VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

DATOVÁ KOMUNIKACE

Úvod do teorie informace a kódování

Prof. Ing. Dalibor Biolek, CSc.

ÚSTAV TELEKOMUNIKACÍ

PŘEDMLUVA

Skripta pokrývají první cyklus přednášek předmětu „Datová komunikace“, věnovaný základním poznatkům z teorie informace a úvodu do zdrojového a kanálového kódování. Skriptum je třeba chápat jako rozšiřující text k základní literatuře [1]. K lepšímu pochopení teoretických principů je výklad prokládán řadou řešených příkladů. Velkou inspirací při psaní tohoto učebního textu byla čtivá skripta [4], která studentům doporučuji k studiu dalších technik kódování.

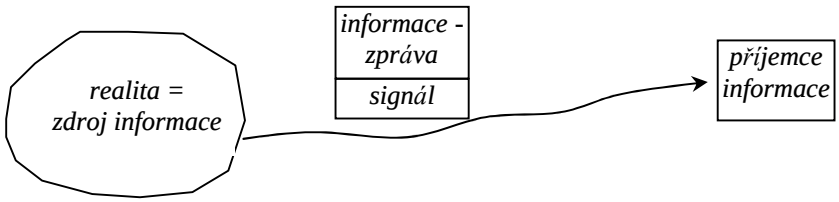
V Brně dne 22.9.2002 **Dalibor Biolek**

DATOVÁ KOMUNIKACE - Úvod do teorie informace a kódování.

OBSAH

ZÁKLADNÍ POZNATKY Z TEORIE INFORMACE

Informace, zpráva, signál.
Získávání informací je proces probíhající mezi **příjemcem informace** (člověkem, případně strojem) a **zdrojem informace (**okolní realitou). Je-li konečným příjemcem člověk, pak má informace obvykle charakter **zprávy** (tj. sdělení v určité formě). Zpráva se šíří daným prostředím díky nosiči – **signálu**.



Obr. 1. Zjednodušené znázornění procesu přenosu informace od zdroje k příjemci.

s Příklad 1: Přenos informací kouřovými signály:

Souvislý tenký kouř:	ahoj – chci Ti něco říct.
Přerušovaný tenký kouř:	ulovili jsme mamuta.
Souvislý čadivý kouř:	snědli jsme mamuta.

V 1. sloupci je uveden signál, který představuje pro příjemce zprávu (informaci) uvedenou v 2. sloupci.
Signál může příjemce interpretovat různě – může vnímat barvu kouře, jeho „vlnění“ na horizontu atd. Pro přenos informace jsou však důležité jen některé atributy: zda je kouř tenký nebo čadivý, souvislý nebo přerušovaný. Myšlenkově tedy může příjemce původní signál nahradit jiným beze změny informačního obsahu přenášené zprávy, např.:

———————	ahoj – chci Ti něco říct.
- - - - - - - - - - - - - - - - - - - -	ulovili jsme mamuta.
▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬	snědli jsme mamuta.

þ

Z hlediska vysílání, přenosu, příjmu a uchovávání zpráv lze obecně definovat **signál** jako fyzikální veličinu(y), v jejíž některých parametrech je zakódována daná zpráva.
Z uvedeného vyplývá, že obvykle je potřeba na straně příjemce provést závěrečnou konverzi přijatého signálu na přijatou zprávu (informaci). To je úkol koncového zařízení (TV, reproduktor, dekodér), v případě kouřového signálu dochází ke konverzi přímo „v hlavě“ příjemce.
Z hlediska informace „nanesené“ na signálu není důležité, jak vlastní signál „vypadá“, z hlediska použitého prostředí šíření signálu však ano. Toho se běžně využívá při přenosu dat, kdy na trase od zdroje k příjemci signál často střídá řadu forem (analogový, digitální, elmag. vlna, akustická vlna, laserový paprsek...). Je však nutné zařídit, aby při konverzi jedné formy signálu v jinou zůstala zachována původní přenášená informace.

Sémantický a syntaktický význam informace [4], [9].
Informace je chápána ve dvojím významu: sémantickém (co zpráva říká, jaký jej její obsah) a syntaktickém (jaká je forma zprávy).

s Příklad 2: zpráva ULOVILI JSME MAMUTA – sémantické pojetí: příjemce spěchá na oběd, syntaktické pojetí: příjemce konstatuje, že vidí na obzoru tenký přerušovaný kouř.

þ

Teorie informace se zabývá pouze syntaktickým pojetím a nezajímá ji sémantika.

Základní tématické oblasti teorie informace.
Popis a řešení problémů spojených se sběrem, přenosem, zpracováním a archivací informace v syntaktickém smyslu.
Součástí teorie informace je teorie kódování.
Teorie informace je součástí kybernetiky.
Pozn.: V současné době jsme svědky globalizace – propojování informačních technologií, komunikačních technologií, poznatků z teorie zpracování signálů a teorie obvodů a systémů do jednoho celku.

Kvantifikace informace. Realita jako množina stavů [9].
Jak popsat množství informace ve zprávě? Paradox: Musíme se odtrhnout od smyslu zprávy (sémantiky). Pak zbude jen forma, kterou lze popsat jako posloupnost dovolených stavů. Vysvětlení následuje.
Zpráva je zakódována v určitých parametrech signálu.

s Příklad 3: U primitivního kouřového signálu z Př. 1 těmito parametry byly: tloušťka kouře T (tenký nebo čadivý) a charakter kouře CH (souvislý nebo přerušovaný). Tyto parametry mohou nabývat určitého počtu stavů, konkrétně zde $N_T=2$, $N_{CH}=2$. Pro přenos zpráv je použito vzájemných kombinací obou parametrů (tenký souvislý..). Celkový počet všech kombinací je $N=N_T.N_{CH}=2.2=4$. Zadání z příkladu 1 lze zapsat do přehledné tabulky:

stav č.	Tloušťka	CHarakter	zpráva
1	tenký	souvislý	ahoj – chci Ti něco říct
2	tenký	přerušovaný	ulovili jsme mamuta
3	čadivý	souvislý	snědli jsme mamuta
4	čadivý	přerušovaný	?

Tab. 1. Informační model kouřového signálu z Př. 1.

Stav č. 4 není využit, lze jej využít dodatečně po přiřazení konkrétní zprávy.

þ

s Příklad 4: Ve Windows98 je zvukový klip *Zvuk Microsoft.wav* s parametry:

Velikost 693 212 bajtů, délka záznamu 7,859s, zvukový formát PCM, vzorkovací kmitočet 22050 Hz, kvantování na 16 bitů.
Signál je tedy možno chápat jako posloupnost (7.859x22050)=173 291 vzorků. Každý z vzorků může nabývat 2^{16}= 65536 hladin (stavů). Zvuková informace je uložena v jediném parametru signálu – v jeho výšce (a samozřejmě ve vzorkovacím kmitočtu).
Pokud by se jednalo o analogový nekvantovaný signál, byl by počet možných hladin (stavů) signálu teoreticky nekonečný. Z praktického hlediska je však potřebný počet stavů vždy omezen i u analogového zpracování (konečné rozlišení dílčích bloků přenosového řetězce, šum, ..).

þ

Zobecnění: **V abstraktním pojetí** je signál množinou po sobě jdoucích povolených stavů (resp. kombinace stavů) dané fyzikální veličiny. Povolený stav je u číslicového přenosu vybírán vždy z diskrétní množiny stavů.
Informačním modelem signálu je množina po sobě jdoucích informačních prvků, pomocí nichž jsou určitým způsobem kódovány jednotlivé povolené stavy.
Fyzickou realizací informačních prvků, resp. jejich kombinací (viz dále definici informačního slova) jsou tzv. **signálové prvky**.

Sémantická a syntaktická abeceda.
Syntaktická abeceda je množina všech možných informačních prvků signálu. Skupina po sobě jdoucích informačních prvků tvoří informační slovo. V případě binárních (dvoustavových) dat se informační prvek nazývá bit (nikoliv Shannon) a některá vyhrazená informační slova (např. osmibitová) byte (bajt).
Sémantická abeceda je množina všech možných prvků zprávy. Prvky zprávy se označují jako písmena. Skupina po sobě jdoucích písmen tvoří slovo zprávy.
V teorii informace se abecedou rozumí abeceda syntaktická.
Někdy se hovoří ještě o fyzikální abecedě: je to množina všech možných signálových prvků.

Při přenosu dat mohou nastat m.j. tyto případy:

1. Každému písmenu sémantické abecedy je přiřazen jeden informační prvek (atypické), který je fyzicky realizován signálovým prvkem.
2. Každému písmenu sémantické abecedy je přiřazeno povolené informační slovo (časté), které se nazývá **kódové slovo**. To je realizováno signálovým prvkem, případně jejich posloupností.

Obecně je možno říci, že informační model signálu se skládá z posloupnosti kódových slov (protože informační prvek je speciální – nejkratší možné – kódové slovo).

s Příklad 5: Kouřový signál z Př. 3, Tab. 1:

Informační prvky jsou 4: tenký, čadivý, souvislý, přerušovaný. Tvoří syntaktickou abecedu.
Prvky zprávy jsou tvořeny základními znaky české abecedy a znaky pomocnými (mezera, interpunkce,..).

þ

s Příklad 6: Zvukový signál z Př. 4, *Zvuk Microsoft.wav*:

Informační prvky jsou pouze dva: nula a jednička. Tvoří syntaktickou binární abecedu. Každá kombinace 16 po sobě jdoucích informačních prvků tvoří povolené kódové slovo. Informační model zvukové nahrávky se skládá z posloupnosti 173 271 šestnáctibitových kódových slov.
Prvky zprávy je obtížné definovat, závisí na způsobu, jak se díváme na rozklad hudební nahrávky na nezávislé elementy (spektrální hledisko, LPC rozklad,…).

þ

s Příklad 7: Vyjádření zjednodušeného ASCII textu linkovým kódem.

Pro účely přenosu textu obsahujícího symboly z 32-prvkové sémantické abecedy {A B C D.. Z , . ? ! / _ } bipolárním RZ linkovým kódem po vedení jsou písmena zprávy průběžně převáděna na pětibitová slova. Každý bit je pak převáděn linkovým kodérem na příslušný signálový prvek. Informační model linkového signálu se skládá z posloupnosti informačních prvků 0 a 1 z dvouprvkové syntaktické abecedy {0 1}. Fyzikální abeceda je tvořena známou dvojicí signálových prvků bipolárního kódu RZ.

þ

Informační kapacita soustavy podle Hartleye. Aditivní vlastnost [9].
Metoda kouřových signálů popsaná v Př. 1 umožňuje přenos jen tří jednoduchých zpráv. Není například jasné, jak by se přenesla zpráva, že vzápětí byl uloven další mamut atd. K sdělení větších podrobností, např. jak je mamut veliký, by bylo nutné informační soustavu podstatně rozšířit. Intuitivně cítíme, že dané signály mají malou „informační kapacitu“.
Přejdeme-li na jiné příklady, které již mají blíže k současnému přenosu dat (např. Př. 6), pak je zřejmé, že informační kapacitu signálu můžeme zvyšovat jeho prodlužováním (čím více vzorků bude obsahovat zvukový klip, tím větší informaci „pojme“).

Nechť signál je složen z posloupnosti n kódových slov o celkové délce n_I informačních prvků. Pak říkáme, že délka signálu je n_I informačních prvků, resp. n kódových slov.
(Syntaktická) abeceda nechť obsahuje N informačních prvků. Pro N=2 hovoříme o binární abecedě. Dále nechť celkový počet všech povolených kódových slov je N_K. Pak celkový počet N_Z všech navzájem různých zpráv, které je možné signálem vyjádřit, je

$$N_Z = N_K^n . \qquad (1)$$

s Příklad 8: Telegram využívá 32-prvkovou abecedu (N=32). Kolik lze sestavit různých telegramů o délce 50 znaků (n=50)?

Zde každé písmeno abecedy odpovídá jednomu kódovému slovu, které je současně informačním prvkem. Proto

$N_Z = 32^{50} = 10^{\log(32^{50})} \doteq 10^{75}$ různých telegramů.

Poznámka: z hlediska sémantického, které je teorií informace ignorováno, je však většina z těchto telegramů nesmyslných.

þ

s Příklad 9: Kolik je možné vytvořit různých zvukových klipů o parametrech klipu *Zvuk Microsoft.wav* z Př. 4 (délka záznamu 7,859s, zvukový formát PCM, vzorkovací kmitočet 22050 Hz, kvantování na 16 bitů)?

N=2, N_K=2^{16}, n=7,859x22050=173 291, $N_Z = \left(2^{16}\right)^{173291} \doteq 10^{834652}$ různých klipů.

Poznámka: V tomto nepředstavitelném počtu jsou kromě znělky Windows zahrnuty všechny možné segmenty dané délky všech existujících hudebních nahrávek i těch, které teprve budou vymyšleny, a všech možných i nemožných výkřiků, skřeků a jiných zvuků a jejich vzájemných kombinací.

þ

s Příklad 10: Odhadněte počet různých půlminutových telefonních hovorů. Předpokládejte, že telefonní signál je kmitočtově omezen do F_M=3,4kHz a přenášen technikou PCM, tj. vzorkován 8000 krát za sekundu, přičemž každý vzorek je kvantován 8 bity na 256 úrovní.

N=2, N_K=2^8=256, n=30x8000=240000, $N_Z = 256^{240000} \doteq 10^{577978}$ různých telefonních hovorů.

þ

Výsledky příkladů 7 až 9 poskytují číselné údaje o jakési „informační kapacitě“ daného signálu (kolik různých zpráv umožňuje vyjádřit).

Informační kapacita soustavy podle Hartleye (1928). Do pojmu soustava zahrnoval Hartley kromě signálů v našem „informačním“ pojetí i sdělovací kanály a zprávy, vůbec vše, co se může nacházet v množině diskrétních stavů.
Informační kapacita soustavy

$$C = \log_2 N_S , \qquad (2)$$

kde N_S je počet všech možných stavů soustavy. Původní jednotka informační kapacity byla bit, nyní (bohužel) Shannon (čti Šenon) [Sh].

s Příklad 11: Jestliže je za soustavu považován telegram z Př. 8, pak počet jeho všech možných stavů N_S je roven počtu všech různých telegramů N_Z =32^{50}. Informační kapacita telegramu je $\log_2 32^{50}$=250 Sh.

þ

Aditivní vlastnost informační kapacity. Uvažujme dvě navzájem oddělené soustavy o informačních kapacitách C_1 a C_2. Jejich sloučením vznikne soustava o informační kapacitě

$$C=C_1+C_2. \tag{3}$$

Tato vlastnost je logická: informační kapacita dvou telegramů musí být rovna součtu informačních kapacit telegramů dílčích. Definiční vztah (2) informační kapacity, v němž se objevuje logaritmus, tuto vlastnost automaticky zajišťuje bez ohledu na základ logaritmu: Jsou-li počty stavů jedné a druhé soustavy N_{S1} a N_{S2}, pak celkový počet stavů soustavy vzniklé jejich sloučením bude $N_S = N_{S1} . N_{S2}$. Kapacita výsledné soustavy pak bude podle (2)

$$C = \log_2\left(N_{S1}N_{S2}\right) = \log_2 N_{S1} + \log_2 N_{S2} = C_1 + C_2 .$$

Základ logaritmu ovlivňuje pouze multiplikativní konstantu informační kapacity (a tím její jednotku), nikoliv však vlastnost aditivity. Dvojka se používá s ohledem na používanou binární syntaktickou abecedu: zvětší-li se délka binárního signálu o jeden informační prvek, pak jeho informační kapacita vzroste v důsledku dvojkového základu o 1 Sh.

Ověřte si, jak souvisí informační kapacita zvukového souboru *Zvuk Microsoft.wav* z příkladu 4 s jeho velikostí (693 212 bajtů). Připomínáme, že nahrávka je stereofonní.

Informační kapacita podle Hartleye sice umožňuje kvantifikovat množství informace, které „se vejde“ do dané soustavy, zcela však pomíjí hledisko příjemce informace, tj. nakolik je pro něj přijatá informace důležitá. Je totiž možné, že v důsledku určitých skutečností může příjemce znát dopředu s velmi vysokou pravděpodobností, co přijme. Pak je pro něj informační obsah přijaté zprávy prakticky bezcenný. Jinak je tomu u sdělení, jehož informační obsah není schopen predikovat. Tuto „kvalitu“ informace nelze popsat vzorcem (2). Musíme použít aparát pravděpodobnostního počtu a náhodných jevů.

Úplný soubor vzájemně se vylučujících nezávislých náhodných jevů [5,8,9,10].

Fyzikální jevy sledujeme na základě signálů, které jsou těmito jevy generovány. Pokud nejsme schopni dopředu přesně určit, jakých hodnot tyto signály nabudou v daných časových okamžicích, prohlásíme tyto jevy za náhodné.

Při přenosu dat se často setkáváme se signály diskrétními v hodnotách, kdy v daných časových okamžicích dopředu neznáme úroveň přijatého signálu (0?1, resp. A?B?C?D?..). Těmito případy se budeme dále zabývat.

Uvažujme soubor N náhodných jevů:

$$\{X\} = \{x_1, x_2, ..., x_N\}.$$

Výskyt jednoho z jevů v daném okamžiku vylučuje výskyt jiných jevů: v daném okamžiku může nastat pouze jeden z nich. Pak soubor *{X}* nazveme **úplným souborem vzájemně se vylučujících náhodných jevů**.

Jestliže pravděpodobnost výskytu daného jevu nezávisí na způsobu, jakým se dílčí jevy objevovaly před tímto výskytem, pak náhodné jevy označíme za **vzájemně nezávislé**. Tento případ budeme dále uvažovat.

Pravděpodobnosti výskytu dílčích jevů v daném okamžiku nechť jsou

$P(x_1), P(x_2), \ldots, P(x_N)$.

Pak pro pravděpodobnosti úplného souboru vzájemně se vylučujících náhodných jevů platí

$$\sum_{i=1}^{N} P(x_i) = 1. \tag{4}$$

Úplný soubor jevů spolu s pravděpodobnostmi jejich výskytu se někdy zapisují do tzv. konečného schématu [1]:

$$\begin{pmatrix} x_1 & x_2 & x_3 & \mathbf{K} & x_i & \mathbf{K} & x_N \\ P(x_1) & P(x_2) & P(x_3) & \mathbf{K} & P(x_i) & \mathbf{K} & P(x_N) \end{pmatrix} \tag{5}$$

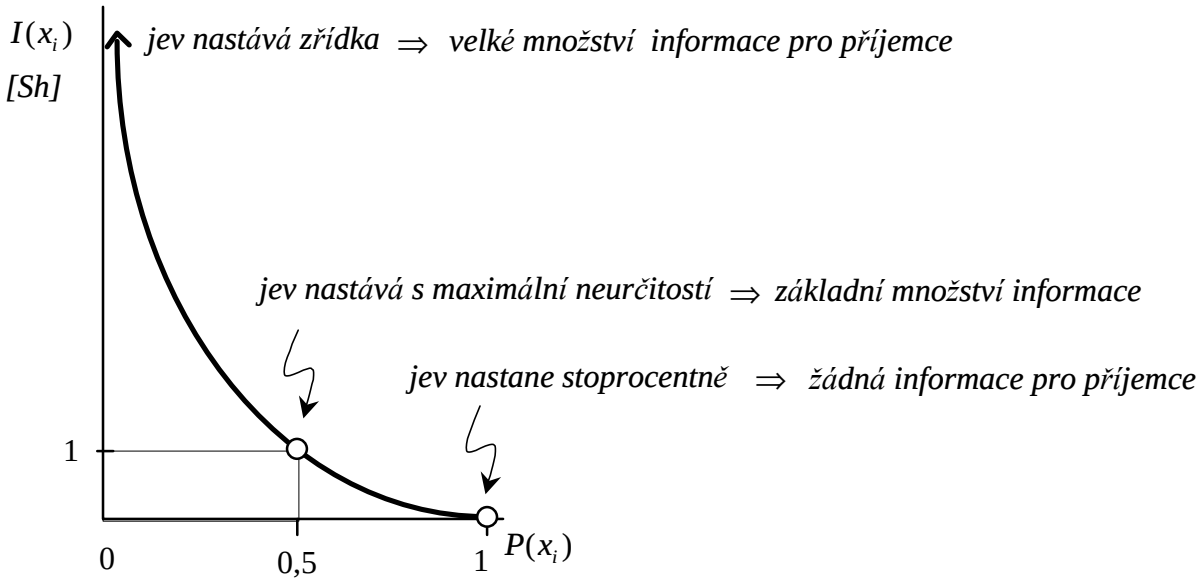
s Příklad 12: Demodulátor přijímače signálu BPSK vyhodnocuje v rozhodovacích okamžicích příjem nuly nebo jedničky. S procesem vyhodnocování je spjat úplný soubor dvou vzájemně se vylučujících náhodných jevů:

Jev x_1: vyhodnotí se příjem nuly.
Jev x_2: vyhodnotí se příjem jedničky.
Při demodulaci lze průběžně registrovat četnost vyhodnocování nul a jedniček, které konvergují k pravděpodobnostem $P(x_1)$ a $P(x_2)$. Jejich součet musí být z pochopitelných důvodů jednička. Při přenosu dat vzniklých digitalizací reálných signálů není důvod, aby četnosti výskytu nul a jedniček byly různé. Pak bývá splněna rovnost $P(x_1) = P(x_2)$, která však může být narušena např. systematickou chybou vnášenou do vyhodnocování dekodérem.

þ

Kvantitativní vyjádření množství informace na základě pravděpodobnosti [6,8,9,10].
Množství informace, kterou získá příjemce přijetím zprávy, že v daný okamžik došlo k výskytu jevu x_i z úplného souboru vzájemně se vylučujících jevů $\{X\}$, je dáno vzorcem

$$I(x_i) = \log_2 \frac{1}{P(x_i)} = -\log_2 P(x_i) \ \text{[Sh]} \qquad (6)$$



Obr. 2. Grafické znázornění vzorce (6) – závislost množství informace na pravděpodobnosti výskytu jevu.

Logaritmus v definičním vzorci zajišťuje aditivitu množství informace v případě současného výskytu několika vzájemně nezávislých jevů.

Chápání vzorce (6) z pohledu „užitečnosti zprávy“ [10].
Jestliže přijmeme zprávu o tom, že došlo k výskytu daného jevu, který ovšem musel zákonitě nastat, protože jeho pravděpodobnost je 1 (např. že slunce vyšlo na východě), „užitečnost“ této zprávy ohodnotíme tak, že jí přiřadíme nulovou informační hodnotu (na obr. 2 pravý krajní bod). Naopak informačně zajímavá pro nás bude zpráva o udání něčeho, co se stává velmi zřídka ($I \to \infty$ pro $P \to 0$). Těmto představám plně vyhovuje vzorec (6), který navíc zajišťuje výše zmíněnou aditivitu.

Chápání vzorce (6) z pohledu návrháře digitálního přenosu zprávy [10].
Snažíme-li se co nejúsporněji zakódovat přenášenou informaci do signálu tak, aby její přenos trval co nejkratší dobu, pak zjistíme samozřejmou věc, že množství informace ve zprávě je úměrné času potřebnému pro přenos zprávy: přenášení dvojnásobného množství informace trvá dvojnásobnou dobu. Snaha o co nejúspornější kódování nás vede k této strategii: znaky, které se vyskytují

v průměru velmi často, kódujeme co nejkratšími kódovými slovy (jejich přenos pak potrvá krátkou dobu), zatímco znaky vyskytující se zřídka lze kódovat slovy delšími.
Uvažujme pouze dva znaky a_1 a a_2, které mají stejné pravděpodobnosti výskytu. Tyto znaky reprezentujme nejjednoduššími kódovými slovy 0 a 1, tj. jednobitovým kódem.
V druhé fázi uvažujme 4 znaky a_1, a_2, a_3 a a_4 , opět se stejnou četností výskytu. Nyní je zakódujme kódovými slovy 00, 01, 10 a 11, tj. dvoubitovým kódem.
Úvahu zobecníme na n znaků a_1 až a_n , kde n je celá mocnina dvou, které lze vyjádřit kódovými slovy délky $\log_2 n$.
Je zřejmé, že přenos binární posloupnosti, složené z dvoubitových kódových slov, bude trvat dvakrát tak dlouho než přenos posloupnosti po jednobitovém kódování. „Dvoubitová“ posloupnost je dvakrát tak delší a obsahuje dvojnásobné množství informace.
Z opačného pohledu, k zakódování každého z n znaků (které se vyskytují se stejnou pravděpodobností) potřebujeme $\log_2 n$ bitů. Ze stejných pravděpodobností plyne jejich velikost, totiž $1/n$. Proto každý znak, kterému přísluší pravděpodobnost výskytu P, je nutno zakódovat $\log_2(1/P)$ bity. Množství informace nesené tímto znakem (resp. spojené s výskytem tohoto znaku) tedy musí být úměrné výrazu $\log_2(1/P)$:

$$I = k.\log_2 \frac{1}{P}.$$

Konstantu k lze nepřímo modifikovat volbou základu logaritmu, podle úmluvy je jednotková. Při dvojkovém základu je jednotkou množství informace 1 shannon (dříve bit), pří základu 10 je jednotkou 1 hartley (3,32 shannonů), a při základu e 1 nat (1,44 shannonů).
Pozn.: Výše uvedená úvaha byla pro jednoduchost provedena pro znaky se stejnou pravděpodobností, vzorec (6) je však platný pro libovolné pravděpodobnostní rozložení. Množství informace I, jak poznáme dále, lze definovat nejen pro jednotlivé znaky, ale i pro celé zprávy, skládající se z těchto znaků. Pak množství informace v binární zprávě je rovno minimálnímu počtu bitů potřebnému k zakódování této zprávy.

s Příklad 13: Na výstupu demodulátoru BPSK se objevují jedničky a nuly s pravděpodobnostmi P(1) = 0,9 a P(0) =0,1. Jaké množství informace získáme zjištěním, že byla přijata jednička?

$I(1) = -\log_2 0{,}9 \approx 0{,}152$ Sh.

þ

s Příklad 14: Na výstupu demodulátoru BPSK se objevují jedničky a nuly s pravděpodobnostmi $P(1) = 0{,}9$ a $P(0) = 0{,}1$. Jaké množství informace získáme přijetím zprávy 1101?

Existuje celkem 16 možných zpráv délky 4. Jejich přijetí je možné chápat jako úplný soubor 16 vzájemně se vylučujících náhodných jevů. Hledaná informace se určí pomocí (6) z pravděpodobnosti výskytu dané zprávy.

Za předpokladu nezávislosti pravděpodobností P(1) a P(0) na předchozích výskytech nuly a jedničky se pravděpodobnost přijetí zprávy vypočte podle vzorce

$P(1).P(1).P(0).P(1) = P^3(1).P(0) = 0{,}9^3 \,.\, 0{,}1 = 0{,}0729$. Pak

$I = -\log_2 0{,}0729 \approx 3{,}78$ Sh.

Pokud by byly pravděpodobnosti stejné, tj. $P(1) = P(0) = 0{,}5$, pak by pravděpodobnost přijetí jakékoliv čtyřbitové zprávy byla

$P^4(1) = 0{,}0625$,

čemuž by odpovídala informace

$I = -\log_2 0{,}0625 = 4$ Sh.

þ

Z příkladu 14 je zřejmé, různě pravděpodobné zprávy stejné délky budou mít pro příjemce různý informační obsah. Užitečné bude například zjistit, jaké je průměrné množství informace ze všech 16 čtyřbitových zpráv a toto číslo srovnávat s průměrným množstvím informace neseným zprávami jiné délky. Za tím účelem se používá speciální charakteristika úplného souboru náhodných jevů – entropie.

Entropie úplného souboru náhodných jevů. Fyzikální význam, vlastnosti [9].
Zatímco množství informace *I* popisuje vlastnost jednoho z úplného souboru jevů, entropie *H* je číslo, které charakterizuje úplný soubor jako celek.
Uvažujme úplný soubor *N* jevů s konečných schématem podle (5):

$$\begin{pmatrix} x_1 & x_2 & x_3 & \mathbf{K} & x_i & \mathbf{K} & x_N \\ P(x_1) & P(x_2) & P(x_3) & \mathbf{K} & P(x_i) & \mathbf{K} & P(x_N) \end{pmatrix}.$$

Rozložení dílčích pravděpodobností ve spodním řádku vypovídá o neurčitosti v očekávání příjemce, který z jevů nastane. Například při rovnoměrném rozložení pravděpodobností, kdy

$$P(x_1) = P(x_2) = \ldots = P(x_N) = \frac{1}{N} \tag{7}$$

bude tato neurčitost maximální možná. Také je možné si představit, že neurčitost poroste s růstem počtu jevů *N*. Opakem bude situace, kdy jedna z pravděpodobností bude jednotková a ostatní tudíž nulové: takovýto úplný soubor jevů již nebude náhodný, nýbrž deterministický, protože bude pravidelně docházet pouze k výskytu jevu s pravděpodobností 1. Pak je neurčitost příjemce v očekávání toho, který jev se objeví, nulová.
Daná neurčitost příjemce se dá jednoduše vyčíslit a nazývá se entropie. Neurčitost v očekávání příjemce zaniká v okamžiku, kdy jeden z jevů nastane. Protože současně příjemce získává informaci o objevení se daného jevu z úplného souboru, můžeme říci, že entropie úplného souboru jevů se číselně rovná množství informace připadající na výskyt jednoho jevu.
Entropie jako funkce tedy musí splňovat tyto požadavky:

1. Musí být funkcí všech pravděpodobností $P(x_i)$, i=1..*N*.
2. Musí nabývat maximální hodnoty při rovnoměrném rozdělení pravděpodobností (7). Tato maximální hodnota musí růst při rostoucím počtu jevů *N*.
3. Musí být nulová při deterministickém rozložení, kdy jedna z pravděpodobností je jedničková.
4. Musí být zachována kompatibilita s definicí množství informace (6), neboť entropie se rovná množství informace připadající na výskyt jednoho jevu.

Uvedeným požadavkům vyhovuje definice entropie jako průměrné množství informace z úplného souboru náhodných jevů $\{X\}$. V níže uvedeném vzorci představuje symbol *E* matematickou naději (střední hodnotu):

$$H(X) = E\{I(x_i)\}\Big|_{i=1,2,..,N} = \sum_{i=1}^{N} P(x_i) I(x_i) \tag{8}$$

Po dosazení (6) vyjde konečný vztah

$$H(X) = -\sum_{i=1}^{N} P(x_i) \log_2 P(x_i) \tag{9}$$

Jednotkou takto definované entropie je [Sh]/jev. Je-li jevem výskyt jednoho z možných symbolů zprávy, resp. prvků signálu, pak je jednotkou [Sh]/symbol, resp. [Sh]/prvek. Později se seznámíme i s jednotkou [Sh]/sekunda.

s Příklad 15: Entropie úplného souboru dvou vzájemně se vylučujících jevů (binárního souboru).

