

ETHERNET driver pro RealTime systém

Diplomová práce

Martin Cvek



Květen 2015

Vedoucí práce:	Ing. Stanislav Flígl, Ph.D.
Studijní program:	Elektrotechnika a informatika (magisterský), strukturovaný
Obor:	Výpočetní technika

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra počítačů

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Martin Cvek**

Studijní program: Elektrotechnika a informatika (magisterský), strukturovaný
Obor: Výpočetní technika

Název tématu: **ETHERNET driver pro RealTime systém**

Pokyny pro vypracování:

- 1) Prostudujte specifiky tvorby programů pro práci v reálném čase.
- 2) Seznamte se s konfiguračními registry Ethernet řadiče v PC modulu CML147786CX650HR-128.
- 3) Vytvořte "hard real time" kompatibilní ovladač. Vycházejte z poznatků diplomové práce [6].
- 4) Rozdělte program vhodným způsobem do modulů.
- 5) Naimplementujte inicializační a výkonné metody pro odesílání a příjem UDP packetů.
- 6) Naimplementujte podporu pro funkci DHCP klient.
- 7) Minimalizujte použití dynamické alokace paměti a strukturujte zdrojový program dle dodané šablony.
- 8) Při vývoji ukládejte změny ve zdrojových souborech verzovacím programem Mercurial.

Seznam odborné literatury:

- [1] <http://rtd.com>
- [2] Messmer, H., P.: Dembowski, K.: Velká kniha hardware. Computer press, Praha 2005.
- [3] Reinhart, D.: Řídicí software systému s maticovým měničem. Diplomová práce ČVUT, Praha 2006.
- [4] Fligl, S.: RT PC-based control system - The 2nd generation. Technická dokumentace Ruhr-Universität, Bochum 2003.
- [5] Klik, L.: Ovladač PC104 I/O modulu s možností ukládání sad záznamů měřených dat. Diplomová práce ČVUT, Praha 2011.
- [6] Martínek, R.: RT rozšíření OS DOS pro platformu PC104. Diplomová práce ČVUT, Praha 2013.
- [7] Svatoš, J.: Diagnostická aplikace regulátoru založená na platformě Netbeans. Diplomová práce ČVUT, Praha 2012.
- [8] Mejdrech, L.: Networking and TCP/IP stack for HelenOS system, Diplomová práce KU, Praha 2009.
- [9] Buttazzo, G. C.: Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications. New York, USA: Springer Science+Business Media, třetí vydání, 2011, ISBN 978-1-4614-0675-4.

Vedoucí: Ing. Stanislav Flígl, Ph.D.

Platnost zadání: do konce letního semestru 2015/2016

doc. Ing. Filip Železný, Ph.D.
vedoucí katedry



prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 31. 10. 2014

Poděkování

Rád bych poděkoval vedoucímu diplomové práce Ing. Stanislavu Flíglovi, Ph.D. za ochotu a připomínky při tvorbě této práce. Dále bych chtěl poděkovat své rodině a všem, kteří mi pomáhali při studiu.

Prohlášení

Prohlašuji, že jsem práci vypracoval samostatně a použil jsem pouze podklady uvedené v příloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 11.5.2015

Abstract

The aim of this work is to design and implement a driver for the network adapter situated on the CML147786CX650HR-128 module produced by RTD Embedded Technologies, Inc. The driver is intended to operate under the second generation of a RT kernel developed by Mr. R. Martinek in the case of his diploma thesis. The driver meets all time requirements for a selected particular hard real-time application.

The integral parts of the driver are: network card service routine, communication via IP and UDP protocols and DHCP client functionality. Other applications running under the RT kernel can utilize it in order to communicate via UDP. The driver is represented by static binary library. The communication from and towards to other modules can be conducted by using the mail service function provided by the kernel itself.

Abstrakt

Cílem této práce je navrhnout a implementovat ovladač pro síťový adaptér obsažený na modulu CML147786CX650HR-128 firmy RTD Embedded Technologies, Inc. Ovladač pracuje pod RT jádrem druhé generace vytvořeným R. Martínkem v rámci jeho diplomové práce a splňuje časové požadavky na maximální dobu exekuce povolené pro danou hard-real time aplikaci.

Součástí ovladače je obsluha hardware síťové karty, komunikace prostřednictvím protokolů IP a UDP, a funkce DHCP klient. Aplikace běžící pod RT jádrem může pomocí tohoto ovladače komunikovat prostřednictvím protokolu UDP. Ovladač je reprezentován binární statickou knihovnou, se kterou ostatní moduly komunikují pomocí zpráv předávaných prostřednictvím mail funkce RT jádra.

