(1) Internet - vznik

Internet je informační prostor

1.1 Historie

- Účelem je poskytování služeb založených na výměně zpráv a sdílení prostředků
- Počátky v USA (studená válka, ~1963)
- Strategický záměr - zajištění kontinuity státní správy po nukleárním útoku
- RAND Corporation předložila řešení: Komunikační síť je od začátku považována za nespolehlivou a je navržena tak, aby toto dokázala překonat
- Všechny uzly sítě jsou si rovnocenné (peer-to-peer), každý představuje autoritu pro vytváření, předávání a přijímání zpráv
- Zpráva je rozdělena na samostatně přenášené elementární části (pakety), obsahující kompletní sadu informací potřebnou k doručení paketu do cíle
- **Paket je blok dat se kterým se z hlediska směrování manipuluje jako s celkem.**
- Konkrétní cesta po které se pakety dostávají do cíle je nepodstatná => nová technologie přepojování paketů. Umožňuje aby se zpráva dostala k cíli i v případě, že je část sítě nefunkční

Internet je silně decentralizovaná, robustní síť založena na přepojování paketů (packet switching) ve variantě datagramové služby.

Internet se dále vyvíjel pod záštitou organizace *ARPA* přidělující některým vysokým školám granty na vývin sítě *ARPANET*. Funkční se síť stala v roce 1969 a účelem propojení bylo sdílení dostupu k univerzitním výpočetním kapacitám (superpočítače). Vlastní uzel byl realizován (školním) počítačem naprogramovaným tak, že plnil funkci IMP (Interface Message Processor). Pro vzájemnou komunikaci používali uzly IMP pevné okruhy o rychlosti **50kbps** a přenosový protokol NCP. V roce 1971 měla síť **15** uzlů, v roce 1972 již **37**. Původní záměr (přístup k superpočítačům) však ustoupil do pozadí, protože uživatelé síť začali využívat primárně na komunikaci.

(2) Internet - adresy

Pokud v tomto prostoru potřebujeme něco najít, musíme vědět, kde je to uloženo - potřebujeme znát cílovou adresu. Adresa musí být koncipována tak, aby identifikovala cíl nějakým použitelným způsobem. Například každá síťová karta má MAC adresu. To je unikátní číslo, které ji jednoznačně identifikuje. Tato adresa však neříká kde se karta fyzicky nachází.

2.1 Adresace v Internetu - adresy IP

- **IPv4**
  - 32 bitů, čtyři oktety zapsané v desítkové soustavě, od roku 1983
  - Obecně má adresa dvě části - adresu sítě a uzlu
  - Adresy mohou být individuální (unicast), skupinové (multicast) a všeobecné (anycast)
  - Třída adresy určuje kolik bitů z adresy tvoří adresu sítě
  - Je definováno 5 tříd adres: A, B, C a D, E
    - Třída A má nejvyšší bit prvého bajtu 0 a zbylých 7 bitů je adresa sítě (126 sítí, každá 2^24 adres
    - Třída B má nejvyšší bit prvního bajtu 10 a dalších 6+8 bitů je adresa sítě
    - Třída C má v prvním bajtu 110 a dalších 5+16 bitů je adresa sítě
- **IPv6** *(r. 1995, RFC 1883)*
  - 128 bitů, reprezentováno hexadecimálním řetězcem znaků

Síťová maska

- Čtyřbajtové číslo, binární jedničky určují adresu sítě ve které je stanice s danou IP adresou

- Od roku 1993 platí nové specifikace *(RFC 1517 a 1520)*

Speciální IP adresy:

- Programová smyčka (loopback): 127.0.0.1
- Třída D slouží pro skupinovou adresaci
- Třídy E slouží pro rezervní a experimentální účely

Autonomní systém:

- Vydělená (regionálně/kontinuálně) část internetu
- Správci autonomních systémů:
  - RIPE (Evropa)
  - ARIN (Severní Amerika)
  - APNIC (Asie)
  - AFNIC (Afrika)
  - LACNIC (Latinská Amerika)
- Každý má svůj interval adres, což je základní podmínka směrování datagramů
- Interval adres je možné agregovat na jednu adresu supersítě

Příklad: Jak určit adresu sítě, na které leží počítač o IP adrese 170.85.255.24

Nejprve se podíváme do tabulky tříd a zjistíme, že naše adresa je třídy B, užijeme standardní masku a provedeme logickou operaci *adresa AND maska*

Popis	Desítkově				Binárně			
Adresa	170	85	255	24	10101010	01010101	11111111	00011000
Maska	255	255	0	0	11111111	11111111	00000000	00000000
Síť	170	85	0	0	10101010	01010101	00000000	00000000

Maska nemusí být vyrovnána na hranici oktetu, směrování nemusí být založeno na třídách.

Realizace beztřídního směrování vyžaduje, aby byla IP adresa zadána ve tvaru např. 192.168.0.0/21 (maska je 21 souvislých jedniček). Standardní síťové masky se používaly do roku 1993, změna z důvodu velké nehospodárnosti.

2.2 Přidělování adres

Adresy se přidělují pomocí delegátů. Koncový uživatel dostane IP adresu přidělenou od svého ISP.

(3) Internet – Identifikace

3.1 URI – Uniform Resource Identifier

Jednoduchý způsob určení objektu v informačním prostoru.

Základní vlastnosti:

- Dovoluje určit rozdílné typy objektů, ke kterým se přistupuje odlišným způsobem
- Není žádné omezení pro typ identifikovaného objektu - může to být elektronický dokument, obraz, zvuková nahrávka, informační zdroj a dokonce se k němu nemusí ani přistupovat přes Internet (může to být např. knížka v knihovně)
- Pouze identifikuje objekt, neříká nic o jeho dostupnosti
- Jedná se o hierarchickou sekvenci komponent: *scheme, authority, path, query a fragment*
- Obecně se skládá ze *schématu* a na něm závislé *specifické části*
- **IRI** - Mezinárodní verze URI (Internationalized Resource Identifier) => rozšiřuje URI o možnost zápisu v UTF-8

Použití URI:

- URI reference je řetězec znaků reprezentující URI a potažmo předmět zájmu
- Absolutní reference obsahuje schéma i specifickou část, relativní pouze specifickou, na schématu závislou část
- Použití relativní reference předpokládá existenci báze – kombinací báze a relativní reference získáme absolutní referenci
- Báze může být odvozena implicitně nebo definována explicitně
- Získání předmětu zájmu předpokládá rozřešení odkazu – dereference (URI resolution)

Příklad URI

http://admin:heslo@www.techbox.cz:80/clanek.php?page=3&rubric=31

Scheme (schéma)

- Reprezentuje určitou síťovou službu
- Určuje syntaxi a sémantiku identifikátoru URI
- Nedefinuje přístupový protokol
- Přiděluje IANA

Authority (autorita)

- Reprezentuje jmenný prostor, kde se zdroj nachází
- Jmenný prostor může být vyjádřen registrovaným jménem nebo IP adresou serveru
- Pole začíná `//` a končí `/` nebo `?` nebo `#`
- Má tři části:
  - userinfo – musí být ukončeno znakem @
  - host – IP adresa v hranatých závorkách nebo jméno
  - port – odděluje se dvojtečkou

Path (cesta)

- Pole začíná znakem `/` a končí ? Nebo #
- Hierarchická sekvence segmentů oddělená lomítky
- Může být relativní nebo absolutní
- Relativní cesta může obsahovat znaky `.` a `..`
- Při relativní cestě musí být definována báze

Query (dotaz)

- Dotaz jsou data, která nemají hierarchickou strukturu
- Pole začíná znakem `?` a končí `#`
- Vetšinou má strukturu KLÍČ = HODNOTA
- Obvykle se realizuje dotazem do databáze

Fragment

- Nepřímo identifikuje sekundární zdroj
- Pole začíná znakem `#` a končí koncem URI
- Může představovat část primárního zdroje nebo jiný způsob prezentace primárního zdroje

3.2 URL (Uniform Resource Locator)

- Zastaralý podtyp URI
- Popisuje konkrétní umístění objektu zájmu
- Lokátor začínající schématem je absolutní – obsahuje úplnou informaci o umístění objektu
- Relativní lokátor obsahuje pouze cestu (path), dotaz a fragment
  - nemá *schéma* - vždy se posuzuje v kontextu
  - umístění objektu získáváme kombinací relativního URL a báze
  - *báze* – levá část absolutního URI (schéma + autorita)

3.3 URN (Uniform Resource Name)

Obecnější mechanizmus identifikace objektu zájmu, cílem je jednoznačné jméno.

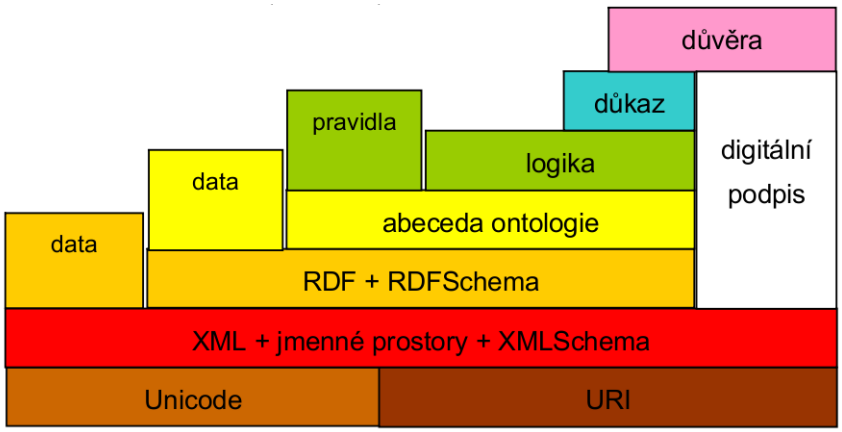
- Technicky zatím nerealizovatelné
- Řeší základní nedostatek URI – situaci, kdy odkazuje na neexistující (přemístěný/smazaný) zdroj
- Použití URI předpokládá vytvoření mechanizmu překladu jména na lokátor (aktuálního) místa uložení (popř. i způsob přístupu)

(4) Sémantický web

Cílem je sémantický informační prostor – přidáním metadat, která specifikují sémantický obsah objektu.

První generace webu byla založena na ručně psaných hypertextových stránkách. Druhá generace poskytuje programově generované (dynamické) služby. Společné pro obě generace je, že procesy vyhledávání a třídění informací jsou stále řízeny člověkem. Současný Internet je charakterizován jako druhá generace.

4.1 Struktura sémantického webu



Obr. 1: Vize sémantického webu

Účel: Učinit zde dostupná data srozumitelná strojům.

Princip: Vývoj standardů, které umožňují systémům najít spojitosti mezi distribuovanými daty.

Cílem je doplnit síť stránek sítí výroků, které lze automatizovaně zpracovávat, zapisovat v RDF . Výsledkem by měla být síť znalostí namísto současné sítě stránek.

Existuje zde začarovaný kruh. Autoři stránek nemají motivaci značky uvádět, protože vyhledávače s nimi nepracují a vyhledávače nepracují se značkami protože je autoři neuvádějí.

4.2 Základní požadavky pro zlepšení

1) **Inteligentní informační služby**
2) **Univerzální vyjadřování**
   – práce s jakoukoliv formou dat
   – nová metoda - Fuzzy logika
3) **Syntaktická interoperabilita softwarových komponent (jednotné API)**
   – nutná podmínka pro komunikaci stroj – stroj i člověk – stroj
   – nikoli uspořádání pomocí syntaktických pravidel nebo uspořádání do katalogu
4) **Sémantická interoperabilita zdrojů**
   – definice vztahů mezi pojmy pomocí metadat
   – sémantická organizace webu
   – automatizované sbírání znalostí

Problémy vyhledávání obecně

- Rozhraní je u každého vyhledávajícího stroje jiné
- Stroje nerozumí obsahu příkazů – hledá pomocí řetězců znaků
- Vyhledávání v textových souborech
- Stroje si nerozumí mezi sebou

Možnosti vyhledávání

- **Fulltext**
  - hledání v textových souborech
  - zpracování obrovského množství dokumentů vyhledávacím robotem
  - indexování dokumentů
  - hledání podle klíčových slov
- **Katalog**
  - utříděný seznam položek

Problémy současných webových vyhledávačů:

- Vysoký koeficient úplnosti, malý koeficient přesnosti
- Nízký koeficient úplnosti (existuje – nenalezeno)
- Výsledky jsou závislé na abecedě modelu (terminologie)
- Výsledkem jsou celé stránky (integrace/rozsáhlost)
- Udávají místo uložení nikoli informaci samotnou (nalezne jen místo kde by informace měla být)
- Relevance odkazů (mám na mysli něco jiného)

Modely vyhledávání

- **Boolean**
  - Rozlišuje pouze to, zda se v dokumentu hledané slovo vyskytuje nebo ne. Stavy pouze 0 a 1.
- **Vector**
  - Uvažuje i to, jak moc se hledané slovo v dokumentu vyskytuje. To vyžaduje mít jednak mechanismus, jak tu váhu zjistit a pak jak z toho něco vypočítat (jedná se v podstatě o page-rank)
  - Souřadnice v n-tém rozměru říká, jak moc je dokument pro n-té slovo relevantní.