Uvažujme úplný soubor dvou náhodných jevů z Př. 12:
Jev x_1: vyhodnotí se příjem nuly.
Jev x_2: vyhodnotí se příjem jedničky.

Konečné schéma souboru bude ve tvaru

$$\begin{pmatrix} x_1 & x_2 \\ P & Q \end{pmatrix}, \tag{10}$$

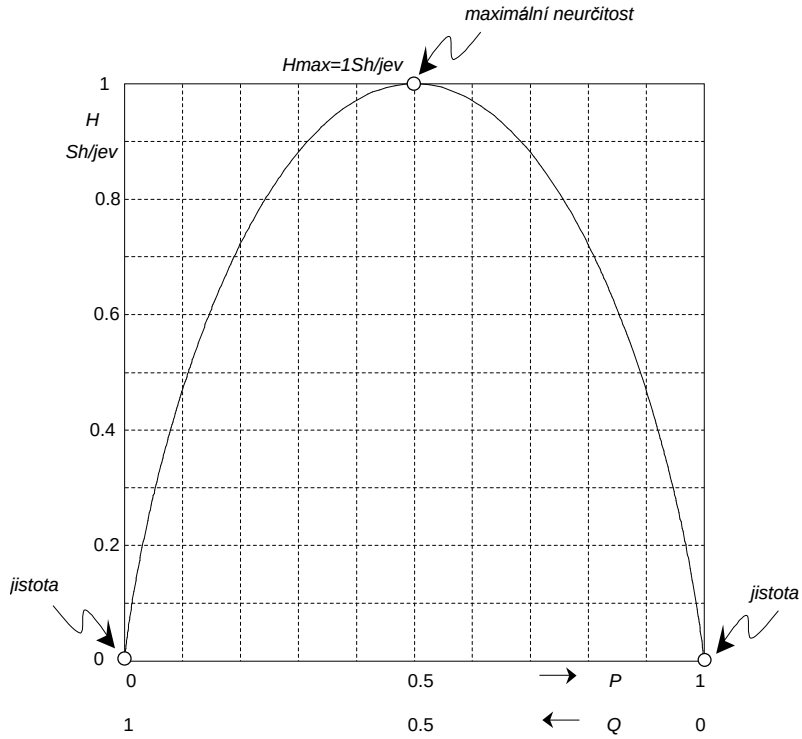
kde podle (4)

$$P + Q = 1. \tag{11}$$

Dosazením do vzorce pro entropii (9) dostáváme výsledek

$$H(X) = -(P\log_2 P + Q\log_2 Q) = -P\log_2 P - (1-P)\log_2(1-P) \tag{12}$$

Grafické znázornění je na obr. 3. Graf potvrzuje první a třetí podmínku ze str. 6 a první část podmínky 2.



Obr. 3. Grafické znázornění vzorce (12) – závislost entropie binárního souboru na pravděpodobnostech výskytů *P* a *Q*.

þ

Věta o maximální entropii při rovnoměrném rozdělení pravděpodobností platí obecně pro soubory s *N* jevy. Matematický důkaz je možné nalézt např. v [3]. Dosadíme-li do vzorce (9) pro entropii podmínku rovnoměrného rozdělení (7), vyjde po úpravě

$$H_{\max} = H(X)\Big|_{P(x_i)=1/N,\, i=1,2,..,N} = \log_2 N \tag{13}$$

Maximální entropie roste s rostoucím počtem jevů *N* v souboru, což je v souladu s druhou částí podmínky 2 na str. 6.

Po porovnání (13) a (2) zjišťujeme, že

Maximální možná entropie úplného souboru jevů se číselně rovná jeho informační kapacitě podle Hartleye.

Někdy se hovoří o (bezrozměrné) relativní entropii úplného souboru náhodných jevů, což je poměr entropie a její maximální hodnoty:

$$h = \frac{H}{H_{\max}} \in \langle 0,1 \rangle . \tag{14}$$

Na základě relativní entropie se zavádí ještě redundance (nadbytečnost)

$$r = 1 - h = \frac{H_{\max} - H}{H_{\max}} \in \langle 0,1 \rangle . \qquad (15)$$

s Příklad 16: entropie abecedy.

Uvažujme zjednodušenou abecedu sestavenou z 32 ASCII znaků podle Př. 7:
{A B C D.. Z , . ? ! / _ }.
Její maximální entropie v případě stejné pravděpodobnosti výskytu všech znaků by byla
$H_{max} = \log_2 32 = 5$ Sh/znak.
Ve skutečnosti by bylo průměrné množství informace na znak psaného textu menší ze dvou důvodů:

1. Různé znaky se vyskytují s různou pravděpodobností.
2. Výskyt znaku na daném místě textu je vázán na předchozí text řadou syntaktických a gramatických pravidel. To znamená, že výskyt znaků abecedy na daném místě nepředstavuje soubor nezávislých jevů, čímž je porušen základní předpoklad pro naše výpočty. Toto omezení volnosti ve volbě znaků z abecedy nutně znamená další snížení reálné informační kapacity zprávy. Úměrně tomu se však zvyšuje její tzv. redundance (nadbytečnost), která naopak přispívá k zvýšení její srozumitelnosti a v případě výskytu chyb může být předpokladem k jejich odstranění nebo alespoň detekci.

þ

Entropie abecedy versus informace nesená zprávou

Entropie byla v teorii informace původně zavedena pro úplný soubor vzájemně se vylučujících jevů. Z tohoto pohledu je v pořádku hovořit o entropii abecedy, čímž se myslí entropie souboru jevů, spojených s náhodným výskytem jednoho ze znaků abecedy na sledované pozici ve zprávě. Pak entropie abecedy bude průměrné množství informace, nesené jedním prvkem zprávy.
Případ nezávislých výskytů (např. zprávy vzniklé digitalizací analogových dějů):
V případě, že pravděpodobnosti $P(x_i)$, i = 1,2, .., N výskytu prvků abecedy na daném místě zprávy budou zcela nezávislé na skutečnosti, jaké prvky se objevily na předchozích místech, budou náhodné jevy spojené s výskytem prvků abecedy na různých místech zprávy nezávislé. Pak vynásobíme-li entropii abecedy délkou zprávy L, tj. počtem jejích prvků, získáme průměrné množství informace nesené celou zprávou:

$$I_{zprávy} = L.H_{abecedy} \qquad (16)$$

Slovo průměrné je důležité (viz Př. 14): konkrétní množství informace závisí totiž na pravděpodobnosti výskytu konkrétní zprávy dané délky.

Případ závislých výskytů (např. literární texty):
Zde pravděpodobnosti $P(x_i)$, i = 1,2, .., N výskytu prvků abecedy na daném místě zprávy mohou být silně závislé na tom, jak vypadá předchozí text. Pak průměrné množství informace nesené celou zprávou bude menší než součin délky zprávy a entropie abecedy:

$$I_{zprávy} < L.H_{abecedy} \qquad (17)$$

Je tomu tak proto, že vazby mezi jednotlivými znaky zprávy snižují neurčitost příjemce zprávy. Vzorec (17) lze interpretovat také tak, že informační obsah zprávy dané délky klesl v důsledku vazeb mezi znaky proto, že klesla „efektivní“ entropie abecedy oproti entropii bez uvažování těchto vazeb.

s Příklad 17: množství informace ve zprávě.

Uvažujme přenos telefonního signálu modulací PCM. Binární zpráva má délku $L=10^6$ prvků. Jedničky a nuly se vyskytují s pravděpodobnostmi P(1) = P(0) =0,5 nezávisle na historii zprávy.

Entropie této dvouprvkové abecedy je maximální možná a její hodnota je 1Sh/znak. Množství informace přenášené zprávou je 1.10^6 =1MSh (dříve 1Mbit).
Pokud by např. v důsledku nevhodného kódování došlo k změně pravděpodobností na P(1) = 0,3 a P(0) = 0,7, pak by se entropie abecedy snížila na hodnotu
$-(0{,}3\log_2 0{,}3+0{,}7\log_2 0{,}7)=0{,}881$ Sh/znak.
To by mělo za následek pokles množství přenášené informace na velikost 0,881 MSh/znak.

þ

Entropie a redundance zdroje zpráv.
Na počátku sdělovacího řetězce obvykle stojí zdroj zprávy, který může mít mnoho forem (senzor, přehrávač, počítač, člověk..). Navzdory různorodému charakteru zpráv je možné nalézt jejich společný rys: vždy existuje určitá množina možných „sdělení“ spolu s pravděpodobnostmi jejich výskytu. Pak lze ovšem použít entropii k ohodnocení informačního obsahu generovaných zpráv. U diskrétních zdrojů zpráv je pak jednoduše entropie zdroje rovna entropii používané abecedy.
Entropii zdroje můžeme určovat jednak klasicky v Sh/znak, nebo také v Sh/sekundu. Přepočet je snadný – entropie v Sh/znak násobená počtem generovaných znaků za sekundu je rovna entropii v Sh/s. Druhá z uvedených entropií tak souvisí s přenosovou rychlostí v bitech/s.
Z entropie zdroje lze pomocí (15) vypočítat jeho redundanci, která vyjádří jeho „informační rezervu“.

s Příklad 18: zdroj čtyřstavových digitálních dat.

Uvažujme zdroj generující čtyřstavová data v úrovních 1V, 2V, 3V a 4V, které odpovídají syntaktické abecedě {A B C D}. Délka trvání každého signálového prvku je 1ms. Pravděpodobnosti výskytu jednotlivých prvků abecedy jsou popsány konečným schématem

$$\begin{pmatrix} A & B & C & D \\ 1/2 & 1/4 & 1/8 & 1/8 \end{pmatrix}. \qquad (18)$$

Klasickým výpočtem bez využití teorie informace dospějeme k modulační rychlosti

$$M = \frac{1}{1ms} = 1\ \text{kBd}$$

a přenosové rychlosti

$$R = 2M = 2\ \text{kbit/s} \qquad (19)$$

(čtyřstavová data v každém signálovém prvku jsou reprezentována dvojicí bitů).
Ze schématu (18) však vyplývá, že prvky *C* a *D* se oproti prvku *A* budou ve zprávě vyskytovat jen „občas“. Kdyby ve zprávě nebyly vůbec, znamenalo by to vlastně jen dvoustavové, tj. jednobitové kódování s přenosovou rychlostí $1kbit/s$. Nerovnoměrné rozložení pravděpodobnosti výskytu prvků abecedy tak vede k snižování „efektivní“ přenosové rychlosti oproti teoretickému výsledku (20). Tento jev lze přesně popsat s využitím entropie zdroje.
Podle (18) bude entropie zdroje
$H_{zdroje} = -(0{,}5\log_2 0{,}5 + 0{,}25\log_2 0{,}5 + 2.0{,}125\log_2 0{,}125) = 7/4$ Sh/znak.
Entropii zdroje v Sh/s získáme vynásobením výsledku počtem signálových prvků za sekundu, t.j. modulační rychlostí:

$$H'_{zdroje} = H.M = 7/4\ \text{kSh/s} = 1{,}75\ \text{kSh/s} \qquad (20)$$

Po formální náhradě jednotky Shannon za praktický bit vidíme pokles „efektivní“ přenosové rychlosti z teoretické hodnoty 2 kbit/s na 1,75 kbit/s.

þ

Fyzikální a informační rozhraní. Překódování zpráv na rozhraní.
Fyzikální rozhraní je místo, v němž dochází k fyzické změně formy signálu, nesoucího informaci.
Informační rozhraní je místo, v němž dochází k změně informačního modelu signálu, nesoucího informaci.

Příkladem fyzikálního rozhraní je místo konverze optického signálu na odpovídající elektrické impulsy.
Příkladem informačního rozhraní je místo, v němž dochází k překódování digitálního signálu za účelem jeho komprese.
Z hlediska teorie informace jsou podstatné děje probíhající na informačním rozhraní.

Změny informačního modelu signálu můžeme dosáhnout dvojím způsobem: buď změnou syntaktické abecedy, nebo smluvenou transformací celých informačních slov na jiná informační slova. V každém případě hovoříme o tzv. kódování (podrobnosti viz kapitoly o kódování).
Za jakým účelem toto kódování provádíme? Kódováním můžeme pozměnit entropii (a tím i redundanci) abecedy. Tak lze hledat kód, který povede na největší entropii zprávy, neboli na nejmenší počet znaků na dané množství generované informace. Tento kód pak bude „nejekonomičtější“ pro přenos informací, neboť k přenosu potřebuje nejmenší počet znaků a nejkratší dobu přenosu.

s Příklad 19: překódování zdroje čtyřstavových digitálních dat z Př. 18 [9].

Zprávy ze zdroje z Př. 18, který využívá čtyřprvkovou abecedu {A B C D} a je popsán konečným schématem (18)

$$\begin{pmatrix} A & B & C & D \\ 1/2 & 1/4 & 1/8 & 1/8 \end{pmatrix},$$

převedeme do binární abecedy {0 1} dvěma různými kódy:

Kód č. 1: A – 00, B – 01, C – 10, D – 11 (všechna slova jsou stejně dlouhá).
Kód č. 2: A – 0, B – 10, C – 110, D – 111 (slova s častým výskytem jsou kratší než slova objevující se méně často).

Po překódování budou ve zprávě figurovat v obou případech pouze nuly a jedničky. Určíme jejich pravděpodobnosti výskytu. Za tím účelem vytvoříme z původní abecedy „typickou“ zprávu tak, aby frekvence výskytu jednotlivých znaků odpovídala jejich pravděpodobnosti výskytu. Pak tuto zprávu překódujeme zvlášť kódem č. 1 a 2. Z četnosti výskytu nul a jedniček stanovíme pravděpodobnosti.

„Typická“ zpráva:
A A A A B B C D.
Po překódování kódem č. 1:
00 00 00 00 01 01 10 11.
Po překódování kódem č. 2:
0 0 0 0 10 10 110 111.

Výpočet pravděpodobností výskytu nuly a jedničky:
Kód č. 1: P(0) = 11/16, P(1) = 5/16.
Kód č. 2: P(0) = 1/2, P(1) = 1/2.

Nová konečná schémata:
Kód č. 1

$$\begin{pmatrix} 0 & 1 \\ 11/16 & 5/16 \end{pmatrix}.$$

Kód č. 2

$$\begin{pmatrix} 0 & 1 \\ 1/2 & 1/2 \end{pmatrix}.$$

Výpočet entropií:
Původní entropie zdroje před kódováním:
H_0 = 7/4 Sh/znak =1,75 Sh/znak (viz Př. 18).
V „typické“ zprávě, která má délku 8, je tedy množství informace 14 Sh.
Entropie po překódování zdrojem č. 1:
$H_1 = -(11/16 \log_2 11/16 + 5/16 \log_2 5/16) = 0{,}875$ Sh/znak (=H_0/2).
V „typické“ zprávě po překódování, která má délku 16, je tedy množství informace 14 Sh.

Entropie po překódování zdrojem č. 2:
$H_2 = -(1/2 \log_2 1/2 + 1/2 \log_2 1/2) = 1$ Sh/znak (>H_0/2).
V „typické“ zprávě po překódování, která má délku 14, je tedy množství informace 14 Sh.

Informace se překódováním „neztratila“, kód č. 2 s ní však „hospodaří“ lépe než kód č. 1. O tom svědčí maximální entropie H_2. Každý znak v kódu č. 2 nese větší množství informace než znak v kódu č. 1. Entropie po překódování je nutno porovnávat s polovinou původní entropie H_0: zdroj má k dispozici 4 prvky abecedy, kdežto po překódování jsou ve zprávě pouze prvky dva. Entropie - informace na znak před kódováním je totéž jako dvojnásobek informace na znak po kódování.
Nyní provedeme dvojí překódování zdroje podle níže uvedeného schématu:

$$\{A\ B\ C\ D\} \xrightarrow{\text{kód č. 2}} \{0\ 1\} \xrightarrow{\text{kód č. 1}} \{A\ B\ C\ D\}$$

Nejprve zdroj překódujeme výhodnějším kódem č. 2 do abecedy {0 1}. Pak se vrátíme k původní čtyřprvkové abecedě inverzním použitím kódu č. 1. Po takovém dvojím překódování dostaneme pro původní abecedu nové konečné schéma s novými pravděpodobnostmi, které určíme následující úvahou:
Po překódování kódem č. 2 získáme dvouznakový zdroj s entropií 1 Sh/znak. Protože následující kód č. 1 přiřadí každé dvojici binárních znaků jeden ze znaků A, B, C nebo D, dvakrát se tím zmenší délka zprávy. Množství informace se však nezmění, dvakrát tedy vzroste entropie jakožto množství informace na jeden znak. Po dvojím překódování tedy bude entropie 2 Sh/znak, čemuž musí odpovídat rovnoměrné rozdělení pravděpodobnosti podle tohoto konečného schématu:

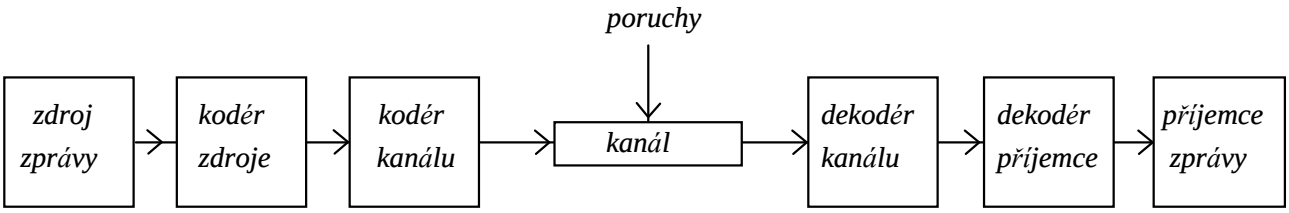
$$\begin{pmatrix} A & B & C & D \\ 1/4 & 1/4 & 1/4 & 1/4 \end{pmatrix},$$

Překódováním jsme získali zdroj informace s nulovou redundancí.
þ

Ukazuje se tedy, že pouhým překódováním lze „vylepšit“ zdroj informace v tom smyslu, že se zvětší jeho entropie čili množství informace na 1 znak, nebo jinými slovy že k přenesení určitého množství informace pak potřebujeme menší počet znaků, čímž lze přenos zrychlit. To je jeden z mnoha dobrých důvodů, proč se budeme kódováním hlouběji zabývat v dalším textu.

Shannonova koncepce komunikačního systému [2,3,5].

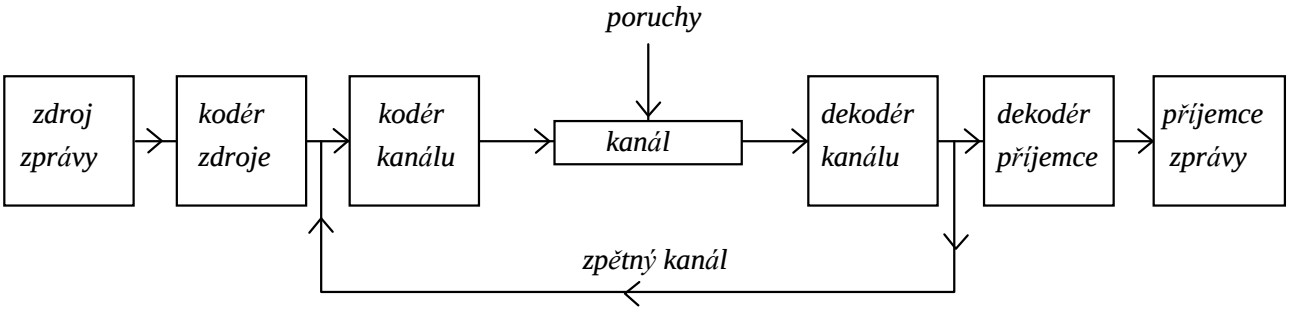
Shannon ukázal v 50. letech, že všechny komunikační systémy používané v minulosti, přítomnosti i později vytvořené jsou pouze zvláštní případy obecného komunikačního systému na obr. 4.



Obr. 4. Shannonovo schéma obecného komunikačního systému.

Úkolem kodéru zdroje je provést kódování zprávy s maximální hospodárností tak, aby pro přenos zprávy byl použit co nejmenší počet znaků. Jinými slovy, je třeba minimalizovat redundanci zprávy a zvýšit tak její entropii, tj. množství informace na jeden znak. Příkladem je komprese textových souborů před jejich přenosem.
Častým úkolem kodéru zdroje je převést původní signál – zdroj informace na signál elektrický, případně transformovat jej do digitální podoby (AD převodník).
Úkolem kodéru kanálu je zabezpečit spolehlivost přenosu tím, že doplňuje informační znaky o přídavné znaky podle určitého algoritmu bezpečnostního kódu. Ten může být buď detekční (příjemce je schopen zjistit, že při přenosu došlo k chybě), nebo korekční (příjemce je navíc schopen lokalizovat místo chyby a opravit ji).
Úkolem dekodéru kanálu je detekovat či opravovat případné chyby při přenosu a rekonstruovat signál tak, aby odpovídal výstupnímu signálu kodéru zdroje.
Úkolem dekodéru příjemce je upravit dekódovanou zprávu na tvar vhodný pro příjemce.
Do kanálu jsou zahrnuty ostatní transformace signálu při přenosu (modulátor a demodulátor, přenosové médium, působení rušení atd.).

V [3] jsou ukázány příklady moderních komunikačních systémů, které byly vyvinuty až daleko po vzniku Shannonovy koncepce a přesto jsou jejími speciálními případy: Systém digitální družicové televize a digitální veřejný celulární radiotelefonní systém GSM.
Není-li extrémní požadavek na spolehlivost přenosu zprávy (netrváme-li na správném přenosu „každého bitu“, např. při přenosu řeči) nebo je-li úroveň rušení při přenosu relativně malá, vystačíme s koncepcí na obr. 4 při současném zabezpečení dat korekčním kódem. Takové systémy přenosu se nazývají FEC (Forward Error Correction, dopředná korekce chyb).
Požadavky na vysoce spolehlivý přenosu dat (např. přenos binárních souborů pomocí Internetu) vedly k doplnění schématu na obr. 4 o tzv. zpětnovazební kanál (viz obr. 5). Data jsou zde většinou zabezpečena jen detekčním kódem. Zpětnovazební systémy se pak označují zkratkou ARQ – Automatic Request for Repetition (automatická žádost o opakování přenosu). Jsou dvojího druhu:



Obr. 5. Shannonovo schéma obecného komunikačního systému doplněné o zpětnovazební kanál.

Systémy s rozhodovací zpětnou vazbou DFB (Decision Feedback): Přijímač vyhodnocuje zprávu po daných slovech a posuzuje její věrnost s využitím detekčního kódu. Není-li zjištěna chyba, vyšle přijímač zpětným kanálem vysílači tzv. poděkování ACK (Acknowledgment). Je-li výsledek prověrky negativní, zpětnovazebním kanálem je vyslán příkaz NACK (Negative Acknowledgment – negativní poděkování) k opakování přenosu daného slova.

Rozhodnutí o případném opakování přenosu tedy vzniká na straně ***přijímače***. Zpětný kanál přenáší pouze jednoduché řídicí příkazy, a proto může být pomalý. Nevýhodou je, že nejsou opraveny chyby, které daný kód není schopen detekovat. Druh kódu je proto třeba volit velmi pečlivě s ohledem na charakter rozložení chyb.
Systémy s informační zpětnou vazbou IFB (Information Feedback): Přímým kanálem jsou vysílána pouze nezabezpečená slova zprávy. Zabezpečující část je ponechána v paměti vysílače. Na základě přijatého slova (které může být narušeno) je na straně přijímače vypočtena zabezpečující část, která je vyslána zpětným kanálem k přijímači. Zde dojde k porovnání s údajem v paměti. Je-li výsledek porovnání negativní, vysílání se opakuje. V opačném případě vyšle vysílač pokyn k uvolnění dat v paměti přijímače a pokračuje ve vysílání dalšího slova.
Zpráva, která se posílá zpět a která je odvozena od přijatého slova, se nazývá kvitance. V nejjednodušším případě může být kvitancí celé přijaté slovo.
Rozhodnutí o případném opakování přenosu tedy vzniká na straně ***vysílače.*** Nevýhodou je, že zpětný kanál musí zabezpečit přenosovou rychlost srovnatelnou s přenosovou rychlostí dopředného kanálu. Další nevýhodou je případné zbytečné opakování vysílání, dojde-li k chybě ve zpětném kanálu. Výhodou je však vysílání pouze nezabezpečených dat. Protože k rozhodování o případné chybě dochází na straně vysílače na základě porovnání atributů vyslaného a přijatého slova, tento systém je značně spolehlivý.

Systémy bez zpětného kanálu jsou úspornější co do požadavků na přenosovou rychlost, resp. šířku pásma dopředného kanálu, účinnost zabezpečení přenosu je však menší. Používají se tam, kde je realizace zpětného kanálu obtížná nebo nemožná (např. družicová komunikace). V praxi se rovněž používají systémy kombinované, kdy se zpětný kanál vytváří adaptabilně v závislosti na momentální chybovosti spoje.

Druhy sdělovacích kanálů.
Kanál je souhrn prostředků pro přenos signálu od zdroje k příjemci. Diskrétní/spojité kanály jsou určeny pro přenos diskrétních/spojitých zpráv.
Odpovídá-li informační prvek přijatého signálu za všech okolností informačnímu prvku vyslaného signálu, hovoříme o bezhlukovém (bezšumovém) kanálu. V opačném případě připouštíme výskyt chyb při přenosu, pak se jedná o hlukový (šumový) kanál.
Diskrétní kanál bez paměti má tu vlastnost, že výsledek přenosu znaku ze vstupu na výstup nezávisí na předchozích znacích na vstupu. V opačném případě jde o kanál s pamětí.
Jestliže přenosové vlastnosti kanálu jsou časově nezávislé, je kanál stacionární, jinak je nestacionární.
V dalším se budeme až na výjimky zabývat diskrétními stacionárními kanály bez paměti, které reprezentují dostatečně přesný a přitom jednoduchý model některých používaných sdělovacích kanálů.

Model diskrétního sdělovacího kanálu [4].

Vstup kanálu je množina znaků $\{X\}$ spolu s jejich pravděpodobnostmi výskytu, které je třeba kanálem přenést.
Výstup kanálu je množina znaků $\{Y\}$ spolu s jejich pravděpodobnostmi výskytu, které získává příjemce.
Pokud jsou množiny $\{X\}$ a $\{Y\}$ dvouprvkové, jedná se o binární kanál.
Informační schéma binárního hlukového kanálu je na obr. 6 a). Jednotlivé podmíněné pravděpodobnosti mají následující význam:

$P(y_1 / x_1)$ .. pravděpodobnost přijetí znaku y_1, byl-li vyslán znak x_1 (správně přenesený znak x_1)
$P(y_2 / x_2)$ .. pravděpodobnost přijetí znaku y_2, byl-li vyslán znak x_2 (správně přenesený znak x_2)
$P(y_2 / x_1)$ .. pravděpodobnost přijetí znaku y_2, byl-li vyslán znak x_1 (chybně přenesený znak x_1)

$P(y_1 / x_2)$ .. pravděpodobnost přijetí znaku y_1, byl-li vyslán znak x_2 (chybně přenesený znak x_2)

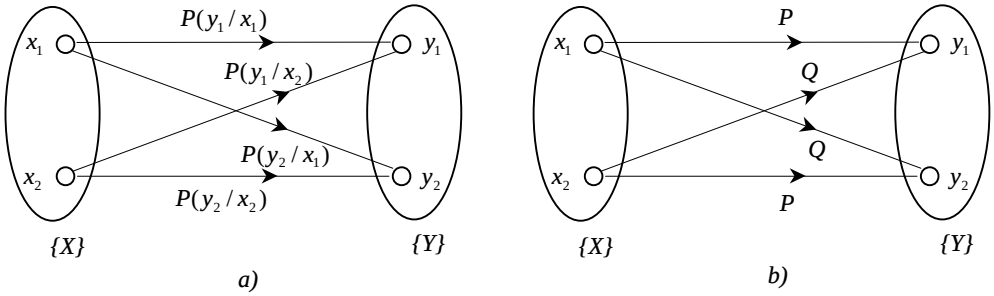
Pravděpodobnosti jsou vázány výrazy

$$P(y_1 / x_1) + P(y_2 / x_1) = 1 \tag{21}$$

(vyšleme-li x_1, pak přijmeme buď y_1 nebo y_2),

$$P(y_2 / x_2) + P(y_1 / x_2) = 1 \tag{22}$$

(vyšleme-li x_2, pak přijmeme buď y_1 nebo y_2).



Obr. 6. Informační model binárního hlukového kanálu, a) obecného bez paměti, b) symetrického.

Z těchto pravděpodobností lze sestavit (přímou) matici kanálu

$$\mathbf{K}_{XY} = \begin{pmatrix} P(y_1 / x_1) & P(y_2 / x_1) \\ P(y_1 / x_2) & P(y_2 / x_2) \end{pmatrix} \tag{23}$$

Z (21) a (22) vyplývá pravidlo, že součet prvků každého řádku matice kanálu je jednička.
S rostoucím vlivem poruch na přenos budou klesat pravděpodobnosti správně přenesených symbolů v hlavní diagonále matice kanálu a současně růst pravděpodobnosti chybných přenosů. Dané pravděpodobnosti jsou určeny jednak šumovými poměry při přenosu (odstupem výkonu vysílaného signálu a šumu), jednak technickými parametry kanálu (anténní systém, způsob modulace, citlivost a princip přijímače…). Prvky matice se dají stanovit experimentálně statistickým vyhodnocováním správně, resp. nesprávně přenesených znaků x_1 a x_2.
Jestliže platí

$$P(y_1 / x_1) = P(y_2 / x_2) = P\,,\; P(y_1 / x_2) = P(y_2 / x_1) = Q = 1 - P\,, \tag{24}$$

pak hovoříme o symetrickém kanálu (viz obr. 6 b). Pravděpodobnosti P budeme říkat spolehlivost kanálu, doplňkové pravděpodobnosti Q pak chybovost kanálu.
Známe-li pravděpodobnosti výskytů symbolů x_1 a x_2 na vstupu kanálu $P(x_1)$ a $P(x_2)$, pak můžeme určit pravděpodobnosti výskytu symbolů y_1 a y_2 na výstupu kanálu:

$$P(y_1) = P(x_1).P(y_1 / x_1) + P(x_2).P(y_1 / x_2)\,, \tag{25}$$

$$P(y_2) = P(x_2).P(y_2 / x_2) + P(x_1).P(y_2 / x_1)\,. \tag{26}$$

Součet dvou členů na pravé straně znamená, že daný symbol se může objevit na výstupu jako důsledek buď správného, nebo chybného přenosu. Každý z přenosů je podmíněn současným výskytem dvou náhodných jevů – jednak je to objevení se znaku na vstupu kanálu s danou pravděpodobností, jednak transformace znaku na výstup s danou podmíněnou pravděpodobností.

s Příklad 20: Symetrický binární kanál bez paměti, spolehlivost 95%. Binární znaky {0, 1} se na vstupu objevují s pravděpodobnostmi $P(x_1) = P(0) = 0{,}4$, $P(x_2) = P(1) = 0{,}6$. Vypočtěte pravděpodobnosti výskytu těchto znaků na výstupu kanálu.

