VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

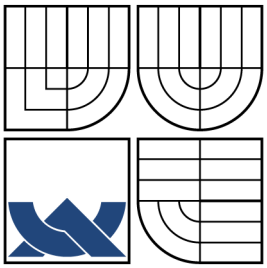
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF BIOMEDICAL ENGINEERING

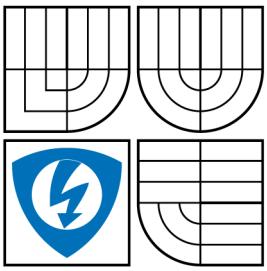
PŘENOS KOMPRIMOVANÉHO EKG SIGNÁLU PO SÍTI ETHERNET

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE Bc. ZUZANA BOHATCOVÁ
AUTHOR



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

FACULTY OF ELECTRICAL ENGINEERING AND
COMMUNICATION
DEPARTMENT OF BIOMEDICAL ENGINEERING

PŘENOS KOMPRIMOVANÉHO EKG SIGNÁLU PO SÍTI ETHERNET

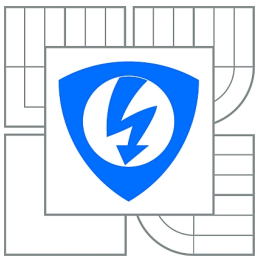
ECG SIGNAL TRANSMISSION VIA ETHERNET

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR
Bc. ZUZANA BOHATCOVÁ

VEDOUCÍ PRÁCE
SUPERVISOR
Ing. JIŘÍ SEKORA

BRNO 2012



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav biomedicínského inženýrství

Diplomová práce

magisterský navazující studijní obor
Biomedicínské inženýrství a bioinformatika

Studentka: Bc. Zuzana Bohatcová ***ID:*** 106162
Ročník: 2 ***Akademický rok:*** 2011/2012

NÁZEV TÉMATU:

Přenos komprimovaného EKG signálu po síti Ethernet

POKYNY PRO VYPRACOVÁNÍ:

1) Proveďte literární rešerši v oblasti metod komprese EKG signálů. Zaměřte se na metody určené k úpravě signálů za účelem přenosu komunikačními kanály. 2) Seznamte se s podstatou komunikace v síti Ethernet, zejména strukturou TCP a UDP paketů s ohledem na danou aplikaci. 3) Navrhněte metodu pro kompresi, odesílání, příjem a rekonstrukci EKG signálu. 4) Metodu realizujte za pomocí dostupných bloků v prostředí LabVIEW. 5) Vytvořenou aplikaci ověřte na datovém přenosu reálných EKG signálů. 6) Proveďte diskuzi získaných výsledků a možností praktického využití.

DOPORUČENÁ LITERATURA:

[1] LEE, S., LEE, M. A real-time ECG data compression algorithm for a digital holter system. Conf Proc IEEE Eng Med Biol Soc., 2008: 4736-9.
[2] National Instruments: Industrial Control and Distributed I/O. LabVIEW Real-Time Controller Interfaces with Ethernet. 2004. Dostupné z: <www.ni.com>.

Termín zadání: 6.2.2012 ***Termín odevzdání:*** 18.5.2012

Vedoucí práce: Ing. Jiří Sekora

prof. Ing. Ivo Provazník, Ph.D.
Předseda oborové rady

UPOZORNĚNÍ:

ABSTRAKT

V teoretické části diplomové práce jsou popsány metody komprese EKG záznamu určené k úpravě záznamu za účelem přenosu komunikačními kanály. Práce obsahuje úvod do sítě Ethernet a seznámení s podstatou komunikace v této síti. Podrobněji jsou zde rozebrány transportní protokoly TCP a UDP. V praktické části práce byly vytvořeny dvě samostatné aplikace. První aplikace v počítači odesílatele otevírá textový soubor s EKG signálem a načtený EKG signál pak filtruje kaskádou filtrů za účelem odstranění rušení. Výsledný signál je zobrazen. Součástí aplikace je detekce R vlny, výpočet délky RR intervalu a tepové frekvence. Aplikace dále umožňuje EKG signál komprimovat. EKG signál je po síti Ethernet odesílán prostřednictvím protokolu UDP po jednotlivých vzorcích. Aplikace v počítači příjemce přijímá vzorky signálu ze sítě, je-li přijatý signál komprimovaný, tak jej rekonstruuje. Výsledný EKG signál je zobrazen a jsou v něm opět detekovány R vlny, vypočtená délka RR intervalů a vzorkovací frekvence.

KLÍČOVÁ SLOVA

EKG, Ethernet, komprese dat, TCP, UDP

ABSTRACT

The semestral thesis describes ECG signal compression methods designed to modify the data for transmission via communication channels. The thesis contains an introduction to Ethernet and explanation of communication in the network. The transport protocols TCP and UDP are discussed in more detail. In the practical part of the thesis was created two separate applications. The first application in the sender's computer opens a text file with the ECG signal. Loaded ECG signal is filtered by cascade of filters to eliminate interference. The resulting signal is displayed. A part of the application is the R wave detection, calculating the length of RR interval and heart rate. The application also allows to compress an ECG signal. ECG signal is sent via Ethernet network via UDP protocol for individual samples. Applications in the recipient's computer receives signal samples from the network. Recieved compressed data is reconstructed. The resulting ECG signal is displayed and there are again detected R waves, the length of RR intervals and sampling frequency are calculated.

KEYWORDS

ECG, Ethernet, data compression, TCP, UDP

BOHATCOVÁ, Zuzana. *Přenos komprimovaného EKG signálu po síti Ethernet*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav biomedicínského inženýrství, 2012. 76 s. Vedoucí práce byl Ing. Jiří Sekora.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Přenos komprimovaného EKG signálu po síti Ethernet“ jsem vypracovala samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autorka uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušila autorská práva třetích osob, zejména jsem nezasáhla nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědoma následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

Brno ....

(podpis autora)

PODĚKOVÁNÍ

Na tomto místě bych ráda poděkovala panu Ing. Jiřímu Sekorovi, vedoucímu své diplomové práce, za jeho podporu, trpělivost a čas, který mi věnoval, cenné rady a především samotný námět diplomové práce.
Dále bych chtěla poděkoval svým rodičům, kteří mě vychovali a neustálně mě morálně i finančně podporovali při studiu. V neposlední řadě děkuji svému partnerovi Patrikovi za trpělivost, toleranci a lásku, kterou mi během nelehkého studia projevoval.

V Brně dne .....

.....
(podpis autora)

OBSAH

SEZNAM OBRÁZKŮ

ÚVOD

V současné době již pokročily informační technologie natolik, že lékaři mohou ze svého pracoviště v reálném čase prohlížet záznamy, které vznikly na jiných odděleních nebo pracovištích. Stejně tak dávají k dispozici svůj materiál jiným kolegům.

Pojem telemedicína je označení pro elektronickou výměnu informací mezi poskytovateli zdravotní péče a pacienty nebo mezi poskytovali zdravotní péče navzájem s cílem zdokonalení zdravotní péče. Často se jedná o výměnu obrazových, zvukových nebo textových informací na velké vzdálenosti. Telemedicína je užitečná v případech, kdy fyzické bariéry brání rychlému přenosu infomací zejména mezi poskytovateli zdravotní péče. Mezi její přínosy patří také lepší přístup k informacím pro vzdálené pacienty, snížení cestovních nákladů poskytovatelů zdravotní péče a lepší možnosti vzdělávání v medicíně.

Podle způsobu jakým jsou data přenášena, telemedicínu rozdělujeme do tří oblastí: zasílání informací, dálkové monitorování a dálková terapie. Zasílání informací je technicky nejsnáze realizovatelné. Data jsou nejprve zaznamenána a teprve poté jsou odeslána. V souvislosti s rozmachem Internetu doznala tato oblast bouřlivého rozvoje. Elektronické služby umožňují rychle a pružně přenášet obrazová data na vzdálená místa. Výhodou rychlého sdílení informací je snadná dostupnost konzultace s expertem, který může řešit nejasné nebo obtížné případy. Dálkové monitorování se vyznačuje tím, že měřená data jsou vyhodnocována v reálném čase. Příkladem je snaha spojit kardiostimulátor s online monitorováním srdeční akce. Technicky nejnáročnějším konceptem je dálková terapie. Postatou je to, že pacient a lékař jsou od sebe geograficky vzdálení a diagnostické a terapeutické výkony jsou zprostředkovány výhradně technickými prostředky.

První kapitola diplomové práce je věnována popisu EKG křivky, metodám snímání EKG signálu, zájmovým oblastem signálu z pohledu diagnostiky a objemu snímaných dat. Ve druhé kapitole jsou představeny obecné metody komprimace dat a následně metody komprese EKG signálu. Rešerže je zaměřena na aktuálně používané kompresní metody určené k úpravě signálů za účelem přenosu komunikačními kanály. Třetí kapitola obsahuje úvod do sítě Ethernet a seznámení s podstatou komunikace v této síti. Podrobněji jsou zde diskutovány transportní protokoly. Další kapitola je již věnována návrhu uživatelských aplikací pro odesílání dat po síti Ethernet a pro jejich příjem. V samostatné kapitole jsou vysvětleny jednotlivé kroky realizace návrhu v programu LabVIEW. Postupně jsou popsány dílčí části programu, jak z pohledu vytváření blokového diagramu, tak z pohledu uživatelského čelního panelu. V závěru jsou shrnuty dosažené výsledky a výstupy z testování programu v různých sítích.

1 EKG

1.1 Elektrický srdeční vektor

Rozhraní mezi depolarizovanou a klidově polarizovanou buňkou představuje dipól. Tento dipól můžeme považovat za vektor, který má velikost a směr. Je orientován od depolarizované tkáně k polarizované. Součtem všech těchto elementárních vektorů v celém srdci vzniká okamžitý integrální vektor srdeční. Tento vektor v každém okamžiku elektrické aktivity mění svou velikost a směr podle toho, jak se vzruch šíří srdcem. U zdravého srdce probíhá srdeční vektor cyklus od cyklu konstantně.

Jsou-li všechny buňky klidově polarizovány, nebo naopak všechny depolarizovány, je elektrický vektor nulový. Mezi těmito stavy vektor postupně opisuje čtyři smyčky: při depolarizaci síní, repolarizaci síní, depolarizaci komor a repolarizaci komor. [40]

1.2 Snímání EKG

Napětí tvořené elektrickým polem srdečním je tak silné, že je lze snímat nejen z povrchu srdce, ale i z povrchu těla, čehož využívá elektrokardiografie (EKG). Jedná se o základní vyšetřovací metodu v kardiologii založenou na snímání elektrické aktivity srdečního svalu a jejím záznamu ve formě elektrokardiogramu. K registraci elektrického srdečního pole slouží elektrody, které se přikládají na povrch těla. Rozdíl napěťových potenciálů mezi dvojicí elektrod se označuje jako svod. EKG se obvykle zaznamenává pomocí vícesvodové konfigurace, která zahrnuje unipolární nebo bipolární svody, popřípadě oboje. Unipolární svody zaznamenávají napětí mezi měřící a referenční elektrodou. Svody bipolární měří napětí mezi dvěma měřícími elektrodami.

V současnosti existuje řada svodových systémů se standardizovanými pozicemi elektrod. Standardní 12-svodové EKG je nejčastěji používaným svodovým systémem v praxi. Je tvořeno kombinací tří různých svodových konfigurací: bipolárními končetinovými svody, rozšířenými unipolárními končetinovými svody a unipolárními prekordiálními svody. Jednotlivé svody jsou zaznamenávány pomocí 10 elektrod umístěných na standardizovaných pozicích na povrchu těla. Tři bipolární končetinové svody I, II a III se získávají měřením napětí mezi levou paží, pravou paží a levou nohou. Pozice tří měřících elektrod lze považovat za vrcholy rovnostranného trojúhelníku se srdcem uprostřed. Rozšířené unipolární končetinové svody aVF, aVL a aVR využívají stejné elektrody jako bipolární končetinové svody, ale jsou definovány napětím mezi jedním z vrcholů trojúhelníku a průměrem zbývajících

dvou vrcholů. Prekordiální neboli hrudní svody jsou umístěny v řadě na přední levé straně hrudníku. Poskytují detailnější pohled na srdce než svody končetinové. Prekordiálních svodů je šest a označují se V1 až V6. Svody jsou unipolární a jsou vztažené k centrální svorce, která je definována jako průměr napětí měřených na pravé paži, levé paži a levé noze.

1.3 Elektrokardiogram

Grafický záznam časového průběhu elektrické činnosti srdce, zaznamenaný elektrodami ve strategických místech tělesného povrchu, se nazývá elektrokardiogram. EKG křivka znázorňuje elektrické proudy, jejich směr a velikosti, stejně jako frekvenci srdečních stahů.

Na EKG křivce (Obr. 1.1) rozeznáváme následující útvary:

- **Vlny** – mají oblý tvar a nižší amplitudu (vlna P, T, U)
- **Kmity** – jsou většinou hrotnaté a mají větší amplitudu (kmity Q, R, S)
- **Segmenty** – zahrnují úsek mezi koncem jedné vlny nebo kmitu do začátku následující vlny nebo kmitu (segment PQ, ST)
- **Intervaly** – zahrnují segment a přiléhající vlnu nebo kmit (interval PQ, QT, ST)

Část elektrokardiogramu, kde není přítomna žádná vlna nebo kmit, se označuje jako izoelektrická linie. [21]

1.4 EKG křivka

Vlny a kmity

- **P vlna** – pozitivní vlna způsobená depolarizací síní, má amplitudu do $2,5\,\mathrm{mV}$ a netrvá déle než $0,10\,\mathrm{s}$
- **QRS komplex** – podkladem QRS komplexu je depolarizace komor; depolarizace probíhá postupně, nejprve je depolarizováno septum z větve levého Tawarova raménka, následně jsou aktivovány subendokardiální vrstvy obou komor, odkud se podráždění šíří napříč pracovním myokardem až k epikardu; QRS komplex se sestává ze 3 kmitů – Q, R, S a délka trvání komplexu je $0,06 - 0,10\,\mathrm{s}$
- **Q kmit** – první negativní kmit komplexu QRS (pod izoelektrickou linií), který vždy předchází kmit R, vyjadřuje depolarizaci septa a papilárních svalů, je

široký do 0, 03 s, výškou nepřesahuje 1/4 výchylky R v tomtéž svodu

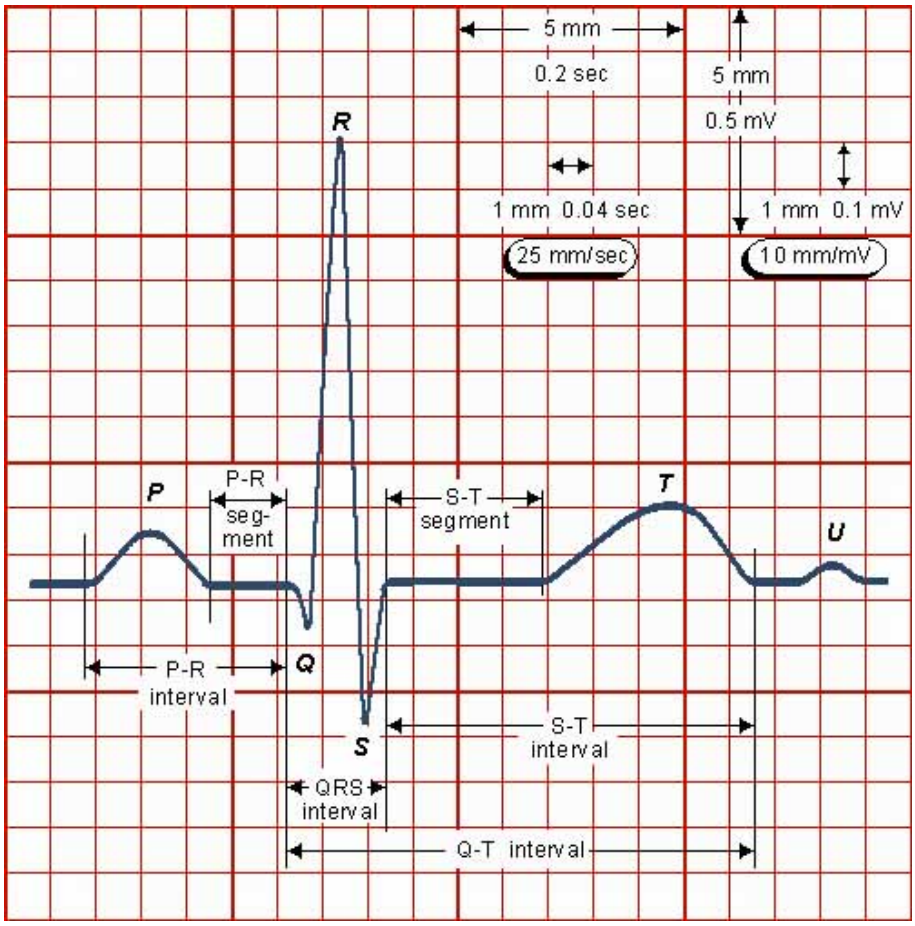
- **R kmit** – první pozitivní kmit komplexu, amplituda výchylky je závislá na místě snímání EKG signálu, délka trvání je do 0, 2 ms
- **S kmit** – druhý negativní kmit, který následuje po kmitu R, nevyskytuje se vždy, amplituda je v rozsahu $0 - 0,8$ mV, délka trvání do 0, 05 ms
- **T vlna** – nachází se za QRS komplexem a vzniká při repolarizaci komor, vlna je pozitivní, amplituda běžně dosahuje 0,2 až 0, 8 mV, délka trvání $0,1 - 0,25$ s
- **U vlna** – nachází se za vlnou T, ale je patrná jen na některých EKG záznamech, například u mladých lidí a sportovců; příčina vlny U není jasná, přisuzuje se pozdější repolarizaci septa nebo opožděné repolarizaci některých oblastí komory, může být také výsledkem repolarizace Purkyňových vláken

Segmenty a intervaly

- **RR (PP) interval** – vzdálenost mezi dvěma kmity R (P) po sobě jdoucích komorových (síňových) komplexů (vln), při pravidelném rytmu je tato vzdálenost relativně neměnná
- **PQ(R) interval** – interval, který měříme od začátku vlny P k začátku komorového komplexu (kmitu Q nebo R), představuje dobu, za kterou přejde vzruch ze SA uzlu přes AV uzel k odstupu komorových ramének z Hisova svazku, informuje nás tedy o trvání síňokomorového vedení, interval PQ je normálně izoelekrický a trvá $0,12 - 0,20$ s
- **QRS interval** – měříme od začátku kmitu Q(R) na konec kmitu S, je to doba trvání komorové depolarizace
- **QT interval** – představuje elektrickou systolu, interval měříme od začátku Q(R) na konec vlny T, hodnoty QT výrazně ovlivňuje tepová frekvence
- **ST segment** – zaujímá izoelektrickou linii mezi koncem QRS komplexu a začátkem vlny T v době repolarizace komor, pokles nebo vzrůst do 0, 1 mV vůči izoelektrické linii je fyziologický
  Pozn. Bod, kterým končí QRS komplex a začíná úsek ST, se označuje jako **junkční bod (bod J)**.
- **QTU interval** – měříme v případě, kdy se vlna U promítá do konce vlny T a interval QT nelze přesně stanovit

1.5 Objem snímaných dat

Standardní EKG signál je snímán třemi bipolárními končetinovými svody. Kromě těchto svodů se používají i tři zesílené unipolární končetinové svody. Rozdíl je v za-



Obr. 1.1: EKG křivka [43]

pojení elektrod, pro účely komprese a stanovení objemu snímaných dat jsou jen další tři kanály. EKG je často snímáno také pomocí šesti hrudních svodů. Kromě uvedených existují i další svodové systémy, ty však nejsou tak rozšířené (např. ortogonální svodový systém).

Každý svod generuje data, která jsou dále digitalizována. Vzorkovací frekvence takových dat se podle povahy vyšetření pohybuje obvykle od 125 Hz do 500 Hz. Pokud snímaný EKG signál obsahuje impulsy kardiostimulátoru, je nutné použít výrazně vyšší vzorkovací frekvenci. V takových případech se vzorkuje s frekvencí až 2000 Hz. Amplituda signálu se kvantizuje s použitím osmi až dvanácti bitů pro zápis každého vzorku. V extrémním případě, kdy snímáme data ze všech dvanácti svodů při vzorkovací frekvenci 2000 Hz a hloubce 12 bitů, dostáváme datový tok 288 kbps. To znamená, že každou minutu se na pevný disk bez jakékoliv komprese zapíše 2, 16 MB. Za jednu hodinu je to potom 129, 6 MB. Taková velikost dat představuje problém zvláště při přenosu těchto dat po síti.

Záznamy EKG nejsou snímány jen po dobu několika hodin. V případě ambu-

lantního monitorování EKG, kdy se používají přenosné osobní přístroje Holterova typu, je signál snímán např. 24 hodin. Holterovské monitory jsou schopné snímat signál i s impulsy kardiostimulátoru, proto používají značně nadsazenou vzorkovací frekvenci 2000 Hz. Při ambulantním monitorování EKG se nevyužívá všech dvanácti svodů. Holterovskými funkcemi jsou dnes vybaveny i některé kardiostimulátory.

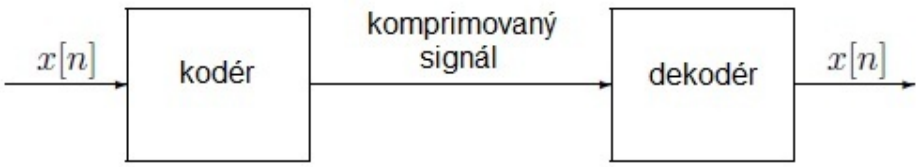
2 KOMPRESE

2.1 Komprese dat

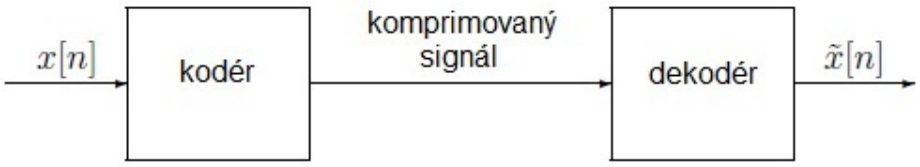
Komprimace neboli komprese dat je postup, jehož cílem je zmenšit výslednou velikost dat. Využívá se při ukládání dat na fyzická média, ale také při přenosu dat přes počítačové sítě, kde výrazně zkracuje dobu přenosu.

Základním principem komprese dat je odstranění nadbytečné informace ze souboru dat. Jedná se buď o data, která se v souboru opakují nebo o data, která je možné odvodit z dat zachovaných.

V praxi se komprese skládá ze dvou zpravidla samostatných algoritmů. Jeden slouží ke kompresi a obvykle se nazývá kodér, druhý k dekompresi a v tomto případě mluvíme o dekodéru. Oba dva algoritmy mohou mít rozdílné požadavky na výpočetní výkon, z čehož vyplývá, že i doba komprese a dekomprese se může lišit.