XML

- Formát pro ukládání strukturovaného textu
- Vyvinuto konsorciem W3C
- Dokument XML má logickou a fyzickou strukturu
  - Logická – pojmenované jednotky – elementy
  - Fyzická – samostatné části dokumentu – entity

RDF (Resource Description Framework)

- Technologická základna pro reprezentaci metadat (tzv. metadata model)
- Vyvinuto W3C (existují i jiné snahy, DARPA)
- Je založen na výrocích o zdrojích ve tvaru **subject-predicate-object** *(předmět-výrok-objekt)*
  - Předmět je zdroj (odkazuje na něj URI)
  - Predikát (výrok) je binární relace mezi zdrojem a hodnotou (vlastnost)
  - Objekt specifikuje pro nějaký předmět hodnotu vlastnosti
- Příklad: Výrok „Obloha má modrou barvu“ zapíšeme jako trojici řetězců. Předmět je „obloha“, výrok „má barvu“ a objekt je „modrá“.

(5) Internet - služby

5.1 Multipurpose Internet Mail Extensions (MIME)

- Původně vyvinuto pro e-mail kvůli kódování, postupně převzato do dalších protokolů (http)
- Protokol SMTP garantuje pouze přenos 7 bitových znaků (čisté ASCII)
- Úkolem konvence MIME je určení, jakého typu jsou doručovaná data (např. text/javascript)
- MIME řeší dva problémy
  - Jak vytvořit ze zprávy, která obsahuje binární data, zprávu přepravitelnou používanými přenosovými protokoly
  - Jak rozlišit jednotlivé druhy zpráv (zavádí klasifikaci zpráv), právě tato vlastnost se ukázala jako velmi užitečná a je používána i v jiných protokolech
- *Popsáno v dokumentech RFC2029 – RFC2045*

Hlavičky MIME

- **MIME-Version** (Formát: *MIME-Version: 1.0*)
  - indikuje, že zpráva je sestavena podle konvence MIME
  - je povinná a existuje z důvodu zachování kompatibility
  - může být dlouhá maximálně jeden řádek – 72 znaků
- **Content-Type** (Formát: *Content-Type: typ/podtyp; parametry*)
  - definuje typ a podtyp dat obsažených v těle zprávy
  - typ může být jednoduchý nebo kompozitní (složený) a určuje o jaká data se jedná
    - 6 jednoduchých typů – text, application, image, audio, video, model
    - 3 kompozitní typy – message, multipart, report
  - podtyp udává konkrétní formát dat
- **Content-Transfer-Encoding**
  - specifikuje mechanizmus převodu obecných dat na čistý ASCII kód
  - dva převodní mechanizmy: Quoted-Printable, Base64
  - celkem 6 možností – quoted-printable, base64, 7bit, 8bit (včetně nových řádků), binary a x-rozšíření (experimentální kódování)
- **Content-ID**
  - volitelná hlavička jednoznačně identifikující zprávu
- **Content-Description**
  - volitelná hlavička popisující popisné informace o přenášené zprávě (např. název obrázku přenášeného v těle zprávy)
- **Content-Disposition**
  - určuje zda jsou přenášená data určená k automatickému zobrazení (inline) nebo je má příjemce zpracovat ručně (attachment)

Hlavičky MIME mohou obsahovat ještě další informace, např. jméno souboru, datum vytvoření, čtení a modifikace, velikost souboru)

Standardní kódovací mechanizmy

- **Quoted-Printable**
  - poskytuje člověku výsledný text ve srozumitelné formě
  - transformace se provádí následovně – ASCII znaky zůstanou beze změny, ostatní se nahradí znakem „=“ a hexadecimální hodnotou nahrazovaného znaku
  - zakódovaný řádek musí mít maximálně 72 znaků, pokud nemá, vloží se =CRLF (měkký konec řádku)
- **Base64**
  - používá kódovací tabulku o 64 znacích (6 bitů)
  - zpráva je zpracována jako proud bitů, který se rozdělí po šesti bitech a podle tabulky Base64 zakóduje (tabulka je jednoduchá 0=A, 1=B, 2=C …)

5.2 Telnet

Umožňuje připojení ke vzdálenému síťovému uzlu prostřednictvím protokolu TCP/IP.

V dokumentu RFC855 jsou definovány tři základní služby telnetu

- Virtuální síťový terminál – představuje standardní rozhraní vůči vzdáleným systémům
- Schopnost vyjednávání o nastavení určitých voleb
- Symetrické zobrazení terminálu a procesu

Služba pracuje se spojením TCP na portu 23 a přenáší mezi vzdáleným a lokálním systémem osmibitové znaky. Je to velice jednoduchá, platformě nezávislá služba – pouze **otevřené spojení na program pro interpretaci příkazové řádky**.

Po spuštění relace telnetu se aplikace na počítači uživatele stává klientem. Ten naváže spojení TCP se vzdáleným Telnet serverem pomocí standardního 3-way TCP handshake. Klient komunikuje pomocí klávesnice a na svém monitoru vidí reakce vzdáleného systému. Server využívá normální terminálové rozhraní pro přístup k programu telnet, který přenáší data do jiného systému. Komunikace má vysokou režii, každý stisk klávesy generuje znak, který jako datagram putuje mezi systémy. Nicméně, objem komunikace je minimální, proto to zásadně nevadí. Ač telnet představuje málo propracovanou službu (nulové možnosti programování a nízká úroveň konfigurovatelnosti), stále se používá především k odlaďování.

Síťový virtuální terminál zajišťuje transparentnost všech operací prováděných uživatelem. Je to imaginární zařízení, které převede konkrétní zařízení na kterém uživatel pracuje, na standardní typ (emuluje určitý terminál, např. VT-100). Definovaný formát NVT používá sedmibitový ASCII kód pro znaky a zobrazovací zařízení (128 kódů, 95 tisknutelných a 33 řídících). Přenášejí se jako osmibitové položky (nejvýznamnější bit je nastaven na nulu). Právě proto může telnet pracovat pod různými OS (potlačuje heterogenitu zařízení a systémů). Řídící příkazy NVT upravují interakci mezi klientem a serverem a jsou začleněny do proudu dat. Příkaz je sekvence dvou nebo tří oktetů. Prvý oktet je indikátor příkazu, druhý pak vlastní příkaz a třetí upřesňuje význam. Příkaz je uveden znakem IAC (Interpret As Command, 255 desítkově). Řídících znaků je zhruba 20.

Volby telnetu

Nejdříve ze všeho probíhá tzv. vyjednávání o volbách, kde se dohodnou parametry komunikace. Obě strany jsou rovnoprávné. Pokud požadavek nemůže být splněn, je zamítnut. Voleb celkem je asi 40 (záleží na implementaci).

Existují čtyři volby

- **WILL** *(251)* – odesílatel chce danou volbu zapnout
- **DO** *(253)* – odesílatel chce, aby příjemce danou volbu zapnul
- **WONT** *(252)* – odesílatel chce danou volbu vypnout
- **DONT** *(254)* – odesílatel chce, aby příjemce danou volbu vypnul

Příklad: Chceme zapnout volbu **Echo** – 255*(IAC)* 251*(WILL)* 1*(ECHO)*
Příjemce odpoví

- **ANO:** 255*(IAC)* 253*(DO)* 1*(ECHO)*
- **NE:** 255*(IAC)* 254*(DONT)* 1*(ECHO)*

5.3 FTP (File Transfer Protocol)

TFTP

Původní návrh předpokládající, implementaci ve firmwaru (jako Boot-ROM) v bezdiskových síťových stanicích. Kromě rychlého a jednoduchého přenosu nenabízí nic dalšího.

FTP

- Poskytuje služby pro přenos souborů mezi vzdálenými systémy
- Používá TCP spojení, pro přístup ke vzdálenému systému je nutné přihlášení
- Jeden z nejstarších internetových protokolů (počátky v roce 1971)
- Dnešní podoba definována dokumentem *RFC 959*
- Heterogenita systémů mezi kterými se přenáší je řešena podporou několika typů souborů a struktur
- Otevírá dvě TCP spojení, jedno řídící na portu 21 a jedno pro samotný přenos na portu 20 (u aktivního spojení)

Relace FTP – přenesení souboru z jednoho systému na druhý

Předpokládá spolupráci 5ti komponent:

1) Uživatelské rozhraní klienta (řízení interpretu příkazů protokolu)
2) Interpret protokolu na straně klienta (předávání příkazů serveru a řízení přenosu na straně klienta)
3) Interpret protokolu na straně serveru (reakce na příkazy klienta a řízení přenosu na straně serveru)
4) Klientský proces přenosu dat (spojka mezi vzdáleným serverem a místním souborovým systémem)
5) Serverový proces přenosu dat (spojka mezi klientem a souborovým systémem serveru)

Interpret serveru naslouchá na známém portu (21) na řídící příkazy. Řídící spojení zůstává aktivní po celou dobu přenosu. Datové spojení na známém portu (20 v případě aktivního spojení). Je aktivní pouze po dobu přenosu.

Komunikující strany musí dále vyjednat způsob vyjádření předávaných dat (4+3 způsoby), jejich uložení a režim přenosu.

Reprezentace dat – je dána jednak způsobem kódování znaků a formátem.

Formátem se rozumí vertikální formát (stránkování). Řízení formátu má smysl jen u textových souborů (kódovaných ASCII a EBCDIC). Existují tři možnosti:

1) Neurčeno pro tisk (výchozí)
2) Formát Telnet
3) Formát Fortran

Přenášená data mohou být kódována čtyřmi možnými způsoby:

1) ASCII (výchozí)
2) EBCDIC (sálové počítače)
3) image (pro systémy stejného typu)
4) místní typ (data o různé délce bajtu, která musí zůstat zachována)

Datové struktury – přenášené soubory mají jistou vnitřní strukturu, která se přenosem nesmí změnit. Procesy přenosu (na obou stranách), zodpovídají za správné mapování mezi přenášenými a místními strukturami. Jsou možné tři struktury:

1) soubor - není interně nijak členěno, přenáší se jako proud bajtů
2) záznamy – pouze textové soubory
3) stránky – nezávisle číslované stránky (zastaralé, nepoužívá se)

Režimy přenosu – určují jakým způsobem jsou data přenášena. Tři možné způsoby:

1) proud bajtů (výchozí), konec přenosu dán znakem EOF
2) blok, série hlavičkou uvedených bloků
3) komprimovaný režim (používá se jen velmi zřídka)

Příkazy FTP

Řetězce ASCII znaků zasílané po řídícím kanálu. Jsou uvedeny třemi nebo čtyřmi velkými písmeny a ukončeny znaky CRLF.

Je jich zhruba 30, dělí se do tří skupin:

1) **řízení přístupu** – určují kdo může přistupovat k čemu, nejpoužívanější příkazy: USER – název uživatele, PASS – přístupové heslo, CDUP – přechod do nadřazeného adresáře, CWD – změna pracovního adresáře
2) **nastavení parametru přenosu** – nejpoužívanější jsou: MODE – režim přenosu (S,B,C – proud, blok, komprimováno), STRU, TYPE
3) **služby** (např. vytvoření složky, zrušení souboru)

Odpovědi na příkazy FTP

Příkazy mohou být generovány ve skupinách a dojde-li k chybě, je třeba opakovat celou sekvenci příkazů. Odpovědi jsou trojmístná čísla, za kterými může následovat textová oblast čímž je zajištěno, že jsou odpovědi srozumitelné i člověku.

Formát odpovědi: *trojmístné číslo, mezera, textová část, CRLF*

Odpovědní kód (trojmístné číslo) je kódován takto:

První číslice

- 1 – kladná nebo záporná předběžná odpověď (čeká se na další příkazy)
- 2 – kladná konečná odpověď
- 3 – kladná dočasná odpověď (nutný další příkaz)
- 4 – záporná dočasná odpověď (příkaz nebyl dokončen, opakujte)
- 5 – záporná trvalá odpověď

Druhá číslice – specifikuje čeho se odpověď týká

- 0 – chyba syntaxe
- 1 – informace
- 2 – stav spojení
- 3 – ověřování a evidence
- 4 – nezadáno
- 5 – stav souborového systému

Třetí číslice – upřesňuje stav

- 200 – příkaz O.K.
- 331 – uživatel potvrzen (zadejte heslo)
- 500 – chyba syntaxe (kombinací je mnoho)

TFTP

Původní verze protokolu pracuje s bloky pevné délky (512 bajtů) a používá UDP spojení na portu 69 *(RFC 1350)*. Nemá žádné příznaky o doručení nebo zabezpečení. Přenos je iniciován klientem, který otevře port proměnného čísla a zašle žádost na známý port 69. Odeslaný datagram definuje požadavek a specifikaci souboru. Server požadavku přiřadí nový UDP port a zahájí přenos. Soubor je přenesen jako kontinuální proud datagramů o pevné délce. Závěrečný blok dat je kratší, což signalizuje ukončení přenosu. Příjemce potvrzuje přijetí každého bloku (ACK) – další blok nebude odeslán, dokud nebude doručení předcházejícího potvrzeno (lock-step). Bloky jsou číslovány vzestupně, počínaje od jedničky. Dojde-li při přenosu k jakékoliv chybě je přenos ukončen a musí se celý opakovat. Chybu signalizuje chybový datagram, který specifikuje chybu a ukončuje spojení.

Protokol rozlišuje tři typy chybových stavů:

1) Požadavek nelze obsloužit
2) Zpožděný nebo duplikovaný datagram
3) Během přenosu došlo ke ztrátě přístupu k některému zdroji (chybových kódů je celkem 7)

Později byl původní protokol rozšířen. V úvodní fázi komunikace je možné zapnutí nebo vypnutí určitých voleb (v praxi pouze jedna – velikost bloku). Rozšíření je však natolik pružné, že výrobce může zavádět vlastní volby.

5.4 SMTP (Simple Mail Transfer Protocol)

Formát zprávy

Skládá se ze záhlaví zprávy a těla zprávy. Jsou od sebe odděleny jedním prázdným řádkem. Každá hlavička v záhlaví je uvedena klíčovým slovem začínajícím na první pozici řádku, za klíčovým slovem následuje dvojtečka a parametry.

