Kapitola 23

Architektura počítačů a sítí

Zdroje:

- zdroj Siti — Peterkovy slidy 3.1 (`http://www.earchiv.cz/l214/index.php3`)
- Peterkovy TCP/IP slidy 2.3(`http://www.earchiv.cz/l215/`)
- architektura pocitacu — Obdrzalkovy slidy (neumi odpovedet na mail), souborkove texty 3.99k, Jirovskeho slidy
- wikipedia (`http://www.wikipedia.org/`)
- Peterkuv archiv obecne `http://www.earchiv.cz/`
- Základ byl převzat z Majklových státnicových výsucků (v 0.5) — viz `http://mff.modry.cz/statnice/`

23.1 Von Neumannova architektura a její alternativy

see also wen:Von Neumann architecture, nebo taky `http://www.earchiv.cz/a94/a406c500.php3`

- Vychází z koncepce vzniklé v USA v letech 1940/1945, jejíž hlavní tvůrce je John von Neumann.
- Vnitřní strukturu počítače tvoří pamět, vstup, výstup a procesor. Struktura počítače se nemění s typem úlohy. Je jednotná pamět pro text programu i data. Rozdíl mezi programem a daty je dán jen jejich interpretací. Pamět je posloupnost stejně velkých pamětových míst. Počítač je řízen tokem instrukcí. Ty se provádějí, kdyz na ně dojde řada. V každém okamžiku probíhá jen jedna činnost.
- **nevýhody** — zastaralá, navrhována jako sekvenční, von Neumann bottleneck (propustnost dat mezi RAM a CPU je prilis nizka (relativne vzhledem k velikosti pameti a rychlosti cpu)
- Na Von Neumanove architekture jsou zalozeny vsechny dnesni mainstreamove pocitace. I kdyz od te doby doslo ke rade uprav (asynchronni I/O, cache, virtualni pamet)
- **alternativy** — počítač řízený tokem dat — operace se provede, jakmile jsou známy všechny její operandy

`http://www.root.cz/clanky/jak-pracuje-pocitac/`

Harvardska architektura

see also wen:Harvard architecture

- od Von Neumannovy se liší tím, že program ukládá do jiné paměti než data
- např DSP procesory a mikrokontrolery
- nehrozí přepsání programu sebou samým, ale je náročnější na výrobu kvůli většímu počtu paměťových sběrnic
- Jiná velikost slova (word) v programové a paměťové částí, t.j. instrukce s konstantou vejde do jediného načtení z programové paměti.
- Načítání instrukce a argumentu (program a data) se nepere o sběrnici (narozdíl od von Neumannovy arch.)

23.2 Multiprocesory

HW Architektura

- Volně spojené (Loosely coupled, grids) (propojené skupiny počítačů)
- Těsně (Tightly coupled) spojené multiprocesory
  - UMA (Uniform memory access, symetrický přístup k paměti)
    - Propojené sběrnicí
    - Propojené n^2 přepínači (crossbar nebo crosspoint switched)
    - Propojené $log_2(n)$ přepínači (multistage switched, omega networks)
    - Propojené jinak, třeba hyperkrychle, klika...
  - NUMA (Non uniform memory access, každý procesor svoji paměť)
    - S-COMA, skupina procesorů má svoji "bližší" paměť, požadavek na obsah paměti náležející jiným procesorům se propaguje vyšší vrstvě
    - ccNUMA, globální adresace, hodně řeší cache, direct memory — speciální paměť instancí a jejich kopií a platností.

Crossbar switched (propojeni pamětí a procesorů)

- Nákladé, kvadratický počet přepínačů
- Možnost zapojit více počítačů

```
M1  M2
|   |
o---o-- P1
|   |
o---o-- P2
```

Multistage switched (omega networks) (propojeni pamětí a procesorů)

- Postupně přepínaná cesta mezi procesorem a pamětí
- Při větším počtu stupňů pomalé
- n log(n) přepínačů

```
P1--                       --M1
    \                     /
     --SWITCH --- SWITCH -
    /        \  /         \
P2--          \/           --M2
P3--          /\           --M3
    \        /  \         /
     --SWITCH --- SWITCH -
    /                     \
P4--                       --M4
```

Rozložení vykonávaného kódu

- OS na jednom, aplikace na druhém (master CPU je botle neck),
- symetricky rozložené OS i aplikace na všech, spousta synchro problémů

Synchro problémy

- Instrukce TSL (test and set lock) musí mít podporu na HW sběrnice (jinak dva CPU mohou nesynchronizovaně volat TSL)
- Cache ping pong při spin locku
- Spin lock je smysluplný, někdo jiný ho může uvolnit (na uniprocesoru se přeplanuje)

Plánování

- Dva rozměry CO a KDE
- time-sharing, v systému jednotná fronta ready procesů, každý procesor si bere co může
- space-sharing — skupina procesů běží na skupině procesorů
- gang-sharing procesy sdruženy do gangů a všichni členové gangu jedou souběžně na různých procesorech

Out-of-order execution

wen:Memory barrier Kratce: procesor muze vykonovat operace v jinem poradi nez jsou v opcode. Na jednoprocesoru to nevidime, ale na multiprocesoru muze dochazet k tomu ze procesory vidi takove, rekneme, sekvencne nekonzistentni mezivysledky. Barierova instrukce prave zajisti synchronizaci.

```
Processor #1:
 loop:
  load the value in location f, if it is 0 goto loop
 print the value in location x

Processor #2:
 store the value 42 into location x
 store the value 1 into location f
```

Pri out-of-order vykonavani instrukci tento kod muze vypsat 0 (BUNO predchozi hodnota prom. x) nebo 42.

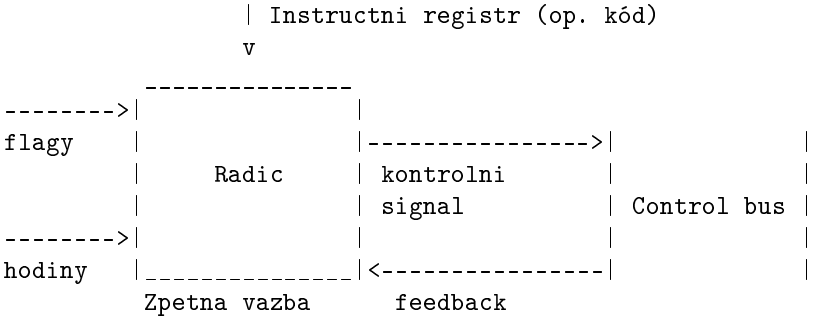
z wikipedie

- – MIMD multiprocessing architecture is suitable for a wide variety of tasks in which completely independent and parallel execution of instructions touching different sets of data can be put to productive use. For this reason, and because it is easy to implement, MIMD predominates in multiprocessing.
  – SIMD multiprocessing is well suited to parallel or vector processing, in which a very large set of data can be divided into parts that are individually subjected to identical but independent operations. A single instruction stream directs the operation of multiple processing units to perform the same manipulations simultaneously on potentially large amounts of data.

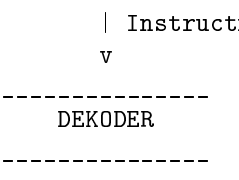
23.3 Mikroprogramové a klasické řadiče, mikroprogramování

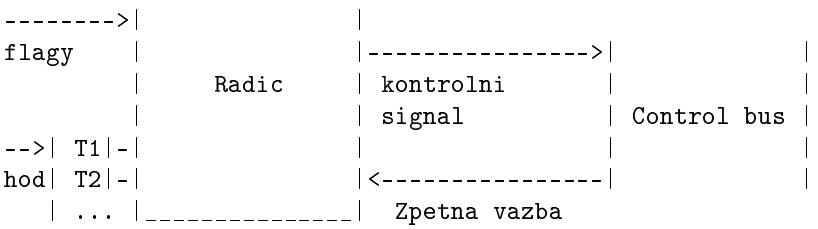
see wen:Microcode

- Klasický řadič má všechny postupy vykonávání instrukcí zadrátované v HW. Pro každou instrukci a stav je naprogramováno, jaké řídící signály mají být dány. Jde vlastně o stavový automat.



- Mikroprogramový řadič má místotoho mikrokód, který popisuje proces vykonávání instrukcí na úplně nejnižší úrovni
- Trochu komplikace s pipeliningem, řídící signály ještě závisí na instrukčním cyklu



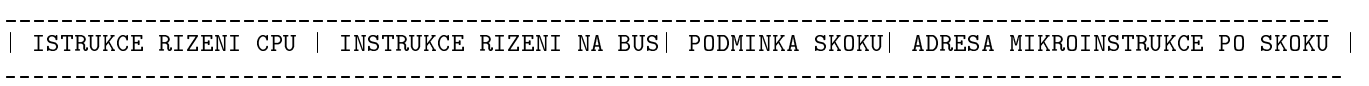


Mikroprogram vypadá nějak takto: Propoj registr 1 se vstupem ALU "A", propoj registr 2 se vstupem ALU "B", nastav ALU na sčítání, vynuluj carry bit v ALU, propoj výstup ALU s registrem 8, aktualizuj registr příznaků podle příznaků v ALU, posuň čítač instrukcí, načti další instrukci, ...

Mikroinstrukce jsou uloženy v oblasti zvané control store, z něhož vybírá instrukce mikroseqeuncer – ten typicky nějak spojuje počítadlo, aktuální hodnotu instrukce kterou CPU zpracovává a pole v mikroinstrukci které říká kam dál.

- horizontální mikrokód – široký, mikroinstrukce obsahuje za sebou pokyny pro všechna propojování, nastavování, skok dál, etc; instrukce jsou široké 56 bitů nebo i víc

Format může vypadat třeba takhle



Provedení instrukce:

- Nastav řídící signály
- Podle podminky se rozhodni
  - Buď proveď další instrukci
  - Nebo skoč
- vertikální mikrokód – mikroinstrukce jsou zakódované, takže jsou kratší, ale vyžadují další zpracování, aby se z nich dostaly ty pokyny
  - někdy to je assembler jednoduššího počítače, který emuluje počítač složitější
  - další možná forma je dvojice (jednotka, pokyn_jednotce) – je potřeba míň paměti, ale CPU je pomalejší
- Nanoprogramování, instrukce mikrokódu se komprimuje tak, že je vlastně jen indexem v adresáři nanoinstrukcí, které něco dělají?