Obr. 2.1: Blokový diagram algoritmu bezeztrátové komprese



Obr. 2.2: Blokový diagram algoritmu ztrátové komprese

Dělení na základě počtu průchodů souborem dat:

- **Jednoprůchodové** – soubor dat projde algoritmem pouze jednou a ihned je komprimován
- **Dvou a víceprůchodové** – soubor dat projde algoritmem dvakrát nebo vícekrát, přičemž komprimován je až při druhém průchodu, první slouží pouze ke zjištění parametrů vstupních dat

Dělení podle symetrie komprese a dekomprese:

- **Symetrické** – procesy komprese i dekomprese jsou víceméně stejně výpočetně náročné, tzn. pro jeden soubor dat se délka trvání komprese a dekomprese výrazně neliší
- **Asymetrické** – komprese, nebo dekomprese je výrazně náročnější, tzn. trvá mnohem déle, náročnější obvykle bývá komprese

Dělení je podle kvality dekomprimovaných dat:

- **Ztrátové** – část informace se během komprese ztratí nebo zkreslí
- **Bezeztrátové** – výsledná informace je identická s informací původní, nedochází ke ztrátě ani zkreslení

2.1.1 Bezeztrátová komprese

Bezeztrátová komprese je jeden ze dvou základních přístupů ke kompresi dat. Jedná se o algoritmy, které umožňují přesnou zpětnou rekonstrukci komprimovaných dat, na rozdíl od ztrátové komprese, kde to možné není. Dekomprimovaná data jsou totožná s daty vstupními. Nevýhodou bezeztrátové komprese je nízký kompresní poměr (CR).

$$CR = \frac{V_{orig}}{V_{komprim}} \tag{2.1}$$

Bezeztrátová komprese se používá všude, kde je nezbytné zachovat úplnou informaci. Příkladem použití bezeztrátové komprese je komprese textů, kde je nepřípustná i sebemenší ztráta kvality.

Algoritmy bezeztrátové komprese se dělí podle typů dat, pro které jsou určeny. Algoritmy zpracovávají čtyři základní typy dat – textová data, obrazová data, zvuková data a videa. Existují také univerzální algoritmy, které mohou zpracovávat jakýkoliv typ vstupních souborů. V praxi se ale nepoužívají, neboť obvykle nejsou schopny docílit takového kompresního poměru jako specializované algoritmy. Většina bezeztrátových komprimačních programů nepoužívá jen jeden algoritmus, ale hned několik najednou. U některých komprimačních programů jsou data napřed transformována a až poté komprimována. Transformace se používá za účelem dosažení lepších kompresních poměrů.

Rozdělení algoritmů bezeztrátové komprese:

- **Statistické algoritmy** – pracují na základě četnosti znaků, pro znaky s vyšší pravděpodobností výskytu vyhradí algoritmus menší počet bitů k jejich zapsání, pro znaky s nižší pravděpodobností výskytu vyhradí naopak větší počet bitů k jejich zapsání, např. Shannon-Fanovo kódování, Huffmanovo kódování, Aritmetické kódování (známé taktéž jako aritmeticko logické kódování), Range coding (RC), ACB, Prediction by partial match (PPM)

  Statistické metody se dále dělí na metody se statickým modelem a metody s adaptivním modelem. Model slouží pro vypočítávání pravděpodobnosti výskytu znaků.

  - **Metody se statickým modelem** - vytvoří před komprimací dat určitý model a podle něho zkomprimují celý soubor dat, zpravidla dvouprůchodové.
  - **Metody s adaptivním modelem** - průběžně model aktualizují, zpravidla jednoprůchodové.

- **Slovníkové algoritmy** – vytvářejí v průběhu komprese slovník (na základě dat již zkomprimovaných), v něm pak hledají data, která se teprve mají komprimovat, opakující se symboly následně nahrazují odkazem na jejich předchozí pozici; do této kategorie patří Lempel-Ziv kodéry, např. Lempel-Ziv 77 (LZ77), Lempel-Ziv 78 (LZ78), Lempel-Ziv-Welch 84 (LZW84), LZMA

- **Transformace** – algoritmy, které data přímo nekomprimují, ale slouží k jejich modifikaci, aby se dala lépe zkomprimovat; ke každé transformaci musí existovat transformace inverzní, která je schopna obnovit původní data, např. Burrows-Wheelerova transformace (BWT), Move-to-front transformace (MTF I a MTF II), Weighted frequency count (WFC), Distance coding (DC), Inverse frequency coding (IF)

- **Ostatní algoritmy** – Run length encoding (RLE), Potlačení nul, Bitové mapy, Půlbajtové kódování

2.1.2 Ztrátová komprese

Ztrátová komprese je algoritmus, který zmenšuje objem dat na zlomek původní velikosti. Přitom se některé méně důležité informace ztrácejí a z vytvořených dat je již není možné zrekonstruovat. Ztrátová komprese se nejčastěji používá pro ukládání obrazových a zvukových záznamů. Přestože se část informace při ztrátové kompresi nevratně ztrácí, je tento způsob ukládání dat často velmi výhodný. Ztrátu některých informací kompenzuje mnohem vyšší kompresní poměr, tedy výrazné zmenšení komprimovaných dat a výrazná úspora místa. Díky vysokému kompresnímu poměru je ztrátová komprese mnohem rozšířenější než bezeztrátová. Ztrátová komprese se využívá při digitálním televizním vysílání, u filmových DVD a při přenášení dat přes Internet a jiné datové sítě. Další využití je při ukládání do multimediálních přehrávačů. Ztrátová komprese je nepoužitelná v případě, kdy je potřeba uchovat přesnou kopii původních dat, například text knihy, program nebo výsledky měření.

Obecný postup ztrátové komprese je jednoduchý. Vstupní data se po úvodním předzpracování nejprve transformují a teprve následně se uplatňují komprimační algoritmy. Při dekomprimaci je postup opačný. Nejdříve se na data aplikují dekomprimační algoritmy a teprve potom inverzní transformace.

Transformace slouží k oddělení důležitých informací od nedůležitých. K transformaci původních nebo předzpracovaných dat se obvykle používá některá z ortonormálních nebo téměř ortonormálních transformací.

- **Diskrétní kosinová transformace (DCT)**
- **Rychlá Fourierova transformace (FFT)**
- **Diskrétní vlnková transformace (DWT)**

Tyto transformace převedou původní data do jiných domén, například z časové do frekvenční. Většina z důležitých informací je poté uchována v mnohem menším objemu než původně. Pokud zbytek dat nahradíme nějakými předem známými nebo vypočitatelnými daty (někdy se pro tento účel hodí samé nuly), data se po zpětné transformaci budou podobat datům původním. Při transformaci nemusí docházet k degradaci původních dat. Celočíselné verze transformací pracují obvykle o něco hůře, ale jsou snadněji implementovatelné, a proto se používají v jednoduchých zařízeních. Ztráta způsobená zaokrouhlováním reálných čísel nebývá velká a obvykle ji vynahradí kvalitnější výsledek transformace pracující s reálnými čísly.

Ve fázi potlačení méně důležitých informací (kompresní část algoritmu) je rozhodující kvalitní psychovizuální nebo psychoakustický model. Ten určuje, která data mohou být potlačena nebo úplně odstraněna.

2.2 Komprese EKG signálů

Jak již bylo řečeno v úvodu kapitoly, obecně platí, že lze kompresní metody rozdělit na ztrátové a bezeztrátové. Pro kompresi EKG signálů se používají ztrátové kompresní algoritmy, a to kvůli jejich schopnosti výrazně snížit objem dat. EKG kompresní metody lze podle [27] rozdělit do tří skupin. První z nich jsou metody přímé komprese dat. Vstupní data jsou analyzována a komprimována v časové oblasti. Druhou skupinou jsou transformační metody, které převádí časovou oblast signálu na frekvenční, nebo do jiné domény, a analyzuje se distribuce energie. Třetí skupina zahrnuje extrakční kompresní metody, které extrahují prvky a parametry signálů.

Metody přímé komprese dat:

- Turning point (TP)
- Amplitude zone time epoch coding (AZTEC)
- Coordinate reduction time
- Encoding system (CORTES)
- The delta algorithm
- The fan algorithm

Transformační metody:

- Fourierova transformace - Fourier transform
- Fourier descriptor
- Karhunen–Loeve transform (KLT)
- The Walsh transform
- Diskrétní kosinová transformace - The discrete cosine transform (DCT) [7]
- Vlnková transformace - The wavelet transform (WT) [28]

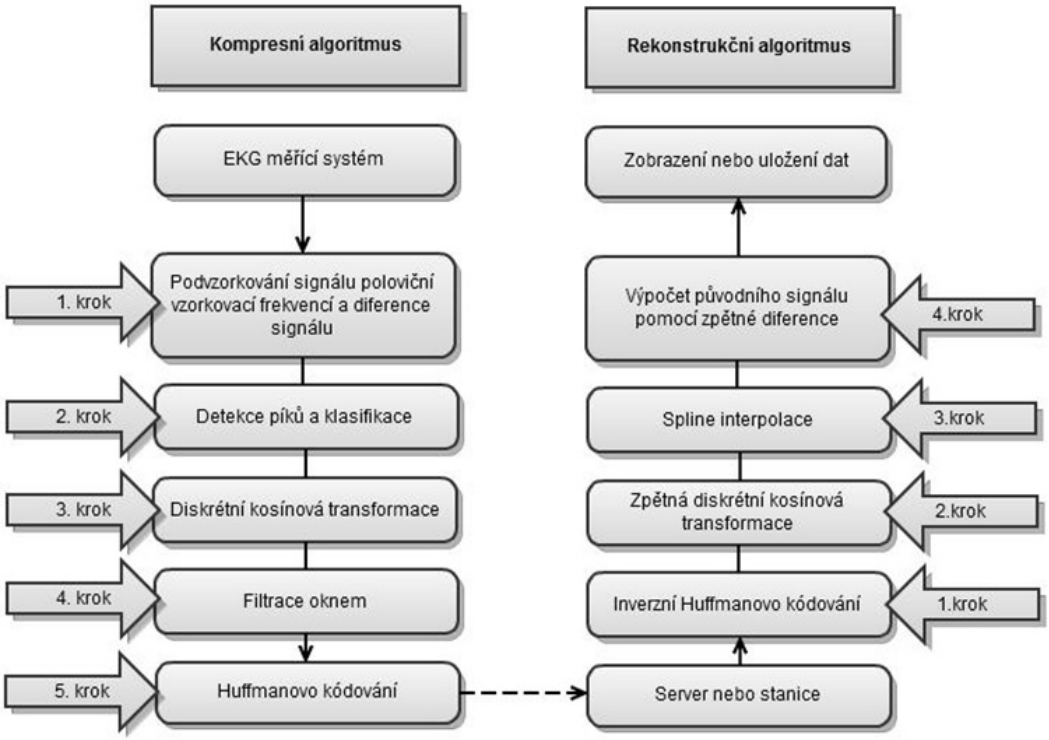
Extrakční kompresní metody

- Peak picking method
- The linear prediction method
- Syntactic method
- The neural network method

Algoritmus pro kompresi EKG signálu navrhovaný v článku [27] využívá metodu DCT. Nejprve jsou detekovány vrcholy EKG signálu, poté je provedena DCT mezi stávajícími a předcházejícími vrcholy datového toku pro kompresi v reálném čase.

2.2.1 Algoritmus pro kompresi EKG signálu v reálném čase

Algoritmus pro kompresi v reálném čase a rekonstrukci signálu EKG je naznačen na Obr. 2.3. Skládá se z pěti kompresních kroků a čtyř rekonstrukčních kroků. V prvním kroku komprese se získá diference po podvzorkování EKG signálu poloviční vzorkovací frekvencí. V druhém kroku probíhá detekce píků, klasifikace a uložení výsledných dat. Ve třetím kroku jsou uložená data transformována použitím diskrétní kosinové transformace (DCT). Výstupem jsou transformovaná data. Čtvrtým krokem je filtrace dat získaných v předchozím kroku pomocí filtrace oknem. Posledním krokem algoritmu je aplikace Huffmanova kódovacího algoritmu.



Obr. 2.3: Blokový diagram kompresního a rekonstrukčního algoritmu [27]

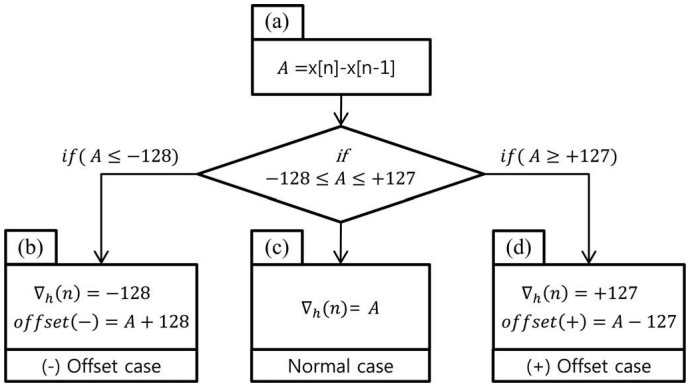
Obr. 2.3 naznačuje také postup rekonstrukce komprimovaného EKG signálu. Algoritmus rekonstrukce je obdobný jako algoritmus komprese, jednotlivé kroky jsou však inverzní a jsou prováděny v opačném pořadí. Prvním krokem rekonstrukčního algoritmu je aplikace inverzního Huffmanova kódovacího algoritmu na komprimovaná a transformovaná data. Druhým krokem je inverzní diskrétní kosinová transformace. Ve třetím kroku je interpolován obnovený signál v čase. V této proceduře se využívá spline interpolace. Konečným krokem rekonstrukčního algoritmu je výpočet původního signálu pomocí inverzního rozdílu. [27]

Podvzorkování s poloviční vzorkovací frekvencí a signálový zpětný rozdíl

EKG je analogový signál, vzorkovaný obvykle při 200 Hz až 1 kHz v závislosti na účelu použití. Navzorkovaná data jsou většinou reprezentována jako 2-bajtová. V navrhovaném kompresním algoritmu je získaný EKG signál nejprve podvzorkován jednou polovinou a reprezentován jako 1-bajtová informace po výpočtu zpětného rozdílu, což snižuje velikost dat o 75 %. Vzhledem k tomu, že 1-bajtová data mohou být v desítkové soustavě reprezentována od -128 do 127, mohou být data původního signálu uložena ve formátu uvedeném v Tab. 2.1. První dva bajty jsou vyhrazeny pro navzorkovaná data originálního signálu a následující bajty pro 1-bajtová rozdílová data. Ve výjimečných případech, kdy je signál narušen impulsivním šumem nebo jakýmkoliv jiným druhem vysokofrekvenčního šumu, musíme pečlivě prozkoumat následky. Obr. 2.4 ukazuje, jak problémy s šumem vyřešit. V těchto situacích algoritmus přiřazuje 2-bajty. V prvním bajtu zapisuje -128 (+127). V druhém bajtu minus (plus) hodnota offsetu, kde rozdílová hodnota je mimo rozsah -128 až +127. Obr. 2.4 b) a d) ukazuje způsob získání minus nebo plus kompenzace. [27]

Tab. 2.1: Datový formát pro původní signál reprezentovaný použitím rozdílu vzorků [27]

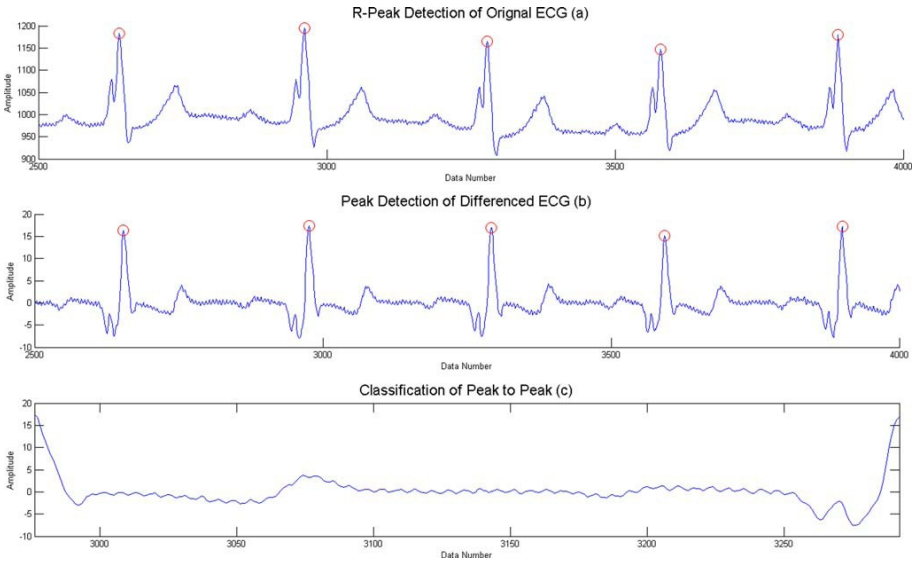
Data Header		Normal	**Under -128 case**		Normal	**Over 127 case**		Normal	Normal	Normal
2-Bytes (Header)		1-Byte	**1-Byte**	**1-Byte**	1-Byte	**1-Byte**	**1-Byte**	1-Byte	1-Byte (end-5)	1-Byte
Start (High-Byte)	Start (Low-Byte)	$\bigtriangledown_h(n)$	**-128**	**Offset(-)**	$\bigtriangledown_h(n+2)$	**127**	**Offset(+)**	$\bigtriangledown_h(n+4)$	...	$\bigtriangledown_h(n+end)$



Obr. 2.4: Schéma výpočtu rozdílu offsetu pro získání 1-bajtových bezeztrátových dat [27]

1-cyklová klasifikace rozdílových dat

Jednou z nejdůležitějších vlastností EKG signálu je jeho periodicita, která je zachována i po diskrétní kosinové transformaci. Periodicita EKG signálu umožňuje při Huffmanově kódování dosáhnout vysokého kompresního poměru, tedy vysoké komprese dat. Typický EKG signál má tři důležité opakující se úseky: P-vlnu, QRS komplex a T-vlnu. Pozice kmitu R je dobře viditelná i po rozdílu EKG signálu a může být použita jako referenční bod pro extrakci periody EKG signálu (Obr. 5.10). [27]



Obr. 2.5: Detekce píků. (a) Detekce R-vlny v původním signálu. (b) Detekce R-vlny v rozdílovém signálu. (c) Klasifikace úseků vln použitím rozdílových dat. [27]

Algoritmus *Down slope trace waveform* (DSTW) se používá pro detekci píků rozdílového EKG signálu, aby bylo možné snadno detekovat píky signálu v reálném čase. Algoritmus DSTW může být použit v různých oblastech. K detekci P-vlny, QRS komplexu, T-vlny v EKG, za účelem snížení nebo potlačení síťového rušení (brumu), kolísání nulové izolínie (driftu), nebo k detekci píků v signálu, který obsahuje různé druhy šumu. [23]

Diskrétní kosinová transformace

Poté, co algoritmus určí interval od píku k píku, aplikuje se na něj DCT v reálném čase. Aby mělo provedení vysoký kompresní poměr, velikost okna DCT, která je úměrná k výsledku DCT, se nastavuje od 15 % do 100 %. [27]

Huffmanovo kódování

Konečným krokem kompresního algoritmu je Huffmanův kódovací algoritmus. Huffmanovo kódování je využíváno pro bezeztrátovou kompresi dat. Algoritmus konvertuje znaky vstupního souboru do bitových řetězců různé délky. Znaky, které se ve vstupním souboru vyskytují nejčastěji, jsou konvertovány do bitových řetězců s nejkratší délkou. Nejfrekventovanější znak tak může být konvertován do jediného bitu. Znaky, které se vyskytují velmi zřídka, jsou konvertovány do delších řetězců. Mohou být i delší než 8 bitů. Nejjednodušší metoda komprese touto metodou probíhá ve dvou fázích. V první fázi algoritmus projde soubor a vytvoří statistiku četností každého znaku. Ve druhé fázi se využije této statistiky pro vytvoření binárního stromu a k následné kompresi vstupních dat.

Rekonstrukce komprimovaných EKG dat

Rekonstrukce komprimovaných dat se provádí ve stejných krocích jako komprese, ale v opačném směru. Postup rekonstrukce je znázorněn na pravé straně Obr. 2.3. Prvním krokem je inverzní Huffmanovo kódování. Druhým krokem postupu je inverzní DCT a třetím rekonstrukce podvzorkovaného signálu pomocí kubického spline algoritmu. Posledním krokem je zpětná rekonstrukce rozdílu s použitím vzorců (3.2) a (3.3). $X_r(0)$ jsou první obnovená data získaná přidáním 2 bajtů z originálních dat (start) a první rozdílová data $\bigtriangledown_h(0)$. Další originální vzorek se získá pomocí vzorce (3.3), kde jsou další rozdílová data přidaná do zrekonstruovaných hodnot v předchozím kroku. [27]

$$X_r = Start + \bigtriangledown_h(0), n = 0 \tag{2.2}$$

$$X_r(n) = X_r(n-1) + \bigtriangledown_h(n), n \neq 0 \tag{2.3}$$

2.2.2 Komprese EKG s využitím vlnkové transformace

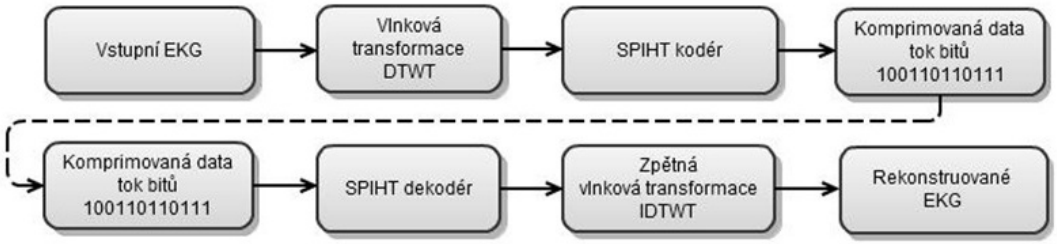
Vlnková transformace je vhodná pro kompresi signálů obsahující vysokofrekvenční složky s relativně krátkým trváním. Takovými signály jsou právě EKG záznamy. QRS komplex obsahuje nejvyšší frekvenční složky signálu a pokrývá přibližně 10 % každého srdečního cyklu. Z tohoto důvodu se určuje nejnižší vzorkovací frekvence EKG signálu na 500 Hz. Užití dyadické vlnkové transformace umožňuje analyzovat signál rozkladem do frekvenčních pásem s dokonalou časovou lokalizací vysokofrekvenčních složek.

V následujícím textu je stručně popsán v současné době nejperspektivnější algoritmus pro kompresi EKG signálů - algoritmus SPIHT.

Algoritmus SPIHT

Kompresní algoritmus SPIHT (z angl. Set Partitioning In Hierarchical Trees) byl původně navržen pro kompresi obrazových dat. Modifikace tohoto algoritmu pro kompresi jednorozměrných dat (především signálů EKG) pak byla představena v [28].

