EVROPSKÁ UNIE
MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
OP Vzdělávání pro konkurenceschopnost
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Vysoká škola báňská – Technická univerzita Ostrava

PŘENOS DAT

učební text

Marek Dvorský, Pavel Nevlud

Ostrava 2012

Recenze: Ing. Petr Machník, Ph.D.,
Ing. Aleš Oujedský, Ph.D.

Název: Přenos dat
Autor: Marek Dvorský, Pavel Nevlud
Vydání: první, 2010
Počet stran: 114
Náklad: 20

Studijní materiály pro studijní obor Informační a komunikační technologie fakulty elektrotechniky a informatiky
Jazyková korektura: nebyla provedena.

Určeno pro projekt:
Operační program Vzděláváním pro konkurenceschopnost
Název: Personalizace výuky prostřednictvím e-learningu
Číslo: CZ.1.07/2.2.00/07.0339
Realizace: VŠB – Technická univerzita Ostrava
Projekt je spolufinancován z prostředků ESF a státního rozpočtu ČR

ISBN 978-80-248-2604-2

POKYNY KE STUDIU

Přenos dat

Pro předmět Přenos dat 3. semestru bakalářského oboru Telekomunikační technika jste obdrželi studijní balík obsahující

- integrované skriptum pro distanční studium obsahující i pokyny ke studiu
- CD-ROM s doplňkovými animacemi vybraných částí kapitol
- harmonogram průběhu semestru a rozvrh prezenční části
- rozdělení studentů do skupin k jednotlivým tutorům a kontakty na tutory
- kontakt na studijní oddělení

Cílem předmětu

je seznámení se základními pojmy v oblasti přenosu dat. Po prostudování modulu by měl student být schopen samostatně a pomocí materiálů řešit problémy a úkoly různých způsobů přenosu dat.

Pro koho je předmět určen

Modul je zařazen do bakalářského studia oboru Telekomunikační technika studijního programu Informační a komunikační technologie, ale může jej studovat i zájemce z kteréhokoliv jiného oboru, pokud splňuje požadované prerekvizity.

Skriptum se dělí na části, kapitoly, které odpovídají logickému dělení studované látky, ale nejsou stejně obsáhlé. Předpokládaná doba ke studiu kapitoly se může výrazně lišit, proto jsou velké kapitoly děleny dále na číslované podkapitoly a těm odpovídá níže popsaná struktura.

Při studiu každé kapitoly doporučujeme následující postup:

Čas ke studiu: xx hodin

Na úvod kapitoly je uveden **čas** potřebný k prostudování látky. Čas je orientační a může vám sloužit jako hrubé vodítko pro rozvržení studia celého předmětu či kapitoly. Někomu se čas může zdát příliš dlouhý, někomu naopak. Jsou studenti, kteří se s touto problematikou ještě nikdy nesetkali a naopak takoví, kteří již v tomto oboru mají bohaté zkušenosti.

Cíl: Po prostudování tohoto odstavce budete umět

- popsat ...
- definovat ...
- vyřešit ...

Ihned potom jsou uvedeny cíle, kterých máte dosáhnout po prostudování této kapitoly – konkrétní dovednosti, znalosti.

VÝKLAD

Následuje vlastní výklad studované látky, zavedení nových pojmů, jejich vysvětlení, vše doprovázeno obrázky, tabulkami, řešenými příklady, odkazy na animace.

Shrnutí pojmů

Na závěr kapitoly jsou zopakovány hlavní pojmy, které si v ní máte osvojit. Pokud některému z nich ještě nerozumíte, vraťte se k nim ještě jednou.

Otázky

Pro ověření, že jste dobře a úplně látku kapitoly zvládli, máte k dispozici několik teoretických otázek.

Úlohy k řešení

Protože většina teoretických pojmů tohoto předmětu má bezprostřední význam a využití v databázové praxi, jsou Vám nakonec předkládány i praktické úlohy k řešení. V nich je hlavní význam předmětu a schopnost aplikovat čerstvě nabyté znalosti při řešení reálných situací hlavním cílem předmětu.

Další zdroje

Seznam další literatury, www odkazů ap. Pro zájemce o **dobrovolné** rozšíření znalostí popisované problematiky.

Buď na konci kapitoly, nebo celého textu.

Úspěšné a příjemné studium s touto učebnicí Vám přejí autoři výukového materiálu

Marek Dvorský, Pavel Nevlud

OBSAH

1. INFORMAČNÍ VLASTNOSTI ZPRÁV

Čas ke studiu: 2 hodiny

Cíl Po prostudování tohoto odstavce budete umět

- definovat základní pojmy pro přenos dat
- popsat jednotlivé bloky a rozhraní
- analyzovat informační vlastnosti zpráv

Výklad

1.1. Základní poznatky a pojmy

Historie přenosu dat pochází ze starého Řecka, kdy byl přenos uskutečňován pomocí kouřových signálů. Již v počátcích byla snaha o co nejspolehlivější přenosy informací na velké vzdálenosti.

Než se budeme zabývat problémy přenosu dat je potřeba se seznámit s několika základními pojmy a vysvětlit možnosti jejich použití.

Data – jsou zprávy ve tvaru vhodném pro jejich přenos, uložení do paměti nebo zpracování.

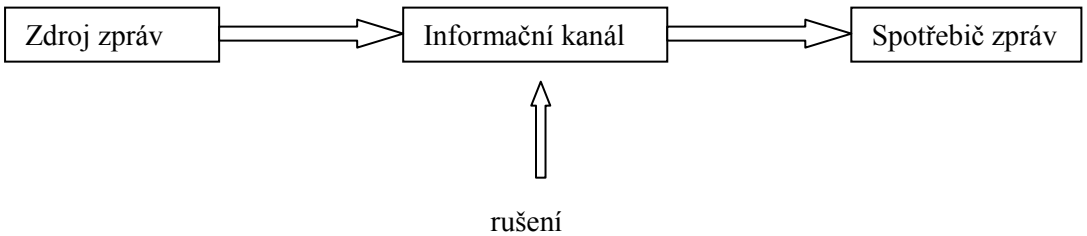
Zpráva – skládá se z jednotlivých prvků, které jsou vybírány z množiny prvků.

Diskrétní zpráva – uspořádaná posloupnost znaků, určená pro sdělování informace.

Přenos dat – spolehlivý přenos diskrétních zpráv o malé nadbytečnosti.

1.2. Informační vlastnosti zpráv

Sdělování je proces přenosu zpráv z místa vzniku – zdroj zpráv, do místa využití – spotřebič zpráv. Následující obrázek znázorňuje způsob přenosu zpráv, včetně rušení, které negativně ovlivňuje přenášené zprávy.



Obr. 1.1 Blokové schéma přenosu zpráv

Informační kanál představuje souhrn prostředků pro přenos zpráv. Tyto prostředky zahrnují jednotlivé přenosové a spojovací systémy, včetně dalších podpůrných mechanismů.

Objem (míra) informace – vyjadřuje množství informace obsažené ve zprávě.

Pokud budeme uvažovat zprávu ze zdroje zpráv s pravděpodobnosti výskytu P_i, potom množina všech zpráv tvoří úplnou soustavu náhodných jevů:

$$\sum P_i = 1 \tag{1.1}$$

Objem informace zprávy I_i je funkcí pravděpodobnosti P_i .

Objem informace dané zprávy vyjadřuje stupeň zvýšení určitosti u pozorovatele, které výskyt zprávy vyvolá.

Jistá zpráva nemá žádnou neurčitost, a tudíž nemá žádný objem informace.

Je-li jednotkový objem informace dán neurčitostí výskytu jednoho ze dvou stejně pravděpodobných jevů:

$$I_i = - \log_2 P_i \ \text{[shannon]} \tag{1.2}$$

$$I_i = - \log_{10} P_i \ \text{[hartley]} \tag{1.3}$$

Zpráva o velmi pravděpodobném jevu nese v sobě tedy velmi málo informace a naopak zpráva o málo pravděpodobném jevu obsahuje velké množství informace.

Informační entropie – je průměrný objem informace.

Chceme-li charakterizovat neurčitost množiny zpráv, pak použijeme k vyjádření průměrné hodnoty:

$$H = \sum P_i I_i = - \sum P_i \log_2 P_i \ \text{[shannon]} \tag{1.4}$$

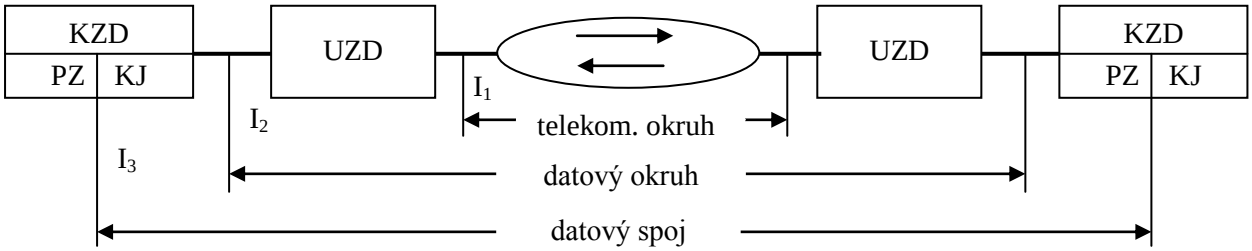
1.3. Datový řetězec pro přenos dat

Sdělování neboli komunikace, je přenos informace mezi několika místy podle dohodnutých pravidel. V souvislosti se zpracováním informace je informace vyjadřována ve tvarech vhodných ke komunikaci, uložení nebo zpracování.

Zařízení pro přenos dat mohou být velmi složitá a neustále jsou na ně kladené přísnější požadavky. Řešení těchto úkolů využívá metod, jež vyplývají z teorie systémů.

V současné době se pomocí prostředků, původně určených pro přenos dat přenášejí i jiné digitalizované signály, než pouze signály datové.

Cesta pro přenos dat mezi datovým zdrojem a datovým spotřebičem je zajišťována systémem přenosu dat, jehož blokové schéma je na následujícím obrázku.



Obr. 1.2 Obecné schéma systému přenosu dat

Systém přenosu dat je realizován různými fyzickými zařízeními a také souborem programového vybavení pro řízení tohoto systému. Význam jednotlivých částí je následující:

KZD – koncové zařízení pro přenos dat, je tvořeno vstupním a výstupním periferním zařízením PZ a komunikační jednotkou

UZD – ukončující zařízení pro přenos dat, je umístěné v objektu uživatele, navazující na přípojné vedení a zajišťující všechny funkce potřebné pro vytvoření, udržení a zrušení požadovaného spoje.

PZ – periferní zařízení, které zajišťuje vstup, nebo výstup dat. Může být realizováno různými vstupními prvky, jako je např. klávesnice, čidlo s převodníkem, paměťové médium apod. Nebo je realizováno jako výstupní zařízení, kde to může být monitor, akční člen, paměťové médium.

KJ – komunikační jednotka, je zařízení, které zajišťuje různé funkce, jako je např. synchronizace, zabezpečení přenosu dat proti chybám, přizpůsobení KZD k telekomunikačnímu okruhu.

I_1, I_2, I_3 - datová rozhraní, která definují jednotlivé mechanické, elektrické, funkční a protokolové vlastnosti.

Rozhraní I_3 – toto rozhraní je vnitřním rozhraním koncového zařízení. Zde je většinou využívaný sběrnicový způsob přenosu dat.

Rozhraní I_2 – toto rozhraní je typicky na krátké vzdálenosti, cca. jednotky metrů. Mezi využívané typy rozhraní je používané například sériové rozhraní RS232, univerzální sériová sběrnice USB (Universal Seriál Bus), apod.

Rozhraní I_1 – toto rozhraní využívá technologie pro velké vzdálenosti. Toto rozhraní je většinou mezinárodně standardizované. Může se použít například rozhraní pro analogové signály, digitální rozhraní ISDN (Integrated Service Digital Network), bezdrátové rozhraní pro UMTS (Universal Mobile Telecommunication System).

Shrnutí pojmů

Data – jsou zprávy ve tvaru vhodném pro jejich přenos, uložení do paměti nebo zpracování.

Zpráva – skládá se z jednotlivých prvků, které jsou vybírány z množiny prvků.

Diskrétní zpráva – uspořádaná posloupnost znaků, určená pro sdělování informace.

Přenos dat – spolehlivý přenos diskrétních zpráv o malé nadbytečnosti.

Objem (míra) informace – vyjadřuje množství informace obsažené ve zprávě.

Informační entropie – je průměrný objem informace.

Otázky

1. Co znamená pojem malá nadbytečnost?
2. Co nazýváme průměrným objemem informace?

Další zdroje

[1] Němec, K. *Datová komunikace*. VUT Brno 2000, ISBN 80-214-1652-1

[2] Svoboda, J. *Základy teleinformatiky.* ČVUT Praha 1998, ISBN 80-901936-3-3

[3] Shannon, E.C. *A Mathematical Theory of Communication*.
http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf

2. ABECEDY A KÓDY POUŽITÉ PŘI PŘENOSU DAT

Čas ke studiu: 3 hodiny

Cíl Po prostudování tohoto odstavce budete umět

- definovat základní typy abeced
- popsat jednotlivé druhy kódování dat
- analyzovat použitý typ abecedy

Výklad

2.1. Základní definice použitých abeced

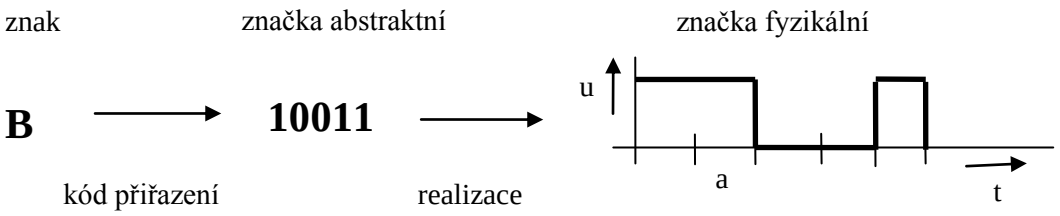
Již od počátku přenosu dat, byla snaha vytvořit vhodnou abecedu pro přenos dat. Postupně vznikaly a stále vznikají nové typy. Pro přenos dat se používají různé druhy abeced a kódů.

Prvek signálu – nejkratší možný úsek, po který může nastat změna stavu, označuje se **a**

Znak – písmeno, číslice, interpunkční znaménko, stav určitého objektu, povel signálu.

Značka – abstraktní nebo fyzikální obraz znaku, složený z prvků kódu nebo prvků signálu.

Vztah mezi jednotlivými pojmy je na následujícím obrázku.



Obr. 2.1 Vztah mezi značkou a znakem

Abeceda – sestava převodních vztahů mezi znaky a odpovídajícími značkami.

Kód – soustava dohodnutých pravidel, podle kterých se tvoří jednotlivé značky

Nerovnoměrný kód – jednotlivým znakům jsou přiřazeny značky různých délek, např. podle četnosti výskytu.

Rovnoměrný kód – jednotlivým znakům jsou přiřazeny značky stejných délek.

2.2. Typy kódů pro přenos dat

V případě rovnoměrného kódu, je kapacita abecedy dána počtem použitých prvků, které odpovídá jednotlivým značkám.

Pětiprvkový kód – kapacita kódu $2^5 = 32$ znaků. Tento kód umožňuje mít 32 různých kódových kombinací.

Mezi tyto kódy patří nejstarší Mezinárodní telegrafní abeceda, označovaná jako MTA 2. Tato abeceda se používala při dálnopisné technice. Byla přijata jako standard CCITT (dnes ITU).

Protože tato abeceda není schopna pokrýt veškerou množinu znaků, používají se řídící znaky přepínání registrů. Abeceda obsahuje tzv. registr písmen, všechna písmena (26), registr číslic (10), kde jsou jednotlivé číslice + znaménka a registr pro zabezpečené číslice, nebo transparentní provoz.

Tato abeceda má nulovou redundanci a není zabezpečena proti chybám, které vznikají při přenosu dat. Pouze, když je použit speciální registr je možno přenášet pouze některé kódové kombinace. Když je přijata nepoužívaná kombinace, je to vyhodnoceno jako chyba.

Šestiprvkový kód – kapacita kódu je $2^6 = 64$ znaků. Tento kód umožňuje celkem 64 různých kódových kombinací.

Mezi tento kód patří např. abeceda BCD, která kóduje číslice ve dvojkové soustavě a pomocí dvou změn umožňuje teoreticky 124 znaků. V praxi se tyto kódy moc nevyužívají.

Sedmiprvkový kód – kapacita kódu je $2^7 = 128$ znaků. Tento kód umožňuje celkem 128 různých kódových kombinací.

Mezi tyto kódy patří např. Mezinárodní abeceda pro přenos dat, označovaná jako MA5 a velmi používaná ASCII (American Standard Code for Information Interchange) tabulka.

Tyto abecedy obsahují 26 písmen malé abecedy, 26 písmen velké abecedy, 10 číslic, 22 znamének, 34 řídících, ovládacích a služebních znaků a 10 znaků pro národní použití.

ASCII tabulka definuje znaky anglické abecedy a jiné znaky používané v informatice. Řídící znaky jsou neviditelné znaky, které jsou určeny pro různé ovládání a mohou být někdy využívány pro různé účely.

ASCII tabulka

Výpis ascii tabulky se provádí pomocí příkazu **man ascii**

0 – 31 znak:	**ŘÍDÍCÍ ZNAKY**	
4 znak:	**EOT** [End Of Transmition]	Ctrl – D (konec přenosu)
9 znak:	**HT** [Horizontal Tabulator]	Ctrl – I (tabelátor)
10 znak:	**LF** [Line Feed]	Ctrl – J (nový řádek)
13 znak:	**CR** [Carriage Return]	Ctrl – M (návrat „vozíku“)
27 znak:	**ESC** [Escape]	Ctrl – [(únikový kód)

(Pozn.: při tisku se obvykle používají t aké řídící znaky LF a CR)
(Pozn.: čísla znaků odpovídají dekadické hodnotě (Dec sloupci))

32 znak:	**SPACE** (mezera)
33 – 47 znak:	INTERPUNKCE
48 – 57 znak:	ČÍSLICE
58 – 64 znak:	INTERPUNKCE
65 – 90 znak:	VELKÁ ABECEDA
91 – 96 znak:	INTERPUNKCE
97 – 122 znak:	MALÁ ABECEDA
123 – 126 znak:	INTERPUNKCE
127 znak:	**DEL** (výmaz)

Osmiprvkový kód – kapacita kódu je 2^8 = 256 znaků. Tento kód umožňuje celkem 256 různých kódových kombinací.

Mezi tyto kódy patří např. kód EBCDIC (Extended Binary Coded Decimal Interchange Code), nebo ISO-8, který vznikl rozšířením kódu ISO-7.

Tyto kódy většinou obsahují další rozšíření pro jiné národní abecedy a další speciální řídící znaky. Např. pro kódování češtiny je používán kód ISO-8859-2, který obsahuje české písmena s diakritikou.

Další typy kódů – dnes již existují další typy kódů, které mohou mít i delší kódové kombinace, např. 16, nebo 32 bitů. Mezi tyto kódy patří **Unicode.**

Tento kód vznikl jako naléhavá potřeba sjednotit různé kódové tabulky znaků pro národní abecedy. Například český jazyk používal v informatice nejméně 5 různě kódovaných tabulek. Vznikaly značné komplikace při spolupráci aplikací a při přenosech dat mezi programy.

V současné době existuje Unicode ve verzi 6.1, která vyšla v roce 2012. Celkem obsahuje přes 100 tisíc znaků a symbolů z 90 různých jazyků a abeced. Unicode Consorcium již v této době zaručuje, že všechny nové verze budou zpětně kompatibilní s předchozími verzemi. Tzn., že nové standardy budou přidávat další znaky, ale žádné již nebudou odstraňovat, nebo měnit.

2.3. Přenos informací

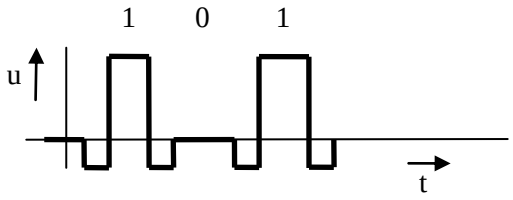
Při přenosu jsou jednotlivé značky přenášeny sériově od prvku s nejnižší váhou, případný paritní bit je na konci značky.

Sériový přenos – jednotlivé prvky jsou v časové posloupnosti vysílány po jediném přenosovém kanálu.

Paralelní přenos – vyžaduje pro každý prvek samostatný přenosový kanál. Tento typ přenosu je vhodný na velmi krátké vzdálenosti.

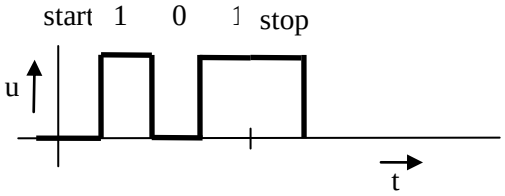
Přenos dat můžeme z hlediska synchronizace rozdělit do několika skupin.

Asynchronní přenos - do vysílaných dat se vkládají synchronizační oddělovací prvky, které nenesou žádnou informaci. Na následujícím obrázku je naznačen průběh asynchronního přenosu dat.



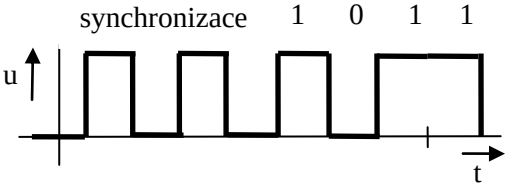
Obr. 2.2 Asynchronní přenos dat

Arytmický přenos – jednotlivé značky jsou přenášeny asynchronně, značkové prvky v rámci jedné značky se přenášejí synchronně. Využívají se tzv. start - stop prvky. Na následujícím obrázku je naznačen průběh arytmického přenosu dat.



Obr. 2.3 Arytmický přenos dat

Synchronní přenos – prvky a jednotlivé značky jsou přenášeny synchronně, pro udržení synchronizace se skládají do bloků, kterým předchází synchronizační sled prvků, které nenesou žádnou informaci. Na následujícím obrázku je naznačen průběh synchronního přenosu dat.



Obr. 2.4 Synchronní přenos dat

Pro přenos dat se používají různé provozní módy. Nejčastěji se využívají tři typy módů.

Simplexní přenos – data se přenášejí pouze jedním směrem. Pro přenos stačí pouze jeden přenosový kanál.

Poloduplexní přenos – data se přenášejí střídavě v jednom směru a opačném směru. Pro přenos také stačí pouze jeden přenosový kanál.

Duplexní přenos – přenos dat probíhá oběma směry současně. Pro přenos musí být použity dva nezávislé kanály.

U duplexního přenosu lze realizovat dva současné kanály např. metodou frekvenčního dělení, nebo časového třídění. U frekvenčního třídění jsou použity dvě různé frekvence pro každý směr přenosu. U časového dělení se využívá jiný časový interval pro každý směr přenosu.

Pro přenos dat se používají různé definice rychlosti.

Modulační rychlost - udává počet možných změn modulačních stavů za jednotkový čas. Je definována jako převrácená hodnota délky nejkratšího prvku **a.**

$$v_m = 1/a \text{ [Bd, } s^{-1}] \tag{2.1}$$

Přenosová rychlost - vyjadřuje objem přenesených dat za jednotku času.

$$v_p = v_m \log_2 m \text{ [bit/s]} \tag{2.2}$$

Přenosový výkon – vyjadřuje objem užitečných dat přenesených za jednotku času.

Shrnutí pojmů

Abeceda – sestava převodních vztahů mezi znaky a odpovídajícími značkami.

Kód - soustava dohodnutých pravidel, podle kterých se tvoří jednotlivé značky

Sériový přenos – jednotlivé prvky jsou v časové posloupnosti vysílány po jediném přenosovém kanálu.

Paralelní přenos – vyžaduje pro každý prvek samostatný přenosový kanál. Tento typ přenosu je vhodný na velmi krátké vzdálenosti.

Asynchronní přenos - do vysílaných dat se vkládají synchronizační oddělovací prvky, které nenesou žádnou informaci.

Arytmický přenos – jednotlivé značky jsou přenášeny asynchronně, značkové prvky v rámci jedné značky se přenášejí synchronně.

Synchronní přenos – prvky a jednotlivé značky jsou přenášeny synchronně, pro udržení synchronizace se skládají do bloků, kterým předchází synchronizační sled prvků, které nenesou žádnou informaci.

Simplexní přenos – data se přenášejí pouze jedním směrem. Pro přenos stačí pouze jeden přenosový kanál.

Poloduplexní přenos – data se přenášejí střídavě v jednom směru a opačném směru. Pro přenos také stačí pouze jeden přenosový kanál.

Duplexní přenos – přenos dat probíhá oběma směry současně. Pro přenos musí být použity dva nezávislé kanály.

Modulační rychlost - udává počet možných změn modulačních stavů za jednotkový čas.

Přenosová rychlost - vyjadřuje objem přenesených dat za jednotku času.

Otázky

1. Co nám udává kapacita použitého kódu?
2. Jakým způsobem lze zvýšit přenosovou rychlost?
3. Jakým způsobem lze realizovat duplexní přenos?

Další zdroje

[1] Němec, K. *Datová komunikace*. VUT Brno 2000, ISBN 80-214-1652-1

[2] Svoboda, J. *Základy teleinformatiky.* ČVUT Praha 1998, ISBN 80-901936-3-3

[3] The Unicode Consorcium, http://unicode.org/

3. ZPŮSOB PŘENOSU A ZABEZPEČENÍ ZPRÁV

Čas ke studiu: 3 hodiny

Cíl Po prostudování tohoto odstavce budete umět

- definovat základní typy rušení datového signálu
- popsat jednotlivé druhy zabezpečení zpráv
- analyzovat schopnost kódu detekovat nebo opravit chybu

Výklad

3.1. Rušení datového signálu

Při přenosu zpráv může docházet ke vzniku chyb, které mohou vznikat jako důsledek různých vlivů, které působí na datový signál.

Rušení z externích zdrojů je zejména způsobeno různými přeslechy, vazbami a přírodními vlivy. Dále je rušení také způsobováno použitými přenosovými prostředky.

Všechny tyto vlivy zhoršují základní parametry datového signálu.

Telegrafní zkreslení – označuje časovou deformaci digitálního signálu.

Individuální telegrafní zkreslení – je definováno, jako časová odchylka skutečného charakteristického okamžiku, vzhledem k ideálnímu okamžiku.

$$\sigma_i = \Delta t / a \,.\, 100\ [\%] \tag{3.1}$$

Arytmické telegrafní zkreslení – je definováno, jako maximální absolutní hodnota individuálního telegrafního zkreslení arytmického signálu.

$$\sigma_i = |\Delta t_{max}| / a \,.\, 100\ [\%] \tag{3.2}$$

Izochronní telegrafní zkreslení – definováno, jako algebraický rozdíl mezi maximálním a minimálním individuálním zkreslením.

$$\sigma_i = |\Delta t_{max} - \Delta t_{min}| / a \,.\, 100\ [\%] \tag{3.3}$$

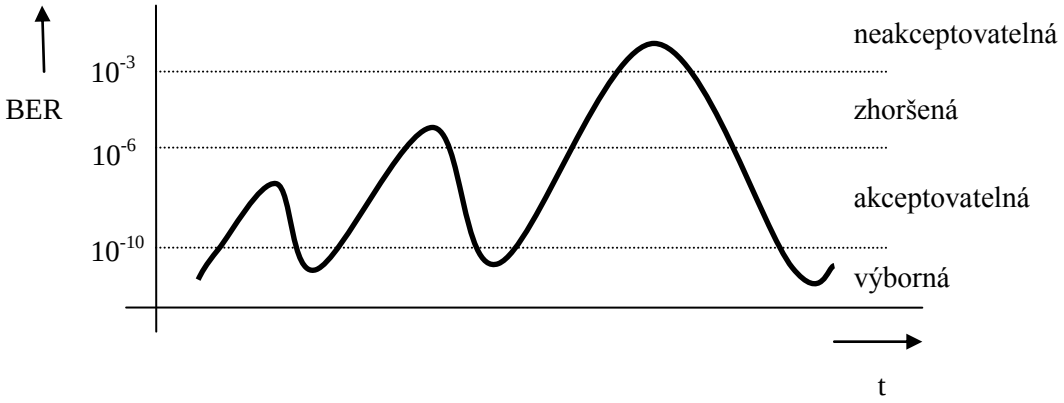
Chybovost (četnost chyb) - je dána poměrem chybně přijatých symbolů k celkovému počtu přijatých symbolů za určitou dobu pozorování.

$$BER = m_{ch} / (v_p \cdot t_m) \quad (3.4)$$

Pro hlubší analýzu chybovosti je velmi důležité časové rozložení výskytu chyb. Získání těchto průběhů je časově velmi náročné.

Chybová sekunda ERS (Errored Second) – sekunda použitelné doby, během níž nastala alespoň jedna chyba.

Neakceptovatelná sekunda SES (Severely Errored Second) – sekunda použitelné doby, během které chybovost překročila hodnotu 10^{-3}.



Obr. 3.1 Časové rozložení výskytu chyb

Poznatky o chybách lze získat několika způsoby.

První metoda využívá měření chyb v použitém kódu. Např. kód HDB3 může mít pouze definované bitové kombinace, a pokud jsou narušeny, je to vyhodnoceno jako chyba.

Další metoda využívá vyhodnocování tzv. synchroskupin, které nenesou žádné informace a pravidelně se vyskytují v přesně definovaných časových okamžicích.

Poslední metoda využívá měření věrnosti přenosu bit po bitu. Tato metoda vyžaduje znalost vysílaných dat, a tím neumožňuje přenášet žádné nové informace. Využívá se při testování nových datových okruhů, nebo v případě změn.

3.2. Způsob zabezpečení zpráv

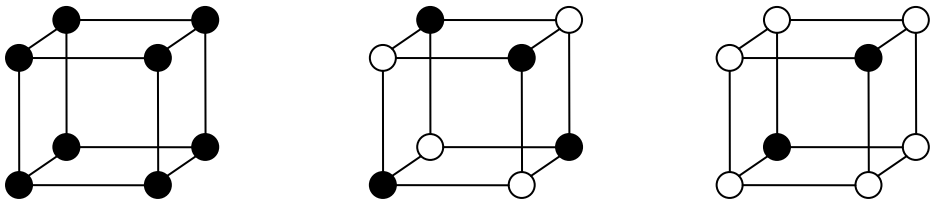
Zprávy lze zabezpečit proti chybám různými způsoby. Vždy záleží na důležitosti přenášených dat, která budeme zabezpečovat.

První metoda, kterou můžeme využít je úpravou přenášené zprávy. Zpráva obsahuje nadbytečné informace a příjemce může vyhodnotit přijatou zprávu jako narušenou. Pokud budeme posílat informaci o nějaké důležité schůzce, můžeme vložit informaci o datu, dni a hodině. Např. “Sejdeme se zítra, 1.1.2011 ve tři hodiny, 15:00.“

Další metoda využívá změnu přenosové rychlosti. Některá zařízení automaticky přizpůsobují přenosovou rychlost v závislosti na kvalitě přenosového kanálu. Při nižší rychlosti je ovlivňován nižší počet přenášených symbolů.

Nejpoužívanější metodou je využití bezpečnostního kódování, které je založeno na nadbytečnosti digitálního signálu.

Princip bezpečnostního kódování se nejsnadněji předvádí na tzv. kódové krychli. Tato krychle představuje kód s délkou 3 bitů. Černě označené vrcholy znamenají využití kódové kombinace. Prázdné vrcholy označují nevyužité kódové kombinace. Na následujícím obrázku jsou znázorněny tři typy kódů a jejich využití.



Nezabezpečený kód | Detekční kód | Korekční kód

Obr. 3.2 Princip bezpečnostního kódování

Nezabezpečený kód využívá všechny kódové kombinace. Pro tři bity existuje celkem 8 různých kombinací. Tento kód není schopen ani detekovat ani opravit jednoduchou chybu.

Detekční kód využívá pouze některé kombinace. Ty jsou zvoleny tak, aby mezi dvěma libovolnými kombinacemi byly alespoň dvě změny. Detekční kód umí pouze chybu detekovat, ale neumí ji opravit.