Tělo zprávy je kódováno v US-ASCII a řádek nesmí být delší než 1000 znaků. Maximální velikost zprávy je taktéž dána.

Některé znaky mají speciální význam:

- Středník a dvojtečka jsou oddělovače v seznamu
- Špičaté závorky použité v adrese znamenají, že se použije pouze to, co je mezi nimi (zbytek se ignoruje)
- Kulaté závorky slouží k označení komentáře
- Hranaté závorky znamenají, že není nutný překlad jména

Celkem existuje asi 16 hlaviček, zvláštní postavení má hlavička RECEIVED, která je připisována do záhlaví každým relay serverem, kterým zpráva prošla. Hlavička RESENT se přidá při strojové odpovědi (např. oznámení o nedoručení zprávy).

5.5 POP3 (Post Office Protocol)

Jednoduchý protokol pro manipulaci s obsahem lokální poštovní schránky. Příkazy se zadávají jako US-ASCII kódovaná klíčová slova. Klient otevírá komunikační kanál na TCP portu 110. Následuje autentizace klienta jménem a heslem, která je na rozdíl od SMTP vyžadována vždy. Pokud tato skončí úspěšně, přejde se do transakční fáze, kdy klient může pracovat se svými zprávami ve své poštovní schránce. V jeden okamžik se může připojit pouze jeden klient z toho důvodu, že po přihlášení se celý obsah schránky zkopíruje na lokální médium a po ukončení se relace klient vyvolá synchronizaci.

5.6 IMAP (Internet Message Access Protocol)

Vylepšený protokol pro práci s poštovními schránkami. Popsán v RFC 1730, poslední verze (4) pak RFC 2060.

Umožňuje současný přístup ke schránce z více lokací. Ačkoliv umožňuje uložení lokální kopie zprávy, tato je považována za dočasnou – autoritou je obsah schránky uložený na serveru a proti této autoritě se vždy synchronizuje. Pokud je schránka otevřena druhým klientem je status prvnímu klientovi změněn na read-only.

Využívá se TCP portu 143, komunikace probíhá příkazy kódovanými v US-ASCII (jsou však odlišné od POP3 – mají zcela jinou filozofii). Díky tomu, že klient každý příkaz označuje identifikátorem, který server posílá zpět v těle odpovědi mohou příkazy přicházet ve skupinách a odpovědi na ně mimo pořadí.

Server posílá dva druhy odpovědí:

1) neoznačené odpovědi (mají hvězdičku na místě označení odpovědi), které nesou požadované odpovědi

2) označené odpovědi, které začínají označením příkazu na který se odpovídá sdělují jak plnění příkazu dopadlo

Průběh komunikace probíhá zhruba následovně. Po navázání spojení na TCP portu 143 nastane neautentizovaný stav a klient se musí serveru přihlásit. Autentizace může proběhnout pomocí jména a hesla nebo jinou metodou pomocí příkazu AUTHENTICATE (pokud je metoda podporována, server odpoví znakem plus). Autentizační data jsou kódována v Base64. Po dokončení identifikace je možné posílat příkazy *(např. CREATE – vytvoření schránky, LIST – seznam adresáře, STATUS – informace o schránce, SELECT – otevření schránku, SEARCH – hledání).*

5.7 Hypertext

Seskupení textů, které jsou mezi sebou propojeny pomocí odkazů (linků). To umožňuje čtenáři procházet textem nelineárně. Bez jasně definovaného konce nebo začátku.

Princip poprvé představil Vannevar Bush v roce 1945 v článku „As We May Think“, v němž popisuje zařízení Memex (Memory Extender), umožňující ukládání informací a jejich vzájemné propojování. Samotný termín hypertext zavedl až v roce 1965 Theodor Nelson, který je rovněž autorem první počítačové realizace (informační katalog Xanadu).

Na výše zmíněný princip si vzpomněl až Tim Berners-Lee, když měl ve švýcarském CERNu vymyslet způsob, jak by mohlo společenství fyziků jednoduše a efektivně sdílet informace. Vyšel z myšlenky hypertextu a lineární texty rozdělil na dokumenty, které vzájemně provázal odkazy – vazbami, které začínají na určitém místě dokumentu a směřují na jiné místo v tom samém nebo úplně jiném dokumentu.

5.8 WWW

Jedná se o službu jejímž základem je Hypertext Transfer Protocol (HTTP) definující pravidla komunikace mezi klientem a serverem.

Mezi základní vlastnosti patří: architektura client/server, TCP spojení (port 80), stateless (jednotlivé relace jsou nezávislé, relace je transakce dotaz-odpověď), podpora proxy.

Vše začíná akcí uživatele, podle které se určí URI zdroje (jaký protokol použít, jaký server kontaktovat atd.). Server obdrží dotaz a odpoví na něj, čím pro něj skončila jedna ucelená http transakce. Stavové informace (např. URL stránky) si musí pamatovat klient.

5.9 Hypertext Transfer Protocol (http)

Je to protokol aplikační vrstvy sloužící ke komunikaci mezi klientem a WWW serverem. Specifikuje podobu přenášených dat, pravidla dotazů a odpovědí komunikujících stran.

Klient naváže spojení na známém portu 80 a odešle dotaz jako prostý textový řetězec. Server jej zpracuje a reaguje na něj příslušnou odpovědí.

První verze protokolu (v0.9) byla velmi primitivní – umožňovala v podstatě jen prostý přenos dat ze serveru ke klientovi a v současnosti je zastaralá. Mezi základní nedostatky této verze patří:

1) špatná identifikace typu přenášených dat
2) klient nezná objem přenášených dat – jinak musí klient číst, dokud server neukončí spojení
3) problematická efektivita – dáno absencí mechanizmu zjištění poslední aktualizace zdroje

Verze 1.0 *(RFC 1945)* přinesla podporu MIME (možnost doplnit dokument o metadata). Nyní aktuální verze 1.1 *(RFC 2616)* přinesla zvýšení efektivity protokolu (trvalé spojení a řetězení dotazů), vylepšenou podporu neanglických jazyků a virtuální servery, díky kterým je možné definovat více WWW serverů pro jednu IP adresu.

HTTP relace

Každý **dotaz** obsahuje dotazový řádek, záhlaví (hlavičky následované prázdným řádkem) a tělo dotazu (může být a většinou je prázdné). Dotazový řádek má tvar: *metoda cesta_k_dokumentu HTTP/1.1*. **Odpovědi** jsou zasílány se stavovým řádkem, záhlavím (hlavičkami a prázdným řádkem) a tělem odpovědi. Stavový řádek odpovědi má tvar: *HTTP/1.1 stavový_kód stavový_text*.

Metoda je druh služby, kterou klient po serveru vyžaduje. Server nemusí všechny metody podporovat a při dotazu na nepodporovanou metodu odpoví chybou. Metod je celkem osm (běžně používáno jen šest).

1) **OPTIONS** – dotaz na komunikační možnosti serveru nebo požadovaného URI. Umožňuje klientovi určit možnosti serveru
2) **GET** – požadavek na získání informací uložených na serveru
3) **HEAD** – metoda podobná GET, ale server posílá jen stavový řádek a záhlaví (neposílá tělo zprávy), většinou se používá k získání doplňkových informací o zdroji
4) **POST** – slouží k uložení dat na server (například pole formuláře), je použit v případě kdy má cílový server přijmout data z požadavku. Skutečná funkce závisí na URI. Formát posílaných dat není nijak omezen a může být pomocí hlaviček popsán v záhlaví.
5) **PUT** – používá se velmi zřídka, definuje požadavek uložení posílaných dat na specifikované URI přičemž uložená data budou dostupná přes dotazy GET
6) **DELETE** – zruší zdroj, na který ukazuje URI (logicky se nepoužívá)
7) **TRACE** – metoda sloužící k testování serveru, je to obdoba tracepath

5.10 Proxy

Proxy server je server, který obsluhuje požadavky svých klientů přeposíláním jejich požadavků na dalším serverům. Klient se připojuje k proxy serveru a požaduje nějakou službu jako například program, připojení, webovou stránku nebo jiný zdroj dostupný z třetího serveru. Proxy server poskytne zdroj tím, že se sám připojí na daný zdroj a požádá o službu jménem klienta. Volitelně může pozměnit požadavek klienta anebo odpověď serveru a někdy také může obsloužit požadavek bez kontaktování vzdáleného serveru za pomoci informací uložených v cache.

(6) Internet - protokoly

6.1 Protokolová architektura TCP/IP

Rozhraní sítě

Zajišťuje přístup k sdílenému přenosovému médiu. Využívá všechny známé přenosové prostředí a všechny známé typy sítí (LAN, WAN, MAN) pro podporu TCP/IP.

Mezisíťová vrstva

Plní úkoly logické adresace, směrování a předávání datagramů (segmentace, sestavování). Protokol IP poskytuje síťovou službu bez spojení a nezaručuje doručení (spoléhá na protokoly vyšších vrstev). Každý datagram je samostatná jednotka, která musí obsahovat všechny informace potřebné k doručení.

Transportní vrstva

Představuje mechanizmus pro přenos dat mezi dvěma stanicemi. Nabízí službu se spojením (TCP, Transmission Control Protocol) nebo bez spojení (UDP, User Datagram Protocol). Na této vrstvě pracují také některé směrovací protokoly (např. RIP, Routing Information Protocol).

Aplikační vrstva

Obsahuje protokoly dávající uživatelům konkrétní aplikace. Aplikační protokoly jsou většinou závislé na transportní službě (například HTTP, Telnet a FTP užívají TCP).

6.2 Protokol IP

Jeho funkcí je dopravovat datagramy mezi jednotlivými sítěmi. Je tvořen několika dílčími protokoly. Základní je protokol IP a služební protokoly ICMP (signalizace mimořádných stavů) a IGMP (doprava adresných oběžníků). Patří sem i protokoly ARP a RARP avšak jejich rámce nemají standardní hlavičku IP.

+	0-3	4-7	8-15	16-18	19-31
0	Verze	Délka hlavičky	Type of Service	Délka paketu	
32	Identifikace paketu			Fragmentace	Fragment offset
64	TTL		Protokol	Kontrolní součet hlavičky	
96	Zdrojová adresa				
128	Cílová adresa				

Tab. 1: Struktura hlavičky IP datagramu

Protokol ICMP

Protokol řídících hlášení. Záhlaví datagramu je vždy dlouhé 8 bajtů. Prvé čtyři bajty obsahují typ zprávy (8), kód zprávy (8) a kontrolní součet (16). Pole typ je hrubé dělení zpráv (0 – echo, 3 – chyba, 4 – žádost), kód pak upřesňuje (0 – network unreachable atd.).

Protokol IGMP

Protokol správy skupin. Slouží k šíření adresných oběžníků. Skupinové vysílání se používá s cílem redukce nebo optimalizace provozu.

6.3 Protokol TCP

Transmission Control Protocol

+	0-3	4-7	8-15	16-31
0	Zdrojový port			Cílový port
32	Sekvenční číslo (Sequence number)			
64	Číslo potvrzení (Acknowledgment number)			
96	Data offset	Rezervováno	CWR, ECE, URG, ACK, PSH, RST, SYN, FIN	TCP Okno (window)
128	Kontrolní CRC součet			Ukazatel důležitosti (urgent pointer)

Tab. 2: Struktura hlavičky TCP datagramu

TCP protokol pracuje na transportní vrstvě a je se spojením. IP protokol zajišťuje spojení mezi jakýmikoli datovými stanicemi na Internetu a protokol TCP pak spojení mezi příslušnými aplikacemi.

Plní funkce asociace portů se spojeními, navazuje (3-Way handshake) a ukončuje (4W), řídí tok dat (segmentace, číslování, potvrzování příjmu). Portů je 65535, 0-1023 jsou privilegované. Pořadová čísla přenášených bajtů nezačínají od nuly, ale od náhodně vygenerované hodnoty nastavené při sestavování spojení. Velikost okna (window size) určuje kolik dat je možné přenést bez potvrzení.

(7) Překlad adres

Pokud se mezi komunikujícími klienty nachází jeden nebo více routerů pak se pro určení směru cesty použije IP adresa cíle a pro adresu prvního skoku hardwarová adresa příslušného routeru, Směrovač postup zopakuje a pomocí IP adresy zjistí směr k cíli – IP adresu následujícího směrovače (next hop) a pošle rámec na jeho hardwarovou adresu. Toto se cyklicky opakuje dokud paket nedorazí do cíle určení. Je dobré si uvědomit, že se vždy pracuje se dvěma adresami (MAC a IP). Ať jsou oba uzly ve stejné dílčí síti nebo ne, je algoritmus doručení paketu stejný.

7.1 Protokoly pro překlad adres

ARP (Address Resolution Protokol)

Překlad logické (IP) adresy na hardwarovou (MAC) adresu.

RARP (Reverse ARP)

Překlad hardwarové (MAC, linková) adresy na logickou (IP) adresu. Byl navržen pro tenké klienty, při zapnutí totiž stanice nezná nic než svoji MAC adresu a proto vyšle linkový oběžník s dotazem jakou IP má používat – je to zase jistá varianta inicializace síťového uzlu. Dnes již zastaralý (nahrazován DHCP).

Protokoly ARP i RARP jsou nezávislé na IP – komunikují nad linkovou vrstvou.

BOOTP (Bootstrap protocol)

Zastaralé, předchůdce DHCP původně určený pro tenké klienty.

DHCP

Společně s BOOTP slouží k počátečnímu nastavení síťového uzlu.