$P = 0{,}95$, $Q = 0{,}05$,

$P(y_1) = 0{,}4.0{,}95 + 0{,}6.0{,}05 = 0{,}41$, … výskyt nuly na výstupu

$P(y_2) = 0{,}6.0{,}95 + 0{,}4.0{,}05 = 0{,}59$ … výskyt jedničky na výstupu.

Součet obou pravděpodobností musí dát jedničku, protože se jedná o vzájemně se vylučující náhodné jevy.

þ

Pravděpodobnosti v matici kanálu $\mathbf{K}_{XY}$ a odpovídající schéma na obr. 6 orientované ze vstupu na výstup popisují pozici pozorovatele na straně vstupu kanálu, který se snaží určit pravděpodobnosti výskytu znaků na výstupu, zná-li četnost vysílaných znaků. Podobně se dá kanál popsat z pohledu příjemce informace, který je na výstupu kanálu: má k dispozici pravděpodobnosti přijatých znaků a z nich se snaží usoudit pravděpodobnosti znaků vyslaných. Proto je možné pracovat s následujícími podmíněnými pravděpodobnostmi:

$P(x_1 / y_1)$ .. pravděpodobnost vyslání znaku x_1, byl-li přijat znak y_1 (správně přenesený znak x_1)
$P(x_2 / y_2)$ .. pravděpodobnost vyslání znaku x_2, byl-li přijat znak y_2 (správně přenesený znak x_2)
$P(x_1 / y_2)$ .. pravděpodobnost vyslání znaku x_1, byl-li přijat znak y_2 (chybně přenesený znak x_1)
$P(x_2 / y_1)$ .. pravděpodobnost vyslání znaku x_2, byl-li přijat znak y_1 (chybně přenesený znak x_2).

K výpočtu těchto pravděpodobností se použijí následující vztahy:

$$P(x_i, y_j) = P(x_i).P(y_j / x_i) = P(y_j).P(x_i / y_j)\,,\ i, j \in \{1,2\}. \tag{27}$$

Na levé straně (27) je pravděpodobnost současného výskytu jevů x_i a y_j (tzv. simultánní pravděpodobnost), na pravých stranách pak vždy součin pravděpodobností dvou dílčích jevů, které musí nastat současně.
Z rovnice (27) pak vyplývají hledané pravděpodobnosti:

$$P(x_i / y_j) = P(y_j / x_i)\frac{P(x_i)}{P(y_j)}\,,\ i, j \in \{1,2\}. \tag{28}$$

Pravděpodobnosti ve jmenovateli se určí pomocí vzorců (25) a (26). Je možné je zapsat do (zpětné) matice kanálu

$$\mathbf{K}_{YX} = \begin{pmatrix} P(x_1 / y_1) & P(x_1 / y_2) \\ P(x_2 / y_1) & P(x_2 / y_2) \end{pmatrix}. \tag{29}$$

Tentokrát platí, že součet pravděpodobností ve sloupcích je roven jedné.
Ukazuje se však, že prvky této matice (na rozdíl od matice $\mathbf{K}_{XY}$) už závisí na pravděpodobnostech výskytu znaků na vstupu x_1 a x_2. Po dosazení vzorců (25) a (26) do (28) a úpravě vyjdou konečné vzorce pro symetrický kanál:

$$P(x_1 / y_1) = \frac{1}{1 + \frac{P(x_2)}{P(x_1)}\frac{Q}{P}}\,,\ P(x_2 / y_2) = \frac{1}{1 + \frac{P(x_1)}{P(x_2)}\frac{Q}{P}}\,, \tag{30}$$

$$P(x_1 / y_2) = \frac{1}{1 + \frac{P(x_2)}{P(x_1)}\frac{P}{Q}}\,,\ P(x_2 / y_1) = \frac{1}{1 + \frac{P(x_1)}{P(x_2)}\frac{P}{Q}}\,.$$

s Příklad 21: Určete přímou a zpětnou matici symetrického binárního kanálu bez paměti o spolehlivosti 95% z příkladu 20. Uvažujte opět $P(x_1) = P(0) = 0{,}4$, $P(x_2) = P(1) = 0{,}6$.

$P = 0{,}95$, $Q = 0{,}05$,

$$\mathbf{K}_{XY} = \begin{pmatrix} P & Q \\ Q & P \end{pmatrix} = \begin{pmatrix} 0{,}95 & 0{,}05 \\ 0{,}05 & 0{,}95 \end{pmatrix}.$$

$$P(x_1 / y_1) = \frac{1}{1 + \frac{0,6}{0,4}\frac{0,05}{0,95}} \doteq 0,9268\,,\quad P(x_2 / y_2) = \frac{1}{1 + \frac{0,4}{0,6}\frac{0,05}{0,95}} \doteq 0,9661\,,$$

$$P(x_1 / y_2) = \frac{1}{1 + \frac{0,6}{0,4}\frac{0,95}{0,05}} \doteq 0,0339\,,\quad P(x_2 / y_1) = \frac{1}{1 + \frac{0,4}{0,6}\frac{0,95}{0,05}} \doteq 0,0732\,,$$

$$\mathbf{K}_{YX} \doteq \begin{pmatrix} 0,9268 & 0,0339 \\ 0,0732 & 0,9661 \end{pmatrix}.$$

þ

s Příklad 22: Na vstup symetrického binárního kanálu bez paměti o spolehlivosti 95% z příkladu 20 je vyslána binární zpráva. Pravděpodobnosti výskytu nul a jedniček jsou stejné a nezávisí na předchozích znacích. Vypočtěte entropii zprávy na vstupu a výstupu.

P = 0,95, Q = 0,05,

$P(x_1) = P(x_2) = 0,5$, $H(X)$ =1 Sh/znak.

$P(y_1) = P(x_1).P + P(x_2).Q = 0,5(P + Q) = 0,5$ (viz 25),
$P(y_2) = P(x_2).P + P(x_1).Q = 0,5(P + Q) = 0,5$ (viz 26).
$H(Y)$ = $H(X)$ =1 Sh/znak.

Zdálo by se, že jestliže zpráva vstupuje do hlukového kanálu, část jejího informačního obsahu se „ztratí“. Z příkladu však vyplývá, že množství informace je ve vstupní i výstupní zprávě stejné: stejně vyšly entropie, tj. průměrné množství informace na znak, a obě zprávy jsou stejně dlouhé.

Problém je v tom, že v důsledku šumu došlo sice k poklesu množství informace při přenosu, hlukový kanál však doplnil do zprávy shodou okolností stejné množství „dezinformace“. Informační obsahy vstupní a výstupní zprávy se jeví jako stejné. Uživatele však zajímá jen množství informace skutečně přenesené od zdroje k příjemci, tzv. vzájemná informace. Její výpočet si usnadníme pomocí tzv. podmíněných a simultánních entropií.

þ

Podmíněné a simultánní entropie [4,5,8,9,10].

Uvažujme hlukový kanál na obr. 6 a). Výskyt znaků na vstupu kanálu můžeme popsat úplným souborem vzájemně se vylučujících náhodných jevů $\{X\}$, výskyt znaků na výstupu pak popisujeme obdobně souborem $\{Y\}$.
Je-li kanál bezhlukový, tj. P = 1, Q = 0, pak soubory $\{X\}$ a $\{Y\}$ mají zcela totožné pravděpodobnostní charakteristiky. Naopak při $P = Q$ = 0,5, kdy se znak přenese na výstup správně nebo chybně se stejnou pravděpodobností, je kanál pro přenos zcela nepoužitelný (příjem jedniček a nul nesouvisí s tím, co je vysíláno, a je možno jej nahradit např. házením mincí). Pak soubory $\{X\}$ a $\{Y\}$ jsou statisticky nezávislé.
Známe-li pravděpodobnostní rozložení v souborech $\{X\}$ a $\{Y\}$, je možné spočítat entropie těchto souborů. V případě statistické závislosti mezi těmito soubory je však užitečné definovat další druhy entropií, které toto propojení modelují.
Podmíněná entropie vstupního souboru $\{X\}$ při známém výstupu y_j:

$$H(X / y_j) = -\sum_i P(x_i / y_j) \log_2 P(x_i / y_j)\,. \qquad (31)$$

Neurčitost příjemce na výstupu kanálu o tom, co je vysláno do vstupu, je $H(X)$. Přečte-li si však výstupní znak a zjistí, že výstup je y_j, pak tato neurčitost je zmenšena na hodnotu $H(X/y_j)$. V případě bezhlukového kanálu je dokonce neurčitost zmenšena na nulovou hodnotu.
Zprůměrujeme-li entropie (31) pro všechny možné výstupy y_j, dostaneme tzv. podmíněnou entropii vstupu po čtení výstupu:

$$H(X/Y) = \sum_j P(y_j) H(X/y_j) . \tag{32}$$

Po dosazení (31) do (32) a s využitím vztahu (27) dostaneme

$$H(X/Y) = -\sum_i \sum_j P(x_i, y_j) \log_2 P(x_i/y_j) . \tag{33}$$

K výpočtu této entropie tedy potřebujeme mj. znát prvky zpětné matice kanálu. Konkrétně pro binární kanál vede dvojitá suma v (33) na čtyři členy:

$$H(X/Y) = -[P(x_1,y_1)\log_2 P(x_1/y_1) + P(x_1,y_2)\log_2 P(x_1/y_2) + P(x_2,y_1)\log_2 P(x_2/y_1) + \\ + P(x_2,y_2)\log_2 P(x_2/y_2)]. \tag{34}$$

Podmíněná entropie vstupu po čtení výstupu představuje průměrnou neurčitost pozorovatele o stavu vstupu kanálu po přečtení výstupu. Před přečtením byla jeho neurčitost $H(X)$. Rozdíl těchto hodnot je tedy informace o vstupu, kterou pozorovatel získal čtením a která „prošla“ kanálem k příjemci. Hovoříme o vzájemné informaci ze vstupu na výstup $I(X,Y)$:

$$I(X,Y) = H(X) - H(X/Y) \text{ [Sh/znak]}. \tag{35}$$

Podmíněnou entropii $H(X/Y)$ lze chápat jako průměrné množství informace na znak zprávy, které se „ztratilo“ na cestě od vstupu kanálu k příjemci.

s Příklad 23: Na vstup symetrického binárního kanálu bez paměti o spolehlivosti 95% z příkladu 22 je vyslána binární zpráva. Pravděpodobnosti výskytu nul a jedniček jsou stejné a nezávisí na předchozích znacích. Entropie zprávy na vstupu i výstupu jsou 1 Sh/znak (viz Př. 22). Vypočtěte průměrné množství informace na znak zprávy, které se „ztratí“ na cestě od vstupu kanálu k příjemci, a vzájemnou vstupně-výstupní informaci.

Z příkladu 22 převezmeme tyto mezivýsledky:
$P = 0{,}95$, $Q = 0{,}05$,
$P(x_1) = P(x_2) = P(y_1) = P(y_2) = 0{,}5$, $H(X) = H(Y) = 1$ Sh/znak.
Nejprve vypočteme simultánní pravděpodobnosti $P(x_i, y_j)$ pomocí vzorce (27). Budeme je potřebovat pro výpočet $H(X/Y)$ podle (34).

$$P(x_1,y_1) = P(x_1).P(y_1/x_1) = 0{,}5.0{,}95 = 0{,}475 ,$$

$$P(x_1,y_2) = P(x_1).P(y_2/x_1) = 0{,}5.0{,}05 = 0{,}025 ,$$

$$P(x_2,y_1) = P(x_2).P(y_1/x_2) = 0{,}5.0{,}05 = 0{,}025 ,$$

$$P(x_2,y_2) = P(x_2).P(y_2/x_2) = 0{,}5.0{,}95 = 0{,}475 .$$

Nyní můžeme určit pravděpodobnosti ze zpětné matice kanálu podle (28):

$$P(x_1/y_1) = P(y_1/x_1)\frac{P(x_1)}{P(y_1)} = 0{,}95 ,$$

$$P(x_1/y_2) = P(y_2/x_1)\frac{P(x_1)}{P(y_2)} = 0{,}05 ,$$

$$P(x_2/y_1) = P(y_1/x_2)\frac{P(x_2)}{P(y_1)} = 0{,}05 ,$$

$$P(x_2/y_2) = P(y_2/x_2)\frac{P(x_2)}{P(y_2)} = 0{,}95 .$$

Dosadíme do (34) a určíme podmíněnou entropii:

$$H(X/Y) = -[P(x_1,y_1)\log_2 P(x_1/y_1) + P(x_1,y_2)\log_2 P(x_1/y_2) + P(x_2,y_1)\log_2 P(x_2/y_1) +$$
$$+ P(x_2,y_2)\log_2 P(x_2/y_2)] = -[0{,}475\log_2 0{,}95 + 0{,}025\log_2 0{,}05 + 0{,}025\log_2 0{,}05 + 0{,}475\log_2 0{,}95] \approx$$
$$\approx 0{,}286 \text{ Sh/znak.}$$

Vzájemná informace ze vstupu na výstup vyjde podle (35)

$$I(X,Y) = H(X) - H(X/Y) \approx 1 - 0{,}286 = 0{,}714 \text{ Sh/znak.}$$

þ

Podmíněnou entropii $H(X/Y)$ jsme zavedli jako určitou neurčitost příjemce, který je na výstupu kanálu a podle toho, co přijme, se snaží uhodnout, co bylo vysláno. Na celý problém se můžeme ale dívat i z pozice vysílače informace na vstupu kanálu.

Podmíněná entropie výstupního souboru $\{Y\}$ při známém vstupu x_i:

$$H(Y/x_i) = -\sum_j P(y_j/x_i)\log_2 P(y_j/x_i). \qquad (36)$$

Neurčitost pozorovatele na vstupu kanálu o tom, co je přijato na výstupu, je $H(Y)$. Zjistí-li však, že byl vyslán znak x_i, pak tato neurčitost je zmenšena na hodnotu $H(Y/x_i)$. V případě bezhlukového kanálu je dokonce neurčitost zmenšena na nulovou hodnotu.
Zprůměrujeme-li entropie (36) pro všechny možné vstupy x_i, dostaneme tzv. podmíněnou entropii výstupu po čtení vstupu:

$$H(Y/X) = \sum_i P(x_i)H(Y/x_i). \qquad (37)$$

Po dosazení (36) do (37) a s využitím vztahu (27) dostaneme

$$H(Y/X) = -\sum_i \sum_j P(x_i,y_j)\log_2 P(y_j/x_i). \qquad (38)$$

K výpočtu této entropie tedy potřebujeme nyní znát kromě simultánních pravděpodobností jen prvky přímé matice kanálu. Konkrétně pro binární kanál vede dvojitá suma v (38) na čtyři členy:

$$H(Y/X) = -[P(x_1,y_1)\log_2 P(y_1/x_1) + P(x_1,y_2)\log_2 P(y_2/x_1) + P(x_2,y_1)\log_2 P(y_1/x_2) +$$
$$+ P(x_2,y_2)\log_2 P(y_2/x_2)]. \qquad (39)$$

Podmíněná entropie výstupu po čtení vstupu představuje průměrnou neurčitost pozorovatele o stavu výstupu kanálu po přečtení vstupu. Před přečtením byla jeho neurčitost $H(Y)$. Rozdíl těchto hodnot je tedy informace o výstupu, kterou pozorovatel získal čtením vstupu a která „prošla zpět“ kanálem z výstupu k příjemci. Nazvěme ji vzájemnou informací z výstupu na vstup $I(Y,X)$:

$$I(Y,X) = H(Y) - H(Y/X) \text{ [Sh/znak].} \qquad (40)$$

Podmíněnou entropii $H(Y/X)$ lze chápat jako průměrné množství informace na znak zprávy, které se do výstupní zprávy dostalo rušivým působením kanálu a které nemá původ ve vstupní zprávě. Pro pozorovatele, který na stav výstupu usuzuje pozorováním vstupu, je tedy tato informační složka nedosažitelná (odečítá se od $H(Y)$ v rovnici (40)).

Simultánní entropie vstupního a výstupního souboru:

$$H(X,Y) = -\sum_i \sum_j P(x_i,y_j)\log_2 P(x_i,y_j) \qquad (41)$$

Je to neurčitost pozorovatele, který pozoruje „nad kanálem“ zároveň vstupy i výstupy, o stavech těchto vstupů a výstupů.
Pokud by vstupy a výstupy byly statisticky nezávislé (tj. kanál by „nefungoval“, $P = 0{,}5$), pak by se tato neurčitost rovnala součtu dílčích entropií vstupu a výstupu. Při statistické závislosti tato neurčitost klesá: například při bezhlukovém kanálu ($P = 1$) by se simultánní entropie rovnala entropii vstupu a zároveň entropii výstupu. Platí tedy nerovnost

$$MIN\{H(X), H(Y)\} \le H(X,Y) \le H(X) + H(Y) \qquad (42)$$

Simultánní entropie se však dá přesně určit pomocí entropií podmíněných:

$$H(X,Y) = H(X) + H(Y/X) \tag{43}$$

Neurčitost pozorovatele o stavech *X* a *Y* se dá snížit odečtením vstupu, tj. získáním informace *H*(*X*). To, co zbude, je neurčitost o stavu *Y* za podmínky, že jsme již odečetli vstup.

$$H(X,Y) = H(Y) + H(X/Y) \tag{44}$$

Neurčitost pozorovatele o stavech *X* a *Y* se dá snížit odečtením výstupu, tj. získáním informace *H*(*Y*). To, co zbude, je neurčitost o stavu *X* za podmínky, že jsme již odečetli výstup.

Odečtením rovnic (43) a (44) dostaneme výsledek

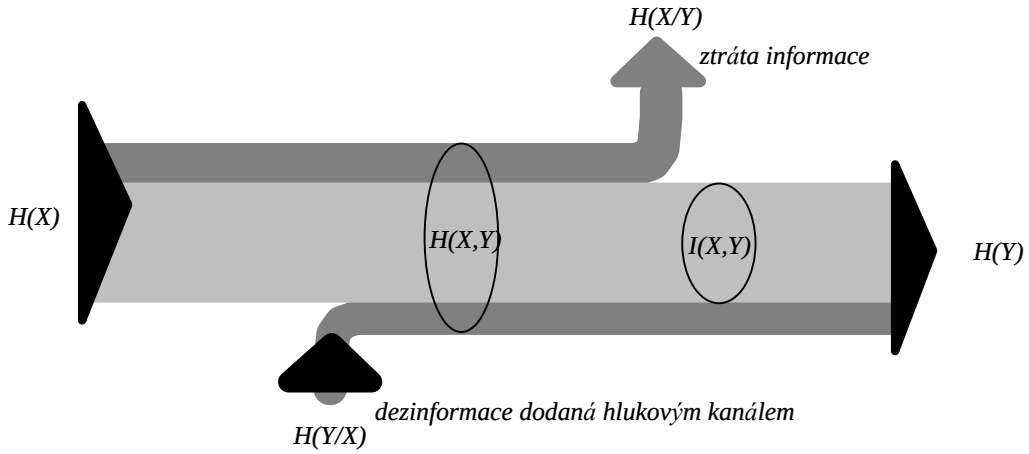
$$H(X) - H(X/Y) = H(Y) - H(Y/X), \tag{45}$$

neboli

$$I(X,Y) = I(Y,X). \tag{46}$$

Obě vzájemné informace jsou si číselně rovny. Tato symetrie se promítá i do jejich společného názvu – vzájemná vstupně-výstupní informace.

Vztah mezi entropiemi na vstupu a výstupu, podmíněnými entropiemi, simultánní entropií a vzájemnou vstupně-výstupní informací je znázorněn na obr. 7. Obrázek představuje syntézu vzorců (35), (40), (43), (44) a (45). Jsou z něj zřejmé i nerovnosti (42).



Obr. 7. Schéma informačních poměrů v hlukovém kanálu.

Porovnáme-li obr. 7 s výsledky příkladu 23, vidíme, proč navzdory poruchám v kanálu mohly být entropie na vstupu i výstupu stejné. Vzájemná informace, společná pro vstupní i výstupní zprávu, je však menší.

s Příklad 24: Vypočtěte vzájemnou vstupně-výstupní informaci a podmíněné entropie *H*(*Y*/*X*) a *H*(*X*/*Y*) pro binární symetrický kanál bez paměti o spolehlivosti a) 100%, b) 50%, c) 0% za předpokladu stejných pravděpodobností výskytu symbolů na vstupu.

Opakujeme-li postup z příkladu 23, vyjde:

Pro všechny tři případy platí $P(x_1) = P(x_2) = P(y_1) = P(y_2) = 0{,}5$, $H(X) = H(Y) = 1$ Sh/znak,

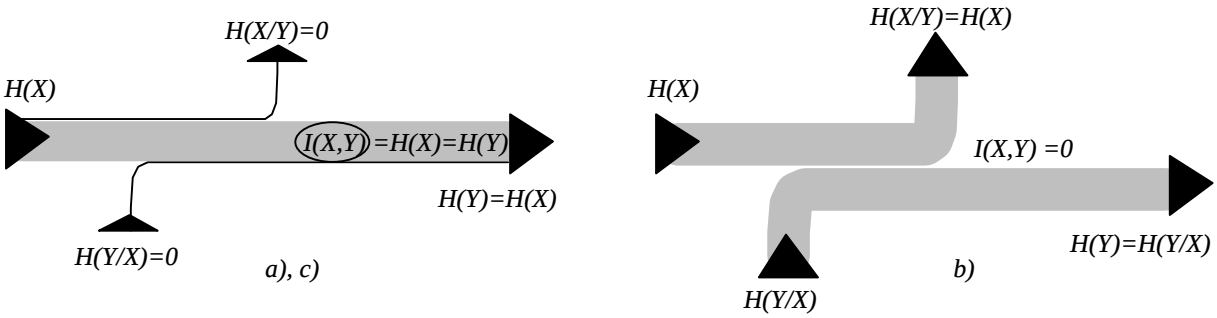
$$I(X,Y) = 1 + P\log_2 P + Q\log_2 Q, \tag{47}$$

$$H(X/Y) = H(Y/X) = -P\log_2 P - Q\log_2 Q. \tag{48}$$

a) $I(X,Y) = 1$ Sh/znak, $H(X/Y) = H(Y/X) = 0$.

b) $I(X,Y) = 0$, $H(X/Y) = H(Y/X) = 1$ Sh/znak.

c) $I(X,Y) = 1$ Sh/znak, $H(X/Y) = H(Y/X) = 0$.



Obr. 8. Informační poměry v kanálu z příklad 24 pro spolehlivost kanálu a) 100%, b) 50%, c) 0%.

Totožnost informačních schémat a) a c) je zarážející jen na první pohled. Případ c) znamená, že při vyslání jedničky je vždy přijata nula a naopak. Kanál má tedy stoprocentní spolehlivost, jen je třeba uvažovat „inverzní“ kód.

Pro případ b) není žádná statistická souvislost mezi vstupem a výstupem, čemuž odpovídá nulová vzájemná informace. Kanál je tedy pro přenos informace neprůchodný. Můžeme si ověřit, že nula vyjde při libovolném rozložení pravděpodobností $P(x_1)$ a $P(x_2)$.

Þ

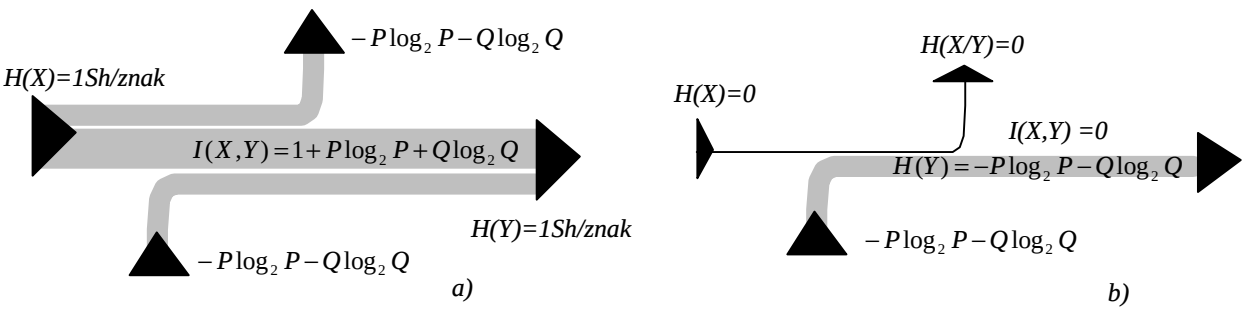
s Příklad 25: Vypočtěte entropii na vstupu a výstupu a vzájemnou vstupně-výstupní informaci pro binární symetrický kanál bez paměti o spolehlivosti P, jestliže jedničky a nuly se objevují na vstupu s pravděpodobnostmi a) stejnými, b) $P(1) = 1$, $P(0) = 0$.

Výsledky:

a) $H(X) = H(Y) = 1$ Sh/znak, $I(X,Y) = 1 + P\log_2 P + Q\log_2 Q$,

$H(X/Y) = H(Y/X) = -P\log_2 P - Q\log_2 Q$.

b) $H(X) = 0$, $H(Y) = -P\log_2 P - Q\log_2 Q$, $I(X,Y) = 0$.



Obr. 9. Informační poměry v kanálu o spolehlivosti P z příkladu 25, rozložení pravděpodobností výskytu znaků na vstupu je a) rovnoměrné, b) deterministické.

Při rovnoměrném rozložení pravděpodobností na vstupu (a) je entropie na vstupu maximální možná, kdežto při deterministickém rozložení (b) je nulová. Entropie na výstupu je tak v případě (b) nenulová díky chybám při přenosu. Vzájemná informace musí být nulová, protože není co přenášet: informační obsah vstupní abecedy je nulový.

Þ

Kapacita kanálu [4,5,8,9,10].

V předchozích příkladech bylo ukázáno, že velikost vzájemné informace $I(X,Y)$ procházející kanálem o dané spolehlivosti závisí na pravděpodobnostním rozložení vstupních znaků. To zase závisí na způsobu kódování před kanálem.

Kapacita kanálu C je maximální možná vzájemná informace, kterou můžeme „protlačit“ tímto kanálem při uvažování všech možných statistických vlastností vstupních informačních posloupností. Pro stacionární diskrétní kanál bez paměti stačí hledat maximum vzájemné informace pro různá rozložení pravděpodobnosti vstupních znaků:

$$C = \max_{P(x_i)} \{I(X,Y)\}. \tag{49}$$

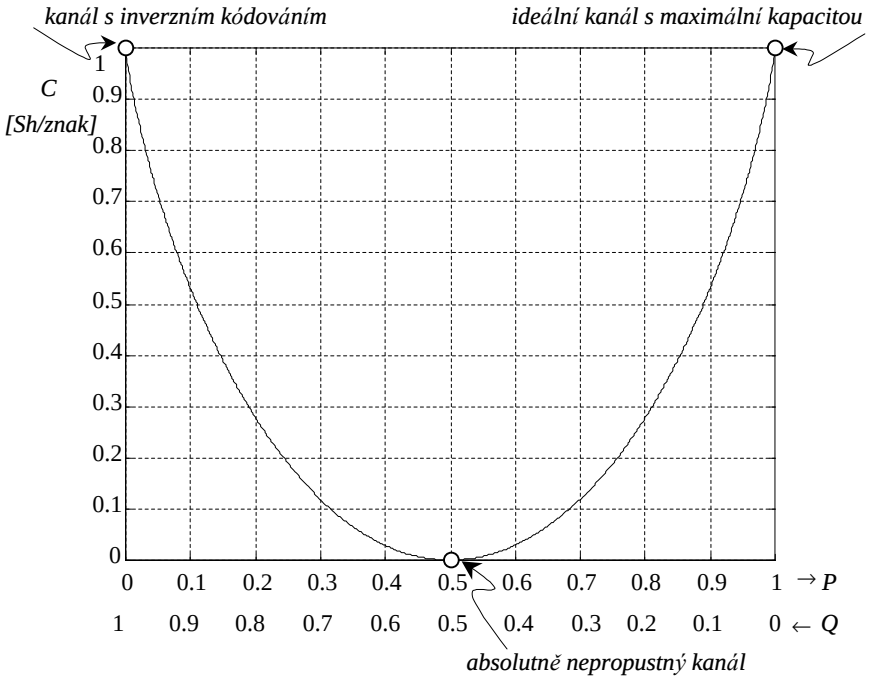
K tomuto maximu dochází při rovnoměrném rozdělení pravděpodobností (důkaz je možné nalézt např. v [5]). V případě symetrického binárního kanálu pak musí být proto kapacita dána vzorcem (47):

$$C = 1 + P\log_2 P + Q\log_2 Q \text{ [Sh/znak]}. \tag{50}$$

Je-li informační znak realizován signálovým prvkem délky T_S, je možné vyjádřit kapacitu kanálu v Sh/s jako množství informace přenesené za jednotku času:

$$C' = \frac{C}{T_S} \text{ [Sh/s]}. \tag{51}$$

Zde vyniká „nešikovnost“ používané jednotky Shannon – veličina C‘ je běžně chápána jako přenosová rychlost v bitech za sekundu (viz též příklad 18, týkající se vztahu modulační rychlosti, klasicky pojaté přenosové rychlosti a entropie zdroje).



Obr. 10. Závislost kapacity symetrického binárního kanálu na jeho spolehlivosti.

Grafické znázornění vzorce (50) je na obr. 10. Maximální možná kapacita binárního kanálu je 1 Sh/znak a dochází k ní při stoprocentní spolehlivosti kanálu. Při spolehlivosti 50% je kanál nepropustný. Další snižování spolehlivost směrem k nule však vede na růst kapacity až k maximu 1 Sh/znak v důsledku „inverzního kódování“, které bylo popsáno v příkladu 24.
Připomeňme, že spolehlivost P jakožto pravděpodobnost správného přenesení znaku kanálem závisí na řadě faktorů, m.j. na poměru signál-šum u rádiových kanálů.

s Příklad 26: Telefonní signál je modulací PCM přenášen symetrickým binárním kanálem. Je požadována chybovost přenosu BER ≤ 10^{-4}. Navrhněte kapacitu kanálu.

BER (Bit Error Rate) je možné pro tento případ ztotožnit s chybovostí kanálu Q:

$Q = 10^{-4}$,

$C \geq 1 + (1\text{-}10^{-4}) \log_2(1\text{-}10^{-4}) + 10^{-4} \log_2 10^{-4} \approx 0{,}9985$ Sh/znak.

þ

Shannonův – Hartleyův vzorec pro kapacitu kanálu [5,8,9,10].

Binární symetrický kanál bez paměti je jeden z jednoduchých modelů reálných sdělovacích kanálů. Diskrétní kanál je zvláštní případ spojitého kanálu využívaného diskrétně v čase, kdy sice přenášíme signál opět jen v definovaných okamžicích, ale jeho hodnoty nejsou kvantovány.