Se klesající cenou tranzistoru začal dominovat horizontální mikrokód.

Wilkesova řídíci jednotka. Založena na řídící matici. Mikroinstukce vybere řádek, podle toho jestli je na dané pozici v řádku jednička se provedou řídící signály odpovídající sloupcům v první části matice a vybere se nasledující instrukce podle jedniček a nul ve sloupcích v příslušném řádku druhé části matice.

23.4 Paměťová hierarchie, vyrovnávací paměti

- obdrzalkovy slidy na principy 07
- registry — cpu, nejrychlejsi — 1ns, nejmensi ~B
- vyrovnavaci pamet — L1, L2 cache, 10ns ~kB
- operacni pamet — RAM 10-100ns ~MB
- sekundarni pamet — HDD 10ms ~GB
- archivni pamet — pasky, DVD, nejpomalejsi nejvetsi 100+ ms, TB
- **vyrovnávací paměti**
  - obvykle pouziti — tam, kde je vyrazny rozdil v rychlostech — procesor vs ram, pristupy na disk.
  - Vyuziva se casove lokality pristupu — data jednou pouzita budou pravdepodobne pouzita podruhe a pametove lokality — kdyz pouziju nejaka data, tak asi budu pouzivat i data ulozena v nejblizsim okoli (pamet pracuje po strankach, takze typicky nactu i okoli) data v cache ulozena spolu se svou adresou. k vyhledavani je mozno pouzit nekolik druhu mapovani (plne asociativni, prime, ...)

- cache ma typicky mnohem mensi kapacitu nez ta pamet, ktera je cacheovana (cpu ma radove 0,5MB cache vs RAM 0,5GB; HDD ma 8MB cache a kapacitu 200GB).
- RAM je v podstate taky cache sekundarni pameti (disku).
- Pro uvolnovani bloku z cache mozno pouzit ruzne algoritmy (LRU, LFU, FIFO, ...)
- typy cache:
  - **write-thru** — používat průpis (co se zapíše do vyrovnávací paměti, zapíše se okamžitě i do hlavní)
  - **write-back** — uklízet modifikované bloky do hlavní paměti (vyšší výkon). Při zápisu do vyrovnávací paměti se pouze nastaví (dirty) příznak a do hlavní paměti se data zapíší, až když jsou vyhozena z cache (a pouze pokud mají nastaveno dirty).
- Vyhledávání v cache:
  - **plně asociativní mapování** — při hledání hledána přímo adresa v asociativním adresáři
  - **přímé mapování** — každý blok v cache své místo
  - **skupinové asociativní mapování** — kombinace předchozích dvou (2-way atd.)
- U multiprocesoru je potřeba zaručit konzistenci dat mezi procesorovými cache.
- Efektivni pristupova doba — "amortizovany" cas potrebny k pristupu do pameti — pomoci cetnosti vypadku stranek a prumerneho casu pristupu do cache a cacheovne pameti se spocita, jaky byl prumerny pristupovy cas ke strance.

23.5 Stránkování a segmentace

Virtualni pamet — zajišťuje větší adresový prostor, než je k dispozici fyzicke pameti. Existují dvě základní metody:

Stránkování

Prime strankovani

Procesy mají k dispozici velký virtuální adresový prostor. Operační paměť tvoří fyzický adresový prostor. Virtuální i fyzická pamět je rozdělena na bloky stejné velikosti — virtuální na stránky, fyzická na rámce. Proces se odvolává pouze na adresy ve svém virtuálním prostoru. Operační systém plní stránkovací tabulky informacemi o tom, které stránky jsou v kterých rámcích — mapování. Při pokusu procesu o přístup k paměti jednotka řízení paměti(MMU — memory management unit) dle stránkovacích tabulek zjistí, zda se stránka obsahující požadované místo nalézá v operační paměti. Pokud ano, provede překlad adresy na fyzickou a instrukci předá procesoru. Pokud ne, vygeneruje přerušení — nastal výpadek stránky(page fault).

OS nalezne volný rámec (případně nějaký uvolní tím, že jeho obsah zapíše do odkládací paměti — viz strategie vymeny stranek) a z odkládací paměti natáhne požadovanou stránku do rámce. Procesor nyní zopakuje instrukci, která přerušení vyvolala. Celý postup je pro proces naprosto transparentní. Pro 32bitovy prostor a 4kB stranky maji strankovaci tabulky uz velikost 4MB, takze se pouziva vice urovnove strankovani (strategie jako indexovani souboru)

32bitova virtualni "pametova adresa" pouzivana procesem je rozdelena na adresu stranky a adresu ve strance. U 4kB stranek se prvnich 20bitu pouzije pro mechanismus hledani fyzicke stranky a az je nalezena, tak poslednich 12 bitu urcuje poradi v ramci stranky, kde je to, co hledame. zvysovani vykonu: mmu primo soucasti procesoru, TLB — transaction lookaside buffer — naposledy pouzita mapovani jsou v malem bufferu, kde se vyhledaji rychleji.

Strategie vymeny stranek:

- OPT — vyhodit stranku, ktera bude potreba za nejdelsi casovy usek — neimplementovatelna teoreticka strategie.
- FIFO — klasicka fronta, trpi Beladyho anomálií (012301401234 pro 3/4 sloty – přidání paměti vede k více výpadkům).
- LRU — Least Recently Used — nejlepsi aproximace OPT
  - casovy "citac" — obsahuje timestamp posledniho pristupu — moc nefunguje při procházení velkých struktur.
  - zasobnik implementovany jako dvojity spojovy seznam. Pokud pouziju stranku tak se presune na vrchol. Nevic pouzivana stranka je na vrcholu, nejmin pouzivana je na dne. Vse je v O(1).
- NRU — Not Recently Used – stránky mají čítač přístupu a špinavosti, čítač přístupu se občas maže, pak se první vyhazují stránky nečtené a čisté, pak nečtené a špinavé atd.
- One Hand Clock – kruhový seznam, při nutnosti vyhodit stránku se pustí ručička, která nuluje access bit pokud je nastavený a vyhazuje stránku pokud je nulový.
- Two Hand Clock – první ručička maže accessed bity, druhá (v pevném odstupu za ní) vyhazuje stránky s nulovým bitem, odstup ručiček určuje agresivitu.

- LFU — Least Frequently Used — stranky maji citac pristupu. Hlavni nedostatek je ten ze stranka hodne pouzita minule a uz nikdy nepouzita v budoucnu ma malo sanci na vyhozeni.

NRU, one hand clock a two hand clock algoritmy muzou byt chapane jako aproximace LRU, ktery je aproximaci OPT. FIFO a LFU jsou na tom nejhure.

Inverzni strankovani

inverzni strankovani (inverted page tables) — prevazne u 64bit. architektur napr. UltraSPARC 64 a Power PC — kvuli velikosti virtualniho prostoru organizuje se adresovani pres ramce fyzicke pameti, ne pres stranky.

Problemy:

- Pristupuje se na virtualni adresu
  - naivni reseni — linearni prohledavani polozek (protoze neumime rychle urcit ktery fyzicky ramec odpovida virtualni adrese)
  - efektivni reseni — pouzije se hashovani — nejdrive sahneme na hash table a tak zjistime kde hledat polozku v page table.
- Sdileni pameti. Uz to nejde tak jako u primych page table — fyzicky ramec odkazuje na virtualni stranku jako 1-1. Pouzije se stejna virtualni adresa stranky ve vsech ASID (???). Anglicky receno: " ... allow the page table to contain only one mapping of a virtual address to the shared physical address.".

Working set – stránky, které proces právě používá. Když běží moc aplikací, tak se jejich working sets nevejdou do paměti, každé přepnutí procesu vede k mohutnému swapování a vše je v háji. Tomu se říká thrashing.

Segmentace

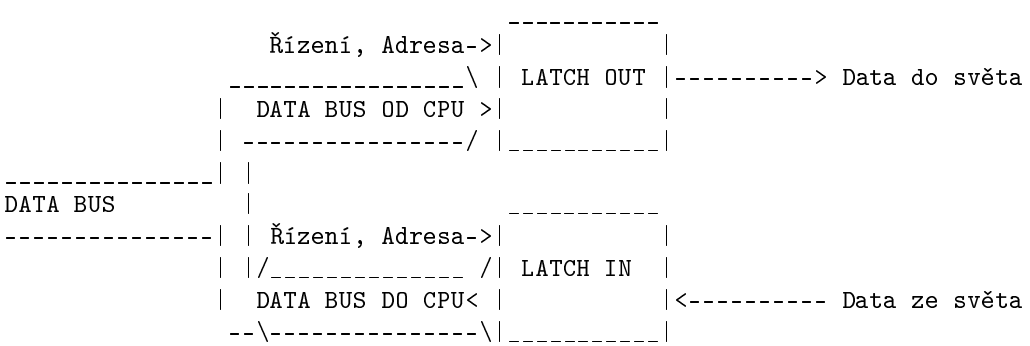
Paměť je rozdělena na segmenty. Program přistupuje na adresy rozdělené na segment/offset. Segmenty mohou být různě velké, a kromě umístení ve fyzické paměti mají i příznaky co na ně může sahat. Segmenty jdou přesouvat i zvětšovat aniž by o tom aplikace musela nezbytně vědět (stačí je jinak přeházet v paměti, číslo segmentu i offset v kódu se nemění).

Stránkování + segmentace: Kazdy proces ma iluzi pro sebe vyhrazeneho prostoru — segmentu. Pro hledani fyzicke stranky se pak pouziva i cislo segmentu prirazene procesu. MMU nejdrive v tabulce segmentu najde, kde zacina tabulka stranek pro dany segment a pak se pokracuje jako vyse.

23.6 Vstupně výstupní subsystémy, přerušení, DMA

- Art of Assembly — Chapter Seven The I/O Subsystem
- Všechno je připojeno vhodně na sběrnici (pomocí "latch" nebo flip-flop).
- Např. output port se může jevit jako místo v paměti, akorát má ještě spojení na vnější svět
- Jakmile je do latch zapsána datová posloupnost, zpřístupní je na sadě drátů co jdou z počítače.