7.2 ARP

Slouží ke zjištění linkové (MAC) adresy síťového uzlu při znalosti IP adresy. Protokolová data jsou balena přímo do linkového rámce Ethernetu, výsledkem je faktická nezávislost na protokolu IP.

Definován v RFC 826.

Datový formát

Datové pole typ určuje jaký formát je v dílčí síti používán, pole *typ síťového protokolu* specifikuje síťový protokol (IP má 800H), délka linkové adresy (většinou 6 oktetů/bajtů), délka síťové adresy (obvykle 4). Typ operace nabývá hodnot 1 (žádost) nebo 2 (odpověď). Žádost je posílána linkovým oběžníkem.

+	0-7	8-15	16-31
0	Hardware type		Protocol Type
32	Hardware length	Protocol length	Operation
64	Hardwarová adresa odesílatele		
96	Protokolová adresa odesílatele (IP adresa u protokolu IP)		
128	Hardwarová adresa cíle (pole je vyplněno nulami při požadavku)		
196	Protokolová adresa cíle		

Tab. 3: Struktura hlavičky ARP datagramu

Komunikace

Při zjišťování linkové adresy cílového uzlu odesílatel nejdříve prověří svoji ARP tabulku, kde jsou uloženy záznamy o linkových adresách. Tabulka ma dvě části – statickou, která je tvořena ručně vloženými položkami a část dynamickou,

která je výsledkem činnosti protokolu. V případě, že hledaná adresa není nalezena je vyslán ARP dotaz v podobě linkového oběžníku na který odpoví buď přímo cílový uzel nebo implicitní směrovač. Odpověď je zanesena do ARP tabulky (kladná i záporná). Dynamická část je realizována jako cache.

7.3 DHCP

Dynamic Host Configuration Protocol, RFC 2131 a 2132

Primární parametry, které musí uzel znát, chce-li komunikovat jsou **IP adresa** a **maska sítě** což následně umožní zjištění adresy pro všesměrové vysílání (broadcast). Spolu s tím, se nastaví i ostatní související parametry: a**dresa implicitního směrovače** a **adresy DNS serverů** – pokud se tyto parametry nemění, lze stanice nastavit staticky. Pokud existují mobilní uživatelé nebo se často přidávají zařízení je takové řešení nepřijatelné.

DHCP je protokol sloužící k automatizaci procesu konfigurace síťových uzlů. Je to aplikační protokol založený na architektuře client-server. Konfigurace uzlu může být automatická (trvalé přidělení parametrů) nebo dynamická. Dynamická alokace umožní přidělit parametry jen na určitou dobu (lease value) – ideální pro mobilní klienty.

Zprávy DHCP

- Discover (pátrání) – klienti vysílají tento druh při své inicializaci jako oběžník určený všem DHCP serverům v síti
- Offer (nabídka) – server posílá nabídku jako odpověď na pátrací oběžník, kde nabídne sadu konfiguračních parametrů
- Request (žádost) – odesílá klient serveru a očekává potvrzení nebo zamítnutí, posílá se jako odpověď na nabídku, nebo jako žádost o prodloužení pronájmu (lease) parametrů nebo získání nových
- ACK (potvrzení příjmu) – odpověď serveru na žádost klienta
  - kladná nebo záporná (server nemůže požadované parametry konfigurace akceptovat)
  - odmítnutí ze strany klienta – přidělené parametry jsou již používány jiným uzlem
- Release (uvolnění) – klient oznamuje serveru uvolnění přidělených parametrů
- Information – zasílají klienti s pevnou IP, když potřebují získat další parametry

7.4 DNS

Syntaxe

Doménové jméno se uvádí v tečkové notaci. První řetězec je jméno počítače, další je jméno nejnižší vnořené domény. Zcela vpravo je doména nejvyšší úrovně (přesněji, zcela vpravo je root, tečka, která se většinou vynechává). Celé jméno může mít maximálně 255 znaků, řetězec pak 63 znaků. Povolené znaky jsou písmena, číslice a pomlčka, která nesmí být na začátku ani na konci.

Pozn. Autonomní systémy dělí Internet z hlediska směrování, domény podle jmen uzlů.

Historie

v roce 1983 vyvinul Paul Mockapetris DNS protokol, který je popsán v dokumentech *RFC 882 a 883.* V roce 1987 byl protokol aktualizován *(RFC 1034 a 1035)*. Dnes existuje cca 30 RFC dokumentů týkajících se DNS.

Struktura

Je to hierarchický systém jehož realizaci zabezpečují servery DNS. Umožňuje lepší orientaci lidem a udržování decentralizovaných databází doménových jmen a jejich překlad z/na IP adresy.

Každý uzel struktury obsahuje data o své části jména, které je mu přiděleno a odkazy na své subdomény. Root je kořenová doména (zapisuje se jako tečka). Hierarchicky níže se nacházejí domény nejvyšší úrovně (TLD). Ty jsou buď generické (tematické) nebo národní. Celá struktura se administrativně dělí do zón, kde každá ma svého správce a autoritativní server. Výhodou je možnost snadné delegace pravomocí.

Každá zóna ma nejméně dva DNS servery. Primární je server, na kterém data vznikají – pokud je třeba provést nějaké změny, tak se provádějí zde. Sekundární server je kopií primárního (pro případ výpadku). Pomocný (caching only) server slouží jako vyrovnávací paměť pro snížení zátěže a uchovává odpovědi dokud nevyprší jejich platnost. Odpovědi primárního a sekundárního serveru jsou autoritativní, pomocného nikoliv.

Systém se skládá ze tří hlavních komponent:

- soustava hierarchicky uspořádaných **jmenných prostorů** (realizováno jako distribuovaná databáze záznamů)
- **jmenný server**, program který zná strukturu jmenných prostorů a umí manipulovat se záznamy databáze
- **resolver**, program který komunikuje se jmennými servery, od kterých dostává informace

Z hlediska klienta se DNS systém jeví jako statický, z pohledu jmenného serveru je systém dynamický, protože server musí periodicky aktualizovat svá data, která organizuje do zón.

DNS dotaz

Relace přeložení jména na IP adresu (popřípadě naopak).

Relace probíhá následujícím způsobem:

- klient má požadavek na přeložení doménového jména www.yyy.zzz
- resolver pošle dotaz lokálnímu jmennému serveru (LNS) a očekává jednoznačnou odpověď
- LNS zná adresy root serverů, pošle tedy některému dotaz
- root NS odpoví seznamem NS domény yyy
- LNS odešle dotaz NS subdomény yyy, který odešle autoritativní odpověď

DNS funkce

Primární server načtením souboru z lokálního disku získá informace o své zóně. Sekundární server tyto informace získává komunikací s primárním serverem. Tato data jsou označována za autoritativní. Dále se načte root.hint obsahující informace o kořenových serverech.

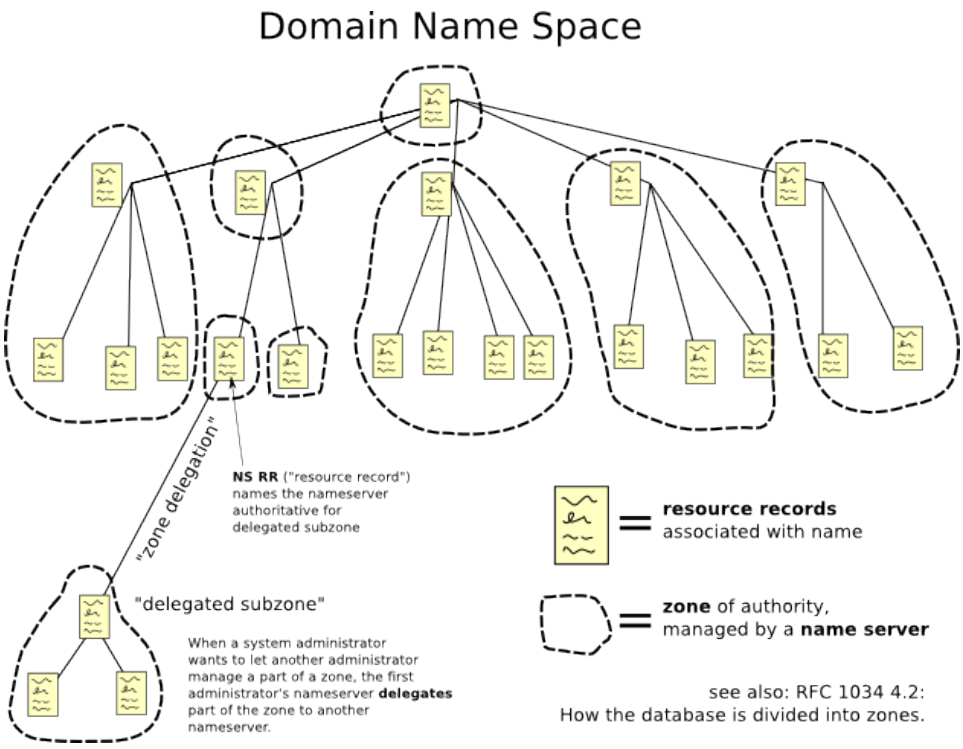
Součástí systému je paměť cache. Do ní se ukládají kladné (případně i záporné, negativní caching) odpovědi na dotazy. Tato data jsou neautoritativní. Pro přenos dotazů se používá protokol UDP, vyšle se datagram prvnímu serveru a pokud nepřijde během velmi krátké doby odpověď, pošle se dotaz dalšímu serveru (to se cyklicky opakuje do získání odpovědi nebo vypršení časového intervalu). Platí ta odpověď která přijde jako první, i když je negativní. Délka UDP datagramu je omezena na 512B a pro zónové přenosy se používá protokol TCP.

DNS záznamy

Doménový jmenný prostor se skládá ze stromu jmen. Každý list má jeden nebo více zdrojových záznamů (resource records), které obsahují informace o každém doménovém jméně (např. fyzický vlastník a jeho adresa)

Typy zdrojových vět (resource record)

- SOA – určuje autoritativní jmenný server zóny (uvozuje data zóny)
- A – přiřazení IP adresy doménovému jménu
- NS – věta definující jmenné servery zóny
- CNAME – uvádí synonymum (alias) k doménovému jménu = umožňuje přiřadit k jednomu jménu více IP adres
- PTR – reverzní záznamy
- AAAA – přiřazení IPv6 adresy
- WKS – popisuje ostatní služby počítače /Well Known Services/
- MX – určuje server elektronické pošty a číselné vyjádření jeho priority, doručuje se podle priority /Mail eXchange/
- TEXT – textové pole, má pouze informativní charakter



Shrnutí

Kořenové servery mají autoritativní informace o doménách nejvyšší úrovně. Dotaz je tedy následně směrován na autoritativní server TLD, v níž se nachází cílové jméno. Ten poskytne informace o v hierarchii nižším autoritativním serveru. Výsledkem je autoritativní odpověď.

DNS protokol

Dotaz (query) má stejný formát jako paketu pro dotaz i odpověď. Skládá se ze záhlaví a až čtyřech dalších sekcí. Záhlaví je povinné a skládá se ze šesti 16bitových polí.

	0-15	16-32
0	ID (slouží k párování dotaz-odpověď)	pole řídících bitů
32	QDCOUNT (počet položek v dotazu)	ANCOUNT (počet zdrojových vět)
64	NSCOUNT (počet autoritativních NS)	ARCOUNT (počet položek v doplňující odpovědi)
96	QNAME	QTYPE
128	QCLASS	TTL, RDLENGHT a RDATA

Řídící bity mají následující význam:

- QR – dotaz/odpověď
- AA – autoritativní odpověď
- TC – zkráceno (odpověď se nevešla do 512 bitů)
- RD – klient požaduje rekurzivní překlad
- RA – server umožňuje rekurzivní překlad

Čtyřbitové pole OPCODE specifikuje typ dotazu (standardní, inverzní, na status serveru) a RCODE charakterizuje odpověď (bez chyby, chyba formátu dotazu, server neumí odpovědět, jméno neexistuje, server nepodporuje dotaz, odmítá odpovědět).

Doménové jméno není zde zapsáno v tečkové notaci, ale každá jeho část je uvedena bajtem, který uvádí délku následujícího řetězce. Konec je signalizován nulovou hodnotou. Pro dosažení minimální délky paketu je doménové jméno komprimováno (uvede se jen poprvé a každé další už jen odkazují).

Paket dotazu se skládá ze záhlaví a sekce dotazu. Paket odpovědi má záhlaví a sekce dotazu, odpovědi a případně sekce autoritativní servery.

(8) Směrování

Účel: Předávání datagramů mezi lokálními sítěmi do místa určení.

Princip: Rozhodování, kudy dál poslat datový paket i při neznalosti celé cesty. Zjistí-li směrovač, že zvolená cesta není optimální, uvědomí o tom odesílatele (většinou směrovač), který upraví svou směrovací tabulku.

Směrovací tabulka je soubor záznamů ve tvaru adresa sítě – adresa následujícího směrovače (next hop)

Dva základní mechanizmy:

- přímé směrování (direct routing) – oba počítače se nacházejí ve stejné LAN síti, řeší většinou uzel samotný
- nepřímé směrování (indirect routing) – kdy se nacházejí v obecně různých sítích, řeší většinou směrovač

Příklad - nepřímé směrování: Uzel A chce poslat paket uzlu B. Rozloží IP adresy pomocí masky a získá adresu cílové sítě. Porovnáním zjistí, že je jiná než jeho vlastní. Paket odešle na směrovač (bránu) své lokální sítě čímž jeho úloha končí a o zbytek se postará směrovač. Každá LAN síť připojená k Internetu (nebo další, větší síti má alespoň jeden směrovač. Pokud je v síti směrovačů více, tak klient v každém případě posílá pakety svému implicitnímu směrovači. Ten může, v případě že je výhodnější paket poslat přes jiný směrovač, odpovědět ICMP zprávou definující jiný implicitní směrovač.