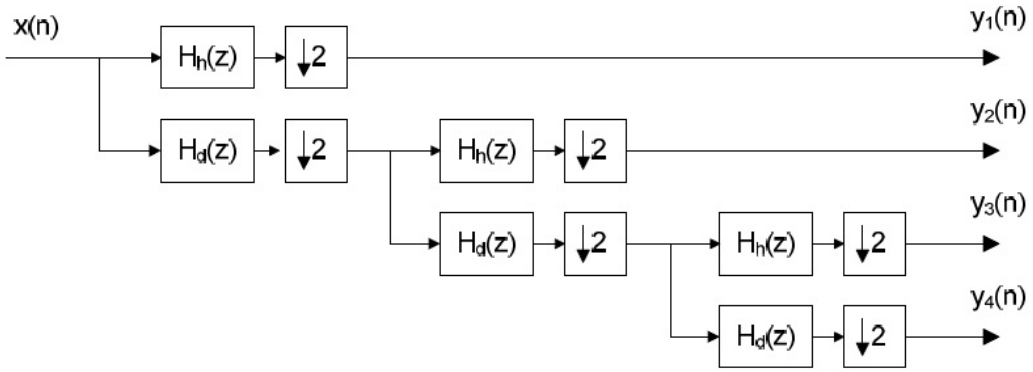
Kompresní metodu signálů EKG využívající algoritmus SPIHT (Set Partitioning in Hierarchical Trees) je možno popsat obecným schématem dle Obr. 2.6. [28]



Obr. 2.6: Schéma zpracování signálu pomocí algoritmu SPIHT [15]

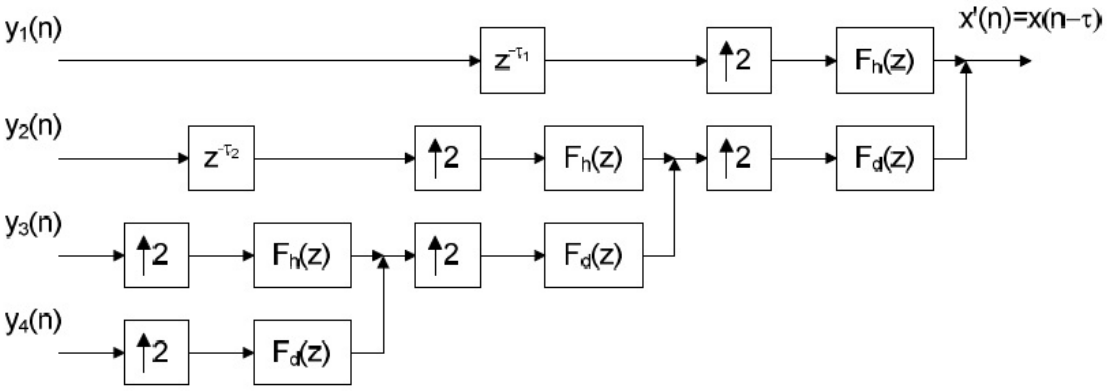
Originální signál EKG je nejprve rozložen pomocí vlnkové transformace na jednotlivá pásma. Pro kompresi signálů se výhradně používá dyadická forma rychlé vlnkové transformace s diskrétním časem DTWT. Na rozklad vlnkovou transformací navazuje kodér algoritmu SPIHT, který pro kompresi využívá dočasných orientovaných stromů a jejich rozdělování. Výstupem kodéru jsou komprimovaná data ve formě binární posloupnosti. Jedná se o progresivní kódování, které může být kdykoliv ukončeno. Algoritmus je možné použít i jako bezeztrátový. Komprimovaná data lze následně rekonstruovat s využitím dekodéru algoritmu SPIHT, na který navazuje zpětná vlnková transformace s diskrétním časem IDTWT.

Na Obr. 2.7 je naznačena realizace dyadické DTWT s podvzorkováním a třemi stupni rozkladu, kde $H_h(z)$ značí rozkladovou horní propust a $H_d(z)$ rozkladovou dolní propust.



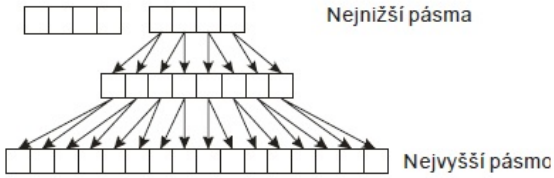
Obr. 2.7: Realizace rychlé dyadické DTWT se třemi stupni rozkladu [15]

Realizace odpovídající dyadické IDTWT je uvedena na Obr. 2.8, kde $F_h(z)$ značí rekonstrukční horní propust a $F_d(z)$ rekonstrukční dolní propust.



Obr. 2.8: Realizace zpětné DTWT[15]

Podstatou ztrátové komprese je co možná nejúspornější vyjádření vlnkových koeficientů $y(n)$, při kterém je ještě poškození rekonstruovaného signálu v přijatelných mezích. Na účinnost komprese a výsledné zkreslení rekonstruovaného signálu má vliv zejména volba rozkladových a rekonstrukčních filtrů a počtu stupňů rozkladu [15]. Na rozklad signálu dyadickou DTWT navazuje kodér algoritmu SPIHT. Koeficienty v jednotlivých pásmech vlnkové transformace spolu souvisí. Tyto souvislosti je možné vyjádřit pomocí stromové struktury (Obr. 2.9).



Obr. 2.9: Stromová struktura souvislostí koeficientů vlnkové transformace [15]

U signálů EKG bývá nejvíce energie soustředěno v nízkofrekvenčním pásmu vlnkové transformace (nejnižší pásmo), které obsahuje nejméně vlnkových koeficientů, tzv. kmenů stromové struktury. Algoritmus kóduje důležité koeficienty, což jsou koeficienty, které jsou nadprahové a přitom sníží jejich velikost o hodnotu prahu. Dále jsou kódovány stromy nedůležitých koeficientů. V okamžiku, kdy nezbývá žádná nadprahová hodnota, se sníží velikost prahu na polovinu. Znovu se kódují koeficienty, které byly v předcházejících iteracích nadprahové a nově i hodnoty nyní překračující práh. Postupně tak dochází ke štěpení stromů nedůležitých koeficientů a kódování stále větších podrobností.

Algoritmus SPIHT pracuje se třemi seznamy: LIP, LIS, LSP.

- LIP: Seznam nedůležitých koeficientů (List of Insignificant Pixels)
  - obsahuje jednotlivé nedůležité koeficienty
- LIS: Seznam nedůležitých množin (List of Insignificant Sets)
  - obsahuje stromy nedůležitých koeficientů
- LSP: Seznam důležitých koeficientů (List of Significant Pixels)
  - obsahuje jednotlivé důležité koeficienty

Stromy koeficientů mohou být dvojího typu:

- Typ D
  - zjišťuje se důležitost všech následovníků kořenového (výchozího) prvku
- Typ L
  - zjišťuje se důležitost všech následovníku kořenového prvku kromě přímých potomků

Během řadícího průchodu jsou zkontrolovány záznamy v seznamech LIP a LIS a důležité koeficienty jsou přesunuty do seznamu LSP. Pokud je důležitý koeficient součástí stromu, je aplikováno pravidlo na rozdělení množiny a koeficient se stává kořenovým prvkem nového stromu. Jelikož kořenový prvek stromu bývá většinou důležitý, je nezávisle otestován, zda je opravdu důležitý.

Následuje upřesňující průběh, při kterém je pro zvýšení přesnosti koeficientů ze seznamu LIP poslán na výstup následující bit z binární reprezentace jejich hodnot. [15]

Průběh algoritmu

1. Výpočet počátečního prahu.

- Inicializace množiny LIP – všechny koeficienty, které jsou kořenovými prvky stromů v nejnižších pásmech rozkladu.
- Inicializace množiny LIS – všechny stromy, na začátku označeny jako typ D.
- Množina LSP na začátku neobsahuje žádné koeficienty.

2. Otestuje se zda jsou koeficienty v LIP důležité:

- ANO – koeficient je důležitý: na výstup se pošle 1 následována znaménkovým bitem a koeficient je přesunut do LSP,
- NE – koeficient není důležitý: na výstup se pošle 0.

3. Otestuje se důležitost všech stromů v LIS – strom je důležitý pokud je některý následovník důležitý (testují se následovníci dle typu stromu D/L):

Typ D:
- ANO – strom je důležitý: na výstup se pošle 1 a přímí potomci jsou otestování obdobně jako v bodě 2):
  - ANO – přímý potomek je důležitý: na výstup se pošle 1 a je přidán do LSP,
  - NE – přímý potomek není důležitý: na výstup se pošle 0 a je přidán na konec LIP,
  - Pokud přímí potomci mají další následovníky:
  - ANO: přesunutí stromu na konec LIS jako typ L,
  - NE: odstranění stromu z LIS,
- NE – strom není důležitý, na výstup se pošle 0,

Typ L:
- ANO (strom je důležitý): na výstup se pošle 1 a všichni přímí potomci jsou přidání na konec LIS jako typ D; rodičovský strom se odstraní z LIS
- NE (strom není důležitý): na výstup se pošle 0

4. Snížení prahu a opakování postupu od bodu 2., dokud není dosaženo požadovaného počtu bitů na výstupu. [15]

2.3 Hodnocení komprimačních algoritmů

V publikacích [8], [11], [12], [17], [28], [36] a [39] byly představeny různé techniky komprese EKG signálů, které se liší různou účinností algoritmu komprese dat. Obecně lze kvalitu komprimačního algoritmu hodnotit na základě dvou parametrů. Prvním parametrem je účinnost komprese, tím druhým je zkreslení rekonstruovaného signálu. Je zřejmé, že se jedná o protichůdné požadavky. Vyšší účinnost komprese obecně vede k vyšší míře poškození rekonstruovaného signálu. Úkolem je tedy nalezení takové úrovně komprese, při které je poškození rekonstruovaného signálu ještě přijatelné. Podle [27] se k hodnocení komprimačních algoritmů používá následujících šest parametrů.

1. CR - Kompresní poměr (Compression ratio) je definován:

$$CR = \frac{N_{inp}}{N_{out}} \tag{2.4}$$

kde N_{inp} je velikost původního signálu a N_{out} velikost komprimovaného signálu.

2. PRD hodnota (Percent Root-mean-square Difference) je reprezentovaná:

$$PRD(\%) = 100 \times \sqrt{\frac{\sum_{N-1}^{n=0}(X_s(n) - X_r(n))^2}{\sum_{N-1}^{n=0}(N_s(n))^2}} \tag{2.5}$$

3. PDRN hodnota je definovaná:

$$PRDN(\%) = 100 \times \sqrt{\frac{\sum_{N-1}^{n=0}(X_s(n) - X_r(n))^2}{\sum_{N-1}^{n=0}(N_s(n) - \bar{X})^2}} \tag{2.6}$$

4. RMS - Kvadratická odchylka (Root mean square) je definována:

$$RMS = \sqrt{\frac{\sum_{N-1}^{n=0}(X_s(n) - X_r(n))^2}{(N-1)}} \tag{2.7}$$

5. SNR - Poměr signálu a šumu (Signal-to-noise ratio) může být vypočítán podle:

$$SNR = 10 \times \log(\frac{\sum_{N-1}^{n=0}(X_s(n) - \bar{X})^2}{\sum_{N-1}^{n=0}(N_s(n) - X_r(n))^2}) \tag{2.8}$$

6. QS - Skóre kvality (Quality-score) je rovno:

$$QS = \frac{CR}{PRD} \tag{2.9}$$

3 TECHNOLOGIE ETHERNET

Ethernet je souhrnný název představující v současné době nejrozšířenější technologii pro budování počítačových sítí typu LAN (lokálních sítě - domácí nebo firemní sítě). Komerčně byl zaveden v roce 1980 a brzy nahradil konkurenční kabelové LAN technologie. Systémy komunikující před Ethernet rozdělí datový tok do jednotlivých paketů, tzv. rámců. Každý rámec obsahuje zdrojové a cílové adresy, kontrolu dat, která detekuje poškozená data a zamezuje tomu, aby byla přenášena. Standardy definují několik variant zapojení. Původní 10BASE5 Ethernet používá koaxiální kabel jako sdílené médium. Později byly koaxiální kabely nahrazeny kroucenými dvojlinkami a optickými vlákny ve spojení s rozbočovači a přepínači. Přenosové rychlosti dat vzrostly z původních 10 megabitů za sekundu na 100 gigabitů za sekundu.

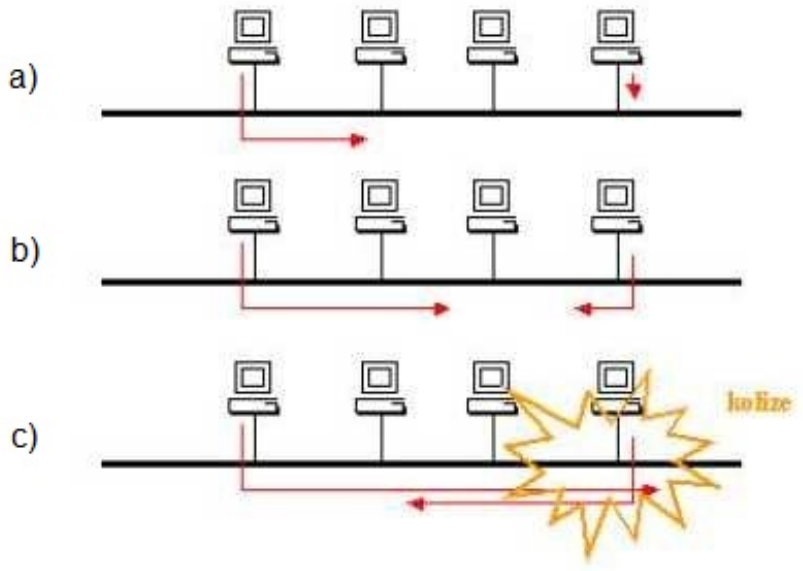
3.1 Princip CSMA/CD

Technologie Ethernetu je nezávisle na tom, zda se jedná o klasický 10 Mbps Ethernet nebo jeho rychlejší mutaci (Fast a Gigabit Ethernet), založena na velice jednoduchém principu, nazývaném CSMA/CD.

CSMA (Carrier Sense Multiple Access) - stanice připravená vysílat data si „poslechne“, zda přenosové médium (kabel) nepoužívá jiná stanice. V případě, že ano, stanice zkouší přístup později (v náhodně stanovenou dobu) až do té doby, dokud není médium volné. V okamžiku, kdy se médium uvolní, začne stanice vysílat svá data.

CD (Collision Detection) - stanice během vysílání sleduje, zda je na médiu signál odpovídající vysílaným úrovním. Tedy, aby se např. v okamžiku, kdy vysílá signál 0, nevyskytl signál 1. Případ, kdy dojde k interakci signálů více stanic, se nazývá kolize. V případě detekce kolize stanice generuje signál JAM a obě, popřípadě všechny stanice, které v daném okamžiku vysílaly, generují náhodnou hodnotu času, po níž se pokusí vysílání zopakovat.

Na Obr. 3.1 si stanice vlevo, která potřebuje vysílat, poslechla, zda někdo vysílá. Protože zjistila, že je na přenosovém médiu klid, začala posílat data. V okamžiku, kdy ještě její signál nedorazil ke stanici vpravo, si stanice vpravo ověřila stav média a zahájila vysílání. Dochází tedy k situaci, kdy data posílají obě stanice současně a nastane kolize. Stanice vpravo generuje signál JAM, všechny vysílající stanice zastavují vysílání a generují náhodné číslo.



Obr. 3.1: Princip Collision Detection[44]

Výše uvedené řešení přináší významnou nevýhodu. S narůstajícím počtem uzlů narůstá počet kolizí, a tím i teoretická propustnost sítě. Soubor uzlů, jejichž vzájemná činnost může vygenerovat kolizi, se nazývá kolizní doména, ta by měla být co nejmenší. Používané aktivní prvky mají ke kolizní doméně rozdílný vztah. Některé kolizní doménu rozšiřují, jiné kolizní domény oddělují. Jejich volbou lze propustnost sítě ovlivnit.

Vedle pojmu kolizní doména existuje pojem broadcastová doména. Na počítačové síti se vyskytují principielně dva typy paketů – tzv. unicasty a nonunicasty. Unicasty jsou pakety, které mají konkrétního adresáta vyjádřeného regulérní síťovou adresou. Nonunicasty používají skupinovou adresu a jsou určené buď všem uživatelům sítě (broadcasty), nebo vybrané skupině uživatelů (multicasty). Problém je v tom, že nonunicastu se musí počítač věnovat, i když není určen pro něj. S nárůstem počtu uzlů v broadcastové doméně narůstá i množství nonunicastů. Z tohoto důvodu je nutné udržet velikost broadcastové domény v rozumné velikosti. Používané aktivní prvky mají k broadcastové doméně opět rozdílný vztah. [44]

3.2 Přenosová média

3.2.1 Koaxiální kabel

Původní Ethernet byl propojován tzv. tlustým koaxiálním kabelem a označoval se jako 10Base5. Jeden segment mohl být dlouhý až 500 metrů. Na kabel byly napi-

chovány transceivery, které se připojovaly na AUI port síťové karty.

K masovému používání Ethernetu došlo se zavedením tzv. tenkého koaxiálního kabelu. Tato varianta se označuje jako 10Base2. Propojovací kabely se zakončují BNC konektory, mezi ně se vkládají BNC-T konektory. Ty se připojují přímo na síťovou kartu, nebo adaptérem na AUI port. Délka segmentu je maximálně 185 metrů, ve speciálních případech až 300 - 400 metrů. Segment se nesmí nijak větvit, je to pouze jeden dlouhý kabel pospojovaný z jednotlivých úseků mezi stanicemi, je zakončený na obou koncích terminátorem.

3.2.2 Kroucená dvojlinka

Kroucená dvojlinka je dnes zdaleka nejrozšířenější druh Ethernetové kabeláže. Topologie sítě se změnila ze sběrnicové na hvězdicovou, v jejímž středu je rozbočovač (hub) a na koncích jednotlivých spojů připojené počítače. Chování sítě napodobuje sběrnici – rozbočovač kopíruje signál přicházející z jednoho rozhraní do všech ostatních. Data vysílaná jednou stanicí jsou proto rozšířena všem ostatním, stejně jako v případě jejich přenosu po sdílené sběrnici.

Rozbočovače jsou dnes většinou nahrazovány přepínači (switch), které jsou na rozdíl od nich inteligentní. Pracují na principu „ulož a předej“. Přijmou ethernetový rámec, uloží si jej do vyrovnávací paměti, analyzují adresu jeho příjemce a následně jej odvysílají do rozhraní, kterým je připojen jeho adresát. Tabulky s fyzickými adresami a jim odpovídajícími rozhraními si udržují automaticky. Učí se na základě adresy odesílatele v rámcích. Přepínač nepředává rámec rovnou, ale po uložení jej sám odvysílá až v okamžiku, kdy bude na cílovém rozhraní volno. Počítače (či sítě) připojené k jeho rozhraním spolu navzájem nesoutěží o přístup k médiu. Na každém rozhraní přepínače běží nezávislý algoritmus CSMA/CD a o médium spolu soutěží jen zdejší počítače. Přepínač tzv. odděluje kolizní domény. Důsledkem je vyšší propustnost sítě a také vyšší bezpečnost, protože data jsou doručována jen tam, kde sídlí jejich příjemce.

Rozvod kroucené dvojlinky v budovách se nazývá strukturovaná kabeláž. Každá zásuvka je propojena s centrálním rozvaděčem samostatným kabelem, který umožňuje její využití i pro jiné účely. Délka jednoho spoje je maximálně 100 metrů, ve strukturované kabeláži se používá limit 90 metrů a 10 m se ponechává pro propojení mezi zásuvkou a počítačem. Ethernet používající kroucenou dvojlinku se označuje příponou T nebo TX.

3.2.3 Optické vlákno

Ethernet je definován i pro optické vlákno. Používají se jednovidová i mnohovidová vlákna v závislosti na požadované rychlosti a vzdálenosti. Vybudování optické trasy je dražší než strukturovaná kabeláž, ale umožňuje přenos na vyšší vzdálenosti. Spojení je odolné proti elektromagnetickému rušení a koncové body spoje jsou galvanicky oddělené. Je tedy vhodné pro budování LAN sítí mezi budovami a vzdálenými lokalitami.

Skleněná vlákna jsou zakončena tzv. media konvertory, které převedou optický signál na elektrický. Převodník bývá obvykle součástí přepínače jako rozšiřující modul. Pro každý spoj se použijí dvě vlákna, pro každý směr jedno. Lze použít také jen jedno, kdy se využívá dvou vlnových délek pro přenos informací. V praxi se pokládá vždy několik vláken navíc jako rezerva pro rozšíření nebo poruchu. Délka optického spoje bývá od stovek metrů až po mnoho kilometrů. Rychlost přenosu může být od 10 Mbit/s až po gigabitové rychlosti. Optický Ethernet se označuje v příponě písmenem F, FX, SX, LX, EX a dalšími.

Kabely mohou být nestíněné (UTP - Unshielded Twisted Pair) a stíněné (STP - Shielded Twisted Pair). Stíněné se používají v průmyslovém prostředí, protože jsou odolnější proti rušení. Používá se stínění celého kabelu, nebo i jednotlivých párů. Provedení strukturované kabeláže se dělí na kategorie podle svých elektrických a přenosových vlastností. Na kategorii závisí maximální možná přenosová rychlost.

3.3 Verze Ethernetu

3.3.1 Ethernet

Jedná se o původní variantu s přenosovou rychlostí 10 Mbit/s. Je definována pro koaxiální kabel, kroucenou dvojlinku a optické vlákno.

3.3.2 Fast Ethernet

Fast Ethernet je rychlejší verze Ethernetu s přenosovou rychlostí 100 Mbit/s. Převzala maximum prvků z původního Ethernetu (formát rámce, algoritmus CSMA/CD apod.), aby se usnadnil, urychlil a zlevnil vývoj. V současnosti ji lze považovat za základní verzi Ethernetu. Je k dispozici pro kroucenou dvojlinku a optická vlákna.

3.3.3 Gigabitový Ethernet

Gigabitový Ethernet zvýšil přenosovou rychlost na 1 Gbit/s. Opět obsahuje upravené prvky původního Ethernetu, teoreticky i algoritmus CSMA/CD. V praxi je ale gigabitový Ethernet provozován pouze přepínaně s plným duplexem. Důležité je především použití stejného formátu rámce. Původně byl definován pouze pro optická vlákna, později byla doplněna i varianta pro kroucenou dvojlinku.

3.3.4 Desetigigabitový Ethernet

Desetigigabitový Ethernet představuje zatím poslední standardizovanou verzi. Přenosová rychlost činí 10 Gbit/s, jako médium zatím slouží hlavně optická vlákna a opět používá stejný formát rámce. Algoritmus CSMA/CD byl definitivně opuštěn, tato verze pracuje vždy plně duplexně. Byla vyvinuta i jeho specifikace pro kroucenou dvojlinku.

4 TCP A UDP PAKETY

4.1 TCP/IP protokoly

TCP/IP (Transmission Control Protocol/Internet Protocol) v překladu znamená primární transportní protokol - TCP a protokol síťové vrstvy - IP. Rodina protokolů TCP/IP obsahuje sadu protokolů pro komunikaci v počítačové síti a je hlavním protokolem celosvětové sítě Internet. Komunikační protokol je množina pravidel, které určují syntaxi a význam jednotlivých zpráv při komunikaci.

4.1.1 Architektura TCP/IP

Síťová komunikace je rozdělena do tzv. vrstev, které znázorňují hierarchii činností. Výměna informací mezi vrstvami je přesně definována. Každá vrstva využívá služeb vrstvy nižší a poskytuje své služby vrstvě vyšší.

Komunikace mezi stejnými vrstvami dvou různých systémů je řízena komunikačním protokolem za použití spojení vytvořeného sousední nižší vrstvou. Architektura umožňuje výměnu protokolů jedné vrstvy bez dopadu na ostatní. Příkladem může být možnost komunikace po různých fyzických médiích - Ethernet, optické vlákno, sériová linka.