Níže jsou uvedeny důležité vzorce pro kapacitu kanálu, které je možné za určitých podmínek aplikovat i na kanály diskrétní. Byl však odvozen za předpokladu, že na užitečný signál, který má normální rozdělení a konstantní výkonovou spektrální hustotu v kmitočtovém pásmu *B*, působí aditivní normální bílý šum:

$$C = \frac{1}{2}\log_2\left(1+\frac{S}{N}\right)\text{[Sh/vzorek]} \tag{52}$$

$$C' = B\log_2\left(1+\frac{S}{N}\right)\text{[Sh/s]} \tag{53}$$

Zde S/N je poměr výkonů užitečného signálu a šumu a *B* je šířka kmitočtového pásma kanálu v hertzích. Vzorkem se v (52) rozumí vzorek analogového signálu o výkonu *S* přenášený spojitým kanálem, případně odpovídající informační slovo reprezentované jedním ze signálových prvků tvořících signál o výkonu *S* přenášený kanálem diskrétním. Z (52) byl odvozen vzorec (53) na základě úvahy plynoucí z vzorkovací poučky, že je-li analogový signál omezen kmitočtovým pásmem *B*, je možno jej reprezentovat beze ztráty informace posloupností jeho 2*B* vzorků za sekundu.

Vzorce popisují horní teoretickou hranici kapacity kanálu, ke které je možno se limitně blížit vhodným kódováním při daných výkonech signálu a šumu, a to při chybovosti kanálu libovolně se blížící k nule. Nedávají však návod, jak toto kódování provést. Navíc je třeba pamatovat na to, že u skutečných sdělovacích kanálů nejsou přesně splněny předpoklady, na základě nichž bylo provedeno odvození (šum není normální, může být multiplikativní apod.). Například u binárních kanálů nemá užitečný signál charakter náhodného signálu s normální rozdělením.

Požadovanou kapacitu kanálu je možné dosáhnou různými kombinacemi parametrů *S*, *N* a *B*. Pro tradiční radiokomunikační systémy je charakteristická volba vysokých poměrů signál/šum (vysoké vysílací výkony). Pak si můžeme dovolit relativně malé šířky pásma. Opačným přístupem jsou malé vysílací výkony, mnohdy pod úrovní šumu, a velká šířka pásma. To je charakteristické pro perspektivní komunikační systémy s rozprostřeným spektrem.

s Příklad 27: Uvažujte spojitý kanál s šířkou pásma 4 kHz pro přenos telefonních signálů. Odhadněte maximální dosažitelnou kapacitu kanálu, je-li poměr signál-šum 30 dB.

$C' = 4000\log_2(1+1000) \approx 39{,}9$ kSh/s (rozuměj 39,9 kbit/s).

þ

Z (52) a (53) vyplývá, že při růstu výkonu užitečného signálu $S \to \infty$ by se měla neohraničeně zvětšovat i kapacita kanálu (viz obr. 11). U diskrétních kanálů však nemá užitečný signál normální rozložení, takže je porušen předpoklad, na němž jsou vzorce založeny. Důsledkem je např. to, že u binárního kanálu existuje konečná limitní hodnota kapacity při zvětšování výkonu signálu.

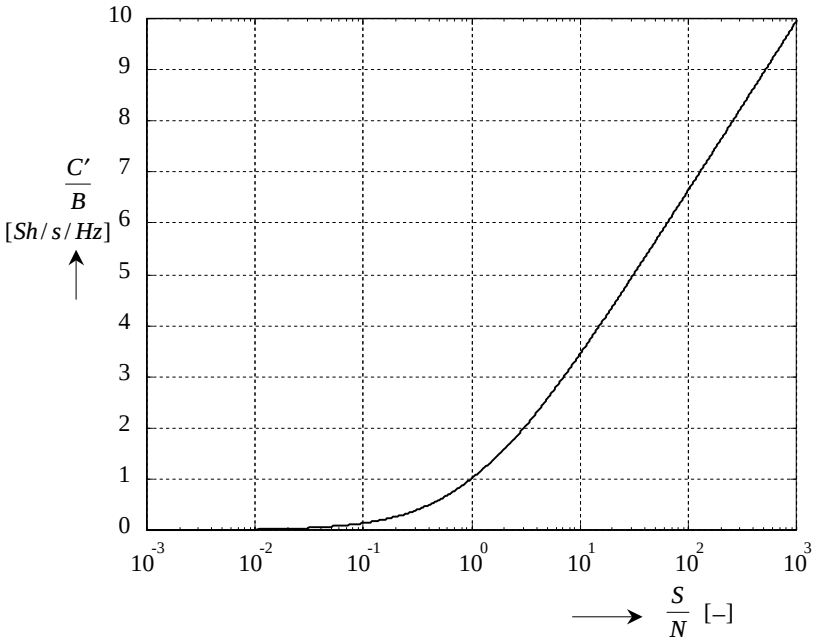
s Příklad 28: Vypočtěte maximální dosažitelnou kapacitu symetrického binárního kanálu bez paměti [10].

Zvětšováním výkonu binárního signálu můžeme limitně zvětšovat spolehlivost kanálu až k jejímu maximu:

$S \to \infty \Rightarrow P \to 1, Q \to 0$.

Pak podle (50) $C \rightarrow 1$ Sh/znak $\Rightarrow C' \rightarrow 2B$ bit/s. Například k dosažení přenosové rychlosti cca 40kbit/s bychom potřebovali binární kanál o minimální šířce pásma 20kHz (u spojitého kanálu stačí 4 kHz – viz příklad 27).

þ



Obr. 11. Závislost kapacity kanálu na výkonu užitečného signálu.

Z vzorce (53) by mohlo na první pohled vyplývat, že kapacitu kanálu mohu neomezeně zvětšovat zvětšováním šířky pásma. Ve skutečnosti je to složitější, protože s růstem B roste i výkon šumu N. Normální bílý šum má konstantní spektrální výkonovou hustotu G, takže mezi šířkou pásma a výkonem šumu platí přímá úměra (za předpokladu konstantní kmitočtové charakteristiky kanálu v pásmu propustnosti)

$$N = B.G\,. \tag{54}$$

Dosadíme-li (54) do (53), dostaneme po úpravě

$$\frac{C'}{S/G} = \frac{\log_2\left(1+\frac{S/G}{B}\right)}{\frac{S/G}{B}} \rightarrow \ln 2 \approx 1{,}443 \text{ pro } B \rightarrow \infty \left(\lim_{x\rightarrow 0}\frac{\log_2(1+x)}{x} = \ln 2\right), \text{ takže}$$

$$C' \rightarrow 1{,}443\frac{S}{G} \text{ při } B \rightarrow \infty\,. \tag{55}$$

Z obrázku 12 a vzorce (55) plyne, že zvyšováním šířky pásma lze zvětšovat kapacitu kanálu jen do určité míry. Dalšího zvětšení mohu dosáhnout zvětšením výkonu užitečného signálu.
Z obr. 12 je zřejmé, že klasické přenosové systémy s velkým odstupem signál – šum zdaleka nedosahují teoretického maxima kapacity. K tomuto maximu se naopak blíží systémy s rozprostřeným spektrem, pro něž je charakteristická nerovnost $S < N$.

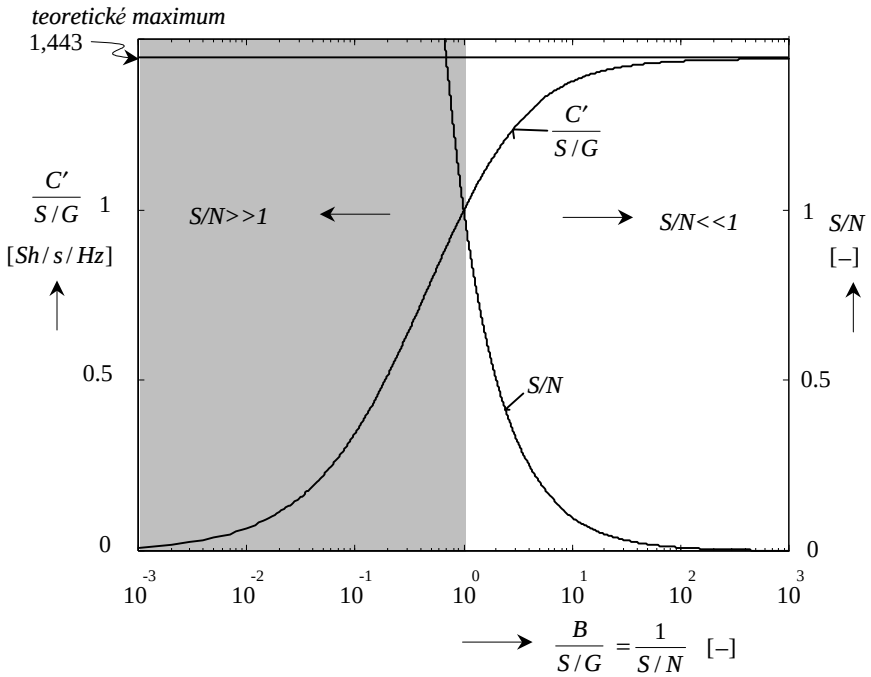
Shannonova věta o kódování v šumovém kanále [5]
V šumovém kanále s kapacitou C můžeme pro zdroj informace s entropií H za předpokladu
$H < C$
nalézt takový způsob kódování dlouhých zpráv, že pravděpodobnost výskytu chyby P_e je menší než předem dané libovolně malé číslo $\varepsilon > 0$, avšak nikoliv rovna nule.

Shannonova obrácená věta o kódování v šumovém kanále

V šumovém kanále s kapacitou C nelze pro zdroj informace s entropií H za předpokladu

$$H > C$$

nalézt takový způsob kódování dlouhých zpráv, aby pravděpodobnost výskytu chyby P_e byla menší než předem dané libovolně malé číslo ε>0.



Obr. 12. Závislost kapacity kanálu na jeho šířce pásma a poměru signál-šum [2].

Shannonovy věty ukazují, že jakýkoliv „špatný“ (šumový) kanál je možno vhodným kódováním zprávy „vylepšit“ tak, že stlačíme chybovost na libovolně malou hodnotu. To však platí za předpokladu, jestliže nebudeme požadovat přenášení většího množství informace, než je kapacita kanálu. Při překročení této kapacity chybovost prudce poroste.

Uvedené věty neposkytují návod na optimální kódování zpráv. V dalším textu ukážeme některé možnosti využití kódů ke kompresi a zabezpečení zpráv.

KÓDOVÁNÍ V SYSTÉMECH PRO PŘENOS INFORMACE

Klasifikace kódů podle typu aplikace.
Kódování plní v různých místech sdělovacího řetězce různorodé funkce. Od toho se odvíjí následující možná klasifikace kódů:

Kódy pro přizpůsobení informačního zdroje na kanál.
Používají se při změně syntaktické abecedy zprávy. Úkolem těchto kódů je zároveň maximalizace entropie abecedy před vstupem zprávy do přenosového kanálu.
Kódy pro snižování nadbytečnosti a kompresi dat.
Úkolem je „zhuštění" zprávy snížením její redundance tak, aby byl její přenos co nejefektivnější.
Kódy pro zabezpečení dat proti chybám.
Informační slova se zabezpečují korekčním nebo detekčním kódem s cílem zajištění definované chybovosti spoje.
Šifrovací kódy.
Takové zabezpečení zprávy, aby představovala zdroj informace pouze pro oprávněné příjemce.
Kódy pro zrovnoměrnění spektra datových signálů (skramblování…).
Při přenosu dat analogovým médiem, např. telefonním kabelem, je důležité efektivní využití jeho kmitočtové šířky pásma. Kvaziperiodický charakter přenášeného signálu však způsobuje, že spektrum je shlukováno zejména v okolí určitých kmitočtů a kanál je tak využíván neefektivně. Tomu lze odpomoci *skramblováním*, kdy se posiluje náhodný charakter signálu jeho slučováním s pseudonáhodnou posloupností. Spektrum přenášeného signálu se pak zrovnoměrní. V přijímači se pak za demodulátorem provede inverzní proces – deskramblování za účelem získání původních dat.
Kódy pro zabezpečení procesu modulace v modemech (Trellis klíčování..).
Používají se různé speciální metody zabezpečení, které bojují jak proti tzv. osamoceným chybám, tak proti shlukům chyb. Některé metody: prokládání (interleaving), trellis klíčování, řetězové kódování.

Shannonovy teorémy zdrojového a kanálového kódování [2].
Z Shannonovy koncepce komunikačního kanálu vyplývá další klasifikace kódů na zdrojové a kanálové kódy. Příklad zdrojového kódu je kompresní kód, kanálové kódy jsou např. kódy bezpečnostní. Poznatky z předchozích kapitol je možné slovně shrnout do dvou teorémů. První z nich říká, že vhodným kódováním je možné „zhustit" každou zprávu tak, že se její informační obsah může libovolně přiblížit její teoretické informační kapacitě podle Hartleye. Druhý teorém je vlastně Shannonova věta o kódování v šumovém kanále.

Teorém zdrojového kódování: Počet bitů, nezbytných k jednoznačnému popisu určitého zdroje dat, se může vhodným kódováním blížit k odpovídajícímu informačnímu obsahu tak těsně, jak je požadováno.
Teorém kanálového kódování: Frekvence výskytu chyb v datech přenášených pásmově omezeným kanálem se šumem může být vhodným kódováním dat redukována na libovolně malou hodnotu, pokud je rychlost přenosu informace menší než činí kapacita přenosového kanálu.

V dalším textu se budeme zabývat metodami zdrojového a posléze kanálového kódování.

Zdrojové kódování - kódy pro snižování nadbytečnosti.

Kódování jako předpis.

V úvodu definujme pojem kódování. Předpokládejme následující struktury:
A .. množina znaků tzv. primární (zdrojové) abecedy.

Posloupnost znaků primární abecedy tvoří primární (zdrojovou) zprávu, kterou je třeba kódovat, např. DOBRY_DEN.

B.. množina znaků tzv. sekundární (kódové) abecedy. Znakům sekundární abecedy se též říká kódové znaky. Příkladem je binární kódová abeceda, kde B={0 1}. Jak uvidíme následně, většinou se tyto znaky nevyskytují v sekundární zprávě samostatně, nýbrž pouze v povolených kombinacích (tzv. kódových slovech).

B* množina tzv. kódových slov. Kódové slovo, někdy též nazývané značka, je povolená posloupnost určitého počtu kódových znaků. Například pro binární abecedu může být množina kódových slov následující: B*={0 10 111}. Počet kódových znaků v kódovém slovu se nazývá délka kódového slova.

Definice kódování: Kódování je předpis, který každému prvku z množiny A přiřazuje právě jedno kódové slovo z množiny B*.
Kód *k* je tedy zobrazení
k: A $\rightarrow$ B*.
Jestliže je kódování v souladu s definicí prováděno jednoznačně, tzn. jestliže různým znakům primární abecedy přísluší různá kódová slova, pak říkáme, že kódování je prosté.

s Příklad 29: Prostý binární kód.

$A=\{a\ b\ c\ d\},\ B=\{0\,1\},\ B^*=\{00\ 01\ 10\ 11\}$ podle přiřazení

$$k:\ \begin{matrix} a & & 00 \\ b & \rightarrow & 01 \\ c & & 10 \\ d & & 11 \end{matrix}$$

V přiřazení se šipka často vynechává a celý zápis se provede formou tzv. kódovací tabulky.

þ

Praktická definice kódu.
V praxi se často definuje kód pouhým výčtem kódových slov z kódovací tabulky.

Blokové a ostatní kódy.
Blokový kód je kód, jehož všechna kódová slova mají stejnou délku (viz např. kód z příkladu 29).
Z ostatních kódů s nestejnými délkami kódových slov jsou pro zdrojové kódování nejdůležitější prefixové kódy.

Prefixové kódování.
Prefix („předpona“) je jeden z atributů kódového slova.
Uvažujme kódové slovo v tomto zápise posloupnosti kódových znaků:

$$b_1\ b_2\ b_3\ b_4\ \ldots\ b_{n-1}\ b_n$$

Toto kódové slovo má následující prefixy („předpony“):

b_1
$b_1\ b_2$
$b_1\ b_2\ b_3$
$b_1\ b_2\ b_3\ b_4$
$b_1\ b_2\ b_3\ b_4\ \ldots\ b_{n-1}$
$b_1\ b_2\ b_3\ b_4\ \ldots\ b_{n-1}\ b_n$

V posledním případě je kódové slovo samo sobě prefixem.

Definice: Kódování se nazývá prefixové, jestliže:
1. je prosté,
2. žádné kódové slovo není prefixem jiného kódového slova.

s Příklad 30: ternární (tříznakový) prefixový kód [4]

A	0
B	1
C	20
D	21
E	220
F	221

Vlastnost 2 umožňuje rychlé dekódování sekundární zprávy „zleva doprava znak po znaku“, i když kódová slova mají variabilní délku. Například zpráva 1122112211 představuje po dekódování původní zprávu BBFBFB.

þ

Vztah prefixového a blokového kódu.
Každý blokový kód je zároveň prefixovým kódem, neboť nemohou existovat dvě stejná kódová slova.
Prefixový kód ovšem nemusí být kódem blokovým, protože nemusí používat kódová slova stejné délky.

s Příklad 31: kód, který není ani blokový ani prefixový: vyjádření teploty vody binárním kódem [4].

ledová	0
studená	01
vlažná	011
teplá	111

Kód byl sestaven tak, aby stavy s vysokou pravděpodobností výskytu byly vyjádřeny krátkými kódovými slovy. Tím se docílí toho, že skutečné zprávy budou po zakódování relativně krátké.
Kód však není prefixový, protože kódové slovo 0 je prefixem slov 01 i 011 a slovo 01 je prefixem slova 011. Vzniká otázka, zda je zakódovaná zpráva vůbec jednoznačně dekódovatelná.
Zakódujeme-li zprávu „ledová, ledová, studená, ledová“, tj. 00010, pak dekódování je snadné.
Jak dekódovat např. zprávu 01111111? Dekódujeme-li zleva doprava, není jisté, zda prvním znakem je 0 nebo 01 nebo 011. Museli bychom prověřit všechny kombinace, což by ovšem prakticky znemožňovalo dekódování dlouhých zpráv.
Kódová slova by tvořila prefixový kód, kdybychom je četli „pozpátku“. Proto je dekódování snadné, postupujeme-li od konce zprávy na její začátek. Výsledek bude „studená, teplá, teplá“. To ale znamená, že dekódovat nelze průběžně během přijímání zprávy, nýbrž se musí počkat, až proběhne celý její přenos. Přitom by stačilo málo – „otočit“ kódová slova tak, abychom získali prefixový kód.

þ

Věta o jednoznačné dekódovatelnosti.
Kódování je jednoznačně dekódovatelné, jestliže ze znalosti zakódované zprávy lze vždy jednoznačně odvodit zdrojovou zprávu.

Poznámky:

- Věta pouze definuje pojem jednoznačné dekódovatelnosti. Nepodává návod, jak zjistit, zda je daný konkrétní kód jednoznačně dekódovatelný nebo ne.

- Srovnáme-li kód z příkladu 31 s prefixovým kódem, který z něj vznikne „otočením“ kódových slov, pak je zřejmé, že oba jsou jednoznačně dekódovatelné, ale… . V praxi jde tedy také o to, jak je kód dekódovatelný.
- Při kompresi dat se používají techniky „ztrátové“ komprese a „bezeztrátové“ komprese. Je zřejmé, že ztrátová komprese využívá kódování, kdy nelze zpětně jednoznačně určit původní zprávu před kódováním. V řadě případů (např. komprese videa nebo zvuku) to ale nemusí být na závadu. Komprese textových a binárních souborů však musí být bezeztrátová.
- Prosté kódování zajišťuje jednoznačnou dekódovatelnost pouze u blokových kódů, kde jsou všechna kódová slova stejně dlouhá.

s Příklad 32: Kódování cifer 0 až 9 binárním prefixovým kódem [4].

Zpráva je tvořena znaky 0 až 9. Přitom cifry 0 a 1 se vyskytují velmi často, cifry 8 a 9 jen zřídka. Je proto vhodné cifry 0 a 1 zakódovat kódovými slovy co nejkratšími, pro cifry 8 a 9 si můžeme dovolit vyčlenit kódová slova relativně nejdelší.

Příklad jednoho z prefixových kódů je uveden v kódovací tabulce:

A	B*	l
0	00	2
1	01	2
2	1000	4
3	1001	4
4	1010	4
5	1011	4
6	1100	4
7	1101	4
8	1110	4
9	1111	4

délka kódového slova

Cifry 0 a 1 nemůžeme kódovat jednomístnými kódovými slovy 0 a 1, protože všechna ostatní binární kódová slova obsahují nulu nebo jedničku jako svůj prefix. Z toho důvodu jsou pro nulu a jedničku použita kódová slova délky 2. Protože začínají nulou, musí všech ostatních 8 kódových slov začínat jedničkou. V tabulce je využito skutečnosti, že 8 různých kódových slov se dá vyjádřit osmi možnými kombinacemi tříbitových slov. Výsledný kód je přitom prefixový. Zřídka se vyskytující osmička a devítka jsou kódovány slovy o délce 4. Taktéž však cifry 2 až 7, takže vznikají otázky, zda by kódování nebylo efektivnější, kdyby některé z cifer 2 až 7 byly kódovány slovy délky 3 a cifry 8 a 9 slovy délky 5 nebo 6. K tomu bychom však asi potřebovali znát konkrétní pravděpodobnosti výskytu jednotlivých cifer a některé zákonitosti prefixového kódování, které si ukážeme dále.

þ

Věta o Kraftově nerovnosti.
Má-li sekundární abeceda n znaků, můžeme sestavit prefixový kód obsahující r kódových slov o délkách $l_1, l_2, .., l_r$, právě když platí Kraftova nerovnost

$$n^{-l_1} + n^{-l_2} + ... + n^{-l_r} \leq 1. \qquad (56)$$

s Příklad 33: Ověření Kraftovy nerovnosti pro prefixový kód z příkladu 32.

$n = 2$, $l_1 = l_2 = 2$, $l_3 = l_4 = \ldots = l_{10} = 4$,

$2.2^{-2} + 8.2^{-4} = 1.$

Kdybychom chtěli zakódovat cifry 2 až 7 pouze třímístnými slovy a cifry 8 a 9 pětimístnými, pak by vyšlo

$n = 2, l_1 = l_2 = 2, l_3 = l_4 = \ldots = l_8 = 3, l_9 = l_{10} = 5,$

$2.2^{-2} + 6.2^{-3} + 2.2^{-5} = 1{,}375 > 1.$

Kraftova nerovnost nyní neplatí, takže takovýto prefixový kód neexistuje a nemá cenu jej tedy hledat.

þ

Poznámka: platí-li pro určitý kód Kraftova nerovnost, neznamená to, že je tento kód prefixový. Např. pro kód z příkladu 31 (teplota vody) je Kraftova nerovnost splněna, přesto kód není prefixový.

s Příklad 34: Využití Kraftovy nerovnosti pro hledání „krátkých“ prefixových kódů.

Pokusíme se modifikovat kód z příkladu 32 tak, že prodloužíme délku kódových slov zřídka se vyskytujících cifer 8 a 9 na pět a budeme hledat co nejkratší kódová slova cifer 2 až 7 (v příkladu 32 mají jednotnou délku 4).

A	B*	*l*	*příspěvek do K.N.*
0	00	2	$2^{-2}+2^{-2}$ = = 0,5
1	01	2	
2	?	?	zbývá do jedničky 1-0,5--0,0625= =0,4375= =7.2^{-4}
3	?	?	
4	?	?	
5	?	?	
6	?	?	
7	?	?	
8	?	5	$2^{-5}+2^{-5}$ = = 0,0625
9	?	5	

Situace je znázorněna v neúplné kódovací tabulce, kde v posledním sloupci je uveden „příspěvek“ kódových slov k vytváření členů na levé straně Kraftovy nerovnosti (56). Vidíme, že na šest cifer 2 až 7 zbývá hodnota 7.2^{-4}, což reprezentuje 7 kódových slov délky 4. Při šesti kódových slovech délky 4 z příkladu 32 je zde ještě určitá rezerva vyjádřená číslem 1.2^{-4}.

Kdybychom zvolili n_3 slov délky 3 a ostatních $n_4 = 6 - n_3$ slov délky 4, vyšlo by

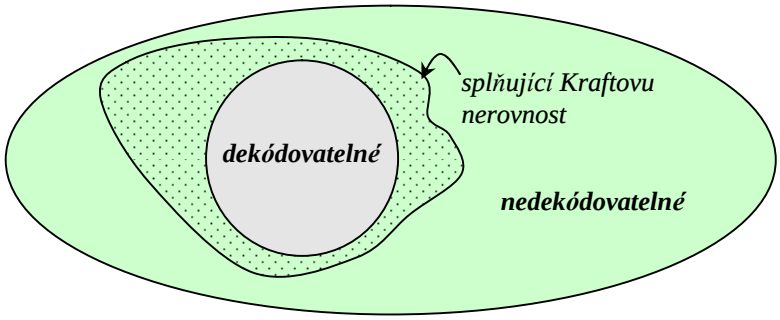
$n_3.2^{-3} + (6-n_3).2^{-4} \leq 7.2^{-4}$. Jediné kladné celočíselné řešení je $n_3 = 1$ a $n_4 = 5$. Zakódujeme tedy například cifru 2 třímístným slovem a cifry 3 až 7 čtyřmístnými slovy. Příklad takového kódu, který je „úspornější“ ve srovnání s původním kódem z příkladu 32, je zde:

A	B*	*d*
0	00	2
1	01	2
2	100	3
3	1010	4
4	1011	4
5	1100	4
6	1101	4
7	1110	4
8	11110	5
9	11111	5

þ

K tomu, abychom byli schopni rozhodnout, který z dvou kódů v příkladech 32 a 34 vede na kratší zprávy, neboli který lépe komprimuje zprávu, bychom potřebovali znát pravděpodobnosti výskytu cifer 0 až 9. Tuto úlohu vyřešíme později jako přípravu k tzv. Huffmanovu kódování.

Mc Millanova věta
Pro každé jednoznačně dekódovatelné kódování platí Kraftova nerovnost.



Pozor! Z McMillanovy věty opačně plyne, že:

- jestliže pro kód *neplatí* Kraftova nerovnost, pak *není jednoznačně dekódovatelný*.
- jestliže pro kód *platí* Kraftova nerovnost, pak *může, ale také nemusí být jednoznačně dekódovatelný*.

Kraftova nerovnost je tedy nutnou, nikoliv však postačující podmínkou jednoznačné dekódovatelnosti!

s Příklad 35: Využití McMillanovy věty k rozhodování o jednoznačné dekódovatelnosti kódů.

kód č. 1

A	B*	*l*
ledová	0	1
studená	01	2
vlažná	011	3
teplá	111	3

kód č. 2

A	B*	*l*
x	0	1
y	1	1
z	01	2

kód č. 3

A	B*	*l*
x	00	2
y	10	2
z	1	1

kód č. 4

A	B*	*l*
x	00	2
y	11	2
z	1	1

Kód č. 1: $2^{-1}+2^{-2}+2.2^{-3} = 1$, nelze rozhodnout o jednoznačné dekódovatelnosti.
Kód č. 2: $2.2^{-1}+2^{-2} = 1{,}25>1$, kód není jednoznačně dekódovatelný.
Kód č. 3: $2.2^{-2}+2^{-1} = 1$, nelze rozhodnout o jednoznačné dekódovatelnosti.
Kód č. 4: $2.2^{-1}+2^{-2} = 1$, nelze rozhodnout o jednoznačné dekódovatelnosti.

Ve skutečnosti není kromě kódu č. 2 jednoznačně dekódovatelný ještě kód č. 4, ostatní kódy jsou jednoznačně dekódovatelné.

þ

Průměrná délka značky prefixového kódu.
Bude záviset jak na délkách jednotlivých kódových slov, tak i na pravděpodobnostech jejich výskytu.
Uvažujme prefixový kód obsahující r kódových slov (značek) o délkách l_1, l_2, .., l_r s pravděpodobnostmi výskytu P_1, P_2, …, P_r. Pak průměrná délka značky bude

$$\bar{l} = \sum_{i=1}^{r} P_i l_i \,. \qquad (57)$$

s Příklad 36: Průměrná délka značky u prefixových kódů z příkladů 32 a 34.

Pravděpodobnosti výskytu cifer 0..9 nechť jsou tyto:
P(0) = 0,4, P(1) = 0,3, P(2) = 0,1, P(3) = 0,07, P(4) = 0,05, P(5) = 0,038, P(6) = P(7) = 0,02, P(8) = P(9) = 0,001.

Pak podle (57) bude průměrná délka značky:
Kód z příkladu 32: $\bar{l} = 0{,}4.2 + 0{,}3.2 + (0{,}1 + 0{,}07 + 0{,}05 + 0{,}038 + 2.0{,}02 + 2.0{,}001).4 = 2{,}6$ znaků.
Kód z příkladu 34: $\bar{l} = 0{,}4.2 + 0{,}3.2 + 0{,}1.3 + (0{,}07 + 0{,}05 + 0{,}038 + 2.0{,}02).4 + 2.0{,}001.5 = 2{,}502$ znaků.

Druhý z kódů bude komprimovat zprávy o něco efektivněji. Je ovšem otázka, zda by bylo možné nalézt jiný prefixový kód, který by měl ještě kratší průměrnou délku značky. Odpověď nám poskytnou Huffmanovy kódy.

þ

Huffmanovo kódování.

Definice: Uvažujme primární abecedu A = {a_1, a_2, .., a_r} s pravděpodobnostmi výskytu znaků P_1, P_2, .. , P_r a kódovou abecedu B = { b_1, b_2, .., b_n}. Huffmanovým *n*-znakovým kódováním se rozumí prefixové kódování abecedy A pomocí *n* kódových znaků z abecedy B, které má nejkratší možnou průměrnou délku kódového slova.

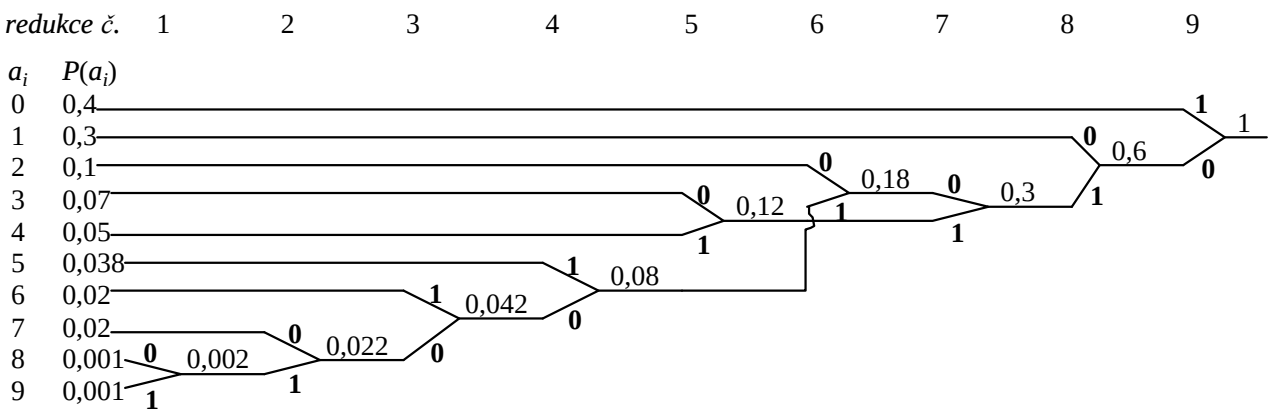
Definice pouze říká, že kód generující nejkratší možné zprávy v abecedě B se nazývá Huffmanův. Nic však neříká o tom, jak tento kód sestrojit.
V dalším se budeme zabývat především binárními Huffmanovy kódy (n = 2).