Obsah

1	Úvod	1
2	Analýza a návrh	3
2.1	Požadavky	3
2.2	Vývojové prostředí	4
2.3	Vlastnosti real-time systémů	4
2.4	Komunikace modulu ovladače s RT jádrem a aplikacemi	6
2.5	Síťový adaptér Intel 82559-ER	7
2.5.1	Intel 82559-ER ze softwarového pohledu	8
2.6	Síťová komunikace	13
2.6.1	Ethernetové rámce	13
2.6.2	Internetový protokol	14
2.6.3	Protokol UDP	16
2.6.4	Protokol ARP	18
2.6.5	Protokol DHCP	19
3	Realizace	23
3.1	Modul INTEL8255X	24
3.2	Modul NET_IP	26
3.2.1	Modul NET_IP – ARP	26
3.2.2	Modul NET_IP - IP	27
3.3	Modul NET_DHCP	30
3.4	Modul NET_UTILS	32
3.4.1	Čas v síťovém ovladači	32
3.5	Modul INTEL82559_DRV	33
3.5.1	Inicializace ovladače	33
3.5.2	Řídící a stavové funkce ovladače	34
3.5.3	Funkce pro práci s UDP pakety	35
3.5.4	Popis činnosti ovladače	36
3.6	Sestavení ovladače	38
4	Testy ovladače	39
4.1	Demonstrační aplikace UDPTest	39
4.2	Demonstrační aplikace NET_TEST	40
4.3	Testy komunikace pomocí UDP	40
4.4	Testy DHCP klienta	41
5	Závěr	43
6	Literatura	45
A	Seznam zkratk	47
B	Obsah příloženého CD	49
C	Obrazovka aplikace NET_TEST	51
D	Aplikace UDPTest	53

Seznam obrázků

Obr. 2.1 - Zasílání zpráv prostřednictvím RT jádra.....	6
Obr. 2.2 - Blokový diagram síťového adaptéru	7
Obr. 2.3 - DHCP relace – získání IP adresy	20
Obr. 3.1 - Začlenění síťového ovladače v systému.....	23
Obr. 3.2 - Fronta odchozích rámců	24
Obr. 3.3 - Vývojový diagram DPCP klienta	31
Obr. 3.4 - Vývojový diagram činnosti ovladače	36
Obr. 4.1 - Nefragmentovaný UDP paket	40
Obr. 4.2 - Fragmentovaný UDP paket	41
Obr. 4.3 - Záznam DHCP komunikace	42
Obr. C.1 - Snímek aplikace NET_TEST	51
Obr. D.2 - Aplikace UDPTest.....	53

Seznam tabulek

Tab. 2.1 - PCI konfigurační registry	8
Tab. 2.2 - PCI konfigurační registr Command	9
Tab. 2.3 - Stavové a řídicí registry adaptéru (SCR).....	10
Tab. 2.4 - Formát struktury Command Block.....	11
Tab. 2.5 - Formát struktury Receive Frame Descriptor	12
Tab. 2.6 - Formát Ethernetového rámce	14
Tab. 2.7 - Formát IP hlavičky	15
Tab. 2.8 - Formát UDP hlavičky.....	17
Tab. 2.9 - Formát IP pseudohlavičky.....	17
Tab. 2.10 - Formát ARP paketu	18
Tab. 2.11 - Formát DHCP zprávy.....	21
Tab. 4.1 - Formát UDP paketu testovacích aplikací	39

Seznam algoritmů

Alg. 3.1 - Zpracování přijatého ARP paketu	26
Alg. 3.2 - Sestavení IP paketu z fragmentů	28

1 Úvod

Možnost komunikace prostřednictvím síťových protokolů z rodiny IP je v současné době běžnou součástí aplikací. Proto vznikla potřeba přidat k Real Time (RT) jádru ovladač pro síťovou kartu a funkce pro příjem a odesílání UDP paketů. Tato práce navazuje na diplomovou práci Radomíra Martínka RT rozšíření OS DOS pro platformu PC104 implementující RT jádro [1]. Toto rozšíření je specifická aplikace pracující pod systémem DOS na platformě Intel x86.

Ovladač je implementován pro ethernetový modul Intel 82559-ER nacházející se na počítači CML147786CX650HR-128 firmy RTD Embedded Technologies, Inc. Použití tohoto počítače je v zadání této diplomové práce a vychází z předchozích prací a existujících úloh vytvořených pro tento počítač.

Modul ovladače spolu s funkcemi pro příjem a odesílání UDP paketů tvoří jednu binární knihovnu. Samotné RT jádro a případné ovladače dalších periférií také tvoří binární knihovny. Tento přístup umožňuje jednoduché sestavení výsledné aplikace pracující pod RT jádrem a podporuje konfigurační management díky možnosti verzovat jednotlivé binární knihovny.

Protože aplikace vyžaduje běh v reálném čase, tak tvorba ovladače a obslužných funkcí se musí řídit specifiky vývoje software pro běh v reálném čase. V první řadě to znamená, že nesmí být překročen čas běhu přidělený plánovačem RT jádra. V opačném případě by mohl být narušen očekávaný běh aplikace a způsobeny problémy v aplikaci řízeném systému.

Knihovna modulu ovladače obsahuje i podpůrné funkce pro práci s protokolem IP. Jedná se funkce o DHCP klient, pro automatické nastavení IP adresy zařízení. Dále je to implementace protokolu ARP, který slouží ke zjištění fyzické adresy (MAC) protistrany na základě znalosti IP adresy.

2 Analýza a návrh

2.1 Požadavky

Na tvorbu ovladače síťového adaptéru jsou následující požadavky. Tyto požadavky vycházejí ze zadání diplomové práce a ze zachování jednotnosti vývoje software v rámci projektu.