Korekční kód využívá pouze dvě kombinace z osmi možných. Např. lze využít pouze dvě platné kódové kombinace “000“ a “111“. Tento opravný kód je schopen opravit jednoduchou chybu. Pokud nastane více chyb, jsou opraveny na chybnou kódovou kombinaci.

Hammingova vzdálenost - označuje se jako **d** a udává počet míst, ve kterých se liší dvě kódové kombinace.

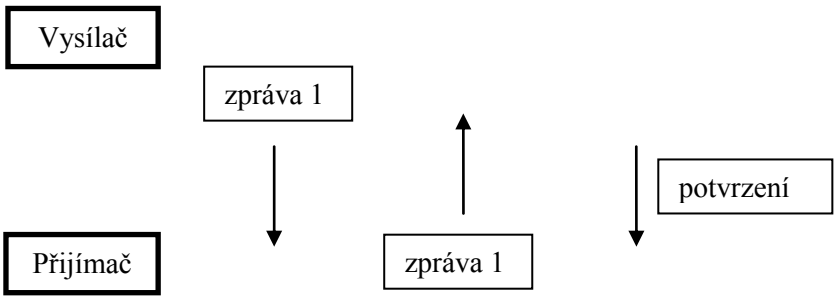
Minimální Hammingova vzdálenost - označuje se jako $\mathbf{d_{min}}$ a udává počet míst, ve kterých se liší **libovolné** dvě kódové kombinace.

Hammingova váha – označuje se jako **w** a udává počet nenulových míst v kódové kombinaci.

3.3. Zpětnovazební metody pro zabezpečení datových přenosů

Pro tyto metody musí existovat tzv. zpětný kanál. Musí být k dispozici minimálně poloduplexní provoz.

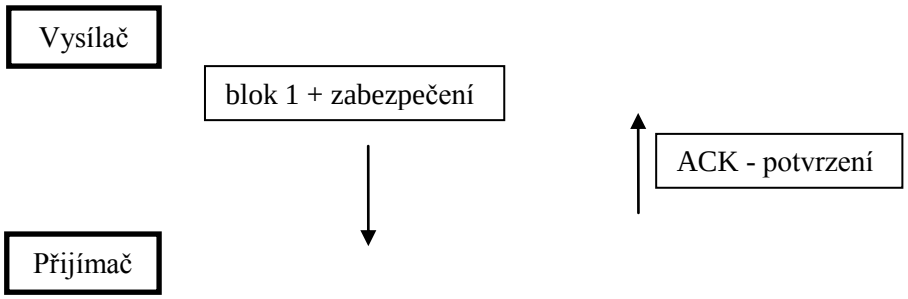
Informační zpětná vazba – vysílaný blok nemusí obsahovat žádné zabezpečení, na přijímací straně se zpráva jednak uloží do paměti a zároveň se posílá zpět k vysílači. Vysílač potom pošle potvrzení o správnosti doručení.



Obr. 3.3 Informační zpětná vazba

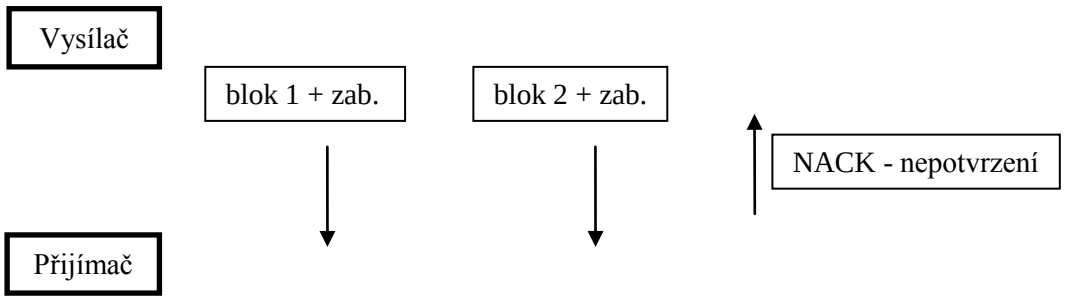
Rozhodovací zpětná vazba - vyžaduje zabezpečení vysílaných dat, označuje se jako metoda ARQ (Automatic Request for Repeat). Tato metoda může využívat různé způsoby potvrzování přenášených dat.

Pozitivní potvrzování - ARQ-ACK, metoda označovaná jako zastav a čekej na potvrzení. Po každém odeslaném bloku dat se očekává kladné, nebo záporné potvrzení přijatého bloku dat.



Obr. 3.4 Rozhodovací zpětná vazba s pozitivním potvrzováním

Negativní potvrzování – ARQ-NACK, metoda využívá spojité opakované vysílání. Vysílač vysílá bloky dat bez přerušení. Pokud dojde při přenosu k chybě, pomocí negativního potvrzení, jsou bloky znova vysílány od chybného bloku.



Obr. 3.5 Rozhodovací zpětná vazba s negativním potvrzováním

Selektivní potvrzování - tato metoda selektivně opakuje pouze chybně přijaté bloky dat. Má vyšší nároky na řízení průběhu přenosu dat, ale je lepší využití přenosových kanálů a paměti vysílače a přijímače.

Shrnutí pojmů

Telegrafní zkreslení – označuje časovou deformaci digitálního signálu.

Chybovost (četnost chyb) - je dána poměrem chybně přijatých symbolů k celkovému počtu přijatých symbolů za určitou dobu pozorování.

Chybová sekunda ERS (Errored Second) – sekunda použitelné doby, během níž nastala alespoň jedna chyba.

Neakceptovatelná sekunda SES (Severely Errored Second) – sekunda použitelné doby, během které BER překročila hodnotu 10^{-3}.

Hammingova vzdálenost - označuje se jako **d** a udává počet míst, ve kterých se liší libovolné dvě kódové kombinace.

Hammingova váha – označuje se jako **w** a udává počet nenulových míst v kódové kombinaci.

Informační zpětná vazba – vysílaný blok nemusí obsahovat žádné zabezpečení, na přijímací straně se zpráva jednak uloží do paměti a zároveň se posílá zpět k vysílači.

Rozhodovací zpětná vazba - vyžaduje zabezpečení vysílaných dat, označuje se jako metoda ARQ (Automatic Request for Repeat).

Otázky

1. Co znamená pojem Hammingova vzdálenost?
2. Která zpětnovazební metoda nevyžaduje zabezpečení vysílaných dat?

Další zdroje

[1] Němec, K. *Datová komunikace*. VUT Brno 2000, ISBN 80-214-1652-1

[2] Svoboda, J. *Základy teleinformatiky.* ČVUT Praha 1998, ISBN 80-901936-3-3

4. BEZPEČNOSTNÍ KÓDY

Čas ke studiu: 4 hodiny

Cíl Po prostudování tohoto odstavce budete umět

- definovat základní typy detekčních kódů
- popsat jednotlivé druhy bezpečnostního kódování
- analyzovat schopnost kódu detekovat nebo opravit chybu

Výklad

4.1. Detekční kódy

Detekční kódy mají schopnost rozpoznat jednu nebo několik chyb. Schopnost kódu detekovat chyby vychází z nadbytečnosti vložené informace.

Nesystematické kódy – nemají rozdělenou zvlášť informační a zabezpečovací část přenášené zprávy.

Mezi tyto kódy patří např. tzv. **izokódy.** Například kód "M z N" má konstantní Hammingovu váhu. V kódové kombinaci je vždy použit stejný počet jedniček a nul. Pokud je toto narušeno, je to vyhodnoceno jako chyba.

Systematické kódy - skládají se z informační části a zabezpečovací části. Zabezpečovací část bývá většinou za informační části, ale může také být i v informační části.

Paritní kódy – datový blok se doplní jedním paritním bitem. Používá se lichá, nebo sudá parita.

Lichá parita – obsahuje lichý počet "1", včetně paritního bitu.

Sudá parita – obsahuje sudý počet "1", včetně paritního bitu.

Iterační kódy - využívají paritních kódů. Pro první iteraci se použije značková parita pro jednotlivé značky. Pro druhou iteraci se použije bloková parita. A pro třetí iteraci je použita tzv. spirálová parita, která počítá značkovou a blokovou paritu. Tento speciální typ kódu umožňuje nejenom detekovat jednu chybu, ale dovede i opravit jednu chybu v přenášené datové posloupnosti.

Parita a spirálová parita jsou součástí této příručky, jako samostatné animace.

Cyklické kódy – tyto kódy se vyznačují jednoduchou realizací kodérů a dekodérů. Vlastní realizace těchto kódů může být řešena jednak softwarově nebo hardwarově. Při hardwarové realizaci jsou použity zpětnovazební posuvné registry. Tyto kódy vykazují velkou účinnost zabezpečení.

Cyklický kód je určen tzv. generujícím (vytvářecím) polynomem **G(x).** Tento mnohočlen musí být primitivní a stupně **r**, kde **r** udává počet zabezpečovacích prvků kódu.

Pokud **B(x),** je kódová kombinace cyklického kódu, definovaného generujícím mnohočlenem **G(x)**, potom je tato kódová kombinace dělitelná **G(x)** beze zbytku.

$$\frac{B(x)}{G(x)} = Q(x) \tag{4.1}$$

Kódové slovo cyklického kódu odpovídá bloku informačních symbolů **M(x)** a zbytku po dělení generujícím mnohočlenem **G(x).**

$$B(x) = x^r \,.\, M(x) \oplus R(x) \tag{4.2}$$

kde R(x) je zbytek po dělení.

$$\frac{x^r \,.\, M(x)}{G(x)} = Q(x) \oplus \frac{R(x)}{G(x)} \tag{4.3}$$

Zabezpečovací symboly získané z tohoto vztahu se označují jako kontrolní slovo CRC (Cyclic Redundancy Check).

Na přijímací straně je přijaté kódové slovo prověřováno, zjištěním zbytku po dělení tohoto slova

$$\frac{B'(x)}{G(x)} = Q'(x) \oplus \frac{S(x)}{G(x)} \tag{4.4}$$

kde S(x) je kontrolní syndrom.

Je-li zbytek po dělení nulový, pak při přenosu buď **nedošlo k chybě,** nebo **došlo k chybě nedetekovatelné.**

4.2. Korekční kódy

Korekční kódy se označují jako FEC (Forward Error Correction). Pro zajištění své funkce nepotřebují zpětný kanál. Chyba, nebo skupina chyb jsou opraveny na základě vazeb mezi jednotlivými prvky kódové kombinace.

Tyto kódy mají větší množství zabezpečovacích prvků, než kódy detekční. Mezi nejjednodušší korekční kódy patří **opakovací kód** s počtem opakování **n.** Tento kód má nejnižší účinnost. Tzn., že potřebuje pro opravu chyb, nejvyšší počet zabezpečovacích prvků.

Pokud potřebujeme opravit jednu chybu, musí být minimální Hammingova vzdálenost $\mathbf{d_{min} = 3.}$
V praxi se můžeme setkat např. s opakovacím kódem **n = 3**, který dokáže opravit jednu chybu. Pokud použijeme opakovací kód s **n = 5**, potom tento kód dokáže opravit 2 chyby.

Hammingův kód (n,k) je lineární binární perfektní kód, který má nejmenší redundanci. Tzn., že má nejnižší počet zabezpečovacích prvků pro opravu jedné chyby. Nevýhodou je, že nedokáže opravit více chyb.

Kódové slovo se tvoří pomocí kontrolní matice. Tato matice obsahuje nenulové a různé sloupce. Například Hammingův kód (7,4) může mít kontrolní matici ve tvaru:

$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Dekódování Hammingova kódu se provádí vynásobením kontrolní matice a přijatého kódového slova. Výsledkem je syndrom, který určuje, zda nastala chyba a určí i pozici, kde případná chyba nastala.

4.3. Výpočet CRC

Cyklický redundantní součet, označovaný také **CRC** (zkratka anglického Cyclic redundancy check) je speciální hašovací funkce, používaná k detekci chyb během přenosu či ukládání dat. Jedná se o detekční kód. Pro svou jednoduchost a dobré matematické vlastnosti jde o velmi rozšířený způsob realizace kontrolního součtu. Kontrolní součet bývá odesílán či ukládán společně s daty, při jejichž přenosu nebo uchovávání by mohlo dojít k chybě. Po převzetí dat je znovu nezávisle spočítán. Pokud je nezávisle spočítaný kontrolní součet odlišný od přeneseného nebo uloženého, je zřejmé že při přenosu nebo uchovávání došlo k chybě. Pokud je shodný, tak téměř jistě k žádné chybě nedošlo. V určitých případech je možné chybu pomocí CRC opravit.

CRC je vhodný pro zjišťování chyb vzniklých v důsledku selhání techniky, avšak jako metoda pro odhalení záměrné změny dat počítačovými piráty je příliš slabý. V tomto případě je třeba používat speciální hašovací funkce určené pro kryptovací algoritmy.

Základem je operace **MODULO 2**:

- $x^i + x^i = 0$
- $x^i + x^i + x^i = x^i$
- $x^i \cdot x^y = x^{i+y}$
- $x^i \div x^y = x^{i-y}$

Základní značení prvků:

- $B(x)$ - Kódové slovo
- x^r - Řád polynomu
- $M(x)$ - Blok informačních symbolů
- $R(x)$ - Zbytek po dělení
- $G(x)$ - Generující (vytvářecí) polynom (musí se jednat o primitivní polynom)
- $B'(x)$ - Přijaté kódové slovo
- $E(x)$ - Chybový polynom
- $S(x)$ - Kontrolní syndrom

Základní vztahy (popsané na přednáškách)**:**
Vysílání kódového slova:

$$B(x) = x^r M(x) \oplus R(x)$$

$$\frac{B(x)}{G(x)} = Q(x)$$

$$\frac{x^r M(x)}{G(x)} = Q(x) + \frac{R(x)}{G(x)}$$

Příjem kódového slova:

$$B'(x) = B(x) \oplus E(x)$$

$$\frac{B'(x)}{G(x)} = Q'(x) + \frac{S(x)}{G(x)}$$

$S(x) = 0 \rightarrow$ při přenosu nedošlo k chybě, nebo došlo k chybě nedetekovatelné

$S(x) \neq 0 \rightarrow$ při přenosu došlo k chybě

Řešený příklad

Vysílání kódového slova

Určete vysílané kódové slovo $B(x)$ *,chceme-li přenést bitovou posloupnost* $(88)_H$. *Je zadán vytvářecí polynom* $G(x) = x^4 + x + 1$ *(řád polynomu je tedy 4).*

$$(88)_H = (10001000)_B \rightarrow M(x) = 1 \cdot x^7 + 0 \cdot x^6 + 0 \cdot x^5 + 0 \cdot x^4 + 1 \cdot x^3 + 0 \cdot x^2 + 0 \cdot x^1 + 0$$

$$\frac{x^r M(x)}{G(x)} = \frac{x^4 (x^7 + x^3)}{x^4 + x + 1} = (x^{11} + x^7) / (x^4 + x + 1) = x^7 + x^4 + x + 1$$

$$\frac{\oplus\ (x^{11} + x^8 + x^7)}{x^8}$$

$$\frac{\oplus\ (x^8 + x^5 + x^4)}{x^5 + x^4}$$

$$\begin{array}{r} \oplus\,(x^5+x^2+x) \\ \hline x^4+x^2+x \\ \oplus\,(x^4+x+1) \\ \hline x^2+1 \end{array} \rightarrow R(x)=x^2+1$$

$$B(x)=x^r M(x)\oplus R(x)=(x^4(x^7+x^3))\oplus x^2+1=x^{11}+x^7+x^2+1$$

$$B(x)=x^{11}+x^7+x^2+1$$

Řešený příklad

Příjem kódového slova

Zjistěte kontrolní syndrom $S(x)$ pro přijaté kódové slovo $B'(x)=x^7+x^2+1$ a na základě syndromu určete, zda došlo při přenosu k chybě. Je zadán vytvářecí polynom $G(x)=x^4+x+1$.

(Resp. Vytvořte přijaté kódové slovo $B'(x)$ na základě znalosti vysílaného kódového slova $B(x)=x^{11}+x^7+x^2+1$ a chybového polynomu $E(x)=x^{11}$. Pro toto slovo určete kontrolní syndrom $S(x)$. Je zadán vytvářecí polynom $G(x)=x^4+x+1$)

$$B'(x)=B(x)\oplus E(x)=(x^{11}+x^7+x^2+1)\oplus x^{11}=x^7+x^2+1$$

$$\frac{B'(x)}{G(x)}=\frac{x^7+x^2+1}{x^4+x+1}=(x^7+x^2+1)/(x^4+x+1)=x^3+1$$

$$\begin{array}{r} \oplus\,(x^7+x^4+x^3) \\ \hline x^4+x^3+x^2+1 \\ \oplus\,(x^4+x+1) \\ \hline x^3+x^2+x \end{array} \rightarrow S(x)\neq 0 \rightarrow \textbf{při přenosu došlo k chybě}$$

4.4. Výpočet Hammingova kódu H(n,k)

Hammingův kód je důležitým lineárním binárním kódem se schopností opravy chyb. Jeho vlastnosti jsou dány kódovou vzdáleností $d_{min}=3$. (Vznikají tedy výběrem jedné osminy slov z přirozeného binárního kódu.) Kód s minimální vzdáleností 3 detekuje až dvojnásobnou chybu nebo opravuje jednonásobnou chybu.

Hammingovy kódy se snadno dekódují a jsou perfektní, tj. mají nejmenší myslitelnou redundanci. Jestliže počet kontrolních bitů m roste po jedné, je celková délka Hammingových kódů n= 2m-1, takže dostáváme binární (3,2)-kód, (7,4)-kód, (15,11)-kód, atd.

n – počet všech prvků
k – počet informačních prvků
m – počet zabezpečujících prvků

$$k=n-m$$

$$n=2^m-1$$

m	1	2	3	4
n	1	3	7	15
k	0	1	4	11

HAMMINGŮV KÓD (15,11)

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Provedeme záměnu příslušných sloupců abychom dostaly na posledních sloupcích jednotkovou matici. Jedná se o záměnu sloupců 1-15, 2-14, 4-13 a 8-11

$$H_K = \left[\begin{array}{ccccccccccc:cccc} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right]$$

Z každého řádku matice vytvoříme rovnici. Při tvorbě rovnic vynecháváme část jednotkovou maticí. (tzn. rovnicí vyjádříme pouze sloupce 1-11)

$$n_{12} = n_1 \oplus n_2 \oplus n_4 \oplus n_8 \oplus n_9 \oplus n_{10} \oplus n_{11}$$
$$n_{13} = n_1 \oplus n_2 \oplus n_4 \oplus n_5 \oplus n_6 \oplus n_7 \oplus n_8$$
$$n_{14} = n_1 \oplus n_2 \oplus n_3 \oplus n_6 \oplus n_7 \oplus n_{10} \oplus n_{11}$$
$$n_{15} = n_1 \oplus n_3 \oplus n_4 \oplus n_5 \oplus n_7 \oplus n_9 \oplus n_{11}$$

Dostaneme-li zadán blok informačních souborů
$M(x) = M_k \cdot x^k + \cdot M_{k-1} \cdot x^{k-1} + ... + M_1 \cdot x^1 + M_0 \cdot x^0$

S každou rovnicí postupně provedeme výpočet
$n_x = n_1 \cdot M_k \oplus n_2 \cdot M_{k-1} \oplus ..... \oplus n_{k-1} \cdot M_1 \oplus n_k \cdot M_0$

Čímž získáme prvky $n_{12}, n_{13}, n_{14}, n_{15}$, z jejichž pomocí vytvoříme kódové slovo B(x)
$B(x) = x^m M(x) \oplus n_{12} \cdot x^3 \oplus n_{13} \cdot x^2 \oplus n_{14} \cdot x^1 \oplus n_{15} \cdot x^0$

Dekódování informace:

Pro přenos platí $H_K \cdot B'(x)^T = S(x)$
$S(x) = 0 \rightarrow$ při přenosu nedošlo k chybě, nebo došlo k chybě nedetekovatelné
$S(x) \neq 0 \rightarrow$ při přenosu došlo k chybě

Řešený příklad

Vytváření kódového slova

Pro zadaný blok informačních souborů M(x)=01000001 určete kódové slovo B(x).

$M(x) = 01000001 = 0000100000\,1 = x^6 + x^0$

S každou rovnicí postupně provedeme výpočet:

$$n_{12} = n_5 \cdot M_6 \oplus n_{11} \cdot M_0 = 0 \cdot 1 \oplus 1 \cdot 1 = 1$$
$$n_{13} = n_5 \cdot M_6 \oplus n_{11} \cdot M_0 = 1 \cdot 1 \oplus 0 \cdot 1 = 1$$
$$n_{14} = n_5 \cdot M_6 \oplus n_{11} \cdot M_0 = 0 \cdot 1 \oplus 1 \cdot 1 = 1$$
$$n_{15} = n_5 \cdot M_6 \oplus n_{11} \cdot M_0 = 1 \cdot 1 \oplus 1 \cdot 1 = 0$$

Teď již můžeme vytvořit kódové slovo B(x):

$$B(x) = x^4 M(x) \oplus 1 \cdot x^3 \oplus 1 \cdot x^2 \oplus 1 \cdot x^1 = x^{10} + x^4 + x^3 + x^2 + x^1 = 0000100000\ 11110$$

Řešený příklad

Příjem kódového slova

Zjistěte, zda při přenosu kódového slova $B'(x) = 0100100000\ 11110$ *došlo k chybě, případně chybu opravte.*

Provedeme operaci $H_K \cdot B'(x)^T = S(x)$

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

$S(x) = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$, což nám odkazuje na příslušný sloupec, v kterém došlo k chybě. Jelikož se jedná o druhý sloupec, dojde k opravě příslušného bitu z kódového slova, čímž dostaneme opravené kódové slovo $B'(x) = 0000100000\ 11110$

Pozor!(v případě S(x) se nejedná o žádné binární vyjádření sloupce, ve kterém nastala chyba. Sloupec odpovídající vzoru S(x) je v tomto případě druhý, binární hodnota je 14).

Řešený příklad

Zjistěte, zda při přenosu kódového slova $B'(x) = 0101100000\ 11110$ *došlo k chybě, případně chybu opravte. (v tomto případě se jedná o dvojnásobnou chybu na pozici 2 a 4.*

Provedeme operaci $H_K \cdot B'(x)^T = S(x)$

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

$S(x) = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$, což nám odkazuje na příslušný sloupec, v kterém došlo k chybě. Jedná se o třetí sloupec, dojde k opravě příslušného bitu z kódového slova, čímž dostaneme opravené kódové slovo $B'(x) = 0111100000\ 11110$. Jelikož se jednalo o vícenásobnou chybu došlo k nekorektní opravě přijatého kódového slova.

Pozor!(opět se zde nejedná o žádné binární vyjádření sloupce, ve kterém nastala chyba. Sloupec odpovídající vzoru S(x), který je v tomto případě třetí, což náhodou odpovídá binární hodnotě)

4.5. Výpočet Reed-Müllerova kódu RM(n,k)

Reed-Muller kódy jsou z rodiny lineární chybě-opravných kódů používané v komunikaci. Jsou pojmenované podle svých objevitelů, Irving S. Reed a DE Muller. Muller objevily kódy, Reed a navrhl většinu logiky dekódování poprvé. Prvním cílem Reed-Muller kód je rovnocenný s Hadamard kód. Reed-Muller kódy jsou uvedeny jako **RM *(n,k)***, kde **n** – počet všech prvků a **k** – počet informačních prvků. RM je schopen opravit vícenásobné chyby.

$d_{\min} = 2^{m-r}$ – Hammingova vzdálenost

Počet chyb, které můžeme opravit = $2^{m-r-1}-1$

Základní výpočet hodnot:

$n = 2^m$

$m > r$

$k = 1 + \binom{m}{1} + \ldots\ldots + \binom{m}{r}$

$\binom{m}{r} = \frac{m!}{(m-r)! \cdot r!}$

m	4	4	3	3
n	16	16	8	8
r	1	2	1	2
k	5	11	4	7

REED-MŰLLERŮV KÓD (8, 4)

$$G = \begin{bmatrix} G_0 \\ G_1 \\ \vdots \\ G_r \end{bmatrix}$$

G_0 – vektor 2^m

$G_0 = [1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1]$

G_1 – matice $m \cdot 2^m$

$$G_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

G_2 – násobení jednotlivých řádků G_1

G_3 – násobení vždy 3 řádků G_1

VÝPOČET KÓDOVÉHO SLOVA POMOCÍ REED-MŰLLERŮVA KÓDU:

Kódové slovo pak určíme operací

$B(x) = M(x) \cdot G$

Budeme-li se odkazovat na příslušný bit kódového slova, použijeme toto označení

$B(x) = [b_0 \quad b_1 \quad b_2 \quad b_3 \quad b_4 \quad b_5 \quad b_6 \quad b_7]$

Budeme-li se odkazovat na příslušný informační bity, použijeme toto označení

$M(x) = [x_0 \quad x_1 \quad x_2 \quad x_3]$

DEKÓDOVÁNÍ REED-MŰLLERŮVA KÓDU:
Při dekódování určíme majoritní hodnotu z rovnic vytvořených pro každý informační bit. Rovnice jsou určeny ze sloupců (resp. vektorů) s Hammingovou vzdáleností $d_{min}=1$, které se liší právě v požadované hodnotě.

$$x_{31}=b'_0\oplus b'_1 \qquad x_{21}=b'_0\oplus b'_2 \qquad x_{11}=b'_0\oplus b'_4$$
$$x_{32}=b'_2\oplus b'_3 \qquad x_{22}=b'_1\oplus b'_3 \qquad x_{12}=b'_1\oplus b'_5$$
$$x_{33}=b'_4\oplus b'_5 \qquad x_{23}=b'_4\oplus b'_6 \qquad x_{13}=b'_2\oplus b'_6$$
$$x_{34}=b'_6\oplus b'_7 \qquad x_{24}=b'_5\oplus b'_7 \qquad x_{14}=b'_3\oplus b'_7$$

Korespondenční úkol

Zkuste si tyto rovnice odvodit sami. Podívejte se na matici G_1. Hledejte sloupce s Hammingovou vzdáleností jedna a rozdílem ve třetí hodnotě (resp. třetím řádku). Ano, jsou to sloupce 0 a 1, 2 a 3, 4 a 5, 6 a 7. Z těchto hodnot následně určíme majoritní hodnotu.

Určení majoritní hodnoty pro x_1, x_2, x_3 je předpokladem pro následné určení x_0. x_0 je dané operací **MODULO 2** hodnoty b_0 a příslušných majoritních hodnot x_1, x_2, x_3 reprezentovaných v daném sloupci.

$$x_{01}=b'_0$$
$$x_{02}=b'_1\oplus x_3$$
$$x_{03}=b'_2\oplus x_2$$
$$x_{04}=b'_3\oplus x_2\oplus x_3$$
$$x_{05}=b'_4\oplus x_1$$
$$x_{06}=b'_5\oplus x_1\oplus x_3$$
$$x_{07}=b'_6\oplus x_1\oplus x_2$$
$$x_{08}=b'_7\oplus x_1\oplus x_2\oplus x_3$$

Korespondenční úkol

Zkuste si odvodit i tyto rovnice. První rovnice je samozřejmě daná rovnou nultým sloupcem, neboť hodnoty v řádcích jsou u tohoto sloupce nulové. V případě dalšího sloupce je nutné provést operaci MODULO 2 s majoritní hodnotou třetího řádku. V případě třetího sloupce s majoritní hodnotou druhého řádku.

Řešený příklad

Výpočet Reed-Müllerůva kódu (8, 4):
Pro zadaný blok informačních souborů M(x)=1100 určete kódové slovo B(x).

Provedeme operaci $B(x)=M(x)\cdot G$

$$B(x)=M(x)\cdot G=\begin{bmatrix}1 & 1 & 0 & 0\end{bmatrix}\cdot\begin{bmatrix}1&1&1&1&1&1&1&1\\0&0&0&0&1&1&1&1\\0&0&1&1&0&0&1&1\\0&1&0&1&0&1&0&1\end{bmatrix}=\begin{bmatrix}1110000\end{bmatrix}$$

Řešený příklad

Dekódování Reed-Müllerův kódu (8, 4):

Proveďte dekódování pro přijaté kódové slovo $B'(x) = 10110000$ *a určete blok informačních symbolů M(x).*

$$x_{31} = b'_0 \oplus b'_1 = 1 \qquad x_{11} = b'_0 \oplus b'_4 = 1 \qquad x_{21} = b'_0 \oplus b'_2 = 0$$
$$x_{32} = b'_2 \oplus b'_3 = 0 \qquad x_{12} = b'_1 \oplus b'_5 = 0 \qquad x_{22} = b'_1 \oplus b'_3 = 1$$
$$x_{33} = b'_4 \oplus b'_5 = 0 \qquad x_{13} = b'_2 \oplus b'_6 = 1 \qquad x_{23} = b'_4 \oplus b'_6 = 0$$
$$x_{34} = b'_6 \oplus b'_7 = 0 \qquad x_{14} = b'_3 \oplus b'_7 = 1 \qquad x_{24} = b'_5 \oplus b'_7 = 0$$

Určením majoritní hodnoty dostaneme výsledek $x_3=0$, $x_2=0$, $x_1=1$

$$x_{01} = b'_0 = 1$$
$$x_{02} = b'_1 \oplus x_3 = 0$$
$$x_{03} = b'_2 \oplus x_2 = 1$$
$$x_{04} = b'_3 \oplus x_2 \oplus x_3 = 1$$
$$x_{05} = b'_4 \oplus x_1 = 1$$
$$x_{06} = b'_5 \oplus x_1 \oplus x_3 = 1$$
$$x_{07} = b'_6 \oplus x_1 \oplus x_2 = 1$$
$$x_{08} = b'_7 \oplus x_1 \oplus x_2 \oplus x_3 = 1$$

Určením majoritní hodnoty dostaneme výsledek $x_0=1$.
Vysílaný blok informačních souborů M(x)=1100

Řešený příklad

Proveďte dekódování pro přijaté kódové slovo $B'(x) = 10110100$ *a určete blok informačních symbolů M(x).*

$$x_{31} = b'_0 \oplus b'_1 = 1$$
$$x_{32} = b'_2 \oplus b'_3 = 0$$
$$x_{33} = b'_4 \oplus b'_5 = 1$$
$$x_{34} = b'_6 \oplus b'_7 = 0$$

Již z tohoto je patrné, že nemůžeme určit majoritní prvek. Ve výpočtu tedy již nemá smysl pokračovat.

Shrnutí pojmů

Detekční kódy mají schopnost rozpoznat jednu nebo několik chyb.

Nesystematické kódy – nemají rozdělenou zvlášť informační a zabezpečovací část přenášené zprávy.

Systematické kódy - skládají se z informační části a zabezpečovací části.

Paritní kódy – datový blok se doplní jedním paritním bitem.

Korekční kódy – mají schopnost opravit jednu, nebo více chyb.

Otázky

1. Mohou paritní kódy opravovat chyby?
2. Jaká je minimální Hammingova vzdálenost pro opravu tří chyb?

Další zdroje

[1] Němec, K. *Datová komunikace*. VUT Brno 2000, ISBN 80-214-1652-1

[2] Svoboda, J. *Základy teleinformatiky.* ČVUT Praha 1998, ISBN 80-901936-3-3

[3] Vlček, K. *Komprese a kódová zabezpečení v multimediálních komunikacích.* BEN 2006, ISBN 80-86056-68-6

5. PARAMETRY A CHARAKTERISTIKY DATOVÝCH SIGNÁLŮ

Čas ke studiu: 2 hodiny

Cíl Po prostudování tohoto odstavce budete umět

- definovat základní parametry datových signálů
- popsat jednotlivé druhy signálů pro přenos dat
- analyzovat charakteristiky datových signálů

Výklad

5.1. Základní parametry datových signálů

Signál je fyzický nosič zprávy. Rozlišujeme různé druhy datových signálů. Mezi základní typy patří signály elektrické, optické, rádiové a akustické.

Signály můžeme popsat v časové nebo kmitočtové oblasti. V kmitočtové oblasti tomu říkáme kmitočtové spektrum signálu.