Konstrukce Huffmanova kódu – algoritmus.

V literatuře je možné nalézt několik způsobů sestavení Huffmanova kódu. Uvedeme jeden z nejnázornějších algoritmů.

1. Znaky primární abecedy seřadíme do sloupce shora dolů v pořadí od nejvyšší po nejnižší pravděpodobnost jejich výskytu (seřazení není nutné, řešení však bude graficky přehlednější). Pravděpodobnosti vepíšeme do sousedního sloupce vpravo.
2. Dva znaky s nejnižšími pravděpodobnostmi sdružíme do kombinovaného znaku, jehož pravděpodobnost výskytu je rovna součtu pravděpodobností slučovaných znaků (provedeme redukci dvou znaků na 1 znak). Graficky to provedeme spojením pravděpodobností slučovaných znaků lomenou čarou. K zalomení vepíšeme výslednou pravděpodobnost. K čáře vycházející od vyšší pravděpodobnosti zapíšeme nulu, k druhé čáře jedničku. Pokud jsou obě pravděpodobnosti stejné, zapíšeme nulu a jedničku podle svého uvážení. Oba slučované znaky již vyloučíme z dalšího postupu.
3. Bod dva periodicky opakujeme, až po poslední redukci dostaneme jediný znak s pravděpodobností výskytu 1 (konečný znak). Počet redukcí je o jedničku menší než počet znaků původní abecedy.
4. Kódové slovo k příslušnému znaku získáme tak, že postupujeme po existujících spojnicích zleva doprava od daného znaku až ke konečnému znaku. Přitom zapisujeme nuly a jedničky u větví, kterými procházíme. Vzniklé kódové slovo pak „otočíme“, tj. zapíšeme pozpátku.
   Pozn.: Kódové slovo samozřejmě můžeme přímo číst zprava doleva, u složitějších kódů to však mnohdy připomíná hledání cesty z bludiště.

s Příklad 37: Konstrukce Huffmanova kódu abecedy z příkladu 36.



Příklad kódování cifry „7“: 0001010, čteno pozpátku 0101000
Výsledná kódovací tabulka:

A	P(a_i)	B*	*d*
0	0,4	1	1
1	0,3	00	2
2	0,1	0100	4
3	0,07	0110	4
4	0,05	0111	4
5	0,038	01011	5
6	0,02	010101	6
7	0,02	0101000	7
8	0,001	01010010	8
9	0,001	01010011	8

Průměrná délka značky:

$\bar{l} = 0{,}4.1 + 0{,}3.2 + (0{,}1 + 0{,}07 + 0{,}05).4 + 0{,}038.5 + 0{,}02.6 + 0{,}02.7 + 2.0{,}001.8 = 2{,}346$ znaků.

Prefixový kód z příkladu 36, resp. 34 měl průměrnou délku značky 2,502 znaků. Nyní jsme našli kód o kratší průměrné délce značky. Podle definice Huffmanova kódu již nelze najít „kratší“ kód.

Všimněte si, že Huffmanův kód využívá extrémně krátkých značek ke kódování nejpravděpodobnějších znaků. Znaky velmi málo pravděpodobné kóduje relativně velmi dlouhými značkami. Srovnejte s kódem z příkladu 34, jehož nejdelší značka má délku 5, průměrná délka je ale přesto větší než u Huffmanova kódu.

þ

Z algoritmu konstrukce Huffmanova kódu a z příkladu 37 je zřejmé, že při sestavování kódu máme k dispozici určitou volnost – např. při slučování dvou znaků se stejnými pravděpodobnostmi je lhostejné, jakým způsobem jim přiřadíme nulu a jedničku. Z toho vyplývá, že řešením může být i několik odlišných Huffmanových kódů, které však budou mít stejnou průměrnou délku slova.

Využití entropie k vyhodnocení komprimačních schopností kódu

Uvažujme primární abecedu A = {a_1, a_2, .., a_r} s pravděpodobnostmi výskytu znaků P_1, P_2, .. , P_r a kódovou abecedu B = { b_1, b_2, .., b_n}. Prefixový kód nechť má průměrnou délku slova $\bar{l}$.

Primární abeceda má entropii H(A) danou vztahem (9). Je to průměrné množství informace na jeden znak primární abecedy. Protože kódováním se každému znaku přiřadí kódové slovo, je to i průměrné množství informace nesené kódovým slovem. Provedeme-li dělení průměrnou délkou kódového slova, dostaneme průměrné množství informace na kódový znak, což je entropie kódové abecedy:

$$H(B) = \frac{H(A)}{\bar{l}}. \tag{58}$$

Míru „hustoty“ informace před kódováním a po kódování můžeme vyjádřit poměrem skutečné a maximální možné entropie primární a kódové abecedy, neboli relativními entropiemi, případně redundancemi:

$$h(A) = \frac{H(A)}{H_{\max}(A)},\ r(A) = 1 - h(A), \tag{59}$$

$$h(B) = \frac{H(B)}{H_{\max}(B)},\ r(B) = 1 - h(B). \tag{60}$$

Pokud by relativní entropie před a po kódování byly stejné, znamenalo by to, že kód neprovádí vůbec žádnou komprimaci zprávy. Čím je poměr relativních entropií po a před kódováním větší než 1, tím je komprimace účinnější. K ohodnocení komprimačních schopností kódu tedy zavedeme komprimační poměr kódu *KP* jako

$$KP = \frac{h(B)}{h(A)}. \tag{61}$$

s Příklad 38: Vyhodnocení komprimačních schopností prefixových kódů č. 1, 2 a 3 (z příkladů 32, 34 a 37).

		kód 1	kód 2	kód 3
A	P(a_i)	B*		
0	0,4	00	00	1
1	0,3	01	01	00
2	0,1	1000	100	0100
3	0,07	1001	1010	0110
4	0,05	1010	1011	0111
5	0,038	1011	1100	01011
6	0,02	1100	1101	010101
7	0,02	1101	1110	0101000
8	0,001	1110	11110	01010010
9	0,001	1111	11111	01010011

$H(A) = -\sum_{i=0}^{9} P(a_i)\log_2 P(a_i) \doteq 2{,}2917$ Sh/znak, $H_{\max}(A) = \log_2 10 \doteq 3{,}322$ Sh/znak,

$h(A) \doteq \frac{2{,}2917}{3{,}322} \doteq 0{,}6899$, $r(A) \doteq 1 - 0{,}6899 = 0{,}3101$.

Průměrná délka značky $\bar{l}_1, \bar{l}_2, \bar{l}_3$ kódu č. 1, 2 a 3:

$\bar{l}_1 = 2{,}6$ znaků, $\bar{l}_2 = 2{,}502$ znaků, $\bar{l}_3 = 2{,}346$ znaků.

Entropie, relativní entropie a redundance kódové abecedy v případě kódů č. 1, 2 a 3:

$H_1(B) \doteq \frac{2{,}2917}{2{,}6} \doteq 0{,}8814$, $h_1(B) \doteq \frac{0{,}8814}{1} = 0{,}8814$, $r_1(B) \doteq 1 - 0{,}8814 = 0{,}1186$,

$H_2(B) \doteq \frac{2{,}2917}{2{,}502} \doteq 0{,}916$, $h_2(B) \doteq \frac{0{,}916}{1} = 0{,}916$, $r_2(B) \doteq 1 - 0{,}916 = 0{,}084$,

$H_3(B) \doteq \frac{2{,}2917}{2{,}346} \doteq 0{,}9769$, $h_3(B) \doteq \frac{0{,}9769}{1} = 0{,}9769$, $r_3(B) \doteq 1 - 0{,}9769 = 0{,}0231$.

Komprimační poměry kódů č. 1, 2 a 3:

$KP_1 \doteq \frac{0{,}8814}{0{,}6899} \doteq 1{,}278$, $KP_2 \doteq \frac{0{,}916}{0{,}6899} \doteq 1{,}328$, $KP_3 \doteq \frac{0{,}9769}{0{,}6899} \doteq 1{,}416$.

þ

Z příkladu je zřejmé, že Huffmanův kód č. 3 má největší komprimační poměr. Všimněme si však, že ani on neprovedl dokonalou kompresi, neboť redundance kódové abecedy je 2,31%. Uvědomme si, že i když je daný Huffmanův kód ze všech kódů nejefektivnější, neznamená to, že danou zprávu již nelze více komprimovat. Ve skutečnosti můžeme Hufmannův kód použít „šikovněji“ tak, že lze redundanci stlačit na libovolně malou kladnou hodnotu. Ukážeme si metodu kódování tzv. rozšířené abecedy.

Uvažujme primární abecedu A = {a_1, a_2, .., a_r} s pravděpodobnostmi výskytu znaků P_1, P_2, .. , P_r a kódovou abecedu B = { b_1, b_2, .., b_n}. Pak *x*-násobně rozšířenou primární abecedou A^x nazveme abecedu, jejíž znaky jsou všechna možná slova délky *x* tvořená znaky primární abecedy. Pravděpodobnost výskytu takovéto *x*-tice je pak rovna součinu pravděpodobností výskytů dílčích znaků v ní obsažených.

Význam této definice bude zřejmý pro takový způsob kódování, kdy se primární zpráva rozdělí na úseky délky *x* a každý úsek se kóduje prefixovým kódem (kódují se celé úseky, nikoliv pouhé znaky jako dosud).

s Příklad 39: Komprimace metodou rozšířené primární abecedy.

Zpráva je složena ze znaků 0 a 1. Jedničky značně převažují: pravděpodobnosti výskytu jsou P(1) = 0,9, P(0) = 0,1. Entropie této abecedy má proto značně daleko do teoretického maxima 1 Sh/znak:

$H_1 = -(0{,}9\log_2 0{,}9 + 0{,}1\log_2 0{,}1) \doteq 0{,}469$ Sh/znak.

Komprimaci provedeme tak, že celou zprávu rozdělíme na slova délky 2. Každé takové slovo budeme pokládat za znak rozšířené abecedy A^2 a budeme jej kódovat Huffmanovým kódem. Existují celkem 4 takovéto znaky, jejichž pravděpodobnosti výskytu jsou rovny součinům pravděpodobností výskytu původních znaků primární abecedy, např. P(01)=P(0).P(1). Huffmanův kód nalezneme známou konstrukcí. Zde je výsledek:

A^2	P	*značky*	l_2
00	0,01	101	3
01	0,09	100	3
10	0,09	11	2
11	0,81	0	1

Průměrná délka značky je

$\bar{l}_2 = 0{,}01.3 + 0{,}09.3 + 0{,}09.2 + 0{,}81.1 = 1{,}29$ znaků.

„Dvojznak“ rozšířené abecedy nese průměrné množství informace

$2H_1 \doteq 0{,}938$ Sh.

Stejné množství informace je tedy neseno značkou o průměrné délce 1,29 znaků. Jeden znak sekundární zprávy tedy nese průměrnou informaci

$H_2 \doteq \frac{0{,}938}{1{,}29} \doteq 0{,}727$ Sh/znak.

Oproti výchozí situaci bez kódování došlo k podstatnému zlepšení.

Nyní zkusíme rozdělit původní zprávu na slova délky 3. Rozšířená abeceda A^3 bude mít 8 znaků. Výsledek je zde:

A^3	P	*značky*	l_3
000	0,001	11111	5
001	0,009	11110	5
010	0,009	11101	5
011	0,081	100	3
100	0,009	11100	5
101	0,081	101	3
110	0,081	110	3
111	0,729	0	1

Průměrná délka značky je nyní

$\bar{l}_3 = 1{,}598$ znaků.

„Trojznak“ rozšířené abecedy nese průměrné množství informace

$3H_1 \approx 1{,}407$ Sh.

Stejné množství informace je tedy neseno značkou o průměrné délce 1,598 znaků. Jeden znak sekundární zprávy proto nese průměrnou informaci

$$H_3 \approx \frac{1{,}407}{1{,}598} \approx 0{,}88 \text{ Sh/znak.}$$

Opět jsme se více přiblížili k teoretickému maximu 1 Sh/znak.

þ

Výsledky z příkladu 39 lze zobecnit:
Zprávu obsahující znaky primární abecedy *A* rozdělíme na úseky délky *x*, které chápeme jako znaky *x*-násobně rozšířené primární abecedy A^x. Tyto znaky kódujeme binárním Huffmanovým kódem o průměrné délce značky $\bar{l}_x$. Je-li entropie primární abecedy *H*, pak entropie rozšířené abecedy bude *H.x* a entropie kódové binární abecedy

$$H_x = H\frac{x}{l_x}. \tag{62}$$

Lze dokázat, že

$$\lim_{x\to\infty} H_x = H \lim_{x\to\infty} \frac{x}{l_x} = 1, \tag{63}$$

neboli

$$\lim_{x\to\infty} \frac{l_x}{x} = H. \tag{64}$$

Vzorec (63) říká, že

Zvětšováním délky kódovaných úseků můžeme neomezeně zvětšovat entropii kódové abecedy až k jejímu teoretickému maximu, resp. snižovat její redundanci k nule.

Interpretace vzorce (64) může být následující:

Počet binárních znaků (tj. bitů), kterými můžeme zakódovat jeden znak primární abecedy, lze zvětšováním délky kódovaných úseků libovolně zmenšovat až k hodnotě, která se rovná entropii primární abecedy.

Jedná se o důsledek Shannonova teorému zdrojového kódování ze str. 27.

Metody snižování nadbytečnosti ve zprávě používané ke komprimaci počítačových souborů [11].

Komprimační algoritmy jsou založeny na hledání určitého řádu v uložených datech. Je-li tímto řádem frekvence výskytu jednotlivých znaků, pak se uplatní Huffmanovo kódování. To je vhodné pro komprimaci např. textových souborů. Při komprimaci obrázků je vhodnější sledovat jinou zákonitost – výskyty dlouhých bloků stejných dat, případně opakované sekvence určitých znaků.

Používají se metody bezeztrátové komprimace (lossless compression, z komprimovaného souboru lze jednoznačně rekonstruovat původní soubor) a ztrátové komprimace (lossy compression, z komprimovaného souboru nelze přesně obnovit původní soubor).
Příklad bezeztrátové komprimace: textové a binární soubory, kde si nemůžeme dovolit jakoukoliv ztrátu dat (programy PKZIP, RAR, ARJ, …), z obrazových formátů jsou to např. PCX, TIFF, GIF a PNG.
Příklad ztrátové komprimace: komprimace obrázků, zvuku a videa.

Bezeztrátová komprimace:

Komprimační algoritmy mohou být neadaptivní a adaptivní.
Neadaptivní algoritmy jsou určeny pouze pro kompresi určitého typu dat. Většinou pracují s předdefinovanými slovníky často se vyskytujících řetězců, které jsou v průběhu kódování nahrazovány kratšími slovy. Příkladem je Huffmanovo kódování nebo příbuzné Shannon-Fanovo kódování.
Některé neadaptivní algoritmy nepotřebují slovníky a dlouhé řetězce opakujících se znaků „zhušťují“ okamžitě, jakmile na ně algoritmus narazí. Příkladem je algoritmus RLE (Run-Length Encoding), využívaný grafickým formátem PCX. Řetězec stejných znaků se nahradí jedním znakem a doplní se údajem o počtu opakujících se znaků.
Adaptivní algoritmy lze použít ke komprimaci širší třídy souborů, neboť si vytvářejí slovníky často se vyskytujících řetězců dynamicky během komprese. K nejznámějším adaptivním algoritmům patří algoritmus LZW (Lempel-Ziv-Welch). Algoritmus LZW je využíván v grafických formátech GIF a TIFF a je zabudován do komprimačních programů typu PKZIP a ARJ.

Nejznámější metody komprese založené na využití pravděpodobností výskytu znaků: Huffmanovo kódování a tzv. aritmetické kódování (zakóduje celou zprávu jako jediné číslo s velkým počtem cifer; značné nároky na hardware). Samostatně se dají použít jen pro kompresi textových souborů. Dobrých výsledků dosahují v kombinaci s adaptivními algoritmy typu LZW.

Ztrátová komprimace:
JPEG (Joint Photographic Experts Group) – sada metod ztrátové komprese vhodných zejména pro složité barevné předlohy s velkou hloubkou barev. Cíl komprese: ponechat informaci o intenzitě a jasu (oko je na tyto atributy citlivé), potlačit informaci o malých změnách barev blízkých bodů (oko nerozezná). Metoda je nevhodná pro komprimaci vektorových obrázků a jednoduchých kreseb s malou hloubkou barev a s rozsáhlými jednobarevnými plochami (lepší výsledky se dosáhnou bezeztrátovou kompresí, viz GIF, PCX).
Waveletová (vlnková) transformace –ztrátová komprese budoucnosti. Podstata jde za rámec tohoto předmětu. Při srovnatelné kvalitě s formátem JPEG může dávat zhruba dvojnásobný kompresní poměr.
MPEG (Moving Picture Expert Group) – komprimace videa spolu se zvukovým doprovodem.
Dnes existují 4 normy MPEG:
MPEG-1: je navržena pro technologii CD tak, aby umožňovala přenosovou rychlost do 1,5 Mb/s (1,2Mb/s pro obrazová data, 0,3 Mb/s pro audio data). Podporuje rozlišení obrázků do 4095x4095 bodů pří 60 snímcích za sekundu.
MPEG-2: je navržena pro dálkové a satelitní přenosy signálu při zachování televizní kvality. Podporuje rozlišení do 16383x16383 bodů, umožňuje prokládání snímků.
MPEG-3: je navržena pro HDTV (TV s vysokým rozlišením). Dnes se nepoužívá, tuto podporu převzala norma MPEG-2.
MPEG-4: je navržena pro přenos videa na pomalých linkách s přenosovou rychlostí od 4800 do 64000 bitů/s. Rozlišení jen 176x144 bodů při 10 snímcích za sekundu.
Komprese zvuku může probíhat ve třech „vrstvách“ – Layer 1 až 3. Čím vyšší vrstva, tím vyšší kvalita zvuku. Příslušné soubory se zvukovým záznamem pak mají přípony .mp1 až .mp3. Kvalita komprimovaného souboru .mp3 (přenosová rychlost od 32kb/s do 320 kb/s) se blíží kvalitě audio

CD (1,4 Mb/s!). Na jedno datové CD je možné uložit 12 až 13 klasických audio CD při prakticky nezměněné zvukové kvalitě. Komprese zvuku je ztrátová, vynechávají se „detaily“, které lidské ucho nevnímá (v přítomnosti silného signálu nevnímáme signál slabší, silný vjem má setrvačnost působení a přehluší jiné složky ještě chvíli po jeho zániku, …).
Komprese podle norem MPEG je časově náročná, může probíhat čistě softwarově, hardwarově nebo kombinovaně.

Kanálové kódování - kódy pro zabezpečení dat proti chybám.

Detekční a korekční kódy.

V kapitole „*Shannonova koncepce komunikačního systému*“ (str. 13-15) je uvedeno dělení bezpečnostních kódů na detekční (používané v systémech ARQ) a korekční (pro systémy FEC). Data se rozdělí na úseky stejné délky, každý z úseků je doplněn o tzv. zabezpečovací bity podle určitého matematického algoritmu. Jsou-li data z úseku narušena během přenosu, dokáže dekodér detekčního kódu tento fakt buď pouze identifikovat (ví se, že došlo k chybě, ale neví se, na kterém místě), nebo v případě korekčního kódu dokáže lokalizovat chybu a opravit ji.

Zabezpečení zprávy korekčním kódem je obecně náročnější než v případě detekčního kódu – jednak počet zabezpečujících bitů bývá větší, jednak je složitější kódovací a dekódovací mechanismus.

Je dobré uvědomit si, že neexistuje kód, jehož nasazení by stoprocentně zaručovalo správnou detekci či korekci chyb. Lze jen garantovat procentuální úspěšnost detekce či korekce a zdokonalováním bezpečnostního kódu se neomezeně přibližovat k hranici stoprocentní úspěšnosti.

K zabezpečení dat se používají převážně blokové kódy, tj kódy využívající značek stejné délky. Použití konvolučního, tedy neblokového kódování, bylo diskutováno v předmětu „Teorie sdělování a přenosu dat“ (TSD).

Druhy chyb.

V předmětu TSD byly chyby obecně rozděleny na chyby ojedinělé (nezávislé) a shluky chyb (angl. Independent Error a Burst Error). Je-li v daném úseku dat pouze jedna chyba, hovoříme o jednoduché chybě, v případě n chyb jde o n-násobnou chybu. Při rostoucím n prudce klesá pravděpodobnost n-násobné chyby v daném úseku dat. Toho se využívá k zabezpečování těchto úseků jednoduššími kódy, které jsou schopny stoprocentně detekovat, resp. korigovat jen několikanásobné chyby, například jednoduché, dvojnásobné a trojnásobné. Pokud dojde k vícenásobné chybě, kód selže. Spoléhá se ale na to, že pravděpodobnost je mizivá.

Shluky chyb se objevují např. při rádiovém přenosu na VKV, při poškození média pro záznam dat apod. Na shluky chyb je třeba nasadit jiný typ bezpečnostních kódů než na chyby ojedinělé.

V dalším se budeme věnovat pouze binárním blokovým kódům, které kódují binární slova primární abecedy na binární značky.

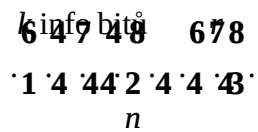
Struktura kódového slova blokového kódu.

Primární zpráva je zabezpečována po úsecích délky k. Říkáme, že informační slovo obsahuje k bitů. Kodér k těmto bitům přidá podle určité zákonitosti r zabezpečovacích bitů, které jsou odvozeny z informačního slova. Vznikne n-bitové kódové slovo, kde

$$n = k + r\,. \tag{65}$$

Jsou dvě možné struktury kódového slova délky n:

1. Zabezpečovací část je připojena bezprostředně za informační část. Kód se pak nazývá systematický.



Obr. 13. Struktura značky blokového systematického kódu.

2. Informační a zabezpečující bity jsou vzájemně promíchány. Kód je pak nesystematický.

Přidáváním zabezpečovací části roste délka slova, takže roste doba přenosu celé zprávy. Tím platíme za její zabezpečení.

Poměr počtu informačních bitů k celkovému počtu bitů ve značce se nazývá informační poměr kódu (bohužel se značí stejně jako přenosová rychlost):

$$R = \frac{k}{n} = \frac{n-r}{n} \in (0,1\rangle . \qquad (66)$$

Umění je nalézt takový kompromisní kód, který by při co nejvyšším informačním poměru dovedl co nejlépe zabezpečit zprávu.
Blokový kód o parametrech *n, k, r* se označuje jako kód (*n*,*k*). Tak kód z následujícího příkladu má označení (4,2).

s Příklad 40: blokový systematický kód.

informační slovo	kódové slovo
00	0000
01	0110
10	1001
11	1111

$k = 2, r = 2, n = 4$

$R = \frac{2}{4} = 0{,}5$.

Všimněme si, že čtyřmístná kódová slova jsou 4, i když existuje celkem 16 možných čtyřmístných binárních slov. Jak uvidíme dále, výběr čtyř kódových slov z 16 možných slov je zvolen tak, aby kód dovedl detekovat a opravovat případné chyby podle určitého kritéria.

þ

Na základě příkladu 40 provedeme následující zobecnění:
Uvažujme binární blokový kód, jehož kódové slovo má délku *n*, z toho *k* informačních a *r* zabezpečovacích bitů. Pak:
Počet všech možných informačních slov délky *k* je 2^k. Je to zároveň počet všech kódových slov délky *n*.
Kódová slova jsou podmnožinou 2^n všech možných slov délky *n*.
Jestliže dojde vlivem chyby k modifikaci kódového slova na slovo, které nepatří do kódu, je dekodér schopen detekovat chybu.
Jestliže je kód zkonstruován tak dobře, že existuje jen jedno kódové slovo, které se liší od přijatého slova v nejmenším počtu bitů, pak bude kodér schopen chybu opravit.

Základní myšlenka bezpečnostního kódování.
Nyní blíže rozvedeme jednu z možných strategií bezpečnostního kódování, naznačenou v předchozí kapitole. Nejprve definujme některé nutné pojmy.
Délka kódu *L* – je to počet všech různých kódových slov. Tuto délku je nutno odlišit od délky kódového slova. Z předchozího textu vyplývá vztah mezi délkou kódu a počtem informačních bitů:

$$L = 2^k . \qquad (67)$$

Hammingova váha *w*(*v*) kódového slova *v* je počet jedniček v kódovém slovu.
Hammingova vzdálenost $d(v_1, v_2)$ dvou kódových slov v_1 a v_2 je počet bitů, v nichž se kódová slova liší.
Minimální vzdálenost kódu d_{min} je minimální možná vzdálenost všech dvojic značek v kódu.

s Příklad 41: blokový systematický kód (4,2) z příkladu 40.

Hammingovy váhy jednotlivých kódových slov jsou uvedeny v tabulce.

i	informační slovo	kódové slovo v_i	$w(v_i)$
1	00	0000	0
2	01	0110	2
3	10	1001	2
4	11	1111	4

Hammingovy vzdálenosti:

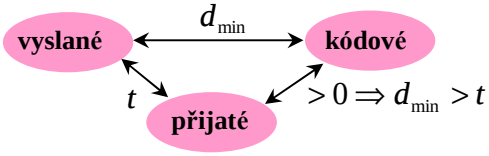
$d(v_i, v_j)$	0000	0110	1001	1111
0000	-	2	2	4
0110		-	4	2
1001			-	2
1111				-

Proto $d_{\min} = 2$.

Všimněme si, že objeví-li se v libovolném kódovém slovu jen 1 chyba, dostaneme značku, která nepatří do kódu. Tento kód tedy je schopen stoprocentně detekovat jednoduché chyby. Nedovede však jednoznačně opravovat jednoduché chyby. Objeví-li se např. při přenosu slova 0000 chyba na posledním místě, tj. přijmeme-li 0001, „nejbližší“ ve smyslu Hammingovy vzdálenosti jsou dvě kódová slova: 0000 (správné) a 1001 (nesprávné). Dekodér nemá možnost jednoznačně se rozhodnout pro správnou variantu.

þ

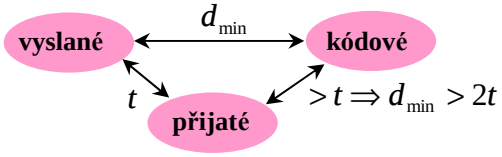
Algoritmus detekce chyb:
Přijaté slovo se porovná s množinou kódových slov. Nezjistí-li se shoda s žádným kódovým slovem, je detekována chyba.
Je zřejmé, že pokud $d_{\min} = 1$, není zaručena stoprocentní detekce chyby, protože vlivem chyby může kódové slovo přejít v jiné kódové slovo.



Pokud $d_{\min} = 2$, je zaručena stoprocentní detekce všech jednoduchých chyb.
Pokud $d_{\min} = 3$, je zaručena stoprocentní detekce všech dvojitých chyb.
Obrázek znázorňuje, že přijaté slovo se v důsledku t-násobné chyby vzdálí od vyslaného kódového slova o vzdálenost t. Aby mohla být chyba detekovatelná, musí být vzdálenost přijatého slova od každého kódového slova větší než nula. Proto platí následující tvrzení:
Obecně je zaručena stoprocentní detekce všech t-násobných chyb, jestliže

$$d_{\min} \geq t + 1. \qquad (68)$$

Algoritmus korekce chyb:
Přijaté slovo se porovná s množinou kódových slov. Pokud je nalezeno jediné kódové slovo, které má od přijatého slova nejmenší Hammingovu vzdálenost, je toto kódové slovo prohlášeno za slovo vyslané. Jsou tak automaticky opraveny chyby na místech, v nichž se liší přijaté a nejbližší kódové slovo.



Je zřejmé, že pokud $d_{\min} = 1$, není zaručena stoprocentní detekce a tudíž ani korekce chyby.
Je-li slovo zatíženo jednoduchou chybou, pak musí mít kód minimální vzdálenost 3, aby každou takovou chybu byl schopen opravit. Při d_{min} = 2 se totiž může stát, že jedna chyba „odchýlí“ přijaté slovo od vyslaného o stejnou vzdálenost (1), o jakou se přiblíží k jinému kódovému slovu.
Obrázek znázorňuje, že přijaté slovo se v důsledku t-násobné chyby vzdálí od vyslaného kódového slova o vzdálenost t. Aby mohla být chyba opravitelná, musí být vzdálenost přijatého slova od

každého kódového slova větší než *t*. Pak totiž bude mít přijaté slovo nejkratší vzdálenost právě ke slovu vyslanému. Proto platí následující věta:
Obecně je zaručena stoprocentní korekce všech *t*-násobných chyb, jestliže

$$d_{\min} \geq 2t + 1. \tag{69}$$

s Příklad 42: detekce a korekce chyb blokovým systematickým kódem (4,2) z příkladu 41.

Protože $d_{min} = 2$, jakákoliv jednoduchá chyba v jakémkoliv kódovém slovu má za následek, že slovo se nezmění v jiné kódové slovo. Každou jednoduchou chybu tedy můžeme snadno detekovat.
Objeví-li se např. v kódovém slovu 0000 dvě chyby na začátku a na konci, tj. přijmeme 1001, chyby nejsou detekovány, protože se jedná o kódové slovo. Jiná bude situace, objeví-li se dvě chyby na konci: 0011. Pak je nesprávný přenos detekován.
Rovnice (68) tedy garantuje detekci všech *t*-násobných chyb, nevylučuje však detekci některých větších skupin chyb.
Podle vzorce (69) daný kód negarantuje korekci všech jednoduchých chyb, protože jeho minimální vzdálenost je pouze 2. Skutečně, přijmeme-li např. kódové slovo 1111 s chybou na posledním místě, tj. 1110, je stejná pravděpodobnost, že bylo vysláno slovo 1111 jako 0110. Od obou kódových slov má přijaté slovo stejnou vzdálenost 1. Konkrétně daný kód neumožňuje korekci žádné jednoduché chyby.