8.1 Historie

V počátcích Internetu představoval původní Arpanet jeho páteřní síť a každá lokální síť se připojovala prostřednictvím jediného směrovače, který plnil funkci default gateway a znal celý Internet. S rozšiřováním sítě nastal problém udržování směrovacích tabulek – řešením jsou *autonomní systémy.*

Současný internet je důsledně rozdělen na autonomní systémy spravované jednotlivými ISP. Každý autonomní systém je označen zkratkou AS a dvoubajtovým číslem. Každá autonomní oblast má svou správu, která žádá autoritu o přidělení intervalu IP adres. Pro směrování mezi AS se používají protokoly EGP nebo BGP a za každý autonomní systém odpovídá jeho provozovatel.

8.2 Směrovací protokoly

Aplikační protokoly sloužící směrovačům k automatickému naplnění směrovacích tabulek.

- Protokoly skupiny IGP (Interior Gateway Protocol)
- EGP slouží pro výměnu směrovacích dat mezi AS

Dělení podle způsobu určení optimální cesty:

- RVP (Routing Vector Protocol) – definuje kvalitu cesty počtem přeskoků, nevýhodou je vyšší režie, prudce rostoucí s počtem směrovačů
- LSP (Link-State Protocol) – testuje v pravidelných intervalech průchodnost cest, má mnohem menší režii a jsou stabilnější, problémem je velká složitost nastavení

8.3 Směrovací tabulky

- Rozhodnutí každého síťového uzlu jsou řízena směrovací tabulkou protokolu IP
- Směrovací tabulka je složena ze záznamů
  - první část udává cílovou adresu, může to být adresa jednotlivého uzlu nebo celé dílčí sítě, součástí je i síťová maska
  - druhá část udává rozhraní
- Akce mohou být dvou typů
  - doručit přímo adresátovi (pokud jsou ve stejné síti)
  - předat sousednímu routeru, který je blíže k adresátovi
- Směrovací rozhodnutí probíhají zcela samostatně pro každý procházející paket
- Uvedená cílová adresa se porovná se směrovací tabulkou, vyberou se záznamy jejichž maska (prefix) odpovídá cílové adrese a konečný záznam s nejdelší (nejpřesnější) maskou (prefixem)
- Směrovací algoritmy lze rozdělit na statické a dynamické

Statické (neadaptivní) směrování

- Pracuje s neměnnou tabulkou (zadána ručně nebo přes DHCP)

- Většina uzlů v dílčích sítích Internetu má statické směrovací tabulky – stačí to, protože většinou obsahují pouze dva záznamy: na adresy ve stejné dílčí síti doručovat přímo a na adresy mimo posílat na výchozí bránu

Dynamické (adaptivní) směrování

- Reaguje na změny v topologii sítě a upravuje směrovací tabulky podle situace
- Dva základní principy
  - hierarchické členění sítě
  - distribuované směrování (dnes je to standardní přístup ke směrování v Internetu)
- Data o změnách se v síti předávají postupně mezi sousedními směrovači
  - představiteli distribuovaných protokolů jsou např. RIP, OSPF nebo BGP
- Objem dat by byl při použití distribuovaného směrování příliš velký což řeší hierarchické členění autonomních systémů do několika relativně samostatných oblastí (area), data o změně topologie se pak šíří jenom zde
- Výměnu souhrnných dat obstarávají hraniční směrovače a každá oblast dostává pouze tato souhrnná data

8.4 Směrovací protokoly

IGP – směrování v rámci autonomních systémů

- **RIP** (Routing information Protocol)
  - používá broadcast, pracuje s vektorem vzdálenosti, určen spíše pro malé sítě
- **OSPF** (Open Shortest Path First)
  - založen na algoritmu pracujícím s kvalitou přenosové cesty (LSA)
  - při rozhodování bere do úvahy
    - šířku přenosového pásma
    - zátěž
    - spolehlivost
    - transportní zpoždění
    - velikost MTU
  - používá skupinové vysílání (multicast, adresa 224.0.0.0/4), spouštěné aktualizace, síťové masky proměnné délky a je schopen podporovat směrování s normovanou kvalitou služby (QoS)
  - podpora ověřování vzdáleného směrovače
- **EIGRP** (Enhanced Interior Gateway Protocol)
  - vyvinut firmou Cisco, kombinuje výhodu obou algoritmů směrování (VDA a LSA)

EGP – výměna souhrnných dat mezi systémy

- **BGP** (Border Gateway Protocol)
  - pracuje s pevně nastavenými pravidly

8.5 Protokol RIP (Routing Information Protocol)

- Nejstarší a nejrozšířenější protokol, existuje ve dvou verzích
  - první verze: RFC 1058, rok 1988
  - druhá verze: RFC 2453, rok 1998
- původní návrh je implementací směrovacího algoritmu Bellman-Ford (Ford-Fulkerson) realizovaný Xeroxem
- Bellman, Ford a Fulkerson položili teoretické základy algoritmů směrování s vektorem vzdálenosti (1962, Princeton)

Základní skutečnosti

- Výhodou je snadnost nastavení a uvedení do provozu
- Nevýhodou, je neschopnost pracovat v rozsáhlých sítích
  - nejvyšší počet přeskoků je 15 (sítě s přeskokem 16 a více jsou považovány za nedostupné
  - se zvětšováním síťové struktury může výměna dat o trasách mezi RIP směrovači výrazně zatížit síť
  - dlouhá doba zotavení – když dojde ke změně v topologii propojených sítí, může úprava trvat až několik minut
  - v rámci automatických úprav mohou vznikat uzavřené směrovací smyčky způsobující nedoručitelnost dat

Směrovací tabulka (lokální databáze tras)

- Ve výchozím stavu obsahuje pouze sítě, ke kterým je směrovač fyzicky připojen (ručně nakonfigurované)
- Tabulka obsahuje
  - adresa cílové sítě
  - metrika (počet směrovačů na cestě k cílové síti resp. *číslo vyjadřující vhodnost cesty k přenosu*)
  - adresa směrovače (adresa rozhraní směrovače – 1. hop)
  - časovač sledující dobu od poslední aktualizace
- Aktualizace probíhá jen s nejbližšími sousedy – směrovač prohlédne získaná data a vybere cílové adresy, které ve své tabulce nemá, zvětší jejich metriku o jedna (+1 hop) a přidá do své tabulky; pokud adresu už v tabulce má, ale nová je kratší (menší metrika) provede výměnu; nepoužité položky zahazuje
- Délka cesty je dána počtem přeskoků (což neříká nic o kvalitě cesty)
- Nedostane-li směrovač aktualizaci pro určitou cestu po dobu 6-ti aktualizací, prohlásí cestu za nedostupnou, nastaví metriku na 16 a zapne časovač (garbage-collection) na vymazání cesty z tabulky
- Omezení metriky na 15 je dáno charakterem protokolu – informace o neplatné cestě se musí rozšířit co nejrychleji jinak může vzniknout smyčka (ke které jsou náchylné všechny protokoly založené na měření počtu přeskoků)
- Problém je právě v postupném šíření směrovacích dat, trvá určitý čas, než se směrovače po aktualizaci sjednotí na stavu sítě, mezitím se neaktualizované směrovače snaží přesvědčit aktualizované, že cesta existuje – předávání chybných dat je signalizováno zvyšováním metriky až dojde na metriku 16, cesta je zneplatněna a směrovače se začnou sjednocovat
- V pravidelných intervalech (~30s) každý směrovač rozesílá obsah své tabulky nezávislé na aktualizacích
- Verze 1 používá všesměrové oběžníky, verze 2 přidává skupinové *(multicast)* oběžníky (adresa 224.0.0.0/4)
- Všechna zařízení naslouchají na UDP portu 520 a aktualizují své tabulky

Problémy a nevýhody protokolu RIP

Struktura sítě se dynamicky mění a trvá nějakou dobu, než se změna lavinovým systémem dostane na všechny směrovače v síti. Nefunkční směrovač neposílá pravidelné aktualizace a v důsledku toho je cesta prohlášena za neplatnou. Důsledkem je dočasná nekonzistence v systému.

Pokud k cíli vede několik tras, může vzniknout směrovací smyčka – na cestě k cíli se paket dostane zase na směrovač, kterým již prošel.

Rozdělení horizontu (split horizon) - určuje, že směrovač nesmí informovat svého souseda o cestách, které vedou přes něj samotného. Důsledek je ten, že se neodesílají celí směrovací tabulky, ale jen položky o cestách, které přes přijímající směrovač nevedou.

Příklad: směrovač A neodešle směrovači B tu položku své směrovací tabulky, která má jako první hop uvedenu adresu směrovače B. Toto pravidlo má zabránit vzniku smyček mezi sousedními směrovači.

Otrávení zpětných dat (poison reverse) – zabraňuje vzniku rozsáhlých smyček. Příznakem rozsáhlé smyčky je postupné navyšování metriky cesty. Tento předpis umožňuje směrovači porušit pravidlo rozdělení horizontu, avšak u předávaných položek (které porušují pravidlo rozdělení horizontu) je však metrika nastavena na 16. Pokud příjemce má ve své tabulce stejnou cestu s lepší metrikou, bude cestu ignorovat.

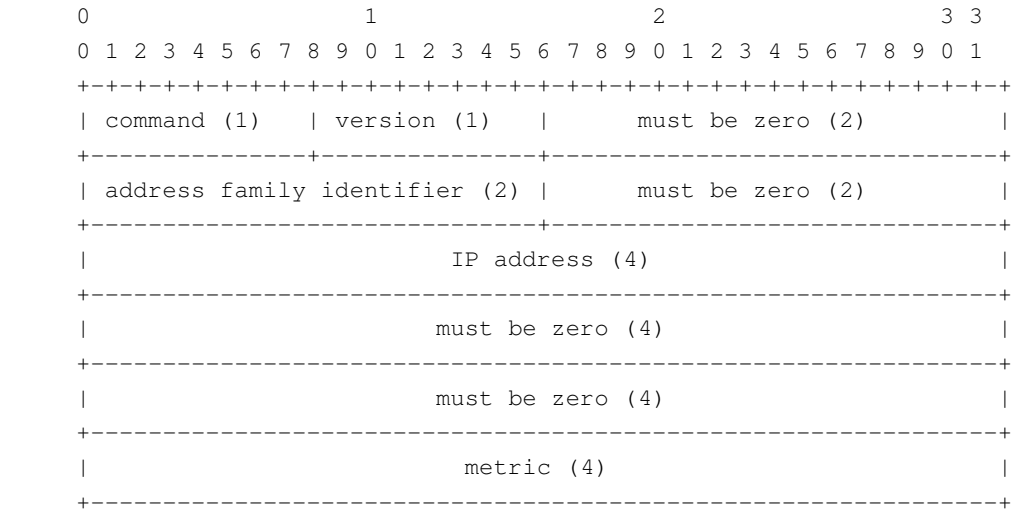
Spouštěné aktualizace (trigger update) – když směrovač zjistí změny v topologii sítě a nečeká na pravidelnou aktualizaci a odešle data, která se změnila.

Zadržovací časovač – při označení cesty za neplatnou trvá ještě dalších 90 vteřin, než je z tabulky vymazána (garbage-collection) a v tomto čase směrovač nereaguje na změny v dané cestě, kromě zprávy, že je cesta v původním stavu.

Cílem těchto úprav, je zabránit šíření nepravdivých směrovacích dat, během konvergence sítě na novou topologii.