Obecná představa o vztahu mezi časovou délkou signálu a šíří jeho spektra vyplývá z vlastností Fourierovy transformace.

Základem spektrální analýzy periodických signálů je vyjádření časové funkce ve tvaru Fourierovy řady.

Datová posloupnost "1 0 1 0 1 0 1 0 1 0 …." představuje z hlediska přenosu, nejméně příznivý případ při dané šířce pásma.

Při volbě praktické šířky kanálu je nutno respektovat, jednak požadavek na energii signálu, a také i požadavek na zachování tvaru signálu.

Pro reálný přenos číslicového signálu má vždy k dispozici datový signál reálný kanál s omezenou šířkou pásma.

Základním parametrem, který omezuje rychlost kanálu je šířka použitého kmitočtového pásma.

Pro diskrétní signály může každý vzorek nabývat **V** diskrétních hodnot. Pro kapacitu kanálu **C** platí Nyquistova věta:

$$C = 2.W.\log_2 V \quad [bit/s, Hz] \qquad (5.1)$$

Kde **W** označuje použitou šířku pásma.

Teoretický limit kapacity kanálu s ohledem na odstup signálu od šumu udává Shannonova věta:

$$C = W.\log_2(1 + S/N) \quad [bit/s, Hz] \qquad (5.2)$$

kde **S** označuje spektrální výkonovou hustotu datového signálu a **N** představuje spektrální výkonovou hustotu šumu datového kanálu.

5.2. Rozdělení datových signálů

Datové signály můžeme dělit do několika skupin. Základní dělení signálů je na signály náhodné a signály nenáhodné.

Náhodné signály (stochastické) můžeme charakterizovat tím, že jejich hodnotu v určitém časovém okamžiku nemůžeme přesně předpovídat. Můžeme pouze vyjádřit pravděpodobnost výskytu daného prvku.

Nenáhodné signály (deterministické) můžeme charakterizovat tím, že hodnota signálu je v libovolném časovém okamžiku přesně známa. Tento signál nenese žádnou informaci. Slouží jako stavební prvky pro vytváření náhodných signálů.

Podle časových průběhů dělíme datové signály na tři základní typy.

Spojité signály jsou popsány ve všech časových okamžicích a jsou popsány spojitou funkcí času. Tyto signály se většinou nepoužívají pro přenos dat.

Diskrétní signály jsou popsány tím, že ke změnám dochází pouze v určitých časových okamžicích. Může nabývat různých napěťových hodnot.

Digitální signály jsou popsány tím, že nabývají pouze určitých, předem definovaných hodnot. Tyto signály se mohou dále dělit na signály izochronní, nebo anizochronní. Izochronní signály mění své hodnoty pouze v určitých okamžicích. Anizochronní signály mění své hodnoty v libovolných časových okamžicích.

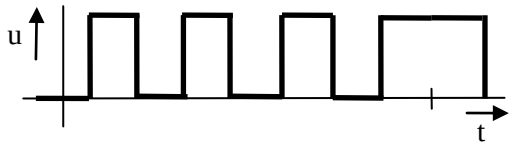
5.3. Datový signál v základním pásmu

Signály v základním pásmu mají frekvenční spektrum omezeno pouze daným přenosovým médiem. Tyto signály mohou obsahovat i stejnosměrnou složku. Jejich kmitočtové spektrum bývá velmi široké.

Stavy signálů, které přísluší jednotlivým signálovým prvkům, jsou určeny velikostí elektrické veličiny a její polaritou.

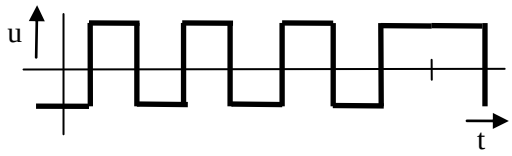
U binárních signálů mohou existovat a používají se následující tvary a průběhy datových signálů.

Unipolární signál je definován, jako signál, který má pouze dvě různé hodnoty stejné polarity.



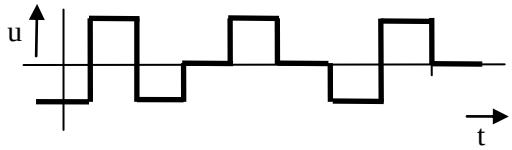
Obr. 5.1 Časový průběh unipolárního signálu

Polární signál je definován, jako signál, který má dvě různé polarity a stejné hodnoty.



Obr. 5.2 Časový průběh polárního signálu

Bipolární signál je definován tak, že může nabývat nulové hodnoty pro jeden datový symbol a dvě různé polarity pro další symbol.



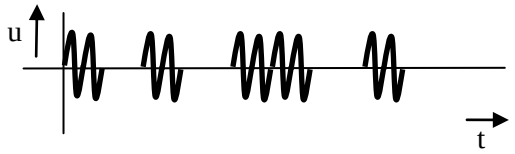
Obr. 5.3 Časový průběh bipolárního signálu

Vícestavový signál se používá v případě, že skupině binárních symbolů odpovídá jeden stav.

5.4. Datový signál v přeloženém pásmu

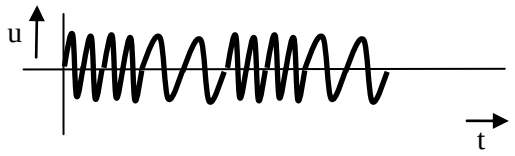
Při vytváření signálů v přeloženém pásmu se používá modulace signálu v základním pásmu na harmonický nosný signál. Používá se metoda, která se nazývá klíčování. Existují tři základní metody při vytváření datových signálů v přeloženém pásmu.

Klíčování amplitudovým posunem – ASK (Amplitude Shift Keying), tato metoda využívá toho, že datový signál, nesoucí informace ovlivňuje amplitudu nosného signálu.



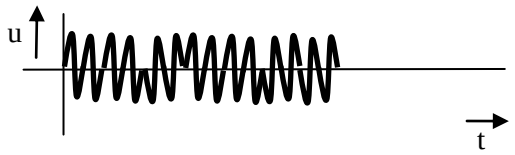
Obr. 5.4 Časový průběh ASK signálu

Klíčování frekvenčním posunem – FSK (Frequency Shift Keying), tato metoda využívá toho, že datový signál ovlivňuje frekvenci nosného signálu.



Obr. 5.5 Časový průběh FSK signálu

Klíčování fázovým posunem – PSK (Phase Shift Keying), tato metoda využívá toho, že datový signál ovlivňuje fázi nosného signálu.



Obr. 5.6 Časový průběh PSK signálu

V praxi se nejčastěji používá kombinace ASK a PSK metoda. Této kombinované metodě se říká kvadraturní amplitudová modulace neboli QAM. Při této metodě se může měnit jak amplituda, tak i fáze přenášeného signálu. Tato metoda využívá vícestavové modulace a jeden symbol nese několik bitů informace.

Shrnutí pojmů

Signál je fyzický nosič zprávy.

Náhodné signály (stochastické) můžeme charakterizovat tím, že jejich hodnotu v určitém časovém okamžiku nemůžeme přesně předpovídat.

Nenáhodné signály (deterministické) můžeme charakterizovat tím, že hodnota signálu je v libovolném časovém okamžiku přesně známa.

Spojité signály jsou popsány ve všech časových okamžicích a jsou popsány spojitou funkcí času.

Diskrétní signály jsou popsány tím, že ke změnám dochází pouze v určitých časových okamžicích.

Digitální signály jsou popsány tím, že nabývají pouze určitých, předem definovaných hodnot.

Unipolární signál je definován, jako signál, který má pouze dvě různé hodnoty stejné polarity.

Polární signál je definován, jako signál, který má dvě různé polarity a stejné hodnoty.

Bipolární signál je definován tak, že může nabývat nulové hodnoty pro jeden datový symbol a dvě různé polarity pro další symbol.

Otázky

1. Jaký typ klíčování se používá nejčastěji?
2. Proč nemůže nést deterministický signál žádnou zprávu?

Další zdroje

[1] Němec, K. *Datová komunikace*. VUT Brno 2000, ISBN 80-214-1652-1

[2] Svoboda, J. *Základy teleinformatiky.* ČVUT Praha 1998, ISBN 80-901936-3-3

[3] Hanus, S. *Bezdrátové a mobilní komunikace.* VUT Brno 2003, ISBN 80-214-1833-8

6. ZPŮSOBY MODULACE A DEMODULACE DATOVÝCH SIGNÁLŮ

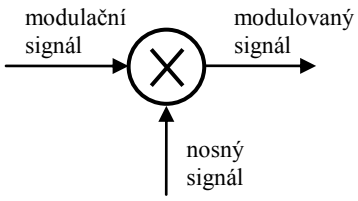
Čas ke studiu: 2 hodiny

Cíl Po prostudování tohoto odstavce budete umět

- definovat základní způsoby modulace a demodulace
- popsat jednotlivé druhy modulací
- analyzovat charakteristiky modulovaných signálů

Výklad

Modulace je proces, při kterém dochází k ovlivňování parametrů jednoho signálu (nosná), jiným signálem (modulačním). Výsledkem modulace je modulovaný signál. Zařízení, které provádí modulaci, se nazývá modulátor a musí vždy obsahovat nelineární prvek, na kterém se modulace realizuje.



Obr. 6.1 Základní princip modulace

Demodulace je proces, fyzikálně shodný s modulací, používá se ke změně formy elektrického signálu na přijímací straně. Při tomto procesu se zpětně získává z vysokofrekvenčního signálu původní modulační signál.

6.1. Amplitudová modulace

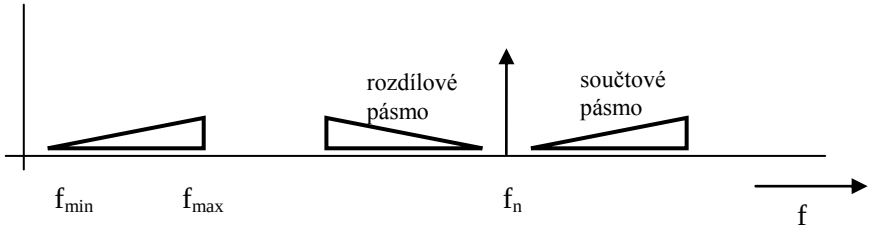
Amplitudová modulace je proces, při němž je ovlivňována amplituda nosného signálu modulačním signálem. Jedná se o historicky nejstarší typ modulace. Začala se používat při experimentech s rádiovým vysíláním. Klasická amplitudová modulace obsahuje nosnou vlnu a dvě postranní pásma (součtové a rozdílové). V praxi se většinou některé z těchto složek odstraňují a vzniká tak modulace s jedním postranním pásmem, nebo s potlačenou nosnou.

Amplitudovou modulaci můžeme popsat matematicky následujícím vztahem:

$$f_{AM}(t) = A \,.\, [1 + m \,.\, \cos(\omega_m t)] \,.\, \cos(\omega_n t) \qquad (6.1)$$

kde A je amplituda nosného signálu, m je hloubka modulace, ω_m je úhlový kmitočet modulačního signálu a ω_n je úhlový kmitočet nosného signálu.

Na následujícím obrázku je znázorněn modulační signál v kmitočtovém pásmu f_{min} až f_{max}, dílčí postranní pásma a nosná frekvence.



Obr. 6.2 Frekvenční analýza amplitudové modulace

6.2. Frekvenční modulace

Frekvenční modulace je proces, při němž je ovlivňována frekvence nosného signálu. Amplituda zůstává stejná. Maximální amplitudě napětí modulačního průběhu odpovídá maximální změna kmitočtu nosné. Tyto změnu označujeme jako frekvenční zdvih.

Frekvenční modulaci můžeme popsat matematicky následujícím vztahem:

$$f_{FM}(t) = A \cdot \cos[\omega_n t + \Delta\omega \cdot \sin(\omega_m t)] \qquad (6.2)$$

kde A je amplituda nosného signálu, ω_n je úhlový kmitočet nosného signálu, $\Delta\omega$ je modulační index a ω_m je úhlový kmitočet modulačního signálu.

Frekvenční modulace jednou frekvencí vytváří nekonečně mnoho postranních frekvencí. Amplitudy nosné vlny i jednotlivých postranních frekvencí jsou dány hodnotami Besselovských funkcí a jsou tedy závislé na modulačním indexu. Pro dostatečně kvalitní přenos pak stačí přenášet pouze několik prvních postranních kmitočtů. Potřebná šířka přenosového pásma je závislá na modulačním indexu.

6.3. Fázová modulace

Fázová modulace je proces, při němž je ovlivňována fáze nosného signálu. Amplituda zůstává stejná. Fázová modulace patří spolu s frekvenční modulací k takzvaným úhlovým modulacím. Spojitá fázová modulace se příliš nepoužívá, ale velmi často se využívají různé varianty, založené na fázové modulaci. Zejména varianta s diskrétním modulačním signálem, nazývaná PSK.

Fázovou modulaci můžeme popsat matematicky následujícím vztahem:

$$f_{PM} = A \cdot \cos[\omega_n t + \Delta\varphi \cdot \sin(\omega_m t)] \qquad (6.3)$$

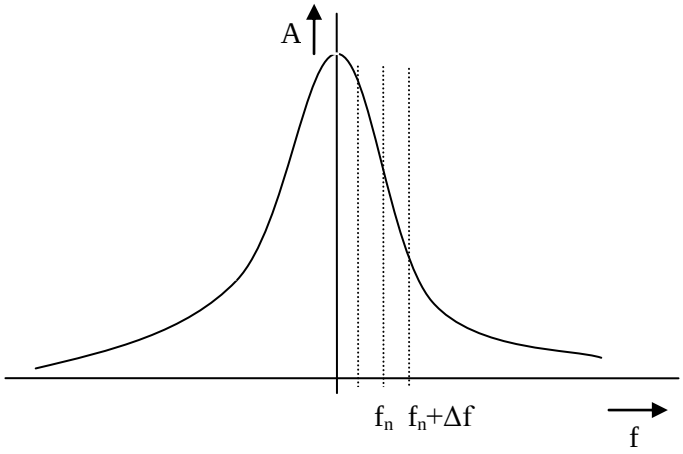
kde A je amplituda nosného signálu, ω_n je úhlový kmitočet nosného signálu, $\Delta\varphi$ je modulační index a ω_m je úhlový kmitočet modulačního signálu.

6.4. Modulace digitálním signálem – praktické využití

Jedna z prvních využívaných digitálních modulací je dvoustavová FSK modulace. Někdy se tato modulace označuje jako BFSK (Binary Frequency Shift Keying). Tato metoda používá dvě frekvence, pro jeden stav je určena jedna frekvence, pro druhý stav je určena druhá frekvence.

Modulátor bývá realizován, jako oscilátor řízený napětím, při odvození řídícího signálu z modulačního signálu. V našem případě je modulačním signálem digitální signál v základním pásmu.

Demodulátor bývá realizován jako frekvenční diskriminátor, nebo se využívá metoda demodulace na principu detekce průchodu nulou. Frekvenční diskriminátor využívá sklonu boku rezonanční křivky oscilačního obvodu k převodu frekvenční modulace na amplitudovou modulaci.



Obr. 6.3 Princip demodulace pomocí frekvenčního diskriminátoru

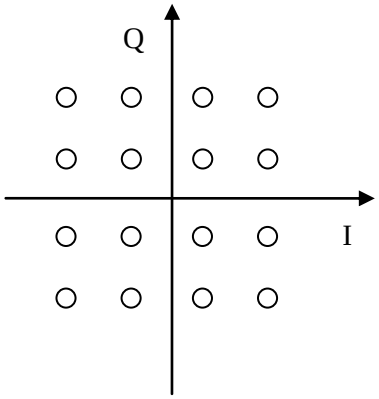
Dalším využívaným typem digitální modulace je dvoustavová PSK modulace. Tato modulace se označuje jako BPSK (Binary Phase Shift Keying). Tato metoda využívá dvě fázové změny. Existují dvě varianty, buď koherentní, nebo diferenciální PSK.

Modulátor může být realizován jako jednoduchý součinový modulátor, nebo modulátor na principu fázové inverze nosné vlny.

Demodulátor využívá násobiče a zpožďovacího článku.

Další typy digitálních modulací jsou založeny na vícestavových PSK modulací.

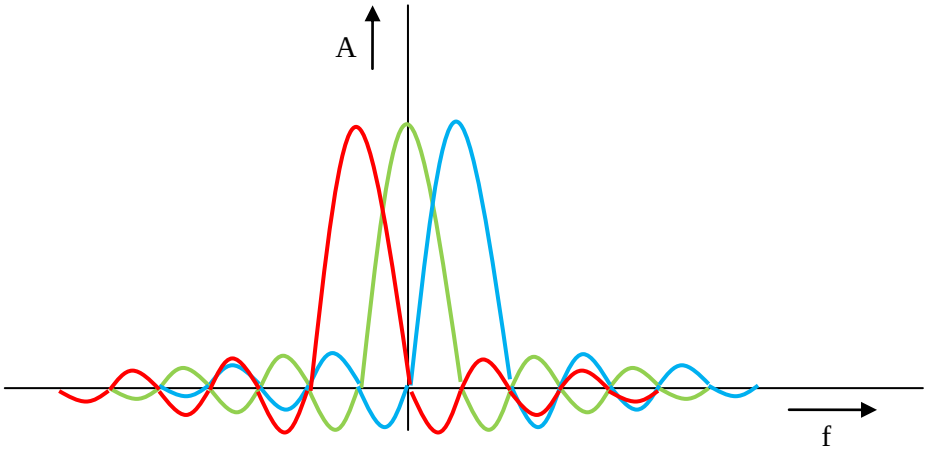
Pokud chceme dosáhnout lepšího využití přenosového pásma, využívají se kombinace ASK a PSK. Tyto metody se nazývají QAM (Quadrature Amplitude Modulation). Více můžeme vidět na následujícím obrázku, který zobrazuje rozložení jednotlivých symbolů.



Obr. 6.4 Konstelační diagram 16-stavové QAM modulace

Konstelační diagram zobrazuje koncové body fázorů, které odpovídají jednotlivým stavům. Na předchozím obrázku představuje každý symbol čtveřici bitů. Přijímač musí být schopen rozlišit jednotlivé stavy.

Nejnovější a nejperspektivnější metodou modulace je metoda OFDM (Orthogonal Frequency Division Multiplexing). OFDM je založena na více samostatných nosných, které jsou ortogonálně rozděleny po celé šířce dostupného pásma. Jednotlivé nosné jsou rozloženy na nezávislých frekvencích tak, aby se vzájemně co nejméně ovlivňovaly. Každá nosná používá samostatně některou z QAM modulačních metod.



Obr. 6.5 Frekvenční spektrum OFDM modulace

Využití OFDM je dnes prakticky ve všech přenosových technologiích. Například tuto metodu využívají technologie xDSL, PowerLine, WLAN, WiMAX, UMTS, DVB, LTE atd.

Shrnutí pojmů

Modulace je proces, při kterém dochází k ovlivňování parametrů jednoho signálu (nosná), jiným signálem (modulačním).

Demodulace je proces, fyzikálně shodný s modulací, používá se ke změně formy elektrického signálu na přijímací straně.

Amplitudová modulace je proces, při němž je ovlivňována amplituda nosného signálu modulačním signálem.

Frekvenční modulace je proces, při němž je ovlivňována frekvence nosného signálu.

Fázová modulace je proces, při němž je ovlivňována fáze nosného signálu.

Konstelační diagram zobrazuje koncové body fázorů, které odpovídají jednotlivým stavům.

Otázky

1. Jaký je rozdíl mezi modulací a demodulací?
2. Jaký typ modulace se používá v mobilních telefonech?

Další zdroje

[1] Němec, K. *Datová komunikace*. VUT Brno 2000, ISBN 80-214-1652-1

[2] Svoboda, J. *Základy teleinformatiky.* ČVUT Praha 1998, ISBN 80-901936-3-3

7. KONCOVÁ ZAŘÍZENÍ PRO PŘENOS DAT

Čas ke studiu: 2 hodiny

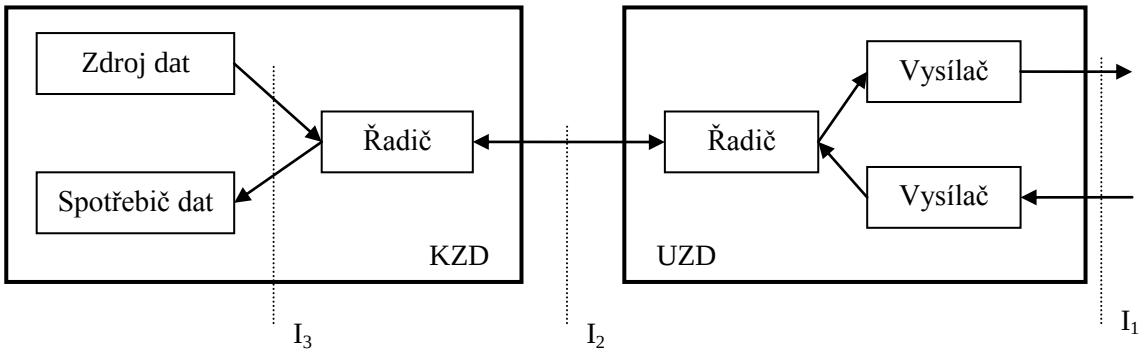
Cíl Po prostudování tohoto odstavce budete umět

- definovat základní části koncového zařízení
- popsat jednotlivé typy terminálů
- analyzovat přenosové protokoly pro různá rozhraní

Výklad

7.1. Základní prvky koncového zařízení pro přenos dat

Koncové zařízení pro přenos dat se skládá z několika dílčích částí. Na následujícím obrázku je zobrazeno zjednodušené schéma koncového zařízení pro přenos dat, včetně jednotlivých rozhraní. KZD je zkratka pro Koncové zařízení pro přenos dat, UZD je zkratka pro Ukončující zařízení pro přenos dat. I_3, I_2 a I_1 jsou definovaná rozhraní.



Obr. 7.1 Koncové zařízení pro přenos dat

Zdroj dat je zařízení, ve kterém se data vytváří. Může to být například záznamové zařízení, klávesnice, dotyková obrazovka, čidlo s A/D převodníkem atd.

Spotřebič dat je zařízení, ve kterém se data zpracovávají. Může to být například záznamové zařízení, zobrazovací jednotka, akční člen atd.

Řadič je část koncového zařízení, které umožňuje, aby zdroj a spotřebič mohly vysílat a přijímat data. Řadič může obsahovat několik modulů. Např. převodník kódu, formátování dat do bloků, zabezpečovací modul, řídící jednotku.

Terminál je speciální koncové zařízení. Je definován jako prostředek k dálkovému styku člověka, nebo řízeného objektu, s prostředím zpracování dat.

Dávkový terminál má omezené řídící funkce. Je ale žádoucí, aby pracoval bez obsluhy. Tento terminál musí být vybaven jednotkou pro automatické navazování, udržování a ukončení spojení.

Konverzační terminál využívá přítomnost obsluhy, a proto vyžaduje minimálně poloduplexní spojení.

Jako koncové zařízení může být využit i osobní počítač. Musí obsahovat komunikační program, který musí umožňovat příslušné funkce. Mezi tyto funkce patří emulace funkce koncového zařízení, možnost nastavení různých funkcí ukončujícího zařízení přenosu dat. Dále musí mít vhodné rozhraní pro uživatele. Mezi další funkce patří možnost vysílání a příjem datových souborů prostřednictvím přenosového protokolu a v neposlední řadě je důležité zajistit bezpečný přenos dat.

7.2. Přenosové protokoly

Pro přenos souborů je možno využít některý z následujících přenosových protokolů. Tyto protokoly slouží k zabezpečenému přenosu souborů mezi dvěma koncovými zařízeními. Před přenosem jsou data rozdělena do menších bloků, které mohou být různě dlouhé a mohou mít různou strukturu.

X-modem je jednoduchý, znakově orientovaný protokol. Jednotlivé znaky jsou uspořádány do bloků délky 128 znaků. Používá zabezpečení cyklickým kódem. Při příjmu bloku se vyšle vysílací straně potvrzení, nebo nepotvrzení přijatého bloku dat. Existuje i varianta pro větší bloky, která se označuje jako X-modem-1k. Tato varianta má délku bloku 1024 znaků.

Y-modem rozšiřuje X-modem o možnost jednak posílat několik souborů najednou a také posílá informace o souborech. Jedná se o časové informace a názvy jednotlivých souborů.

Z-modem je přenosový protokol s možností přenášet několik souborů najednou. Velikost souboru může být až 8kB, s možností potvrzovat několika bloků najednou. Pokud dojde při přenosu k přerušení datového spoje, dokáže tento protokol pokračovat v přenášení dat, které ještě nebyly přeneseny.

V některých implementacích Z-modemu, existují i varianty s velikostí bloku 32 kB, nebo až 64 kB. Některé varianty také umožňují i komprimaci přenášených dat.

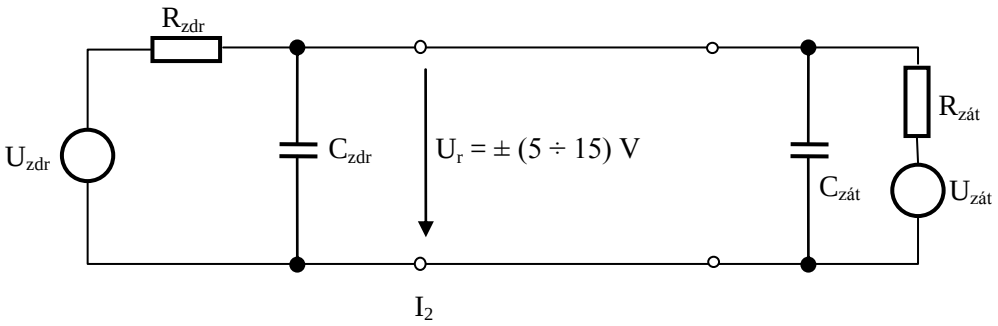
7.3. Datová rozhraní

V datovém řetězci pro přenos dat jsou definována tři rozhraní. Rozhraní I_3 je vnitřní rozhraní koncového zařízení. Toto rozhraní si definuje výrobce koncového zařízení, není standardizováno. Mezi koncovým zařízením pro přenos dat KZD a ukončujícím zařízením pro přenos dat UZD je definováno rozhraní I_2. Toto rozhraní bývá standardizováno. Posledním rozhraním je rozhraní I_3, které je mezinárodně standardizováno, většinou v rámci ITU (International Telecommunication Union).

Rozhraní I_2 je rozhraní mezi KZD a UZD. Jsou definovány čtyři základní charakteristiky, které popisují toto rozhraní.

Mechanická charakteristika určuje a popisuje fyzické vlastnosti rozhraní. Tzn., určuje typ použitého konektoru, přiřazení jednotlivých vývodů apod.

Elektrická charakteristika přiřazuje logickým binárním stavům jejich elektrické vyjádření. Definuje polaritu signálu, rozmezí amplitudy, impedanční přizpůsobení apod. Na následujícím obrázku je ekvivalentní obvod rozhraní pro doporučení V.28.



Obr. 7.2 Ekvivalentní obvod rozhraní I_2

Funkční charakteristika definuje seznam obvodů rozhraní a jejich funkce. Například doporučení V.24 definuje 42 vazebních obvodů pro všeobecné použití a 13 vazebních obvodů pro automatické volání. Každý z těchto obvodů je připojen k určitému pólu konektoru a plní určitou funkci, která může být iniciována buď ze strany KZD, nebo UZD. Mezi základní funkce patří vysílání dat, příjem dat, synchronizace, řízení toku dat, atd.

Protokolová charakteristika popisuje proces komunikace. Protokolovou charakteristiku lze rozdělit do pěti základních bodů. Nejdříve musí dojít k vytvoření spojení, poté se zřizuje datový okruh, následně je možné přenášet vlastní data. Po přenosu dat dochází k zrušení datového okruhu a následně k rozpojení spojení, pokud je to požadováno.

Shrnutí pojmů

Zdroj dat je zařízení, ve kterém se data vytváří.

Spotřebič dat je zařízení, ve kterém se data zpracovávají.

Terminál je speciální koncové zařízení. Je definován jako prostředek k dálkovému styku člověka, nebo řízeného objektu, s prostředím zpracování dat.

X-modem je jednoduchý, znakově orientovaný protokol.

Mechanická charakteristika určuje a popisuje fyzické vlastnosti rozhraní.

Elektrická charakteristika přiřazuje logickým binárním stavům jejich elektrické vyjádření.

Funkční charakteristika definuje seznam obvodů rozhraní a jejich funkce.

Protokolová charakteristika popisuje proces komunikace.

Otázky

1. Jaké má základní funkce řadič v koncovém zařízení?
2. Může být použit dávkový terminál bez obsluhy?

Další zdroje

[1] Němec, K. *Datová komunikace*. VUT Brno 2000, ISBN 80-214-1652-1

[2] Svoboda, J. *Základy teleinformatiky.* ČVUT Praha 1998, ISBN 80-901936-3-3

8. DATOVÉ MĚNIČE SIGNÁLU V ZÁKLADNÍM PÁSMU

Čas ke studiu: 2 hodiny

Cíl Po prostudování tohoto odstavce budete umět

- definovat základní rozdělení datových měničů
- popsat jednotlivé typy datových měničů
- analyzovat různé typy signálů v základním pásmu

Výklad

Hlavním úkolem ukončujícího zařízení pro přenos dat, je přeměnit původní datový signál, vyskytující se na rozhraní I_2, na signál vhodný pro přenos po daných telekomunikačních kanálech.

Ukončující zařízení pro přenos dat nazýváme datový měnič s ohledem na jeho základní funkce.

Datové měniče v základním pásmu mění kódování, impedanci a úroveň vysílaného signálu, tak aby byl přizpůsoben pro dálkový přenos.

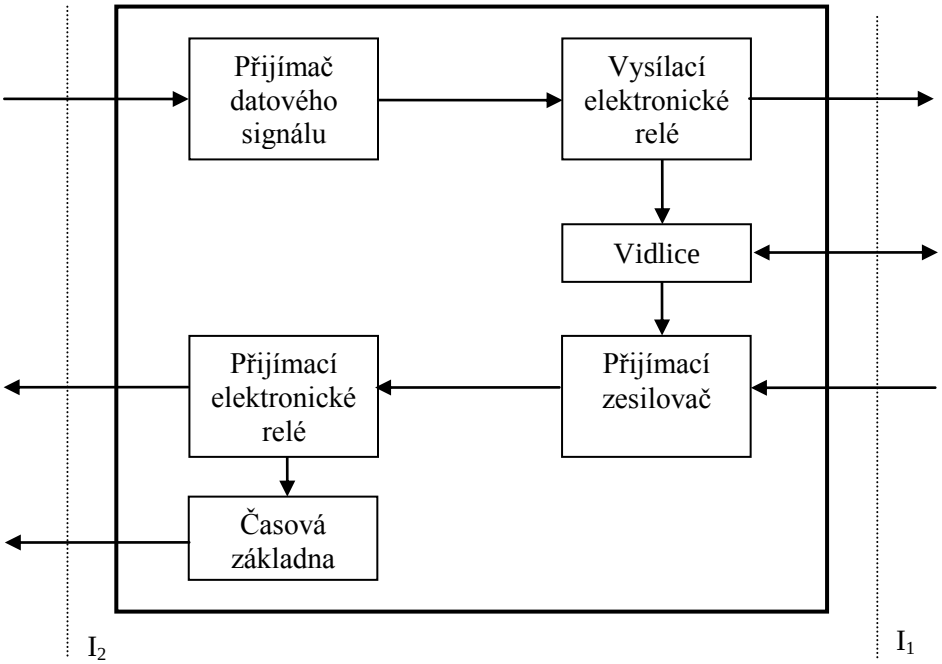
8.1. Datový měnič typu GDN

Datový měnič typu GDN je odvozen z německého názvu Gleichstrom Daten Niedrig, což v překladu znamená, stejnosměrný datový přenos s nízkou impedancí a výstupní úrovní.

Datový signál, vystupující z datového měniče obsahuje stejnosměrnou složku a využívá duplexní provoz. Díky nízké výstupní úrovni signálu, jsou velmi malá přeslechová napětí. Vlivem nízké výstupní a vstupní impedance vysílače a přijímače je velký útlum hlukových napětí.