Obecně jde Input i Output zařízení sdílející jednu adresu nakreslit následovně, na BUS je někde napojen CPU. Data nemusí jít úplně nutně do CPU, mohou jít třeba jinám (v režiji DMA)



Přerušení

see also Operační systémy (státnice)#Mechanismus přerušení v OS

Přerušení je událost, která mění sekvenci, ve které CPU vykonává instrukce; je generováno HW nebo SW. Je pouzivano pro praci s I/O, virtualni pameti, DMA, ...
Přerušení lze generovat:

- synchronně (trap) | vyvolala ho instrukce v běžícím programu, např. výpadek stránky
- asynchronně (interrupt) | vnější událost, např. dokončení V/V operace, porucha HW: : :
- výjimkou (exception) | nesprávné chování programu, např. dělení nulou

Postup při generování přerušení:

- 1) vznik příčiny, vyslání žádosti
- 2) rozhodnutí o přijetí či nepřijetí — přerušení lze maskovat — přerušení se uplatní až později
- 3) identifikace zdroje přerušení a určení adresy obslužného programu — rutina, ktera vyresi duvod preruseni
- 4) úschova aktuálního stavu procesoru na zásobník — je možno povolit víceúrovňové přerušení, ale je nutno zabránit zacyklení procesor obvykle při přijetí přerušení maskuje všechna přerušení
- 5) provedení obslužného programu
- 6) obnova stavu procesoru
- 7) návrat do přerušeného programu | obvykle pomocí zvláštní strojové instrukce, která obnoví to, co CPU uklidil

Programové přerušení se využívá ke zpřístupnění služeb OS, instrukce call se nepoužívá kvůli ochraně | vstupní body jsou při volání přes přerušení pevně dány.

DMA (Direct Memory Access)

see also Operační systémy (státnice)#DMA

- je způsob rychlého přenosu dat bez ucasti procesoru při I/O operacích přesunech dat v paměti, zobrazování, refreshi . . .
- Přenos není řízen strojovými instrukcemi, ale řadičem DMA.
- Přenos probíhá přímo mezi pamětí a I/O zařízením, procesor je odpojen od sběrnic.
- Jak se řadič DMA a procesor dohodnou, kdy provést přenos?
  - dávkový režim — řadič o sběrnice žádá, po dobu přenosu procesor na sběrnici nesahá, pak dostane od řadiče zpávu že je hotovo
  - kradení cyklů — řadič je schopen uspat procesor, provést přenos a pak ho probudit; náročné na hardware, nejde na dlouho
  - transparentní režim — řadič pozná, kdy procesor nepracuje se sběrnicemi, a v této době provede přenos — procesor o tom ani neví

23.7 Způsoby obsluhy periferií

Metody

- polling, periodické dotazování na stav zařízení
- Přerušení, zařízení samo zatahá procesor za nožičku, (většinou verkrze řadič přerušení)
- DMA, to je rozebráno výše samostatně

Abstrakce

- Většinou se programátorovi zařízení jeví jako stream nebo jako soubor.

Typy přístupu

- Adresované (Memmory mapped), jeví se jako nějaká část paměti

  - Přímé, adresa je součást instrukce
  - Nepřímé, součástí instrukce je adresa registru, kde je adresa dat

- Port mapped, k přístupu se používá specielní instrukce (wen:Input/output), `http://webster.cs.ucr.edu/AoA/Windows/HTML/IO.html`
  - Intel 80x86 IN and OUT instrukce

23.8 Vstupně-výstupní topologie

- daisy chain (řetězec počítač – zařízení – další zařízení – poslední zařízení)
  - SCSI se tak fyzicky zapojuje, i když elektricky je to sběrnice
  - MIDI má takovou formu (viz `http://www.root.cz/clanky/rozhrani-midi-na-osobnich-pocitacich/`)
- sběrnice (všechno visí na jednom drátu)
- stromová struktura (USB)

23.9 Sběrnice a jejich řízení (SCSI, USB, AGP, ...)

Dřív prostě dráty, obsahovaly následující linky:

- datové (přenáší data)
- adresové (kam mají data jít)
- řídící (režie přenosu)

Jde o sdílené médium, takže je potřeba nějak zajistit, aby nedocházelo ke kolizím.
přenos:

- synchronní (podle předem daných hodin)
- asynchronní (signál: teď bude zpráva, pak je zpráva)

Ke sběrnici se jednoduše přidávají další (i vcelku různá) zařízení, ale je to potenciální bottleneck (všechno na sběrnici se musí bavit stejnou rychlostí).
Přenos dat:

- s účastí CPU, které to řídí
- bez účasti CPU, řízeno řadičem (třeba DMA), nebo přímo zařízením (bus mastering)

Původně CPU přímo na sběrnici, později oddělena CPU/pamět na rychlejší, a za řadičem pomalejší zbytek. K rychlejším komponentám se pak přidala i grafická karta na vyhrazené sběrnici AGP nebo PCIe.

Sériové sběrnice

Přenáší najednou jen jeden bit (víc jen v nějakém složitějším kódování).
Výhody: jednodušší (nemusí sychronizovat mezi jednotlivými kanály), menší nároky na místo; dá se dosahovat vyšších přenosových rychlostí na kanál. Některé (PCI Express) začínají používat full-duplex p2p spojení, číž eliminují kolize.
Zástupci:

- USB
- FireWire
- SATA
- PCI Express

Paralelní sběrnice

Přenáši se najednou více bitů (asi ve více vodičích, tedy)
Zástupci:

- AGP
- ISA
- PCI
- ATA (aka IDE)
- SCSI

SCSI

- SCSI (Small Computer System Interface) je standardní rozhraní a sada příkazů pro výměnu dat mezi externími nebo interními počítačovými zařízeními a počítačovou sběrnicí. SCSI se vyslovuje „skazi “. Komunikace probíhá mezi **iniciátorem** a **cílem** pomocí příkazů.
- SCSI commands are sent in a Command Descriptor Block (CDB). The CDB consists of a one byte operation code followed by five or more bytes containing command-specific parameters.
- There are 4 categories of SCSI commands: N (non-data), W (writing data from initiator to target), R (reading data), and B (bidirectional). There are about 60 different SCSI commands.

AGP

- AGP (Accelerated Graphics Port) je PCI sběrnice, které běží na 66 MHz, a data přenáší na vzestupné i na sestupné hraně hodin, čimž zdvojuje rychlost přenosu. Obrázek v Ceresovo (`http://dsrg.mff.cuni.cz/~ceres/sch/osy/notes.php`) handoutech.

USB

- Mezi dvěma zařízeními vždz jen jeden kabel, jsou tam HUBy. Logicky je to tedy sběrnice, ale fyzicky tam nic sdíleného není. Jedná se o strom, v jeho kořenu je jedno zařízení připojené na vnitřní sběrnici, pracující v režimu MASTER. ve stromě může být celkem 127 jiných zařízení v režimu SLAVE.
- Řešení napájení z USB
- Zařízení, která se dají připojit na USB spadají do tříd. Tj, OS po připojení a resetu zařízení umí rozpoznat z jaké třídy a podtřídy zařízení je a podle toho zvolit ovladač. Ten je stejný bez ohledu na výrobce.

23.10 Mezipočítačová komunikace

pres site s pouzitim nejake sitove architektury — dnes prevazne TCP/IP.

Stejne vrstvy sitove architektury na ruznych pocitacich spolu “jako” komunikuji (rovnobezna komunikace). Ve skutecnosti kazda vrstva krome fyzicke vyuziva sluzeb te pod ni. Fyzicka vrstva umi posilat 0 a 1 nekam.

Kazda dve zarizeni komunikuji pomoci nejakeho (predem znameho) protokolu — “jazyka” s nejakou gramatikou — napr. IP, TCP, UDP, HTTP, FTP, ...

Komunikujici strany si predavaji PDU — protocol data unit, ktery ma vzdy dve slozky (hlavicku a obsah), i kdyz se muze jmenovat ruzne (packet, ramec, http data request, ...). Hlavicka obsahuje rezijni informace, napr. druh dat, adresa odesilatele a prijemce, poradove cislo, ...

23.11 Sériové a paralelní kanály

Asi viz #Sběrnice a jejich řízení (SCSI, USB, AGP, ...)

RS-232, Centronics, USB

- Seriovy port RS-232C, seriova sběrnice, viz `http://www.root.cz/clanky/komunikace-pomoci-serioveho-portu-rs-23`
  - Nemaj vyvedený samostatný hodinový signál pro synchronizaci
  - Posílá se start a stop bit, mezi ně posloupnost bitů, velké nároky na přesnost krystalky na obou stranách
  - Nutné jen dva datové vodiče, (tři s “nulákem”), volitelně a řídícími vodiči pro HW řízení toku, handshaking.
  - Simplexní, poloduplexní či plně duplexní asynchronní přenos dat

- Paralelní port, (Centronics rozhraní, LPT Port), viz `http://www.root.cz/clanky/paralelni-port-a-rozhrani-centro`
  - Konektor s dvaceti pěti piny, dvě řady (13+12), označení DB25
  - Osm datových vodičů
  - Čtyři řídicí vodiče
  - Pět stavových vodičů, kterými připojené zařízení posílá počítači zpět informaci o svém stavu
  - Připojen přes kartu např na PCI sběrnici

23.12 Modemy

see also wen:Modem

Modem je zarizeni pouzivane k prevodu digitalniho signalu na analogovy, jeho kodovani pro prenos pres metalicke medium a konverze zpet do digitalni podoby na druhe strane linky.

MOdulator a DEModulator, odtud modem. Modulovani je "nanaseni" signalu nesouciho informaci na nejakou nosnou vlnu, ktera se dobre siri mediem. Samotna nosna (typicky sinusoida) nenese informaci. Informace se uklada az do zmen teto vlny tzv. modulovanim.