þ

s Příklad 43: detekce a korekce chyb blokovým systematickým kódem (6,3).

i	informační slovo	kódové slovo v_i	$w(v_i)$
1	000	000000	0
2	001	001110	3
3	010	010101	3
4	011	011011	4
5	100	100011	3
6	101	101101	4
7	110	110110	4
8	111	111000	3

Minimální vzdálenost kódu je 3:

$d(v_i, v_j)$	000000	001110	010101	011011	100011	101101	110110	111000
000000	-	3	3	4	3	4	4	3
001110		-	4	3	4	3	3	4
010101			-	3	4	3	3	4
011011				-	3	4	4	3
100011					-	3	3	4
101101						-	4	3
110110							-	3
111000								-

Znamená to, že lze jednoznačně detekovat jednoduché a dvojité chyby (viz vzorec 68) a opravovat všechny jednoduché chyby (viz vzorec 69).
Kromě toho však kód umožňuje detekci např. trojnásobné (nebo i šestinásobné) chyby, která z kódového slova 000000 udělá nekódové slovo 000111 (nebo 111111).

þ

V následující části ukážeme některé metody kódování zprávy, které umožní podstatné snížení chyb při přenosu binárním kanálem.

Kódování a přenosový kanál.
Uvažujme binární symetrický kanál bez paměti o spolehlivosti *P*, resp. chybovosti *Q* = 1-*P*.

s Příklad 44: Vztah mezi spolehlivostí kanálu a spolehlivostí přenosu binárního slova.

Binární symetrický kanál bez paměti má spolehlivost 99%. Vypočtěte pravděpodobnost P_{OK} bezchybného přenesení binárního slova délky 16.

$P_{OK} = 0{,}99^{16} \doteq 0{,}851$.

Při chybovosti kanálu 1% je pravděpodobnost chybného přenesení slova téměř 15%. Při prodlužování binárního slova pravděpodobnost chyby roste.
Z příkladu vyplývá důležitost bezpečnostního kódování. Bez něj by chybovost při přenášení dlouhých zpráv přerostla únosnou mez.

þ

Je tedy vhodné rozlišovat mezi spolehlivostí kanálu *P* a spolehlivostí přenosu slova P_{OK}, resp. mezi chybovostí kanálu *Q* a chybovostí přenosu slova $P_{ERR} = 1 - P_{OK}$. Rovnítko můžeme dát mezi příslušné dvojice pouze za předpokladu, že přenášené slovo má délku 1.

Chybovost přenosu slova kanálem můžeme libovolně snižovat vhodným kódováním zprávy. Mechanismus objasníme na příkladu tzv. opakovacího kódu, což je velmi jednoduchý blokový kód.

Podstata použití opakovacího kódu: každý binární znak vyšleme do kanálu několikrát za sebou. Na výstupu kanálu se rozhodne o správném znaku podle toho, který z nich byl přijat nejvícekrát (metoda hlasování). Z toho důvodu je vhodné, je-li vysílán lichý počet znaků.

s Příklad 45: Znak je opakovaně vysílán za účelem zvýšení spolehlivosti jeho přenosu.

Uvažujme binární symetrický kanál o spolehlivosti P = 0,9. Vypočteme, zda nezvýšíme spolehlivost správného přenesení jednoho binárního znaku pomocí opakovacího kódu s tím, že stejný znak bude vysílán třikrát za sebou.
Na kódovací proces je tedy možné pohlížet pomocí kódovací tabulky blokového kódu:

binární znak	*kódové slovo*
0	000
1	111

Minimální vzdálenost opakovacího kódu je rovna délce kódového slova, v našem případě 3. Podle vzorců (68) a (69) tedy kód umožňuje stoprocentní detekci všech jednoduchých a dvojnásobných chyb a korekci všech jednoduchých chyb.
Pravděpodobnost správného dekódování kódového slova P_{OK} se určí jako pravděpodobnost složeného jevu, spočívající ve správném přenesení alespoň dvou znaků (viz strategie dekódování hlasováním):

$P_{OK} = 0{,}9^3 + 3.0{,}9^2.0{,}1 = 0{,}972$.

První člen znamená pravděpodobnost správného přenosu všech tří znaků, druhý člen je pravděpodobnost správného přenesení dvou znaků a tudíž chybného přenesení zbývajícího znaku (jsou 3 možné kombinace).

$P_{ERR} = 1 - P_{OK} = 0{,}028$.

Vidíme, že kódováním se zvýšila spolehlivost přenesení jednoho znaku z 90% na 97,2% a odpovídající chybovost poklesla z 10% na 2,8%.

Nyní budeme znak vysílat pětkrát za sebou. Tomu bude odpovídat blokový kód s pětimístnými značkami, který je podle teorie blokových kódů schopen detekovat všechny čtyřnásobné a menší chyby a opravovat všechny jednoduché a dvojnásobné chyby:

binární znak	*kódové slovo*
0	00000
1	11111

K správnému dekódování nyní dojde za předpokladu, že budou správně přeneseny alespoň tři z pěti znaků. Tomu odpovídá výpočet příslušné pravděpodobnosti:

$P_{OK} = 0{,}9^5 + 5.0{,}9^4.0{,}1 + \binom{5}{2}.0{,}9^3.0{,}1^2 \doteq 0{,}9914$,

$P_{ERR} = 1 - P_{OK} \doteq 0{,}0086$.

Dá se očekávat, že zvětšováním počtu opakování lze snížit chybovost libovolně k nule.

þ

Použitím opakovacího kódu sice můžeme výrazně zvýšit spolehlivost přenosu, avšak za cenu malého informačního poměru kódu a z toho vyplývajícího značného prodlužování zprávy a času na její přenesení.
Je proto na místě otázka, zda existují efektivnější kódy než opakovací. Máme-li například k dispozici binární kanál, který je schopen přenášet data jen dvakrát rychleji, než jak jsou generována zdrojem zprávy, musíme se omezit na kódy s informačním poměrem 1:2 a větším. Ptáme se, zda je možné zkonstruovat takový kód s pevným informačním poměrem, např. 0,5, který by umožnil snížení chybovosti např. z 10% na 1%. Odpověď nám dá později modifikovaná Shannonova věta o kanálovém kódování. V následujícím příkladu použijeme ke kódování metodu rozšířené abecedy zdroje, která se nám osvědčila již při komprimování zpráv.

s Příklad 46: Kódování zprávy jejím dělením na slova stejné délky.

Uvažujme binární symetrický kanál o spolehlivosti P = 0,9. Hledáme bezpečnostní kód s informačním poměrem 0,5, který by zvýšil spolehlivost přenosu.
Zvolíme tuto strategii kódování: binární zprávu rozdělíme na posloupnost slov délky 2. Každé slovo délky 2 zakódujeme značkou délky 4 podle následující kódovací tabulky systematického blokového kódu:

binární dvojznak	*kódové slovo*
00	0000
01	0111
10	1000
11	1111

Tento kód má minimální vzdálenost pouze 1 (je to vzdálenost první a třetí nebo druhé a čtvrté značky), takže nezaručuje ani detekci, natož korekci všech jednoduchých chyb. Přesto však jeho zabezpečovací schopnosti nejsou zanedbatelné. Přeneseme-li totiž první bit správně a bude-li v dalších třech bitech jen jedna chyba, je možné tuto chybu opravit (v těchto třech bitech je totiž zakódována druhá cifra opakovacím kódem délky 3). Pravděpodobnost správného dekódování značky na původní dvoubitové slovo je tedy
$P_{OK} = 0{,}9\left(0{,}9^3 + 3.0{,}1.0{,}9^2\right) = 0{,}8748$, neboli $P_{ERR} = 0{,}1252$.
Pravděpodobnost P_{OK}, resp. chybovost P_{ERR} je však třeba srovnat s pravděpodobností správného, resp. nesprávného přenesení slova délky 2 bez kódování, což je
$P^2 = 0{,}9^2 = 0{,}81$, neboli $1 - P^2 = 0{,}19$.

Nasazení daného kódování tedy sníží chybovost v poměru

$\frac{1-P^2}{P_{ERR}} \doteq 1{,}52$.

Poznamenejme, že k danému snížení chybovosti dojde při tomto způsobu dekódování: předpokládáme, že první znak je přenesen správně, o druhém znaku dekodér rozhodne na základě větší četnosti znaků na místech 2 až 4.

Nyní zkusíme jiný kód se stejným informačním poměrem 0,5. Rozdělíme zprávu na slova délky 3 a každé budeme kódovat systematickým kódem z příkladu 43 podle uvedené tabulky:

binární trojznak	*kódové slovo*
000	000000
001	001110
010	010101
011	011011
100	100011
101	101101
110	110110
111	111000

V příkladu 43 jsme určili minimální vzdálenost kódu 3. Kód je tedy schopen opravit všechny jednoduché chyby. Pravděpodobnost správného dekódování binárních trojznaků bude

$P_{OK} = 0{,}9^6 + 6.0{,}9^5.0{,}1 = 0{,}885735$, $P_{ERR} = 0{,}114265$.

K správnému dekódování totiž dojde buď při správném přenesení všech šesti znaků (s pravděpodobností $0{,}9^6$), nebo při správném přenesení 5 znaků a chybě v 1 znaku (pravděpodobnost $0{,}9^5.0{,}1$ je třeba uvažovat šestkrát, protože tolik je možností, jak umístit 1 chybu v šestimístné značce).

Spolehlivost a chybovost při přenesení nekódovaného slova délky 3 jsou

$P^3 = 0{,}9^3 = 0{,}729$, $1-P^3 = 0{,}271$.

Použitím tohoto kódu se snížila chybovost přenesení trojznaku

$\frac{1-P^3}{P_{ERR}} \doteq 2{,}37$ krát.

K snížení chybovosti dojde tentokrát při dekódování podle strategie minimální Hammingovy vzdálenosti.

þ

Výsledky z příkladu 46 naznačují, že čím více bitů sdružujeme do slov, která jsou kódována, tím více je možné potlačit původní chybovost při přenosu zprávy. Podrobně o tom hovoří modifikovaná Shannonova věta o kanálovém kódování:

V každém binárním symetrickém kanálu bez paměti o kapacitě C > 0 lze přenášet data s libovolně malou chybovostí, jestliže použijeme vhodný kód s informačním poměrem

$$R < C .$$

Pozn.: Daná věta platí i pro jiné kanály než binární.

V příkladu 46 se jednalo o binární kanál o spolehlivosti 90%, takže jeho kapacita je podle rovnice (50)

$C = 1 + 0{,}9\log_2 0{,}9 + 0{,}1\log_2 0{,}1 \doteq 0{,}531$ Sh/znak.

Podle Shannonovy věty je možné spolehlivost libovolně zvyšovat pomocí kódů s informačním poměrem menším než 0,531. V příkladu byl požadavek na informační poměr 0,5.

Obecný pohled na bezpečnostní kódování a dekódování.

Z předchozích příkladů je zřejmé, že účinnost zabezpečení dat před chybami závisí obecně na dvou vzájemně souvisejících faktorech:

1. Na způsobu kódování.
2. Na způsobu dekódování.

Například v první části příkladu 46 jsme poznali, že daný kód se z hlediska jeho minimální vzdálenosti $d_{min} = 1$ jevil jako bezcenný, protože negarantuje ani detekci všech jednoduchých chyb. Při dekódování zprávy metodou minimální Hammingovy vzdálenosti skutečně naprosto selže. Použijeme-li však jinou strategii dekódování, uvedenou v příkladu, dosáhneme dobrého zabezpečení.

O konkrétních způsobech dekódování a kódování zpráv pro používané typy kódů se zmíníme později. Nejprve analyzujme některé společné rysy dekódovacích a až pak kódovacích procesů.

Dekódovací proces

V důsledku chyb je vyslané kódové slovo přijato v pozměněné podobě. Úkolem dekodéru je transformovat podle jednoznačného algoritmu přijaté slovo v slovo kódové tak, aby byla maximalizována pravděpodobnost, že dekódované slovo bude odpovídat slovu vyslanému.

Obecně dekódování je proces, realizující zobrazení δ množiny K^n přijatých slov délky n do množiny K kódových slov:

$$d : K^n \to K . \tag{70}$$

Každému přijatému slovu musí odpovídat kódové slovo, a to i v případě, že dekodér nedokáže chybu jednoznačně opravit. V takovém případě se musí dekodér rozhodnout pro konkrétní řešení, které se jeví jako nejoptimálnější.

s Příklad 47: Dekódování opakovacího kódu [4].

Znak primární zprávy nechť se vysílá šestkrát. Dekódování je pak snadné, přijmeme-li slovo, kde je většina znaků stejných. Například
$d(001001) = (000000)$.

Při stejném počtu přijatých nul a jedniček však vzniká problém, který se dá řešit několika způsoby: Může se například stanovit, že při rovnosti počtu nul a jedniček se vždy dekóduje jednička. Někdy se omezujeme na částečné (rychlé) dekódování, což znamená, že takovéto stavy se neošetřují algoritmem a přiřadí se jim číslo z generátoru náhodných čísel. Nebo sestavíme speciální dekódovací algoritmus, například:

$$d(v_1\ v_2\ v_3\ v_4\ v_5\ v_6) = \begin{cases} 000000 & \text{je - li } v_i = 0 \text{ pro alespoň 4 indexy } i, \text{ nebo} \\ & v_1 = 0 \text{ a } v_i = 0 \text{ pro 2 indexy } i > 1, \\ 111111 & \text{jinak} \end{cases}$$

Tento algoritmus zaručeně opraví nejen všechny jednoduché a dvojnásobné chyby, ale i všechny trojnásobné chyby, které nemění první znak.

þ

Kódovací proces

Budeme uvažovat pouze kódování, které používá množinu K_i informačních slov a množinu K kódových slov. Pak kódování informačních slov je vzájemně jednoznačné zobrazení φ množiny K_i do množiny K:

$$j : K_i \to K . \tag{71}$$

Zobrazení φ se musí definovat praktickým kódovacím předpisem, jak generovat kódová slova k informačním slovům. Známý je předpis ve formě kódovací tabulky. Často je výhodný jiný předpis ve formě vzorce.

s Příklad 48: Opakovací kód.

Znak *v* primární zprávy nechť se vysílá pětkrát za sebou. Kódování je pak dáno předpisem $j(v) = vvvvv$.
Zápis na pravé straně je možné chápat jako informační znak doprovázený čtyřmi kontrolními znaky.

þ

s Příklad 49: Paritní kód.

Binární informační slova délky 3 jsou doplněna jedním kontrolním bitem tak, aby celkový počet jedniček v kódovém slovu byl sudý. Kódovací tabulka vypadá následovně:

informační slovo	*kódové slovo* (v_1 v_2 v_3 v_4)
000	0000
001	0011
010	0101
011	0110
100	1001
101	1010
110	1100
111	1111

Můžete se přesvědčit o tom, že minimální vzdálenost kódu je 2. Kód tedy odhalí všechny jednoduché chyby. Kromě toho však díky dekódovacímu algoritmu (hlídá se, zda celkový počet jedniček je sudý) odhalí i všechny trojnásobné chyby.
Kódovací algoritmus se dá elegantně zapsat jedinou rovnicí

$$j : v_1 + v_2 + v_3 + v_4 = 0 \qquad (72)$$

za předpokladu, že zavedeme algebraické operace „modulo 2“:
0 + 1 = 1 + 0 = 1, 0 + 0 = 1 + 1 = 0.

þ

s Příklad 50: „Koktavý“ kód [4].

Binární informační slova délky 3 jsou kódována na slova délky 6 tak, že každý znak se jednou zopakuje, například $j(101) = 110011$.
Je-li zápis kódového slova (v_1 v_2 v_3 v_4 v_5 v_6), pak kromě klasické kódovací tabulky můžeme kódovací algoritmus popsat soustavou rovnic

$$\begin{aligned} v_1 + v_2 &= 0 \\ v_3 + v_4 &= 0 \\ v_5 + v_6 &= 0, \end{aligned} \qquad (73)$$

opět za předpokladu aritmetiky „modulo 2“.

þ

Věta o hraniční minimální vzdálenosti systematického kódu.
Opakovací, paritní i „koktavý“ kód z předchozích příkladů jsou kódy blokové. První dva jsou navíc systematické, poslední nikoliv.
Pro systematické kódy platí následující věta [4]:
Minimální vzdálenost d_{min} systematického kódu může být nejvýše o jedničku větší než počet kontrolních znaků:

$$d_{\min} \le r+1 = n-k+1 \tag{74}$$

Nerovnost je vyjádřením kompromisu, před nímž stojíme při výběru vhodného kódu: na jedné straně se snažíme při dané délce značky n o co nejmenší počet kontrolních znaků, aby se do značky „vešlo“ co nejvíce informačních prvků, na straně druhé nám jde o co největší zabezpečení dat, čili o velké d_{min}. Oba požadavky jsou v rozporu. Proto je důležité hledat vhodné kódy, které prodlužují zprávy relativně málo při schopnosti opravy požadovaného počtu chyb. Některé z takových kódů budou popsány v dalším textu. Většinou patří do kategorie tzv. lineárních kódů.

Lineární binární kódy.

V příkladech 49 a 50 byl ukázán elegantní způsob, jak zákonitosti, kterými je vázána struktura kódového slova, jedinou nebo několika rovnicemi. Co měly ukázky v daných příkladech společné?

1. Jednalo se o binární kódy.

2. Nad binárními kódovými prvky jsou prováděny operace „modulo 2“ podle pravidel

$$0 + 1 = 1 + 0 = 1,\ 0 + 0 = 1 + 1 = 0, \tag{75}$$
$$1.1 = 1,\ 1.0 = 0,\ 0.1 = 0,\ 0.0 = 0. \tag{76}$$

3. Soustava rovnic se dá ve všech případech zapsat ve formě maticové rovnice

$$\mathbf{H}.\mathbf{v}^t = \mathbf{0}^t, \text{ resp. } \mathbf{v}.\mathbf{H}^t = \mathbf{0}, \tag{77}$$

kde **H** je jistá matice, kterou nazýváme kontrolní matice kódu, a **v** je tzv. vektor kódové značky. **0** je nulový vektor a znakem ^{t} je označena transpozice matice, resp. vektoru.

Konkrétně pro paritní kód z příkladu 49 se dá rovnice (72) přepsat do maticového tvaru

$$\begin{pmatrix}1 & 1 & 1 & 1\end{pmatrix}\begin{pmatrix}v_1\\v_2\\v_3\\v_4\end{pmatrix}=0, \text{ resp. } \begin{pmatrix}v_1 & v_2 & v_3 & v_4\end{pmatrix}\begin{pmatrix}1\\1\\1\\1\end{pmatrix}=0.$$

Konkrétně pro „koktavý“ kód z příkladu 50 je rovnice (73) v maticovém tvaru

$$\begin{pmatrix}1&1&0&0&0&0\\0&0&1&1&0&0\\0&0&0&0&1&1\end{pmatrix}\begin{pmatrix}v_1\\v_2\\v_3\\v_4\\v_5\\v_6\end{pmatrix}=\begin{pmatrix}0\\0\\0\end{pmatrix}, \text{ resp.} \begin{pmatrix}v_1 & v_2 & v_3 & v_4 & v_5 & v_6\end{pmatrix}\begin{pmatrix}1&0&0\\1&0&0\\0&1&0\\0&1&0\\0&0&1\\0&0&1\end{pmatrix}=\begin{pmatrix}0 & 0 & 0\end{pmatrix}.$$

Opakovací kód z příkladu 48 lze zapsat například takto:

$$\begin{pmatrix}1&1&0&0&0\\1&0&1&0&0\\1&0&0&1&0\\1&0&0&0&1\end{pmatrix}\begin{pmatrix}v_1\\v_2\\v_3\\v_4\\v_5\end{pmatrix}=\begin{pmatrix}0\\0\\0\\0\end{pmatrix}, \text{ resp. } \begin{pmatrix}v_1 & v_2 & v_3 & v_4 & v_5\end{pmatrix}\begin{pmatrix}1&1&1&1\\1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix}=\begin{pmatrix}0\\0\\0\\0\\0\end{pmatrix}$$

(samozřejmě že je více možností, jak postupně zapsat, že všechny znaky ve značce jsou stejné).

Z příkladů je dále zřejmé, že:

- počet řádků kontrolní matice je roven počtu kontrolních prvků $r=n-k$,
- počet sloupců kontrolní matice je roven délce značky n,
- k danému kódu může existovat i několik ekvivalentních kontrolních matic, které vyhovují rovnicím (77): například můžeme libovolně zaměňovat pořadí řádků (tomu odpovídá jen pořadí rovnic), u opakovacího kódu existuje více možností zápisu rovnic s ekvivalentním informačním obsahem apod.

Opakovací, paritní a „koktavý“ kód patří do kategorie tzv. lineárních kódů. Podle rovnice (77) má každá značka, která patří do kódu, tu vlastnost, že násobíme-li ji transponovanou kontrolní maticí, vyjde nulový vektor (délky k). Tímto způsobem se dá identifikovat, zda přijatá značka patří do kódu nebo ne. V případě nekódové značky, např. při výskytu chyby na určitých místech, vede daný součin na nenulový vektor, tzv. syndrom značky (podrobnosti viz dále). Syndrom kódového slova je tedy nulový vektor.

Lineární kód je spojen s pojmem lineární vektorový prostor. Každá značka kódu délky n je představována vektorem o n souřadnicích. Všechny značky, které patří do kódu, vyplňují vektorový prostor kódu. Ostatní značky délky n, které do kódu nepatří, tvoří tzv. doplněk vektorového prostoru. Vektory z tohoto doplňkového prostoru jsou generovány kódem, který je doplňkovým k původnímu kódu. Sjednocení obou prostorů je tvořeno všemi existujícími značkami délky n.

Základním znakem lineárního vektorového prostoru je to, že v něm neexistuje vektor, který by nebylo možné odvodit z ostatních vektorů jejich lineárními kombinacemi. Existuje minimální množina vektorů daného prostoru, jejichž lineárními kombinacemi lze vytvořit všechny ostatní vektory prostoru. Tato minimální množina se nazývá báze vektorového prostoru a počet vektorů v bázi je řád vektorového prostoru.

Teorii lineárních vektorových prostorů lze aplikovat na lineární kódy, což jsou blokové kódy splňující níže uvedené podmínky. Dále se budeme zabývat pouze kódy binárními.

Filozofie lineárních kódů.

1. Značku délky n lze chápat jako vektor o n souřadnicích

$$\mathbf{v} = [v_1\ v_2\ \mathbf{K}\ v_n].$$

Souřadnice jsou binární znaky 0 a 1.

2. Součet dvou značek délky n lze chápat jako součet dvou vektorů podle pravidla

$$\mathbf{v} = \mathbf{v}_1 + \mathbf{v}_2 = [v_{11}\ v_{12}\ \mathbf{K}\ v_{1n}] + [v_{21}\ v_{22}\ \mathbf{K}\ v_{2n}] = [v_{11}+v_{21}\ v_{12}+v_{22}\ \mathbf{K}\ v_{1n}+v_{2n}],$$

kde prvky výsledného vektoru vzniknou součtem odpovídajících prvků dílčích vektorů podle pravidel (75).

3. Násobení značky binárním znakem $a \in \{0\ 1\}$ se provede podle pravidel (76) prvek po prvku:

$$a.\mathbf{v} = a.[v_1\ v_2\ \mathbf{K}\ v_n] = [av_1\ av_2\ \mathbf{K}\ av_n].$$

4. Lineární kombinací vektorů $\mathbf{v}_1$ a $\mathbf{v}_2$ se rozumí vektor

$$\mathbf{v} = a_1\mathbf{v}_1 + a_2\mathbf{v}_2,$$

kde a_1 a a_2 jsou libovolné binární znaky z množiny $\{0\ 1\}$. Pak říkáme, že vektory $\mathbf{v}$, $\mathbf{v}_1$ a $\mathbf{v}_2$ jsou lineárně závislé.

Věta: Každé kódové slovo lineárního kódu je tvořeno lineární kombinací jiných kódových slov.

Důsledek: kódovým slovem každého lineárního kódu musí být slovo obsahující samé nuly (vektorový prostor musí obsahovat „počátek“).

Jakou délku má lineární blokový kód (*n*,*k*) a jeho doplňkový kód?

Připomeňme, že délka kódu je počet jeho kódových slov a označení (*n*,*k*) patří blokovému kódu o *k* informačních znacích, které tvoří součást *n*-místné značky (str. 41).

Odpověď dává vzorec (67) na str. 41: počet kódových slov je roven počtu všech možných informačních slov délky *k*, tj.

$$L = 2^k . \tag{78}$$

Všechna ostatní slova délky *n*, která nepatří do kódu, patří do doplňkového kódu. Délka doplňkového kódu musí být

$$\overline{L} = 2^n - L = 2^n - 2^k , \tag{79}$$

neboť celkový počet všech značek délky *n* je 2^n.

s Příklad 51: Paritní kód z příkladu 49.

informační slovo	*paritní kód*	*doplňkový kód*
000	0000	0001
001	0011	0010
010	0101	0100
011	0110	0111
100	1001	1000
101	1010	1011
110	1100	1101
111	1111	1110

Uvedený paritní kód má parametry $n = 4$, $k = 3$. Tudíž jeho délka je $L = 2^3 = 8$. Doplňkový kód má stejnou délku: $\overline{L} = 2^4 - 2^3 = 8$.

þ

s Příklad 52: Opakovací kód o délce značky 3.

informační slovo	*opakovací kód*	*doplňkový kód*
0	000	001
1	111	010
		011
		100
		101
		110

Opakovací kód má parametry $n = 3$, $k = 1$. Tudíž jeho délka je $L = 2^1 = 2$. Doplňkový kód má délku $\overline{L} = 2^3 - 2^1 = 6$.

þ

Poznámky k příkladům 51 a 52:

Kód doplňkový k lineárnímu kódu je často v praxi nepoužitelný. Například doplňkový kód k opakovacímu kódu se nedá použít k prostému kódování. Naproti tomu doplňkový kód ke kódu sudé parity je kód liché parity.

Doplňkový kód k lineárnímu kódu již nemůže být lineární, protože neobsahuje nulovou značku.

Věta: V každém kódu je možné najít několik množin značek, které mají následující vlastnosti:

- Vektory značek z dané množiny jsou lineárně nezávislé.
- Lineárními kombinacemi vektorů z dané množiny lze vytvořit všechny značky kódu.
- V každé množině je stejný počet značek.
- Známe-li jednu množinu, pak ostatní lze získat na základě lineárních kombinací vektorů této množiny.

Pak každá z uvedených množin vektorů může být prohlášena za bázi vektorového prostoru kódu. Znalost báze je užitečná v tom, že si nemusíme pamatovat všechna slova patřící do kódu. Stačí znát bázi, ostatní kódová slova lze získat z báze lineárními kombinacemi. Báze je tedy „zhuštěný“ zápis celého kódu.

Kolik vektorů obsahuje báze lineárního blokového kódu (*n*,*k*) (jaký je řád vektorového prostoru)?

Počet vektorů báze je roven počtu informačních bitů k. Vysvětlení následuje.

Počet všech informačních slov délky *k* je 2^k. Potřebujeme však jen *k* informačních slov, z nichž lze lineárními kombinacemi vytvořit všechna ostatní informační slova. Například bázi informačních slov délky 3 mohou tvořit tři vektory (1 0 0), (0 1 0) a (0 0 1). Protože kód přiřazuje ke každému informačnímu slovu jedinou *r*-tici zabezpečujících bitů, tvoří bázi celého vektorového prostoru kódu právě *k* vektorů.

s Příklad 53: Báze vektorového prostoru paritního kódu (4,3) z příkladu 49.

informační slovo	*kód*	*báze*
000	0000	
001	0011	0011
010	0101	0101
011	0110	
100	1001	1001
101	1010	
110	1100	
111	1111	

Nejsnadněji bázi získáme jako množinu kódových slov, které odpovídají všem informačním slovům, obsahujícím právě jednu jedničku. Další možné báze složíme z vektorů, které jsou tvořeny lineárními kombinacemi vektorů dané báze, např. (0011), (0101), (1010).

þ

Podstata kódování informačních znaků lineárním binárním kódem.

Uvažujme lineární kód (*n*,*k*) a jednu z jeho bází, složenou z *k* vektorů $\{\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_k\}$. Pak libovolný vektor **v** kódového slova lze generovat jako lineární kombinaci vektorů báze, neboli

$$a_1\mathbf{g}_1 + a_2\mathbf{g}_2 + \mathbf{L} + a_k\mathbf{g}_k = \mathbf{v}, \quad a_i \in \{0\,1\}, i = 1,2,...,k\,. \tag{80}$$

Rovnici (80) lze přepsat do maticového tvaru:

$$\begin{pmatrix} a_1 & a_2 & \mathbf{L} & a_k \end{pmatrix} \begin{pmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \\ \mathbb{M} \\ \mathbf{g}_k \end{pmatrix} = \mathbf{v}\,. \tag{81}$$

Binární vektor délky *k* na levé straně rovnice (81) lze považovat za vektor informačního slova **a**. Matice sestavená z *k* vektorů báze, jejichž délka je *n*, má tedy k řádků a *n* sloupců. Nazývá se generující matice kódu **G** (*k* x *n*). Rovnice (81) pak představuje algoritmus kódování: vstupem algoritmu jsou informační slova **a**, výstupem kódová slova **v**:

$$\mathbf{a}.\mathbf{G} = \mathbf{v}\,. \tag{82}$$

s Příklad 54: kód (4,2).

Generující matice kódu nechť je ve tvaru

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}.$$

Skládá se tedy z dvou vektorů báze

$$\mathbf{g}_1 = (1\,0\,1\,1), \quad \mathbf{g}_2 = (0\,1\,1\,1).$$

Libovolné kódové slovo získáme lineárními kombinacemi podle (80):

a_1	a_2	$a_1\mathbf{g}_1 + a_2\mathbf{g}_2$	*poznámka*
0	0	(0 0 0 0)	= **0**
0	1	(0 1 1 1)	= $\mathbf{g}_2$
1	0	(1 0 1 1)	= $\mathbf{g}_1$
1	1	(1 1 0 0)	= $\mathbf{g}_1 + \mathbf{g}_2$

Tyto operace lze jednodušeji vyjádřit pomocí rovnice (82):

$$(a_1\ a_2).\begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix} = (.\ \ .\ \ .\ \ .).$$

k=2 informačních bitů — *matice kxn, tj. 2x4* — *značka délky n = 4*

þ

Zabezpečovací schopnost lineárního kódu.

Na str. 41 jsme definovali pojmy Hammingova váha značky blokového kódu a Hammingova vzdálenost dvou značek a minimální vzdálenost kódu. Na základě minimální vzdálenosti pak lze usuzovat na schopnost kódu detekovat nebo opravovat *t*-násobné chyby (vzorce 68 a 69).

Protože lineární kód je zároveň blokovým kódem, platí tyto principy i pro něj. Linearita kódu však navíc umožňuje snadný výpočet minimální vzdálenosti kódu.

Věta: Minimální vzdálenost lineárního kódu je rovna minimální Hammingově váze všech kódových značek s výjimkou nulové značky.