Na následujícím obrázku je schéma zapojení datového měniče typu GDN. Tento měnič může pracovat na čtyřdrátových okruzích bez využití vidlice. Při tomto způsobu realizace je možno překlenout větší vzdálenosti.

Pokud budeme mít k dispozici pouze dvoudrátové okruhy, musíme využít v zapojení vidlici. Tím se oddělí vysílací provoz od přijímacího provozu. Nevýhodou je nižší překlenovací vzdálenost mezi jednotlivými měniči.



Obr. 8.1 Datový měnič typu GDN

8.2. Datový měnič typu DMZP

Datový měnič v základním pásmu s potlačenou stejnosměrnou složkou. Tyto datové měniče se někdy nazývají "base-band" modemy.

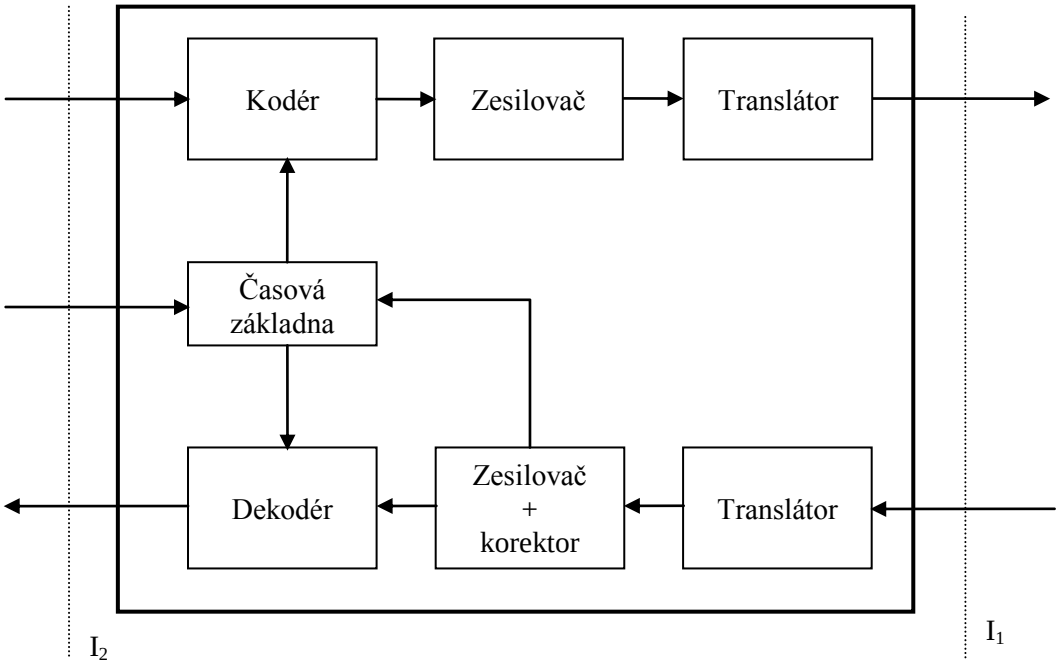
Pro potlačení stejnosměrné složky se používá bipolární AMI (Alternate Mark Inversion) signál, nebo dvoufázový signál.

Bipolární AMI kód je využíván v přenosové technice. Zde je například binární 0 realizována jako stav bez napětí, a binární 1 je realizována pravidelně se střídající kladné a záporné hodnoty napětí.

Dvoufázový signál je využíván například u Ethernetu, kdy pro každý bit je určen jeden kladný a jeden záporný puls, tak aby stejnosměrná složka byla nulová. Nevýhodou je, že jsou větší nároky na šířku pásma.

Tyto typy datových měničů se využívají pro realizaci pevných datových okruhů. Hlavní nevýhodou těchto typů je vzájemná nekompatibilnost těchto zařízení. Pro realizaci datového okruhu se musí vždy využít datového měniče stejného výrobce.

Na následujícím obrázku je principiální schéma zapojení datového měniče v základním pásmu s potlačenou stejnosměrnou složkou.



Obr. 8.2 Datový měnič v základním pásmu s potlačenou stejnosměrnou složkou

8.3. Datové měniče typu HDSL

Datové měniče HDSL (High bit rate Digital Subscriber Line) jsou mezinárodně standardizovány mezinárodní telekomunikační unií ITU.

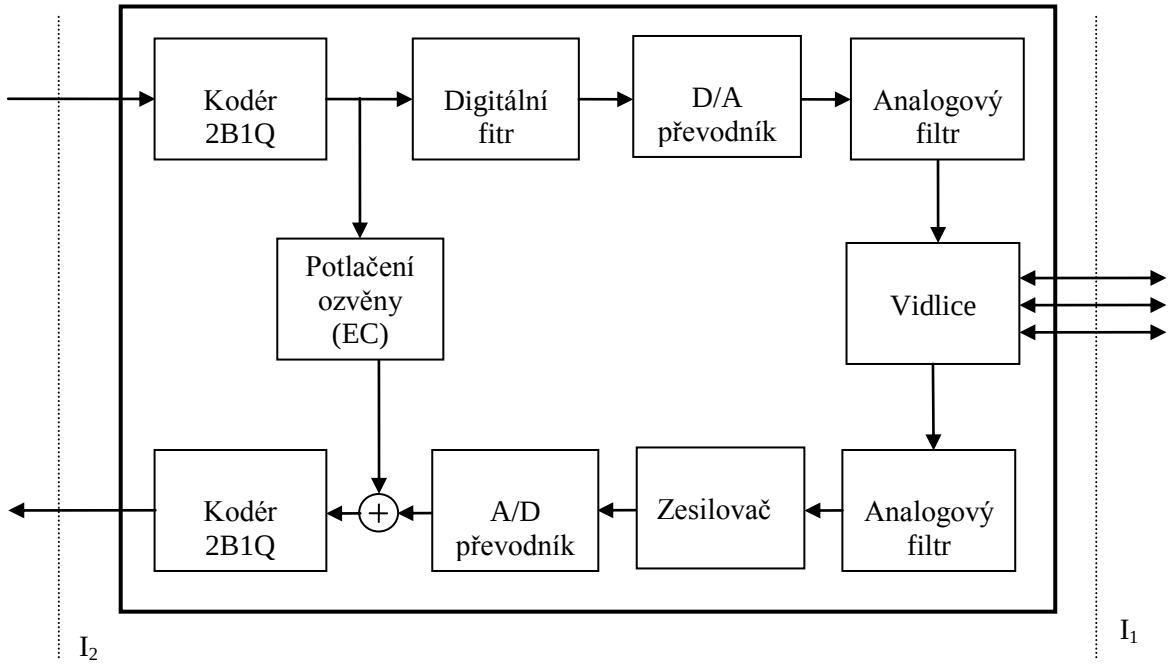
Tento typ datových měničů je zejména využíván na telekomunikačních okruzích.

Používají se na stávajících metalických kabelech. Mohou využívat jeden, dva, nebo tři kroucené páry telekomunikačního kabelu.

HDSL měniče používají každý pár pro plně duplexní provoz s použitím metody potlačení ozvěn. Metoda potlačení ozvěn se označuje EC (Echo Cancellation). Tato metoda slouží k potlačení nežádoucích rušivých signálů, které vznikají jednak nedokonalosti vidlic a nepřizpůsobením impedance k vedení.

Na každé straně HDSL datového měniče jsou potřeba jeden, dva, nebo tři vysílače/přijímače. Přenos může probíhat po několika párech kabelu současně. Vzorkovací frekvence na obou stranách musí být synchronizována.

Analogový filtr, který je ve funkci dolní propusti, slouží pro úpravu spektra přenášeného signálu. V přijímací části jsou také použity obvody pro obnovu taktu v přijímaném signálu. Tato časová základna slouží ke korekci fáze vnitřního oscilátoru datového měniče.



Obr. 8.3 Blokové schéma HDSL měniče

Na předchozím obrázku je blokové schéma HDSL měniče s využitím kódu 2B1Q. Tento linkový kód převádí na jeden kvaternární symbol. Symbol může nabývat jednoho ze čtyř stavů.

Shrnutí pojmů

Datové měniče v základním pásmu mění kódování, impedanci a úroveň vysílaného signálu, tak aby byl přizpůsoben pro dálkový přenos.

Potlačení ozvěny slouží k redukci nežádoucích rušivých vlivů.

Otázky

1. Jaké jsou základní funkce datového měniče?
2. Jakým způsobem lze potlačit nežádoucí rušení?

Další zdroje

[1] Němec, K. *Datová komunikace*. VUT Brno 2000, ISBN 80-214-1652-1

[2] Svoboda, J. *Základy teleinformatiky*. ČVUT Praha 1998, ISBN 80-901936-3-3

9. DATOVÉ MĚNIČE DATOVÉHO SIGNÁLU V PŘELOŽENÉM PÁSMU

Čas ke studiu: 5 hodin

Cíl Po prostudování tohoto odstavce budete umět

- definovat základní rozdělení datových měničů v přeloženém pásmu
- popsat jednotlivé typy datových měničů
- analyzovat různé typy signálů v přeloženém pásmu

Výklad

Datové měniče v přeloženém pásmu převádějí původní datový signál v základním pásmu na jiný signál v přeloženém frekvenčním pásmu, který je vhodný pro přenos daným telekomunikačním kanálem.

Metodou pro tento převod je nejčastěji modulace harmonického nosiče. Protože datové měniče většinou obsahují modulátor a demodulátor, nazývají se modemy.

Modemy jsou zařízení, která převádějí analogový signál na signál digitální. Modemy se používají především pro přenos digitálních dat pomocí analogové přenosové cesty.

9.1. Základní princip modemu

Modem může obsahovat různé části, které mají za úkol vhodným způsobem připravit signál pro dálkový přenos dat přes telekomunikační kanály.

Existuje celá řada typů různých modemů. Původně se používaly modemy pro komutované připojení do standardní telefonní sítě. Tyto modemy převádí digitální signál do pásma pro běžný hovor. Standardní telefonní pásmo je 300 Hz až 3400 Hz. Pro přenos se potom využívá běžná telefonní linka.

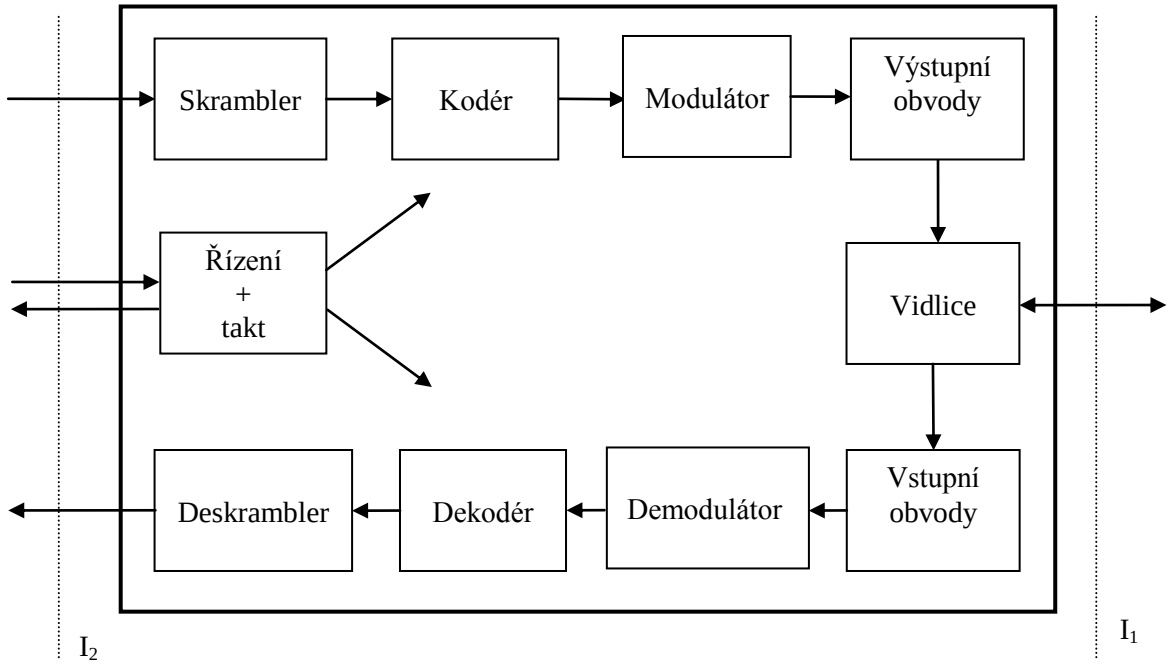
Také se využívají modemy pro pronajaté okruhy, které mají dvoudrátové, nebo čtyřdrátové propojení. Tato zařízení mají většinou lepší vybavení a mají možnost v průběhu spojení měnit přenosovou rychlost a mají také většinou lepší kompresní moduly.

Mezi další typy lze zařadit modemy pro širokopásmové kanály, například kabelové a ADSL modemy. Tyto modemy mají přenosovou cestu, která má k dispozici větší šířku pásma a je možno přenášet data s vyšší přenosovou rychlostí.

Pro bezdrátové připojení existuje celá řada radiomodemů, a modemů pro mobilní sítě. Tyto modemy jsou často přímo součástí koncového zařízení.

Modem může být realizován buď jako samostatné zařízení, nebo jako vestavěné zařízení do počítače.

Na následujícím obrázku je zjednodušené principiální schéma modemu.



Obr. 9.1 Základní princip sériového modemu

Skrambler, deskrambler – mění původní datovou sekvenci na pseudonáhodnou posloupnost. Skramblování dat se používá pro dva základní účely. Jednak lépe rozkládá frekvenční spektrum signálu a také vytváří lepší podmínky pro časovou synchronizaci dat.

Kodér, dekodér se využívá pro vícestavovou modulaci a řadí jednotlivé bity do skupin, které jsou dále modulovány.

Modulátor, demodulátor slouží pro převedení signálu do jiného frekvenčního spektra, které je vhodné pro přenos dat. Nejčastěji je využívána kvadraturní amplitudová modulace QAM.

Vstupní, výstupní obvody slouží pro úpravu signálu. Zde jsou používané různé korektory, zesilovače, omezovače apod.

Vidlice slouží k rozdělení dvoudrátového vedení na čtyřdrátové vedení a naopak.

Při navazování spojení si modemy na začátku ustavují dostupné modulační režimy, dostupné modulační rychlosti. Dále si proměřují telekomunikační okruh a nastavují jednotlivé korektory. V závěrečné fázi si určí maximální přenosovou rychlost a dohodnou se na detekci a korekci chyb, případně se dohodnou na kompresi dat.

9.2. Modemová detekce a korekce chyb

Data jsou rozdělena do jednotlivých bloků a jsou nejčastěji zabezpečena pomocí metody CRC. Pro vyhodnocování chyb, které mohou vznikat při přenosu, se používá rozhodovací zpětná vazba, označovaná jako ARQ.

Pro modemovou detekci a korekci chyb se používají různé protokoly, které definují jakým způsobem se řeší detekce a oprava chyb. Pro asynchronní znakově orientované protokoly se používá protokol MNP1 (Microcom Networking Protocol). Pro bitově orientované přenosy se používá protokol MNP3. Protokol MNP4 se používá v případě adaptivní tvorby bloků. Modemy si průběžně monitorují chybovost a podle toho mění délku zabezpečených bloků. Při nízké chybovosti vytváří delší bloky a při vyšší chybovosti jsou bloky kratší.

Další typ protokolu pro detekci a opravu chyb je protokol LAP-M (Link Access Procedure for Modems), který vychází z bitově orientovaných protokolů. Pro detekci chyby v přijatém bloku se používá zabezpečení cyklickým kódem. Při detekci chyby je chybný blok znova posílán.

Posledním typem protokolu je protokol V. 42, který je standardizován mezinárodní telekomunikační unií. Tento protokol se používá pro duplexní modemy, zahrnuje všechny předchozí protokoly.

9.3. Komprese dat používaná v modemech

Hlavním úkolem komprese dat je redukce původní datové sekvence. Slouží pro zvýšení průchodnosti přenášených datových zpráv v datových kanálech. Tím dochází ke zkrácení doby přenosu.

Účinnost komprese je dána redundancí neboli nadbytečností původních dat. Účinnost je popsána kompresním poměrem, který udává poměr přenosové rychlosti komprimovaných dat k rychlosti původních nekomprimovaných dat.

Existují tři základní typy redundance, znaková, bloková a poziční. Při znakové redundanci je využíván algoritmus, který nahrazuje nejčastěji se vyskytující se znaky, kratším bitovým vyjádřením.

Při blokové redundanci je využíván algoritmus, který nahrazuje často se opakující se skupiny znaků, zkráceným vyjádřením. Mohou to být různé formuláře, hlavičky, tabulky, apod.

Poziční redundance nahrazuje skupinu znaků, které se vyskytují na předem odhadnutelné pozici datové zprávy. Mohou to být například záhlaví souborů apod.

V praxi se využívají různé protokoly, které popisují kompresní metody. U modemů se používají buď protokoly MNP, nebo doporučení V.42bis. Oba typy jsou vzájemně nekompatibilní.

Protokoly MNP používají pro kompresy dat adaptivní Huffmanův algoritmus, který je založen na četnosti výskytu jednotlivých znaků.

Doporučení V.42bis využívá pro kompresi dat slovníkovou metodu LZW, která využívá dynamický slovník. Umí také automaticky rozpoznat náhodná data, která již nelze zkomprimovat a kompresi nepoužívá. Dosahuje vyšších kompresních poměrů, než protokoly MNP.

9.4. Přehled základních typů modemů

Historicky nejstarší modemy jsou popsány v doporučení V. 21. Tyto modemy používají dvoustavovou FSK modulaci. Dosahují maximální přenosové rychlosti 300 bit/s. Modulační rychlost u dvoustavové modulace je stejná jako přenosová rychlost. Tento modem obsahuje pouze modulátor, demodulátor, vstupní a výstupní obvody a vidlici.

Dalším typem modemu, je modem popsaný v doporučení V.22bis, který již používá 16-stavovou QAM modulaci. Tok dat je rozdělen do čtveřice bitů a ty jsou potom společně modulovány. Tento typ

modemu má modulační rychlost 600 Bd a přenosová rychlost je maximálně 2400 bit/s. Modem je možné využít, jak pro komutované linky, tak pro pevné dvoudrátové telefonní okruhy. Tento typ modemu obsahuje všechny části, které jsou na obrázku 9.1.

Mezi velmi využívané modemy, patří modem podle doporučení V.34. Tento typ modemu může používat až 1664-stavovou QAM modulaci. Modulační rychlost je závislá na kvalitě linky a může nabývat hodnot 2400 – 3429 Bd. Přenosová rychlost se může měnit v rozsahu 2400 – 33600 bit/s, podle kvality linky.

Nejnovějším typem modemu je modem podle doporučení V.90. Při tomto doporučení se musí používat dva druhy modemu. Jeden modem je analogový a druhý modem musí být digitální, nejčastěji ISDN modem s podporou V.90. Pro upload je maximální rychlost 33600 bit/s a pro download je maximální rychlost 56700 bit/s.

9.5. Výpočet komprese dat

Bezeztrátové metody komprese

- statistické
- slovníkové (kontextové) metody

Speciální znaky:
Ic indikátor komprese (znak s ascii tabulky z pozic 0 až 31)
k kompresní poměr (poměr počtu bitů před kompresí a počtu bitů po kompresi)

Algoritmus potlačení nul (mezer)

- musí obsahovat alespoň **tři stejné znaky** (nuly)
- umí komprimovat pouze jeden typ znaků
- maximálně můžeme provést kompresi 256 nul
- nahrazuje za sebou vyskytující se znaky za indikátor komprese a počet znaků

Př.:

$$AB00CD00000\,AC000 \rightarrow AB00CDI_C5ACI_C3$$

-kompresní poměr $k = \frac{16 \cdot 8}{12 \cdot 8}$

Algoritmu proudového kódování (RLE – Run Lenght Encoding)

- komprimuje jakýkoliv znak nacházející **se 4x za sebou**
- použití u komprese BMP souborů
- nahrazuje za sebou vyskytující se znaky za indikátor komprese, opakující se znak a počet zbývajících znaků (celkový počet -1)

Př.:

$$ABBBBCCCCCAAA \rightarrow AI_CB3I_CC4AAA$$

-kompresní poměr $k = \frac{13 \cdot 8}{10 \cdot 8}$

Algoritmus upraveného proudového kódování

- komprimuje jakýkoliv znak nacházející se 3x za sebou
- nepotřebuje indikátor komprese
- nevýhodou je možnost kompresního poměru vyššího než jedna
- za trojicí stejných znaků je vždy číselná hodnota vyjadřující počet identických znaků nacházejících se za nimi

Př.:

$ABBBBCCCCCAAA \rightarrow ABBB1CCC2AAA0$

-kompresní poměr $k = \frac{13 \cdot 8}{13 \cdot 8}$

Algoritmus bitové mapy

- provádí kompresi jednoho znaku
- znaky nemusí být za sebou
- před jednotlivými symboly se přenáší bitová mapa
- znak je po kompresi nahrazen v bitové mapě jako „0“

Př.:

$A0AB00CD \rightarrow$ [illegible]$0110011\ \underline{A}ABCD$

-kompresní poměr $k = \frac{8 \cdot 8}{8 + 5 \cdot 8}$

Algoritmus půlbajtové komprimace

- komprimuje číslice – v první půlce bajtu je u číslic vždy stejná kombinace čtyř bitů, které jsou při komprimaci odejmuty
- použití indikátoru komprese
- první číslice za indikátorem komprese udává počet komprimovaných číslic, v případě sudého počtu je v poslední půlce bitu 0, na kterou již první číslice neukazuje

Př.:

$ABC123456\,D \rightarrow ABCI_C$ [illegible]$,1\underline{\ }$[illegible]$,3\underline{\ }$[illegible]$,5\underline{\ }$[illegible]$,0\,\underline{D}$

-kompresní poměr $k = \frac{10 \cdot 8}{5 \cdot 8 + 4 \cdot 8}$

Shannon-Fanovo kódování

- využívá pravděpodobnost výskytu jednotlivých znaků
- nevýhodou komprimace dat jako jeden blok
- využívá **binární strom**, který **se tvoří od kořene k listům**

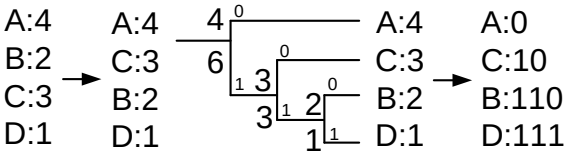
Postup:

1. vytvoříme množinu četností

2. rozdělujeme množinu četností na přibližně stejné části tak dlouho, dokud nevytvoříme celý strom (např. v jedné větvi znak s četností 10, v druhé části 5 znaků s četností: 1 a 4 znaky s četností 2)
3. z důvodu mnohoznačnosti je nutné si znát strom při dekomprimaci
4. spočítáme kódy pro všechny bajty

Př.:

ACCDACBABA



ACCDACBABA

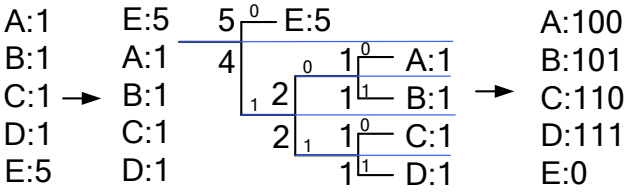
↓

0101011101011001100

- kompresní poměr $k = \frac{10 \cdot 8}{19 + 4 \cdot 8 + 1 + 2 + 3 + 3}$
- kromě komprimované binární posloupnosti (20 bitů) se musí přenést i daný strom, tedy čtyři znaky (4*8bitů) a čtyři pozice (1+2+3+3)

Př.:

ABCDEEEEE



ABCDEEEEE

↓

100101110111100000

Huffmanovo kódování

binární strom se vytváří **od jednotlivých listů ke kořenu**

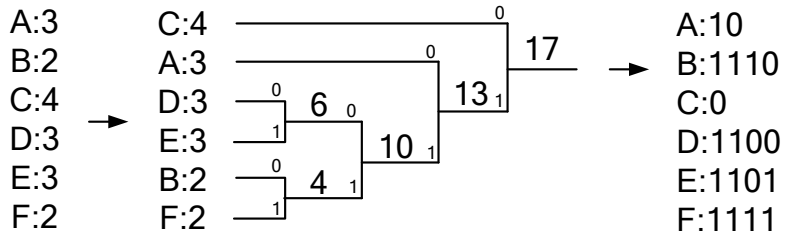
Postup:

1. vytvoříme množinu četností
2. vhodné je setřídit četnosti znaků sestupně podle počtu
3. sečteme dvě nejmenší četnosti **(!!!!ať jsou na kterékoliv úrovni stromu!!!!)** a jejich součet zařadíme do množiny četností **(!!!součet četností dále považujeme za jedinou četnost!!!)**
4. tento krok opakujeme dokud nevytvoříme celý strom
5. určíme kódy pro všechny bajty

6. máme-li možnost vybrat mezi různými znaky se stejnou četností, vybere znak nacházející se níže
7. Z důvodu mnohoznačnosti je nutné si znát strom při dekomprimaci

Př.:

ACDEBEFCADEFACBCD



ACDEBEFCADEFACBCD

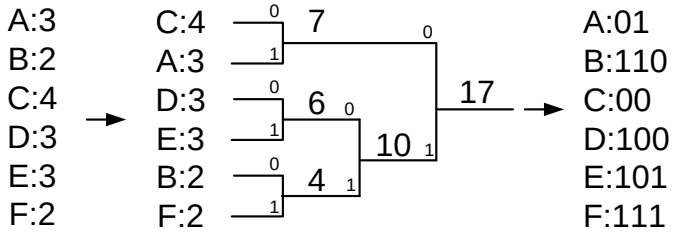
↓

10011001101111011011111010110011011111100111001100

- u některých variant HK se můžeme setkat s upřednostněním znaku (podvětve), který byl v kratší větvi (nebo nejlépe vůbec)

Př.:

ACDEBEFCADEFACBCD



ACDEBEFCADEFACBCD

↓

01001001011101011110001100101111010011000100

Adaptivní Huffmanovo kódování

- binární strom, který se tvoří průběžně
- **NZ** pozice pro *nový znak*

Postup:

1. základním pravidlem je, že **nahoře a napravo musí být větší hodnota**
2. vyskytne-li se hodnota nalevo nebo níže, nastávají v zásadě dvě možnosti:

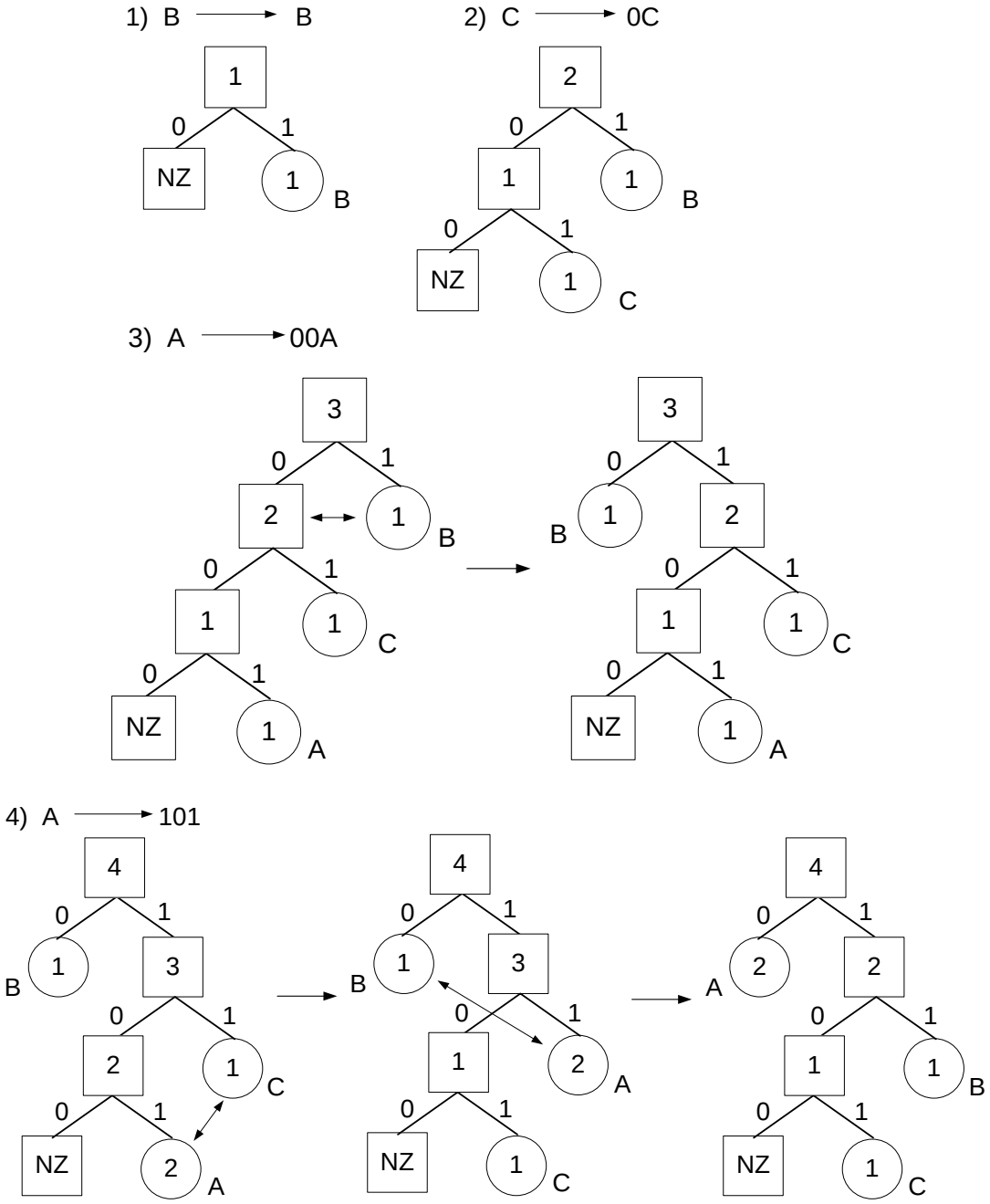
   I) jedná se o jeden znak vyskytující se vícekrát v posloupnosti (v kolečku). **Znak tedy postupně přesuneme nejdříve napravo, poté výše** (atd.).

   II) jedná se o více znaků, které svým celkovým počtem (k čtverci) převyšují počet jednoho znaku (např. v kolečku). Zde dochází k dělení stromu (**záměnu znaku v kolečku za kompletní větev vycházející ze čtverce**)

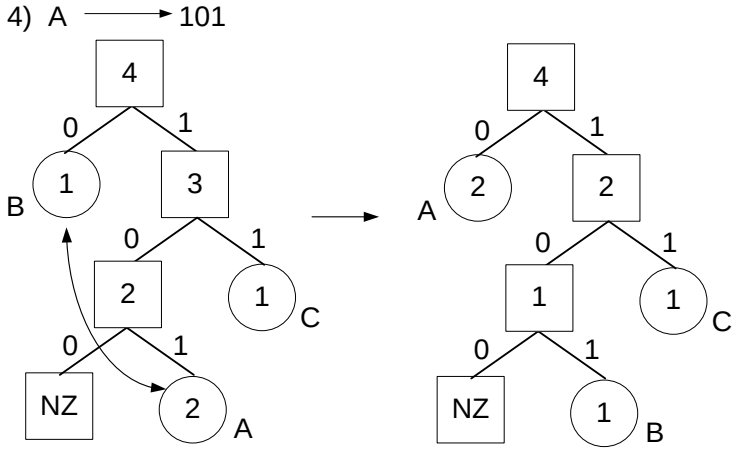
3. **operace 1 má vždy přednost před operací 2**
4. při kódování se znak přenáší pouze v případě prvního výskytu. Při dalším výskytu je již přenesen jen odkaz na jeho pozici

Př.:

BCAA

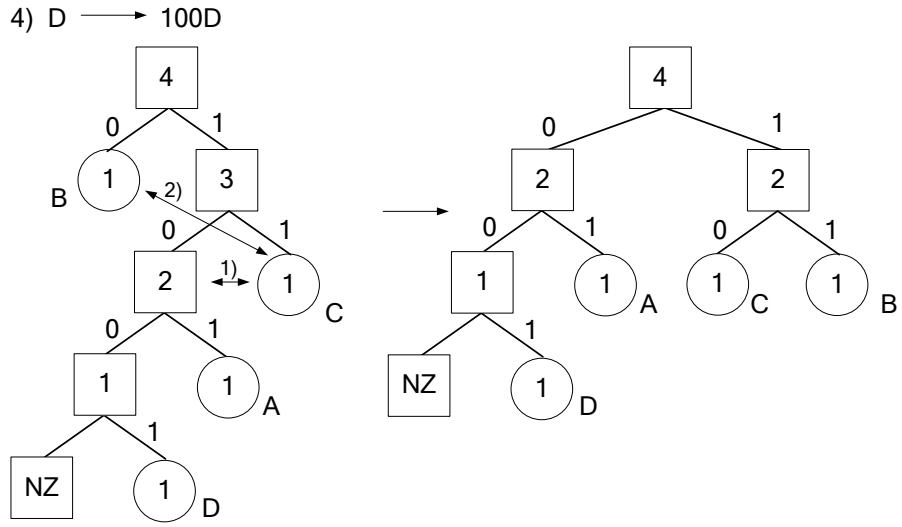


Při použití **progresivních metod** dochází k záměně s nižší hodnotou umístěnou nejblíže kořenu.