Formát paketu RIPv1

viz: http://tools.ietf.org/html/rfc1058



- Prázdná pole jsou pozůstatkem minulost (pole, která ztratila význam)
- V jednom datagramu může být maximálně 25 položek směrovací tabulky
- Význam IP adresy první položky závisí na typu zprávy
  - žádost – IP adresa odesílatele
  - odpověď – jedna z IP adres ze směrovací tabulky odesílatele
- Identifikátor rodiny adres má vždy hodnotu 2 (= protokol IP)
- Pole s číslem verze definuje verzi RIP protokolu

Formát paketu RIPv2

Z důvodu zpětné kompatibility jen minoritní změny. Definován v RFC 1723

- Změny oproti RIPv1
  - prázdného pole za polem IP adresy je maska
  - další prázdné pole je adresa prvního hopu (počátek cesty)
  - pole AFI má dvojí význam
  - značkování externích cest
  - informační bloky
- Hlavní nedostatky RIPv1 oproti v2
  - velmi omezená podpora pro tvorbu podsítí
  - žádná autentizace vysílajícího uzlu
  - metrika omezená na 15 uzlů
  - jen všesměrové vysílání zpráv
- Limit metriky na 15 uzlů je nezměněn (změna by vedla k nekompatibilitě se starší verzí)

Mechanismus autentizace (RIPv2)

- Důvěryhodnost původce směrovacích dat má zásadní bezpečnostní význam, šíření nepravdivých dat z falešných zdrojů způsobí minimálně k poškození směrovacích tabulek nebo hůře, ke kolapsu síťového provozu
- Mechanismus nemění strukturu hlavičky, pouze prvá položka směrovacích dat je použita pro přenos ověřovacích údajů, podle AFI bude má hodnotu FFFFH a pole se značkou cesty se mění na pole vyjadřující typ autentizace
- RFC 1723 definuje pouze jeden způsob autentizace – plaintext password o délce 16 znaků
- Komunikace mezi směrovači není šifrována, což ubírá na robustnosti protokolu – kdokoliv s přístupem k síti může heslo odposlechnout

(9) Organizace

9.1 Internet Society (ISOC)

- Nejvyšší orgán zaštiťující veškeré dění kolem Internetu, podporuje činnosti zaměřené na vývoj a dostupnost Internetu
- Funguje od roku 1992, členská základna více než 100 000 členů

9.2 Internet Architecture Board (IAB)

- Technická poradní skupina ISOC, specifikuje politiku protokolové architektury Internetu
- Spravuje většinu čísel používaných v Internetu (čísla portů, protokolů …)
- Standardizace (normy *de facto*), editace a vydávání RFC dokumentů
- Volí předsedy IRTF a IETF

9.3 Internet Engineering Task Force (IETF)

- Připravuje pod vedením IAB specifikace dokumentů RFC (normy *de facto*)
- Veškerá práce na specifikacích se uskutečňuje pomocí mailing listů (diskusních skupin), zasedání je ~3x ročně
- Proces vzniku nového standardu nebo protokolu
  - prvotní návrh může podat kdokoliv (internet draft), dokument je volně k dispozici pro komentář
  - pokud se na návrhu do dvou měsíců od uveřejnění nepracuje, nebo jej nikdo nekomentuje, je stažen
  - po dvou diskusích se buď ustaví formální pracovní skupina nebo se návrh opustí
  - návrh, který se ujme, se podrobně rozpacovává (internet draft)
  - po minimálně šesti měsících práce vznikne návrh normy (proposed standard)
  - následuje ověřování v praxi pro dobu minimálně čtyř měsíců
  - aby se navrhovaná norma stala normou *de facto* musí existovat minimálně dvě nezávislé implementace ověřující správnost a životaschopnost normy a až poté je vydáno RFC
  - vznik normy trvá obyčejně několik let, výsledku se dosahuje dohodou

9.4 Internet Research Task Force (IRTF)

- Skupina podporující výzkum v oblasti protokolů, aplikací, architektury a technologií využitelných pro Internet
- Skládá se z dlouhodobých úzce zaměřených skupin

(10) Windows XP

10.1 Historické souvislosti

V polovině osmdesátých let minulého století vyvíjela firma Microsoft na zakázku a ve spolupráci s firmou IBM nový operační systém. Byl nazván OS2 a měl nahradit DOS. Což byl velmi úspěšný operační systém. Oficiálně se ho prodalo více než 20 milionů kopií. Odhaduje se, že nelegálních kopií bylo několikanásobek. OS2 byl velmi ambiciózní projekt. Měl vyvinout standardní OS personálních počítačů nejméně na jednu dekádu. vyvíjel jej tým, který vedl Ed Iacobucci. Byl napsán v assembleru pro výpočetní systém na bázi architektury I80286. Nakonec však všechno bylo jinak. V roce 1988 se MS rozhodl, že pro personální počítače vyvine svůj vlastní OS. Zcela novou technologií - NT (New technology). Od počátku se předpokládalo, že nový systém bude mít dvě rozhraní - OS2 a POSIX. Vedoucím týmu se stal Dave Catler (bývalý šéf architektury OS VAX/VMS u firmy DEC). Úspěch Windows, což byla ve své době pouze grafická nástavba DOSu, způsobil, že API OS/2 bylo následně nahrazeno Win32. První verzí NT byly Windows NT 3.1 (a Advanced Server). Windows NT 4.0 převzaly uživatelské rozhraní Win95 a mezi součástmi systému se objevil i webový prohlížeč a webový server. Navíc, programový kód uživatelského rozhraní a grafiky se stal součástí jádra OS. Cílem bylo zvýšení výkonu systému, daní a to pak snížení stability. Byla to také poslední verze NT, která byla portována i na jiné architektury než Intel. Následující verze Win2k přinesla řadu zásadních změn. Především Active Directory, větší podporu sítím a

mobilním zařízením (přenosné počítače), distribuovaný souborový systém, PNP, víceprocesorové systémy a mnoho dalšího. další krok byl víceméně logický. Došlo ke sloučení systémů Win 95, 98, a NT. v září 2001 byly uvolněny Win XP, serverová verze pak v roce 2002 (Windows .NET Server). Verze XP přinesla nové grafické rozhraní, snadnější používání (Ease-of-Use), orientaci na multimédia i větší stabilitu a bezpečnost. OS Win XP je realizace architektury klient-server. Implementuje architekturu podsystémů. Rozumí se tím uživatelsky orientované procesy, které realizují vlastnosti jiného operačního systému. Například POSIX a Win32 API (více OS v jednom). Win XP je víceuživatelský systém založený buď na distribuovaných službách, nebo na vícenásobném přístupu pomocí grafického uživatelského rozhraní (terminal server). XP je také první verze Windows, která existuje v 64b variantě. Výhodou je mnohem větší adresový prostor (různé komponenty Win užívají 64 bitová čísla již dávno, např file systém). Pro běžné uživatele jsou na trhu dvě edice Win. Professional je určena pro kancelářské použití, případně pro kvalifikované domácí práce. Vykazuje rysy pokročilého systému - umí spolupracovat s Active Directory, je stabilnější a umožňuje spouštění POSIX aplikací. Edice Home je jednodušší (především v obsluze) a umožňuje snadný přechod z verzí Win 95 a 98. Je určena pro domácí uživatele s orientací na multimediální aplikace. Serverová verze je na trhu v několika edicích. Základem je Win Server 2003. Ten existuje ve čtyřech verzích. Standard, Enterprise, Datacenter a Web. Liší se rozsahem (množstvím komponent) i parametry. Zvláštní kapitola pak představuje Windows Server 2003 SBS (Small Business Server, dvě edice). Je vhodný pro systémy do 75 počítačů a představuje přítulnou verzi Windows Server. Obsahuje všechny základní komponenty a navíc sadu uživatelsky přívětivých a komfortních utilit pro konfiguraci (víceméně automatickou).

10.2 Principy návrhu

Základními parametry návrhu nového OS byla bezpečnost, spolehlivost, kompatibilita pro Win a POSIX aplikace, vysoký výkon, rozšířitelnost, přenosnost a mezinárodní podporu.

Bezpečnost a spolehlivost

Jedná se o stěžejní parametry návrhu nového OS. Výsledkem je, že verze NT 4.0 získala klasifikaci C-2. Což představuje středně zabezpečený výpočetní systém. Zabezpečený znamená, že programový kód byl důkladně testován na skryté chyby,a že má zvýšenou odolnost proti škodlivému SW. Také byl doplněn systém automatických aktualizací a subsystém monitorování stavu (PC Health).

Kompatibilita pro aplikace

Systém musí podporovat jak aplikace pro starší verze Win, tak i aplikace na bázi standardu POSIX. Starší aplikace většinou prověřují kompatibilitu dotazem na verzi OS.Aby bylo vyhověno všem požadavkům, byla mezi aplikace a API Win32 vložena vrstva zajišťující kompatibilitu (Compatibility Layer). Vrstva se stará o to, aby Win XP vypadaly tak, jak si aplikace přeje. Vrstva také transformuje volání služeb jiných než Win32 API do žádané podoby. Identický postup je implementován při transformaci API Win32 na Win64.

Výkon systému

Principiální je zkrácení programového kódu, zejména v kritických procesech (lepší algoritmy) a jeho důsledná optimalizace. Další použité techniky jsou následující:

- asynchronní I/O
- vylepšené síťové protokoly
- přenesení grafických rutin do jádra systému
- sofistikovanější subsystém vyrovnávacích pamětí souborového systému
- lepší správa (procesy, paměť)

Navíc byla přidána podpora pro multiprocessing.

Rozšiřitelnost

Rozumí se schopnost systému akceptovat, vstřebávat nové technologie (tedy, pomalu zastarávat). Systém má vrstevnatou strukturu. Řídící jádro systému (kernel) poskytuje služby ostatním částem systému. Nad jádrem pracují servery, které emulují prostředí různých operačních systémů (DOS, Win16, POSIX). Protože systém je důsledně modulární, je možné přidávat podle potřeby další servery, bez vlivu na jádro. Ovladače zařízení pracují na stejném principu, takže je možné přidávat další zařízení.

Přenositelnost na jiné platformy

Všechny hlavní části programového kódu jsou napsány v C a C++. Většina na HW závislém kódu je izolována v dynamicky linkovaných knihovnách zvaných HAL (HW Abstract Layer), které poskytují služby ostatním.

10.3 HAL

Slouží ke skrytí rozdílů v hardwaru, poskytuje tak ostatním modulům standardní rozhraní pro přístup k hardwaru a zařízení pak potřebují jen relativně jednoduchý driver (který nepřistupuje přímo k hardwaru).

10.4 Jádro

Základ systému, nemůže být vytěsněno z paměti ani přerušeno (non preemptive).

Plní 4 základní funkce:

- správa procesů
- správa přerušení a výjimek
- synchronizace mezi procesy
- obnova po selhání napájení

Svou činnost jádro vykonává pomocí sady systémových objektů. Podstatnou komponentou je *dispečer*, který provádí plánování procesů a vláken, implementaci synchronizačních primitivů, správu časových intervalů a správu přerušení a výjimek.

Vykonatelný kód je organizován jako proces a vykonáván jako vlákno a což se stará právě dispečer jádra. Vlákno se může nacházet v jednom ze šesti stavů:

- připraven (vlákno může být spuštěno)
- v pohotovosti (do tohoto stavu je převedeno vlákno s nejvyšší prioritou)
- běží (procesor je přidělen a vlákno reálně pracuje)
- čeká (pozastaveno)
  - spuštění vlákna s vyšší prioritou
  - uplynutí časového intervalu
  - blokovací signál dispečera objektů (např. Čekání na ukončení operace přenosu dat)
- přechodný stav (čeká se na přidělení potřebných zdrojů)
- ukončen (uvolňují se alokované zdroje)

Dispečer jádra při rozhodování vychází s prioritního schématu o 32 úrovních a každé vlákno má přidělenu tabulku stavu, ve které je mimo jiné, aktuální priorita, afinita k procesoru, využití kvanta času. Algoritmus rozhodování je poměrně složitý a v podstatě se dá říct, že se jedná o fronty s prioritou.

10.5 Synchronizace v jádře

- Hlavní funkcí synchronizačních mechanizmů je zabránit korupci dat (data corruption)
- Dva typy dispečeru objektu
  - **signalizační** – signalizuje, že se určitá událost stala, reagovat může více vláken (cosi jako signál „start“)
  - **synchronizační** – slouží synchronizaci činnosti vláken, reagovat bude jen jedno vlákno

Dispečer objektů EVENT

- Slouží k řízení vláken
- Zaznamenává události v systému a synchronizuje příslušné akce
- Příklad: vlákno A potřebuje data, která jsou výsledkem činnosti vlákna B. Je evidentní, že cyklické dotazování, zda jsou data již k dispozici není právě efektivní řešení. Výhodnější je vlákno B pozastavit, dokud data nejsou připravena a objekt EVENT dá vědět, až data připravena budou.

Dispečeři objektů MUTANT a MUTEX

- Slouží k řízení bezkonfliktního přístupu k sdíleným datům (**Mut**ual **Ex**clusion – vzájemně se vylučující)
- Dovolí v daný moment pouze jednomu vláknu přístup ke sdíleným datům, ostatní vlákna dožadující se přístupu budou pozastavena, dokud první vlákno nedokončí práci s daty
- MUTANT pracuje v obou režimech (), MUTEX pouze v jádře

Dispečer objektů SEMAPHORE

- Slouží ke správě zdrojů přidělených jednotlivým vláknům

- Vykonává úlohu čítače nebo brány zajišťující souběžný, ale vzájemně se vylučující přístup k chráněnému prostředku – proces je v podstatě správní jednotka systému, nic neprovádí a jen poskytuje kontext pro spuštění vláken

Dispečer objektů PROCESS

- Vlastní správní jednotka prostředí (nic neprovádí, poskytuje kontext pro spouštění dalších vláken)
- Slouží k zapouzdření objektu THREAD a poskytuje mu virtuální adresový prostor

Dispečer objektů THREAD

- Základní běhová jednotka (nejmenší spustitelný kód)
- FIBER je vlákno plánované aplikací

Dispečer objektů TIMER

- Objekt pro odměňování časových intervalů (pro potřeby plánování periodicky se opakujících činností a pro potřeby ukončení akcí, které vyčerpaly svůj čas)

10.6 Jádro systému

- Zajišťuje synchronizaci mezi procesy v multi-procesorových systémech
- Pomocí mechanismu SPINLOCK zajišťuje, aby ke sdíleným datům nepřistupovalo více vláken najednou (pokud běžící vlákno SL nastaví, kontext se nepřepne, dokud vlákno činnost nedokončí – jiné vlákno nebude mít k datům přístup)
- Blokována vlákna jsou ve stavu aktivního čekání, dokud je blokující vlákno v kritické sekci

10.7 Přerušení v jádře

O všechny případy atypických situací, řešených v módu jádra se starají dispečer výjimek a dispečer přerušení. Zpracování výjimek je relativně jednodušší, protože nezávisí na architektuře procesoru (rozdílné architektury používají různý počet přerušení odlišných typů).