Jde o důsledek linearity kódu. Jestliže totiž dvě kódová slova $\mathbf{v}_1$ a $\mathbf{v}_2$ mají Hammingovu vzdálenost $d(\mathbf{v}_1, \mathbf{v}_2)$, pak kódové slovo $\mathbf{v}_1+\mathbf{v}_2$ má Hammingovu váhu $w(\mathbf{v}_1+\mathbf{v}_2) = d(\mathbf{v}_1, \mathbf{v}_2)$. Proto minimální ze všech vzdáleností kódových slov mezi sebou je stejná jako minimální váha nenulového slova.

s Příklad 55: kód (4,2) z příkladu 54.

Kódové slovo o minimální váze je (1 1 0 0). Proto

$$d_{\min} = w_{\min} = 2.$$

Kód je schopen pouze stoprocentně detekovat jednoduché chyby.

þ

Věty o manipulacích s prvky generující matice, které nemění zabezpečovací schopnost kódu.

Uvažujme lineární kód o generující matici **G**. Připomeňme, že každý řádek této matice je vektorem báze daného vektorového prostoru kódu. Zabezpečovací schopnost kódu se evidentně nezmění,

zaměníme-li pouze pořadí zápisu jednotlivých vektorů báze do matice. Dále víme, že lineárními kombinacemi vektorů báze můžeme přecházet na jiné báze téhož kódu. Tedy:

Vzájemné přehazování řádků generující matice nebo náhrada řádku součty s dalšími řádky nemají vliv na zabezpečovací schopnost kódu.

Dále je zřejmé, že jestliže přehodíme v generující matici libovolné dva sloupce, zůstane zachována Hammingova váha všech vektorů v řádcích, a tedy i Hammingova váha všech kódových slov, které mohou vzniknout lineárními kombinacemi bázových vektorů. Důsledkem je to, že touto operací se nezmění minimální vzdálenost kódu. Tedy:

Vzájemné přehazování sloupců generující matice nemá vliv na zabezpečovací schopnost kódu.

Výše uvedenými operacemi vznikají generující matice různých kódů, které jsou však ekvivalentní vzhledem k svým zabezpečovacím schopnostem.

Systematický lineární kód.

Na str. 40 byl vysvětlen pojem „systematický blokový kód“: kódování informačního slova je provedeno tak, že se k němu pouze připojí zabezpečující část.
Je otázka, jak potom musí vypadat generující matice systematického lineárního kódu.

s Příklad 56: „nesystematický“ a systematický kód.

Generující matice kódu (6,3) je

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

Jedná se o systematický kód?

Informační slova musí mít délku 3. Například slovo

(1 0 1)

je zakódováno takto:

$$(1\,0\,1)\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} = (1\,1\,0\,0\,1\,1), \text{ obecně}$$

$$(a_1\ a_2\ a_3)\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} = (a_1\ a_1\ a_2\ a_2\ a_3\ a_3).$$

Vidíme, že se jedná o „koktavý“ kód z příkladu 50, který není systematický.

Generující matice jiného kódu (6,3) je

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}.$$

Slovo (1 0 1) je nyní zakódováno takto:

$$(1\,0\,1)\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} = (1\,0\,1\,1\,1\,0), \text{ obecně}$$

$$(a_1\ a_2\ a_3)\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} = (a_1\ a_2\ a_3\ a_3\ a_1\ a_2).$$

Tento kód je systematický.

þ

Z příkladu 56 je zřejmá platnost následujícího tvrzení.
Systematický lineární kód má generující matici ve tvaru

$$\mathbf{G}_{k,n} = \left(\mathbf{I}_{k,k} \mid \mathbf{P}_{k,n-k}\right), \tag{83}$$

kde $\mathbf{I}_{k,k}$ je jednotková matice o k řádcích a sloupcích a $\mathbf{P}_{k,n-k}$ je matice o *k* řádcích a *n-k* sloupcích.
Lze dokázat, že generující matici libovolného lineárního kódu lze výše uvedenými operacemi s řádky a sloupci, které nemají vliv na zabezpečovací schopnosti kódu, transformovat na matici ekvivalentního systematického kódu [12].

s Příklad 57: Hledání ekvivalentního systematického kódu.

V příkladu 56 je uvedena generující matice nesystematického kódu:

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

Postupně změníme pořadí jednotlivých sloupců takto: 1 3 5 6 2 4. Získáme matici ekvivalentního systematického kódu, která byla rovněž uvedena v příkladu 56:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}.$$

þ

Souvislosti mezi generující a kontrolní maticí lineárního kódu.
Na str. 49 je definována kontrolní matice lineárního kódu **K**. Kód má současně i generující matici **G**. Vzniká tedy otázka, jak lze z kontrolní matice získat matici generující a naopak.
Dané matice jsou definovány rovnicemi (77) a (82):

$$\mathbf{v}.\mathbf{H}^t = \mathbf{0},$$

$$\mathbf{a}.\mathbf{G} = \mathbf{v},$$

které platí pro libovolnou značku kódu **v** a pro libovolné informační slovo **a**.
Násobíme-li druhou rovnici zprava transponovanou kontrolní maticí, dostaneme
$\mathbf{a}.\mathbf{G}.\mathbf{H}^t = \mathbf{0}$.
Má-li rovnice platit pro všechna informační slova **a**, pak musí platit

$$\mathbf{G}.\mathbf{H}^t = \mathbf{0}. \tag{84}$$

Generující matice **G** sestává z *k* vektorů báze prostoru lineárního kódu:

$$\mathbf{G} = \begin{pmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \\ \mathbf{M} \\ \mathbf{g}_k \end{pmatrix} = \begin{pmatrix} g_{11} & g_{12} & \mathbf{L} & g_{1n} \\ g_{21} & g_{22} & \mathbf{L} & g_{2n} \\ \mathbf{M} & \mathbf{M} & \mathbf{M} & \mathbf{M} \\ g_{k1} & g_{k2} & \mathbf{L} & g_{kn} \end{pmatrix}. \tag{85}$$

Obdobně si můžeme představit kontrolní matici **H**, která má rozměr *r* x *n* = (*n-k*) x *n*, sestavenou z *r* vektorů typu **h**:

$$\mathbf{H} = \begin{pmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{M} \\ \mathbf{h}_r \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & \mathbf{L} & h_{1n} \\ h_{21} & h_{22} & \mathbf{L} & h_{2n} \\ \mathbf{M} & \mathbf{M} & \mathbf{M} & \mathbf{M} \\ h_{r1} & h_{r2} & \mathbf{L} & h_{rn} \end{pmatrix}. \tag{86}$$

Definujeme-li skalární součin dvou vektorů **g** a **h** délky *n* předpisem

$$\mathbf{g}.\mathbf{h} = \left(g_1 \quad g_2 \quad \mathbf{L} \quad g_n\right)\left(h_1 \quad h_2 \quad \mathbf{L} \quad h_n\right) = g_1h_1 + g_2h_2 + \ldots + g_nh_n,$$

kde dílčí součiny jsou realizovány podle pravidel „modulo 2“, viz rovnice (76), rovnici (84) můžeme pomocí skalárních součinů přepsat takto:

$$\mathbf{G}.\mathbf{H}^t = \begin{pmatrix} \mathbf{g}_1.\mathbf{h}_1 & \mathbf{g}_1.\mathbf{h}_2 & \mathbf{L} & \mathbf{g}_1.\mathbf{h}_r \\ \mathbf{g}_2.\mathbf{h}_1 & \mathbf{g}_2.\mathbf{h}_2 & \mathbf{L} & \mathbf{g}_2.\mathbf{h}_r \\ \mathbf{M} & \mathbf{M} & \mathbf{M} & \mathbf{M} \\ \mathbf{g}_k.\mathbf{h}_1 & \mathbf{g}_k.\mathbf{h}_2 & \mathbf{L} & \mathbf{g}_k.\mathbf{h}_r \end{pmatrix} = \mathbf{0}. \qquad (87)$$

Obdobně jako generující matice **G** je matice lineárního kódu (*n*,*k*), můžeme kontrolní matici **K** chápat jako generující matici jiného lineárního kódu (*r*,*n*) = (*n*-*k*,*n*). Tento nový kód je charakterizován *r* bázovými vektory, jejichž skalární součiny se všemi bázovými vektory původního kódu jsou nulové. Říkáme, že daný kód je duální k původnímu kódu.
Poznámka: duální kód je něco jiného než doplňkový kód.
Pak analogicky k rovnici (82) platí

$$\mathbf{a}_d.\mathbf{H} = \mathbf{v}_d, \qquad (88)$$

kde $\mathbf{a}_d$ a $\mathbf{v}_d$ jsou informační slovo délky *r* a kódové slovo délky *n* duálního kódu.

s Příklad 58: Duální kód k paritnímu kódu (4,3) z příkladu 49.

V příkladu 53 je odvozena báze prostoru kódu, z níž sestavíme generující matici:

$$\mathbf{G} = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix}.$$

Kontrolní matice kódu bude mít 1 řádek a 4 sloupce (má být rozměru *r* x *n*). Je tedy tvořena jediným vektorem báze duálního kódu

$$\mathbf{H} = \mathbf{h} = (h_1\ h_2\ h_3\ h_4).$$

Skalární součiny tohoto vektoru se všemi bázovými vektory v generující matici musí být nulové. Z toho vyplývají 3 rovnice:

$$h_3 + h_4 = 0,$$
$$h_2 + h_4 = 0,$$
$$h_1 + h_4 = 0$$

s řešením

$$h_1 = h_2 = h_3 = h_4.$$

Protože vektorem báze nemůže být nulový vektor, jediné řešení je

$$\mathbf{H} = (1\,1\,1\,1).$$

Chápeme-li **H** jako generující matici duálního kódu, pak je možno tímto kódem kódovat jen jeden informační prvek podle rovnice (88), která v tomto případě vede na výsledek

$$1.(1\ \ 1\ \ 1\ \ 1) = (1\ \ 1\ \ 1\ \ 1),$$
$$0.(1\ \ 1\ \ 1\ \ 1) = (0\ \ 0\ \ 0\ \ 0).$$

Dospěli jsme k závěru, že duální kód k paritnímu kódu (4,3) je opakovací kód s délkou značky 4.

þ

s Příklad 59: Duální kód ke „koktavému“ kódu (4,3) z příkladu 50.

Na str. 49 byla odvozena kontrolní matice tohoto kódu:

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

a to na základě struktury kódového slova, kde se každý informační znak objevuje dvakrát po sobě.
Kódovací předpis tedy zapíšeme pomocí generující matice takto:

$$(a_1\ \ a_2\ \ a_3)\mathbf{G} = (a_1\ \ a_1\ \ a_2\ \ a_2\ \ a_3\ \ a_3).$$

Je zřejmé, že rovnici vyhovuje matice **G**, která je identická s kontrolní maticí, tedy

$$\mathbf{G} = \mathbf{H}.$$

Dospěli jsme k překvapivému závěru, že uvedený kód má generující matici, která je totožná s kontrolní maticí, a že tudíž tento kód je zároveň sám sobě duálním kódem.

þ

Poznámky a shrnutí:

- kontrolní matice kódu je generující maticí jeho duálního kódu a naopak,
- skalární součin každé značky kódu s každou značkou jeho duálního kódu je nulový,
- některé lineární kódy jsou samoduální, tj. mají shodnou generující a kontrolní matici, takže kód je zároveň i duálním kódem,
- kód a duální kód tedy mohou mít společné značky (vždy mají společnou nulovou značku) na rozdíl od kódu a doplňkového kódu.

Vztah mezi generující a kontrolní maticí systematického kódu.
Uvažujme systematický kód (n,k) o generující matici, která má tvar (83):

$$\mathbf{G}_{k,n} = \left(\mathbf{I}_{k,k} \mid \mathbf{P}_{k,r}\right).$$

Pak kontrolní matice tohoto kódu je ve tvaru

$$\mathbf{H}_{r,n} = \left(\mathbf{P}^{t}_{r,k} \mid \mathbf{I}_{r,r}\right), \tag{89}$$

kde $\mathbf{I}_{r,r}$ je jednotková matice o r řádcích a r sloupcích.
Důkaz je možno nalézt např. v [4].

s Příklad 60: Kontrolní matice systematického kódu (6,3) z příkladu 57.

V příkladu 57 je uvedena generující matice

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

systematického kódu. V souladu s vzorcem (83) si tuto matici představíme v následující struktuře:

$$\left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{array}\right).$$

$\mathbf{I}_{3,3}$ $\mathbf{P}_{3,3}$

Pak kontrolní matice bude podle (89)

$$\mathbf{H} = \left(\begin{array}{ccc|ccc} 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right).$$

$\mathbf{P}^{t}_{3,3}$ $\mathbf{I}_{3,3}$

þ

Dekódování lineárních kódů.
V kapitole „Obecný pohled na bezpečnostní kódování a dekódování“ bylo v příkladu 47 ukázáno, že díky „chytrému“ dekódovacímu algoritmu může dekodér kromě těch chyb, které z principu detekuje či koriguje na základě hledání nejbližších kódových slov (viz vzorce 68 a 69), detekovat nebo i korigovat i některé další chyby. V tomto smyslu „nejchytřejším“ způsobem dekódování lineárních kódů je tzv. standardní dekódování. Na opačném konci dekódovacích technik stojí klasické dekódování popsané na str. 42.

K dekódování lineárních kódů se tedy používají tyto přístupy:

1. Standardní dekódování
2. Standardní dekódování urychlené s využitím syndromů.
3. Dekódování na principu minimální Hammingovy vzdálenosti pomocí syndromů.

Standardní dekódování je nejúčinnější způsob dekódování, kdy dekodér opraví maximální množství chyb. Je ale relativně pomalé, protože dekodér hledá v tabulkách , které jsou pro dlouhé kódy značně rozsáhlé.
Toto hledání lze urychlit, známe-li syndrom přijatého slova, což je součin tohoto slova a transponované kontrolní matice kódu.
Třetí metoda je schopna spolehlivě opravit pouze tolik chyb, kolik popisuje vzorec (69). Opravuje se podle principu minimální Hammingovy vzdálenosti přijaté značky od značek kódu. Dekódování je však velmi rychlé.
U některých kódů metody 2 a 3 splývají.

Standardní dekódování
Uvažujme lineární kód k o délce kódových značek l. Dále uvažujme slovo $\mathbf{w}$ téže délky l, které může, ale nemusí patřit do kódu.
Třída slova $\mathbf{w}$ *podle kódu* k je množina slov, která vznikne sečtením slova $\mathbf{w}$ postupně se všemi kódovými slovy kódu k.

s Příklad 61: Třídy slov podle binárního paritního kódu (4,3) z příkladu 49.

Kód z příkladu 49 obsahuje tato kódová slova:
k = {0000, 0011, 0101, 0110, 1001, 1010, 1100, 1111}.
Slovo $\mathbf{w}_1$ = (1000) do kódu nepatří.
Třída slova (1000) podle kódu k bude podle definice
(1000) + k = {1000, 1011, 1101, 1110, 0001, 0010, 0100, 0111}.

Slovo $\mathbf{w}_2$ = (0100) do kódu rovněž nepatří.
Třída slova (0100) podle kódu k bude
(0100) + k = {0100, 0111, 0001, 0010, 1101, 1110, 1000, 1011}.
Dostali jsme stejnou třídu jako u slova (1000), i když v jiném pořadí zápisu.

Slovo $\mathbf{w}_3$ = (1001) do kódu patří.
Třída slova (1001) podle kódu k bude
(1001) + k = {1001, 1010, 1100, 1111, 0000, 0011, 0101, 0110}.
Dostali jsme opět slova kódu, i když v jiném pořadí zápisu.

þ
Zobecnění z příkladu 61: třída kódového slova podle kódu je původní kód. Vysvětlení: sečteme-li kódové slovo s tímtéž kódovým slovem, dostaneme nulové slovo. Sečteme-li kódové slovo s jiným kódovým slovem, dostaneme jiné kódové slovo. Touto operací tedy nikdy nevybočíme z vektorového prostoru kódu.
Naopak třída nekódového slova podle kódu nemůže být původním kódem: sečtením kódového a nekódového slova vznikne slovo z doplňkového kódového prostoru.
Mohou existovat shodné třídy různých nekódových slov. V příkladu 61 se o tom můžeme přesvědčit porovnáním tříd slov $\mathbf{w}_1$ a $\mathbf{w}_2$.
Nemohou existovat dvě různé třídy, které by obsahovaly některá shodná slova. Důkaz je uveden např. v [4].

Jaký je celkový počet různých tříd podle daného kódu (n,k)?

$$2^{n-k} = 2^r . \qquad (90)$$

Počet slov v každé třídě je totiž stejný jako je počet kódových slov, což je 2^k. Počet všech možných slov délky n je 2^n. Počet slov v třídě krát počet tříd musí být 2^n, z čehož vyplývá (90).

s Příklad 62: Počet různých tříd slov podle binárního paritního kódu (4,3) z příkladu 61.

$n = 4, k = 3 \Rightarrow$ počet různých tříd = $2^{(4-3)} = 2$.
Jedna z tříd je vždy samotný kód:
k = {0000, 0011, 0101, 0110, 1001, 1010, 1100, 1111}.
Druhá třída je vypočtena v příkladu 61:
(1000) + k = {1000, 1011, 1101, 1110, 0001, 0010, 0100, 0111}.
Tato třída je společná pro všechna nekódová slova, kterých je $2^n - 2^k = 8$. Jedná se vlastně o kód liché parity.

þ

Reprezentant třídy je slovo o nejmenší Hammingově váze ze všech slov třídy. Pokud je takových slov více, volíme jako reprezentanta jedno z nich.

Důsledek: reprezentantem kódu je nulové slovo.

s Příklad 63: Paritní kód (4,3) z příkladu 61.

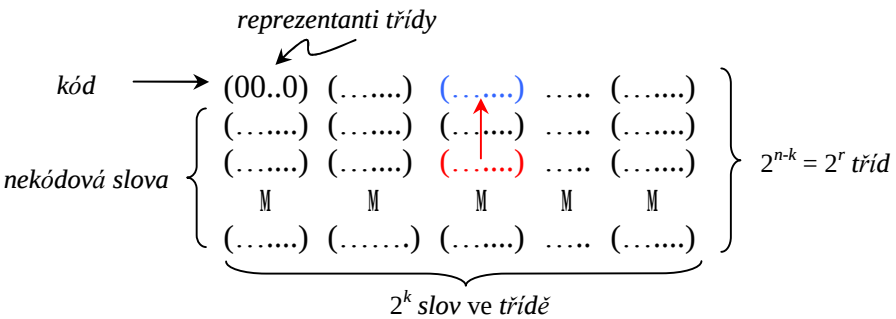
Podle příkladu 62 má tento kód dvě třídy:
{0000, 0011, 0101, 0110, 1001, 1010, 1100, 1111},
{1000, 1011, 1101, 1110, 0001, 0010, 0100, 0111}.
Reprezentantem první třídy je nulové slovo (0000). Reprezentantem druhé třídy může být jedno ze slov (1000), (0001), (0010) nebo (0100).

þ

Algoritmus standardního dekódování.
Přijaté slovo nechť je **w**. Vyhledáme třídu, v níž se slovo nachází. Pak od přijatého slova odečteme chybu, za kterou se považuje reprezentant třídy. Získáme tím kódové slovo jako výsledek standardního dekódování.
Pozn.: odečítání je zde zaměnitelné za přičítání.
Algoritmus lze jednoduše vyjádřit vzorcem

$$d(\mathbf{w}) = \mathbf{w} - [\text{reprezentant třídy } \mathbf{w} + k\ ], \qquad (91)$$

kde d je v souladu s (70) operátor dekódování.
Postup dekódování lze graficky znázornit pomocí tzv. Slepianova standardního rozmístění. Jde o 2^r řádků, v každém řádku jsou uvedena slova jedné z tříd kódu. Pořadí jednotlivých řádků je libovolné až na to, že na prvním řádku musí být uvedena třída kódového slova, tj. kód. Pořadí slov v každém řádku je takové, že první zleva musí být reprezentant třídy. Následují slova v pořadí, které odpovídá kódovým slovům v prvním řádku, k nimž se přičetl reprezentant třídy. Tím se vytvoří schéma obsahující 2^r řádků a 2^k sloupců podle obrázku.



Přijaté slovo **w** nejprve lokalizujeme v tabulce a poté jej dekódujeme tak, že mu přiřadíme kódové slovo, které leží nahoře v prvním řádku a v témže sloupci jako je přijaté slovo. Tento postup je v souladu s rovnici (91), protože všechna slova dané třídy jsou vzdálena od příslušných kódových slov v těchže sloupcích právě o reprezentanta třídy, jemuž přísluší nulové kódové slovo.
Protože reprezentant třídy má ze všech slov třídy nejmenší Hammingovu váhu, znamená tento algoritmus dekódování přijatého slova na nejbližší kódové slovo.
Pokud je možno za reprezentanta třídy vybrat slovo z více možností, můžeme touto volbou ovlivnit charakter dekódovacího procesu. Ukážeme to na příkladu.

s Příklad 64: Paritní kód (4,3) z příkladu 61.

Tento kód má minimální vzdálenost d_{min} = 2, takže podle strategie minimální vzdálenosti přijatého slova od kódových značek (viz vzorce 68 a 69) je schopen pouze jednoznačné detekce, nikoliv korekce chyby. Ukážeme, že standardní dekódování dokáže některé (ne všechny) chyby i korigovat.
Třída nekódových slov obsahuje celkem 4 slova o minimální Hammingově váze 1. Zvolíme-li za reprezentanta například slovo (1000), vyjde Slepianovo rozmístění následovně:

(0000) (0011) (0101) (0110) (1001) (1010) (1100) (1111)
(1000) (1011) (1101) (1110) (0001) (0010) (0100) (0111)

Přijmeme-li například slovo (1101), dekódujeme jej jako (0101). Dekódování tedy bude správné, jestliže došlo k chybě na 1. místě. Z tabulky vyplývá, že dekodér spolehlivě odstraní jednoduchou chybu vždy, pokud se objeví na prvním místě. Ostatní jednoduché chyby opraví nesprávně.
Pokud za reprezentanta zvolíme jiné slovo, např. (0010), vyjde jiné Slepianovo rozmístění:

(0000) (0011) (0101) (0110) (1001) (1010) (1100) (1111)
(0010) (0001) (0111) (0100) (1011) (1000) (1110) (1101)

Tentokrát je přijaté slovo dekódováno jako (1111). Dekodér spolehlivě odstraní jednoduchou chybu pouze objeví-li se na 3. místě.

þ

Poznatky z příkladu 64 lze zobecnit:
Standardní dekódování spolehlivě opravuje právě ta chybová slova, která jsme zvolili za reprezentanty tříd.
Chybovým slovem se přitom rozumí rozdíl mezi vyslaným kódovým slovem a přijatým slovem.

V [4] je dokázáno toto tvrzení:
Každé standardní dekódování d je optimální v tom smyslu, že žádné jiné dekódování neopravuje větší množinu chybových slov než d.

Pro kódy s dlouhými značkami je standardní dekódování technicky těžko realizovatelné. Například pro kódy o n = 64, které se často používají, vychází $2^n = 2^{64} \approx 10^{19}$ 64-místných slov ze všech existujících tříd, která by se musela prohledávat.
Hledání reprezentanta třídy, v níž leží přijaté slovo, lze urychlit, známe-li tzv. syndrom (příznak) přijatého slova. Z rovnice (77) vyplývá, že součin kódového slova a transponované kontrolní matice kódu je nulový vektor délky r. V případě, že slovo **w** nepatří do kódu, tento vektor již bude nenulový. Nazývá se syndrom **s** slova **w**:

$$\mathbf{s} = \mathbf{w}.\mathbf{H}^t . \qquad (92)$$

Lze dokázat, že všechna slova patřící do stejné třídy mají stejný syndrom. Uvažujme dvě slova $\mathbf{w}_1$ a $\mathbf{w}_2$ téže třídy. Tato slova vznikla přičtením reprezentanta w_R třídy ke kódovým slovům $\mathbf{v}_1$ a $\mathbf{v}_2$, která leží ve stejných sloupcích Slepianova rozložení jako slova $\mathbf{w}_1$ a $\mathbf{w}_2$:

$$\mathbf{w}_1 = \mathbf{v}_1 + \mathbf{w}_R\,,\ \mathbf{w}_2 = \mathbf{v}_2 + \mathbf{w}_R\,. \tag{93}$$

Vypočítáme-li syndromy $\mathbf{s}_1$ a $\mathbf{s}_2$ slov $\mathbf{w}_1$ a $\mathbf{w}_2$ podle (92), pak s využitím (77) vyjde

$$\mathbf{s}_1 = (\mathbf{v}_1 + \mathbf{w}_R).\mathbf{H}^t = \mathbf{v}_1.\mathbf{H}^t + \mathbf{w}_R.\mathbf{H}^t = \mathbf{w}_R.\mathbf{H}^t\,,$$
$$\mathbf{s}_2 = (\mathbf{v}_2 + \mathbf{w}_R).\mathbf{H}^t = \mathbf{v}_2.\mathbf{H}^t + \mathbf{w}_R.\mathbf{H}^t = \mathbf{w}_R.\mathbf{H}^t\,.$$

Znamená to, že syndromy všech slov třídy jsou stejné a jsou rovny syndromu reprezentanta třídy. Stačí tedy z kontrolní matice kódu vypočítat syndrom přijatého slova a nalézt reprezentanta o daném syndromu. Dekódování se pak provede přičtením reprezentanta k přijatému slovu podle (91). Daný postup je možný díky faktu, že reprezentanti různých tříd mají různé syndromy. Důkaz je možné nalézt např. v [4].

Standardní dekódování urychlené s využitím syndromů.

Ke každé třídě kódu stanovíme jejího reprezentanta a k reprezentantovi vypočteme jeho syndrom. Tím si připravíme data pro dekódování podle tabulky:

třída číslo	1	2	3	…	m
reprezentant	e_1	e_2	e_3	…	e_m
syndrom	s_1	s_2	s_3	…	s_m

Dekódování přijatého vektoru w pak probíhá v následujících krocích:

a) Přijmeme vektor w.
b) Zjistíme jeho syndrom si přes kontrolní matici.
c) V tabulce vyhledáme reprezentanta e_i se syndromem s_i.
d) Dekódujeme předpisem

$$d(w) = w \,\mathbb{m}\, e_i\,.$$

s Příklad 65: Paritní kód (4,3) z příkladu 64.

Kontrolní matice tohoto kódu je (1 1 1 1).
Uvažujme přijaté slovo w = (0111). Pak standardní dekódování podle Slepianova rozmístění z příkladu 64 vyjde takto:

(0000) (0011) (0101) (0110) (1001) (1010) (1100) (1111)
(1000) (1011) (1101) (1110) (0001) (0010) (0100) (0111)

$d(0111) = (1111)$.

Dekódování pomocí syndromů:

třída číslo	1	2
reprezentant	(0000)	(1000)
syndrom	0	1

a) w = (0111)

b) $s = w.\mathbf{H}^t = (0111)\begin{pmatrix}1\\1\\1\\1\end{pmatrix} = 1$.

c) $e = (1000)$

d) $d(w) = w - e = (0111) - (1000) = (1111)$.

þ

Příklad kódu s rychlým dekódováním: Hammingův (7,4) kód.

Sloupce kontrolní matice tohoto kódu jsou tvořeny binárními rozvoji čísel 0, 1, 2, …, 7:

$$\mathbf{H} = \begin{pmatrix}0&0&0&1&1&1&1\\0&1&1&0&0&1&1\\1&0&1&0&1&0&1\end{pmatrix} \Big\} n-k=r=3 \Rightarrow k=4 \text{ (4 datové a 3 zabezpečovací bity).} \quad (94)$$

1 4 4 4 4 2 4 4 4 4 3

$n = 7$

Najdeme reprezentanty tříd kódu:

e_0 = (0000000) má syndrom (000), je tedy reprezentantem kódu.

e_1 = (1000000) má syndrom (001), neleží tedy ve třídě $e_0 + K$, je tedy reprezentantem jiné třídy.

e_2 = (0100000) má syndrom (010), je odlišný od předchozích syndromů, volíme jej za reprezentanta další třídy.

…

Dvě slova e_i s jedničkou na i-tém místě pro různá i mají navzájem různé syndromy. Syndromem je i-tý sloupec matice **H**, tj. binární rozvoj čísla i. Těchto různých slov e_i je celkem 8. Protože kód má 2^{7-4}=8 tříd, našli jsme všechny reprezentanty:

třída číslo	1	2	3	4	5	6	7	8
reprezentant	e_0	e_1	e_2	e_3	e_4	e_5	e_6	e_7
syndrom	(000)	(001)	(010)	(011)	(100)	(101)	(110)	(111)

Dekódování Hammingova kódu:

a) Přijmeme slovo w.

b) K přijatému slovu vypočteme syndrom, který bude binárním rozvojem některého z čísel i = 0, 1, 2,.., 7.

c), d) Dekódujeme podle předpisu

$d(w) = w - e_i$,

tj. opravíme i-tý znak v přijatém slově.

Dekódování bude správné, nastala-li maximálně jednoduchá chyba (viz poznámka na str. 60: „Standardní dekódování spolehlivě opravuje právě ta chybová slova, která jsme zvolili za reprezentanty tříd.“). Jinými slovy, tento kód spolehlivě opravuje všechny jednoduché chyby. Rychlé dekódování pomocí syndromů je ekvivalentní k dekódování podle kritéria minimální Hammingovy vzdálenosti. Minimální vzdálenost kódu d_{min} je 3.

s Příklad 66: Hammingův (7,4) kód.

Přijaté slovo je

w = (1010111).

Jeho syndrom je

$$w.\mathbf{H}^t = (1010111)\begin{pmatrix}1&0&0\\0&1&0\\1&1&0\\0&0&1\\1&0&1\\0&1&1\\1&1&1\end{pmatrix} = (110)\text{, což odpovídá číslu 6.}$$

Opravíme tedy 6. znak zleva:
$d(1010111) = (1010101)$.

þ

Struktura kódového slova Hammingova (7,4) kódu, jeho kodér a dekodér.