- Vytvoření ovladače síťové karty Intel 82559-ER obsahující prostředky pro příjem a odesílání UDP paketů. Tento ovladač bude pracovat pod RT jádrem. Komunikace modulu ovladače síťového adaptéru s ostatními částmi systému bude prostřednictvím číslovaných zpráv. Tato část software bude pracovat v systému reálného času a její funkce musí dodržovat časové omezení na svůj běh.
- Modul bude obsahovat funkce pro inicializaci ovladače. Funkce budou umožňovat nastavení a přečtení IP adresy, fyzické adresy síťového adaptéru, přečtení statistických informací z adaptéru, zjištění a nastavení stavu síťové linky. Tyto funkce se volají před spuštěním smyčky RT jádra a nepodléhají požadavkům na časové omezení běhu funkce.
- Vytvoření statické funkce pro usnadnění práce s datovou strukturou UDP paketu. Jsou to funkce pro práci s položkami v hlavičce paketu, jejich čtení a nastavení.
- Realizace automatického nastavení parametrů IP rozhraní pomocí DHCP.
- Implementace protokolu ARP, pro zjištění fyzické síťové adresy.
- Vhodné rozdělení programu ovladače do modulů. Hlavní rozdělní bude na modul vlastního ovladače, práce s hardware síťové karty, a modul pro realizaci vyšších síťových vrstev dle referenčního modelu ISO/OSI.
- Použití programovacího jazyka C a vývojových prostředí Open Watcom a Eclipse. Výsledný modul ovladače síťové karty bude tvořit jednu binární knihovnu a hlavičkový soubor. Během tvorby software používat systém Mercurial pro správu verzí.

- Nepoužívat dynamické alokování paměti. Všechna potřebná paměť pro funkci modulu ovladače bude alokována staticky nebo bude při inicializaci ovladače z aplikace předán ukazatel volnou na paměť.

2.2 Vývojové prostředí

Software RT jádra, jakož i ostatní části projektu s jádrem spolupracující, jsou psány v jazyku C. Jako vývojové prostředí je použit Open Watcom verze 1.9. Toto prostředí umožňuje vytvářet programy pro více platform. V projektu RT jádra se využívá Extended DOS, 32 bitové rozšíření OS DOS pod názvem PMODW/W. To zajišťuje běh 32 bitové aplikace pod čistým systémem DOS, který je 16 bitový. Díky tomu může aplikace využívat celou paměť dostupnou na PC104 modulu počítače a ne jen 640kB využitelných pod systémem DOS.

Protože práce s IDE Open Watcom není uživatelsky přívětivá a nesplňuje již současné požadavky na vývojové prostředí, používá se z Open Watcom převážně jen kompilátor a linker. Jako uživatelské prostředí slouží Eclipse doplněné o skripty na kompilaci zdrojových kódů, které se jednoduše spouštějí přímo z Eclipse. Dále je Eclipse doplněno o regulární výrazy rozboru výpisu chyb produkovaného kompilátorem Open Watcom, pro snadnou identifikaci a lokalizaci chyb ve zdrojovém kódu.

2.3 Vlastnosti real-time systémů

Počítačový systém reálného času (real-time systém), je takový počítačový systém, v němž správné chování závisí nejen na správnosti výsledků použitých algoritmů, ale také na skutečném čase, v nichž jsou tyto výsledky dostupné. Počítačový systém pracující v reálném čase musí reagovat na podměty ze svého okolí v časových intervalech, které jsou určeny prostředím, kde tento systém pracuje a funkcí systému. Časový okamžik, kdy musí být výsledek dostupný, se nazývá *deadline*. Podle důsledků nedodržení tohoto okamžiku se systémy reálného času dělí do tří kategorií:

- *Soft* real-time systém – výsledek je použitelný i po *deadline*. Systém může pracovat dále.
- *Firm* real-time systém – výsledek není použitelný po *deadline*, ale nedojde k vážným následkům vlivem tohoto zpoždění.
- *Hard* real-time systém – nedodržení *deadline* může způsobit vážné problémy. U této kategorie systémů musí být výsledek dostupný za všech okolností do *deadline*.

V operačních systémech reálného času se používají různé metody pro plánování úloh. Ty lze rozdělit podle různých kritérií.

Jedno rozdělení je na statické a dynamické plánování. U statického plánování je plán dán při kompilaci a za běhu systému se nemění. Toto plánování vyžaduje znalost všech úloh vyskytujících se v systému pro vytvoření plánu. Dynamické plánování provádí rozhodnutí, kterou úlohu spustit jako další, na základě aktuálního stavu až za běhu systému. Vybere jednu z úloh, které čekají na přidělení času procesoru pro svůj běh. Statické plánování je jednodušší a má nižší vlastní režii než dynamické plánování. U statického plánování je také zajištěn determinismus systému. Dynamické plánování může vést k nedeterministickému chování systému.

Další kritérium rozdělení plánování úloh v systému je preemptivní a nepreemptivní. U nepreemptivního plánování aktuálně běžící úloha není přerušena, dokud sama nevrátí řízení jádru. U preemptivního plánování může být vykonávání úlohy přerušeno, pokud je potřeba provést úlohu s vyšší prioritou.

U *hard* real-time systémů se používá buď dynamické preemptivní, nebo statické nepreemptivní plánování.[2]

Ovladač síťového adaptéru, který je předmětem této diplomové práce, bude součástí *hard* real-time systému. To znamená, že jeho činnost nesmí způsobit překročení *deadline*. Proto se vyžaduje jeho definované a předvídatelné chování za všech okolností. Kvůli snadnějšímu zajištění determinismu celého systému, nejsou povolené přerušení. Z toho vyplývá i chování funkcí ovladače. RT jádro používá nepreemptivní statické plánování. Proto musí funkce ovladače po zavolání jádrem, vrátit řízení zpět jádru do definovaného času.