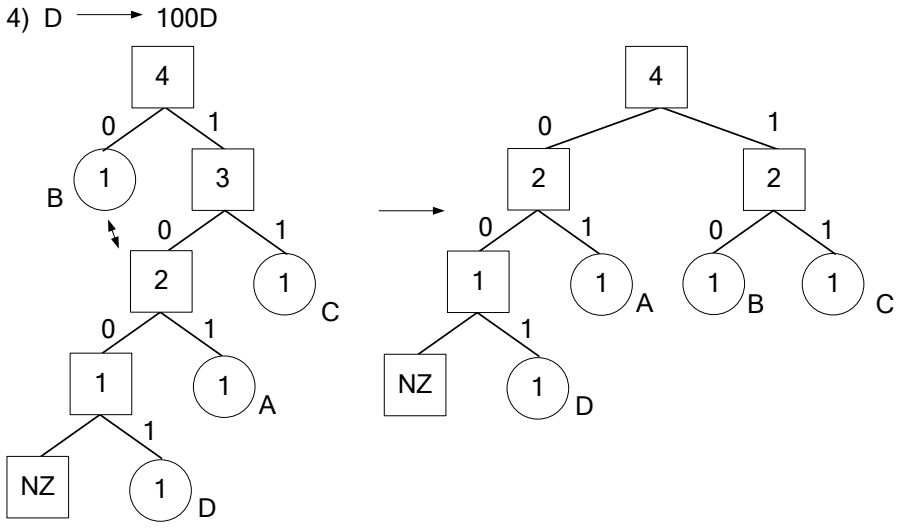


-kompresní poměr $k = \frac{4 \cdot 8}{8+9+10+3} = \frac{32}{30}$

Je-li čtvrtým znakem D, dochází zde k dělení celých větví. Nejdříve se provede záměna větví na stejné úrovni, poté záměna mezi úrovněmi



Při použití **progresivních metod** dochází k záměně s nižší hodnotou umístěnou nejblíže kořenu.



LZ 77 (Abraham Lempel, Jakob Ziv)

- substituční (slovníková) metoda
- principem je hledání, zda se znak a znaky po něm následující, které chceme komprimovat, se vyskytují již dříve v řetězci
- způsob zápisu výstupu: **(xx,yy)NZ**
  kde: **xx** je o kolik kroků zpět se znaky vyskytovaly
  yy je kolik znaků je shodných
  NZ je nový znak
- nevýhodou je velký počet prohledávání
- nový znak je vždy jen jeden (a je vždy rovnou přenesen)
- na konci posloupnosti může být nový znak vynechán, protože poslední kombinace je shodná s kombinací dříve přenesenou

Př.: *AABCBBABC*

krok	pozice	shoda	nový znak	výstup
1	1	-	A	(0,0)A
2	2	A	B	(1,1)B
3	4	-	C	(0,0)C
4	5	B	B	(2,1)B
5	7	ABC	-	(5,3)

-kompresní poměr $k = \frac{9 \cdot 8}{5 \cdot 8 + 4 \cdot 8} = \frac{9}{9} = 1$, při vyjádření čísla pomocí čtyř bitů

- nebo $k = \frac{9 \cdot 8}{5 \cdot 8 + 8 \cdot 8} = \frac{9}{13}$, při vyjádření čísla pomocí osmi bitů

Dekomprese:
-metodou odkazu na pozici již dekomprimovaného znaku a počtu znaků za ním následujících vytvářím výstupní dekomprimované slovo

vstup	výstup
(0,0)A	A
(1,1)B	AAB
(0,0)C	AABC
(2,1)B	AABCBB
(5,3)	AABCBBABC

LZ 78

- vytváří se slovník (každá strana má svůj slovník, které musí být stejné)
- způsob zápisu výstupu: **(index ve slovníku dříve vyskytující se znak(y), rozšiřující znak)**
- výhodou je menší počet prohledávání
- nový znak je vždy jen jeden (a je vždy rovnou přenesen)
- z kombinace, která je přenesena se vytvoří položka ve slovníku
- -na konci posloupnosti může být nový znak vynechán, protože poslední kombinace je shodná s kombinací obsaženou ve slovníku

Př.: *AABCBBABC*

krok	index	slovník	výstup
1	1	A	(0,A)
2	2	AB	(1,B)
3	3	C	(0,C)
4	4	B	(0,B)
5	5	BA	(4,A)
6	6	BC	(4,C)

-kompresní poměr $k = \frac{9 \cdot 8}{6 \cdot 4 + 6 \cdot 8} = \frac{9}{9} = 1$, při vyjádření čísla pomocí čtyř bitů

-nebo $k = \frac{9 \cdot 8}{6 \cdot 8 + 6 \cdot 8} = \frac{9}{12}$, při vyjádření čísla pomocí osmi bitů

Dekomprese:
-pro vytvoření výstupu za sebe řadím znaky z průběžně vznikajícího slovníku

vstup	index	slovník	výstup
(0,A)	1	A	A
(1,B)	2	AB	AAB
(0,C)	3	C	AABC
(0,B)	4	B	AABCB
(4,A)	5	BA	AABCBBA
(4,C)	6	BC	AABCBBABC

LZW (Terry Welch)

- modifikace LZ 78
- ve slovníku jsou již uloženy znaky (ASCII tabulka)
- další znaky (kombinace znaků) se umísťují za ASCII tabulku

- při tvoření nové slovníkové položky se použije slovníkový znak předcházející (pomocí něhož se již vytvořila předchozí položka slovníku) a znak nový
- jsou-li po sobě jdoucí znaky již ve slovníku, přenesou se jako jeden znak
- nikdy nevznikne situace, že bychom novou položku slovníku získali položku slovníku předcházející a další znak
- při vytváření spojení je provedena dohoda mezi přijímačem a vysílačem o počtu bitů na jednotlivý znak

Př.: $AABCBBABC$

vstup	slovník	výstup
A		
A	256=AA	A
B	257=AB	A
C	258=BC	B
B	259=CB	C
B	260=BB	B
A	261=BA	B
B		
C	262=ABC	257
		C

$AABCBBABC \rightarrow AABCBB257C$

-kompresní poměr $k = \frac{9 \cdot 12}{8 \cdot 12} = \frac{9}{8}$

Dekomprese: $AABCBB257C$
-odkazy na slovník se nahrazují vytvářejícím se slovníkem. Tyto znaky použiji dále na tvorbu slovníku
-upravený vstup je tedy výstupem

vstup	slovník	výstup
A		
A	256=AA	A
B	257=AB	A
C	258=BC	B
B	259=CB	C
B	260=BB	B
257 (A)	261=BA	B
(B)		A
C	262=ABC	B
		C

$AABCBB257C \rightarrow AABCBBABC$

9.6. Datové přenosy využívající analogové modemy

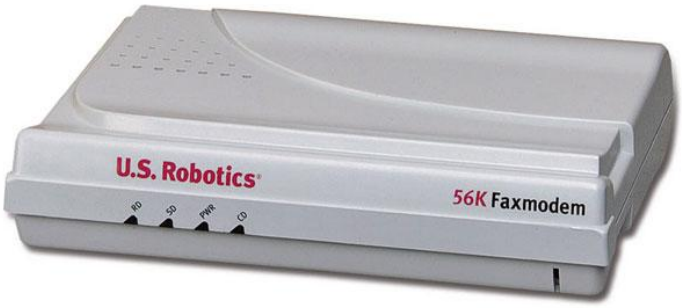
AT příkazy

AT příkazy dnes používá téměř každý modem. Byly vymyšleny za účelem jednodušší konfigurace modemu. Byl definován jednoduchý jazyk nazývaný AT-language, kde každý řádek začíná znaky AT (z anglického attention). K používání je třeba terminálový software (např. hyperterminál, minicom, gtkterm). Jeho příkazy jsou softwarem předávány po sériové lince zasílány přímo k modemu podle normy RS-232.

Modem dokáže pracovat ve dvou režimech:

- **v příkazovém**, sloužícím k zadávání AT-příkazů modemu
- **v přenosovém**, v němž mezi sebou modemy přenášejí data a nemohou přijímat AT-příkazy.

Modem nemůže pracovat v obou zároveň, ale může mezi nimi přepínat bez ztráty spojení. Přepínání se provádí tzv. escape sekvencí, která sestává ze tří stejných znaků stisknutých rychle po sobě. Tento znak je většinou "+", ale není to pravidlem.



Obr. 9.2 Analogový modem Robotics 5630

POPIS MODEMU

Popis modemu (Led diody):

1. RD (Receive Data) – přijímaná data (data přijata od druhého modemu jsou posílána do PC)
2. SD (Send Data) – odesílaná data (data přijata od PC jsou posílána do druhého modemu)
3. PWR (Power) – napájení
4. CD (Carrier Detect) – detekce nosné

Řízení toku dat mezi počítačem a modem (Flow control):
Kabel by měl obsahovat minimálně 3 vodiče (jeden pro každý směr a jeden společný)

1. Hardwarové řízení – je použito pro speciální komunikaci mezi modemem a PC. K tomu jsou potřeba další dva vodiče.
   RST – posílá počítač
   CTS – posílá modem
2. Softwarové řízení – pomocí signálu XON a XOFF
   XON – modem je připraven posílat data
   XOFF – je použito, když je vyrovnávací paměť modemu přeplněna

NASTAVENÍ A OVLÁDÁNÍ MINICOMU

Při prvním spuštění minicomu je vhodné použít příkaz `minicom -s`, pomocí něhož můžeme provést nastavení připojení modemu před jeho inicializací. Po uložení nastavení za defaultní profil je už možné dále použít pouze příkaz `minicom`

Dále je nutné nastavit položky:

- **FILENAME AND PATHS** – zde je třeba nastavit download a upload directory (do adresáře na nějž máte vhodná oprávnění)
- **SERIAL PORT SETUP** – Serial device /dev/ttyS1 (může se lišit)
  - komunikační parametry rozhraní PC/modem 9600 8N1
- **MODEM AND DIALING** – vymažte položky A a B (inicializační a resetační řetězec)

Profil uložte jako defaultní a opusťte menu volbou EXIT.

CTRL+A <klávesa> – Předvolba pro menu (Pozn. Za <klávesa> zvolte následující klávesu)
Z – nápověda minicomu
B – rolování obrazovky
S – přenos dat
Přenosové protokoly:
Z modem – volitelná velikost bloku dat (64B – 8kB), umožňuje přenést více souborů
Y modem
X modem
ASCII
S – příjem dat
O- konfigurace
Q – ukončení minicomu

ZÁKLADNÍ AT PŘÍKAZY

`AT` (Attention) – prověření funkce modemu - odpověď modemu OK

Nápověda:
`AT$` - množina příkazů, které máme k dispozici (ATD, ATIn - (za n doplníme číslo!!!))
`AT&$` - rozšířená množina příkazů (např. AT&Nn, AT&Mn, AT&Kn)
`ATD$` - příkazy pro vytáčení
`ATS$` - funkce S-registrů

Důležité AT příkazy podrobněji:
`ATIn` – (kde z n doplníme příslušné číslo)
0 – kód výrobce
1 – kontrolní součet RAM
2 – test RAM
3 – popis
4 – nastavení aktuální konfigurace (nastavení rychlost, opravy, komprese, atd.…)
5 – konfigurace uložená v NVRAM
6 – diagnostika linky posledního spojení
7 – informace o modemu
11 – zpráva o spojení

`ATDxxxx` – (za xxxx se doplní příslušné číslo)
`ATDPxxxx` – pulsní volba
`ATDTxxxx` – tónová volba
`ATZ` – resetování modemu + načtení konfigurace

Příkazy v příkazovém režimu:
Pauza 1s (nic nemačkat 1 sekundu) + + + (stisknout třikrát +) pauza 1s (nic nemačkat 1 sekundu) – úniková sekvence sloužící k přepnutí z přenosového do příkazového režimu
ATO0 – přechod z příkazového režimu do přenosového
ATO1 – přechod z příkazového režimu do přenosového + nové zasynchronizování
ATH0 nebo (ATH) – položí telefonní linku (ukončení spojení)
ATH1 – zvednutí linky

Výběr souboru:
CTRL+A <S> – přenos souboru. Vhodné je použít Z modem.
Při posílání vyberte předpřipravený adresář se soubory pomocí volby GOTO.
Klávesou SPACE označte zvolený soubor.
Nabídkou OKEY soubor pošlete.

Nastavení přenosových parametrů spojení: (Nejdůležitější příkazy rozšířené množiny příkazů podrobněji)

Tyto nastavení je nutné provést před vlastním navázáním spojení (tzn. mezi bodem 1 a 2). Těmito příkazy přímo definujeme vlastní parametry spojení.

AT&Nn – Nastavení přenosové rychlosti
0 – proměnlivá přenosová rychlost spojení
1 – 300b/s
2 – 1200b/s
3 – 2400b/s
4 – 4800b/s
atd…. více v příslušné nápovědě

AT&Mn – Oprava chyb
0 – režim bez opravy chyb (normální režim)
1 – oprava chyb pro synchronní komunikaci
4 – normální režim nebo oprava chyb s využitím metody ARQ
5 – oprava chyb ARQ nebo modem zavěsí

AT&Kn – Komprese dat
0 – komprese dat zablokována
1 – automatické povolení nebo zablokování
2 – komprese dat povolena
3 – selektivní komprese dat

(!!!nastavení rychlosti, opravy a komprese je nutné provést shodně na obou terminálech!!!)

Postup navázání spojení pomocí AT příkazů

- **Volající zadá:**

(!!!nastavení rychlosti, opravy a komprese je nutné provést shodně na obou terminálech!!!)

1) AT (ověření funkčnosti modemu)
2) AT&Nn (nastavení stanovené rychlost)

3) AT&Mn (nastavení opravu chyb)
4) AT&Kn (nastavení komprese dat)
5) ATDTxxxx (vytočení příslušného čísla)
6) Nyní se nacházíme v datovém režimu v kterém je možné pomocí kombinace CTRL+A a +S přenos data. Při přenosu si změříme celkovou dobu přenosu.
7) pauza 1s + + + pauza 1s (přepnutí do příkazového režimu)
8) zaznamenejte informace zobrazené pomocí příkazu ATI6
9) zaznamenejte informace zobrazené pomocí příkazu ATI11
10) ATH (ukončení daného spojení)
11) Opakujte postup se od bodu 3 (včetně) pro měření s opravou chyb a korekcí dat

♦ **Volaný zadá:**

(!!!nastavení rychlosti, opravy a komprese je nutné provést shodně na obou terminálech!!!)

7) AT
8) AT&Nn (nastavení stanovené rychlost)
9) AT&Mn (nastavení opravu chyb)
10) AT&Kn (nastavení komprese dat)
11) ATA (spojení komunikace v případě přijetí RING)
12) Nyní se nacházíme v datovém režimu, ve kterém je možné přijímat data
13) pauza 1s + + + pauza 1s
14) zaznamenejte informace zobrazené pomocí příkazu ATI6
15) zaznamenejte informace zobrazené pomocí příkazu ATI11
16) ATH (ukončení daného spojení)
17) Opakujte postup se od bodu 9 (včetně) pro měření s opravou chyb a korekcí dat

Po přenosu všech typů souborů je nutné změnit režim opravy chyb z volby „bez opravy chyb“ (AT&M0) na „normální režim nebo oprava chyb s využitím metody ARQ“ (AT&M4). Pro nastavení komprese použijeme režim „automatické povolení nebo zablokování“ (AT&K1) z „původní zablokované komprese dat“ (AT&K0).

Shrnutí pojmů

Datové měniče v přeloženém pásmu většinou využívají některý z typů modulace.

Modem je zkratka pro zařízení obsahující modulátor a demodulátor.

Skrambler, deskrambler – mění původní datovou sekvenci na pseudonáhodnou posloupnost.

Kodér, dekodér se využívá pro vícestavovou modulaci a řadí jednotlivé bity do skupin, které jsou dále modulovány.

Vstupní, výstupní obvody slouží pro úpravu signálu. Zde jsou používané různé korektory, zesilovače, omezovače apod.

Vidlice slouží k rozdělení dvoudrátového vedení na čtyřdrátové vedení a naopak.

Kompresní poměr udává poměr přenosové rychlosti komprimovaných dat k rychlosti původních nekomprimovaných dat.

Otázky

1. Jaké jsou základní funkce modemu?
2. Jakým způsobem lze zvýšit přenosovou rychlost modemu?
3. Jakým způsobem lze přepínat režimy práce modemu?

Další zdroje

[1] Němec, K. *Datová komunikace*. VUT Brno 2000, ISBN 80-214-1652-1

[2] Svoboda, J. *Základy teleinformatiky*. ČVUT Praha 1998, ISBN 80-901936-3-3

10. MODEMY PRO ŠIROKOPÁSMOVÉ KANÁLY

Čas ke studiu: 4 hodiny

Cíl Po prostudování tohoto odstavce budete umět

- definovat základní rozdělení modemů pro širokopásmové kanály
- popsat jednotlivé typy širokopásmových modemů
- analyzovat spektrum signálu pro vysokorychlostní modemy

Výklad

10.1. Analogové modemy pro širokopásmové kanály

Širokopásmové modemy se dříve používaly pro tzv. primární skupinu v systémech pro nosnou telefonii. Tyto modemy využívaly šířku pásma 48 KHz, ve frekvenčním spektru 60 – 108 KHz.

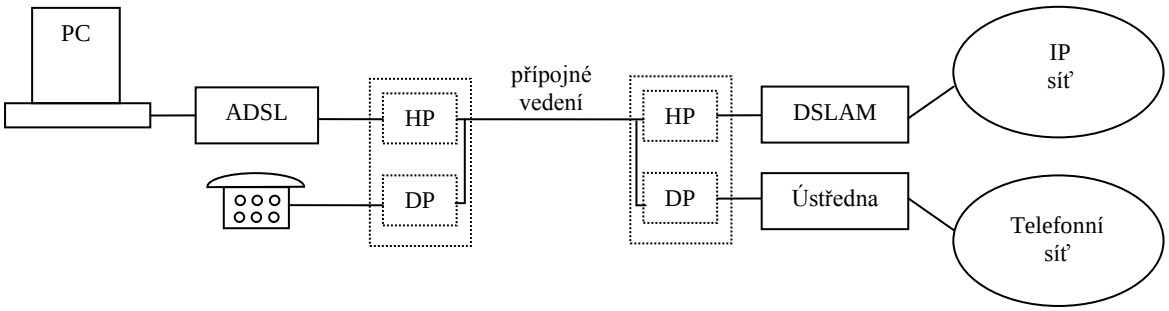
Pro tyto modemy se používala ASK modulace přes celou šířku pásma. Dosahovalo se maximálních rychlostí 168 kbit/s.

Tyto modemy jsou popsány v doporučeních V.35, V.36 a V.37.

10.2. Datové měniče ADSL

Datové měniče typu ADSL (Asymmetric Digital Subscriber Line) jsou digitální datové měniče, které využívají stávající metalické účastnické vedení.

Maximální přenosová rychlost závisí na délce přípojného vedení, průřezu vodiče, použití odboček, útlumu přeslechu a dalších rušivých vlivů.

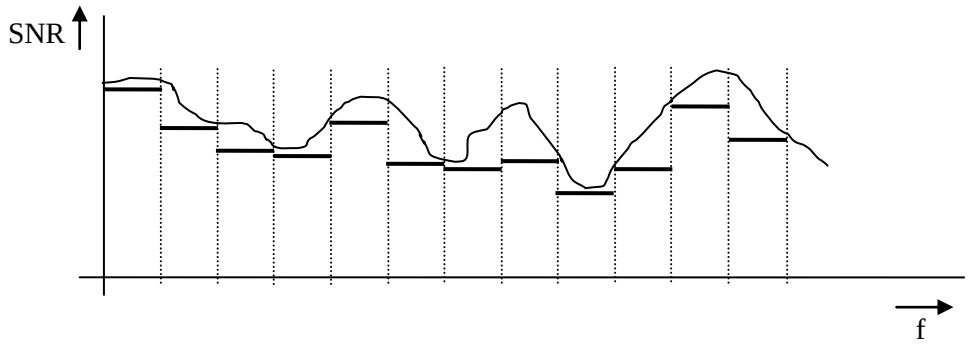


Obr. 10.1 Princip ADSL datového měniče

Aby byl možný současný přenos se stávající telefonní, nebo ISDN linkou, datový signál se moduluje nad hovorové pásmo. Pro oddělení provozu jsou použity filtry typu horní propust a dolní propust.

Současný přenos hovorového signálu a datového signálu může být realizováno dvěma způsoby. Starší způsob využívá frekvenčního multiplexu. U této metody jsou frekvenčně odděleny vysílací a přijímací pásma. Novější metoda využívá metody potlačení ozvěn, kdy je možný současný přenos signálů oběma směry. Tím je k dispozici více kanálů a celková přenosová rychlost je také vyšší.

Celé frekvenční pásmo je rozděleno do jednotlivých subpásem, kde je následně využita QAM modulace. Tato metoda se nazývá DMT – diskrétní multitónová modulace.

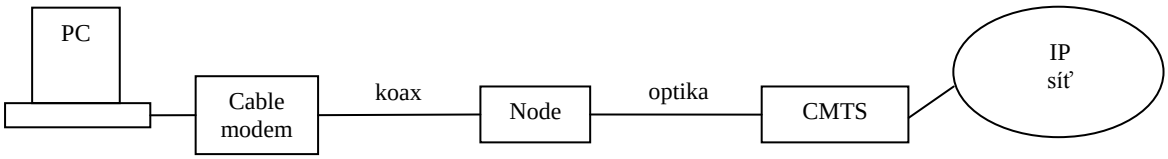


Obr. 10.2 Frekvenční průběh SNR

Na předchozím obrázku je naznačeno, jak je frekvenční pásmo rozděleno do jednotlivých subkanálů. Pro každý subkanál je ze střední hodnoty SNR (Signal to Noise Ratio) odvozen maximální počet bitů na symbol.

10.3. Kabelové modemy

Kabelové modemy využívají kabelových sítí pro televizní rozvody. Nejčastěji jsou připojovány na koaxiální rozvody. Využívají tzv. HFC (Hybrid Fiber Coax) sítí, což je kombinace optických a koaxiálních rozvodů.



Obr. 10.3 Architektura HFC sítí

CMTS (Cable Modem Termination System), někdy se také nazývá headend, je zařízení používané v sítích kabelových operátorů pro širokopásmové služby. Node slouží pro převedení signálu z optického tvaru na elektrický a naopak. K jednomu CMTS může být připojeno několik kabelových modemů.

Kabelové modemy využívají nejnižší dvě vrstvy referenčního modelu OSI. Jsou standardizovány v průmyslovém standardu DOCSIS (Data Over Cable Service Interface Specification), případně EuroDOCSIS. Jsou zde specifikována pravidla pro obousměrné datové přenosy po stávajících kabelových televizních rozvodech.

Tyto modemy využívají několik typů modulací. Nejpoužívanější je však kvadraturní amplitudová modulace QAM a kvadraturní klíčování fázovým posunem QPSK.

Kabelové modemy jsou koncipovány tak, aby nevyžadovaly žádnou konfiguraci ze strany uživatele. Veškeré potřebné parametry si automaticky stahují ze sítě pomocí protokolu TFTP (Trivial File Transfer Protocol).

Přidělování IP adres zajišťuje DHCP (Dynamic Host Control Protocol) a management může být založen na protokolu SNMP (Simple Network Management Protokol). DOCSIS/euroDOCSIS používá na linkové vrstvě MAC (Medium Access Control) adresu, která slouží kromě adresování také pro identifikační a autorizační procedury.

10.4. Měření - Datové přenosy pomocí ADSL

Zadání:

Pomocí příkazu ping zjistěte zpoždění paketů mezi Serverem a Klientem pro poskytovatelem nastavenou přenosovou rychlost. Při přenosu vyšlete vždy 10 dotazů najednou. Postupně volte 10 různých velikostí paketu, tak aby vhodně zastoupili celé spektrum možných velikostí MTU a zaznamenejte všechny dostupné výsledné informace. Proveďte diagnostiku spojení pomocí diagnostických nástrojů v ADSL směrovači. Změřené hodnoty graficky zpracujte, zhodnoťte výsledky měření i diagnostiky.

Použité zařízení pro měření:

- Server – PC11_N312 s OS Linux
- DSLAM – ZyXEL IES-1000
- Simulátor délky vedení – Telebyte 458-LM-E20 (0-6150m/modul)
- ADSL směrovač – ZyXEL Compact P-660R-61C
- Klient – PCx_N312 s OS Linux

Postup pro nastavení DSLAMu:

- Propojte Server seriovým kabelem na terminálový port (Console port) přístupového multiplexeru DSLAM. Po spuštění Serveru s OS Linux spusťte aplikaci `minicom`.
- Po vstupní inicializaci a zadání jména `admin` a hesla `1234` ověřte nastavení IP DSLAMu (v našem případě **192.168.100.1**) příkazem `ip info show`.
- Nakonfigurujte IP adresu serveru na `192.168.100.10`.
- Propojte Server a DSLAM síťovým kabelem a přihlaste se přes webový prohlížeč, zadáním adresy `192.168.100.1`, do konfiguračního menu, kde pro ověření opět zadejte přihlašovací údaje.

Postup měření:

1. Propojte ADSL směrovač s PC pomocí kříženého síťového kabelu.
2. Propojte ADSL směrovač s DSLAMem pomocí telefonního kabelu.
3. Získejte IP z DHCP poskytovaného ADSL směšovačem `dhclient eth1`
4. Zpoždění paketu lze měřit příkazem `ping 192.168.100.10 -c10 -s100`

kde 192.168.100.10 je IP adresa Serveru

–c udává počet vyslaných dotazů ping

–s udává velikost paketu v bytech.

5. Z výpisu zaznamenejte jak nejmenší (min), střední (avr) a největší (max) doba zpoždění i časová odchylka.
6. Zadáním adresy `192.168.x.x` (IP zjistíte v bode 3.) ve vebovém prohlížeči se připojte do konfiguračního menu vašeho ADSL směrovače.
7. Přihlašovací heslo je `12345`

Vyhodnocení měření:

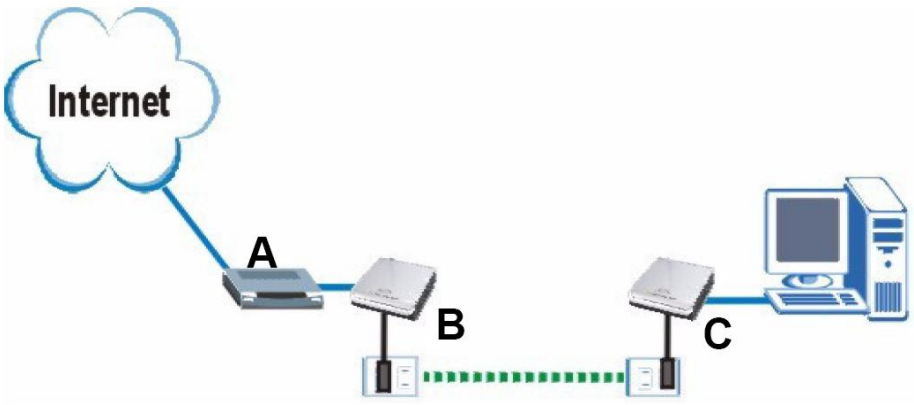
- V položce `Main Menu/Maintenance/System Status` zjistěte nastavenou adresu vnější sítě a použité **VPI/VCI**.
- V položce `Main Menu/Maintenance/Systém Status/Show Statistic` určete poskytovatelem nastavenou rychlost **Up/Down**stream.
- V položce `Main Menu/Maintenance/Diagnostic` zjistěte na jakých subkanálech se přenáší informace ve směru **Up/Down**stream
- Zjistěte přenosový výkon pomocí stažení souboru z adresy http://kat440.vsb.cz/download/pd/

10.5. Měření – Datové přenosy pomocí Powerline modemů

Zadání:

Pomocí příkazu **ping** zjistěte zpoždění paketů mezi Serverem a Klientem pro poskytovatelem nastavenou přenosovou rychlost. Při přenosu vyšlete vždy 10 dotazů najednou. Postupně volte 10 různých velikostí paketu, tak aby vhodně zastoupili celé spektrum možných velikostí MTU a zaznamenejte všechny dostupné výsledné informace. Změřené hodnoty graficky zpracujte, zhodnoťte výsledky měření i diagnostiky. Pokuste se zjistit na jakou vzdálenost uvedený typ modemu dokáže komunikovat (teoreticky z manuálu/prakticky měřením).

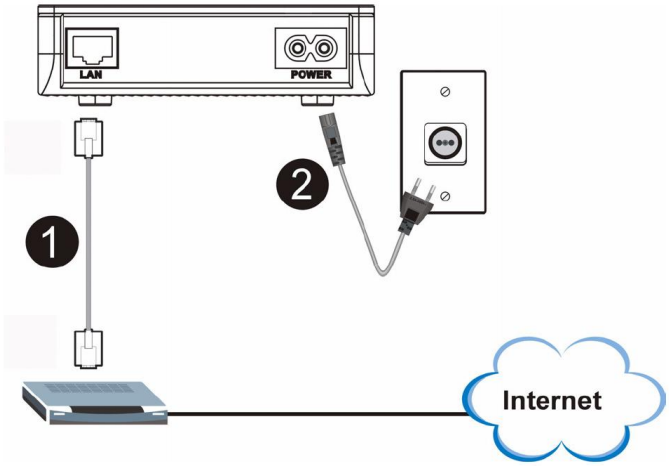
Schéma:



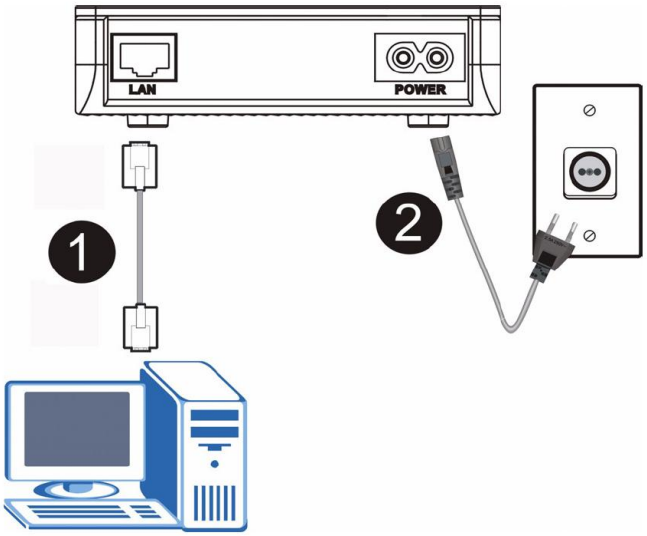
Použité zařízení pro měření:
Router Ovislink Airlive vpn 9000
Powerline modem ZyXEL PLA-400
Klient – PCx_N312 s OS Linux (distribuce Debian)

Postup připojení powerline modemu:

1. Nejprve je nutno připojit „master modem“ (modem jež je připojen k routeru/dslamu/switchy…)



2. Propojit PC s vlastním powerline modemem pomocí kříženého kabelu.



Klientská stanice by měla dostat automaticky přiděleno IP z DHCP serveru (pokud v síti DHCP běží), pokud se tak nestane musí se IP nastavit ručně.