Dispečer přerušení

- Převádí hardwarové přerušení na standardní sadu
- Windows XP má 32 úrovní přerušení
  - 31 je nejvyšší priorita (Machine Check & Bus Error)
  - 8 přerušení je pro účely jádra
  - 24 reprezentují své hardwarové ekvivalenty (skrze HAL)
- Tradiční IRQ PC jsou vedeny jako IRQL 3-26
- **Tabulka přerušení** slouží je svázání úrovně přerušení s konkrétní rutinou
  - může jich být jedna nebo více
  - každé CPU má svoji tabulku a masku pro blokování hardwarového přerušení
- Prioritní schéma zajišťuje, že přerušení jsou obsluhována podle důležitosti
- Softwarové přerušení (instrukce INT Ix) jsou reprezentovány jako úrovně 0-2 (volání některých systémových funkcí)
  - APC (Asynchronous Procedure Call) – umožňuje vykonat systémový kód v kontextu uživatelského vlákna (obsluha procedurou), např. spustit nové vlákno, ukončit proces nebo doručit zprávu o dokončení I/O operace
  - DPC (Delayed Procedure Call) – umožňuje oddálit obsluhu přerušení, odložené požadavky jsou pak řazeny do fronty a zpracovány dispečerem až v okamžiku, kdy na ně přijde řada v prioritním schématu (obsluha procedurou = vláknem)

Dispečer výjimek

- Obsluhuje atypické situace vzniklé v důsledku vykonávání kódu vlákna
- Výjimky jsou zpracovány asynchronně, v kontextu vlákna, kde k výjimce došlo
- Zpracovává tyto události
  - narušení ochrany paměti
  - přetečení v celočíselné aritmetice (FX)
  - přetečení v aritmetice s desetinnou čárkou (FP)
  - dělení nulou (FP a FX)
  - ilegální operace (= operace, které nejsou na dané úrovní povoleny)
  - privilegovaná instrukce
  - chyba stránkování
  - narušení přístupových práv
  - přetečení stránkovacího souboru
  - ladění (krokování)
- Pokud dojde k výjimce v módu jádra, dispečer spustí rutinní vyhledání příslušné obslužné procedury
  - pokud **je** nalezena, tak jí předá řízení
  - pokud **není** nalezena, vznikne fatální chyba (BSOD)
- V uživatelském módu vytvoří subsystém operačního prostředí každému procesu port pro výjimky (případně ladící port)
  - je-li registrován ladící port a je schopen danou výjimku zpracovat, předá dispečer výjimek požadavek na tento port – pokud není pošle signál subsystému operačního prostředí, který se pokusí převést chybu na místně příslušnou

10.8 Exekutiva

- Výkonná (řídící) část systému poskytující systémové služby ostatním subsystémům
- Základní moduly
  - správce objektů
  - správce virtuální paměti
  - správce procesů
  - soubor lokálních obslužných procedur
  - manažer cache
  - správce I/O
  - bezpečnostní manažer
  - správce Plug&Play
  - zavádění systému

Manažer (správce) objektů

- **Objekt je netransparentní datová struktura**
- Implementovaná a manipulovaná určitou komponentou jádra systému (data + množina operací)
- Základ pro manipulaci s objekty (metody pro vznik a zrušení instance objektu, dynamické přidání nového typu objektu, udržuje globální jmenný prostor systému a vytváří jeho hierarchickou strukturu)
- Objekty se rozlišují podle jména nebo systémového identifikátoru (Handle)
- Handle je standardní interface pro objekt libovolného typu, je unikátní pro každý proces
- Pro manipulaci s objekty existují virtuální funkce (Create, Open, Close, Delete, Parse …)

Manažer procesů

- Služby pro vznik, použití a zrušení procesů, vláken a úloh
- Proces je objekt, který reprezentuje instanci vykonávajícího se programu
- Představuje virtuální adresový prostor, kód a data zde uložená a přidělené systémové zdroje
- Každý proces je tvořen jedním nebo více vlákny (vlákno = základní běhová jednotka, nejmenší spustitelný kód)
- **Úlohy** sdružují procesy do větších organizačních jednotek (sleduje se tím lepší využití prostředků)

- Postup při vzniku procesu
  - Iniciátor pošle zprávu Win32 API, že hodlá vytvořit proces
  - Procedura CreateProcess() v původním procesu předá řízení manažeru procesů
  - Manažer procesů předá požadavek manažeru objektů, který vytvoří objekt procesu a předá příslušný handle Win32 API
  - Win32 API znovu zavolá manažera procesů, aby vytvořil vlákno pro proces a vrátil handle vlákna

Struktura paměti

Pro architekturu IA32 platí, že každý proces má adresový prostor o velikosti 4GB. Dolní 2GB je privátní adresový prostor daného procesu, který je pro jiné procesy v systému nepřístupný (User Mode). Zde je uložen programový kód a data daného procesu (prvá stránka je úmyslně nepoužitelná – snazší odhalování chyb v adresaci). Hodní 2GB jsou vyhrazeny pro potřeby systému a jejich struktura je pro všechny procesy v zásadě stejná. Spodní část je vyhrazena pro jádro a ovladače zavádění systému. Paměť je přímo mapována do fyzické paměti a nikdy nemůže být vytěsněna na vnější médium. Velikost se různí. Další část zabírá hyperprostor, což je úsek paměti přístupný pouze v režimu jádra. Slouží k uložení datových struktur specifických pro daný proces, které spravuje VMM (adresář a tabulky stránek). Při přepnutí kontextu je obsah aktualizován. Typická velikost je 4MB. Manažer cache typicky spotřebuje dalších 512MB. Logicky je tento prostor rozdělen na metadata a vyrovnávací paměť. Zbytek je označován jako pool. Dělí se na část, která je stránkovaná a část, která stránkovaná není. Používá je podle potřeby operačního systému.

Manažer virtuální paměti (VMM)

- Spravuje **virtuální adresový prostor, přidělování fyzické paměti, stránkování a paměť cache**
- Základní mechanismus stránkování používá dvouúrovňové překládání a stránku o statické velikosti 4kB
- Adresář stránek obsahuje 1024 položek o délce 4 bajty (= 4kB)
  - každá položka adresuje tabulku stránek o 1024 položkách a 4 bajty
  - *hyperprostor* má tedy 1024 x 4 kB = 4 MB
- Hardwarovou podporu pro mechanizmus stránkování poskytuje MMU (Memory Management Unit)
- V případě architektury IA-64 má stránka 8 kB a překlad je tříúrovňový
  - adresář i tabulky stránek mají 1024 položek a 8 bajtů (8 kB)
  - výsledek je 10 + 10 + 10 + 13 = 43, což poskytuje virtuální adresový prostor 8 TB
- Proces přidělování paměti má dva kroky
  - požadované množství paměti se rezervuje ve virtuální paměti
  - přidělí se fyzická paměť (Commit, mapování virtuální na fyzickou)
- Pro zvýšení výkonu mohou privilegované procesy zakázat vytěsňovat některé bloky přidělené paměti na vnější médium
- VMM také spravuje databázi všech paměťových rámců (PFD, Page-Frame Database)
  - položka databáze udává stav příslušného rámce, počet a identifikaci sdílených stránek
  - pro výměnu stránek se používá algoritmus FIFO

(11) Bezpečnost výpočetních systémů

11.1 Úvod

Informace mají tržní hodnotu, představují majetek a lze je tedy vlastnit, kupovat a prodávat. Ale také je možné je krást, ničit a falšovat.

Hodnota informace je dána především její **výlučností, přesností a vypovídací schopností**.

11.2 Informační bezpečnost

- **Souhrn ochranných opatření zajišťujících principy důvěryhodnosti, integrity, dostupnosti a prokazatelné zodpovědnosti při činnosti informačně-technologického systému**
- **Informačně-technologický systém** tvoří technické a paměťové prostředky, paměťová média a zainteresované osoby
- **Informační bezpečnost** se zabývá ochranou informací během jejich vzniku, zpracování, ukládání, přenosu a likvidace.
  - Zahrnuje bezpečnost **administrativní, komunikační, počítačovou, personální a fyzickou**
  - Komplex logických, technických, fyzických a organizačních opatření, která zabraňují ztrátě důvěrnosti, integrity a dostupnosti obhospodařovaných dat
  - Cílem je zabránit kompromitaci nebo nedovolené modifikaci dat či destrukci systému nebo jeho části, což by následně umožnilo zneužití citlivých informací, použití klamných dat, ze kterých budou vyvozeny chybné výsledky či závěry
- Hrozby jsou všechny výše uvedené skutečnosti – v podstatě to říká, že musíme informační systém chránit tak, aby nedošlo ke škodě
- **Podoby škody**
  - **přímá finanční ztráta** nebo fyzické poškození
  - ztráta **dostupnosti** – informace je nedostupná všem nebo někomu, kdo k ní má mít přístup
  - ztráta **důvěrnosti** – informace je dostupná i těm, kteří k ní neměli mít přístup
  - ztráta **integrity** – data byla modifikována a informace je v důsledku toho zničena nebo změněna, úmyslně či neúmyslně
  - ztráta **autentičnosti** – není možné určit původce nebo zdroj informace
- Data jsou v informačním systému většinou to nejcennější
  - data jsou utajována
  - existence dat je utajována
  - důvod utajování dat je utajován
- ***Počítačová kriminalita*** – jakákoli trestná činnost páchaná pomocí výpočetní techniky nebo ohrožující informační systémy resp. data v nich uložená

11.3 Základní pojmy

Bezpečnost

Vlastnost nebo stav ochrany proti nevyhnutelným ztrátám. Zahrnuje ochranu činnosti zpracování, úschovy, distribuce a prezentace informací. Je to kombinace důvěrnosti, integrity, dostupnosti a zodpovědnosti.

Hrozba

Akce nebo událost, která může ohrozit bezpečnost. Je to zneužití zranitelnosti. Lze také definovat jako pravděpodobnost útoku daná atraktivitou systému pro útočníka.

Zranitelnost

Slabé místo v systému, které může být zneužito k narušení zamýšleného chování systému.

Aktivum (Asset)

Cokoli co má cenu. Dělí se na hmotné (např. počítače a komunikační technika) nebo nehmotné (data, programové vybavení).

Hodnota aktiva (Asset Value)

Představuje ocenění důležitosti a významu objektu pro vlastníka. Nemusí být vyjádřena v penězích.

Citlivá data (informace)

Data vyžadující zvláštní ochranu, protože existuje určitá pravděpodobnost působení hrozeb.

Průnik

Akt neautorizovaného použití informačního systému.

Důvěrnost

Charakteristika informace, která znemožňuje její odhalení neautorizovaným subjektům.

Integrita

Vlastnost, která určuje, že daný objekt byl změněn pouze specifikovaným a autorizovaným způsobem. Vyjadřuje fakt, že daná data jsou stále platnou reprezentací určité informace.

- **Možné hrozby**
  - modifikace
  - přidání
  - smazání
- **Integrita systému**
  - schopnost vykonávat určité funkce neměnným způsobem, bez jakékoli skryté nebo úmyslné neautorizované manipulace se systémem.

Riziko

Pravděpodobnost zničení nebo poškození určitého aktiva, konkrétní hrozbou – pravděpodobnost úspěšného útoku.

Dostupnost

Vlastnost systému, která zabraňuje neautorizovanému zadržení zdrojů, které jsou pak autorizovaným subjektům dostupné pouze se zdržením (zabránit DoS).

11.4 Bezpečnostní politika

Definuje cíle, požadavky principy, omezení a postupy, které určují způsob správy, ochrany a distribuce citlivých informací. Je to soubor kritérií pro aplikaci služeb bezpečnosti. Představuje **základní východisko pro řízení bezpečnosti informačního systému.** Vždy se vztahuje k určité organizační jednotce. Bezpečnostní politika může být například státní, podniková, oddělení. Cílem bezpečnosti politiky je minimalizace rizik. Není to pojistka proti úniku informací, ale snížení výskytu nebezpečí. V zásadě vždy **představuje kompromis mezi omezováním uživatelů a chráněným zájmem** organizace. Bezpečnostní politiku by měla vypracovávat skupina odborníků, ze všech struktur organizace.

Je to zásadní dokument definující základní principy vedoucí k prosazení bezpečnostních funkcí organizace. V dokumentu je uvedeno, jakými mechanismy logického, fyzického a organizačního zabezpečení budou bezpečnostní funkce prosazeny.

Dokument je sestavován na základě analýzy, jež se skládá z následujících bodů:

- **Analýza aktiv** – vymezuje aktiva, která je nutno chránit; může to být dostupnost, důvěrnosti a integrita data, technické a programové prostředky zajišťující podporu obchodním procesům atp.
- **Analýza hrozeb** – definuje proti čemu chceme aktiva chránit; například vyzrazení, zničení, znepřístupnění informací, podvržení nebo modifikaci dat, zneužívání technických prostředků, narušování chodu informačního systému
- **Analýza rizik** – vymezuje, jaká rizika hrozí informačním aktivům, jak velká tato rizika jsou, proti kterým rizikům chce organizace chránit svá aktiva a následně kolik prostředků je ochotna investovat do eliminace těchto rizik
- Výsledkem těchto analýz je návrh **protiopatření**

Bezpečnostní politika může mít charakter povinných nařízení měnitelných pouze velmi omezenou skupinou lidí nebo je to souhrn bezpečnostních doporučení, jejichž následné uplatnění prosazují vlastníci jednotlivých aktiv.

Pro vypracovávání bezpečnostní politiky existuje několik přístupů. U nás se většinou vychází z britského standardu BS7799. Za východiska považujeme především:

- **ČSN ISO/IEC 17799** Informační technologie – Soubor postupů pro řízení informační bezpečnosti
- **ČSN ISO/IEC 13335** Informační technologie – Směrnice pro řízení bezpečnosti IT
- Information Security Management – překlad a interpretace standardu BS7799 pro české prostředí

Bezpečnostní politika odráží přístup organizace k řešení problému bezpečnosti informací. Definuje organizační a systémová opatření, nástroje a technologie, jež mají za úkol příslušná bezpečnostní opatření realizovat. Za bezpečnostní politiku je zodpovědný management organizace.