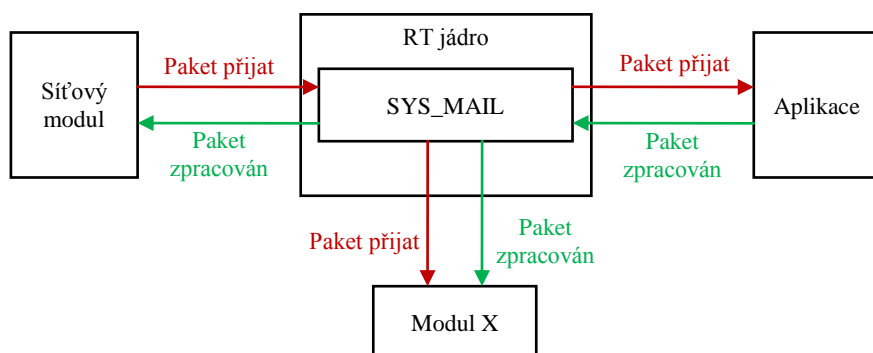
Ke zvýšení determinismu systému postaveného nad RT jádrem a tím i zajištění, že *hard* real-time úloha bude pracovat správně, přispívá také zvolený programovací jazyk a používání pouze staticky alokované paměti. U jazyka C je výsledný nativní strojový kód dán při kompilaci na rozdíl od interpretovaných jazyků. Díky statické alokaci paměti a eliminaci rekurze nemůže dojít k vyčerpání volné paměti a nemožnosti přidělení další paměti za běhu systému. Alokace dynamické paměti s sebou nese režii danou hledáním volné paměti. Čas potřebný na tuto činnost je obecně nedefinovaný a záleží na konkrétních knihovnách.

2.4 Komunikace modulu ovladače s RT jádrem a aplikacemi

Modul ovladače síťového adaptéru tvoří samostatnou binární knihovnu, která je závislá jen na RT jádru. Pro nastavení síťového rozhraní se použijí inicializační funkce ovladače. Tyto se zavolají přímo z aplikace. Pro přenos UDP paketů mezi síťovým ovladačem a jinými komponentami systému se použije služba *SYS_MAIL* RT jádra. Tato služba poskytuje rozhraní pro zasílání zpráv zaregistrovaným příjemcům. Každá zpráva je definována jako struktura *SYS_MAIL_T* obsahující identifikátor zprávy *mailId* a ukazatel na data *pData*. Identifikátor zprávy je přirozené číslo určující význam zprávy a dat, na které ukazuje *pData*.

Po obdržení zprávy si daný modul zapamatuje tuto zprávu, ale ihned ji nezpracovává. Ke zpracování zprávy dochází, až když RT jádro přidělí modulu procesorový čas. Případná odpověď na zaslou zprávu je realizována jako další zpráva s jiným identifikátorem.

Zprávy jsou rozesílány vždy všem, kdo si zaregistroval příjem zpráv. Příjemce rozhoduje,



Obr. 2.1 - Zaslání zpráv prostřednictvím RT jádra

jestli zpráva je nebo není určená pro něj a jestli ji zpracuje nebo ignoruje.

Uvedený obrázek ukazuje, jak je po přijetí paketu síťovým modulem odeslána zpráva *Paket přijat*. Tento paket zpracuje aplikace a následně odešle zprávu *Paket zpracován*.

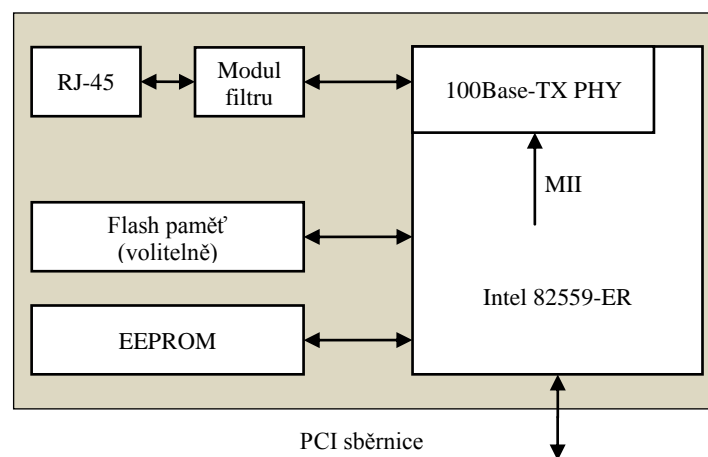
2.5 Síťový adaptér Intel 82559-ER

Síťový adaptér Intel 82559-ER podporuje standard ANSI/IEEE 802.3u pro 100BASE-TX a 10BASE-T provoz. Hlavní části adaptéru jsou:

- Fast Ethernet Media Access Controller (MAC) – adresování a řízení přístupu k přenosovému médium
- Physical Layer (PHY) interface – převádí bitový tok dat na fyzický signál
- Sériová EEPROM paměť – zde je uložena konfigurace adaptéru

Obecně jsou PHY interface a MAC dvě fyzické komponenty. U tohoto síťového adaptéru jsou integrovány do jedné součástky 82559-ER. Logická struktura však zůstává zachována a komunikace mezi MAC a PHY probíhá přes Media Independent Interface (MII). Flash paměť není na počítači CML147786CX650HR-128 osazena. Tato flash paměť má využití při načítání operačního systému ze sítě místo z disku počítače. Síťový adaptér je připojený k počítači přes PCI sběrnici. Kompletní blokové schéma adaptéru je na Obr. 2.2.