Postup měření:

A)

1. Propojte Powerline modem s PC pomocí kříženého síťového kabelu s konektory RJ-45.
2. Propojte Router s „master“ Powerline modemem pomocí kříženého síťového kabelu s konektory RJ-45.
3. Zkontrolujte připojení zařízení do sítě.
4. Spusťte OS Linux.
5. Přihlaste se jako administrátor
6. Získejte IP z DHCP Poolu, poskytovaného routerem *Ovislink Airlive vpn 9000*, pomocí příkazu: **dhclient eth1** Pokud v síti neběží DHCP server, nastavte IP ručně pomocí příkazu: **ifconfig eth1 192.168.1.xx netmask 255.255.255.0** (kde „xx“ značí číslo vašeho pracoviště)
7. Nyní můžeme přistoupit k vlastnímu měření. Zpoždění paketu změříte pomocí příkazu **ping 192.168.1.254 –c 10 –s 100**

kde **192.168.1.254** je IP adresa Routeru
–c udává počet vyslaných dotazů ping
–s udává velikost paketu v bytech.
Z výpisu zaznamenejte jak nejmenší (min), střední (avr) a největší (max) doba zpoždění i časová odchylka

B)

8. Vyhledejte informaci o **maximální možné přenosové rychlosti**, kterou tento typ modemu podporuje a **maximální vzdálenost/velikost sítě** na kterou může modem komunikovat.
9. Rychlost otestujte měřením (například na http://speedtest.net).
10. Otestujte pomocí přenosného laptopu maximální vzdálenost použitelnosti technologie (stačí jedno měření za celou skupinu).

Shrnutí pojmů

Datové měniče typu ADSL jsou digitální datové měniče, které využívají stávající metalické účastnické vedení.

Kabelové modemy využívají kabelových sítí pro televizní rozvody.

Datové měniče v přeloženém pásmu většinou využívají některý z typů modulace.

Modem je zkratka pro zařízení obsahující modulátor a demodulátor.

DOCSIS je mezinárodní standard pro kabelové modemy, definující rozhraní.

Otázky

1. Jaké jsou nevýhody ASK modulace?
2. Na čem závisí maximální přenosová rychlost u ADSL modemů?
3. Jaký typ kabelu je použit pro připojení kabelového modemu?

Další zdroje

[1] Němec, K. *Datová komunikace*. VUT Brno 2000, ISBN 80-214-1652-1

[2] Svoboda, J. *Základy teleinformatiky.* ČVUT Praha 1998, ISBN 80-901936-3-3

[3] The Unicode Consorcium, http://unicode.org/

11. DATOVÉ SÍTĚ

Čas ke studiu: 4 hodiny

Cíl Po prostudování tohoto odstavce budete umět

- definovat základní druhy datových sítí
- popsat jednotlivé typy datových sítí
- analyzovat propustnost datové sítě

Výklad

Pro přenos dat se používá buď hostitelských telekomunikačních sítí, nebo sítí vytvořených pro účely přenosu dat. Dnes vznikají tzv. integrované sítě, které jsou určeny pro všechny typy signálů.

Z hlediska vzdálenosti koncových zařízení rozdělujeme datové sítě na sítě lokální LAN (Local Area Network), metropolitní MAN (Metropolitan Area Network) a rozlehlé WAN (Wide Area Network).

11.1. Lokální datové sítě

Mezi základní vlastnosti lokálních datových sítí je její omezený dosah a velmi vysoké přenosové rychlosti.

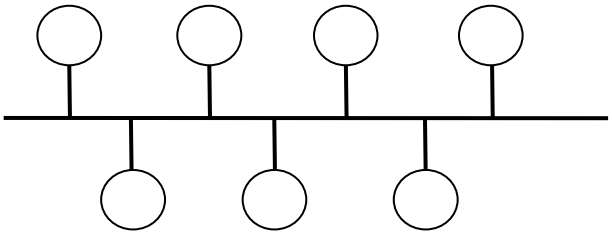
Volba topologie neboli uspořádání koncových zařízení má vliv na řadu vlastností. Mezi základní vlastnosti patří spolehlivost sítě. Je to vlastně odolnost sítě proti poruchám.

Další důležitou vlastností je rozšiřitelnost neboli možnost zařazení dalšího koncového zařízení s minimálními náklady. Také je velmi důležitá výkonnost sítě, která znamená využití kapacity linky, zpoždění signálu apod.

V neposlední řadě hraje důležitou roli i cena. Zde jsou zahrnuty všechny pořizovací a udržovací náklady. Mezi významné vlastnosti lze zařadit i možnost dálkového dohledu a správy sítě.

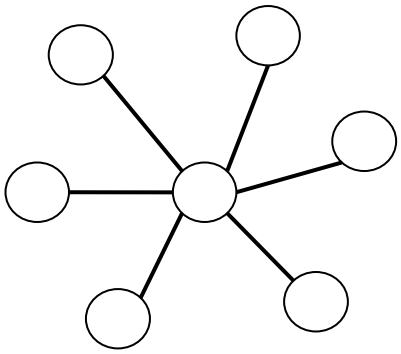
11.2. Topologie sítě

Topologie typu **sběrnice** je jednoduchá, snadno rozšiřitelná. Není ovšem vhodná pro větší počet koncových zařízení. Je také velmi citlivá na výpadek linky.



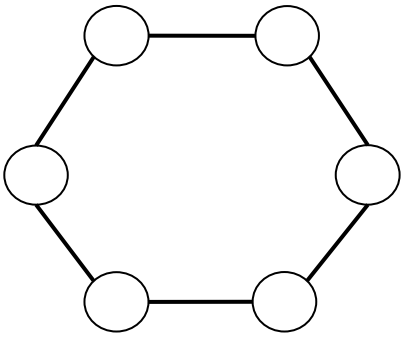
Obr. 11.1 Topologie typu sběrnice

Topologie typu **hvězda** je charakterizováno spolehlivostí, lepší správou. Tato topologie je odolná na výpadek linky, nebo stanice. Je velmi citlivá na výpadek centrálního uzlu.



Obr. 11.2 Topologie typu hvězda

Topologie typu **kruh** je charakterizována vysokou spolehlivostí. Většinou využívá deterministický způsob řízení přístupu ke společnému médiu. Je citlivá na výpadek linky nebo stanice.



Obr. 11.3 Topologie typu kruh

11.3. Řízení přístupu ke společnému médiu

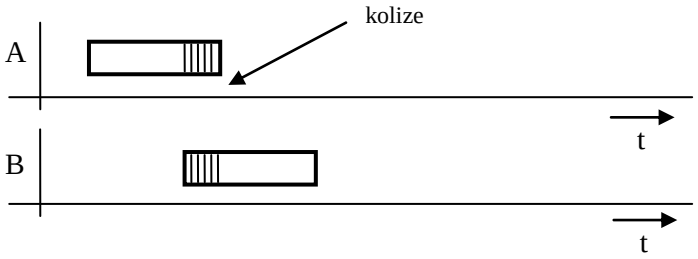
Protože pro datové sítě je charakteristické, že využívají stejné přenosové médium, musí se nějakým vhodným způsobem řešit přístup k tomuto médiu.

Existují dva základní principy řízení přístup ke společnému médiu. Prvním typem jsou tzv. deterministické metody, kde je přesně určeno, kdy bude která stanice vysílat. Druhým typem jsou tzv. stochastické neboli náhodné metody, kdy není přesně určeno, kdy stanice bude mít přístup k médiu.

Deterministické metody řízení k přístupu se dále dělí na centralizované a distribuované. Centralizované metody jsou založeny na tom, že jedna stanice je řídící a přiděluje ostatním stanicím kapacitu kanálu. Přidělování může být buď na výzvu, kdy řídící stanice postupně vyzívá podřízené stanice k odeslání zprávy, nebo přidělování na žádost, kdy řídící stanice rozhodne na základě požadavků od jednotlivých stanic.

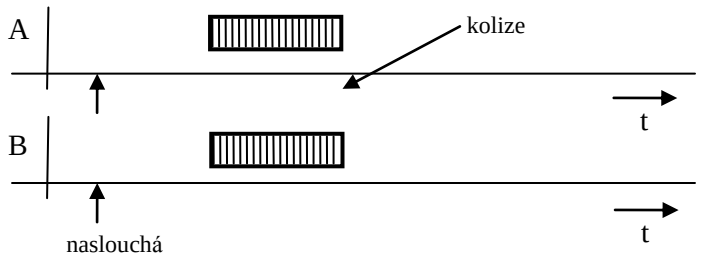
Distribuované metody řízení nemají žádnou řídící stanici. Pro řízení přístupu používají buď rezervační metodu, nebo prioritní přístup.

Stochastické metody řízení přístupu ke společnému přenosovému médiu vyžívají náhodného přístupu. Tyto metody se postupně vyvíjely. První metodou byla tzv. ALOHA (Aditive Link On-line Hawaii Area), která byla právě poprvé použita na Havajských ostrovech. Tato metoda je založena na tom, že pokud chce stanice vysílat, tak začne vysílat, bez ohledu na provozu v daném kanále. Dalším vylepšením této metody je tzv. synchronní ALOHA, která je založena na tom, že stanice vysílá pouze v určitých časových intervalech.



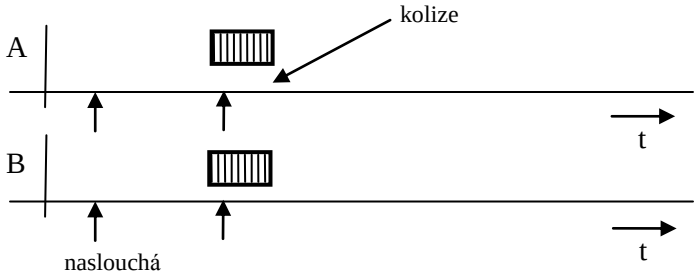
Obr. 11.4 Princip přístupové metody ALOHA

Další metoda je metoda CSMA (Carrier Sense Multiple Access), která se nazývá jako vícenásobný přístup s naslouchání nosné. Tato metoda je založena na tom, že stanice před tím než začne vysílat, musí naslouchat, zdali je sdílené médium volné. Poté může začít s vysíláním svých dat. Vysílá po celou dobu, i když nastala kolize.



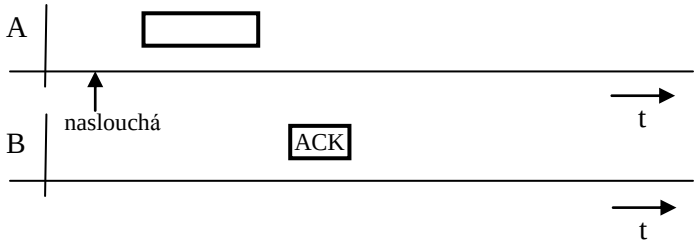
Obr. 11.5 Princip přístupové metody CSMA

Tato metoda byla zdokonalena v metodu CSMA/CD (Carrier Sense Multiple Access/ Collision Detection), která se snaží detekovat případné kolize. Kolize se projeví například zvýšením napětí na lince. Stanice přestane vysílat. Stanice je potom povinna za náhodnou dobu znova odvysílat narušený rámec dat. Tato metoda je využívaná například u technologie Ethernet.



Obr. 11.6 Princip přístupové metody CSMA/CD

Další metodou je metoda CSMA/CA (Carrier Sense Multiple Access/ Collision Avoidance), která na rozdíl od předchozí metody nezjišťuje výskyt kolize. Stanice při volném médiu počká náhodně zvolenou dobu a poté odvysílá svůj datový rámec a počká na jeho potvrzení. Tato metoda je například používaná u bezdrátových sítí Wi-Fi.



Obr. 11.7 Princip přístupové metody CSMA/CA

11.4. Přenosová média

Pro datové sítě se používají různé typy přenosových médií. Základní používané typy médií jsou metalické, optické a bezdrátové způsoby přenosu signálů.

Koaxiální kabely jsou definovány jako nesymetrické vedení. Mezi základní vlastnosti patří charakteristická impedance, která je dána průměrem vnějšího a vnitřního vodiče a použitým dielektrikem. Dalším základním parametrem je měrný útlum, který vyjadřuje, kolikrát se zmenší výkon signálu po průchodu kabelem definované délky. Tento parametr je kmitočtově závislý.

Kroucené páry jsou definovány jako symetrické vedení. Důvodem kroucení vodičů je zlepšení elektrických vlastností kabelu. Minimalizují se přeslechy mezi páry a snižuje se vzájemné ovlivňování mezi kabelem a okolím. Tím je omezeno vyzařování elektromagnetického záření do okolí i jeho příjem z okolí.

Optická vlákna jsou skleněná nebo plastová, která přenáší signály pomocí světla. Vlákna přenášejí signály s menší ztrátou a jsou imunní vůči elektromagnetickému rušení. Mezi základní vlastnosti patří útlum, který je ovlivněn materiálovou absorpcí, rozptylem, ohybem a ztrátami při spojování. Disperze je příčinou zkreslení přenášeného signálu, dochází ke zpožďování impulsů a změně jejich tvaru. Dalším parametrem je numerická apertura, která vyjadřuje schopnost optického vlákna navázat optický výkon do svého jádra.

11.5. Datové převodníky

Datový převodník typu Ethernet – RS232

Standard **RS 232** uvádí jako maximální možnou délku vodičů **15 metrů**, nebo délku vodiče o kapacitě 2500 pF. To znamená, že při použití kvalitních vodičů lze dodržet standard a při zachování jmenovité kapacity prodloužit vzdálenost až na cca 50 metrů. Kabel lze také prodlužovat při snížení přenosové rychlosti, protože potom bude přenos odolnější vůči velké kapacitě vedení. Uvedené parametry počítají s přenosovou rychlostí 19200 Bd.

Tab.4.1: maximální možná délka vodičů dle standardu RS232

• Baud rate [Bd]	• Max length [m]
• 2 400	• 900
• 4 800	• 300
• 9 600	• 150
• 19 200	• 15

Převodníky typu Ethernet-RS232 umožňují rozšířit možnosti použití aplikací používající sériový port umístěný ve vzdálenosti větší, než je povoleno standardem RS232. Různé průmyslové aplikace často používají sériový port. S převodníky typu RS232/Ethernet lze dané aplikace ovládat s použitím internetu prakticky odkudkoliv.

Moxa NPort 5110

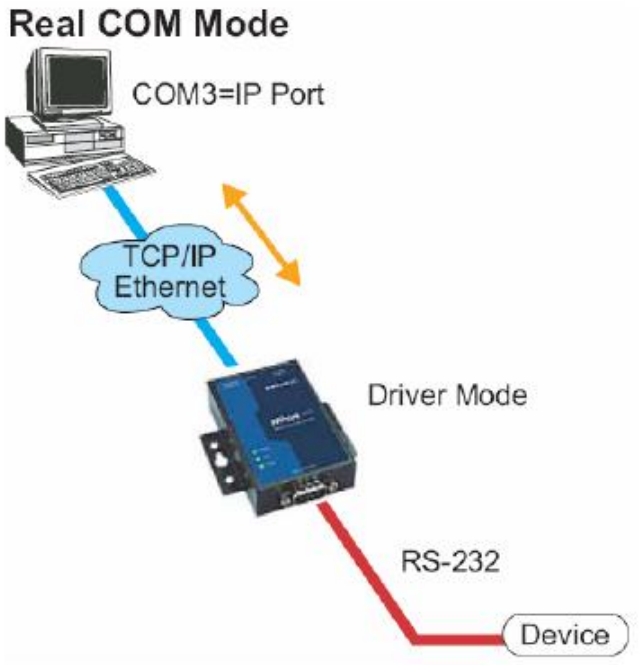
Převodníky typu NPort 5100 jsou navrženy tak, aby je bylo možno jednoduše, rychle a variabilně nasadit k ovládání a datovým přenosům různých průmyslových aplikací používající pro připojení sériové RS-232/422/485 zařízení, jako jsou čtečky karet, platební terminály, různé typy senzorů, ovládání a konfigurace řídících zařízení. Ovládací terminál může být díky použití Ethernetu umístěny kdekoliv na lokální síti LAN, nebo na Internetu.

NPort 5100 podporuje provoz několika režimech, včetně TCP Server, TCP Client, UDP Server/Client, páruje připojení Ethernet-Modem, zajištění slučitelnosti sítí, jenž používá standardní síťové rozhraní API (Winsock, BSD Sockets).

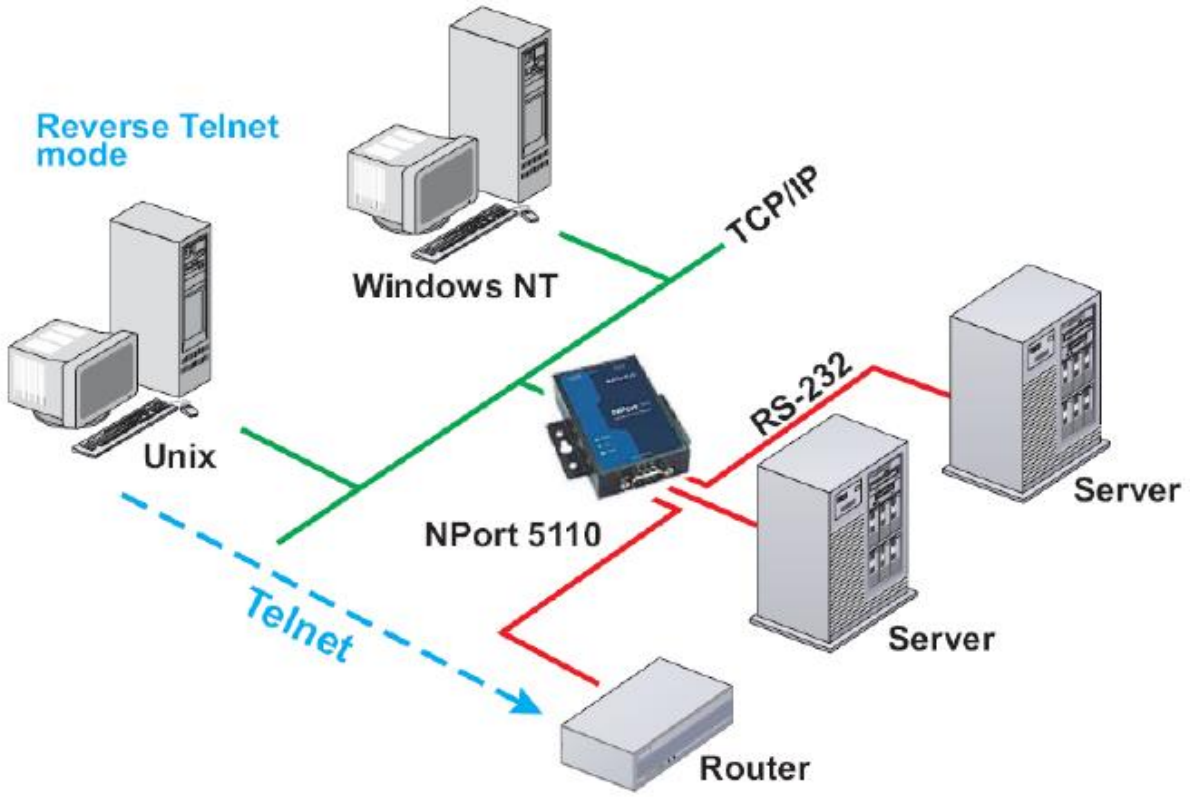
Obr. 11.8 Převodník Moxa NPort 5110

Základní funkce NPort 5100:

- nízkonákladové zařízení (velikosti kreditní karty)
- Real COM / TTY ovladače pro Windows a Linux
- univerzální zásuvka provozní režimy: TCP Server, TCP Client, UDP, a Ethernet Modem
- způsob připojení pro připojení dva sériové zařízení přes počítačovou síť bez PC
- zabudované 15 KV ESD ochrana pro všechny signály sériového rozhraní
- konfigurace přes web / telnet / sériová konzole



Obr. 11.9 Způsoby zapojení převodníku Moxa NPort 5110 – Real COM



Obr. 11.10 Způsoby zapojení převodníku Moxa NPort 5110 – Telnet mode

POPIS KONFIGURACE PŘEVODNÍKU MOXA V OS LINUX

- Nastavení IP adresy pomocí telnetu. Default IP address **192.168.127.254**.

Type **1** to select **IP address** and then press **Enter**.

```
<< Main Menu->Network settings >>
  (1) IP address
  (2) Netmask
  (3) Gateway
  (4) IP configuration
  (5) DNS server 1
  (6) DNS server 2
  (7) SNMP
  (8) SNMP community name
  (9) SNMP contact
  (a) SNMP location
  (b) Auto IP report to IP
  (c) Auto IP report to TCP port
  (d) Auto IP report period
  (v) View settings
  (m) Back to main menu
  (q) Quit

Key in your selection: 1_
```

```
<< Main Menu->Network settings >>
  (1) IP address
  (2) Netmask
  (3) Gateway
  (4) IP configuration
  (5) DNS server 1
  (6) DNS server 2
  (7) SNMP
  (8) SNMP community name
  (9) SNMP contact
  (a) SNMP location
  (b) Auto IP report to IP
  (c) Auto IP report to TCP port
  (d) Auto IP report period
  (v) View settings
  (m) Back to main menu
  (q) Quit

Key in your selection: 1
IP address: 192.168.127.253_
```

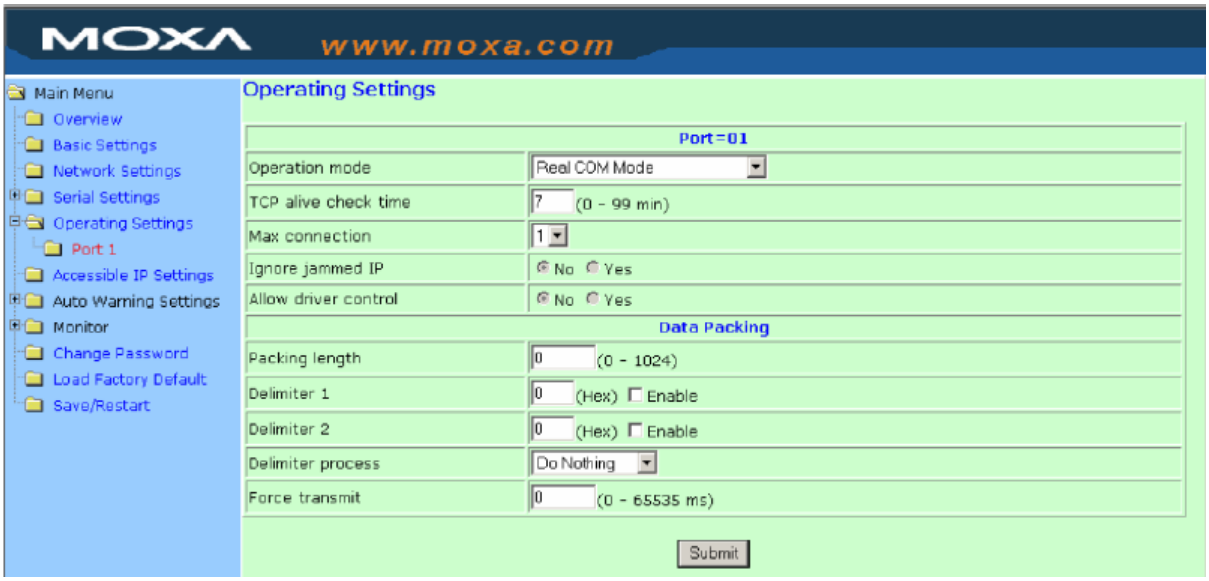
Obr.11.11 Nastavení pomocí telnetu

- Nastavení pomocí sériového rozhraní: nutno nakonfigurovat port na 19 200 a 8n1.
- Nastavení pomocí html rozhraní:

Do adresového řádku prohlížeče zadejte: http://192.168.127.254

Obr. 11.12 Základní menu Moxa

Obr. 11.13 Nastavení seriál portu Moxa



Obr. 11.14 Nastavení Real mode

Dále je potřeba nainstalovat ovladače virtuálního portu do systému. Postup je detailně popsán v souboru „readme.txt“. Výňatek z plné verze návodu:

Installing the Driver File

a. *Copy the driver file from Moodle454.*
b. *Log in to the console as a super user (root).*
c. *Execute* `cd /` *to change to the root directory.*
d. *Copy the driver file* `moxa.tgz to` *the " / " directory.*
e. *Execute* `tar xvfz moxa.tgz` *to copy all files into the system.*
f. *Execute* `/tmp/moxa/mxinst`.
g. *The shell script will install the driver files automatically.*

After installing the driver, you will be able to see several files in the "/usr/lib/npreal2/driver" folder, including:

- **mxaddsvr** (Add Server, map tty port)
- **mxsetsec** (Set secure communication mode)
- **mxdelsvr** (Delete Server, un-map tty port)
- **mxloadsvr** (Reload Server)
- **mxuninst** (Remove tty port and driver files)

Mapping TTY Ports

Before mapping tty ports, you must set the operation mode of your NPort to Real Com Mode. Mapping tty ports automatically: After logging in as a super user, enter the direktory "/usr/lib/npreal2/driver" and then execute "mxaddsvr" to map the target NPort serial port to the host tty ports. The syntax of "mxaddsvr" is:

`mxaddsvr` [NPort IP Address] [Total Ports] ([Data port] [Cmd port])

Example 1:

```
# cd /usr/lib/npreal2/driver
# ./mxaddsvr 192.168.3.4 16
```

16 tty ports will be added, all with IP 192.168.3.4, but with data ports equal to (950, 951, ..., 965), and command ports equal to (966, 967, 968, ..., 981).

Example 2:

```
# cd /usr/lib/npreal2/driver
# ./mxaddsvr 192.168.3.4 16 4001 966
```

6 tty ports will be added, all with IP 192.168.3.4, but with data ports equal to (4001, 4002, ..., 4016), and command ports equal to (966, 967, 968, ..., 981).

Remove Mapped TTY Ports

Remove the mapped tty ports automatically: After logging in as root, enter the directory "/usr/lib/npreal2/driver" and then execute `mxdelsvr` *to delete a server. The syntax of "mxdelsvr" is:*

```
mxdelsvr [IP]
```

Example:

```
# cd /usr/lib/npreal2/driver
# ./mxdelsvr 192.168.3.4
```

Removing the Driver

Removing the driver will remove all driver files, mapped tty ports, and unload the driver. To do this, enter the directory „/usr/lib/npreal2/driver", and then execute "mxuninst" to uninstall the driver. This program will perform the following tasks:

- *Unload the driver.*
- *Delete all files and directories in "/usr/lib/npreal2"*
- *Delete directory "/usr/lib/npreal2"*
- *Modify the system initialization script file.*

Převodník typu Ethernet – optika (používané v laboratoři N312)

Obr. 11.15 Převodník TP-Link MC210CS – SingleMode

Model NO.	Interface	Transmission Distance	Transmission Media	Output Center Wavelength
MC200CM	RJ45--SC	0.55km(50/125um), 0.22km(62.5/125um)	Multi-mode Fiber,TP	850nm
MC210CS	**RJ45--SC**	**15km**	**Single-mode Fiber,TP**	**1310nm**
MC220L	RJ45--LC	0.55km /10km	Multi/Single-mode Fiber, TP	850nm/1310nm

Obr. 11.16 Převodník Tenda TER850S – MultiMode

Shrnutí pojmů

Lokální datové sítě mají omezený dosah a vysoké přenosové rychlosti.

Topologie sítě má vliv na řadu vlastností.

Deterministické metody řízení přístupu ke společnému médiu přesně určují, kdy bude stanice vysílat

Stochastické metody řízení přístupu ke společnému médiu neurčují, kdy stanice začne vysílat.

Distribuované metody řízení přístupu ke společnému médiu nemají řídící stanici.

ALOHA náhodná přístupová metoda

CSMA vícenásobný přístup s nasloucháním nosné.

CSMA/CD vícenásobný přístup s nasloucháním nosné s detekcí kolizí.

CSMA/CA vícenásobný přístup s nasloucháním nosné bez zjišťování kolizí.

Koaxiální kabely jsou definovány jako nesymetrické vedení.

Kroucené páry jsou definovány jako symetrické vedení.

Optické vlákna přenášejí signál pomocí světla.

Otázky

1. Jaké jsou základní typy topologií sítí?
2. Jaký má vliv topologie sítě na její propustnost?
3. Může u přístupové metody CSMA/CA vzniknout kolize?

Další zdroje

[1] Němec, K. *Datová komunikace*. VUT Brno 2000, ISBN 80-214-1652-1

[2] Svoboda, J. *Základy teleinformatiky.* ČVUT Praha 1998, ISBN 80-901936-3-3

12. BEZDRÁTOVÉ DATOVÉ SÍTĚ

Čas ke studiu: 4 hodiny

Cíl Po prostudování tohoto odstavce budete umět

- definovat základní rozdělení bezdrátových datových sítí
- popsat jednotlivé typy technologií pro bezdrátový přenos dat
- analyzovat propustnost bezdrátové datové sítě

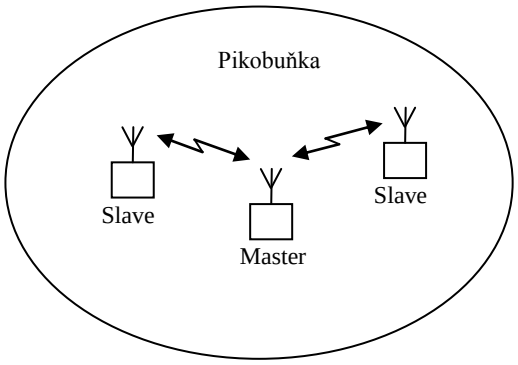
Výklad

Bezdrátové datové sítě se dají rozdělit podle dosahu koncových zařízení na různé typy. U každého typu sítě je potom použita vhodná přenosová technologie.

12.1. Bezdrátové osobní sítě

Bezdrátové osobní sítě se v anglickém originále označují jako WPAN (Wireless Personal Area Network). Tento typ sítí je určen do vzdálenosti do cca 10 m. Vzdálenost většinou závisí na vysílacím výkonu, který bývá předepsán. Tyto WPAN sítě jsou popsány v doporučení IEEE 802.15.

V současné době je nejpoužívanější technologií Bluetooth, která je popsána v doporučení 802.15.1. Pracuje v ISM pásmu na kmitočtu 2.4 GHz. K přenosu využívá metod FHSS (Frequency Hopping Spread Spectrum), kdy využívá 1600 přeskoků za sekundu mezi 79 frekvencemi. Dosahuje maximální přenosovou rychlost 1Mbit/s. V současné době je definována specifikace Bluetooth 4.0, která umožňuje maximální přenosovou rychlost až do 24Mbit/s, má sníženou spotřebu a umožňuje šifrování AES-128.



Obr. 12.1 Princip přenosu dat technologií Bluetooth

Blutooth je paketově orientovaný protokol. Zařízení mezi sebou komunikují přes řídící stanici (Master) v rámci tzv. pikobuňky. Jedna Master stanice může řídit až 7 Slave stanic.

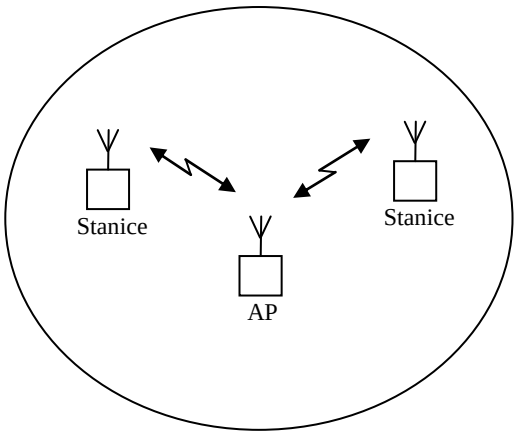
Využití technologie Bluetooth je zejména pro bezdrátové připojení různých příslušenství. V nové verzi se předpokládá i větší využití pro vzájemné propojení různých zařízení.