- baseband — modulují do změn napětí/proudu. Unipolární (0V-5V), bipolární(-5V-5V), Manchester(náběžná hrana, sestupná hrana), Diferenciální Manchester(byla změna, nebyla změna).
- broadband — Modulace na nosnou vlnu, $y = A \cdot sin(\omega t + \varphi)$, tj tři možnosti co měnit (amplituda, frekvence, posunutí=fáze)

Pouziva se frekvencni, amplitudova a fazova modulace. Presneji kombinace techto tri. Napr. QAM — kvadraturni amplitudova modulace je kombinaci 3 ruznych amplitud a 12 fazovych posunu, coz tvori celkem 36 stavu. Pro snazsi rozpoznatelnost se ale pouziva jen 16 z nich s nejvyssi vzajemnou "vzdalenosti". Prenosova rychlost = modulacni rychlost (kolikrat za sekundu se zmeni signal) * log2 n. Kde n je pocet rozpoznatelnych stavu (16 u QAM)

Baud = počet změn stavu za sec != bps (bit per second)... modemy totiz mohou mit vic stavu.

Shanonuv theorem

$$C = B \log_2 \left(1 + \frac{S}{N}\right)$$

where

C is the channel capacity in bits per second;

B is the bandwidth (signal processing) of the channel in hertz;

S is the total signal power over the bandwidth, measured in watt or volt^2;

N is the total noise power over the bandwidth, measured in watt or volt^2; and

S/N is the signal-to-noise ratio (SNR) or the carrier-to-noise ratio (CNR) of the communication signal to the Gaussian noise interference expressed as a linear power ratio (not as logarithmic decibels]]).

Media

- drátová (kabely a optika)
- bezdrátová (wifi, bluetooth,...)
  - narrowband režim (malý rozsah frekvencí, velký odstup signálu a šumu)
  - spreadspectrum režim (velký rozsah frekvencí)
    - Frekvency hopping (změny frekvence)
    - DSSS (Direct Sequece Spread Spectrum), na každý bit sekvence bitů (chip) a z ní se pak něco vyXORuje (WiFi, GPS)

Multiplexing

- Frekvenční
- Časový
- Kódový na DSSS (chiping kódy)

Realita

- 33,6 kbps, protože šířka pásma elefonní linky je 3,1 KHz, AT příkazy
- ISDN (kde je všecko digitální, modem není modem)
- ADSL, subkanály, na každém je nosná frekvence, QAM

23.13 Topologie sítí

see also wen:Network topology

- point-to-point – permanentní nebo spojované okruhy (klasická telefonní síť)
- sběrnice – drát na kterém jsou všichni (původní Ethernet)
- kruh (Token Ring, FDDI – kruh může být koncentrovaný ve středu hvězdy – see wen:Media Access Unit)
- strom (Ethernet 10BASE-T a ty po něm, 100VG-AnyLAN – alternativní varianta 100Mbit Ethernetu)
- hvězda (jednotlivý segment nového Ethernetu)
- klika (každý s každým – asi nějaké bezdráty)
- mesh (ad-hoc wireless sítě – 802.11s, OLPC laptopy mají něco takového)

23.14 Přístupové metody

Resi problem pristupu ke sdilenemu prenosovemu mediu. resi se na urovni linkove vrstvy (podvrstvy MAC v RM ISO/OSI).

Varianty reseni, nekolik ruznych pristupu:

- deterministicke (rizene) vs. nedeterministicke (nerizene)
- centralizovane vs. distribuovane
- vylucujici kolize / zpracovavajici kolize / bez detekce i napravy

Srovnani:

- nerizene metody funguji lepe v mensich sitich s mensi vytizenosti media (LAN)
- rizene funguji lepe ve vetsich sitich s vetsi vytizenosti (paterni site).

Rizene centralizovane

arbitr vyzyva (polling) nebo je dotazovan (RTS/CTS – Request To Send / Clear To Send).

- Vyhody: arbitr muze menit strategie
- Nevyhody: vyssi rezie, single point of failure
- Priklad: 100VG-AnyLAN (pouze stromova topologie s hlavnim korenem)

Rizene distribuovane

jasna pravidla, vsechny uzly rovnocenne,

varianty:

- metody logickeho kruhu (tokenring) — siti putuje opravneni vysilat (token), ktery si uzly pravidelne predavaji.
- rezervacni metody (siti putuje specialni rezervacni ramec, ktery se pravidelne vyhodnocuje vsemi uzly). Ramec muze mit podobu bitmapy, ktera reprezentuje ktery uzel ma zajem o vysilani.
- Vyhody: distribuovanost — nestoji a nepada s jednim uzlem
- Nevyhody: jednotlive uzly toho musi umet vice
- Priklad: FDDI, TokenRing

Nerizene distribuovane

Metoda vysilej co chces (radiem), kdyz nedojde potvrzeni vysilej znovu. Vzniklo na Hawaiskych ostrovech, kde nebyla zadna pouzitelna infrastruktura.

- Vyhody: jednoduchý protokol
- Nevyhody: velka rezie na kolize a znovu posilani
- Priklad: wen:ALOHAnet

CSMA/CD, CSMA/CA a no-stress metody

"Poslechne" si (Carrier Sense), zda nekdo vysila a pokud ne, tak zacne — malo kolizi (ale muzou byt — Multiple Access) vzhledem k vysoke rychlosti sireni signalu. Pokud nekdo vysila, tak pocka az skonci a pak zkusi s pravdepodobnosti p (v zavislosti na metode, pro Ethernet p=1) vysilat znovu. Pokud dojde ke kolizi (Colision Detection), tak se chvili vysila JAM signal pro utvrzeni kolize. Ruzne metody pak muzou zkusit hned vysilat znovu, nebo chvili mlcet. V Ethernetu se oba (vsechny) kolizni uzly odmlci na nahodne zvolenou dobu a pak zacnou vysilat znovu (pokud dojde opet ke kolizi, tak se zdvojnasobi interval, ze ktereho si voli nahodnou dobu k odmlceni).

Na podobném principu funguje i CSMA/CA, která se snaží předcházet kolizím. Poslouchá, počká, nechá rozestup a pak vysílá.

"No-stres" metoda nic neřeší, pokud nepřijde potvrzení, tak je prostě neco špatně a pošle se to znova.

- vyhody: mohou byt velmi efektivni — mala rezie u nizke zateze
- nevyhody: nezarucuji vysledek (mohou byt kolize do te doby, nez to uzel vzda), pri vyssi zatezi prenosoveho media se velmi zvysuje rezie
- Priklad: Ethernet

Kolizni domena — kam az se siri kolize — repeatery siri, bridge, routery ji ohranicuji
Celkove se ocekava nizke vyuziti prenosoveho media a tim i malo kolizi

Bezdratove pristupove metody

- IEEE 802.11 — "casta" ztrata ramcu kvuli spatnemu signalu, neni mozne zjistit ze nastala kolize behem vysilani (radio je poloduplex). Vzdy ceka na ACK ramec, aby se ujistil ze prenos probehl uspesne. Problem "skryte" stanice (A chcevysilat na B ale nevidi ze na nej vysila C protoze signal z C nedosahuje) a predsunute stanice (B chce vysilat na C ale na nekoho vysila A takze medium je 'obsazeno' i kdyz na C by vysilat mohl — ale nevi ze A na C nevidi a tedy nejde o kolizi).
  - Distributed coordination function (DCF), CSMA/CA (Colision Avoidance). 0-perzistentni metoda — pokud nekdo vysila odmlci se na nahodnou dobu. Stale se muze dojit ke kolizi, napr. problem skryteho uzlu (tzn. neni 100% CA metoda). Povinna metoda dle IEEE 802.11
  - RTS/CTS `http://www.marigold.cz/wifi/doku.php/problem_skryteho_uzlu`. Snazi resit problem skryteho uzlu. Uzel X vysila Request-to-Send signal, uzly co jsou v dosahu zachyti signal RTS, nastavi svuj network allocation vector (stopky) a vyslou Clear-to-Send signal ktery zachyti i ty uzly ktere jsou mimo dosah uzlu X.
  - Point coordination function (PCF). Centralizovana metoda rizeni pristpu access pointem (AP).
- Bluetooth — po navazani spojeni preskakuje rychle mezi vysilacimi frekvencemi a tim si "zarucuje" malou pravdepodobnost kolize (u prenosu hlasu nevadi sem tam nedorucenost)

23.15 Síťové technologie — ATM, FDDI, FastEthernet, beztrátové technologie

ATM

see also wen:Asynchronous Transfer Mode

Prenosova technologie pouzivana v nekterych paternich sitich, převážně telekomunikačních společností. Je orientovana na QoS, je draha, jen spojovana, nespolehlivá, nema broadcast, ale zase nemá omezenou prenosovou rychlost. Puvodem ze sveta spoju, ale respektuje cast. i svet pocitacu. Je to bitová roura, žádné potvrzování a řízení toku.

ATM pracuje s bunkami dat. Maji vzdy 53B, z toho 5 je hlavicka, 48 naklad.

Nabizi vyssim vrstvam nekolik trid sluzeb:

- CBR — constant bit rate — garantuje celou kapacitu — emuluje drat — rezervuje max, vhodne pro tel. ustredny, nekomprimovana multimedia
- VBR — variable br — garantuje co, co prenos prave potrebuje — rezervuje max — nevyuzite pasmo vraci (narozdil od CBR), vhodne pro komprimovana multimedia
- ABR — available br — rezervuje min
- UBR — negarantuje nic — best effort — co zbyde

Nabizi virtualni okruhy — snazsi routovani pro ustredny, presmerovani, pokud cesta vypadne, nevyhodou je nutnost dvojiho protokolu aby to fungovalo (jeden pro komunikaci s koncovymi uzly, jeden pro vnitrni komunikaci ustreden mezi sebou).

Z hlediska vrstev je ATM nad fyzickou vrstvou a nad sebou jeste potrebuje AAL — (ATM Adaptation Layer), jejiz hlavnim ukolem je rozsekavat data na kusy po 44-48B pro prenos pres ATM. AAL ma casti AAL1-4 zhruba odpovidajici tridam sluzeb (viz vyse). AAL5 je specializovan na pocitacove prenosy (velke datagramy, nespolehlive, burstmod).

VPI a VCI — virtual path se přepisuje při přechodu přes ústřednu, VCI ne.