Adaptér podporuje plně duplexní i polo duplexní komunikaci. Obsahuje příjmovou a vysílací 3 KB FIFO paměť. Podporuje auto-negotiation, tj. automatické nastavení přenosové rychlosti a duplexního režimu, podle možností sítě. Poskytuje statistické informace o počtu přenesených rámců, chyb při přenosu na straně Ethernetu a počtu chyb při přenosu na straně počítače, například z důvodu zaplnění příjmové fronty.



Obr. 2.2 - Blokový diagram síťového adaptéru

2.5.1 Intel 82559-ER ze softwarového pohledu

2.5.1.1 Konfigurační registry PCI

Proto, aby bylo možné komunikovat s nějakým zařízením na PCI sběrnici, musí být známy jeho báze adresy, v jakém prostoru se tyto adresy nachází (mapované do paměti nebo I/O) a číslo přerušení, které zařízení generuje. Tyto informace lze zjistit u PCI zařízení díky podpoře *Plug and play*. Po startu počítače software, obvykle BIOS, po zjištění přítomných zařízení jim přiřadí báze adresy a linky přerušení. Toto nastavení je přístupné přes PCI konfigurační prostor. Pomocí volání služeb BIOSu lze zjistit přítomná zařízení zapojená na PCI sběrnici a pak u zvolených zařízení přečíst nebo upravit jejich nastavení. Jednotlivá zařízení jsou identifikována dvojicí čísel *Vendor ID* a *Device ID*, případně ještě pořadovým číslem zařízení, pokud je v systému více stejných zařízení. Intel 82559-ER má *Vendor ID* 8086h a *Device ID* 1209h.

31	24	23	16	15	8	7	0	Adresa
Device ID				Vendor ID				00h
Status				Command				04h
Class Code						Revision ID		08h
BIST	Header Type		Latency Timer		Cache Line Size			0Ch
CSR Memory Mapped Base Address Register								10h
CSR I/O Mapped Base Address Register								14h
Flash Memory Mapped Base Address Register								18h
Reserved Base Address Register								1Ch
Reserved Base Address Register								20h
Reserved Base Address Register								24h
Reserved								28h
Subsystem ID				Subsystem Vendor ID				2Ch
Expansion ROM Base Address Register								30h
Reserved						Cap_Ptr		34h
Reserved								38h
Max_Lat		Min_Gnt		Interrupt Pin		Interrupt Line		3Ch
Power Management Capabilities				Next Item Ptr		Capability ID		DCh
Reserved		Data		Power Management CSR				E0h

Tab. 2.1 - PCI konfigurační registry

Tabulka ukazuje konfigurační registry, které používá síťový adaptér. Kromě prvních dvou registrů jsou pro tvorbu ovladače důležité následující registry.

- Command – povoluje se zde funkce *Bus Master*, nezbytná pro správnou činnost adaptéru. Zařízení může pak dočasně převzít kontrolu nad PCI sběrnici a přenést data z/do paměti počítače pomocí DMA. Aktivuje se zde přístup do paměťového I/O prostoru adaptéru, bity *Memory Space* a *I/O Space* mohou být nastaveny nezávisle na sobě.

Bit	Popis
0	I/O Space
1	Memory Space
2	Bus Master Enable
3	0
4	Memory Write and Invalidate
5	0
6	Parity Error Response
7	0
8	SERR# Enable
9	0
10 - 15	Reserved

Tab. 2.2 - PCI konfigurační registr Command

- Latency Timer – určuje maximální čas, po který může zařízení kontrolovat PCI sběrnici, pokud je jako Bus Master. Jednotka je cyklus PCI sběrnice.
- Base Address Register – zde jsou bázové adresy pro přístup k adaptéru. Nejnižší bit určuje typ adresy, hodnota 0 znamená mapovaná do paměti, hodnota 1 mapovaná do I/O prostoru.
- Minimum Grant Register (Min_Gnt) – minimální čas v cyklech PCI sběrnice, kdy zařízení kontroluje PCI sběrnici. Registr je jen pro čtení.
- Maximum Latency Register (Max_Lat) – určuje jak často zařízení potřebuje přístup k PCI sběrnici. Registr je jen pro čtení, jednotka je cyklus PCI sběrnice.

Ostatní registry nejsou z hlediska vývoje tohoto ovladače významné, proto zde není uveden jejich popis. Kompletní popis registrů je v dokumentaci Intel 82559-ER [3].

2.5.1.2 Softwarové rozhraní adaptéru

Software v počítači komunikuje se síťovým adaptérem pomocí registrů i sdílené paměti. Struktury v paměti lze rozdělit do těchto tří částí:

- Control / Status Registers (CSR) – řídicí a stavové registry. Nacházejí se na adaptéru a lze k nim přistupovat přes paměť nebo I/O. Bázová adresa těchto registrů je v konfiguračním prostoru PCI.
- Command Block List (CBL) – spojový seznam příkazových bloků (Command Block - CB) pro adaptér. Nachází se v hlavní paměti počítače.
- Receive Frame Area (RFA) – oblast pro přijímané rámce. Je to spojový seznam složený z popisovačů přijímaných rámců (Receive Frame Descriptor – RFD). Za každým RFD se nachází paměť pro přijímaný rámeček. Nachází se v hlavní paměti počítače.