12.2. Bezdrátové lokální sítě

Bezdrátové lokální sítě se v anglickém originále označují jako sítě WLAN (Wireless Local Area Network. Tento typ sítí je převážně určen pro komunikaci na vzdálenosti do 100 m.

Bezdrátové lokální sítě jsou mezi uživateli velmi oblíbené a často používané. Jsou standardizovány doporučením IEEE 802.11. Pracují v bezlicenčním pásmu ISM, na kmitočtech buď 2.4 GHz, nebo na kmitočtech 5 GHz.

Datové přenosy mohou být realizovány přímo mezi dvěma stanicemi, tzv. "ad-hoc"síti. Ale častější je ale způsob komunikace mezi několika stanicemi, kdy je využitý centrální prvek s jedním, nebo několika přístupovými body, které řídí veškerou komunikaci.



Obr. 12.2 Infrastrukturní zapojení stanic a přístupového bodu AP

Důležitou roli při rozlišování jednotlivých sítí hraje identifikátor, označovaná jako SSID (Service Set Identifier). Tento řetězec může obsahovat až 32 ASCII znaků. V některých případech bývá tento identifikátor skryt a připojující se stanice musí tento řetězec znát.

WLAN sítě používají různé způsoby modulace. Poslední standard například využívá dnes nejrozšířenější metodu OFDM (Orthogonal Frequency Division Multiplexing).

Pro bezdrátové komunikace je velmi důležité zabezpečení sítě, zejména proto, že se signál šíří i mino zabezpečený prostor. Mezi jednoduchý způsob zabezpečení, patří kontrola MAC (Media Access Control) adres na přístupovém bodě.

Další možností zabezpečení je pomocí statických WEP (Wired Equivalent Privacy) klíčů a symetrického šifrování, které jsou ručně nastavovány na jednotlivých zařízeních.

Lepším způsobem zabezpečení je využívání dynamických klíčů, pomocí standardu WPA (Wi-Fi Protected Access). V tomto případě se většinou využívá autentizační server RADIUS (Remote Authentication Dial In User Service). Vhodnější je používat WPA2, která přináší silnější šifrování podle standardu AES (Advanced Encryption Standard).

12.3. Bezdrátové metropolitní sítě

Bezdrátové metropolitní sítě se v anglickém originále označují jako WMAN (Wireless Metropolitan Area Network). Tento typ sítí je typický pro městské aglomerace a používá se na vzdálenosti do 50 km.

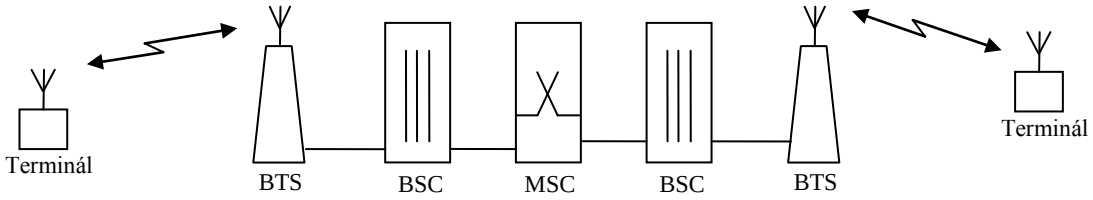
Tyto sítě jsou standardizovány doporučením IEEE 802.16 a označují se WiMAX (Worldwide Interoperability for Microwave Access). Tyto sítě využívají jak bezlicenční, tak licenční frekvenční pásma v rozsahu 2-11 GHz.

WiMAX používá OFDM modulaci a maximální dosahovaná rychlost je 70 Mbit/s. Tato technologie není v současné době mnoho využívaná. Spíše lze očekávat větší využívání v mobilních sítích 4. generace – 4G sítě.

12.4. Bezdrátové rozlehlé sítě

Bezdrátové rozlehlé sítě označované jako WWAN (Wireless Wide Area Network) jsou sítě využívané u mobilních operátorů. Podle použité technologie je lze rozdělit do několika generací.

První typ mobilních technologií je systém GSM (Global System for Mobile communication). Tato technologie byla původně určena převážně pro hlasové služby. Lze zde také přenášet data s přenosovou rychlostí 9,6 kbit/s. Využívá modulaci GMSK (Gaussian Minimum Shift Keying).



Obr. 12.3 Přenos dat v sítích GSM

Na předchozím obrázku je naznačen způsob přenos dat v sítích GSM. Terminál komunikuje se základnovou stanicí BTS (Basic Tranceiver Station). Několik základnových stanic může být propojeno na řídicí systém pro BTS, tzv. BSC (Base System Controller). Jednotlivé řídicí systémy jsou napojeny na MSC (Mobile Switching Centre), který je ve funkci mobilní přepínací ústředny.

Rozšířením GSM vznikla mobilní datová služba GPRS (General Packet Radio Service). Využívá TDMA (Time Division Multiplexing Access) kanály v GSM síti. Podle kódovacího schématu a počtu použitých timeslotů, může dosáhnou teoretické přenosové rychlosti až 160 kbit/s. Skutečná rychlost závisí na vytíženosti dané buňky, ve které se mobilní terminál nachází.

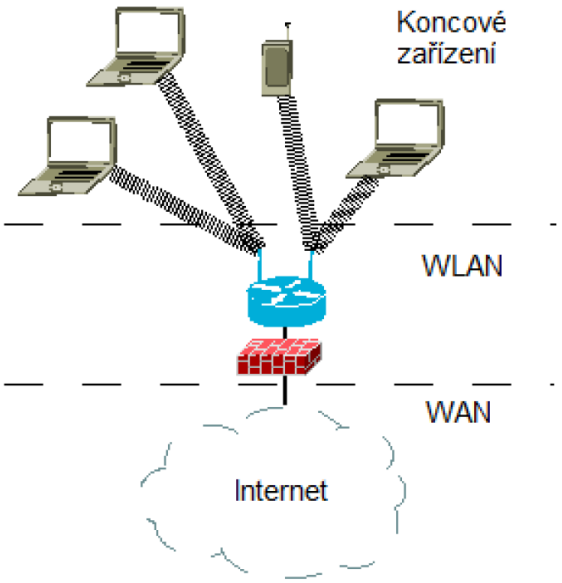
Dalším typem mobilních služeb je UMTS (Universal Mobile Telecommunication System). Je označovaný jako 3G standard mobilních telefonů. Pro přístup používá technologii W-CDMA (Wideband Code Division Multiple Access), která může být dále kombinována TDMA nebo FDMA. Tento standard se stále vyvíjí a má celou řadu variant. Dosahované teoretické přenosové rychlosti jsou řádově v jednotkách Mbit/s.

12.5. Zabezpečený přenos dat v bezdrátových sítích

NEZABEZPEČENÁ SÍŤ

Nezabezpečená síť neboli otevřený systém (open system) znamená, že se k ní může připojit kdokoliv. Uživatelé pošlou požadavek na AP, které vzápětí odpoví a přidělí IP adresu plus další potřebné nastavení. Většina těchto systémů pracuje s protokolem **DHCP** (Dynamic Host Configuration Protocol), který dovoluje přiřazení potřebných nastavení pro připojení do sítě jakémukoliv uživatelskému zařízení. Připojením k takovéto síti posíláme veškeré informace nešifrovaným způsobem. Tudíž kdokoliv, kdo má potřebný hardware a software, může odposlechnout vysílaná či přijímaná data. Je to totožné, jako byste někde v kavárně nechali několik volných ethernetových portů a kdokoliv kdo přijde, si může připojit kabel a tak se lehce dostane do vaší sítě nebo do internetu. Tato metoda zabezpečení sítě je vhodná, pokud chcete mít veřejně přístupný bod. Tudíž vhodné řešení na veřejná místa jakými jsou kavárny, letiště, nádraží, nákupní centra a jiná místa, kde chceme lidem dopřát jednoduché připojení k internetu.

Aby vůbec bylo možno se připojit na jakékoliv **AP**, existuje **SSID** (Service Set Identifier). V jednoduchosti si lze představit **AP** jakoby maják na pobřeží, který světlem oznamuje svou přítomnost. Tak i **AP** oznamuje svou přítomnost vysíláním jednoduchých **SSID** zpráv (tzv. beacon frame). Na základě znalosti **SSID** lze zaslat požadavek na spojení s **AP**. Některá **AP** mají možnost toto vysílání i případně vypnout. Díky vypnutému vysílání **SSID** lze zaručit jistou míru bezpečnosti, protože se standardním vybavením zařízení netuší, že je v blízkosti nějaký přístupový bod. Nesmíme však zapomenout, že když vysílaní **SSID** vypneme, tak se nejedná o žádné šifrování, pouze o utajení našeho zařízení



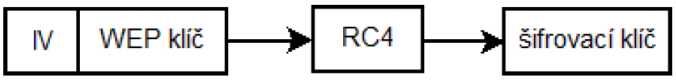
Obr. 12.4 Nezabezpečená **WiFi** síť

SÍŤ ZABEZPEČENA POMOCÍ WEP

Zabezpečení sítě za pomoci **WEP** klíčů je nejslabším možným řešením zabezpečení bezdrátové sítě. Avšak již tato metoda je jakýmsi ukazatelem, že se už nejedná o veřejnou síť, ale o privátní. Tudíž každý, kdo se k této síti nějakým způsobem připojí, nemůže říci, že to byla náhoda. Šifrování probíhá za pomoci proudové šifry **RC4**.

V počátku jsem uvedl, že se jedná o nejslabší řešení ohledně zabezpečení sítě. Není to dáno použitou šifrou, ale způsobem jakým je použita. Ve stručnosti šifra **RC4** má požadavek aby se inicializační klíč neopakoval. V implementaci **WEP** se však běžně stává, že se klíč opakuje po určité době respektive

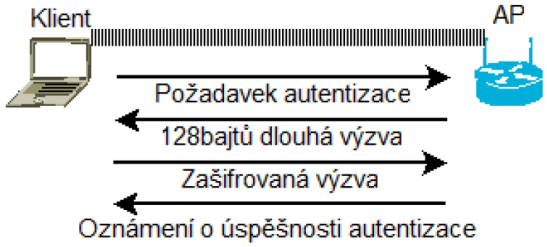
po určitém objemu dat. Ve zkratce ono zmíněné opakování má na svědomí inicializační vektor (IV), který má délku 24bitů a při vyšších rychlostech přenosu se tento prostor rychle vypotřebuje a musí se začít od začátku. Tudíž je porušeno pravidlo šifry **RC4**, že se klíč nesmí opakovat. Z toho plyne, že útočník, který odchytí velké množství dat, je schopen dešifrovat **WEP** klíč a tudíž i celou komunikaci.



Obr. 10.5 Princip šifrování WEP

Průběh autentizace sdíleným klíčem:

- Klient zašle požadavek o autentizaci na **AP**
- AP pošle klientovi výzvu 128bajtů dlouhou
- Klient tuto výzvu zašifruje svým 128bajtovým **WEP** klíčem a pošle zpět na **AP**
- AP ze znalosti **WEP** klíče určí, zda je zpráva zašifrována korektně
- AP oznámí klientovi úspěšnou či neúspěšnou autentizaci



Obr.12.6 Autentizace

SÍŤ ZABEZPEČENA POMOCÍ WPA

Toto zabezpečení vychází taktéž z šifry **RC4**, avšak ji používá jiným způsobem. Proto **WPA** řeší většinu nedostatků **WEP**. Pomoci mechanismů **TKIP**(Temporal Key Integrity Protocol), který se stará o šifrování dat a **802.1x** pro autentizaci uživatelů. **TKIP** odstranil IV a namísto něj mixuje vstup do **RC4** jinými hodnotami. Díky tomu zachovává požadavek šifry na to, aby se vstupní hodnoty klíče neopakovaly.

Vyřešené nedostatky technologie **WEP**:

- Opakování hodnoty IV
- Chybějící správa klíčů
- Podpora pokročilých autentizačních metod
- Chybějící identifikace a autentizace uživatelů
- a další…

SÍŤ ZABEZPEČENA POMOCÍ WPA2

Technologie **WPA2** již plně implementuje kompletní řešení **802.11i**, které bylo schváleno k použití v červnu 2004. **WPA2** opouští proudovou šifru **RC4** a přechází na blokovou šifru **AES** (Advanced Encryption Standard). Pro zabezpečení dat využívá následující komponenty. 802.1x pro autentizaci (s využitím **EAP** a autentizačního serveru), **RSN** (Robust Security Network) pro udržování záznamů

asociací a na **AES** založený **CCMP** (Counter Mode with Cipher Block Chaining Message Authentication Code Protocol) jenž se stará o šifrování dat.

Od roku 2006 musí každé zařízeni, které chce být certifikováno jako **Wi-Fi**, implementovat **WPA2.** Proto jej lze najít v dnešní době na běžných **AP** zařízeních, avšak Windows XP jej podporuje až se service pack 2 a se záplatou KB893357. [1]

VPN

VPN znamená virtuální privátní síť. Virtuální protože existuje jen logicky nikoli však fyzicky. Je to princip přenosu dat na již jiné vybudované síti. Privátní je protože využívá šifrovacích algoritmů a autentizačních mechanismu, tudíž ne každý se do ní dostane. A sítí je protože dokáže propojit několik uživatelů k serveru či několik VPN bran mezi sebou, základní druhy spojení VPN jsou naznačeny v následujícím odstavci.

Obecný přehled o VPN

Virtuální privátní sítě jsou využívány zřejmě nejčastěji pro propojení několika lokálních sítí přes veřejnou síť. Je to jednoduché a hlavně levné řešení propojení již přes vybudovanou infrastrukturu veřejné sítě. Mějme situaci, že firma má pobočku v Praze, Brně, Olomouci a Ostravě. Propojení těchto sítí ať už optikou, bezdrátově či nějakým metalickým médiem je náročné a hlavně nákladné. A taky proč natahovat nový spoj, když už někdo tento spoj vybudoval a za určitý poplatek jej nabízí k používání. V případě že máte připojení do nějaké takovéto sítě, kupříkladu internet, které propojuje ony zmíněné pobočky, pak není problém nakonfigurovat VPN brány tak, aby se těchto několik LAN sítí, chovalo jako by byly propojeny na jedné lokální síti. Ovšem rychlost je dána rychlostí připojení, jež Vám garantuje poskytovatel připojení a ne maximum rychlosti, které lze implementovat na použitém médiu.

METODY AUTENTIZACE

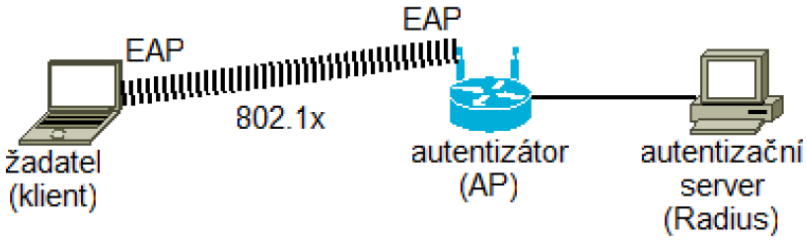
Autentizace je způsob ověření identity uživatele, respektive jeho zařízení, připojujícího se do systému. Toto ověření přístupu se provádí několika způsoby, kupříkladu klasické uživatelské jméno a heslo, nebo sofistikovanější metodou za pomocí certifikátů. V následujících několika odstavcích je výčet technologií používaných v bezdrátových sítích.

- **EAP**

Cílem proč vznikl EAP bylo vytvoření obecné platformy pro různé autentizační metody. **EAP** vychází z protokolu **PPP** (Point to Point Protocol). Jinými slovy by se dalo říci, že **EAP** je **PPP** se „zásuvnými“ moduly pro autentizaci. Díky tomu můžeme používat jakékoliv dostupné ověřovací metody například klasické uživatelské jméno a heslo či otisky prstů nebo otisk oční sítnice případně různé další dostupné metody identifikace uživatele.

- **802.1x**

802.1x je protokol umožňující využití protokolu **EAP** na metalické i bezdrátové síti. Tato technologie je založena na třech komponentách: žadatel, autentizátor a autentizační server. Ve stručnosti žadatelem je klient co žádá o přístup k síti. Autentizátor je obvykle mezičlánek mezi klientem a serverem, například AP. Autentizační server je kupříkladu server **RADIUS** či jiný **AAA** (authentication, authorization and accounting) server.



Obr.12.7 Autentikace pomocí 802.1x

- **MD5**

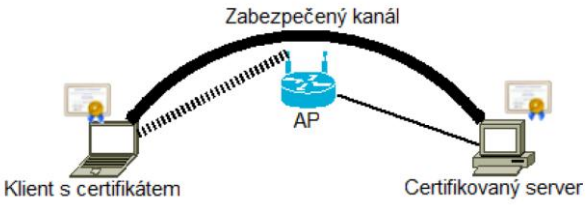
Jedná se o nejnižší možnou úroveň zabezpečení, jelikož hesla se ukládají na server v jednoduché textové podobě a v případě úspěšného útoku na takovýto server je celá autentizace k ničemu. Při tomto druhu autentizace se ověřuje pouze klient nikoli, však **AP**. Z toho vyplývá možné podvrhnutí cílového **AP**. V dnešní době již málo využívané řešení.

- **LEAP (Lightweight Extensible Authentication Protocol)**

Pochází od společnosti Cisco, která toto řešení představila v roce 2000. Bylo to dočasné řešení před příchodem **802.1x**. Umožňuje vzájemnou autentizaci a dynamickou obnovu **WEP** klíčů. Jedinou slabinou je fakt, že toto řešení implementovali pouze Cisco zařízení a tudíž se nerozšířilo i k jiným výrobcům.

- **TLS (Transport Layer Security)**

Představuje nejsilnější řešení z pohledu bezpečnosti, avšak za jistou cenu výpočetního výkonu. Jelikož **TLS** potřebuje pro autentizaci klienta a serveru certifikát na obou stranách. Při obousměrné autorizaci je také zapotřebí mít **PKI** (Public Key Infrastructure), neboli infrastrukturu veřejných klíčů. **PKI** zahrnuje jak správu certifikátů, klíčů tak i asymetrické šifrování a jiné potřebné utility. Díky **PKI** lze také využít technologie elektronického podpisu a autorizovat tak přenášená data.



Obr.12.8 Zabezpečená autentizace

Zmíněné certifikáty jsou standardizovány a lze je najít pod standardem **X.509**. Systémy založené na již zmíněné technologii **PKI** využívají právě certifikáty standardu **X.509**. Ověření certifikátu je de facto výpočet hodnot, které obsahují. Data se pro účel srozumitelnosti výpočetních strojů ještě kódují do binární formy a vkládají mezi uvozující značky, kterými jsou kupříkladu:
`-----BEGIN CERTIFICATE-----` pro začátek
`-----END CERTIFICATE-----` pro konec.

Ukázku kompletních certifikátu pro **TLS** spojení naleznete v příloze A. Tato příloha obsahuje ukázku serverového, uživatelského certifikátu a ukázku certifikační autority. Následující dvě další metody vycházejí z **TLS**, avšak již jen jeho prostřednictvím autentizují **AP**. Díky takovému to spojení následně použije jinou metodu **EAP** pro ověření klienta. Například metodu založenou na uživatelském jménu a heslu. Tyto technologie jsou **TTLS**(Tunnelled Transport Layer Security) a **PEAP** (protected **EAP**).

12.6. Měření – datové přenosy v bezdrátových sítích

Zadání:

- Přes konzolový port nakonfigurujte přístupový bod (Cisco Aironet 1240ag) na libovolný kanál s ESSID: PDxx (xx je číslo pracoviště).
- (pracujte ve dvojicích: 1 AP = 2 pracoviště =2 clienti)
- Připojte své pracoviště pomocí bezdrátové síťové karty do vaší sítě.
- Oscanujte okolí a zjistěte ESSID okolních sítí a pracovní kanály (+kmitočty).
- Pomocí příkazu **ping** zjistěte zpoždění paketů mezi vaším pracovištěm a přístupovým bodem (případně mezi dvěmi pracovišti). Při přenosu vyšlete vždy 10 dotazů najednou. Postupně volte 10 různých velikostí paketu, tak aby vhodně zastoupili celé spektrum možných velikostí MTU a zaznamenejte všechny dostupné výsledné informace.

Použité zařízení pro měření:

Server	PC11_N312 s OS Linux
Cisco AP	Aironet 1200/1240 series
WiFi karta	AirLive_WL-5480USB-50
Klient	PCxx_N312 s OS Linux

Vyhodnocení měření:

- Uveďte postup konfigurace AP a klienta.
- Zapište zjištěné ESSID okolních sítí a pracovní kanály (+kmitočty).
- Zhodnoťte zpoždění spojení uskutečněného přes bezdrátovou síť.
- Porovnejte toto zpoždění se zpožděním v metalických sítích.
- Zjistěte přenosový výkon pomocí stažení souboru z adresy http://moodle.kat440.vsb.cz

Shrnutí pojmů

WPAN sítě jsou využívány na krátké vzdálenosti, typicky do 10m.

WLAN sítě jsou využívány na krátké dosahy s vysokými přenosovými rychlostmi.

WMAN sítě jsou využívány na větší vzdálenosti, řádově desítky km.

GSM globální systém pro mobilní komunikaci.

GPRS rádiová datová služba s přepínáním paketů.

UMTS univerzální mobilní telekomunikační systém.

Otázky

1. Který typ bezdrátových sítí je nejvíce využíván?
2. Na čem závisí výsledná přenosová rychlost u bezdrátových přenosů?
3. Jaké bezdrátové technologie lze využít pro přístup do internetu?

Další zdroje

[1] *IEEE 802.11i* [online]. 2006 , 2. 7. 2008 v 20:40 [cit. 2009-02-20]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/IEEE_802.11i>.

[2] BARKEN, Lee. *Jak zabezpečit bezdrátovou síť Wi-FI*. Miroslav Hausknecht; Jiří Veselský. 28. dubna 48, 635 00 Brno : Computer Press, c2004. 174 s. ISBN 80-251-0346-3.

[3] *Transport Layer Security* [online]. 2007 , 6. 3. 2009 [cit. 2009-03-10]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Transport_Layer_Security>.

[4] *TripleDES* [online]. 2007 , 10. 4. 2009 [cit. 2009-04-12]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/TripleDES>.

[5] *OpenVPN* [online]. 2009 , 6. 2. 2009 [cit. 2009-02-10]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/OpenVPN>.

[6] *RSA* [online]. 2006 , 28. 3. 2009 [cit. 2009-04-02]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/RSA>.

13. ŠIFROVÁNÍ DAT

Čas ke studiu: 4 hodiny

Cíl Po prostudování tohoto odstavce budete umět

- definovat základní principy šifrování
- popsat jednotlivé typy symetrického a asymetrického šifrování
- definovat výhody a nevýhody jednotlivých typů šifrování

Výklad

13.1. Hash funkce

Algoritmy, které hrají velice důležitou roli v moderní kryptografii, jsou Hash-algoritmy. Otisk je **jednocestná funkce**, která z libovolně dlouhého textu vytvoří krátký řetězec o konstantní délce. Výsledný řetězec (otisk) musí maximálně charakterizovat původní text. Velikost výsledného textu je 16 B (např. algoritmus MD-5), 20 B (algoritmus SHA-1). Dnes se již algoritmy jako MD-5 a SHA-1 považují za slabé, proto se stále častěji setkáváme s novými algoritmy, které produkují delší a tím pádem bezpečnější otisky.

Mezi tyto algoritmy patři SHA-224 (otisk dlouhý 28 B), SHA-256 (otisk dlouhý 32 B), SHA-384 (otisk dlouhý 48 B) a SHA-256 někdy taky označovanou SHA-2 (otisk dlouhý 64 B).

Jak bylo výše uvedeno, hash je jednocestná funkce. To znamená, že z výsledné funkce nelze zpětně zjistit funkci původní. Jednocestné algoritmy jsou výpočetně nenáročné. Výpočetně náročné ale je nalézt k výsledku původní text.

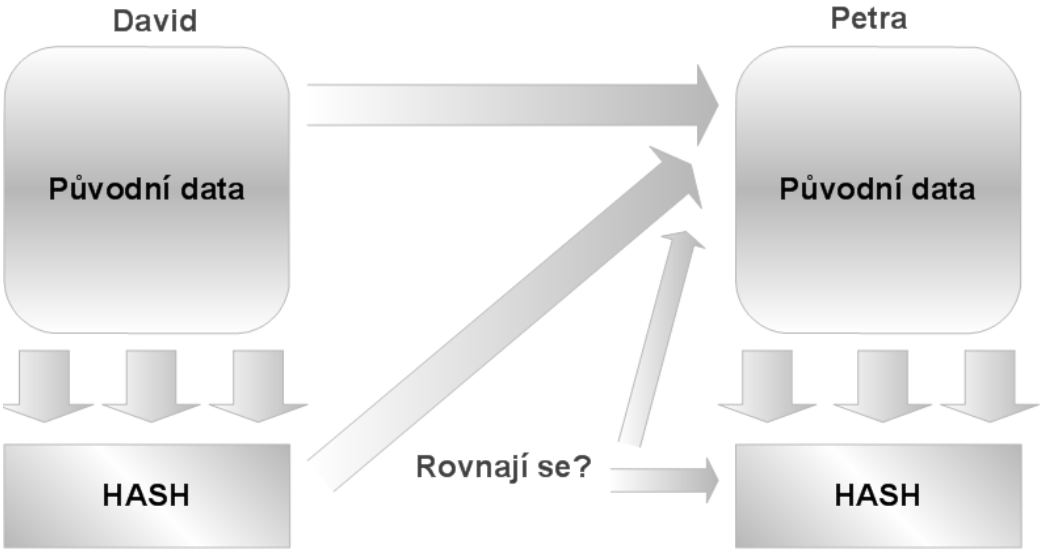
Kvalitní jednocestná funkce musí dát výrazně jiný výsledek i při drobné změně původního textu. Například u digitálního podpisu použitého k platebnímu příkazu by bylo velice špatné, kdyby se po připsání třeba jedné nuly k převáděné částce otisk nezměnil.

Otisk se počítá z libovolně dlouhého textu a to znamená, že by teoreticky mělo být ke konkrétnímu otisku možné najít nekonečně mnoho původních textů.

Hashovací funkce jsou konstruovány na výpočetních operacích nízké úrovně (bitové operace a posuny) a to znamená, že jsou výpočetně velmi rychlé a efektivní. **Tyto algoritmy nejsou v žádném případě šifrovací algoritmy, protože obecně neexistuje inverzní funkce.** Používají se v roli kvalitního „otisku prstu dat“ (fingerprint).

Otisk se může využít v komunikaci jako důkaz, že zpráva nebyla na cestě od jednoho uživatel k druhému změněna. To znamená, že využije otisk jako důkaz integrity dat.

Mějme uživatele Davida a uživatelku Petru. David Petře neodešle pouze samotná data zprávy, ale ke zprávě připojí i otisk z textu zprávy. Petra poté, co zprávu přijme, spočítá otisk z přijaté zprávy a ten srovná s přijatým otiskem od Davida. Pokud se oba otisky shodují, tak to znamená, že zpráva nebyla po cestě nijak změněna.



Obr. 13.1 Otisk jako důkaz integrity zprávy

Jelikož je ale algoritmus pro výpočet otisku veřejně popsán v příslušné technické normě, tak existuje hrozba, že se někdo pokusí zprávu od Davida zachytit, změnit ve svůj prospěch a odeslat Petře.

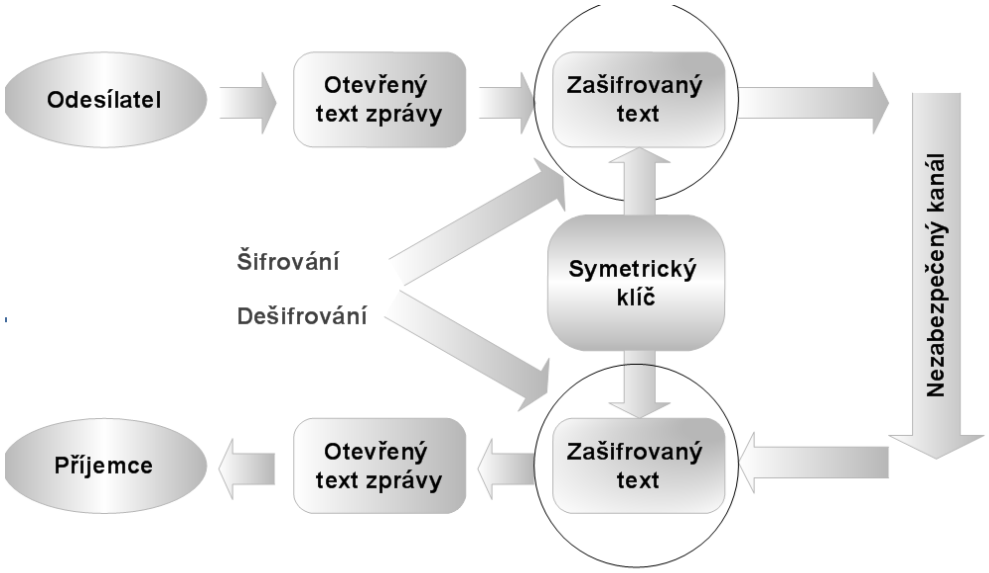
Tomuto zabrání David s Petrou tak, že si osobně vymění nějaké tajemství a vždy, když budou odesílat nějakou zprávu tomu druhému, přidají k ní i tohle sdílené tajemství. Třetí osoba tohle tajemství nezná, tudíž není schopna spočítat takovýto otisk a pozměnit zprávu, aniž by to někdo poznal.

Takovýto otisk se označuje MAC (Message Authentification Code – kryptografický kontrolní součet). Veliče často je MAC použit u protokolu SSL a protokolu IPSec.¨

13.2. Symetrické šifrování

Symetrické šifry se vyznačují tím, že pro šifrování a dešifrování se **používá stejný klíč.** Vstupem je nějaký tajný text a šifrovací klíč. Šifrovací funkce za pomocí klíče převede text na kód, který může být dále předán příjemci. Příjemce poté použije dešifrovací funkci se stejným klíčem, jakým byl původní text zašifrován a získá původní tajný text.

Je důležité, aby měl příjemce pro dešifrování stejný klíč jako ten, kterým byl původní text zašifrován. Bez tohoto klíče zprávu nikdo nerozluští. Při šifrování symetrickými šiframi je nutné, aby si komunikující osoby předaly nějakým bezpečným způsobem klíč.



Obr. 13.2: Symetrické šifrování dat

Pro symetrické šifrovací funkce musí platit, že při znalosti vstupního a zakódovaného textu je obtížné vygenerovat klíč. Při tom vlastní kódovaní a dekódování pomocí tohoto klíče je rychlá záležitost.

Obtížné zjištění klíče záleží na jeho délce. Šifrovaná zpráva by měla odolat takzvanému útoku hrubou silou, to znamená, že se předpokládá vyzkoušení všech možných klíčů. Například pokud jé klíč dlouhý 8 bitů, existuje 2^8 (256) možných klíčů.

Mezi nejběžnější šifrovací algoritmy patří:

- **DES** – délka klíče 56 bitů. Dnes nedostačující. V roce 1997 byl tento algoritmus prolomen.
- **3DES** – délka klíče je 112 bitů, nebo 168 bitů. Algoritmus je odvozen od algoritmu DES.
- **AES** – dnes doporučovaný. Délka klíče je 128, 192 nebo 256 bitů.

Symetrická kryptografie má jednu velkou nevýhodu a to v tom, že chtějí-li dva uživatelé mezi sebou tajně komunikovat, musí se napřed dohodnout na šifrovacím klíči. Tento ale musí být předán nějakou bezpečnou cestou (např. osobní kontakt), nesmí být předán po nezabezpečeném kanále.

Výhody a nevýhody symetrické kryptografie

Výhody :

- Patří mezi ně rychlost a nenáročnost na výpočetní výkon. Je velmi dobrá pro šifrování dat, které se nepotřebují nikam zasílat, nebo přenášet (zašifrování dokumentů na počítači).