- vyhody: rychla, snazi se byt univerzalni
- nevyhody: tezkopadna, draha, IP na ATM není moc efektivní
- Priklad: paterni sit ADSL Ceskeho Telecomu (O2)

FDDI

see also wen:Fiber distributed data interface

Fiber Distributed Data Interface — data prenasena optikou. Nejstarsi vysokorychlostni (100Mbps) technologie, vhodna pro paterni site. Topologii dvojity ring, druhy pro zalohovani, pokud funguji oba, tak max. prenosova rychlost 200Mbps. V praxi se pouziva dvojity kruh stromu, kdy do kruhu jsou zapojeny treba routery a k nim stromem obyc. pocitace plus dalsi aktivni prvky.

Pouziva token passing pro rizeni pristupu. Moznost provozovat na dlouhe vzdalenosti (100km), pro pripojeni na kratke vznikla z ekonomickych duvodu varianta CDDI (Copper DDI). Existuje i novejsi FDDI2, ale ani ta se moc nepouziva. Celkove jde o do budoucna pravdepodobne mrtvou technologii, používá se spíš Fast Ethernet nebo Gigabit Ethernet.

Dva typy uzlů, co se připojují

- DAS oba okruhy jimy přiímo prochází
- SAS zapojené pomocí konektoru

Optické přemostění, pokud je připojená stanice vypnutá

FastEthernet

dnes verze ethernetu pracujici 100Mbps a vice (Gb Ethernet, 10Gb Ethernet), puvodne jen 100BaseT — prvni 100MBps CSMA/CD standard. Vychazi z 10Mbps, jen vse zrychluje a zkracuje intervaly. Verze, ktera i meni mnohe veci dostala nazev 100VG-AnyLAN.

Ethernet pouziva 48bitove adresy — MAC adresy. Puvodne mely byt nemenne, dnes je vetsinou lze menit. Prvni tri byty jsou OUI (organizationally unique identifier) — identifikator organizace, dalsi tri identifikator zarizeni. Mely by byt unikatni na celem svete (coz zjevne nelze zarucit, kdyz je uz lze uzivatelsky menit).

Ethernet je vrstvou na urovni sitoveho rozhrani. Ethernetove ramce je treba rozlisovat na ramce na urovani MAC (nizsi podvrstva) a LLC (vyssi podvrstva). Tj. Linková vrstva ISO-OSI se rozpadá pro ethernet na LLC (Logical Link Control) a MAC (Media Access Controll) podvrstvu Ramec ma 22B hlavicku a az 42-1500B payload.

Struktura packetu

```
-------------------------------------------------------
|PREAMBULE| DEST | SOURCE | TYP | PAYLOAD | CHECKSUM |
-------------------------------------------------------
```

100base-tx kodovani — misto Manchesteru (Ethernet) se zaclo pouzivat kodovani 4b/5b (4 informace, 5 preneseno, tj vždy se posílá pětice s alespoň dvěma jedničkama). Zpetna "kompatibilita" — vsechny sitove prvky ktere umi 100Mb se dokazou prepnout i na 10Mb. Puvodne poloduplex, ale rozsiren na plne duplexni (vysilani muze probihat obema smery) za predpokladu zadnych opakovacu a jen switchu krome koncovych stanic — uz neni pouzivano sdilene medium — kazda stanice ma primy spoj na switch a tim odpada nutnost CSMA/CD.

Rozpoznání 10 vs 100 MB ethetnetu na základě FastLinkPulse, 17-33 pásků o 100 ns (který desetimegabit detekuje jako jeden pulz?)

Mozna delka kabelu uz neni omezena technologii na 200m, ale jen utlumem kabelu.

Gigabit ethernet

Optikou nebo kroucenou dvoulinkou (kat. 5, 4 pary). Varianty poloduplex (s CSMA/CD) i fullduplex (ten se stava technologii vhodnou i jinde nez v LAN). Jiz jen dvoubodove spoje (pres switche se realizuji ruzne topologie).

10Gb ethernet — predpoklada se jen optika a jen fullduplex.

bezdrátové technologie

Je k dispozici omezeny rozsah pouzitelnych frekvenci.

- licencovane (nabizi operatori) provozovatel ma exkluzivni prava na vysilani, zadne ruseni, vydava CTU — 900MHz, 1.8GHz GSM, UMTS, CDMA
- nelicencovane, musi jen respektovat omezeni o max. vykonu; jednotlivi provozovatele si mohou rusit signal — wifi (2,4; 5GHz), WiMAX, bluetooth

druhy deleni bezdratovych siti

- podle druhu mobility (wifi, wimax vs GSM, CDMA vs Iridium; satelitni telefon — primo "bts")
- podle dosahu — cordless, wireless, satelitni
- podle druhu prenosu (1:1, 1:n)

WLAN standardy IEEE 802.11

- 802.11a – 5GHz, 54Mbit/s
- 802.11b – WiFi, 2.4GHz, 11Mbit/s
- 802.11g – 2.4GHz, 54Mbit/s

Rámce v 802.11

- řídící (např ACK)
- administrační (např authentizační)
- datové

Přístupové metody

- DCF, Distributed Coordination Function, založená na CSMA-CA
- PCF Point Coordination Function, koordinace založená na koordinátorovi
  - cez AP, mix periód DCF a CFP (vtedy rozosiela pakety staniciam, ze mozu vysielat)

BSS, prepojene cez (Wireless-)DS; spojenim BSS => ExtendedSS (mobilita)

- SSID, BSSID

rámce: PLCP (rychlost prenosu) | Mac (RTS/CTS, ACK) | management (probe, asociacia,autentizacia) | data frames

4 adresy v MAC (sndr, rcvr + bity to DS, from DS)

kolizie

- 0-perzistencia (CSMA/CA — ACK)

prekryvajuce sa frekv. pasma

DHSS (bit => napr. bakerov kod; vo wifi nie kvoli multiplexu — ale kvoli sumu)

OFDM (ortho. freq. division multiplex — prekryvajuce sa pasma, v kazdom pasme iny nosny signal)

PBCC

23.16 RM ISO/OSI

see also wen:OSI model

Referencni model ISO/OSI (International Organisation for Standardisation / Open Systems Interconnection)

Pokus vytvorit univerzalni sitovou architekturu standardizacni agenturou. Reseni od "zeleneho stolu". Nebylo plne dodelano, protokoly se dodelavaly az pozdeji postupne. Zacal "velkyma ocima", ze kterych bylo postupne ustupovano kvuli neimplementovatelnosti celeho modelu. Ze zacatku hodne prosazovan statnimi institucemi, ktere ve svych zakazkach vyzadovaly kompatibilitu.

RM ISO OSI obsahuje sedm vrstev, kazda vrstva ke svemu fungovani pouziva vrstvy primo pod ni na stejnem pocitaci, mezi pocitaci fakticky komunikuje jen fyzicka vrstva.

1. Fyzicka — prenasi bity. Kvuli tomu resi kodovani, modulaci, synchronizaci, ... na jeji urovni se rozlisuje paralelni a seriovy prenos, synchronni a asynchronni,...
2. Linkova — prenasi ramce k primym sousedum. Kvuli pretizenosti se rozpadla na dve podvrstvy — LLC a MAC. Zajistuje synchronizaci na urovni ramcu (detekce zacatku a konce ramcu). Muze fungovat spojovane i nespojovane, spolehlive i nespolehlive, best eff. i QoS. Ridi tok dat a pristup ke sdilenemu mediu
   - MAC dělá kanál na sdíleném médiu, adresování, řeší kolize
   - LLC multiplexing, řízení toku (kdy vysílat aby se druhý konec neucpal)
3. Sitova vrstva — prenasi datagramy pomoci smerovani do cele site. Muze pouzivat ruzne routovaci algoritmy. Je to nejvyssi vrstva, kterou musi mit smerovace (routery). Routing — hledani cesty k adresatovi, forwarding — poslani packetu tou cestou.
4. Transportni vrstva — muze pridavat sluzby — napr. nizsi vrstvy (patrici nekomu jinemu) nabizi jen nespolehlivou komunikaci a transportni vrstva doda (pripadne zvysi) spolehlivost. Pr. TCP (spolehlivy) nad IP (nespolehlivy) (ten priklad neni z ISO OSI, ale ze transportni vrstvy rodiny TCP/IP). Transportni vrstav pridava "delitelnost" uzlu — uz nestaci jen adresa, ale je treba i port pro identifikaci adresata.
5. Relacni vrstva — muze zajistovat sifrovani, podporu transakci, sessions. Je to nejvice kritizovana soucast RM ISO/OSI. V TCP/IP neni zastoupena a řeší si to aplikace. `http://www.earchiv.cz/a92/a225c110.php3`
6. Prezentacni vrstva — prevod dat do/z tvaru vhodneho k prenaseni — napr. kodovani (ascii, ebcdic, little endian vs big endian) linearizace (dvourozmerne pole -> 1D),...
7. Aplikacni vrstva — puvodne mela obsahova aplikace, ale nakonec jen jadro aplikaci, ktere ma smysl standardizovat — napr. prenos el. posty.

nevyhody modelu: vyroben od zeleneho stolu, prilis slozity, nektere funkce opakovany v ruznych vrstvach, upřednostnuje spojovane spolehlive sluzby sveta spoju, maly duraz na propojovani siti.

23.17 Aktivní prvky (bridge, routery)

Směrovač (router)

Zarizeni fungujici na urovni sitove vrstvy. Propojuje dve nebo vice siti (ma dve nebo vice sitovych rozhrani — adres). Funguje v prostredi prepojovani packetu. Stara se o smerovani packetu (routing) i jejich posilani (forwarding). Sklada se z prepojovaciho pole a smerovaciho procesoru. Procesor urcuje smer dalsiho putovani packetu, pres prepojovaci pole se packet fyzicky prenese z vstupni fronty do vystupni fronty.

Smerovac pro urceni dalsi cesty pouziva smerovaci tabulky (routing tables), pripadne je nepouziva u "nouzovych" metod typu flood nebo random. Smerovaci tabulka obsahuje trojice jake site, jakou cenou (v nejake predem stanovene metrice), pres jakeho primeho souseda.