Architektura TCP/IP je členěna do čtyř vrstev:

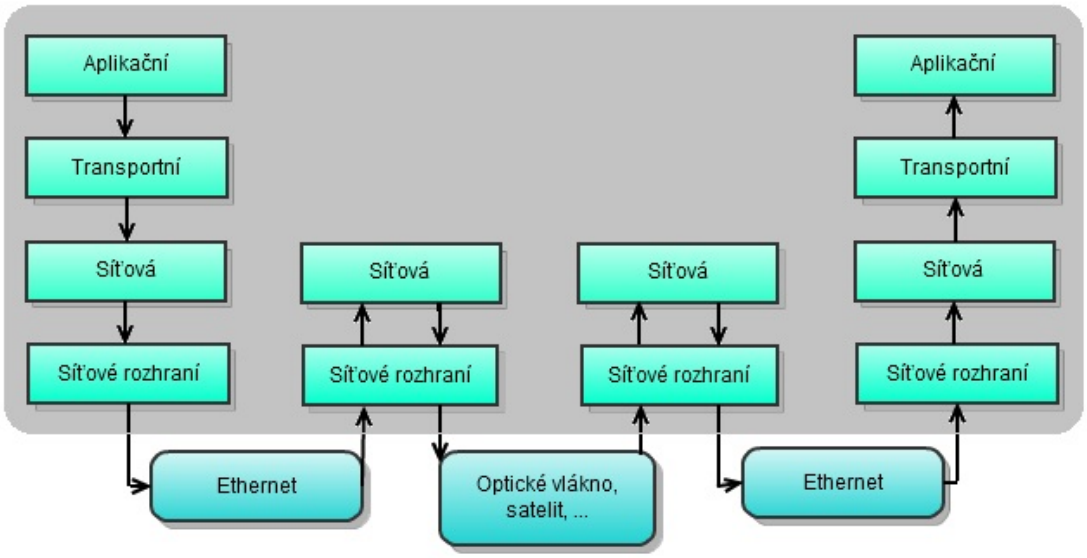
- Aplikační vrstva (Application Layer)
- Transportní vrstva (Transport Layer)
- Síťová vrstva (Network Layer)
- Vrstva síťového rozhraní (Network Interface)

Vrstva síťového rozhraní

Nejnižší vrstva umožňuje přístup k fyzickému přenosovému médiu. Je specifická pro každou síť v závislosti na její implementaci. Příklady sítí: Ethernet, Token ring, FDDI, X.25, SMDS.

Síťová vrstva

Vrstva zajišťuje především síťovou adresaci, směrování a předávání datagramů. Protokoly: IP, ARP, RARP, ICMP, IGMP, IGRP, IPSEC. Je implementována ve všech prvcích sítě - směrovačích i koncových zařízeních.



Obr. 4.1: Architektura TCP/IP

Transportní vrstva

Transportní vrstva je implementována až v koncových zařízeních (počítačích), a umožňuje proto přizpůsobit chování sítě potřebám aplikace. Poskytuje spojované (protokol TCP, spolehlivý) či nespojované (UDP, nespolehlivý) transportní služby.

Aplikační vrstva

Vrstva aplikací. To jsou programy (procesy), které využívají přenosu dat po síti ke konkrétním službám pro uživatele. Příklady: Telnet, FTP, HTTP, DHCP, DNS.

Aplikační protokoly používají vždy jednu ze dvou základních služeb transportní vrstvy: TCP nebo UDP, případně obě dvě (např. DNS). Pro rozlišení aplikačních protokolů se používají tzv. porty, což jsou domluvená číselná označení aplikací. Každé síťové spojení aplikace je jednoznačně určeno číslem portu, transportním protokolem a adresou počítače. [33]

4.2 Transportní protokoly

Síťová vrstva, která poskytuje služby adresace a směrování datagramů, nezajišťuje doručení datagramů adresátům, ani nezajišťuje pořadí jejich doručení či dobu jejich doručení. Proto nad síťovou vrstvou musí existovat vrstva starající se o koncový přenos datových jednotek mezi odesílatelem a příjemcem. Služba, kterou tato vrstva nabízí vyšším vrstvám, může být buď spolehlivá, nebo nespolehlivá. Spolehlivá služba garantuje doručení dat v pořadí jejich odeslání, podporuje libovolně dlouhé bloky dat, synchronizaci a řízení toku mezi odesílatelem a příjemcem na obranu proti zahlcení. Transportní vrstva nabízí transportní službu se spojením, nebo bez spojení za použití jednoho ze dvou protokolů.

- **Transmission Control Protokol (TCP)**
  - poskytuje transportní službu se spojením (spolehlivý protokol)
- **User Datagram Protokol (UDP)**
  - poskytuje transportní službu bez spojení (nespolehlivý protokol)

4.3 TCP - Transmission Control Protokol

Transmission Control Protokol je transportní protokol se spojením. Poskytuje logické spojení mezi koncovými aplikacemi, tedy spolehlivý přenos dat, který nebyl zajištěn datagramově orientovaným IP. TCP využívají ke své práci aplikační protokoly vyžadující spolehlivou transportní službu. TCP přijímá informace od vyšší vrstvy jako souvislý tok bajtů dat, který musí rozdělit do transportních segmentů délky odpovídající velikosti datagramu IP, a ty předá síťové vrstvě.

TCP specifikuje postupy pro spolehlivý přenos dat, formát dat a potvrzení, která si dva komunikující uzly vyměňují pro dosažení spolehlivého přenosu. Vykonává další funkce transportní vrstvy, jako řízení koncového zabezpečení a datového toku, řízení koncového spojení. TCP je v protikladu k UDP složitý protokol.

Jedním z hlavních principů TCP je nezávislost na nižší infrastruktuře, protože na ni spoléhá jen minimálně. Aplikační vrstvě poskytuje TCP službu plného duplexu, tj. data se mohou posílat v obou směrech, nezávisle na sobě.

4.3.1 Vlastnosti podporované TCP protokolem

- **Spolehlivá transportní služba** - doručí adresátovi všechna přenášená data tak, jak je odesílatel vyslal (tzn. bez ztráty nebo zkreslení dat či duplicitních

paketů) s využitím pozitivního potvrzování (na základě pořadových čísel bajtů) a opětovných přenosů

- **Služba se spojením** - fáze navázání spojení mezi dvěma procesy (potřebuje výměnu 3 segmentů), přenosu dat a ukončení spojení (potřebuje výměnu 4 segmentů)
- **Efektivní využití přenosových kanálů** - vysílání s využitím vyrovnávacích pamětí, kdy odesílatel začne vysílat, až se nashromáždí dostatek dat, nebo po vypršení časového limitu, zamezení zahlcení příjemce i sítě
- **Transparentní přenos** - libovolných dat vyššího protokolu
- **Plně duplexní spojení** - současný obousměrný přenos dat, potvrzení o správném příjmu TCP segmentu je součástí datového TCP segmentu vysílaného opačným směrem
- **Rozlišení mezi více potenciálními adresáty** - na daném počítači pomocí portů

4.3.2 Funkce TCP

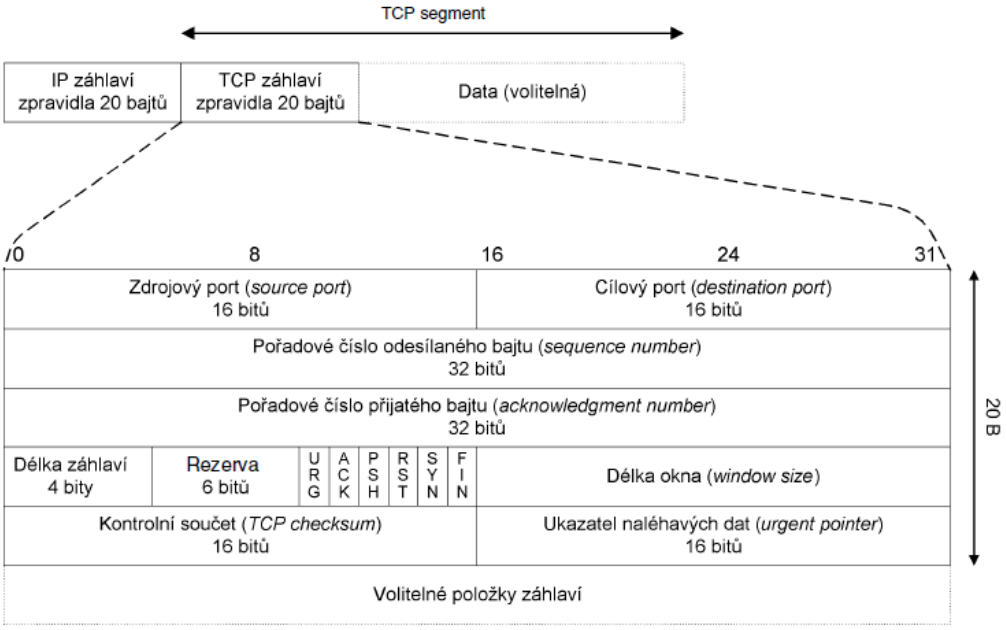
TCP vykonává řadu funkcí:

1. Asociuje porty se spojením
2. Navazuje spojení
3. Segmentuje data
4. Čísluje bajty dat
5. Specifikuje velikost okna (počet bajtů)
6. Vypočítává kontrolní součty
7. Reguluje tok dat
8. Signalizuje urgentní data
9. Pozitivně a kumulativně potvrzuje příjem dosud doručených bajtů dat
10. Nastavuje časomíru pro opětovné vysílání
11. Ukončuje spojení

TCP používají jako základ své činnosti pojem spojení. Spojení je definováno mezi dvěma koncovými body. Koncový bod je dvojice – stanice a port, kdy stanice znamená její IP adresu a port je TCP port na této stanici. Jakmile je spojení jednou navázáno, může být funkční po velmi dlouhou dobu, i když je v klidu a žádná data se po něm neposílají. TCP totiž nemá žádný mechanizmus, který by kontroloval životnost druhé strany.

4.3.3 Segment TCP

Segment TCP se zapouzdřuje do IP datagramu. Záhlaví TCP má délku 20 oktetů.



Obr. 4.2: TCP segment[10]

Záhlaví segmentu TCP obsahuje následující pole:

- **Zdrojový port** - identifikuje zdrojový aplikační proces (známým nebo uživatelským číslem portu)
- **Cílový port** - identifikuje cílový aplikační proces (známým nebo uživatelským číslem portu)
- **Pořadové číslo** - číslo prvního oktetu dat v segmentu, jeho pořadí v toku přenášených bajtů; při navazování spojení se nezačíná číslem 1, ale náhodným počátečním pořadovým číslem, odvozeným např. z vnitřních hodin
- **Číslo potvrzovací** - specifikuje pořadové číslo oktetu dat, které očekává jako následující
- **Délka záhlaví** - délka záhlaví v násobcích počtu 32 bitů
- **Funkce řízení** - každý ze šesti bitů tohoto pole má svůj přidělený význam související s datovým spojením:
  - **URG** - označuje urgentní data pro přednostní doručení v rámci daného spojení
  - **ACK** - označuje platnost pole s číslem potvrzení
  - **PSH** - požaduje okamžité doručení segmentu protokolu vyšší vrstvy

  - **RST** - požaduje okamžité ukončení spojení
  - **SYN** - je žádostí o navázání spojení (segment neobsahuje data)
  - **FIN** - žádost o ukončení spojení
- **Šířka okna** - velikost okna – počet oktetů, které lze vyslat bez průběžného potvrzení, odpovídá velikosti vyrovnávací paměti příjemce
- **Kontrolní součet** - zabezpečení přes celý segment
- **Označení urgentních dat** - specifikuje poslední oktet urgentních dat
- **Volitelné možnosti** - doplňkové informace (až 40 oktetů), např. maximální velikost segmentu

4.3.4 Fáze spojení TCP

TCP rozlišuje tři fáze komunikace s ohledem na spojení: navázání spojení, přenos dat a ukončení spojení.

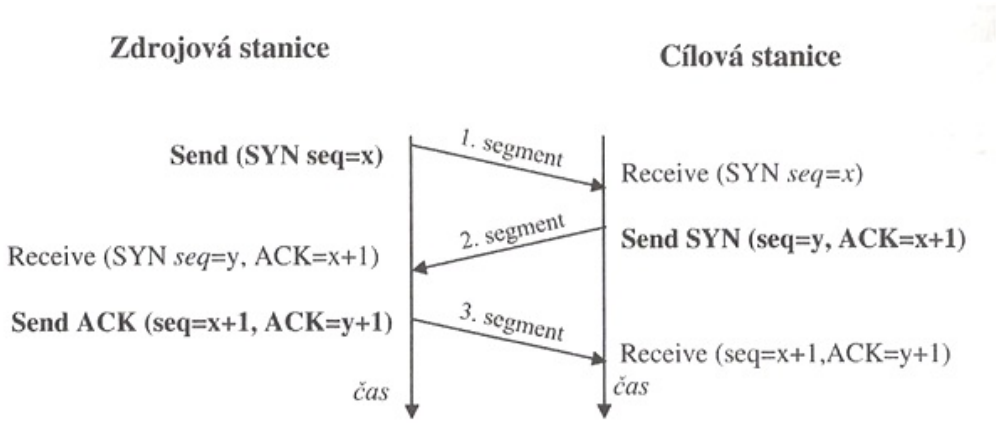
Navázání spojení

Pro navázání (otevření) spojení musí obě strany souhlasit s příjmem dat. Aplikační program na jedné straně provede pasivní otevření. Tím, že kontaktuje svůj operační systém, indikuje, že je ochotný přijmout příchozí spojení. Operační systém v tuto chvíli přidělí číslo portu danému spojení. Aplikační program na straně odesílatele také musí kontaktovat operační systém formou požadavku aktivního otevření spojení.

Každému přenosu podporovanému protokolem TCP musí předcházet povinná výměna tří segmentů v rámci navazování spojení, než dojde k vyslání prvního segmentu s uživatelskými daty. Dokud není spojení navázáno, musí TCP software udržovat data připravení k odeslání. U většiny implementací TCP se přijme žádost o navázání spojení, pokud má příjemce prostor ve své vstupní vyrovnávací paměti.

Mechanismus „trojitého potřásání rukou“ je naznačen na Obr. 4.3. Prvním krokem je vyslání synchronizačního segmentu s označením SYN v poli řízení TCP a se specifikací náhodně vybraného čísla (mezi 0 a $2^{32} - 1$). Odpovědí je segment s potvrzením přijetí počátečního segmentu a nastavení „startovního“ náhodně vygenerovaného pořadového čísla pro opačný směr vysílání (SYN-ACK). Závěr fáze navazování spojení činí potvrzení příjmu odpovědi cílovou stanicí (ACK).

První segment fáze navázání spojení SYN je zcela specifický; neobsahuje žádná data, ale ani číslo potvrzení, ani šířku okna, která má smysl právě jen v souvislosti s potvrzením. Druhý segment je odpovědí na SYN, ten již obsahuje potvrzení, proto se označuje SYN-ACK. SYN a SYN-ACK obsahují také počáteční pořadové číslo



Obr. 4.3: Výměna segmentů ve fázi navazování spojení TCP, převzato z [33]

(ISN) pro vysílání každé z komunikujících stran. Z bezpečnostních důvodů má být ISN náhodné číslo. Třetí segment je klasické potvrzení ACK (s pořadovým číslem ISN+1), definuje velikost okna a posílá jej klient.

Příklad navazování spojení a počátku komunikace:

1. Klient vysílá: SYN ISN = 50
2. Server vysílá: SYN INS = 99, ACK = 51
3. Klient vysílá: ACK = 100
4. Klient vysílá: bajty 51-100
5. Server vysílá: ACK = 101
6. Klient vysílá: bajty 101-150

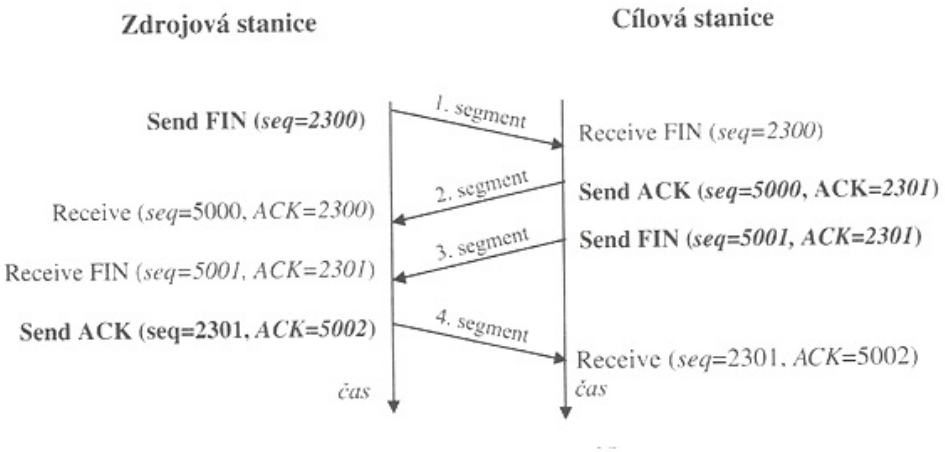
Přenos dat

Fáze ihned navazující na fázi navázání spojení zahrnuje přenos segmentů na základě pořadových čísel, pozitivního potvrzování a opětovného vysílání.

Ukončení spojení

Poslední fází spojení u TCP je ukončení spojení. Musí být provedeno z obou stran. Pokud je provedeno pouze z jedné strany, může druhá stanice dále posílat data, dokud sama spojení také neukončí. Ukončení spojení se provádí nastavením bitu FIN v poli řízení segmentu TCP, který musí být druhou stranou potvrzen jako jakýkoliv jiný segment.

Žádosti o ukončení spojení musí být dvě (z obou stran) a musí se jednotlivě potvrdit Obr. 4.4. Plnohodnotné ukončení spojení potřebuje výměnu čtyř segmentů, protože je třeba uzavřít oba směry plně duplexní komunikace.



Obr. 4.4: Výměna segmentů ve fázi ukončení spojení TCP, převzato z [33]

TCP umožňuje resetovat spojení v případě abnormální situace. Jedna strana může iniciovat okamžité ukončení prostřednictvím nastavení bitu RST. Druhá strana na takový segment reaguje okamžitým ukončením spojení.

4.3.5 Řízení toku TCP

Řízení toku TCP se realizuje několika spolupracujícími mechanizmy: velikostí okna, pořadovými čísly oktetů dat a potvrzováním.

Velikost okna

Záhlaví TCP segmentu obsahuje pole velikost okna. Toto pole specifikuje, kolik oktetů dat se může přenést od odesílatele k příjemci bez průběžného potvrzování doručení jednotlivých segmentů. Velikost okna není pevná hodnota, v průběhu komunikace lze velikost okna měnit, je tedy dynamická.

Počáteční velikost okna se nastaví při navazování spojení na základě velikosti paměti příjemce i odesílatele. Maximální velikost okna je obsažena v každém potvrzení a můžeme si ji snadno představit jako velikost vyrovnávací paměti na straně příjemce.

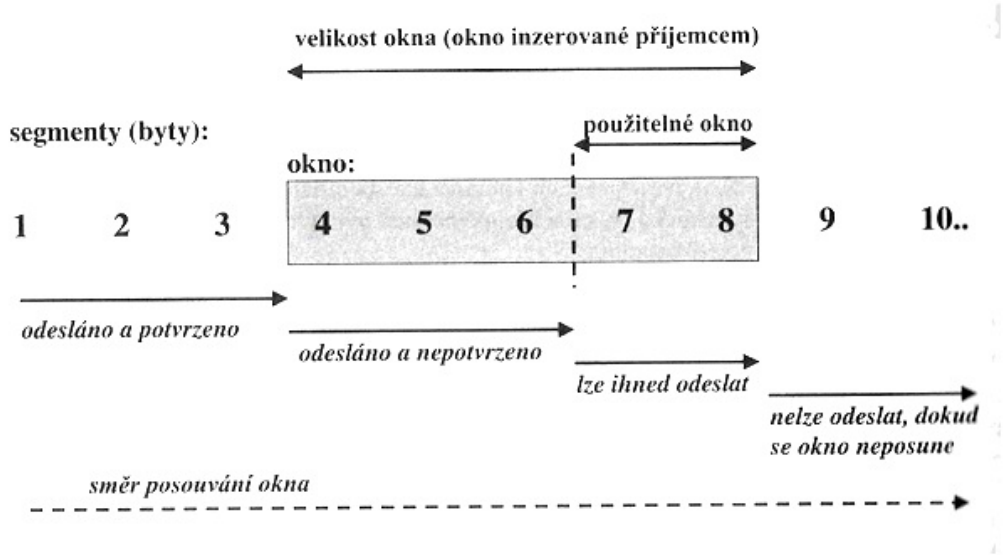
Pokud paměť příjemce na přijímaná data nestačí, inzeruje ve své odpovědi velikost okna nula. Tím signalizuje vysílací stanici, aby přestala vysílat. Po uvolnění paměti pak přijímací stanice opět inzeruje nenulovou velikost okna a indikuje svou připravenost přijímat data.

Velikost okna 1 vyžaduje potvrzovat příjem každého jednotlivého bajtu. Tento způsob přenosu je neefektivní. Z tohoto důvodu se využívá větší šíře okna, tj. většího

množství segmentů, které se mohou odesílat ihned za sebou a teprve poté získat na jejich doručení potvrzení. Typické velikosti okna TCP jsou 8.192 bajtů (stanice) a 24.000 bajtů (velké servery).

Klouzající okno

Po obdržení potvrzení příjmu dat může odesílatel vyslat opět skupinu oktetů podle velikosti okna. Tento mechanismus TCP se nazývá klouzající okno, protože se okno na vysílající straně posunuje (klouže) o příslušný počet bajtů dál v řadě segmentů čekajících na odeslání. Maximální počet dosud nepotvrzených vyslaných bajtů je dán velikostí okna. Vysílající strana si vždy musí ve své vyrovnávací paměti uchovat všechny segmenty, které odesílá, a to do té doby, než přijde potvrzení o jejich doručení. Pokud totiž potvrzení nepřijde do doby, než vyprší časovač pro opětovné vysílání, musí je znovu poslat.

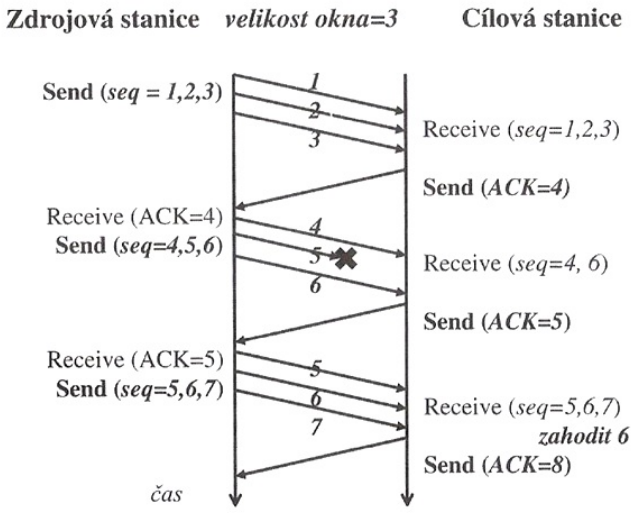


Obr. 4.5: Klouzající okno[33]

Potvrzování a opětovné vysílání

Základem spolehlivosti přenosu TCP je potvrzování přijatých dat. Potvrzování doručených oktetů dat provádí přijímací stanice na základě pořadových čísel bajtů. Potvrzování je pozitivní, tzn. potvrzuje se příjem všech oktetů předcházejících číslu potvrzení (to je číslo oktetu, které se očekává jako další, ACK). Potvrzování je u TCP kumulativní, tzn. potvrzují se postupně bajty v toku dat.

Při odesílání segmentu si vysílající stanice nastartuje svůj časovač pro přijetí potvrzení o obdržení segmentu cílovou stanicí. Pokud časovač vyprší dříve, než dorazí potvrzení přijetí dat segmentu, považuje vysílající stanice segment za ztracený a připraví jej k opětovnému odeslání i s bajty od čísla specifikovaného v posledním potvrzení. Většinou se odesílají i další bajty do velikosti okna, neboť tímto jednoduchým pozitivním potvrzováním lze obtížně specifikovat pouze chybějící segmenty.



Obr. 4.6: Posílání segmentů TCP s pořadovými čísly a jejich potvrzování, převzato z [33]

Potvrzení od cílové stanice slouží vysílající stanici k tomu, aby pokračovala ve vysílání. Chybějící potvrzení může znamenat, že odeslaný segment nebyl doručen, nebo se potvrzení ztratilo po cestě. V druhém případě pak příjemce duplicitní segmenty zahodí, aby zajistil opětovné vyslání segmentů.