Prvních 8 bajtů z CSR se nazývá System Control Block (SCB). SCB představuje hlavní komunikační bod pro řízení adaptéru a zjišťování jeho stavu. Adaptér se skládá ze dvou výkonných jednotek, Command Unit (CU) a Receive Unit (RU). Command Unit provádí příkazy zadané v CBL, dokud nedojde na konec seznamu nebo není zastavena přes SCB. Receive Unit se chová podobně. Přijímá ethernetové rámce a ukládá je do hlavní paměti počítače, dokud je volné místo v RFA nebo není zastavena přes SCB.

2.5.1.3 Stavové a řídicí registry adaptéru

Tabulka Tab. 2.3 ukazuje řídicí a stavové registry (SCR) adaptéru.

31	24	23	16	15	8	7	0	Offset
SCB Command Word				SCB Status Word				00h
SCB General Pointer								04h
Port								08h
EEPROM Control Register				Reserved				0Ch
MDI Control Register								10h
Rx DMA Byte Count								14h
PMDR		Flow Control Register				Reserved		18h
Reserved				General Status		General Control		1Ch
Reserved								20h – 2Ch
Function Event Register								30h
Function Event Mask Register								24h
Function Present State Register								38h
Force Event Register								3Ch

Tab. 2.3 - Stavové a řídicí registry adaptéru (SCR)

- SCB Status Word obsahuje stav jednotek CU a RU, dále indikuje dokončení čtení / zápisu na MDI (Management Data Interface). Pomocí MDI komunikuje MAC s PHY.
- SCB Command Word obsahuje masky přerušení v horním bajtu. V dolním bajtu jsou příkazy pro CU a RU. K jejich provedení nedochází okamžitě, záleží na vnitřním stavu adaptéru. Akceptování příkazu je indikováno vynulováním dolního bajtu SBS Command Word.
- SCB General Pointer se používá u některých příkazů. Jsou v něm ukazatele na CBL nebo RFA.
- PORT registr umožňuje provést reset zařízení a diagnostiku adaptéru.
- EEPROM Control Register slouží ke komunikaci s připojenou EEPROM pamětí.
- MDI Control Register umožňuje konfiguraci a zjištění stavu PHY. Například přečtení stavu linky nebo nastavení přenosové rychlosti.

- General Status obsahuje informace o stavu linky, tj. přenosovou rychlost, plně nebo polo duplexní režim, a zda je linka v aktivním stavu.

2.5.1.4 Příkazy pro Command Unit

Příkazy pro Command Unit se předávají prostřednictvím Command Block List (CBL). Pro spuštění prvního příkazu se nejprve nastaví SCB General Pointer na adresu prvního CB a pak se zápisem do SCB Command Word spustí Command Unit.

31				16				15				0				Offset
Command Word								Status Word								00h
EL	S	I	XXXXXXXXXX				CMD	C	X	OK	XXXXXXXXXXXXXX				04h	
Link Address												08h				
Data																

Tab. 2.4 - Formát struktury Command Block

Command Word obsahuje bity:

- EL – jedničková hodnota znamená, že tento blok je poslední. Po vykonání tohoto příkazu se CU zastaví a je vyvoláno přerušení.
- S – jedničková hodnota znamená, že po vykonání příkazu se CU zastaví a je generováno přerušení. Na rozdíl od předchozího příkazu je načtena hodnota z pole Link Address a po opětovném spuštění se provede příkaz z CB nacházející se na této adrese.
- I – pokud obsahuje jedničku, je generováno přerušení po vykonání příkazu
- CMD – 3 bitová hodnota obsahující příkaz pro CU. Příkazy důležité pro tvorbu ovladače v této diplomové práci jsou Transmit, Configure a Individual Address Setup. Příkaz Transmit odešle jeden Ethernetový rámec. Příkazy Configure a Individual Address Setup slouží k nastavení adaptéru a MAC adresy.
- Ostatní bity mají význam jen u některých příkazů. Pokud se nevyužívají, musí být nastaveny na nulu.

Status Word obsahuje bity C a OK. Bit C je nastaven adaptérem po vykonání příkazu a bit OK určuje, zda byl příkaz proveden bez chyby. Některé příkazy využívají i jiné stavové bity.

Link Address obsahuje adresu následujícího CB. Tato adresa je relativní vzhledem k báze adrese CU. Fyzická adresa se určí jako součet z pole Link Address a báze adresy CU. Po zpracování příkazu z aktuálního CB se načte z této adresy nový CB a začne se s jeho zpracováním.

Pole Data je volitelné a jeho význam se může dle kontextu měnit: obsahuje například odkaz na přenášená data v případě příkazu Transmit, nebo jsou zde uložena konfigurační data u příkazů Configure a Individual Address Setup.

2.5.1.5 Práce s Receive Unit

Jednotka pro příjem rámců pracuje s Receive Frame Area (RFA), tj. oblastí v paměti připravené ovladačem síťové karty, kam se ukládají přijaté rámce. Úkolem ovladače je zajistit, aby byly dostupné volné Receive Frame Descriptors (RFD) pro přijímané rámce. Jinak by docházelo k jejich ztrátě. Indikace ztráty rámce je v SCB Status Word viz Tab. 2.3.