Okruhy bezpečnostní politiky

- ***Popis informačního systému*** představuje výchozí bod v procesu tvorby bezpečnostní politiky. Vychází z konkrétní organizační struktury informačního systému organizace. Definuje jaký je účel informačního systému, jaké úkoly plní, jaké v něm probíhající informační toky, jaká je návaznost na ostatní struktury organizace. Stanoví, co všechno je a co není informační systém, kde jsou jeho hranice a jaký je vliv okolí.
- ***Cíle bezpečnostní politiky*** vycházejí z obchodních cílů organizace, legislativy, smluv a interních požadavků. Jasné cíle umožní vytyčit strategii a rámcové postupy pro jejich dosažení.
- ***Legislativní rámec*** je vymezen jednak právními předpisy, které musí informační systém respektovat a bezpečnostními normativy, které musí splňovat, aby mohl být certifikován.
- ***Definice aktiv.*** Aktiva organizace mají svoji hodnotu, která je ve většině případů pro organizace z hlediska jejího fungování kritická. V případě ztráty nebo závažného poškození některých aktiv může dojít případně i k ukončení činnosti organizace. Klasifikace a správa aktiv pojednává o přidělení hodnoty určitým informacím podle síly dopadu na organizaci při ztrátě nebo nedostupnosti.
- ***Definice hrozeb*** taxativně (= „jsou tam vyjmenovány“) uvádí proti jakým hrozbám hodláme aktiva chránit. To je výchozím bodem pro ohodnocení rizik.
- ***Bezpečnostní funkce.*** Představují postupy vedoucí k minimalizaci rizik. Formulace bezpečnostní funkce probíhá v posloupnosti: určení hodnoty aktiva -> stanovení velikosti hrozby či stupně zranitelnosti -> odpovídající riziko -> bezpečnosti funkce, která eliminuje nebo omezí riziko na požadovanou míru. Formulaci bezpečnostní funkce lze provést výběrem z katalogu nebo jako výstup automatizovaného expertního systému.
- ***Personální bezpečnost*** obecně omezuje rizika od chyb lidí. Má dvě roviny. Jednak řeší problematiku přístupu osob k chráněným aktivům; prověřování osob, definování stupně důvěry, a jednak definuje požadavky na kvalifikaci.
- ***Zásady organizační politiky*** zahrnují organizační opatření – tedy, kdo je zodpovědný za co. Koordinuje jednotlivé bezpečnostní složky organizace – informační, majetkové a osobní.
- ***Technicko-provozní zabezpečení*** řeší finanční a materiální požadavky potřebné pro zavedení požadovaných bezpečnostních opatření.
- ***Plán obnovy po havárii*** je definovaná posloupnost úkonů, které umožní obnovu činnosti informačního systému po havárii.
- ***Metodika řešení krizových stavů*** slouží k řešení bezpečnostních incidentů.

11.5 Bezpečnostní politika státu

Představuje základní strukturu ochrany informací. Tvoří ji zákony a z nich odvozené předpisy, nařízení a směrnice. Jedná se o následující normativy.

- Zákon č. **412/2005** Sb. **O ochraně utajovaných informací a o bezpečnostní způsobilosti**
- Zákon č. **140/1961** Sb. **Trestní zákon**
- Zákon č. **101/2002** Sb. **O ochraně osobních údajů**
- Zákon č. **106/1999** Sb. **O právu na svobodný přístup k informacím**

Státem utajovaná informace (dříve státní, hospodářské či služební tajemství) je definována takto:

Utajovaná informace je informace v jakékoliv podobě zaznamenaná na jakémkoli nosiči a označená v souladu s tímto zákonem jejíž vyzrazení nebo zneužití může způsobit újmu zájmu České Republiky nebo by to mohlo být pro tyto zájmy nevýhodné.

Nutnou podmínkou je, že tato skutečnost je uvedena v **seznamu utajovaných skutečností**. Obecný rámec vytváří příslušný zákon. Utajovanou informací mohou být **fakta politického, vojenského či hospodářského významu**. Klasifikací chráněných informací na základě požadavků vládních institucí provádí Národní Bezpečnostní Úřad (NBÚ), který také kompiluje příslušný seznam. Únik chráněných informací řeší trestní zákon.

Zákon o ochraně utajovaných informací musí být, mimo jiné, v souladu s předpisy vyšší právní síly, zejména článkem 17 Listiny základních práv a svobod, který upravuje právo na informace a přípustné výjimky z něj. Státní normativy musí jednoznačně definovat všeobecně platně principy. Musí však být natolik obecné, aby nesvazovaly ruce bezpečnostním politikám organizací.

Každá organizace má vždy své specifické požadavky, na které musí být brán zřetel. Legislativní politika státu potom představuje právní rámec, ze kterého musí bezpečnostní politiky jednotlivých organizací vycházet.

11.6 Britský standard 7799

Nejznámější, celosvětově rozšířený bezpečnostní standard. Vznikl kolem roku 1995 a byl několikrát revidován. V prosinci roku 2000 byl přijat jako mezinárodní norma ISO/IEC 17799. Představuje standard komplexní ochrany informací. Pokrývá všechny oblasti bezpečnosti, popisuje rozsáhlý počet bezpečnostních opatření a je použitelná pro různé typy organizací. Skládá se ze tří částí. BS7799-1:1999 – Code of Practice for Information Security Management (Katalog bezpečnostních funkcí a opatření). Také ISO/IEC 17799:2000. Představuje podrobné shrnutí praktických zkušeností s řešením informační bezpečnosti. Definuje 120 bezpečnostních funkcí rozložených do 10 zón.

Druhá část, BS7799-2:1999 – Specification for Information Security Management Systems (označuje se zkratkou ISMS), představuje návod k výstavbě systému řízení bezpečnosti informací. V podstatě říká, jak aplikovat prvou část normy (BS7799-1). Třetí část je inovace části druhé. BS7799-2:2002 doporučuje jak používat prvou část normy a jak provozovat, udržovat a vylepšovat existující ISMS.

11.7 České normy

- Vznikají obvykle adopcí norem mezinárodních
- **ČSN BS 7799-2 zrušena** v říjnu 2006
- **Nahrazena** nově vydanou normou **ČSN ISO/IEC 27001**

ČSN ISO/IEC 27001

- Informační technologie – Bezpečnostní techniky – Systémy managementu bezpečnosti informací – Požadavky
- Propojena a harmonizována s
  - **ISO/IEC 9001:2000 (kvalita)**
  - **ISO/IEC 14001:2004 (životní prostředí)**
  - Hlavní část normy **definuje požadavky na vybudování, zavedení, provoz, monitorování, přezkoumávání, udržování, zlepšování a případnou certifikaci** systému managementu bezpečnosti informací
  - Specifikovány požadavky na výběr a zavedení bezpečnostních opatření chránících informační aktiva
- *Příloha A* - cíle a opatření, která jsou přímo propojena s cíli a opatřeními uvedenými v ISO/EIC 17799:2005
- *Příloha B* - vztah mezi principy OECD pro bezpečnost informačních systémů a sítí a fázemi PDCA cyklu
- *Příloha C* - vztah mezi ISO/IEC 9001:2000, ISO/IEC 14001:2004 a ISO/IEC 27001:2005

Je plánováno, že v budoucnu bude ISO/IEC 27xxx (ze které naše norma vychází) obsahovat sedm dokumentů:

- 27000, principy a slovník
- 27001, požadavky na ISMS (ISO ekvivalent BS7799-2:2004)
- 27002, návody pro zavádění
- 27003, analýzy rizik
- 27004, metriky a měření
- 27005, řízení rizik
- 27006, kontinuita podnikání a obnova po havárii

11.8 Systém správy bezpečnosti informačních systémů (ISMS)

Postup vypracování

- Vychází z druhé části standardu BS7799
- **Vedení organizace definuje cíle**, má dvě vazby
  - Neoddělitelnost informace od řízení a firemních procesů jako takových, informace představují v tomto případě zdroje, stejně jako peníze nebo pracovní síla – nezajištěnost vlastních zdrojů pak ohrožuje produkci (=> růst rizik)
  - Marketingová souvislost – společnost, která se snaží zvýšit svou důvěryhodnost na trhu získáním certifikátu jakosti, musí očekávat také dotaz na úroveň zabezpečení informací. Pro státní zákazníky je certifikace nutnou podmínkou.
- **Zjištění legislativního rámce** - získáme podklady jaké právními předpisy musí ISMS respektovat a jaké bezpečnostní normativy musí splňovat pro certifikaci
- **Identifikace a ocenění aktiv** – je nutné rozhodnout, jaká data jsou pro danou organizaci důležitá
  - klasifikace aktiv, rozdělíme je na skupiny podle síly negativního dopadu pro organizaci při jejich ztrátě
- **Definice možných hrozeb**

Bezpečnostní dekompozice systému

Informační systém je tvořen několika standardními částmi:

- data
- programové vybavení
- technické prostředky
- komunikační prostředky
- personál

Bezpečnostní dekompozice je rozložení systému na logické části a jejich analýza s cílem odhalit hrozby, ocenit rizika a vypracovat odpovídající bezpečnostní funkce.

Nejobecnějším rozdělením je vydělení na vnitřní a vnější bezpečnost.

- Vnitřní bezpečnost je systém ochran, kterou realizuje sám informační systém svým programovým nebo technickým vybavením
- Vnější bezpečnost jsou všechna opatření, která si nemůže informační systém zajistit svými prostředky

Bezpečnostní perimetr je část systému vnitřní bezpečnosti, ve které jsou v činnosti a platnosti kontrolní opatření za účelem ochrany aktiv.

Základní složky vnitřní bezpečnosti

- Datová bezpečnost – ochrana dat proti neoprávněné změně, poškození nebo ztrátě, systém řízení přístupu k datům
- Programová bezpečnost – výběr programového vybavení podle bezpečnostních kvalit a pracovní spolehlivosti
- Technická bezpečnost – výběr technických prostředků pro manipulaci s daty na základě spolehlivosti, návrh a realizace s ohledem na spolehlivost a odolnost proti poruchám, ochrana proti elektromagnetickému vyzařování
- Komunikační bezpečnost – souhrn bezpečnostních opatření zajišťujících integritu dat při přenosu mezi komponentami informačního systému

Základní složky vnější bezpečnosti

- Fyzická bezpečnost – zabezpečení aktiv technickými prostředky, ochrana budov, strážná služba
- Personální bezpečnost – souhrn opatření pro minimalizaci hrozeb způsobených lidským faktorem
- Režimová bezpečnost – komplex administrativních opatření a systém kontrol, které jsou zavedeny za účelem zajištění bezpečného chodu systému

Fyzická bezpečnost

Zahrnuje ochranu objektů včetně zařízení a organizace přístupu do nich. Dále pak umístění technologických, zabezpečovacích a monitorovacích zařízení. Především se jedná o

- Zabezpečení budov proti neoprávněném vniknutí (zamykací systém, poplašné zařízení, strážní služba)
- Monitorování pohybu osob (kontrolované zóny, identifikace osob)
- Katastrofám (záplavy, bouře, požár, pád letadla)

Další oblastí je technické zabezpečení provozu: nepřetržitá dodávka médií (vody, el. energie – UPS), ventilace a chlazení (klimatizace). Systém technických ochranných prostředků je tvořen:

- Mechanickými zábrannými prostředky
  - účelem je vytvoření časové prodlevy mezi okamžikem napadení objektu a časem jeho dokončení, velikost prodlevy je kriteriem bezpečnostní úrovně zábrany
  - patří sem části vnějšího uzavření (ploty, vrata), stavební prvky budov (zdi, podlahy, střechy), otvorové výplně (okna, dveře, mříže) a úschovné objekty (schránky, trezory, pokladny)
- Elektrickým signalizačním zařízením
  - požární signalizace, signalizace výskytu hořlavých plynů a zabezpečovací signalizace
- Souborem organizačních opatření
  - pro strážní službu musí být vypracovány směrnice ostrahy spolu s plány činnosti v případě mimořádných událostí
  - směrnice musí zcela jasně deklarovat pravomoci strážných v případě nutnosti ochraňovat střežený objekt
- Ostrahou

Dekompozice systému – Autentizace

- **Zabránění neoprávněným osobám v přístupu k aktivům** pomocí zábranných prostředků
- Autentizace přístupu = ověření a určení identity s požadovanou mírou záruky
- Provádí se na základě (v praxi většinou kombinace více přístupů, dvoufaktorová autorizace)
  - znalost něčeho (something you know)
  - vlastnictví něčeho (something you have)
  - osobní charakteristiky (něco jsme, something you are)

Heslo – důvěrná autentizační informace, provádí se ve dvou krocích:

- prohlášení identity (například uvedení jména)
- sdělení hesla (potvrzení identity)

Heslo je uváděno jako prostý text a může být odhaleno monitorováním nebo paděláním protokolu autentizace. Bezpečnější je použití jednorázového hesla nebo krypto-systému typu výzva/odpověď. Výzva je otevřená zpráva, odpověď její kryptogram. Bezpečnější jsou systémy, které nepoužívají prostou (tedy konstantní) výzvu, ale úvodní frázi (passphrase). Ta je určitým postupem transformována na virtuální heslo (klíč).

Například: „Dnes je pátek, 12. září, 08:09.120“. Podstatný je časový údaj – správná odpověď vznikne jeho transformací. Uvedené postupy předpokládají důvěryhodnost systému, ke kterému se přihlašujeme.