Nevýhody

- Pro šifrovanou komunikaci je nutné předem si bezpečně vyměnit šifrovací klíč.
- Největší nevýhodou je počet klíčů. Pro komunikaci mezi více osobami je nutný velký počet klíčů. Pro komunikaci 2 osob je potřeba 1 klíč, pro 3 osoby jsou to 3 klíče a pro 4 osoby již 6 klíčů. Obecně se potřebný počet klíčů vyjádří podle vzorce
-

$$N = \frac{n^*(n-1)}{2} \tag{13.1}$$

kde *n* je počet klíčů.

13.3. Asymetrické šifrování

Asymetrická kryptografie (*public key cryptography*) je metoda, při které se pro šifrování a dešifrování zprávy používají jiné klíče. **Tyto dva klíče, které jsou spolu matematicky svázány, spolu tvoří pár (*keypair*).** Pro šifrování se tedy používá jeden z párů klíčů (příjemce zprávy jej ani nemusí znát) a pro dešifrování se použije druhý z páru. Výhodou u tohoto typu kryptografie je to, že ten kdo šifruje, nemusí s příjemce sdílet žádné tajemství. Tímto se odstraňuje riziko při výměně šifrovacího klíče jako u symetrické kryptografie.

Nyní si ukážeme blíže, jak tato metoda funguje. Nejprve příjemce (David) si vygeneruje pár klíčů, z nichž jeden je veřejný a je k dispozici všem. David jej veřejným (nezabezpečeným) kanálem předá Petře. Petra s pomocí tohoto veřejného klíče zašifruje zprávu, určenou Davidovi a pošle mu ji. David ji po přijetí dešifruje svým privátním klíčem, který si pečlivě uschoval. Celý proces je zobrazen na obrázku 13.3.

Tuto zprávu nemůže nikdo jiný než vlastník příslušeného soukromého klíče dešifrovat, protože nezbytnou podmínkou pro asymetrickou kryptografii, je praktická nemožnost ze znalosti šifrovacího klíče spočítat dešifrovací.

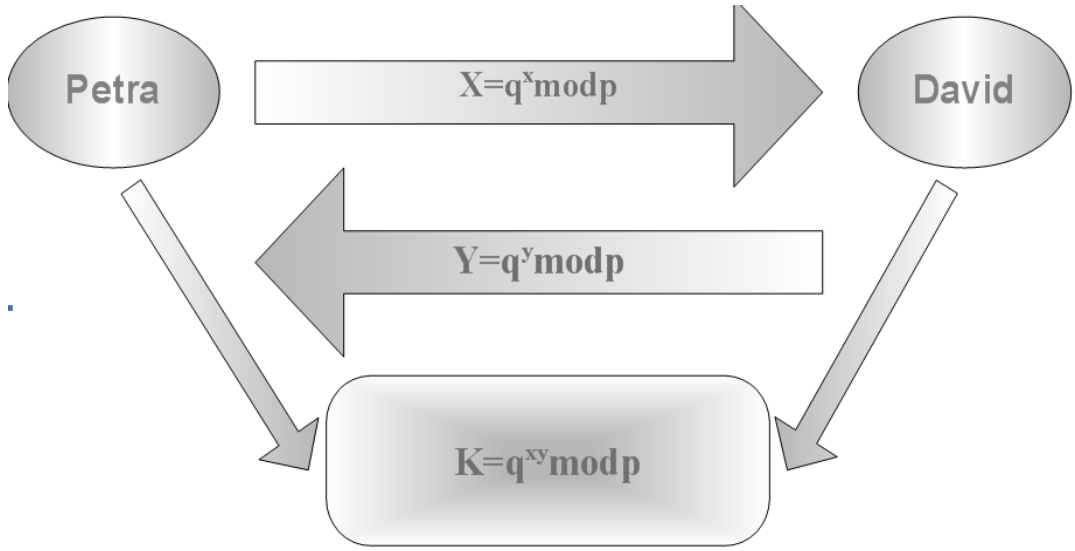
Obr. 13.3 Asymetrické šifrování

13.4. Diffie-Hellman protokol

Jednu z metod asymetrické kryptografie lze využít v symetrické kryptografii. Touto metodou je Diffie-Hellman protokol. Je to protokol pro bezpečnou výměnu šifrovacího klíče pro symetrickou kryptografii. Tento protokol navrhli pánové Whitfield Diffie a Martin Hellman. Postup výměny klíče je znázorněn na obrázku 13.4.

1) Petra a David se dohodnou na velkém prvočíslu **p** a **q**.
2) Petra si zvolí takové číslo **x**, že $1 \leq x < p-1$ a vypočítá $X = q^x mod\ p$, které pošle Davidovi.
3) David si zvolí libovolné takové číslo **y**, že $1 \leq y < p-1$ a vypočítá $Y = q^x mod\ p$, které pošle Petře.
4) Petra vypočítá $Y^x mod\ p$ a David vypočítá $X^y mod\ p$.
5) Petra a David nyní vlastní tajný šifrovací klíč **K** pro symetrickou kryptografii, protože $K = Y^x mod\ p = (q^y)^x mod\ p = q^{xy} mod\ p = (q^x)^y mod\ p = X^y mod\ p = K.$

Pokud by nějaká osoba odposlouchávala komunikaci mezi Petrou a Davidem, musela by pro získání **x,** ze znalosti **X,p,q,** nebo **y** ze znalosti **Y,p,q** umět vypočítat diskrétní algoritmus, to se ale předpokládá jako neřešitelný problém.



Obr. 13.4: Diffie-Hellman protokol

13.5. Algoritmus RSA

Algoritmus RSA vymysleli pánové Ron Rivest, Adi Shamir a Len Adleman v roce 1978. Zkratka RSA je z počátečních písmen jejich příjmení. Tento algoritmus je založen na problému faktorizace velkých čísel. Předpokládá se totiž, že rozložit velké číslo na součin prvočísel je velmi složitá úloha.

Postup vygenerování klíčů pomocí algoritmu RSA:

- David si zvolí dvě dostatečně velká prvočísla **p** a **q**, která mezi sebou vynásobí a získá číslo $n = p.q$.
- Potom začne počítat dvě celá čísla **d** a **e** tak, že si zvolí nějaké náhodné číslo **e**, které není soudělné s $(p-1).(q-1)$. Poté z výrazu $ed = 1 mod (p-1).(q-1)$ vypočítá číslo **d**.
- Potom pošle Petře veřejný klíč, který má podobu páru čísel $\{e,(n)\}$.
- Petra vytvoří zprávu $M \in Z_n$.
- Petra nyní zprávu zašifruje pomocí Davidova veřejného klíče $C = M^e mod\ n$.
- Tuto zprávu pak zašle Davidovi, který pomocí privátního klíče tuto zprávu dešifruje jako $M = C^d mod\ n$ a potom ji převede zpět do textové podoby.

13.6. Praktické nástroje pro šifrování dat v OS linux

ccrypt

Nástroj pro šifrování a dešifrování souborů a toků. Využívá AES.

`ccrypt -e pd.txt`	požádá o encryption key a provede šifrování souboru pd.txt do souboru pd.txt.cpt
`ccrypt -d pd.txt.cpt`	požádá o encryption key a provede dešifrování souboru

md5sum

`md5sum`	vypočítá a ověří MD5 hash
`md5sum pd.txt`	vypočítá MD5 hash
`md5sum pd.txt>pd.md5`	vypočítá MD5 hash a výsledek uloží do souboru pd.md5
`md5sum -c pd.md5`	ověří MD5 hash

sha1sum

`sha1sum`	vypočítá a ověří SHA1 hash
`sha1sum pd.txt`	vypočítá SHA1 hash
`sha1sum pd.txt>pd.sha1`	vypočítá SHA1 hash a výsledek uloží do souboru pd.md5
`sha1sum -c pd.sha1`	ověří SHA1 hash

steghide

steghide steganography program pro formáty JPEG, BMP, WAV, AU

`steghide embed -cf obrazek.jpg -ef secret.txt` uloží text do obrazek.jpg

`steghide info obrazek.jpg`

`steghide extract -sf obrazek.jpg -xf secret.txt` uloží text do souboru secret.txt

ikey3000 instalace a konfigurace

Musí být nainstalovány následující balíčky:

lsusb, openct, openec, pcscd, pcsc-tools, openssl

`openct-tool list`	vypíše název připojeného zařízení
`openct-tool -r 0 atr`	výpis ATR (answer to reset) adresy klíče
`ATR_analysis xxxx`	provede kompletní analýzu klíče. Za xxxx doplňte hodnotu ATR zjištěnou předchozím příkazem

Shrnutí pojmů

Hash je jednosměrná funkce, která vytvoří z libovolného textu krátký řetězec konstantní délky.

Symetrické šifrování používá stejný klíč pro šifrování i dešifrování.

Asymetrické šifrování používá jiný klíč pro šifrování a jiný klíč pro dešifrování.

RSA asymetrická šifra, využívající faktorizaci velkých čísel

Otázky

1. Jaké znáte nástroje pro generování otisku zprávy?
2. Jakým způsobem se tvoří digitální podpis?
3. Proč je asymetrické šifrování časově náročné?

Další zdroje

[1] – LIBERDA, L. *Problematika digitálního podpisu pod Linuxem.* Ostrava: Vysoká škola báňská – Technická univerzita Ostrava. Fakulta elektrotechniky a informatiky. Katedra telekomunikační techniky, 2007. 51 s.

[2] – *Kryptografie* [online]. 20.12.2006 [cit.2009-01-06]. Dostupný z WWW: http://cs.wikipedie.org/wiki/Kryptografie.

[3] - DOSTÁLEK, L., VOHNOUTOVÁ M. *Velký průvodce infrastrukturou PKI a technologií elektronického podpisu.* 1 vyd. Brno: Computer Press, 2006, 536 str. ISBN 80-251-0828-7.

[4] – DOSTÁLEK, L. a kolektiv. *Velký průvodce protokoly TCP/IP:Bezpečnost*. 2 vyd. Brno: Computer Press, 2003, 592 str. ISBN 80-7226-849-X.

[5] – DOLEŽAL, Dušan. *Jak si vybrat certifikační autoritu* [online]. 14.3.2003 [cit. 2009-1-12]. Dostupný z WWW: http://interval.cz/clanky/jak-si-vybrat-certifikacni-autoritu/

[6] – TOXEN, Bob. *Bezpečnost v Linuxu:Prevence a odvrácení napadení systému*. Brno: Computer Press, 2003, 876 str. ISBN 80-7226-716-7

[7] – HONTANÓN, Ramón. *Linux – praktická bezpečnost*. Praha: Grada, 2003, 440 str. ISBN 80-247-0652-0.

14. ZÁKLADY OPERAČNÍHO SYSTÉMU LINUX

Čas ke studiu: 3 hodiny

Cíl Po prostudování tohoto odstavce budete umět

- definovat základní vlastnosti operačního systému
- používat základní příkazy pro práci v Linuxu
- samostatně pracovat s open source nástroji

Výklad

Protože se na cvičeních pracuje s operačním systémem Linux, jsou zde uvedeny některé základní vlastnosti tohoto OS.

Linux představuje reálnou alternativu proti monopolnímu postavení firmy Microsoft na poli operačních serverových systémů a to zejména z důvodu pořizovacích nákladů. Ty jsou totiž velice nízké. Samotný Linux je šířen pod licencí GPL GNU, což znamená, že je zcela zdarma. Navíc jeho hardwarové nároky se v dnešní době dají považovat za minimální. Uživatel není nucen pořizovat drahý značkový server, ale své služby může nabízet na „obyčejném“ PC při stejné kvalitě výkonu, spolehlivosti a dostupnosti jako u komerčních řešení. Základní nedůvěra k OS Linux pramení z toho, že „když je něco zadarmo, jak to může být kvalitní?“. Linux je vyvíjen tisíci odborníky a nadšenci po celém světě a mnohokrát vyšším počtem testerů.

Linux je schopen běžet roky bez jediného restartu (nepotřebuje reboot ani v tak extrémních případech jako je výměna ovladačů, či instalace nového software). Díky tomu, že je systémem typu unix, který byl vyvíjen na akademické půdě, přesně podle pravidel, podle kterých se systémy vyvíjet mají je absolutně bezpečný (neexistují zde prakticky žádné viry (občas sice prosákne informace o linuxovém viru, ty však díky dokonalé ochraně jádra nemají možnost se dále šířit ani měnit jakékoli soubory apod.), spolehlivý a stabilní (díky dokonalé ochraně paměti systému).

Flexibilita – formou „stavebnice“ je možno sestavit systém šitý na míru konkrétnímu použití a nainstalovat jen to, co je nezbytně nutné (s možností dalšího rozšíření), což opět zvyšuje výkon a bezpečnost. Díky rozšiřitelnosti a to nejen o cizí komponenty, ale i komponenty naprogramované vlastními silami je Linux ideálním řešením pro složité a nestandardní použití.

13.1. Základní vlastnosti operačního systému

Linux je svobodný
Svobodný software je software, který zaručuje uživatelům svobody, které jim proprietární software upírá.

- Svoboda používat program tak, jak vy sami chcete. Žádné registrace, aktivace, vázanost licence na hardware, omezení počtu instalací či zálohování, omezení na nekomerční aktivity, apod.
- Svoboda studovat, jak program funguje, a upravovat jej. K tomu je vázán nárok na získání zdrojového kódu. Kromě možnosti cokoliv změnit a nezávislosti na dodavateli má tato

svoboda řadu praktických důsledků. Žádný spyware či malware, žádné odesílání informací z vašeho počítače někam do internetu, žádné skryté funkce, žádná zadní vrátka. Zdrojový kód je pod neustálým dohledem veřejnosti.

- Poslední dvě svobody se týkají distribuce. Máte svobodu program šířit jak v původní, tak upravené verzi. Svobodný software tedy můžete kopírovat kamarádovi nebo pozměnit a změny zveřejnit (nebo prodávat). Tahle svoboda umožňuje spolupráci vývojářů i uživatelů z celého světa a zajišťuje, že kontrolu nad softwarem mají jeho uživatelé.

Stabilita a spolehlivost
Stabilita Linuxu je věc dobře známá. Nepromítá se však jenom jako odolnost systému vůči pádům, ale i jinde. Kupříkladu, Linux nevyžaduje žádné restarty kvůli instalacím či aktualizacím aplikací, a také se nezanáší smetím, a nevyžaduje tedy pravidelné reinstalace kvůli souvisejícímu zpomalení. K tomu v Linuxu nedochází.
Zatímco na trhu s osobními počítači je podíl Linuxu malý, pohybující se mezi 0,5 % až 3 %, mezi superpočítači naopak Linux výrazně převažuje. Na českých webových serverech se jeho podíl pohybuje okolo 70 %.

Když už zmiňujeme výhody GNU/Linuxu, tak musíme také zmínit jeho stinné stránky. Hlavním problémem je jeho malá rozšířenost na desktopech (resp. nadvláda dominantního systému), která ovlivňuje celou řadu faktorů působících na uživatele.

Hardware, software
Ne každý výrobce hardwaru myslí na podporu Linuxu, někteří Linux ignorují, dokonce se někdy i brání linuxovým vývojářům v získání potřebných informací, aby mohli ovladač napsat sami. Může se tedy stát, že některý váš hardware Linux nebude schopen zprovoznit. Většina hardwaru však v Linuxu podporována je.
V případě softwaru je situace podobná. Výrobci mnohdy Linux ignorují, což se dotýká zejména dostupnosti "profesionálního" (=velmi drahého) komerčního softwaru. Jsou sice různá řešení, jak spouštět software určený pro Windows v Linuxu (Wine, virtualizace), popřípadě je možné provozovat Windows a Linux současně (dualboot), ale to uživatelům ne vždy vyhovuje. Na stranu druhou, pro GNU/Linux je k dispozici ohromné množství svobodných aplikací.
Co se podpory Linuxu ze strany výrobců SW a HW týče, toto je záležitost, kterou mohou změnit pouze uživatelé, tedy třeba právě vy. Je nutné apelovat na výrobce, dát jim vědět, že máte o podporu příslušného produktu v GNU/Linuxu zájem.

13.2. Základní příkazy v OS Linux

Proces - program, který se začal provádět

- dentifikace procesu: každý proces má své identifikační číslo (**PID** – Process IDentification)
- výpis provedených procesů: ps (ve výpisu jsou zobrazeny 4 slupce)
  PID – identifikační číslo procesu
  TTY – terminál
  TIME – čas nutný pro provedení procesu
  CMD – příkaz (název procesu)
- výpis všech procesů: ps –A
- top (table of process) zobrazuje i procentuelní využití kapacity paměti a procesoru jednotlivými procesy (pro ukončení provádění příkazu použijte **q**)
- ukončení procesu: kill <PID> (pouze korektní ukončení)
  kill -9 <PID> (definitivní (i nekorektní) ukončení)

Řešený příklad

- *Proveďte příkaz* ps –A *a zaznamenejte si proces prováděný na tty2.*
- *Přihlaste se na druhé konzole (Alt-F2 a zadat jméno uživatele a heslo).*
- *Přepněte se zpět na první konzolu (Alt-F1).*
- *Proveďte příkaz* ps –A *a zaznamenejte si proces prováděný na tty2 a jeho PID po přihlášení by na tty2 měl být prováděn proces bourne again shell (bash)).*
- *Pokuste se ukončit proces prováděný na tty2 příkazem* kill <PID>.
- *Proveďte příkaz* ps –A *a pokud je na tty2 prováděn stále proces bash, pokuste se ho končit příkazem* kill -9 <PID>*, opět ověřte příkazem* ps –A.
- *Správnost ukončení procesu ověřte i pokusným přepnutím na druhou konzolu(Alt-F2).*

Nápověda

- **zabudovaná** (je součástí operačního systému)

(pozn.: tato nápověda je charakteristická proměnlivou kvalitou odvislou od schopností autora daného programu, pro nějž autor sám nápovědu sestavuje).

1. manuálové stránky: vyvolávají se příkazem man <příkaz> (např.: man ps), pozn.: pro ukončení provádění příkazu použijte klávesu **q**
2. informační stránky: vyvolávají se příkazem info <příkaz> (např.: info ps),
3. dokumentace k programu uložená v adresáři.

- **internet**

internetové stránky (např.: www.linux.cz, www.linux.links.cz, www.root.cz),
HOWTO soubory s nápovědou,které je možno stáhnout na internetu (txt,pdf,html...).

- **knihy**

(např.: Jemný úvod do systému UNIX (Kopp, 2001), Rozumíme UNIXU (CPRESS, 2002))

- **lidé-experti** (diskusní skupiny, konference mailové, chatové atd.)

Soubory

Soubor je jakýkoliv zdroj,ze kterého lze data číst nebo kam lze data zapisovat.
- **obyčejný soubor** (např.: textový soubor (ascii 32-126), binární (program,všechny znaky))
- **adresář** (soubor, ve kterém jsou uloženy informace o souborech: velikost, datum, ..atd.)
(Pozn.: na této úrovni lze zabezpečit systém, např. lze i omezit velikost prostoru na disku pro jednotlivé uživatele)
- **speciální soubor** (soubor, v němž jsou uloženy informace o vstupních a výstupních zařízeních, např. klávesnice, monitor, tiskárna, HDD, FDD ... atd.)

Příkaz pro výpis obsahu adresáře a informací o souborech:
ls název_adresáře (výpis)
ls –l název_adresáře (rozšířený výpis) (Pozn.: l – long)
ls –al název_adresáře (rozšířený výpis všech souborů) (Pozn.: a – all)
Označení typů souborů:

.	sám adresář
..	rodičovský adresář
.profile	skrytý soubor
bin/	podadresář
setup*	spustitelný soubor
soubor.txt	soubor

Popis souboru

Popis jednotlivých položek výpisu pomocí příkazu ls:
total (nebo **celkem**) **10** velikost paměti v blocích, blok = 512 bytů
drwxr-xr-x 2 student student 96 Lis 21 9:34 soubor.txt

Význam sloupců:
1. sloupec - obsahuje 10 znaků charakterizujících daný soubor
-1.znak: - obyčejný soubor
d adresář
c speciální soubor pro znaková zařízení (c – character)
b speciální soubor pro bloková zařízení (b – block)
l speciální soubor s odkazy (l – linky)
jiný jiný speciální soubor
- 2. až 10 znak: r - čtení (read), w – zápis (write), x - spouštění
rwx (2. až 4. znak) **rwx** (5. až 7. znak) **rwx** (8. až 10. znak)
práva vlastníka souboru práva skupiny do níž patří vlastník práva ostatních uživatelů

2. sloupec - je číselným označením počtu odkazů na daný soubor
3. sloupec - charakterizuje vlastníka souboru
4. sloupec - charakterizuje skupinu uživatelů, do níž patří vlastník souboru
5. sloupec - označuje velikost souboru
6. až 8. sloupec - datum vytvoření nebo aktualizace souboru
9. sloupec - název souboru

(Pozn.: Název souboru je obvykle do 255 znaků a neměl by obsahovat znaky * / \ ? ani interpunkci, naopak běžně se používá _ - .)

Práce s adresáři

vytvoření (Make Directory) mkdir název_adresáře
odstranění (Remove Directory) rmdir název_adresáře
přejmenování (Move) mv staré_jméno_adresáře nové_jméno_adresáře
kopírování (Copy) cp -R zdrojový_adresář cílový_adresář
změna adresáře (Change Directory)
(relativní či absolutní cesta k adresáři) cd název_adresáře
(návrat do domovského adresáře) cd
(návrat o úroveň zpět) cd ..
(návrat do kořenového adresáře /) cd /
výpis absolutní cesty k pracovnímu adresáři (Print Working Directory) pwd

(Pozn.: Pro doplňování názvů adresářů a souborů lze také používat tlačítko „Tabelátor“.)
(Pozn.: Více příkazů lze zapsat na jeden řádek pomocí znaku **;**, např.: **cd;mkdir pokus**.)

Způsob zadávání cesty

lze používat dva způsoby zadávání cesty k danému souboru:
- absolutní cesta
Je zadána od nejvyššího vrchu / („kořenového adresáře“), zde se nacházejí nedůležitější adresáře, jako jsou: **/bin** (obsahující programy),
/dev (obsahující všechna zařízení),
/etc (obsahující konfiguraci),

/home (obsahující domovské adresáře).
např.: výpis adresářů pomocí absolutní cesty **ls /bin** nebo **ls /home/student**
(Pozn.: Domovské adresáře mohou být fyzicky umístěny na jiném disku či počítači, často jsou pro dané domovské adresáře omezovány počty v nich uložených souborů či maximální obsažený prostor.)

- relativní cesta
Je zadána od pracovního adresáře (adresáře ve kterém se právě nacházíme)
např.: výpis adresáře zadaného pomocí absolutní cesty **ls /home/student/pokus** lze provést také pomocí relativní cesty **ls pokus**, ale pouze za předpokladu, že pracovním adresářem je **/home/student**.
(Pozn.: Pro předchozí případ můžeme využít také toho, že systém „zná“ cestu k domovskému adresáři a lze ji reprezentovat znakem **~**, pomocí něhož pak můžeme provést výpis prováděný příkazem **ls /home/student/pokus** také jednodušeji **ls ~pokus** nehledě na to, jaký je aktuální pracovní adresář!)

Práva přístupu k adresáři

Jsou dána právy přístupu souboru typu adresář charakterizující daný adresář:
rwx ... vše povoleno
r-x ... není povoleno mazání adresáře (pouze výpis obsahu adresáře a přístup k souborům)
--x ... není povolen ani výpis obsahu adresáře (např. pomocí příkazu ls) a je možno pouze přistupovat k některým souborům

Práce se soubory

- příkaz touch název_souboru (vytvoří nový soubor, pokud už existuje, tak aktualizuje datum a zachová obsah)
- příkaz cat název_souboru (slouží pro čtení, vytvoření nebo spojování souborů)

cat soubor1 - vypíše obsah souboru soubor1,
cat > soubor1 - vytvoří nový soubor soubor1 a umožní uživateli zadávat znaky, které budou do souboru uloženy, pro ukončení použijte ***Ctrl-D***.
(Pozn.: pokud už soubor zadaného jména existuje, tak je jeho obsah přemazán nově zadanými znaky.)
cat >> soubor1 - umožní vkládat data za data, která již byla v souboru uložena (nemaže je), ale zobrazuje pouze nově vkládaná data,
cat soubor1 soubor2 > soubor3 – sloučí obsah souborů soubor1 a soubor2 a uloží výsledek do souboru soubor3.
(Pozn.:Podobně lze využít i znaku *,který nám slouží pro označení všech souborů v daném adresáři, např.: cat *, nebo cat * > výsledný_soubor).

Řešený příklad

1 *Vytvořte nový adresář váš_název ve vašem domovském adresáři* mkdir váš_název.
2 *Vytvořte v adresáři váš_název tři nové soubory s libovolným obsahem* cat > nový_soubor.
3 *Připište do některých souborů nová data (při zachování dat původních)* cat >> nový_soubor.
4 *Slučte obsah všech souborů obsažených v adresáři **/home/student/váš_název** do souboru výsledný_soubor v témže adresáři* cat home/student/váš_název/* > výsledný_soubor nebo cat ~/váš_název/* > výsledný_soubor.

Práva přístupu k souboru

Tato práva, jak již bylo řečeno, lze zobrazit rozšířeným výpisem pomocí příkazu ls –l. Nově vytvořeným souborům jsou automaticky přidělena práva **rw- rw- r--.** Pro změnu těchto práv lze použít příkaz chmod xyz název_souboru

chmod	**x**			**y**			**z**		
práva	**r**	**w**	**x**	**r**	**w**	**x**	**r**	**w**	**x**
Příklad									
práva	r	w	-	r	w	-	r	-	-
binárně	1	1	0	1	1	0	1	0	0
dekadicky		6			6			4	
výsledek	chmod 664								

Základní operace se soubory

odstranění (výmaz) souboru (Remove) rm název_souboru
přesun nebo přejmenování souboru (Move) mv zdroj cíl
kopírování souboru (Copy) cp zdrojový_soubor cílový_soubor

Řešený příklad

1 *Přejmenujte jeden ze souborů, které jste vytvořili* mv starý_název nový_název.
2 *Zkopírujte jeden ze souborů, které jste vytvořili* cp zdrojový_soubor cílový_soubor.
3 *Změňte práva u jednoho z vámi vytvořených souborů z **rw- rw- r--** na **r-- r-- r--*** chmod 444 název_souboru.
4 *Pokuste se tento soubor jako vlastník smazat* rm název_souboru.
5 *Lze ho smazat? (Ano ale na dotaz!)*

Práce s textovými soubory

určení typu souboru file název_souboru
počítání znaků wc –c název_souboru (Word Count – Character)
počítání slov wc –w název_souboru (Word Count – Word)
počítání řádků wc –l název_souboru (Word Count – Line)
očíslování řádků souboru nl < starý_soubor > nový_soubor
výpis obsahu souboru more název_souboru
less název_souboru

(Pozn.: pro pohyb použijte šipky nebo „**mezerník**“ a pro **ukončení q**, pokud se neukončí sami.)

Připojení výměnného zařízení

Při práci s *Unixem* se nevyhneme nutnosti přenést data z/do PC pomocí výměnných zařízení jakou jsou *disketa*, *CD* nebo *USB Flash Disk*. K připojení těchto zařízení slouží příkaz mount typ_zařízení kam. (**kam** bývá prázdný předem vytvořený adresář)

Typy zařízení:	místo:
Disketa	**/dev/floppy** **/mnt/floppy**
CD	**/dev/cdrom** **/mnt/cdrom**
USB	**/dev/sda1** (může se lišit)

- **Připojení CD/DVD**

Připojit CD nebo DVD může pouze lokálně přihlášený uživatel. Jste-li lokálně přihlášeni, vložte disk do mechaniky a spusťte:

mount /dev/cdrom /mnt/cdrom

Aby bylo možné CD nebo DVD vyjmout z mechaniky, musíte ukončit všechny procesy pracující s obsahem adresáře **/mnt/cdrom** a spustit

umount /mnt/cdrom

- **Připojení USB Flash Disku**

USB Flash disky (klíčenky) používají protokol *USB mass storage*, který je podporován od jádra verze 2.4, tedy prakticky ve všech novějších linuxových distribucích. Flash disk je možné připojit přímo do adresářové struktury systému jako CD nebo pevný disk. Ovladač využívá SCSI emulace a disk se objevuje jako SCSI zařízení.

Jako které zařízení je disk připojen zjistíte z výpisu příkazu lsusb - seznam všech zařízení na sběrnici USB, s parametrem **-v** velmi podrobný.

Připojit ho můžete jako superuživatel příkazem **mount**. Adresář do něhož bude disk připojen musí být vytvořen předem.

mount /dev/sda1 /mnt/usb

Po skončení práce s diskem je nutné ho před vytažením z počítače odpojit ze systému, jinak je možné, že nebudou zapsána všechna data. To provedete příkazem **umount.**

umount /mnt/flash

Užitečným příkazem je tail –f /var/log/messages který vypisuje aktuálně prováděné akce systému. Spusťte tento příkaz před připojením USB disku do USB postu a sledujte co se bude dít. Z výpisu můžete vysledovat, jak se bude USB disk jevit v systému.

Řešený příklad

1. *Vytvořte adresář v **/home/student/váš_adresář*** mkdir váš_adresář.
2. *Připojte USB Flash Disk do váš adresář* mount /dev/sda1 /home/student/váš_adresář.
3. *Překopírujte soubor z **/home/student** na vaše USB* cp co kam.
4. *Regulerně odpojte USB* umouth /home/student/váš_nazev.

Práce se síťovou kartou

Základním příkazem pro práci se síťovou kartou je příkaz ifconfig. Tento příkaz vypíše všechny dostupné informace o sítových kartách v systému (ifconfig -a). Pokud chcete nastavit IP adresu na příslušný síťový adaptér, proveďte to příkazem: ifconfig eth*n xxx.xxx.xxx.xxx* netmask *yyy.yyy.yyy.yyy*

kde: **n** je číslo síťové karty

xxx.xxx.xxx.xxx je IP adresa

yyy.yyy.yyy.yyy je maska

Pokud chcete aby byla IP adresa udělena prostřednictvím DHCP serveru, použijte příkaz dhclient ethn (kde **n** je číslo síťové karty). Pokud chcete IP adresu uvolnit: dhclient eth*n* release.

route –n zobrazí výpis směrovací tabulky. Pro nastavení výchozí brány použijte příkaz: route add default gw xxx.xxx.xxx.xxx

kde **xxx.xxx.xxx.xxx** je IP adresa výchozí brány.

Vypnutí počítače

poweroff	Ukončí všechny procesy, nezapsaná data zapíše na disk, vypne počítač.	
halt	Totéž jako poweroff.	
shutdown -h now	Také vypne počítač a na první pohled se může zdát, že je příliš složitý. Umí toho ale víc. Provede příslušné ukončovací scripty a má i řadu užitečných voleb.	
Například:	shutdown -h 22:00	znamená "vypni se v deset večer",
	shutdown -r +30	za půl hodiny rebootuj počítač,
	shutdown -r now	rebootuj hned teď.

Shrnutí pojmů

Linux je svobodný operační systém.

Proces je program, který se začal provádět.

Soubor je jakýkoliv zdroj, ze kterého lze data číst nebo kam lze data zapisovat.

Přístupová práva definují práva k jednotlivým souborům.

Otázky

4. Jaké znáte textové editory pro práci s textem?
5. Jakým způsobem lze nastavit práva k jednotlivým souborům v podadresáři?
6. Kterým příkazem si můžeme vyžádat informace o běžících procesech?

Další zdroje

[1] – Petrlík, L. *Jemný úvod do systému UNIX.* KOPP 2001, ISBN 80-85828-28-6
[2] – Vychodil, V. *Operační systém Linux, Příručka českého uživatele.* Computer Press 2003, ISBN 80-7226-333-1
[3] - http://www.root.cz

CD-ROM

Na přiloženém CD jsou k dispozici následující animace:

1. Parita
2. Spirálová parita
3. CRC
4. Hammingův kód
5. Reed-Müllerův kód
6. Adaptivní Huffmanův kód

Rejstřík

N

O

P

Q

R

S

T

U

V

W

X

Y

Z