Rychlé opětovné vysílání a rychlé zotavení

Mechanizmy rychlého opětovného vysílání a rychlého zotavení řeší situaci při doručení segmentu, který je mimo pořadí (s jiným než očekávaným pořadovým číslem). Jakmile příjemce takový segment obdrží, musí vyslat několik duplicitních potvrzení s pořadovým číslem chybějícího segmentu. Pro vysílací stranu znamená duplicitní potvrzení: zahozené segmenty, přerovnání pořadí datových segmentů sítí nebo duplikace segmentu potvrzení nebo dat sítí. Příjemce by měl poslat okamžité potvrzení v případě, kdy dorazí chybějící segment scházející v celkovém pořadí. A to kumulativním potvrzením segmentů, které mezitím mohly dorazit.

Vysílající strana by měla použít mechanizmus rychlého opětovného vysílání na zjištění a nápravu ztrát založených na příchodu duplicitních potvrzení. Tři kopie jednoho potvrzení znamenají, že segment nedorazil. Po přijetí kopií potvrzení TCP vysílající strana okamžitě opětovně vyšle segment, který se jeví jako chybějící, a to bez čekání na vypršení časovače pro opětovné vysílání. Po této situaci následuje rychlé zotavení, protože se s největší pravděpodobností jedná o výjimečnou ztrátu jednoho segmentu, zatímco ostatní jsou již v paměti příjemce.

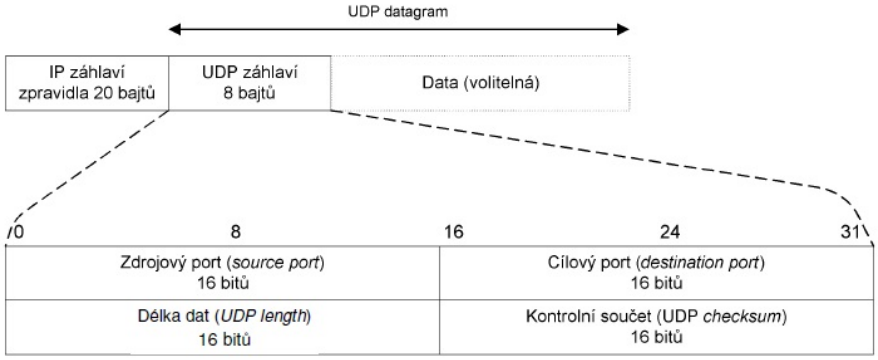
4.4 UDP - User Datagram Protokol

User Data Protokol je jednoduchou alternativou protokolu TCP. Jedná se o transportní protokol, který poskytuje nespolehlivou transportní službu. Využívají ho aplikace, které nepožadují zabezpečení přenosu v takovém rozsahu (doručení v pořadí, zajištění opakovaného přenosu). Odesílatel odešle UDP datagram příjemci a už se nestará o to, zda se datagram náhodou neztratil. Aplikační program využívající UDP musí převzít kompletní zodpovědnost za řešení problémů se spolehlivostí přenosu včetně ztrát zpráv, jejich duplicity, zpoždění, dodání mimo pořadí nebo ztrátu konektivity.

Na rozdíl od protokolu TCP je UDP protokol nespojovaná služba. Přenos dat u UDP protokolu nemá fázi navazování ani ukončování spojení. Již první segment UDP obsahuje uživatelská data. UDP podporuje vysílání na všeobecnou adresu IP a na skupinové adresy.

4.4.1 Formát datagramu UDP

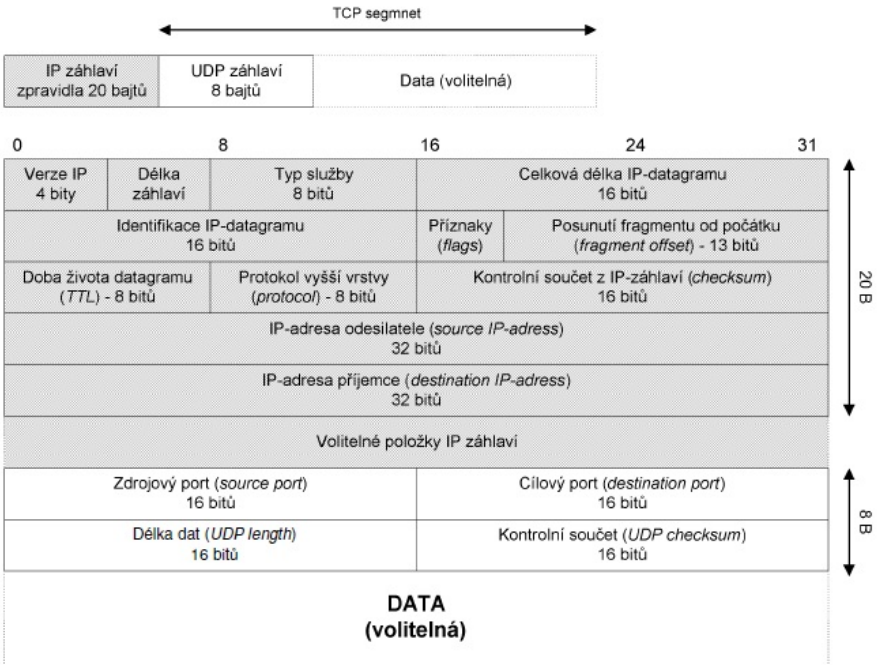
Datová jednotka UDP se nazývá datagram. UDP datagramy jsou baleny do IP datagramu, jak je znázorněno na Obr. 4.7.



Obr. 4.7: Záhlaví UDP datagramu[10]

Celková podoba UDB datagramu včetně jeho IP-záhlaví je znázorněna na Obr.4.8.

Výhodou protokolu UDP je malá hlavička dat. Ta obsahuje 4 položky. Základní informací je číslo portu příjemce. Číslo portu odesílatele je nepovinné, vyplňuje se pouze v případě, je-li požadována odpověď. Čísla portů protokolu UDP nesouvisí s čísly portů protokolu TCP. Protokol UDP má svou nezávislou sadu čísel portů. Další položkou v hlavičce datagramu je délka UDP datagramu (délka záhlaví + délku dat). Minimální délka UDP datagramu je 8 bajtů (UDP datagram obsahující pouze záhlaví a žádná data). Poslední položkou je kontrolní součet, který není povinný.



Obr. 4.8: UDP datagram[10]

4.4.2 Použití UDP protokolu

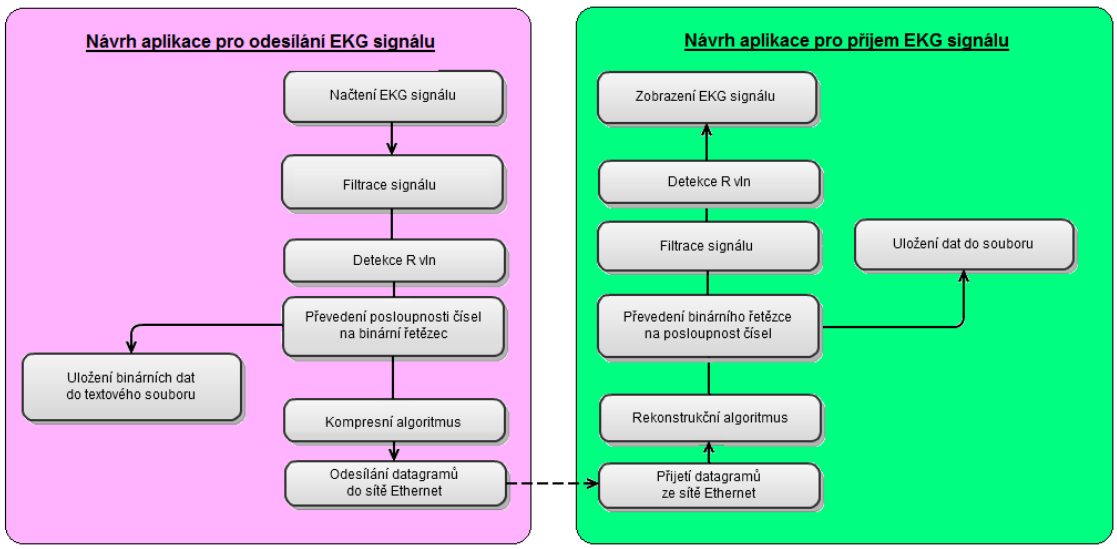
Výhodou UDP protokolu je malé zatížení sítě a schopnost rozlišit jednotlivé datagramy. UDP protokol je vhodné využívat při různých real-time přenosech multimediálních dat. Vhodný je například pro mobilní aplikace, protože klade menší nároky na kvalitu sítě.

5 REALIZACE V LABVIEW

5.1 LabVIEW

LabVIEW je vývojové prostředí využívající grafický programovací jazyk. Programy, které se v LabVIEW vytváří, se nazývají virtuální instrumenty (VI). Základními částmi každého VI je čelní panel a blokový diagram. Čelní panel obsahuje ovládací prvky, zatímco blokový diagram prvky funkční, které vykonávají jednotlivé operace. Obě části jsou vzájemně provázány.

5.2 Návrh programu



Obr. 5.1: Blokové schéma programu

5.3 Realizace návrhu

5.3.1 Popis aplikace

Úkolem vytvořené aplikace je přenos EKG signálu od odesílatele k příjemci po síti Ethernet. Aplikace je tvořena dvěma programy; jedním pro filtraci, kompresi a odesílání dat a druhým pro jejich příjem a rekonstrukci.

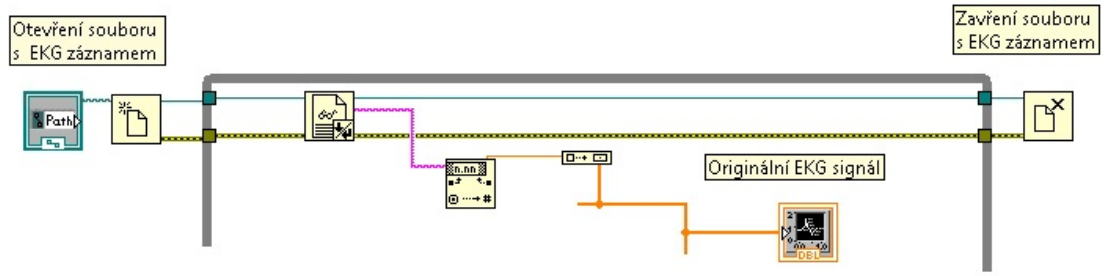
Data jsou do aplikace načítána z textového souboru uloženého v počítači odesílatele. Tento soubor otevírá uživatel v uživatelském prostředí programu před jeho

spuštěním. EKG signál je nejprve filtrován a jsou v něm detekovány R vlny. Následně je převeden na binární řetězec a v této podobě zálohován do textového souboru v počítači odesílatele. Před zahájením přenosu je binární řetězec komprimován. Pro přenos EKG signálu po síti Ethernet byl vybrán jednodušší z transportních protokolů - protokol UDP. Adresu příjemce dat vyplňuje odesílatel v uživatelském prostředí.

Aplikace pro příjem EKG signálu v počítači příjemce obsahuje rekonstrukční algoritmus. Rekonstruovaná data jsou uložena do textového souboru. EKG signál je v aplikaci příjemce filtrován, zobrazen a jsou v něm detekovány R vlny.

5.3.2 Načtení a zobrazení EKG signálu

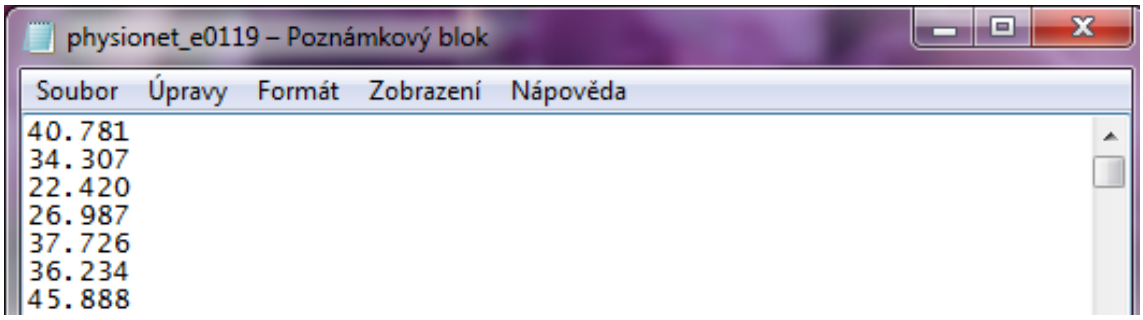
Část blokového diagramu zajišťující načtení a zobrazení EKG signálu je na Obr. 5.2.



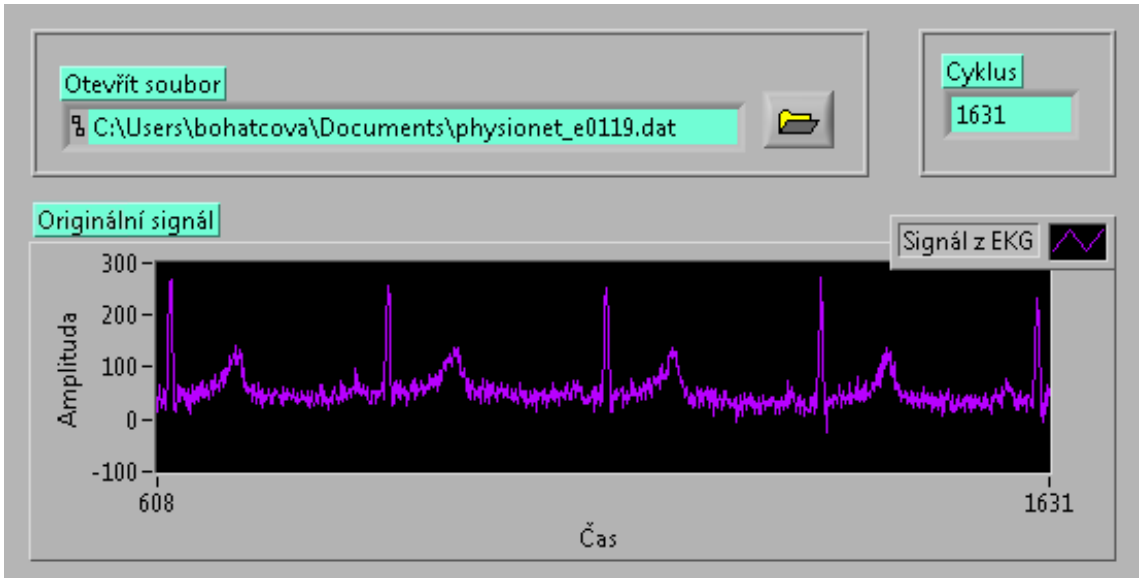
Obr. 5.2: Načtení EKG signálu ze souboru a zobrazení (blokový diagram)

Výchozím bodem aplikace je blok *Open File Function*, který se používá k vytvoření nového souboru, otevření existujícího souboru, a nebo k jeho nahrazení. V aplikaci je blok nastaven tak, aby otevíral existující soubor s EKG záznamem. Cestu k souboru nastavuje uživatel manuálně v dialogovém okně na čelním panelu aplikace (Obr. 5.4). Hodnoty EKG signálu jsou ze souboru do aplikace načteny pomocí bloku *Read from Text File Function*.

V ukázkovém souboru physionet_e0119.dat (Obr. 5.3) je EKG signál uložen jako sloupec amplitud jednotlivých vzorků. Aby bylo možné se signálem dále pracovat, je nutné změnit datový typ z textového řetězce na číslo. Tuto operaci provádí blok *Fract/Exp String To Number Function*. K zobrazení EKG signálu slouží prvek *Waveform Chart* vybraný z palety *Graph Indicators*. V nastavení zobrazovače jsou k dispozici jeho tři varianty: průběžný zapisovač, zapisovač obdobný osciloskopu a zapisovač s možností nastavení svislé značky počátku obnovy dat. V programu je použit průběžný zapisovač *Strip Chart*. Jedná se o výchozí režim, který zobrazuje data v pořadí, v jakém na výstup přišla. Je-li graf naplněn, vykreslená část se postupně posunuje doleva (Obr. 5.4).



Obr. 5.3: Ukázkový soubor physionet_e0119.dat



Obr. 5.4: Načtení a zobrazení originálního EKG signálu (čelní panel)

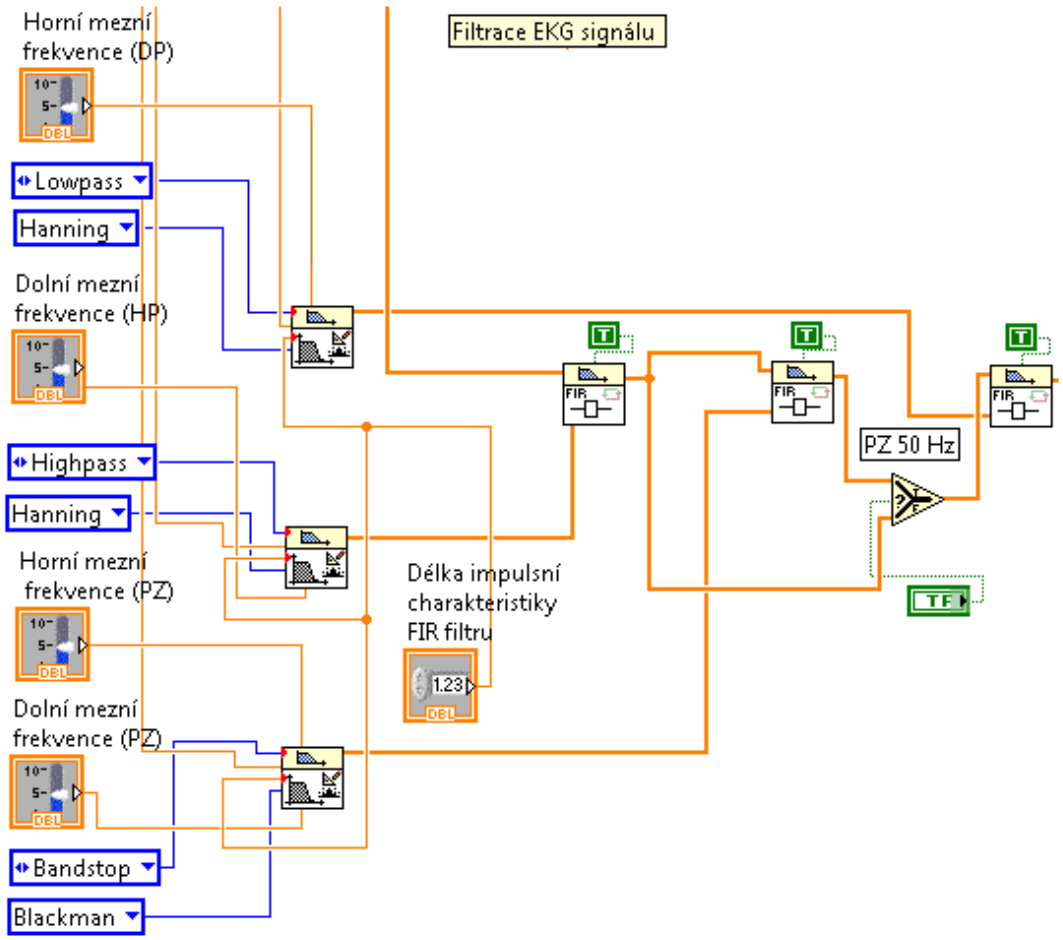
5.3.3 Filtrace EKG signálu

Mezi nejčastější zdroje rušení signálů EKG patří síťový brum (50 Hz + vyšší harmonické složky) a kolísání nulové linie signálu tzv. drift (přibližně do 2 Hz). Jedná se o úzkopásmové signály. Kolísání izolinie (drift) je způsobeno především elektrochemickými procesy na rozhraní kůže - elektroda, dýcháním pacienta (do 0,8 Hz) nebo jeho pomalými pohyby (do 2 Hz). Síťový brum vzniká v důsledku indukce napětí ze silových elektrických rozvodů. Zdrojem širokopásmové rušení EKG signálů (nad 100 Hz) jsou především myopotenciály, které vznikají při aktivním pohybu svalů. Jejich frekvenční pásmo se nachází v intervalu od jednotek Hz do jednotek kHz a překrývá tak frekvenční pásmo užitečného signálu.

Jedním z hlavních kroků předzpracování signálů EKG je jejich filtrace. Na zašuměný signál se aplikují filtry, jejichž cílem je potlačit v záznamu rušení. Zároveň je však nutné dbát na to, aby se nepoškodil užitečný signál, protože kmitočtová

spektra rušení a užitečného signálu se překrývají.

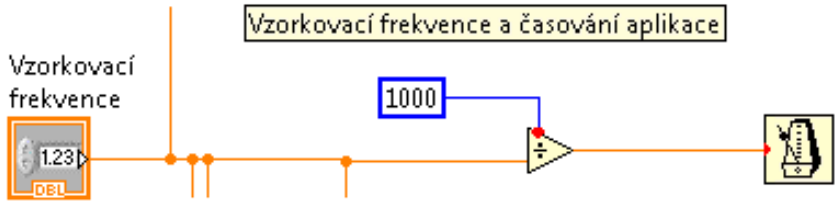
Pro potlačení driftu je v aplikaci použita horní propust (HP). Dalším filtrem je vypínatelná pásmová zádrž (PZ), která slouží k odstranění síťového rušení. Posledním filtrem před zobrazením signálu je dolní propust (DP), která potlačuje rušení vzniklé činností svalů.



Obr. 5.5: Filtrace signálu (blokový diagram)

Filtr se vždy skládá ze dvou částí: *FIR Windowed Coefficients* a *FIR Filter VI* (Obr. 5.5). Blok *FIR Windowed Coefficients* slouží k nadefinování filtru. Vstupem bloku je výběr typu filtru, nastavení délky impulsní charakteristiky, vzorkovacího kmitočtu, horní a dolní mezní frekvence. Výstupem filtru jsou koeficienty, které definují práci samotného filtru. Filtrování signálu zajišťuje blok *FIR Filter VI*. Vstupem filtru je vstupní signál a koeficienty získané z předchozího bloku. Výstupem bloku je filtrovaný signál (Obr. 5.7).

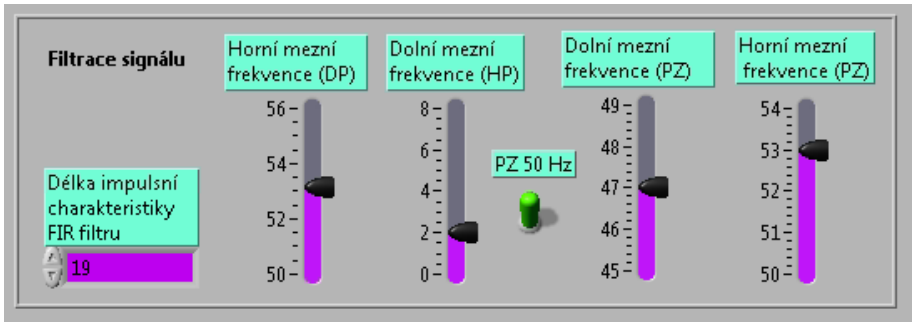
Mezní frekvence filtrů se nastavují pomocí posuvných ovladačů na čelním panelu



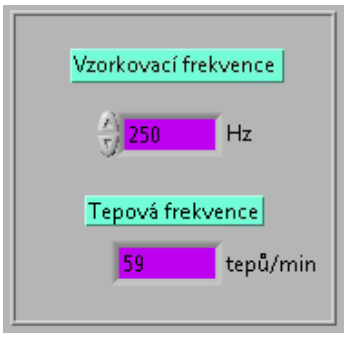
Obr. 5.6: Vzorkovací frekvence a časování aplikace (blokový diagram)

Obr. 5.7: Filtrovaný signál (čelní panel)

(Obr. 5.8). Kromě mezních frekvencí umožňuje čelní panel nastavit také vzorkovací frekvenci (Obr. 5.9). Obvyklé vzorkovací frekvence EKG přístrojů jsou 125, 250, 500 nebo 1000 Hz. Ukázkový záznam physionet_e0119.dat má vzorkovací frekvenci 250 Hz.