Adaptivni algoritmy prubezne upravuji smer. tabulky, neadaptivni ne.

metody smerovani:

- flood — posli vsem krome zdroje — pro aktualizacni informace, pripadne ve vojenskych sitich.
- random — posli komukoli — kdyz se mi zacinaji zahlcovat buffery a nejde to poslat tam, kam to ma jit
- distribuovane smerovani — Nejcasteji uzly spolupracuji na hledani optimalni cesty. Dnes prevazne link state routing metoda.
  - OSPF (open shortest path first) protokol. Poslu vsem! vzdalenosti k primym sousedum a kazdy uzel si z nich pocita nejkratsi cesty sam. Ani tohle neni kvuli velikosti tabulek dostatecne pro opravdu VELKE site.
  - hiearchicke smerovani — rozdelim sit na bloky, ze kterych se nebudou sirit detailni smerovaci informace, ale jen informace typu informace ve smyslu: "pres mne jsou dostupné síte X.Y.Z až X.Y.W". Tuto sumarizaci provadi "hranicni smerovac". Pres paterni sit je nutno pouzit exterior gateway protokol (BGP), v "moji" siti nejaky interior gateway p. (třeba OSPF, nebo něco statického)

Most (bridge), přepínač (switch)

Zarizeni na linkove vrstve. spojujici jednotlive pocitace do "site". Muze byt zapojeno k sobe i vice switchu. Jednotlive site od sebe oddeluji routery. Bufferuje data a tim muze spojovat jednotlive segmenty site, ktere nepracuji stejnymi rychlostmi. Pracuje transparentne (na urovni linkove vrstvy), zastavuje kolize Ethernetu, podporuje broadcast.

Metodou zpetneho uceni se uci topologii nejblizsiho okoli (jen sit — po router). Nejdrive nevi nic a co dostane broadcastuje vsem. Zapamatuje si adresy tech co mu neco poslou a kdyz jim je neco v budoucnu smerovano, tak uz to posle jen jim. Problem moznych cyklu v siti (od A mi to prislo pres B i C) resi inteligentni switche vyhledanim kostry.

Pro posilani v ramci site se muze pouzivat source routing (tedy ne v Ethernetu; neplest s routing v sitove vrstve) — ramec ma v hlavicce navigaci — uplny itinerar uzlu, pres ktere vede cesta k cili. Ten se zjisti specialnim pruzkumnym packetem, ktery se pred tim poslal floodem a vratil se.

Rozdil mezi switchem a bridgem je v tom, ze switch umi jen stejne/podobne technologie (prepoji site 10 a 100mbit ehternet), kdezto bridge je pomalejsi ve sve funkci, ale za to treba spoji token ring a ethernet.

Opakovač (repeater), hub

Zarizeni fyzicke vrstvy, jen hloupy zesilovac signalu — co mu prijde z jedne strany vysle na druhou (nebo na všechny ostatní v případě hubu), a tim "prodluzuje" sit. Mnozstvi repeaterů zapojitelnych v Ethernetu je omezene (max 5 segmentů mezi dvěma stanicemi), aby se stihly detekovat kolize. Dulezite na repeateru je, ze uz se nepouziva, protoze se ze sdileneho media (sbernice — koax) preslo na point2point (kroucene dvoulinky do switche).

23.18 Síťový model TCP/IP, IPv6

Rodina protokolů (asi 100) TCP/IP, na které je postavený dnešní Internet. Ucelena predstava o sitove architekture a poctu a ukolech vrstev. Oproti RM ISO/OSI obsahuje jen ctyri vrstvy.

Vznikalo pomalu, postupnym pridavanim v akademickem prostredi, ale prosadilo se i v komercnim. Dnes je to nejrozsirenejsi sitova architektura fungujici nad jakoukoli linkovou technologii.

Protokoly byly vyvijeny jako definitivni reseni pro provoz vznikajiciho Internetu. Mely nahradit prozatimni Network Control Protocol Arpanetu — site financovane ministertvem obrany USA (Vint Cerf — otec internetu). Filozofie – musi to byt decentralizovane a nejak fungovat, kdyz cast fungovat prestane (bude znicena treba). Nebylo pozadovano zabezpeceni, mobilita... TCP/IP vymysleno tak, aby sly ruzne site s ruznymi technologiemi pripojovat k ARPANETu (a tim tvori Internet).

Vrstvy

1. Aplikacni vrstva — jednotne zaklady aplikaci — email, prenos souboru, http, xmpp, etc
2. Transportni vrstva — jednotne transportni protokoly — TCP a UDP
3. Sitova vrstva — prenosovy protokol IP
4. Vrstva sitoveho rozhrani — TCP/IP nedefinuje — at si tady je co chce, kdyz to bude poskytovat sluzby potrebne pro fungovani IP protokolu. napr. Ethernet, Token Ring, ATM, ...

porovnani s RM ISO/OSI:

- aplikacni vrstva ˜ aplikacni + prezentacni + relacni
- transportni vrstva ˜ transportni v.
- sitova v. (IP vrstva) ˜ sitova v.
- v. sitoveho rozhrani ˜ linkova + fyzicka

IP

Protokol IP – nespolehlivy, nespojovany, best effort, 32bitove adresy (IPv4). Vyssi vrstvy vidi pouze MTU — maximum transfer unit — velikost ramce pro danou technologii pod IP. Pokud ji nerespektuji, dochazi k fragmentaci (IP to rozdeli, ale to je plytvani).

IP adresy logicky dvouslozkove adresa site + adresa uzlu. Predpokladaji se dva typy uzlu v siti — koncove (pc, tiskarny,servery,...) a routery, ktere spojuji jednotlive "site". Adresa site byla puvodne vyjadrena bud 8 bity (trida A), 16 (trida B) nebo 24 (trida C) a jednotlivym zadatelum se pridelovaly jednotlive tridy.

S rozvojem pocitacu zacinaji adresy dochazet, coz se resi:

- Subnetting — deleni jiz pridelenych trid.
- CIDR – uz jde pridelovat jakoukoli 2^n mnozinu adres ne jen tri ruzne tridy.
- privatni IP adresy (10.*; 192.168.*) — jsou videt jen v ramci jedne site, ne pres router, do zbytku světa připojené přes NAT nebo různé proxy
- IPv6

ostatní protokoly a TLA

- TCP — spojovany spolehlivy (emuluje toto chovani nad nespojovanym nespolehlivym IP) protokol v transportni vrstve (vyuziva jej napr. SMTP, FTP, Telnet, HTTP)
- UDP — nespojovany nespolehlivy protokol v transportni vrstve (DHCP, RPC, NFS, větší část DNS), víceméně jen obal IP s porty
- RFC — public domain standardy TCP/IP, vznikaji az na zaklade fungujici technologie (firmy dnes predkladaji ke standardizaci sve technologie)
- Internet Socieety -> IAB (ridi standardizaci, vydava RFC)
- Webove standardy vydava W3C, kopie jsou vydavany jako RFC

Packet header analysis IP header Ethernet frames
TCP/IP obecně má:

- vyhody: efektivni, nevnucuje rezii, kterou nepotrebujeme
- nevyhody: nehodi se na nektere nove technologie (VOIP, Video on Demand), ktere by potrebovaly QoS, nikoli BestEffort (lze resit prioritizaci — MPLS nebo rezervaci RSVP – zajistuje rezervaci zdroju "pod" IP); neposkytuji zadne zabezpeceni (ze zadani)

Srovnání s RM ISO/OSI

TCP/IP	ISO/OSI
prijme jedn. reseni, pak pridava	zacne na velkem, pak ubira, protoze nezvlada
prijima jen realizovane veci	neni nutne overeni realizovatelnosti
standardy jsou volne dostupne	standardy jsou prodavany (draze)

IPv6

Naslednik IPv4 (ten prevazne pouzivany nyni). Asi nejdulezitejsi zmenou je rozsireni adresoveho prostoru z 32 bitu na 128 a tim vyreseni problemu s nedostatkem adres "na vzdy".

Obsahuje ale i jine nove moznosti:

- multicast přímo ve specifikaci
- mobilita (dnes zatím nepoužito)
- link-local adresy
- IPSec – šifrování autentizace přímo ve specifikaci
- autokonfigurace (postup pro získání link-local adresy, pak router solicitation, ...)

Nejedna se o zasadni zmeny od IPv4, spise, podle filosofie protokolu TCP/IP o vylepseni, ktere si "zadaji uzivatele".
Další odlišnosti:

- nemá checksum v hlavičce (kde se neustále mění TTL/hop limit), místo toho spoléhá na vrstvy okolo. routery tak nemusejí neustále počítat měnící se součty
- pakety se nefragmentují na cestě, ale jen u odesílatele, předpokládá se použití Path MTU discovery

Adresace v ipv6

z wcs:ipv6 Adresy IPv6 se dělí do tří kategorií[1]1RFC 2373 — 'Architektura adresování IPv6':

- unicast adresy
- multicast adresy
- anycast adresy

Unicast adresa reprezentuje jednotlivé síťové rozhraní. Paket zaslaný na unicast adresu je doručen konkrétnímu počítači. Následující typy adres jsou IPv6 unicast adresy:

- globální unicast adresy
- adresy místní linky
- adresy místní stránky
- unikátní lokální IPv6 unicast adresy
- speciální adresy

Multicast adresy jsou používány k definování množiny rozhraní obvykle patřících různým uzlům, nikoli pouze jednomu. Paket zaslaný na multicast adresu je protokolem doručen všem rozhraním určeným touto adresou. Multicast adresy mají prefix FF00::/8 a jejich druhý oktet určuje dosah adresy, tzn. rozsah v jakém je multicast adresa zviditelněna. Běžně využívány jsou rozsahy místní linky (0x2), místní stránky (0x5) a globální (0xE). Anycast adresy jsou také přiřazeny více než jednomu rozhraní, patřící rozdílným uzlům. Nicméně paket vyslaný na anycast adresu je obvykle doručen pouze jednomu z členských rozhraní, typicky „nejbližšímu“ vzhledem k představě směrovacího protokolu o vzdálenosti. Anycast adresy nemohou být snadno identifikovány, mají strukturu bežné unicast adresy a liší se pouze zaváděním do směrovacího protokolu na více místech v síti.

Speciální adresy

Existuje množství adres které mají speciální význam v IPv6:

Místní linka

- `::/128` — adresa samých nul je nespecifikovaná adresa a je používána pouze v software.
- `::1/128` — adresa místní smyčky je adresa pro localhost. Pokud aplikace vyšle paket na tuto adresu, IPv6 paket je vrácen zpět na téhož hosta (odpovídá 127.0.0.1 v IPv4).
- `fe80::/10` — prefix místní linky udává platnost adresy pouze v místní fyzické lince. Analogické autokonfigurační IPv4 adrese 169.254.0.0/16.