Předpokládejme tuto strukturu kódového slova:

(c_1 c_2 a_3 c_4 a_5 a_6 a_7),

kde symboly *c*/*a* jsou označeny kontrolní/datové bity (uvažujme, že kód je nesystematický).
Syndrom každého kódového slova musí být nulový:

$$\begin{pmatrix}c_1 & c_2 & a_3 & c_4 & a_5 & a_6 & a_7\end{pmatrix}\begin{pmatrix}1&0&0\\0&1&0\\1&1&0\\0&0&1\\1&0&1\\0&1&1\\1&1&1\end{pmatrix} = \begin{pmatrix}0 & 0 & 0\end{pmatrix}.$$

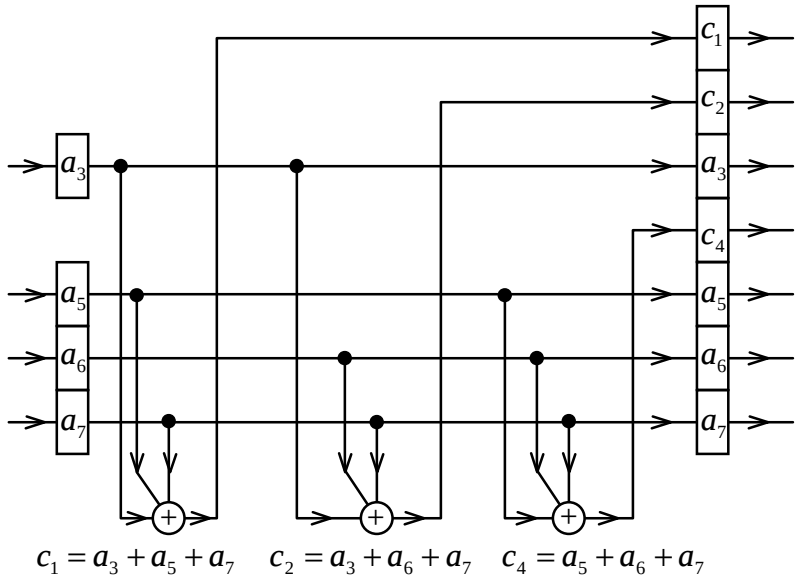
Z maticové rovnice vyplývají tři rovnice skalární pro 3 složky syndromu $s = (s_1\ s_2\ s_3)$. Jsou to **rovnice dekodéru**, který potřebuje počítat složky syndromu z přijatého slova.

$$\begin{aligned} s_1 &= c_1 + a_3 + a_5 + a_7 = 0\,,\\ s_2 &= c_2 + a_3 + a_6 + a_7 = 0\,,\\ s_3 &= c_4 + a_5 + a_6 + a_7 = 0\,. \end{aligned} \qquad (95)$$

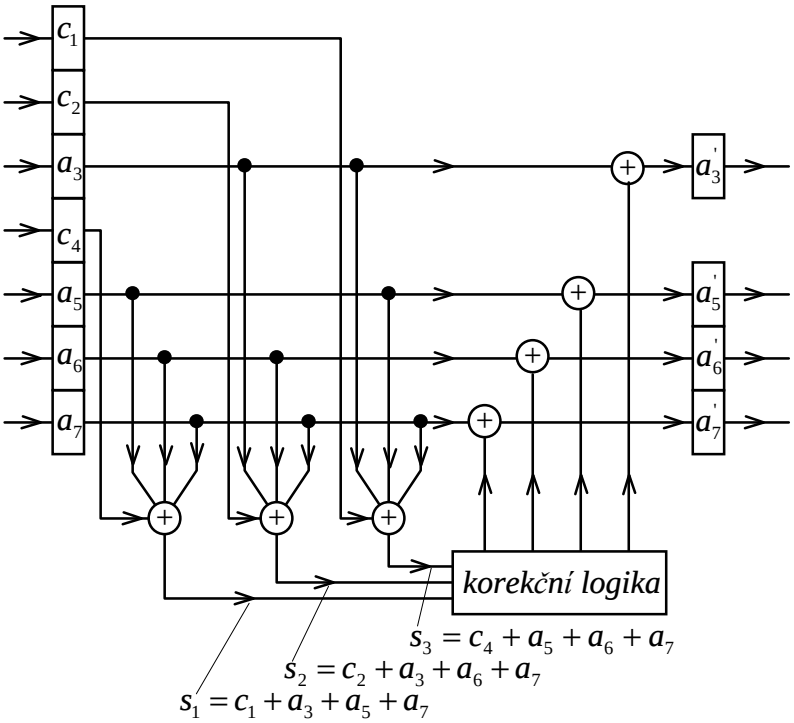
Nuly na pravých stranách jsou pouze při příjmu kódových slov. V případě chyby jsou složky syndromu obecně nenulové a slouží k opravě chyby podle výše uvedeného mechanismu.
S využitím pravidla „+1 = -1“ můžeme psát vzorce pro výpočet kontrolních bitů z datových bitů (**vzorce kodéru**):

$$\begin{aligned} c_1 &= a_3 + a_5 + a_7\,,\\ c_2 &= a_3 + a_6 + a_7\,,\\ c_4 &= a_5 + a_6 + a_7\,. \end{aligned} \qquad (96)$$

Na obrázku jsou principiální schémata kodéru a dekodéru Hammingova kódu, která vyplývají z vzorců (95) a (96).



kodér Hammingova kódu (7,4)



dekodér Hammingova kódu (7,4)

Hammingových kódů může být celá řada. Abychom mohli definovat, co je Hammingův kód, musíme nejprve objasnit pojem „perfektní kód“.

Definice perfektního kódu.

Lineární kód je **perfektní pro *t*-násobné opravy**, jestliže současně:

- je schopen opravit každou *t*-násobnou chybu v přijatém slově,
- má nejmenší možnou redundanci (největší možný informační poměr) ze všech kódů, schopných *t*-násobné opravy (při stejných délkách kódů).

Definice Hammingova kódu kódu.

Hammingův kód je lineární binární kód perfektní pro **jednoduché** opravy.

Pro Hammingovy kódy platí následující věta:

Binární kód se nazývá Hammingův, jestliže má kontrolní matici, jejíž sloupce jsou **všechna nenulová** slova dané délky $n\text{-}k=r$ a žádné z nich se neopakuje. Tento kód pak opravuje všechny jednoduché chyby.

Příkladem je kód (7,4) na str. 62.

Věta poskytuje konkrétní návod, jak sestrojit kód, který opravuje jednoduché chyby a má při daném počtu $r=n\text{-}k$ kontrolních bitů co nejmenší redundanci (co nejvíce informačních znaků):

Za kontrolní matici volíme takovou matici, jejíž sloupce jsou všechna nenulová slova délky r.

Počet sloupců kontrolní matice je n. Současně to má být počet nenulových slov délky r, tedy

$$n = 2^r - 1 = 2^{n-k} - 1. \qquad (97)$$

Vzorec (97) udává vztah mezi počtem datových a kontrolních bitů Hammingova kódu, který nemůže být libovolný. Možná řešení pro celá čísla r, k, n jsou v tabulce:

r	n	k	$R=k/n$
2	3	1	0,333
3	7	4	0,571
4	15	11	0,733
5	31	26	0,839
6	63	57	0,905
6	6	66	6

s Příklad 67: Hammingův (15,11) kód.

Nalezněte kontrolní matici Hammingova kódu, který by zabezpečoval 11-tibitová data.

$k = 11 \Rightarrow n = 15 \Rightarrow$ kód (15,11).

$$\mathbf{H} = \underbrace{\begin{pmatrix} 0&0&0&0&0&0&0&1&1&1&1&1&1&1&1 \\ 0&0&0&1&1&1&1&0&0&0&0&1&1&1&1 \\ 0&1&1&0&0&1&1&0&0&1&1&0&0&1&1 \\ 1&0&1&0&1&0&1&0&1&0&1&0&1&0&1 \end{pmatrix}}_{n=15} \Big\} n-k=r=4$$

þ

Další vlastnosti perfektních kódů.

Platí následující věta:

Lineární kód je **perfektní pro t-násobné opravy**, jestliže reprezentanti jeho tříd tvoří množinu všech slov váhy $\le t$.

s Příklad 68: Hammingův kód.

Hammingův kód má reprezentanty o vahách 0 a 1, je tudíž perfektní pro jednoduché opravy ($t = 1$).

þ

s Příklad 69: Opakovací kód.

Opakovací kód délky $n = 2t+1$ je perfektní pro t-násobné opravy. Ověřte pro $t = 1$ a 2.
Uvědomme si, že: Opakovací kód opravující jednoduchou chybu je kód o $n = 3$ a $k = 1$ a jeho informační poměr je $R = k/n = 0{,}333$. Dalším perfektním kódem je Hammingův kód o $n = 3$ a $k = 1$ o stejném informačním poměru.
þ

Věta Tietäväinenova a Van Lintova:

Jediné netriviální perfektní kódy jsou tyto:

a) Hammingovy kódy pro jednoduché chyby,
b) Golayův kód pro trojnásobné chyby a kódy s ním ekvivalentní,
c) Opakovací kód délky $n = 2t+1$ pro t-násobné chyby, kde $t = 1, 2, 3,..$.

Golayův kód
[Golejův].
Je to systematický kód. Nejznámější jsou dvě varianty:
G_{23}: $n = 23$, $k = 12$, $r = 11$.
G_{24}: je to kód G_{23} doplněný mechanismem celkové kontroly parity.

Generující matice:

$$\mathbf{G}_{23} = \left(\mathbf{I} \left| \frac{\mathbf{B}}{11...11} \right. \right)$$

$$\mathbf{G}_{24} = \left(\mathbf{I} \left| \frac{\mathbf{B}}{11...11} \, \frac{\mathbf{1}}{0} \right. \right)$$

Rozměry matic:
I(12,12) .. jednotková matice
1(11,1) .. sloupcový vektor obsahující samé jedničky
B(11,11) .. matice, její řádky jsou tvořeny cyklickými posuvy prvního řádku (110111000010).
$\mathbf{G}_{23}$(12,23)
$\mathbf{G}_{24}$(12,24)

Minimální vzdálenost kódu $d_{min} = 8$, opravuje tedy všechny trojnásobné chyby.
Dekódování je komplikované.

Cyklické kódy

Jsou to zvláštní lineární kódy, vhodné pro detekci a opravu chyb, vyskytujících se ve shlucích. Používají se například k zabezpečení dat na disketách.

Definice cyklického kódu

Lineární kód je cyklický, jestliže současně s každým kódovým slovem

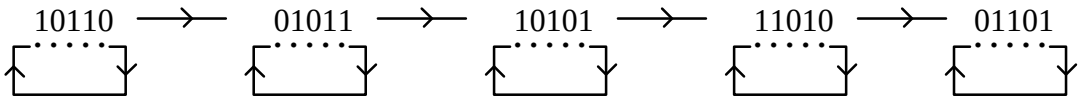
$(v_{n-1}\, v_{n-2}\, \ldots\, v_1\, v_0)$

obsahuje i kódové slovo vzniklé cyklickým posuvem:

$(v_0\, v_{n-1}\, v_{n-2}\, \ldots\, v_1)$

s Příklad 70:

Cyklický kód obsahuje toto kódové slovo:
(10110).
Pak musí obsahovat i tato další kódová slova:
(01011), (10101), (11010), (01101).



Poznámka: to ale neznamená, že daný kód již nebude obsahovat další kódová slova.

þ

Tento princip zaručuje, že vliv chyb na začátku a na konci značky (ojedinělé chyby) se projevuje stejně jako shluk chyb vzniklých spojením těchto chyb po cyklických posuvech.

Jako u každého lineárního kódu, *n*-místná značka cyklického kódu obsahuje *k* datových a $r = n-k$ zabezpečovacích bitů. Cyklický kód obecně nemusí být systematický.
Cyklické kódování je způsob, jak nalézt *r* zabezpečujících bitů k *k* informačním bitům tak, abychom vygenerovali kódové slovo cyklického kódu.
Dekódování je způsob zjištění k informačních bitů z přijatého slova včetně informace, zda jsou v těchto bitech chyby (detekce), případně kde jsou chyby (korekce).

Matematickým prostředkem pro usnadnění kódovacích a dekódovacích procedur je teorie polynomů (mnohočlenů).
n-bitové binární slovo lze zapsat buď jako vektor

$(v_{n-1}\ v_{n-2}\ \ldots\ v_1\ v_0)$,

nebo jako binární polynom

$$v_{n-1}x^{n-1} + v_{n-2}x^{n-2} + \ldots + v_1 x + v_0 .$$

s Příklad 71:

$$(110101) \rightarrow x^5 + x^4 + x^2 + 1.$$

þ

Cyklický kód má svůj **generující (vytvářecí) polynom** $g(x)$, což je obdoba generující matice **G**. Generující polynom musí mít tyto vlastnosti:

- Řád $g(x)$ je $n-k = r$.
- Polynom $x^n + 1$ je beze zbytku dělitelný polynomem $g(x)$.

Z těchto vlastností vyplývá, že ne každý polynom může být generujícím polynomem cyklického kódu.

Pravidla pro násobení a dělení binárních polynomů

Jsou založena na algebře modulo 2“:

1 + 0 = 0 + 1 = 1, 0 + 0 = 0, 1 + 1 = 0.

V polynomiálním zápisu vypadá poslední rovnost takto:

$x + x = x(1+1) = 0$.

s Příklad 72: Násobení polynomů.

$$(x^3 + x^2 + 1).(x+1) = x^4 + x^3 + x + x^3 + x^2 + 1 = x^4 + x^3(1+1) + x^2 + x + 1 = x^4 + x^2 + x + 1 \rightarrow (1\,0\,1\,1\,1).$$

þ

s Příklad 73: Dělení polynomů (s nulovým zbytkem).

$(x^4+x^2+x+1):(x^3+x^2+1)=x+1$

$\underline{x^4+x^3+x}$

x^3+x^2+1

$\underline{x^3+x^2+1}$

0

$(1\ 0\ 1\ 1\ 1):(1\ 1\ 0\ 1) = (1\ 1)$

$\underline{1\ 1\ 0\ 1}$

$1\ 1\ 0\ 1$

$\underline{1\ 1\ 0\ 1}$

0

Postupuje se stejně jako u dekadických polynomů. Využívá se pouze toho, že není třeba rozlišovat mezi sčítáním a odečítáním a mezi znaménky + a -.
Na pravé straně je proces dělení vyjádřený pomocí binárních vektorů.

þ

s Příklad 74: Dělení polynomů (s nenulovým zbytkem).

$(x^4+x^2+x+1):(x^3+x+1)=x+\dfrac{1}{x^3+x+1}$

$\underline{x^4+x^2+x}$

1 ← zbytek

$(1\ 0\ 1\ 1\ 1):(1\ 0\ 1\ 1) = (1) + \dfrac{(1)}{(1\,0\,1\,1)}$

$\underline{1\ 0\ 1\ 1}$ → 1

Postupné dělení je ukončeno, je-li řád polynomu mezivýsledku nižší než řád polynomu dělitele. V případě zápisu pomocí vektorů, počet bitů zbytku (první bit vlevo musí být nenulový) je menší než počet bitů dělitele.
Zajímá-li nás pouze zbytek dělení a ne výsledek dělení, což bude aktuální v následujících procedurách kódování a dekódování, není třeba vůbec psát pravou stranu (viz dále).

þ

s Příklad 75: Příklady generujících polynomů.

Vynásobíme-li tři níže uvedené polynomy, dostaneme:

$(x^3+x^2+1).(x+1).(x^3+x+1)=(x^4+x^3+x+x^3+x^2+1).(x^3+x+1)=(x^4+x^2+x+1)(x^3+x+1)=$

$=x^7+x^5+x^4+x^3+x^5+x^3+x^2+x+x^4+x^2+x+1=x^7+1.$

Můžeme tedy říci, že polynom x^7+1 je beze zbytku dělitelný těmito polynomy:

$g_1(x)=x+1,$

$g_2(x)=x^3+x^2+1,$

$g_3(x)=x^3+x+1,$

$g_4(x)=(x^3+x^2+1).(x+1)=x^4+x^2+x+1,$

$g_5(x)=(x^3+x+1).(x+1)=x^4+x^3+x^2+1,$

$g_6(x)=(x^3+x^2+1)(x^3+x+1)=x^6+x^5+x^4+x^3+x^2+x+1.$

Podle definice základních vlastností generujících polynomů na str. 67 tedy pro $n = 7$ existuje celkem 6 cyklických kódů o 6 generujících polynomech. Získané výsledky jsou uspořádány do tabulky.

$n = 7$

č. kódu	*k*	*r*	*generující polynom*	
1	6	1	$x+1$	(1 1)
2	4	3	x^3+x^2+1	(1 1 0 1)
3			x^3+x+1	(1 0 1 1)
4	3	4	x^4+x^2+x+1	(1 0 1 1 1)
5			$x^4+x^3+x^2+1$	(1 1 1 0 1)
6	1	6	$x^6+x^5+x^4+x^3+x^2+x+1$	(1 1 1 1 1 1 1)

þ

Kódovací procedura.
Je dáno *k*-bitové datové slovo. Hledáme jeho rozšíření o *r* zabezpečovacích bitů cyklického kódu.

1) Datové slovo rozšíříme na *n*-bitovou značku přidáním $r=n-k$ nul. Číslo *r* není možné volit libovolně. Vzniklému slovu odpovídá polynom $f(x)$.
2) Vybereme generující polynom $g(x)$ řádu $r=n-k$.
3) Provedeme dělení $f(x):g(x)$ a vypočteme zbytek dělení. Je to polynom řádu $r-1$, resp. vektor o $r=n-k$ bitech.
4) Namísto původních $r=n-k$ nul na konci *n*-místné značky zapíšeme zbytek dělení. Získáme kódové slovo.

Poznámka: Tímto postupem je zaručeno, že kódové slovo cyklického kódu je beze zbytku dělitelné jeho generujícím polynomem. To je základní vlastnost kódového slova. Využívá se ho k testování, zda přijaté slovo patří do kódu, neboli zda je či není v přijatém slově chyba. Pro bezchybná data musí platit, že zbytek dělení je nulový. Tento zbytek se často označuje zkratkou CRC (Cyclic Redundancy Check).

s Příklad 76: Ukázka kódování pro $n = 7$, $k = 3$, $r = 4$.

Data (101) je třeba zakódovat cyklickým kódem tak, aby tvořil sedmimístnou značku. Postupujeme podle čtyř bodů obecné kódovací procedury:

1) $(1010000) \rightarrow f(x) = x^6 + x^4$.
2) Z tabulky na str. 68 vybereme generující polynom $g(x) = x^4 + x^2 + x + 1 \rightarrow (10111)$.
3) $(x^6 + x^4) : (x^4 + x^2 + x + 1) \rightarrow$ (1010000):(10111) = …

```
10111
 1100   ← zbytek
```

4) Kódové slovo: (1011100).

þ

s Příklad 77: Kontrola, zda slovo patří do cyklického kódu.

Po dělení kódového slova generujícím polynomem musí vyjít nulový zbytek. Vyzkoušíme pro kódové slovo, získané v příkladu 76.

(1011100):(10111) = …

```
10111
    0
```

Simulujme chybu v 2. bitu zleva:

(1111100):(10111) = …

```
10111
 100000
 10111
    1110   zbytek je nenulový, je detekována chyba v přijaté značce.
```

þ

Náznak vysvětlení, proč v kódovací proceduře doplňujeme zbytek dělení za datové bity:
zajišťujeme tak dělitelnost generujícím polynomem.

```
11:3 = 3                ale    (11-2):3 = 9:3 = 3
 9                                  9
 2  zbytek                          0  zbytek je nulový.
```

Shrnutí: Jestliže je zbytek odečten (což je v algebře „modulo 2“ totéž co přičten) od slova (…..000), zbytek po dělení bude nulový.

Souvislosti cyklických kódů s dalšími lineárními kódy.

s Příklad 78: Kód celkové kontroly parity (4,3) je vlastně cyklický kód.

V tabulce jsou uvedena kódová slova kódu sudé parity včetně jejich polynomiálních zápisů. Polynomy jsou rozloženy do součinů již dále nerozložitelných součinitelů.

kódové slovo	*polynomiální vyjádření*
0000	0
0011	$x+1$
0101	$x^2+1=(x+1)(x+1)$
0110	$x^2+x=x(x+1)$
1001	$x^3+1=(x^2+x+1)(x+1)$
1010	$x^3+x=x(x+1)(x+1)$
1100	$x^3+x^2=x^2(x+1)$
1111	$x^3+x^2+x+1=(x+1)(x+1)(x+1)$

Všechny kódové polynomy obsahují společný polynom x+1 řádu 1. Je to tedy generující polynom cyklického kódu (4,3):

$$g(x)=x+1\rightarrow(11).$$

Zkusme například zakódovat data (100). Doplníme je jednou nulou a dělíme vektorem (11):
(1000):(11) = ..

```
 11
  100
  11
   10
   11
    1
```

„jedničkový“ zbytek zapíšeme namísto původní nuly na konci. Kódové slovo bude (1001).

þ

Zobecnění – věta o bázi cyklického kódu.

Každý netriviální cyklický (n,k) kód obsahuje polynom $g(x)$ řádu $r = n-k$. Má tyto vlastnosti:

1) $g(x)$ dělí polynom $x^n + 1$ beze zbytku.
2) Polynomy $g(x)$, $xg(x)$, …, $x^{k-1}g(x)$ tvoří bázi kódu.

Polynom $g(x)$ je současně generujícím polynomem kódu.

s Příklad 79: Získání báze cyklického kódu z příkladu 78.

$g(x)=x+1$
$xg(x)=x^2+x$
$x^2g(x)=x^3+x^2$

(0011)
(0110)
(1100)

báze: lineární kombinací těchto slov vygenerujeme všechna kódová slova.

Generující matice daného kódu je tedy $\mathbf{G}=\begin{pmatrix}0&0&1&1\\0&1&1&0\\1&1&0&0\end{pmatrix}$.

þ

Zobecnění – vztah mezi generujícím polynomem a generující maticí cyklického kódu.

Generující matice cyklického kódu (n,k) vznikne cyklickými posuvy „doleva“ koeficientů $g_{n-k}\, g_{n-k-1} \ldots . g_1\, g_0$ generujícího polynomu až do vyčerpání nul vlevo:

$$\mathbf{G} = \begin{pmatrix} \overbrace{0\,0\,..\;\;0\,0\,0}^{k-1} & \overbrace{g_{n-k}\, g_{n-k-1} \ldots . g_1\, g_0}^{n-k+1} \\ 0\,0\,..\,0 & g_{n-k}\, g_{n-k-1} \ldots . g_1\, g_0\; 0 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots & \\ \underbrace{g_{n-k}\, g_{n-k-1} \ldots . g_1\, g_0}_{n-k+1} & \underbrace{0\,0\ldots..\;\;0\,0\,0}_{k-1} \end{pmatrix} \Bigg\} k \qquad (98)$$

n

Kontrolní polynom $h(x)$ cyklického kódu.

Je obdobou kontrolní matice, pomocí níž se testuje, patří-li slovo do kódu. Zde platí analogie: součin polynomu kódového slova $w(x)$ a kontrolního polynomu $h(x)$ musí být roven polynomu x^n+1:

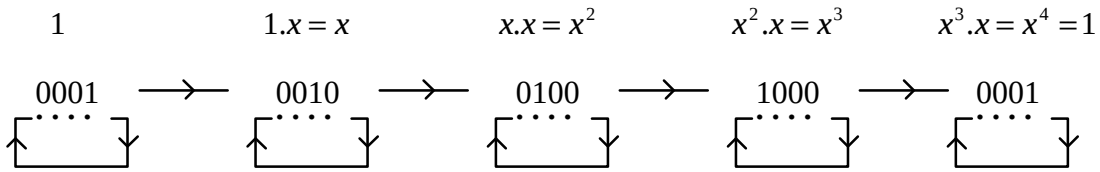
$$w(x)h(x) = x^n + 1\,. \qquad (99)$$

Kupodivu stejný vztah platí i mezi generujícím a kontrolním polynomem:

$$g(x)h(x) = x^n + 1, \qquad (100)$$

takže známe-li generující polynom, můžeme polynomiálním dělením získat polynom kontrolní. Kontrolní polynom je řádu k.

Co vlastně znamená číselně polynom $x^n + 1$? Vysvětlíme pro jednoduchost pro $n = 4$.
Vyjdeme ze slova (0001). Každé násobení symbolem x znamená posuv jedničky o jedno místo doleva. V důsledku cykličnosti platí $x^4 = 1$ – viz obrázek.



Výraz $x^n + 1$ tedy můžeme v důsledku cykličnosti posuvů považovat za nulový:

$$x^n + 1 \equiv 1 + 1 = 0\,. \qquad (101)$$

Z předchozího dále plyne, že

$$x^{n+m} \equiv x^m \text{ pro celé } m. \qquad (102)$$

Vzorce (99) a (100) mají tedy podobu

$$w(x)h(x) = g(x)h(x) \equiv 0\,. \qquad (103)$$

s Příklad 80: Kontrolní polynom cyklického paritního kódu z příkladu 78.

$n = 4,\ k = 3,\ g(x) = x + 1 \Rightarrow$

$$h(x) = (x^4+1) : (x+1) = x^3 + x^2 + x + 1 \rightarrow (1111).$$

$$\begin{array}{r} x^4 + x^3 \\ \hline x^3 + 1 \\ x^3 + x^2 \\ \hline x^2 + 1 \\ x^2 + x \\ \hline x + 1 \\ x + 1 \\ \hline 0 \end{array}$$

Zkontrolujeme pomocí (99), zda značka (1010) patří do kódu. K zjednodušení výrazu využijeme vzorce (101) a (102):

(1010).(1111)$\rightarrow$ $(x^3 + x)(x^3 + x^2 + x + 1) = x^6 + x^5 + x^4 + x^3 + x^4 + x^3 + x^2 + x = x^6 + x^5 + x^2 + x \equiv$

$\equiv x^2 + x + x^2 + x = 0$.

Þ

s Příklad 81: Kontrolní polynomy cyklických kódů z příkladu 75, definovaných tabulkou na str. 68.

Ke generujícím polynomům vypočteme kontrolní polynomy podle (100). Výsledky jsou uspořádány do tabulky:

$n = 7$

č. kódu	*k*	*r*	*generující polynom*	*kontrolní polynom*
1	6	1	$x+1$	$x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$
2	4	3	$x^3 + x^2 + 1$	$x^4 + x^3 + x^2 + 1$
3			$x^3 + x + 1$	$x^4 + x^2 + x + 1$
4	3	4	$x^4 + x^2 + x + 1$	$x^3 + x + 1$
5			$x^4 + x^3 + x^2 + 1$	$x^3 + x^2 + 1$
6	1	6	$x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$	$x+1$

Þ

Z rovnice (100) a příkladu 81 vyplývá, že:

Každý cyklický kód rozkládá polynom $x^n + 1$ v součin $g(x)h(x)$. Naopak každý takovýto součin definuje cyklický kód. Jestliže existuje cyklický kód o generujícím polynomu $g(x)$ a kontrolním polynomu $h(x)$, pak současně existuje cyklický kód o generujícím polynomu $h(x)$ a kontrolním polynomu $g(x)$. Dané kódy se vůči sobě chovají jako duální.
V tabulce jsou vůči sobě duální kódy č. 1 a 6, 2 a 5, 3 a 4.
O duálních kódech víme, že mají zaměnitelné matice **G** a **H** (generující a kontrolní). U cyklického kódu již umíme sestavit generující matici z generujícího polynomu. Proto v následující části ještě ukážeme, jak je možné z kontrolního polynomu určit kontrolní matici.

Vztah mezi kontrolním polynomem a kontrolní maticí cyklického kódu.

Kontrolní matici cyklického kódu (n,k) získáme cyklickými posuvy „doleva“ koeficientů $h_k\ h_{k-1}\ \ldots\ .\ h_1\ h_0$ kontrolního polynomu až do vyčerpání nul vlevo:

$$\mathbf{H} = \begin{pmatrix} \overbrace{0\,0\,..\;\;0\,0\,0}^{n-k-1} & \overbrace{h_k\,h_{k-1}\,\ldots\,.\,h_1\,h_0}^{k+1} \\ 0\,0\,..\,0 & h_k\,h_{k-1}\,\ldots\,.\,h_1\,h_0\;0 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots & \\ h_k\,h_{k-1}\,\ldots\,.\,h_1\,h_0 & 0\,0\ldots..\;\;0\,0\,0 \end{pmatrix} \begin{matrix} \\ \\ \\ \} \; r=n-k \end{matrix} \qquad (104)$$

$k+1$ $\quad$ $n-k-1$

n

s Příklad 82: Generující a kontrolní matice cyklických kódů, definovaných tabulkou na str. 72.

Matice sestavíme podle vzorců (98) a (104). Výsledky uspořádáme do tabulky. Postačí uvést matice pro první tři kódy. Další kódy jsou k nim duální.

$n = 7$

č. kódu	*k*	*r*	*generující matice*	*kontrolní matice*
1	6	1	$\begin{pmatrix} 0&0&0&0&0&1&1 \\ 0&0&0&0&1&1&0 \\ 0&0&0&1&1&0&0 \\ 0&0&1&1&0&0&0 \\ 0&1&1&0&0&0&0 \\ 1&1&0&0&0&0&0 \end{pmatrix}$	$\begin{pmatrix} 1&1&1&1&1&1&1 \end{pmatrix}$
2	4	3	$\begin{pmatrix} 0&0&0&1&1&0&1 \\ 0&0&1&1&0&1&0 \\ 0&1&1&0&1&0&0 \\ 1&1&0&1&0&0&0 \end{pmatrix}$	$\begin{pmatrix} 0&0&1&1&1&0&1 \\ 0&1&1&1&0&1&0 \\ 1&1&1&0&1&0&0 \end{pmatrix}$
3			$\begin{pmatrix} 0&0&0&1&0&1&1 \\ 0&0&1&0&1&1&0 \\ 0&1&0&1&1&0&0 \\ 1&0&1&1&0&0&0 \end{pmatrix}$	$\begin{pmatrix} 0&0&1&0&1&1&1 \\ 0&1&0&1&1&1&0 \\ 1&0&1&1&1&0&0 \end{pmatrix}$

þ

Z tabulky vyplývá, že například cyklický kód č. 1 je paritní kód (7,6). Kód č. 6 je k němu duální, jeho kontrolní matice je generující maticí kódu č. 1 a patří opakovacímu kódu (7,1) (viz též str. 49 a 56).

Literatura

[1] NĚMEC, K.: Datová komunikace. Skriptum VUT, VUTIUM 2000.
[2] ŽALUD,V.: Moderní radioelektronika. BEN, Praha 2000. ISBN 80-86056-47-3. 799 Kč.
[3] ŽALUD,V.: Multimediální přenosy signálů. Skriptum ČVUT Praha, 1995. ISBN 80-01-01338-3. 69 Kč.
[4] ADÁMEK,J.: Kódování a teorie informace. Skriptum ČVUT Praha, 1994. ISBN80-01-00661-1. 42 Kč.
[5] HAVLÍK,J.: Teorie přenosu. Učebnice VA Brno, U-1039, 1976.
[6] ŠEBESTA,V.: Přenos dat. Skriptum FEI VUT Brno, 1990, ISBN 80-14-0229-6.
[7] PRCHAL,J.: Signály a soustavy. SNTL/ALFA 1987.
[8] KROUTL, F.: Signály a hluky. NADAS, 1964.
[9] POLETAJEV, I.A.: Kybernetika. SNTL 1961.
[10] LATHI, B.P.: Modern Digital and Analog Communication Systems. HRW, USA, 1983. ISBN 0-03-058969-X.
[11] MORKES,D.: Komprimační a archivační programy. Computer Press, Brno 1998.
[12] DEMLOVÁ,M.-NAGY,J.: Algebra (Matematika pro vysoké školy technické III). Praha, SNTL 1982.