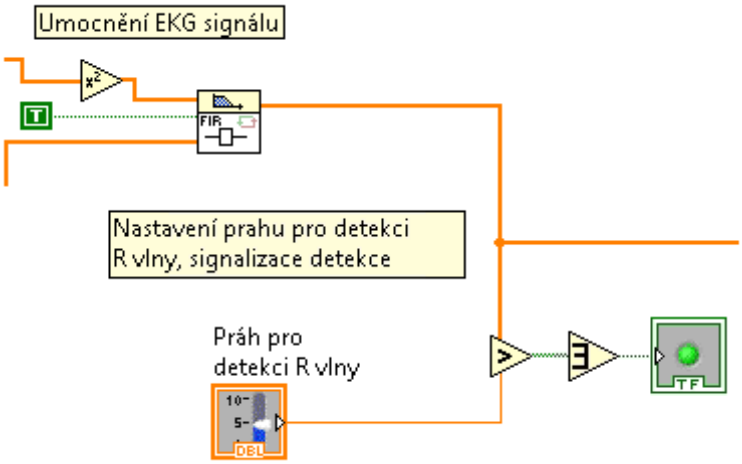
Obr. 5.8: Nastavení parametrů filtrace a prahu pro detekci R vlny (čelní panel)



Obr. 5.9: Nastavení vzorkovací frekvence (čelní panel)

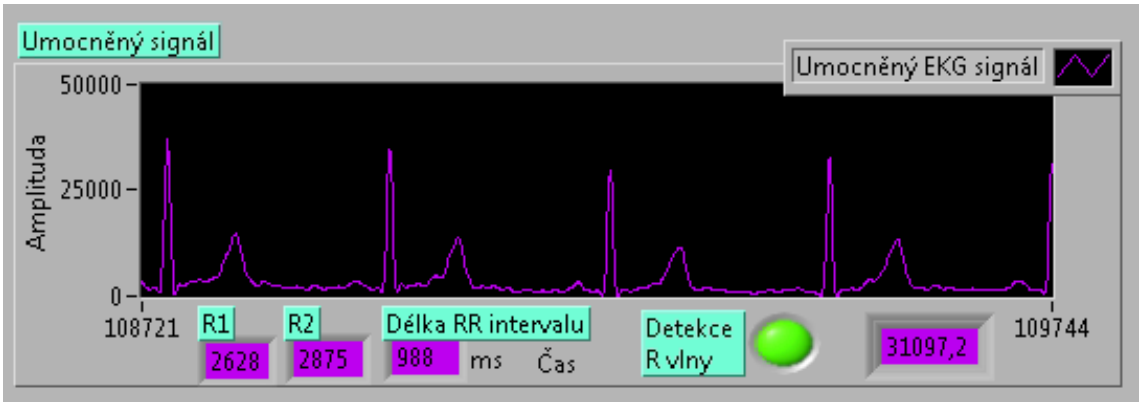
5.3.4 Detekce R vlny pomocí umocnění a prahu

Další filtry, které jsou v aplikaci použity, slouží k vymezení frekvenčního pásma, ve kterém se vyskytuje R vlna. Přibližně se jedná o pásmo mezi 10 až 20 Hz. K tomuto účelu je použita pásmová propust. Následné umocnění signálu se provádí za účelem dosažení kladných hodnot signálu a zvýraznění R vln. Na závěr je signál vyhlazen filtrem typu dolní propust.



Obr. 5.10: Umocnění EKG signálu a detekce R vlny pomocí prahu (blokový diagram)

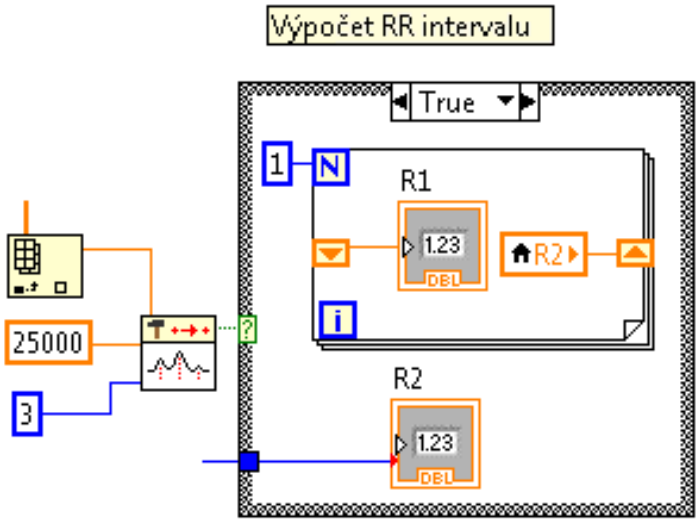
Takto upravený signál je zobrazen na čelním panelu. Detekce R vlny je v programu zajištěna pomocí překročení nastaveného prahu (Obr. 5.10). Výskyt R vlny je signalizován LED diodou (Obr. 5.11).



Obr. 5.11: Umocněný EKG signál po filtraci a detekce R vlny

5.3.5 Výpočet R-R intervalu a tepové frekvence

Pro určení polohy peaku R vlny slouží blok *Peak Detector PtByPt*, který v souboru vstupních dat hledá lokální minima - *Valley* a lokální maxima - *Peak*. Významným parametrem funkce je šířka, neboli velikost zájmové oblasti - *Width*. Výchozí hodnota tohoto parametru je 3. Je-li šířka liché číslo, funkce může najít vrchol nebo údolí pouze na pozici $(šířka + 1)/2$. Pokud je šířka sudé číslo, tak na pozici $šířka/2$. Výstupem bloku je pravdivostní hodnota *True* nebo *False*, která dále řídí strukturu *Case* (Obr. 5.12). Detekuje-li funkce první hodnotu nacházející se na nad prahem, odešle na svůj výstup hodnotu *True*, jinak výstupem zůstává hodnota *False*.



Obr. 5.12: Pozice R vlny

Do struktury je zaveden číselný indikátor označený jako R2, který je připojený

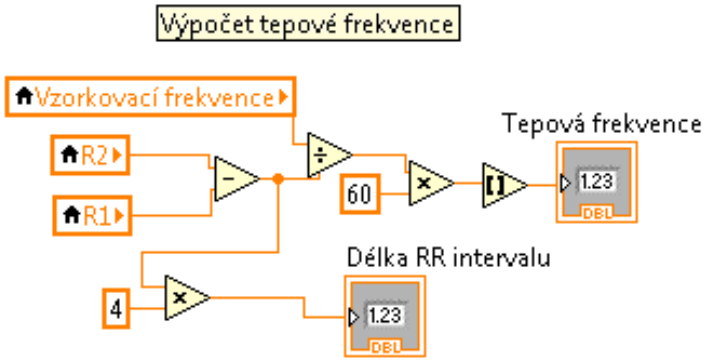
k iteračnímu terminálu cyklu *While*. Indikátor R2 zobrazuje každou hodnotu, kdy signál překročí nastavený práh. Z R2 je vytvořena lokální proměnná. Tato lokální proměnná je umístěná do struktury *For Loop* a připojena na posuvný registr *Shift Register*. Počítací terminál *For Loop* je nastaven na jedničku, aby při každé hodnotě *True* provedl jeden cyklus. Lokální proměnná díky posuvnému registru přenáší svoji hodnotu z jednoho kroku smyčky do dalšího. Pro zobrazení přenesené hodnoty slouží číselný indikátor R1.

Z polohy dvou po sobě následujících R vln je možné vypočítat délku RR intervalu a také tepovou frekvenci (TF) podle vzorce:

$$TF = 60 \times \frac{f_{vz}}{R_2 - R_1}, \tag{5.1}$$

kde R_1 a R_2 udává pozici R vlny ve vzorcích.

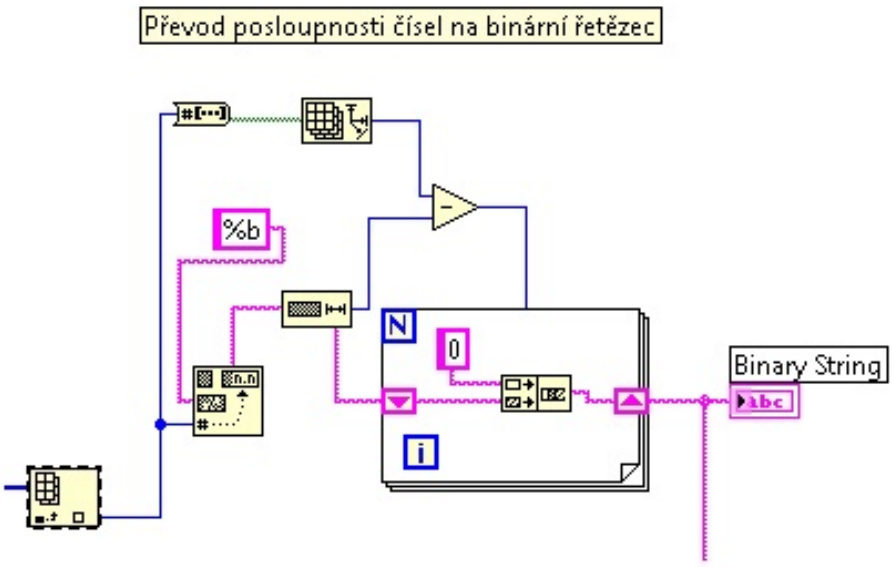
Realizace výpočtu délky RR intervalu a tepové frekvence je na Obr. 5.13.



Obr. 5.13: Výpočet tepové frekvence

5.3.6 Převod posloupnosti čísel na binární řetězec

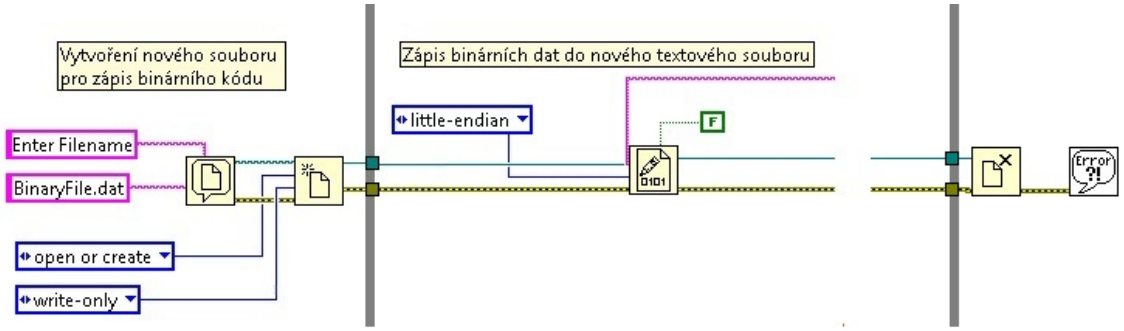
Před odesíláním do sítě Ethernet jsou jednotlivé hodnoty převedeny na binární řetězce. Převod je realizován pomocí funkcí *Index Array Function*, *Format Value Function*, *Array Size Function* a struktury *For Loop* (Obr. 5.14).



Obr. 5.14: Převod posloupnosti čísel na binární řetězec

5.3.7 Uložení binárních dat do textového souboru

Vytvořená aplikace umožňuje binární data před odesláním uložit do textového souboru. K zápisu binárních dat do nového souboru slouží funkce *Write to Binary File Function* (Obr. 5.15). Pro přehlednost výsledného textového souboru byla použita funkce *Concatenate Strings Function*, která sloučí dva vstupní řetězce do jednoho. Jedním ze vstupních textových řetězců funkce je binární řetězec a druhým předdefinovaná "konstanta" *End Of Line Constant*. Tato jednoduchá operace řadí jednotlivé hodnoty kódované binárně pod sebe.



Obr. 5.15: Uložení binárního řetězce do textového souboru

5.3.8 Komprese signálu

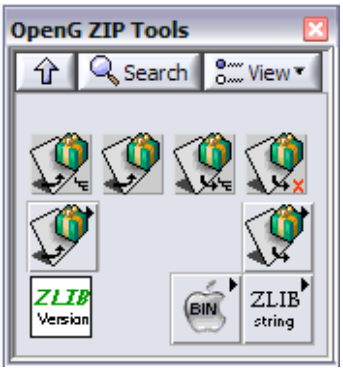
Komprese signálu není nutnou součástí programu, proto je do aplikace zabudovaná jako blok, který uživatel může vypnout. Aplikace je vytvořena tak, aby odesílala signál po jednotlivých vzorcích. Tento způsob přenosu signálu je rychlejší a spolehlivější. Každý vzorek je tedy při odesílání do sítě zabalen do samostatného datagramu. Vzhledem k tomu, že samotná hlavička má větší velikost než přenášený vzorek, není nutné tato data komprimovat.

Při realizaci programu se nabízela možnost EKG signál rozdělit do určitých úseků a ty pak postupně odesílat. V tomto případě by se pravděpodobně využilo detekovaných R vln a vypočtených RR intervalů. Zde by byla komprimace efektivní. Nevýhodou této metody je jednak větší časová náročnost, ale také případné ztráty datagramů při jejich přenosu. Chybějící datagram s větším úsekem EKG způsobí v přijatém a vykresleném signálu velký skok.

V teoretickém úvodu práce byly představeny dva kompresní algoritmy, které by bylo možné využít ke kompresi úseků EKG pro přenos po síti. Kompresní algoritmus SPIHT založený na vlnkové transformaci a kompresní algoritmus založený na diskrétní kosinové transformaci. Oba algoritmy jsou však ztrátové. Pro klinickou praxi je obvykle požadavána bezeztrátová komprese dat.

Řešením je ponechat stávající odesílání EKG signálu po vzorcích a použít ke kompresi bezeztrátový slovníkový algoritmus.

Pro účely komprese byla základní verze LabVIEW rozšířena o knihovnu *OpenG ZIP Tools* (Obr. 5.16). Jedná se o balíček kompresních funkcí pracujících na základě slovníkových algoritmů.



Obr. 5.16: Dialogové okno knihovny OpenG ZIP Tools

Knihovna *OpenG ZIP Tools* nabízí následující funkce:

ZLIB Compress Dictionary

ZLIB Compress Files
ZLIB Extract All Files To Dir
ZLIB Delete Files From Archive
Advanced Compression
Advanced Decompression
ZLIB Get Version
ZLIB String Compression
- *ZLIB Inflate*
- *ZLIB Deflate*

MacBinary

Algoritmy této skupiny vytvářejí v průběhu komprimace slovník na základě dat již zkomprimovaných, v němž se pak snaží najít data, která se teprve mají komprimovat. Pokud jsou data nalezena ve slovníku, algoritmus zapíše pozici dat ve slovníku místo samotných dat. Slovníkové algoritmy se řadí mezi bezeztrátové kompresní metody.

V programu je využit kompresní blok *ZLIB Deflate* a dekompresní blok *ZLIB Inflate* (Obr. 5.17). Funkce *ZLIB Deflate* provádí kompresi procházejícího proudu dat, aniž by se jakkoliv musela ukládat do souboru. Blok nabízí uživateli devět úrovní komprese. Nejnižší stupeň je 0 (bez komprese), nejvyšší stupeň 9. Defaultní nastavení -1 představuje doporučenou úroveň 6. Výstupem bloku je proud komprimovaných dat.

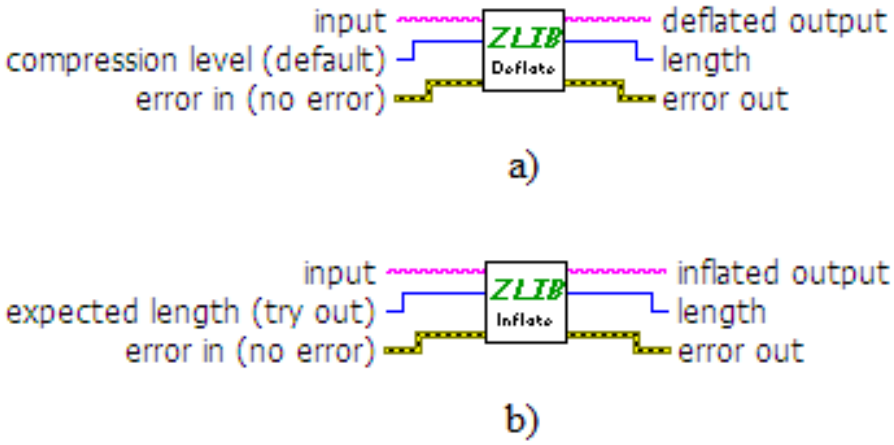
Protikladem *ZLIB Deflate* v programu pro příjem signálu je funkce *ZLIB Inflate*. Do bloku vstupuje řetězec komprimovaných dat a výstupem jsou rekonstruovaná data.

5.3.9 Přenos EKG signálu po síti Ethernet

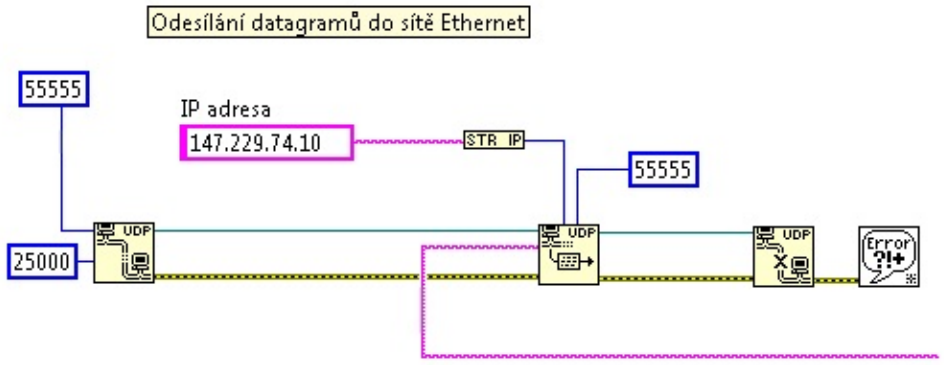
Odesílání datagramů

Pro přenos EKG signálu po síti Ethernet byl zvolen transportní protokol UDP. Odesílání binárních dat je realizováno třemi bloky z palety *Protocols* → *UDP* (Obr. 5.18). Funkce *UDP Open Function* otevírá UDP socket (zásuvku) na portu či názvu služby. Funkce *UDP Write Function* zapisuje do vzdálené UDP zásuvky. Funkce *UDP Close Function* zavírá UDP socket.

Funkce *UDP Open Function* vyžaduje nastavení lokálního portu *Port*, s nímž je požadováno vytvoření soketu (zásuvky). Číslo portu se zadává manuálně na čelním panel a je přednastaveno na 55555.



Obr. 5.17: a) kompresní blok *ZLIB Deflate*, b) rekonstrukční blok *ZLIB Inflate*

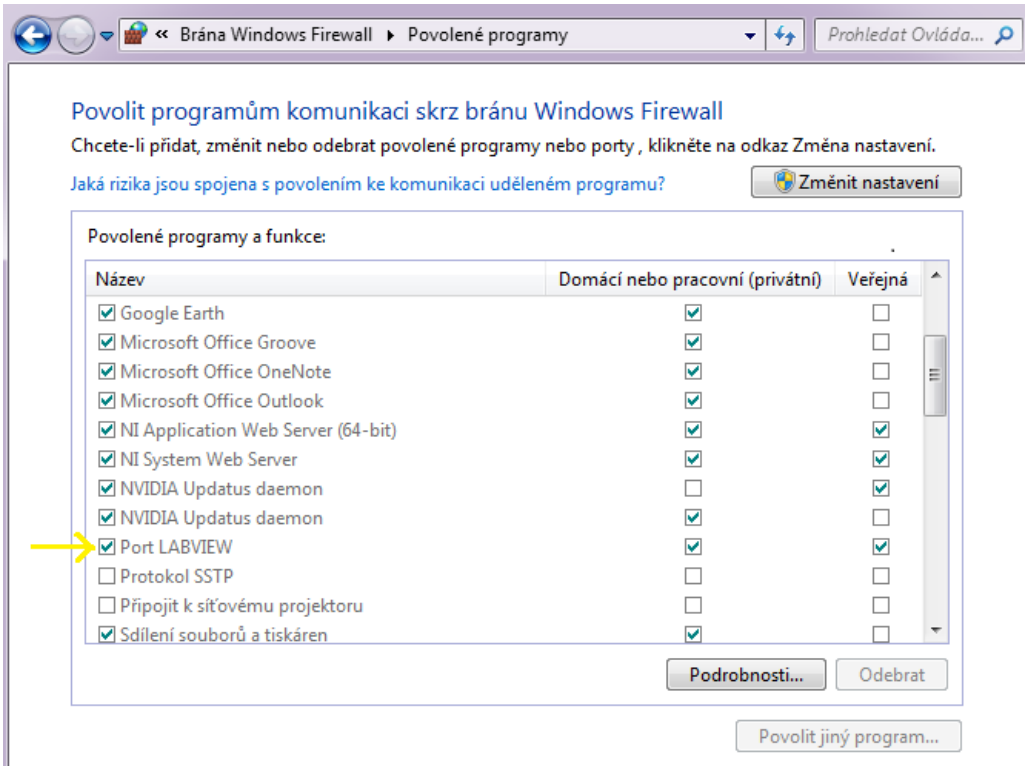


Obr. 5.18: Odesílání datagramů do sítě Ethernet na uvedenou IP adresu

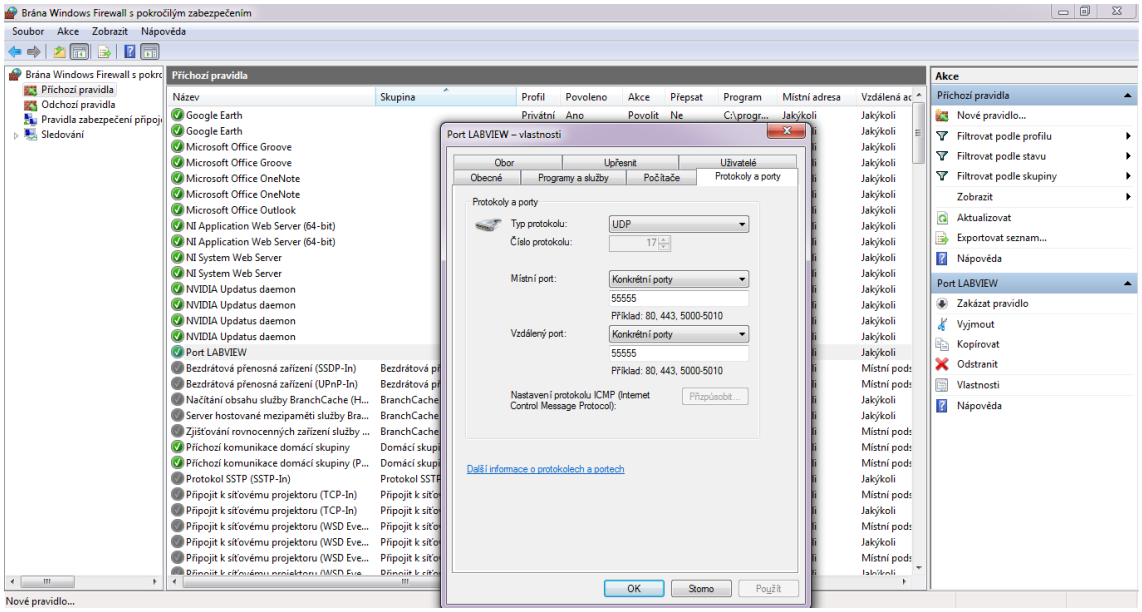
Pro správnou funkci programu je nutné provést následující nastavení počítače:

1. V nastavení *Brány Windows Firewall* umožnit programu NI LabView komunikaci na po6adovan0m portu (*Start* → *Ovládací panely* → *Systém a zabezpečení* → *Brána Windows Firewall*) (Obr. 5.19).
2. V záložce *Upřesnit nastavení* vytvořit nové příchozí a odchozí pravidlo pro port, který chceme pro přenos dat používat. V našem případě tedy nastavujeme pravidlo pro UDP port s číslem 55555 (Obr. 5.20).