31		16				15				0		Offset	
Command Word						Status Word						00h	
EL	S	00000000				H	SF	000		C	0	OK	Status Bits
Link Address												04h	
Reserved												08h	
00	Size				EOF	F	Actual Count				0Ch		

Tab. 2.5 - Formát struktury Receive Frame Descriptor

Struktura je podobná CB. Jsou zde však navíc stavové bity, indikující problémy při příjmu rámce a shodu adresy v rámci s adresou nastavenou v adaptéru. Většina z těchto bitů se nastavuje jen v případě, že je nastaveno ukládání chybných rámců, i když chyby mohou vzniknout až později po přijetí rámce adaptérem (např. nedostatečně velké místo rezervované pro rámeček).

Pole Size určuje velikost vyhrazené paměti pro rámeček. Obsah přijatého rámce je uložen hned za RFD. Bit EOF nastaví adaptér na 1 poté, co jsou data z rámce kompletně nahrána do paměti počítače. V poli Actual Count je po přijetí rámce a jeho uložení do paměti zaznamenána velikost přijatého rámce zvětšená o velikost struktury RFD.

Po nastavení báze adresy RU a adresy prvního RFD se může Receive Unit spustit. Ta si po načtení RFD a její analýze uloží adresu následující RFD z pole Link Address. Při příjmu rámce se pomocí DMA zapisuje jeho obsah za RFD. Když je celý rámeček zapsán do paměti nastaví se Status Word, bit EOF a pole Actual Count. Pak se RU posune na další RFD, podle pole Link Address, a postup se opakuje. Po přijetí a zápisu rámce do paměti počítače je generováno přerušení a nastaven příznak v SCB Status Word.

RU poskytuje akceleraci výpočtů kontrolních součtů TCP a UDP paketů. Pokud je tato funkce zapnutá, tak se u každého přijatého Ethernetového rámce spočítá jeho součet. Součet je dvoubajtová hodnota a počítá ze všech bajtů, co se zapisují do paměti počítače. Adaptér ji pak zapíše za přijatý rámeček a velikost přijatých dat je pak o 2 bajty větší.

Ovladač musí z tohoto součtu odečíst data z Ethernetové a IP hlavičky. Takže místo počítání součtu z celého rámce se počítá součet jen z hlaviček.

2.6 Síťová komunikace

Síťová komunikace je rozdělena do několika úrovní. Toto rozdělení popisuje referenční model ISO/OSI. Model rozděluje komunikaci v počítačových sítích do 7 vrstev. Při komunikaci procházejí požadavky od odesílatelovy nejvyšší vrstvy k nejnižší. Na straně příjemce je tomu analogicky ale v opačném pořadí. Komunikace dvou systémů se řídí protokoly a probíhá mezi stejnými úrovněmi dvou systémů. Ne vždy jsou však všechny úrovně tohoto modelu využity. Na úrovni jednoho systému mezi sebou komunikují jen vrstvy sousední (o jednu úroveň vyšší nebo nižší).

Vrstvy modelu IOS/OSI:

1. Fyzická – specifikuje fyzické parametry komunikace. Tj, vlastnosti kabelů, konektorů, modulace signálu a napětové úrovně.
2. Linková – propojení dvou sousedních systémů. Provádí převod dat z fyzické vrstvy do rámců a zpět.
3. Síťová – realizuje adresování a směrování v síti. Umožňuje komunikaci systémů, které spolu nesousedí.
4. Transportní – zajišťuje přenos dat mezi koncovými uzly.
5. Relační – řídí dialog mezi komunikujícími systémy. Navazuje a ukončuje spojení.
6. Prezentační – provádí mapování dat mezi aplikační a relační vrstvou.
7. Aplikační – tato vrstva poskytuje rozhraní pro komunikaci aplikací.

Síťový ovladač v této diplomové práci pracuje na úrovních linkové až transportní vrstvy.

2.6.1 Ethernetové rámce

Síťový adaptér Intel 82559-ER pracuje na linkové vrstvě s Ethernetovými rámci, přesněji rámci typu Ethernet II. Data těchto rámců se odesílají jako bitová posloupnost. Rámec se skládá z oktětů, tj. osmic bitů. Toto označení vzniklo z důvodu rozdílné délky slova u různých typů počítačů. Rámec začíná preambulí, která slouží k synchronizaci hodin příjemce. Preambule je tvořena 7 oktety střídajících se 1 a 0. Následuje označení začátku rámce, adresy cíle a zdroje, typ rámce a data. Rámec je zakončený kontrolním součtem CRC32. Kontrolní součet vypočítává a ověřuje síťový adaptér. V datové části je vložený paket vyššího protokolu, v této diplomové práci je to buď paket IP, nebo ARP. Z pohledu

ovladače je vidět jen část obsahu rámce, od adresy cíle po datovou část. Ostatní části jsou doplňovány a odebírány při zpracování rámce adaptérem.

Preamble	SFD	Adresa zdroje	Adresa cíle	Typ / Délka	Data	CRC32
7 oktetů	1 oktet	6 oktetů	6 oktetů	2 oktety	46 - 1500 oktetů	4 oktety

Tab. 2.6 - Formát Ethernetového rámce

Pokud je v poli Typ / Délka hodnota větší než 1500, jedná se o Ethernet II a toto pole určuje typ vyššího protokolu dat v rámci. Pokud je přenášených dat méně než 46 oktetů, doplní adaptér datové pole nulami až do minimální velikosti.

2.6.2 Internetový protokol

Internetový protokol (IP) je široce rozšířený protokol pro přenos dat. V současnosti je nejrozšířenější jeho revize 4 (IPv4). Tato revize začíná být postupně nahrazována novější IPv6. V této práci je implementován jen starší a rozšířenější IPv4. Specifikace protokolu IPv4 je v RFC 791 [5].