Místní stránka

- `fc00::/7` — unikátní lokální adresy (ULA) jsou směrovatelné pouze v množině spolupracujících stránek. Adresy zahrnují 40 bitové pseudonáhodné číslo minimalizující nebezpečí konfiktu při sloučení stránek či úniku paketů.

IPv4

- `::ffff:0:0/96` — tento prefix se používá pro IPv4 mapované adresy (viz Mechanismy přechodu níže).
- `2002::/16` — pro adresování tunelu IPv6 v IPv4.

Multicast

- `ff00::/8` — použití pro multicast adresy

Užito v příkladech, neschváleno, či zastaralé

- `::/96` — nulový prefix se používal pro IPv4 kompatibilní adresy. Nyní zastaralý.
- `2001:db8::/32` — použito v dokumentaci. Kdekoliv je dán příklad IPv6 adresy, měl by mít tento prefix.
- `fec0::/10` — prefix místní stránky udává platnost adresy pouze uvnitř místní organizace. Použití bylo v září 2004 odmítnuto dle RFC 3879 a systémy nesmí podporovat tento speciální typ adres.

[1]

V protokolu IPv6 nejsou vymezeny žádné rozsahy adres pro broadcast — aplikace používají multicast ke skupině all-hosts

Zónové rejstříky

Určitý problém představují adresy lokální linky pro systémy s více než jedním rozhraním. Jelikož každé rozhraní může být připojeno k jiné síti a všechny adresy se zdají být na stejné podsíti, objevuje se nejednoznačnost neřešitelná směrovacími tabulkami Například host A má dvě rozhraní a těmto rozhraním jsou při aktivaci automaticky přiřazeny adresy místní linky : fe80::1/64 a fe80::2/64. Pouze jedno z rozhraní je připojeno do stejné fyzické sítě jako host B s adresou fe80::3/64. V případě že host A se pokusí o spojení s fe80::3, jak určí rozhraní které má použít? O tom pak rozhodují zónové rejstříky

23.19 Přenosové služby počítačových sítí

Spolehlivé a nespolehlivé

spolehliva sluzba ten, kdo data prenasi zodpovida za doruceni (proto je nutna detekce chyb, zadosti o opetovny prenos).

nespolehliva sluzba je ji jedno, zda se to doruci nebo ne. Se spolehlivosti je spojena nenulova rezie, ktera muze byt nezadouci. Napr. multimedialni prenos potrebuje dodavat pravidelne hodne dat a pripadna ztrata casti z nich boli mene nez zdrzeni vetsiny kvuli znovuposlani.

best effort typ prenosu, kdy se sit "snazi" a pokud to uz nejde, tak jsou pozadavky stejnomerne kraceny. pr. IP

Quality of Service Obecné oznacení pro variantu, kdy prenosová sí dokáže rozlišovat mezi jednotlivými prenosy a nabízet jim ruznou "kvalitu prenosu" (QoS). Muze nabizet garanci sluzeb (resi se rezervaci zdroju) a take nemusi.

Spojované a nespojované

spojovana komunikace

1. nejdrive se navaze spojeni v rámci navázání spojení je nalezena (a vyznacena) trasa penosu
2. probiha komunikace
3. spojeni je ukonceno, pripadne zdroje vraceny

- komunikace je stavova (alespoň neni spojeni / je spojeni). je treba osetrit prechod mezi jednotlivymi stavy a nestandardni situace (jeden ucastnik spadne apod.)
- zachovava poradi (packety se nemohou zprehazet, kdyz cestuji stejnou cestou)
- analogie — telefonni hovor (vytocim, mluvim, polozim)

Prepojovani okruhu – zpusob spojovaneho prenosu. Jednotlive uzly site "vyriznou" z dostupne kapacity tolik, o kolik si komunikujici strany reknou (a taky to pak nauctuji, rezervace je drahá). Vyrabi iluzi jednoho dratu mezi zdrojem a prijemcem, ktery ma vsude stejnou kapacitu. Odhadnutelne zpozdeni na ceste, vyuziva se ve svete telekomunikaci, protoze je vhodne pro multimedia, malo ve svete pocitacu (snad jen seriove komunikace).

nespojovaná komunikace

nenavazuji spojeni, neoveruji, ze druha strana existuje, neni treba ukoncovat spojeni. komunikace probiha formou posilani zprav — datagramu

- bezestavova
- každý datagram obsahuje v hlavicce plnou adresu prijemce
- nezarucuje poradi doruceni
- trasa pro datagram je hledana vzdy znovu
- analogie — listovni posta

Prepojovani packetu – nic se nevyhrazuje, stale je snaha vyuzit vsechny dostupne zdroje (best effort). Pouziva se prakticky ve vsech sitich, vhodne pro datove prenosy. Datagramy museji obsahovat adresy odesilatele a prijemce. Ma smysl jen blokovy prenos, proudovy se emuluje. Diky slozitejsi logice zpracovani v uzlech ma vetsi prenosove zpozdeni nez prepojovani okruhu a je nerovnomerne. Uzly funguji na principu Store & Forward. Muze fungovat spojovane (virtualni okruhy) i nespojovane (datagramova sluzba).

23.20 Přenos a sdílení dat

není moc jasné co tahle otázka vlastně znamená, následují nějaké spekulace

Z hlediska (lowlevel) techniky prenosu dat

proudovy prenos / blokovy prenos

- Proudovy prenos (stream)
  - predavano po jednotlivych b[iy]tech.
  - Zadna hlavicka, prijemce je ten na druhe strane kanalu. Predpoklada se spojovana komunikace.
- Blokovy prenos
  - prenaseno po blocich. Blok je nazyvan podle vrstvy, ve ktere je prevod realizovan,zpusobu prenosu nebo velikosti bloku:
    - packet — síťová vrstva
    - rámec (frame) — linkova vrstva
    - zpráva — aplikacni vrstva
    - datagram — obecny nespojovane prenaseny blok (ramec je datagram)
  - Velikost bloku je promenna, ale shora omezena

Podle synchronizace

- synchronni — hodinky zdroje a prijemce jsou stale aktualizovany (treba v ethernetu se posilaji data jen kazdou druhou dobu, ty druhe "kazde druhe (sudy/lichy) jsou pro synchronizaci)
- asynchronni — postrada jakoukoli synchronizaci, potrebuje 3hodnotovou logiku (“1”, “0”, “nic”), nepouziva se
- arytmicky — terminologicky zamenovan s asynchronnim — muze velke mezery mezi vysilanim, zesynchronizuji se vzdy na zacatku vysilaneho bloku (ktery ma jen 4-8b, coz vydrzi synchronizovane) “start” bitem.

Podle směrovosti

- simplex — jendosmerny kanal
- duplex — obousmerny,
- semiduplex — lze prenaset v obou smerech, ale ne najednou

Z hlediska prenosu dat pomoci aplikacnich protokolu

Kazdopadne musi obe strany dodrzovat nejaky dohodnuty protokol.

- pro prenos souboru — FTP — na soubory se divame jinak, jsou-li ulozeny lokalne nebo vzdalene. Uzivatel musi explicitne kopirovat data z jedne lokaci do druhe (get/put). Funguje typem klient/server. active/passive(vsechna spojen navazuje klient — kvuli fw) mod. port serveru 21 — prikazy, 20 — data, u klienta nahodny.
- pro sdileni souboru — NFS, AFS, SMB (wokenni sdileni) — soubory lze vnimat stejne (byt muzou mit ruznou dobu pristupu), o stehovani z jednotlivych lokaci se stara system.

Soubory jde posílat přenášet obří hromadou dalších protokolů. Pod sdílení se asi kromě síťových filesystémů dají zahrnout i FTP, web, různé content distribution networks a p2p věci jako je BitTorrent.

23.21 Elektronická pošta

služba, je moznost realizovat ji interne ruzne (smtp vs MS mail vs X.400). V internetu se pouziva SMTP. Krome protokolu pro prenos zprav mezi servery a klientem (SMTP), je treba dodrzovat jeste spravny format (rfc822 — hlavicka, telo, pripadne prilohy), zajistit zpusob stahovani zprav ze schranky na serveru (imap, pop3), rozsireni pro napr. narodni abecedy, urceni typu prilohy (MIME).

Kazda zprava ma:

- header — komu, od koho, predmet, datum poslani, ...
- telo — obsah
- prilohy

Přenos zprávy

1. sestavení zprávy, příkaz k odeslání
2. upload (odeslání) zprávy na poštovní server, pomocí SMTP
3. přenos zprávy mezi poštovními servery
   - pomocí protokolu SMTP
   - zpráva končí v poštovní schránce (mailboxu) příjemce
4. stažení zprávy z poštovní schránky do poštovního klienta
   - pomocí POP3/IMAP
5. čtení přijaté zprávy, v rámci poštovního klienta příjemce

- MX záznamy v DNS — pro kazdou domenu je definovan jeden nebo vice serveru, ktere pro ni prijimaji postu
- SMTP garantuje prenaseni jen 7bitovych znaku (ASCII), pro prenos 8bitovych je treba je prevest do 7bitu — jak to udelat resi MIME.
- IMAP vs POP3 — IMAP dovoluje pracovat s daty na serveru, je urcen pro stala pripojeni typu pevna linka/adsl. POP3 umi zpravy stahovat na lokal — vhodne pro modemisty

Spam

Jak na spam:

- Bayes filtry
- DNSBL
- greylisting
- SPF (Sender Policy Framework)

23.22 Služby zpřístupnění informací

Po Gopheru dnes WWW.

WWW

Sada HTML stránek a dalších médií okolo nich propojených odkazy. Data se přenášejí pomocí protokolu HTTP/HTTPS. Funguje na architektuře klient/server.

HTTP je bezestavový protokol (jen request/response), stavovost se do něj doplňuje typicky přes cookies. HTTP request je nejčastěji GET/POST/HEAD, ale jsou i PUT/DELETE a pár dalších (see wen:Hypertext Transfer Protocol#Request methods).