Dále je nutné u funkce *UDP Open Function* nastavit *Timeout ms* - dobu, po kterou funkce čeká na dokončení. Pro *Timeout ms* je ponecháno standartní nastavení 25 000 ms. Při překročení této doby program vypíše chybovou hlášku. Pokud bychom



Obr. 5.19: Povolení komunikace na daném portu ve Firewallu (MS Windows)

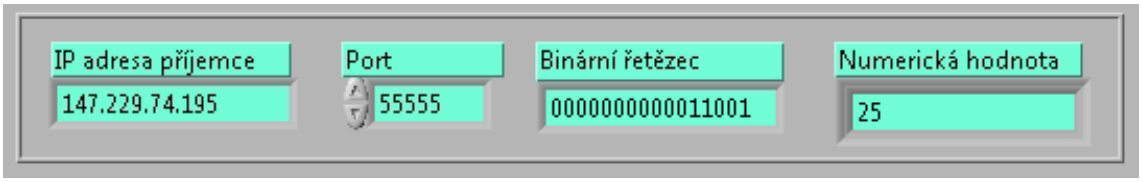


Obr. 5.20: Vytvoření nového pravidla

požadovali nastavení nekonečné doby čekání, zadáme zde hodnotu -1.

EKG záznam je před odesláním do sítě Ethernet rozdělen do datagramů. Datagramy postupně vstupují jako 12 bitové textové řetězce do funkce *UDP Write*

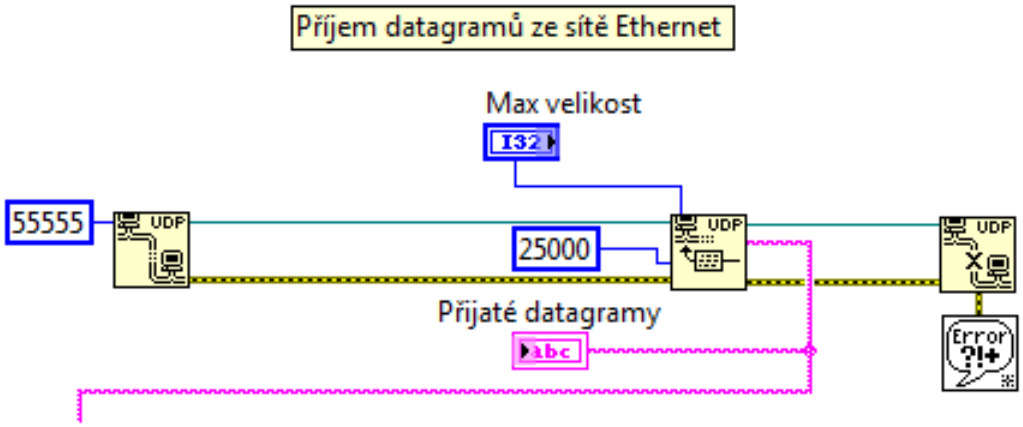
Function. U tohoto bloku je opět přednastaveno číslo portu pro zápis a také adresa počítače, na který požadujeme datagram poslat. IP adresa adresáta se nastavuje manuálně na čelním panelu jako textový řetězec (Obr. 5.21). V blokovém diagramu je pro převedení textového řetězce na IP adresu sítě použita funkce *String To IP Function*.



Obr. 5.21: Nastavení IP adresy a čísla portu

Příjem datagramů

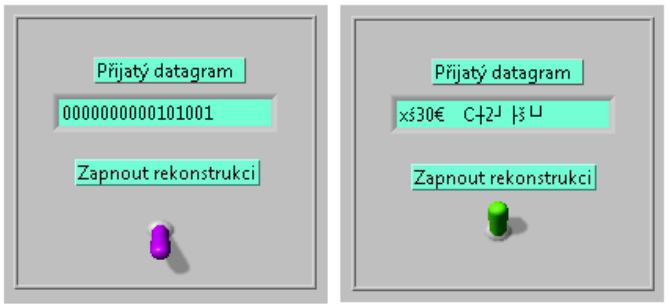
Příjem dat ze sítě Ethernet je v programu stejně jako jejich odesílání realizováno třemi bloky: funkcemi *UDP Open Function*, *UDP Read Function* a *UDP Close Function* (Obr. 5.22).



Obr. 5.22: Příjem datagramů ze sítě Ethernet

Funkce pro otevření a zavření UDP socketu mají stejné nastavení jako v programu pro odesílání datagramů. Funkce *UDP Read Function* načítá datagramy z UDP socketu a odesílá výsledky na výstup. Blok *UDP Read Function* vyžaduje nastavení dvou parametrů: maximální velikosti přijatého datagramu (*Max Size*) a doby čekání na příjem datagramu (*Timeout ms*).

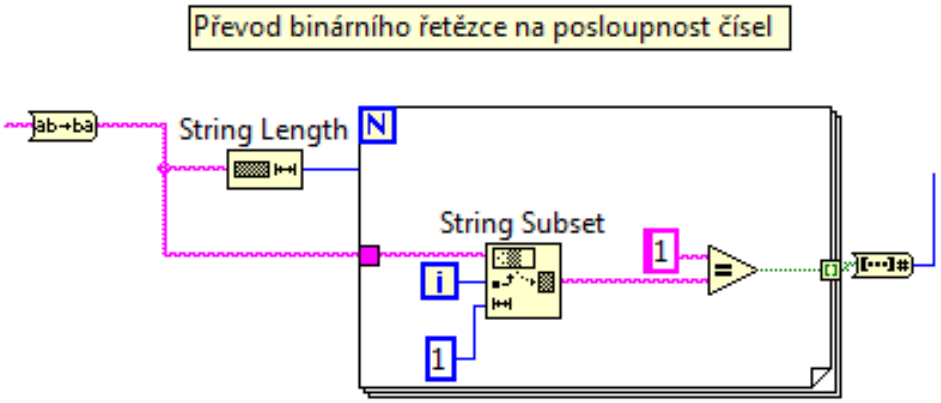
Jestli data přicházejí komprimovaná, pozná uživatel nejen podle grafického výstupu, ale také podle hodnoty, která se zobrazí na čelním panelu v indikátoru nad tlačítkem pro zapnutí rekonstrukce EKG signálu. Nekomprimovaná data se zobrazují jako řetězce 0 a 1, zatímco aktivní komprese a neaktivní dekomprese je patrná přenosem libovolných ASCII znaků (Obr. 5.23). V závislosti na tom pak uživatel spustí dekompresní algoritmus.



Obr. 5.23: Indikace příjmu původních dat (vlevo) a komprimovaných dat (vpravo)

5.3.10 Převod binárního řetězce na posloupnost čísel

Program pro příjem datagramů ze sítě Ethernet přijímá data buď ve formě binárního řetězce a nebo data komprimovaná, která se následně rekonstruují. V obou případech jsou před zobrazením EKG signálu binární řetězce převedeny zpět na posloupnost čísel. Schéma převodu je na Obr. 5.24.



Obr. 5.24: Převod binárního řetězce na posloupnost čísel

5.3.11 Zajištění běhu programů

Všechny výše uvedené části obou programů jsou uzavřeny ve smyčkám *While Loop* (Obr. 5.25). Tyto smyčky zajišťují kontinuální opakování algoritmu po dobu platnosti vložené ukončovací podmínky. Podmínkou pro ukončení běhu programu je změna pravdivostní hodnoty tlačítka *Stop*. Cyklus tedy může být ukončen po stisknutí tlačítka *Stop* na čelním panelu.

Obr. 5.25: Smyčka *While Loop* a ukončení programu

5.3.12 Nápověda

Všechny ikony na čelních panelech jsou opatřeny popiskem, který se zobrazí při spuštění programu a současném zapnutí kontextové nápovědy.

6 DATOVÝ PŘENOS

Program ping (Packet InterNet Groper) umožňuje prověřit funkčnost spojení mezi dvěma síťovými rozhraními v takové počítačové síti, která používá sadu protokolů TCP/IP. Ping při své činnosti periodicky odesílá IP datagramy a očekává odezvu protistrany. Při úspěšném obdržení odpovědi změří délku zpoždění (latenci) a na závěr vypíše statistický souhrn přenosu.

Aplikace byla odzkoušena mezi dvěma počítači s IP adresami 147.229.74.10 (odesílatel) a 147.229.74.195 (příjemce). Výstup prověření spojení mezi těmito počítači je na Obr. 6.1

```
Microsoft Windows [Verze 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Všechna práva vyhrazena.

C:\Users\bohatcova>ping 147.229.74.195

Příkaz PING na 147.229.74.195 - 32 bajtů dat:
Odpověď od 147.229.74.195: bajty=32 čas < 1ms TTL=128
Odpověď od 147.229.74.195: bajty=32 čas < 1ms TTL=128
Odpověď od 147.229.74.195: bajty=32 čas < 1ms TTL=128
Odpověď od 147.229.74.195: bajty=32 čas < 1ms TTL=128

Statistika ping pro 147.229.74.195:
Pakety: Odeslané = 4, Přijaté = 4, Ztracené = 0 (ztráta 0%),
Přibližná doba do přijetí odezvy v milisekundách:
    Minimum = 0ms, Maximum = 0ms, Průměr = 0ms
```

Obr. 6.1: Výstup prověření spojení mezi dvěma počítači

Všechny čtyři odeslané pakety byly přijaty beze ztráty a přibližná doba do přijetí odezvy byla menší než 1 ms.

Program traceroute slouží k analýze cesty (trasy) poslaného datagramu k cílovému počítači. Vypisuje uzly (respektive směrovače) na cestě datagramů od zdroje až k zadanému cíli (Obr. 6.2). Uzly jsou zjišťovány pomocí snížení hodnoty TTL v hlavičce datagramů.

```
C:\Users\bohatcova>tracert 147.229.74.195

Výpis trasy k PC-E330-371.ubmi.feec.vutbr.cz [147.229.74.195]
s nejvýše 30 směrováními:

  1    < 1 ms    < 1 ms    < 1 ms  PC-E330-371.ubmi.feec.vutbr.cz [147.229.74.195]

Trasování bylo dokončeno.
```

Obr. 6.2: Výpis uzlů mezi počítači

Traceroute zvyšuje hodnotu "time to live"(TTL) po každém úspěšném odeslání balíčku paketů. První tři pakety mají jednotnou hodnotu TTL nastavenou na 1 (odesílají se současně), další tři pakety mají hodnotu TTL 2 atd. Při cestě k cíli paket

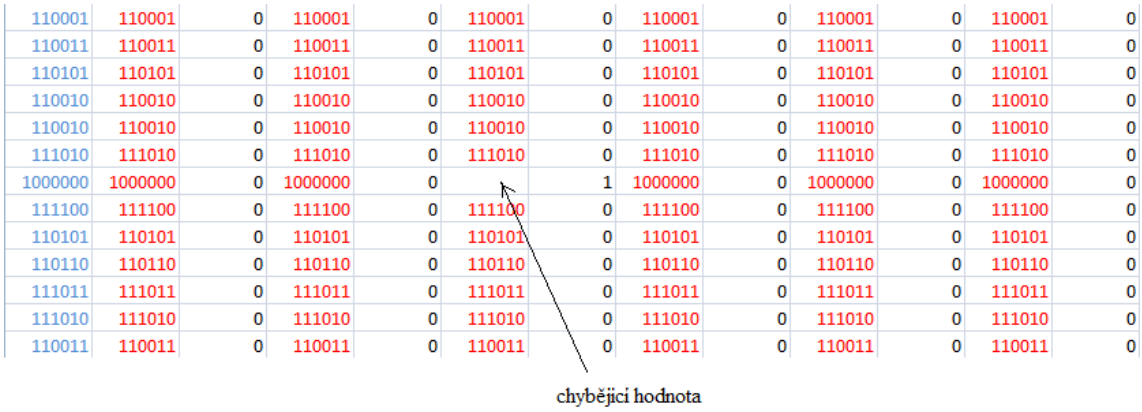
prochází jednotlivými směrovači (uzly). Při průchodu směrovač sníží hodnotu TTL o 1 a pošle ho dál. Je-li hodnota TTL paketu nula a není v cílové IP síti, pak je paket zahozen a směrovač pošle chybovou ICMP zprávu odesílateli. Traceroute využívá právě těchto chybových hlášení, aby sestavil tabulku cesty paketu od odesílatele k cíli. Ve výpisu jsou tak zobrazeny všechny uzly, které položku TTL snižují.

Datagramy odeslané z počítače s IP 147.229.74.10 na počítač s IP 147.229.74.195 putují do cíle pouze přes jeden uzel.

7 SHODNOCENÍ VÝSLEDKŮ

Součástí aplikace je zápis odesílaných a následně přijímaných dat do textového souboru v podobě binárních řetězců. Tyto výstupy jsou užitečné k závěrečnému vyhodnocení kvality přenosu EKG signálu.

Pro vyhodnocení výsledků byl mezi počítači odesílatele a příjemce přenos stejného EKG signálu opakován 20krát. V každém opakování bylo odesláno prvních 10 000 vzorků EKG signálu. Všechny odeslané a přijaté hodnoty byly zapsány do tabulky. Ukázka tabulky je na (Obr. 7.1).



110001	110001	0	110001	0	110001	0	110001	0	110001	0	110001	0
110011	110011	0	110011	0	110011	0	110011	0	110011	0	110011	0
110101	110101	0	110101	0	110101	0	110101	0	110101	0	110101	0
110010	110010	0	110010	0	110010	0	110010	0	110010	0	110010	0
110010	110010	0	110010	0	110010	0	110010	0	110010	0	110010	0
111010	111010	0	111010	0	111010	0	111010	0	111010	0	111010	0
1000000	1000000	0	1000000	0		1	1000000	0	1000000	0	1000000	0
111100	111100	0	111100	0	111100	0	111100	0	111100	0	111100	0
110101	110101	0	110101	0	110101	0	110101	0	110101	0	110101	0
110110	110110	0	110110	0	110110	0	110110	0	110110	0	110110	0
111011	111011	0	111011	0	111011	0	111011	0	111011	0	111011	0
111010	111010	0	111010	0	111010	0	111010	0	111010	0	111010	0
110011	110011	0	110011	0	110011	0	110011	0	110011	0	110011	0

Obr. 7.1: Ukázka tabulky odeslaných a přijatých hodnot s vyznačenou chybou

Při přenosu originálních dat bez použití komprese se během dvaceti přenosů neztratil žádný datagram. Použitá metoda komprese EKG signálů při doporučené úrovni 6 průměrně způsobuje ztrátu 23.75 vzorků z 10 000, což představuje ztrátu 0.24 % dat. Těchto výsledků bylo dosaženo na stejné síti, kde vzhledem ke krátké trase mezi odesílatelem a příjemcem (jeden aktivní prvek) chyby představuje rekostrukční algoritmus.

ZÁVĚR

Cílem teoretické části diplomové práce bylo provést literární rešerši v oblasti metod komprese EKG signálů. Seznámit se s podstatou komunikace v síti Ethernet, zejména se strukturou TCP a UDP paketů. Navrhnout metodu pro kompresi, odesílání, příjem a rekonstrukci EKG signálu.

V diplomové práci byly blíže představeny dvě metody komprese EKG signálů. Tyto metody jsou vhodné pro zpracování EKG signálu za účelem následného přenosu EKG záznamů komunikačními kanály. První metoda komprimující EKG signál v reálném čase využívá diskrétní kosinovou transformaci a Hufmanovo kódování. Druhá metoda je založena na vlnkové transformaci a algoritmu SPIHT. Obě dvě kompresní metody jsou však ztrátové.

V praktické části diplomové práce byly vytvořeny dvě samostatné aplikace. První aplikace v počítači odesílatele otevírá textový soubor s EKG signálem. Načtený EKG signál pak filtruje kaskádou filtrů za účelem odstranění rušení. Výsledný signál je zobrazen. Součástí aplikace je detekce R vlny, výpočet délky RR intervalu a tepové frekvence. Aplikace dále umožňuje EKG signál komprimovat. EKG signál je do sítě Ethernet odesílán prostřednictvím protokolu UDP po jednotlivých vzorcích. Ovládání aplikace prostřednictvím čelního panelu je intuitivní. Posuvnými ovladači se nastavují mezní frekvence filtrů a práh pro detekci R vlny, číselnými ovladači délka impulsní charakteristiky filtru a vzorkovací frekvence. Manuálně se také zadává IP adresa příjemce a číslo portu.

Aplikace v počítači příjemce přijímá vzorky signálu ze sítě, je-li přijatý signál komprimovaný, tak jej rekonstruuje. Výsledný EKG signál je zobrazen a jsou v něm opět detekovány R vlny, vypočtená délka RR intervalů a vzorkovací frekvence. Čelní panel aplikace příjemce má podobný vzhled jako čelní panel aplikace odesílatele.

Pro návrh aplikace byla zpočátku vybrána první z metod představených v teoretickém úvodu, metoda založená na diskrétní kosinové transformaci. Kompresní algoritmus se zde skládá z pěti kroků, během kterých je signál transformován a poté komprimován. Nejprve je EKG signál podvzorkován s poloviční vzorkovací frekvencí. Druhým krokem algoritmu je detekce píků v podvzorkovaném signálu. V dalším kroku jsou uložená data transformována použitím diskrétní kosinové transformace. Transformovaná data jsou filtrována oknem a následně je aplikován Huffmanův kódovací algoritmus. Výstupem algoritmu jsou komprimovaná data.

Postup rekonstrukce z komprimovaných dat se sestává ze čtyř kroků. Jednotlivé kroky rekonstrukčního algoritmu jsou inverzní ke krokům algoritmu komprimačního a jsou prováděny v opačném pořadí. Na transformovaná a komprimovaná data je aplikován inverzní Huffmanův kódovací algoritmus. Druhým krokem je inverzní ko-

sinová transformace. Ve třetím kroku je obnovený signál interpolován v čase pomocí spline interpolace. Posledním krokem rekonstrukčního algoritmu je výpočet původního signálu pomocí inverzního rozdílu.

Převážnou část kompresního programu se podle návrhu podařilo realizovat, ale nebylo dosaženo uspokojivých výsledků. Klinická praxe vyžaduje při zpracování signálů použití bezeztrátových kompresních algoritmů, což tato metoda nesplňuje. Řešením bylo použít pro kompresi EKG signálu metodu ze skupiny slovníkových algoritmů. Algoritmy této skupiny vytvářejí v průběhu komprimace slovník na základě dat již zkomprimovaných, v němž se pak snaží najít data, která se teprve mají komprimovat. Pokud jsou data nalezena ve slovníku, algoritmus zapíše pozici dat ve slovníku místo samotných dat. Slovníkové algoritmy se řadí mezi bezeztrátové kompresní metody. K realizaci kompresní metody byl vybrán algoritmus, který provádí kompresi procházejícího proudu dat, aniž by se data jakkoliv musela ukládat do souboru. Aplikace nabízí uživateli devět úrovní komprese. Teprve použití slovníkového algoritmu přineslo očekávané výsledky.

Aplikace byla testována na datovém přenosu mezi dvěma počítači připojenými v lokální síti i průchodem přes několik sítí s delší odezvou. Originální EKG data bez použití komprese byla přenesena bez ztráty datagramů a téměř bez zpoždění. Použití komprese při doporučeném nastavení úrovně komprese způsobuje průměrnou ztrátu 0,24 % signálu. S rostoucím počtem uzlů v síti dochází k mírnému nárůstu počtu ztracených datagramů (až na 0,5 %), což mohou kromě chyb rekonstrukčního algoritmu způsobovat i ztracené datagramy.

Vzhledem k sestavenému programu pro odesílání a příjem EKG záznamů, aplikovanému kompresnímu i rekonstručnímu algoritmu a vyhodnocení provedeného přenosu se domnívám, že byly cíle práce splněny.

LITERATURA

[1] Abo-Zahhad, M. and B. A. Rajoub *An effective coding technique for the compression of one-dimensional signals using wavelet transforms.* Medical Engineering Physics, Vol. 24, No. 3, pp. 185-199, 2002. Times Cited: 14. Impact factor: 1.909.

[2] Alesanco, A. and J. Garcia *A simple method for guaranteeing ECG quality in real-time wavelet lossy coding.* Eurasip Journal on Advances in Signal Processing, 2007. Times Cited: 1. Impact factor: 1.053

[3] Alesanco, A., S. Olmos, et al. *A novel real-time multilead ECG compression and de-noising method based on the wavelet transform.* Computers in Cardiology 2003, Vol 30. A. Murray. 30, pp. 593-596, 2003. Times Cited: 2.

[4] Alshamali, A. and A. Al-Smadi *Combined coding and wavelet transform for ECG compression.* Journal of Medical Engineering Technology, Vol. 25, No. 5, pp. 212-216, 2001. Times Cited: 4.

[5] Arnavut, Z. *ECG signal compression based on Burrows-Wheeler transformation and inversion ranks of linear prediction.* IEEE Transactions on Biomedical Engineering, Vol. 54, No. 3, pp. 410-418, 2007. Times Cited: 5. Impact factor: 1.790.

[6] Arnavut, Z., Ieee, et al. *Lossless and near-lossless compression of ECG signals.* Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Vols 1-4: Building New Bridges at the Frontiers of Engineering and Medicine. 23, pp. 2146-2149, 2001. Times Cited: 1.

[7] Batista, L. V., E. U. K. Melcher, et al. *Compression of ECG signals by optimized quantization of discrete cosine transform coefficients.* Medical Engineering Physics, Vol. 23, No. 2, pp. 127-134, 2001. Times Cited: 21. Impact factor: 1.909.

[8] Bilgin, A., M. W. Marcellin, et al. *Compression of electrocardiogram signals using JPEG2000.* Ieee Transactions on Consumer Electronics, Vol. 49, No. 4, pp. 833-840, 2003. Times Cited: 30. Impact factor: 1.057.

[9] Brechet, L., M. F. Lucas, et al. *Compression of biomedical signals with mother wavelet optimization and best-basis wavelet packet selection.* Ieee Transactions on Biomedical Engineering, Vol. 54, No. 12, pp. 2186-2192, 2007. Times Cited: 19. Impact factor: 1.790.

[10] Dostálek, L., Kabelová, A. *Velký průvodce protokoly TCP/IP a systémem DNS* Praha: Computer Press, 2000. 2. vydání, 426 s. ISBN 80-7226-323-4.

[11] Filho, E. B. L., N. M. M. Rodrigues, et al. *ECG signal compression based on dc equalization and complexity sorting.* Ieee Transactions on Biomedical Engineering, Vol. 55, No. 7, pp. 1923-1926, 2008. Times Cited: 6. Impact factor: 1.790.

[12] Fira, C. M. and L. Goras *An ECG signals compression method and its validation using NNs.* IEEE Transactions on Biomedical Engineering, Vol. 55, No. 4, pp. 1319-1326, 2008. Times Cited: 6. Impact factor: 1.790.

[13] Giurcaneanu, C. D., I. Tabus, et al. *Using contexts and R-R interval estimation in lossless ECG compression.* Computer Methods and Programs in Biomedicine, Vol. 67, No. 3, pp. 177-186, 2002. Times Cited: 14. Impact factor: 1.238.

[14] Hampton, John. R. *EKG stručně, jasně, přehledně.* Praha: Grada, 2005. 2. vydání, 149 s. ISBN 80-247-0960-0.

[15] Hrubeš, J., Kozumplík, J. *Možnosti algoritmu SPIHT při kompresi 1D signálů.* Elektrorevue - Internetový časopis (http://www.elektrorevue.cz), 2007, roč. 2007, č. 55, s. 55-1 (12 s.), ISSN: 1213-1539.

[16] Hrubeš, J., Vítek, M., Kozumplík, J.. *Vliv komprese signálů EKG na diagnózu.* Elektrorevue - Internetový časopis (http://www.elektrorevue.cz), 2010, roč. 2010, č. 36, s. (4 s.) , ISSN: 1213-1539.

[17] Chou, H. H., Y. J. Chen, et al. *Effective and efficient compression algorithm for ECG signals with irregular periods.* IEEE Transactions on Biomedical Engineering, Vol. 53, No. 6, pp. 1198-1205, 2006. Times Cited: 22. Impact factor: 1.790.

[18] Istepanian, R. S. H. and A. A. Petrosian *Optimal zonal wavelet-based ECG data compression for a mobile telecardiology system.* IEEE Transactions on Information Technology in Biomedicine, Vol. 4, No. 3, pp. 200-211, 2000. Times Cited: 45.

[19] Iwata, A., Y. Nagasaka, et al. *Data-compression of the ECG using neural network for digital Holter monitor.* IEEE Engineering in Medicine and Biology Magazine, Vol. 9, No. 3, pp. 53-57, 1990. Times Cited: 12.

[20] Jan, J. *Číslicová filtrace, analýza a restaurace signálů.* Brno: Vutium Brno, ISBN: 80-214-081.

[21] Khan, M. Gabriel. *EKG a jeho hodnocení.* Praha: Grada, 2005. 2. vydání, 348 s. ISBN 80-247-0910-4.

[22] Kim, B. S., S. K. Yoo, et al. *Wavelet-based low-delay ECG compression algorithm for continuous ECG transmission.* IEEE Transactions on Information Technology in Biomedicine, Vol. 10, No. 1, pp. 77-83, 2006. Times Cited: 19.

[23] Kim, J., M. Kim, I. Won, S. Yang, K. Lee, and W. Huh *A biomedical signal segmentation algorithm for event detection based on slope tracing.* Proc. IEEE Conf. Eng. Med. Biol. Soc., pp. 1889–1892, 2009.

[24] Ku, C. T., H. S. Wang, et al. *A novel ECG data compression method based on nonrecursive discrete periodized wavelet transform.* IEEE Transactions on Biomedical Engineering, Vol. 53, No. 12, pp. 2577-2583, 2006. Times Cited: 14. Impact factor: 1.790.

[25] Kumar, V., S. C. Saxena, et al. *Direct data compression of ECG signal for telemedicine.* International Journal of Systems Science, Vol. 37, No. 1, pp. 45-63, 2006. Times Cited: 4.

[26] Lee, S., J. Kim, et al. *A Real-Time ECG Data Compression and Transmission Algorithm for an e-Health Device.* IEEE Transactions on Biomedical Engineering, Vol. 58, No. 9), pp. 2448-2455, 2011. Times Cited: 0. Impact factor: 1.790.

[27] Lee, S., Lee, M. *M. A real-time ECG data compression algorithm for a digital holter system.* Conf Proc IEEE Eng Med Biol Soc., 2008., 8 s. 4736-9.

[28] Lu, Z. T., D. Y. Kim, et al. *Wavelet compression of ECG signals by the set partitioning in hierarchical trees algorithm.* IEEE Transactions on Biomedical Engineering, Vol. 47, No. 7), pp. 849-856, 2000. Times Cited: 138. Impact factor: 1.790.

[29] Miaou, S. G. and C. L. Lin *A quality-on-demand algorithm for wavelet-based compression of electrocardiogram signals.* IEEE Transactions on Biomedical Engineering, Vol. 49, No. 3, pp. 233-239, 2002. Times Cited: 28. Impact factor: 1.790.

[30] Miaou, S. G. and H. L. Yen *Multichannel ECG compression using multichannel adaptive vector quantization.* IEEE Transactions on Biomedical Engineering, Vol. 48, No. 10, pp. 1203-1207, 2001. Times Cited: 5. Impact factor: 1.790.

[31] Miaou, S. G., H. L. Yen, et al. *Wavelet-based ECG compression using dynamic vector quantization with tree codevectors in single codebook.* IEEE Transactions on Biomedical Engineering, Vol. 49, No. 7, pp. 671-680, 2002. Times Cited: 36. Impact factor: 1.790.

[32] Nielsen, M., E. N. Kamavuako, et al. *Optimal wavelets for biomedical signal compression.* Medical Biological Engineering Computing, Vol. 44, No. 7, pp. 561-568, 2006. Times Cited: 11.

[33] Pužmanová, Z. *TCP/IP v kostce* České Budějovice: Kopp, 2009. 2. vydání, 619 s. ISBN 978-80-7232-388-3.

[34] Rajoub, B. A. *An efficient coding algorithm for the compression of ECG signals using the wavelet transform.* IEEE Transactions on Biomedical Engineering, Vol. 49, No. 4, pp. 355-362, 2002. Times Cited: 47. Impact factor: 1.790.

[35] Salomon, D. *Data Compression - The Complete Reference.* London: Springer, 2007. 1092 p. 4.ed. ISBN 1-84628-602-5.

[36] Tai, S. C., C. C. Sun, et al. *A 2-D ECG compression method based on wavelet transform and modified SPIHT.* IEEE Transactions on Biomedical Engineering, Vol. 52, No. 6, pp. 999-1008, 2005. Times Cited: 25. Impact factor: 1.790.

[37] Tchiotsop, D., D. Wolf, et al. *ECG data compression using Jacobi polynomials.* 2007 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Vols 1-16, pp. 1863-1867, 2007. Times Cited: 2.

[38] Vokurka, M., Hugo, J. *Velký lékařský slovník.* Praha: Maxdorf, 2006. 6. vydání, 1017 s. ISBN 80-7345-105-0.

[39] Wei, J.-J., C.-J. Chang, N.-K. Chou, and G.-J. Jan *ECG data compression using truncated singular value decomposition.* IEEE Trans. Inf. Technol. Biomed., Vol. 5, No. 4, pp. 290–299, 2001.

[40] Wilhelm, Z. *Stručný přehled fyziologie člověka pro bakalářské studijní programy.* Brno: Masarykova univerzita, 2005. 3. vydání, 115 s. ISBN 80-210-2837-8.

[41] Zigel, Y., A. Cohen, et al. *ECG signal compression using analysis by synthesis coding.* IEEE Transactions on Biomedical Engineering, Vol. 47, No. 10, pp. 1308-1316, 2000. Times Cited: 34. Impact factor: 1.790.

[42] Zigel, Y., A. Cohen, et al. *The weighted diagnostic distortion (WDD) measure for ECG signal compression.* IEEE Transactions on Biomedical Engineering, Vol. 47, No. 11, pp. 1422-1430, 2000. Times Cited: 51. Impact factor: 1.790.

[43] *Archive for the 'DIY ECG' Category* [online]. Poslední aktualizace 2009 [cit. 2011-12-08].
Dostupný z URL: <http://www.swharden.com/>.

[44] *Svět sítí* [online]. Poslední aktualizace 2003 [cit. 2011-12-08].
Dostupný z URL: <http://www.svetsiti.cz/>.

[45] *Výukový web EKG* [online]. Poslední aktualizace 2006 [cit. 2011-12-08].
Dostupný z URL: <http://www.ekg.kvalitne.cz/>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

AZTEC Amplitude zone time epoch coding

BWT Burrows-Wheelerova transformace

CORTES Encoding system

DC Distance coding

DCT Diskrétní Fourierova transformace

DWT Diskrétní vlnková transformace

EKG elektrokardiografie, elektrokardiogram

FFT Rychlá Fourierova transformace

IF Inverse frequency coding

IP Internet Protocol

KLT Karhunen-Loeve transformace

MTF Move-to-front transformace

RLE Run lenght encoding

TCP Transmission Control Protocol

TF Tepová frekvence

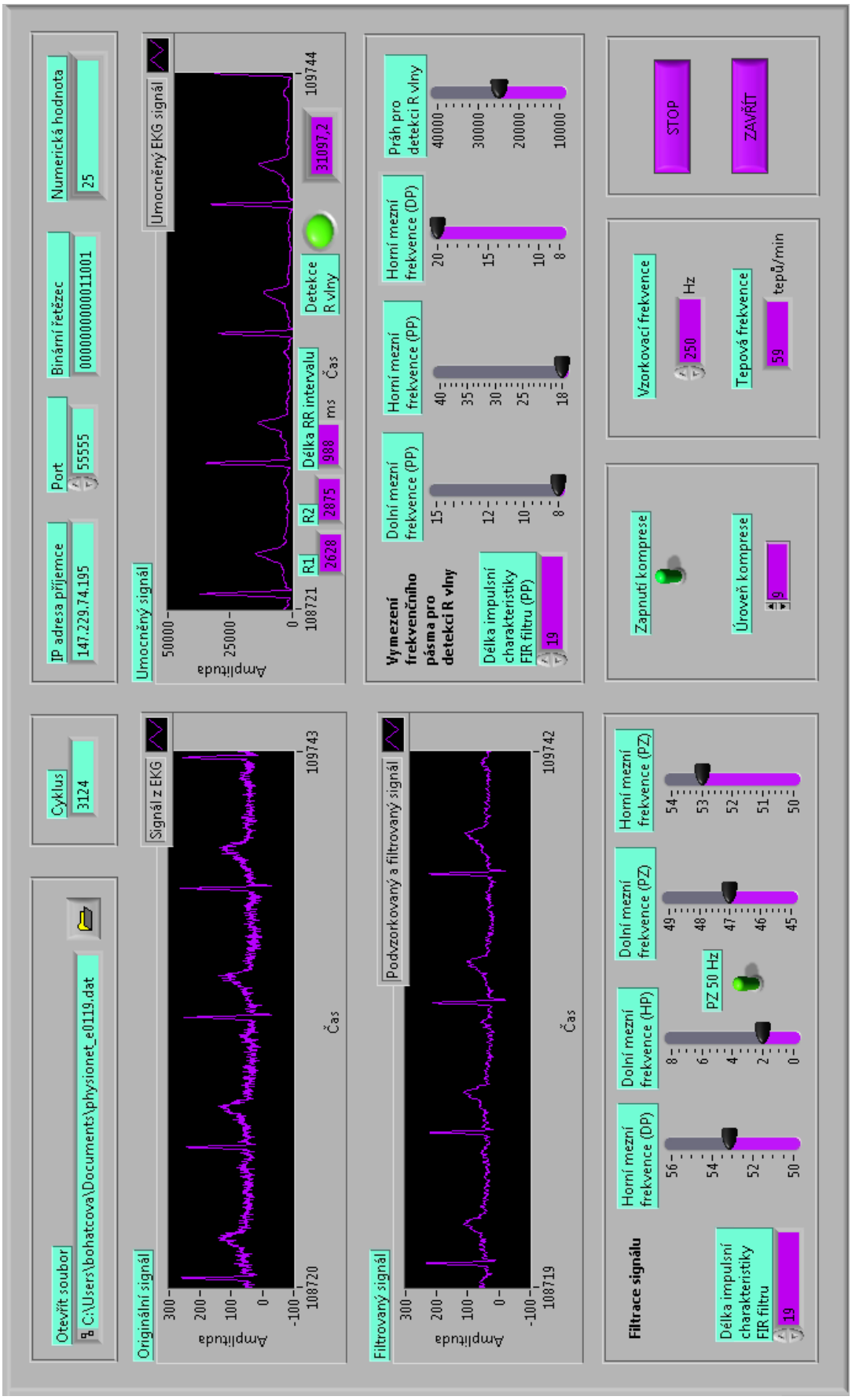
TP Turning point

TTL Time To Life

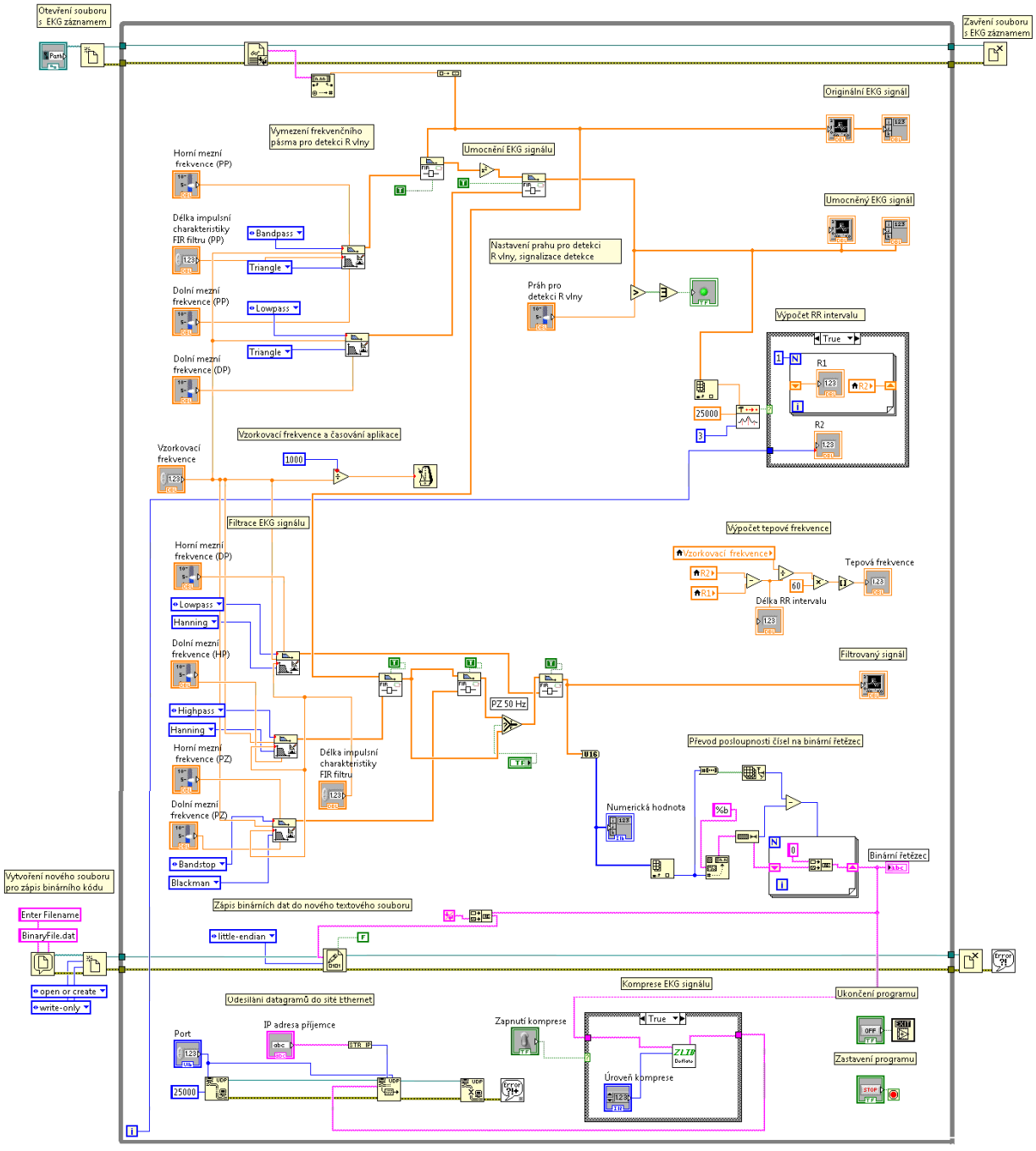
UDP User Datagram Protokol

WFC Weighted frequency count

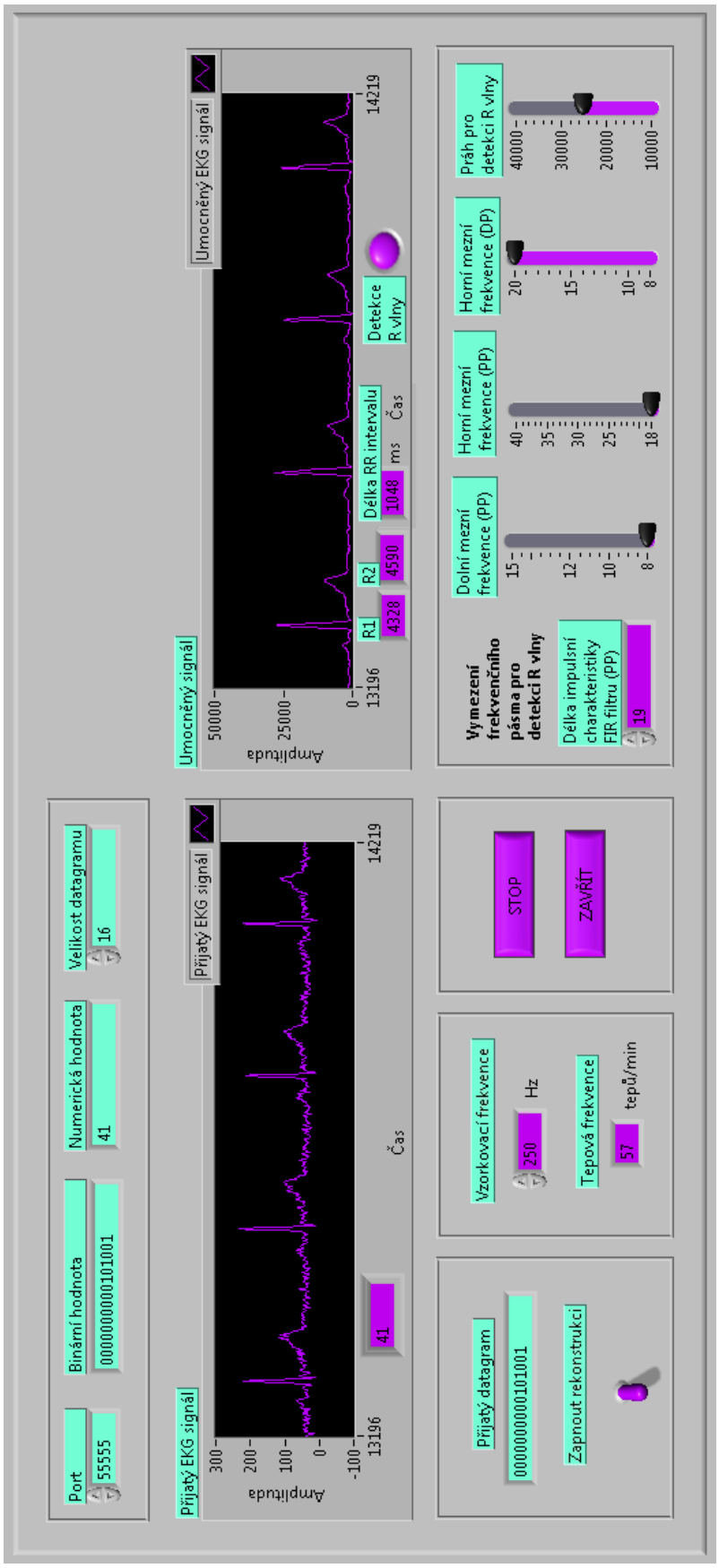
PŘÍLOHY



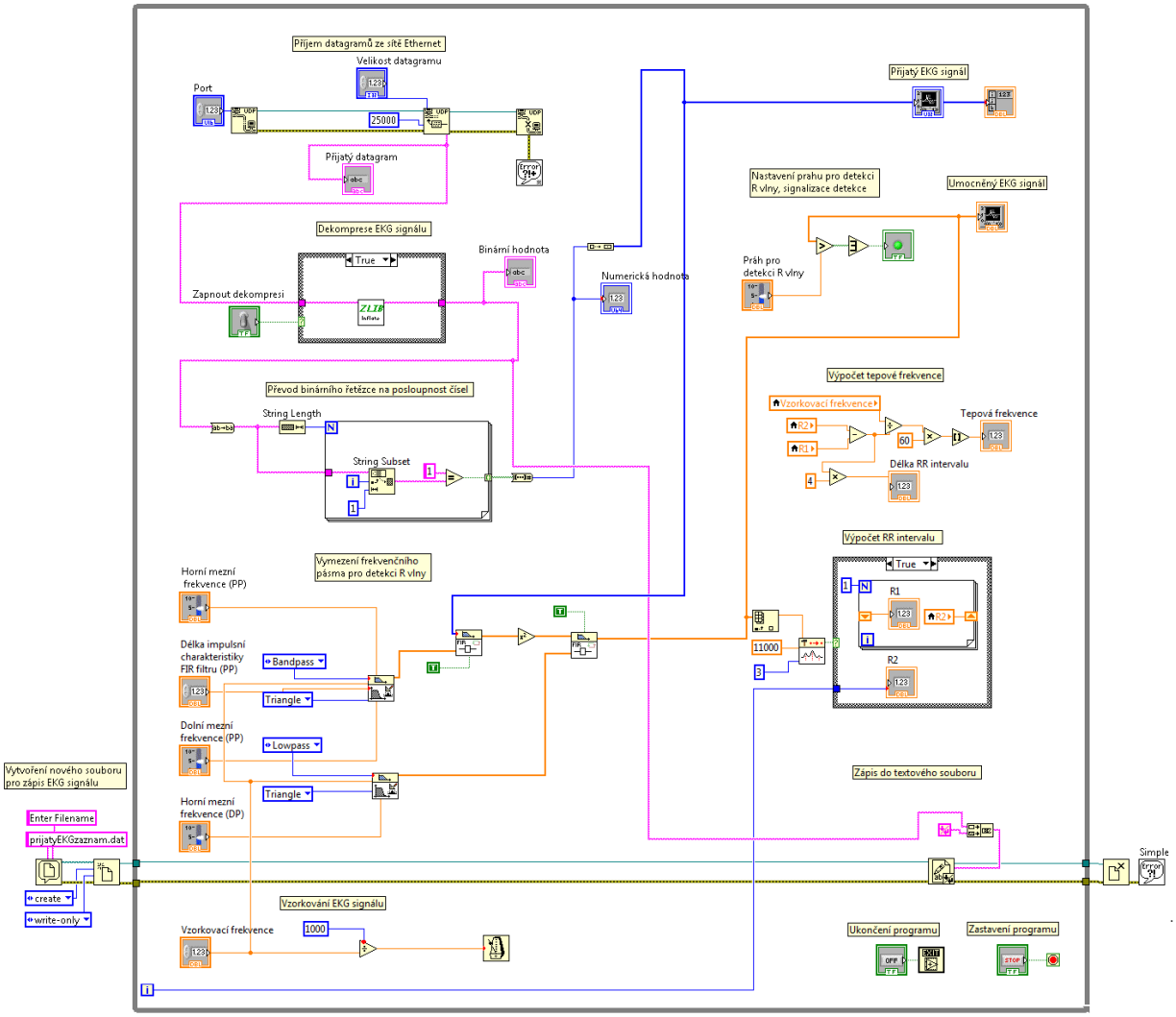
Obr. 2: Čelní panel aplikace pro kompresi a odesílání EKG signálu



Obr. 3: Blokový diagram aplikace pro kompresi a odesílání EKG signálu



Obr. 4: Čelní panel aplikace pro příjem a rekonstrukci EKG signálu



Obr. 5: Blokový diagram aplikace pro příjem a rekonstrukci EKG signálu