V odpověd přijde nějaký ze status kódů (200, 404, 302, ...) a obsah.

Proxy, P2P

proxy proxy server je jakysi prostrednik mezi klientem a serverem, ktery vyrizuje pozadavek. Klient posle pozadavek proxy serveru (o cemz uzivatel ani nemusi vedet a typicky si nemuze vybrat, zda proxy server pouzije ci nikoli) a ten jej vyridi bud ze sve cache nebo dotazem na dany server a pak preposlanim odpovedi. Pouziva se z duvodu bezpecnostnich (jediny bod site pristupny z internetu, vsechny lokalni pocitace jsou az "za nim"), vykonovym (cacheovanim muze usetrit spoustu prenosove kapacity), pripadne jinym (napr. povoleni jen urcitych stranek — pokus o pristup na jine proxy server nedovoli)

Peer to peer site (Peer2Peer, P2P) vymenne datove site (bittorrent, KaZaa, Napster, DirectConnect...), kde si data predavaji dva(vice) koncovych uzivatelu, nejsou stahovana z zadneho centralniho serveru.

23.23 Bezpečnost síťového přístupu, zabezpečené protokoly

Tady je stranka, ktera slusne pokryva toto tema. Jasne a do hloubky napsane: RAD security

Bezny sitovy provoz je tvoren v "dobre vire". Kdokoli je na trase, tak muze do zprav nakukovat, i je menit. To samozrejme neni pro nektere typy komunikaci dostatecne. Aby byla sitova komunikace bezpecna, mela by zajistovat tyto veci:

- identifikace (zjisteni identity ucastniku — ten druhy je tim, za koho se vydava),
- autorizace (overeni pristupovych prav — nelez/nesahej kam nemas)
- duvernost (sifrovani komunikace — nikdo si to neprecte)
- integritu (taky sifrovani — nikdo to nezmeni).

Protokoly pro bezpecny prenos informaci — z datoveho toku neni poznat, co je prenaseno. Resi se sifrovanim. Pro "uplne" (kazda sifra muze byt prolomena) bezpecne pouziti je nutne identifikatory ucastniku (fingerprinty, jmena-/hesla) prenest bezpecnou cestou (rucne lokalne opsat, nikoli stahnout nezabezpecene z webu). Bezpecnost zavisi na bezpecnosti pouzite sifry a korektni implementaci algoritmu na obou stranach.

Identita ucastniku a navazani spojeni se resi pomoci asymetrickeho sifrovani, vlastni vymena pak pomoci symetrickeho(pretoze symetricke su rychlejsia nez asymetricke). Identita je dana public klicem ucastnika (nebo jeho fingerprintem). To se typicky deje jen pro server a uzivatel svou identitu serveru "prokazuje" pozdeji pomoci jmena a hesla.

SSL, secure socket layer protokol v rodine tcp/ip slouzici k bezpecnemu provozu jinych protokolu. Je zalozen na sifrovani a muze pouzit nekolik ruznych algoritmu. Pouziti se sestava ze tri kroku: 1. Dohodnuti se na algoritmech, 2. autentikace ucastniku pomoci verejnych klicu (certifikátů), serveru povině a klienta volitelně 3. vymena dat pomoci symetrickych sifer. Myšlenaka byla vytvořit zabezpečené sockety typu berkley, původně hlavně pro protokol HTTP.

https — secure obdoba http protokolu jde o vymenu dat beznym http protokolem provozovanym nad SSL (secure socket layer)

ssh (secure shell) pro bezpecne terminalove spojeni se se vzdalenym serverem, dá se použít pro tunelování protokolů na TCP/IP, např HTTP.

sftp secure file transfer protocol, je nad ssh

FTP jde pomocí SSH zabezpečit řídící spojení, to je nad TCP, ale datové ne, to je nad UDP.

Do SSL/TLS se dá balit i SMTP, POP3, XMPP, ... Pro paranoiky jde i Tor a tak.

23.24 Překlad adres

NAT — network address translation — technika prekladu sitovych adres, ktery se deje na firewallu nebo routeru. Pri pouziti se vzdy projevi tyto cile/dusledky:

- setreni jmeneho prostoru IPv4 — privatni adresy mohou byt pouzity pro spoustu pocitacu za NATem.
- sdileni internetoveho pripojeni — internet provider poskytuje pouze jednu sitovou adresu a my chceme pripojit vice pocitacu.
- bezpecnost — pocitace za NATem nemaji verejnou adresu a neda se na ne z internetu poslat request, protoze skonci na NAT routeru.

Krome techto "vlastnosti", ktere se projevi pri kazdem pouziti NATu, lze tuto techniku pouzit i pro dosazeni jinych cilu:

- LoadBalancing — nekolik fyzickych serveru za jednou ip adresou a "NATovadlo :)", ktere rozdeluje pozadavky
- Failover — server a jeho backup za jednou ip adresou. Kdyz server selze, tak se zacne pouzivat backup, aniz by o tom kdokoli vedel.

priklad — dnesni domaci routery, iptables.

Rozliseni, ktery pocitac v privatni casti ma dostat odpoved na svuj pozadavek se dela pomoci cisel portu (proto nekdy nazyvano PAT) — router pro pozadavek pocitace prideli nejaky port a "z nej" to potom posle do internetu. Kdyz prijde odpoved na tento port, tak router pak vi, komu to ma preposlat. Preposilani obema smery se deje prepisovanim adres v hlavickach a ponechani tela (payload) packetu. Proto protokoly, ktere maji adresu i v tele (IPSec) nebo navazují dodatečná spojení směrem ke klientovi (active mód FTP) nefunguji za NATem (bez explicitni pomoci od NATovadla).

Routeru se dá nastavit aby některé požadavky na sebe přeposílal daným počítačům ve vnitřní síti (port forwarding). Toho se dá využít ke zpřístupnění některých zdrojů ve vnitřní síti ven (třeba pro p2p sítě ;-). Protokol UPnP umožňuje tyto tunely vytvořit zevnitř bez speciálního nastavování routeru.

23.25 Firewally

Firewall je zařízení slouzici ke kontrole a případnému povolení/zakázání nějakého síťového provozu. V podstatě odděluje vnější síť (typicky Internet) od vnitřní sítě (LAN nebo i jeden počítač). Lze řešit v HW i SW.

Ve firemní síti firewall třeba zakazuje veškerý HTTP provoz všemu kromě proxy serveru, a zabraňuje navazování spojení zvenku. Dále se filtruje třeba SMTP, nebo některý druh přístupu pouze z důvěryhodných sítí, ...

V jednom PC je firewall softwarove který kontroluje příchozí a odchozí spojení i na úrovni aplikací – můžu povolit spojení ven prohlížeči, ale už ho třeba zakázal malwaru.

23.26 Certifikáty

Digitální certifikát: obecné označení pro údaje, týkající se určitého subjektu a stvrzené jiným subjektem, který se zaručuje za jejich pravost (tzv. certifikační autoritou — u nas 1.CA, ve svete VeriSign). Nejčastěji je v certifikátu obsažen veřejný klíč vlastníka certifikátu, který má být veřejně přístupný (ale mohou zde být obsaženy i další údaje).

Údaje v certifikátu jsou chráněny pomocí asymetrických šifrovacích technik — jsou zašifrovány privátním klíčem vydavatele certifikátu (certifikační autority), a mohou být dešifrovány s použitím veřejného klíče certifikační autority (který je veřejně známý). Význam a věrohodnost certifikátu jsou závislé jak na věrohodnosti samotné certifikační autority, tak i na způsobu, jakým tato autorita získává a ověřuje údaje, které svým certifikátem stvrzuje.

Treba pro certifikat pouzitelny pro digitalni podpis pro komunikaci se statni spravou musite jit osobne s obcankou do 1.CA. Nikdo jiny zatim nema "licenci" na to, aby tyto certifikaty vydaval. Naproti tomu Verisign vydava vice "druhu" certifikatu, podle toho jak byla overena identita.

Pro uplnou funkci je treba pouzivat i casove znamky (timestamp). Jinak bych u cehokoli mohl tvrdit, ze to bylo podepsano nekym jinym mym certifikatem az potom, co jsem jej prohlasil za neplatny a vygeneroval si jiny.

Ověření certifikátů se tedy dáj zřetězovat. Řekněme, že certifikát autority A má každý, např už jako součást prohlížeče nebo OS. A podepíše certifikát autoritám AA, BB, CC a do certifikátu je zaznamenenáno, kdo certifikát podepsal (jak sehnat certifikát podepisovače). AA nebo BB nebo CC podepíše certifikáty AAA, BBB, CCC, DDD, EEE, a tak dále. Pokud jsme schopni sehnat všechny certifikáty v tomto stromě od listu ke kořeni, tak jsem schopni ověřit, zda je certifikát v listu důvěryhodný (je tak důvěryhodný jako nejméně důvěryhodná autorita na cestě do kořene)

23.27 VPN

viz http://cs.wikipedia.org/wiki/Virtu%C3%A1ln%C3%AD_priv%C3%A1tn%C3%AD_s%C3%AD%C5%A5

Z fyzickeho pohledu jde o součást jiné sítě. Z logického pohledu se jedná o samostatnou sit. Muze mit vlastni adresovaní, do jinych siti pristup jen pres branu aj.

Duvod vzniku: uzivatel chce mit vlastni sit, ale nevyplati se mu ji fyzicky budovat. Jeji "simulace" je levnejsi. Provoz ve VPN je sifrovan, aby byl bezpecny. poskytuje typicky sluzby identifikace (zjisteni identity), autentizace (overeni pristupovych prav) u kazdeho vstupu uzivatele (pripojeni z domova, z jine pobocky) pres verejnou sit (napr. internet). Dale zajistuje duvernost (sifrovani komunikace — nikdo si to neprecte) a integritu (taky sifrovani — nikdo to nezmeni, aniz by se to poznalo).

Priklady pouziti:

- uzivatel z domu do firemni site, nekolik pobocek dohromady se tvari jako jedna sit
- ve svete telekomunikaci se vyskytuji sluzby jako "volani zdarma v ramci firmy", jejiz telefony tvori VPN (coz jsou teda asi spis jen jina cisla a tarifikace nez specialni sifrovani)