Protokol se používá v sítích s přepojováním paketů. Tento protokol nezaručuje doručení přenášených dat. Není zaručeno pořadí přijímaných datagramů a je možný případný výskyt duplicitního přijetí datagramu. Není zaručena integrita přenášených dat. Zaručení spolehlivosti přenosu má na starosti vyšší protokol nebo aplikace, pokud je ji potřeba zajistit.

Jednotlivé koncové uzly v IPv4 jsou identifikovány 4 bajtovou adresou. Tato adresa je rozdělena na dvě části, identifikaci sítě a identifikaci koncového uzlu. Identifikace sítě je ve vyšších bitech adresy. Rozdělení mezi tyto dvě části je dáno buď třídou sítě (A, B, C, D a E) nebo podle novějších standardů je vyjádřeno počtem bitů identifikace sítě (Classless Inter-Domain Routing – CIDR). Adresa se nejčastěji zapisuje v tečkové notaci v desítkové soustavě po bajtech. Síťová adresa je určena buď maskou sítě, nebo počtem bitů adresy sítě. Například 192.168.1.5/24, kde 24 znamená, že adresa sítě je prvních 24 bitů, tj. 192.168.1.0. Masky této sítě je 255.255.255.0. K propojení sítí slouží směrovače (routry). Některé rozsahy adres sítí jsou vyhrazené jako privátní sítě a pakety z těchto sítí nejsou přenášeny dále za směrovač. Pro snadnější identifikaci se používá doménové jméno, které se dotazem na DNS server přeloží na IP adresu. V diplomové práci toto není implementováno. Jsou podporovány jen IP adresy v numerické formě.

IP paket se skládá z hlavičky a datové části. Obsah hlavičky je zabezpečen kontrolním součtem. Datová část zabezpečena není a spoléhá se na nižší vrstvu nebo zabezpečení spolehlivosti přenosu zajišťuje vyšší vrstva. U Ethernetu je každý rámec zajištěn CRC32.

Offset	0	7	8	15	16	23	24	31
0	Version	IHL	DSCP	ENC	Total Length			
32	Identification				Flags	Fragment Offset		
64	Time To Live		Protocol		Header Checksum			
96	Source IP Address							
128	Destination IP Address							
160	Options							

Tab. 2.7 - Formát IP hlavičky

- Version označuje verzi protokolu a má fixní hodnotu 4 pro IPv4.
- IHL je 4 bitové pole, kde je uložena velikost hlavičky. Hodnota je v 32 bitových slovech. Minimální hodnota je 5, tj. 20 bajtů.
- DSCP (Differential Services Code Point) obsahuje značku pro mechanismy zajišťující službu s definovanou kvalitou (QoS)
- ENC (Explicit Congestion Notification) je volitelné pole, dle RFC 3168.
- Total Length obsahuje celkovou délku paketu v bajtech. Pokud je paket rozdělen do více fragmentů, je zde uložena velikost fragmentu. Celková velikost paketu je v případě jeho fragmentace známa až po přijetí posledního fragmentu.
- Identification obsahuje identifikaci paketu. Slouží k rozpoznání fragmentů paketu při jeho sestavování při příjmu. Všechny fragmenty mají toto pole stejné.
- Flags je 3 bitové pole pro řízení a identifikaci fragmentování paketu.
 - Bit 0 má název MF (More Fragments) – označuje fragmentovaný paket. Poslední fragment paketu má tento bit nulový.
 - Bit 1 je označen jako DF (Don't Fragment) – nefragmentovat. Pokud není možné takovýto paket odeslat bez fragmentace, tak je takovýto paket zahozen.
 - Bit 2 musí být vždy 0.
- Fragment Offset určuje pozici fragmentu v paketu vzhledem k jeho počátku. Je udávána v osmibitových blocích.
- Time To Live (TTL) určuje maximální stáří paketu v sekundách. Pokaždé když paket opustí směrovač, sníží se tato hodnota o jedničku. Pokud je tato hodnota nulová, je paket zahozen. Slouží jako prevence proti posílání paketu v kruhu a také při sestavování paketu z jeho fragmentů (pro případ že nějaký fragment nedorazí).

- Protocol identifikuje použitý vyšší protokol. Například UDP (17) nebo TCP (6).
- Header Checksum obsahuje kontrolní součet hlavičky. Je to jedničkový doplněk 16 bitového součtu obsahu hlavičky. Při vytváření kontrolního součtu je pole Header Checksum nulové. Při ověřování kontrolního součtu musí být výsledek součtu obsahu hlavičky roven 0. Pokud není, je paket zahozen.
- Source IP Address obsahuje odesílatele paketu.
- Destination IP Address obsahuje adresu příjemce paketu.
- Options jsou volitelné části hlavičky.

Adresy adresáta a příjemce se mohou během přenosu paketu v síti měnit. Například při přenosu paketu z privátní sítě do internetu se ve směrovači přepíše adresa odesílatele na internetovou adresu směrovače.

Datová část paketu následuje bezprostředně za hlavičkou. Číselné hodnoty protokolů a Options spravuje organizace Internet Assigned Numbers Authority (IANA), <http://www.iana.org>.

S fragmentací IP paketu souvisí hodnota maximum transmission unit (MTU). Určuje maximální velikost fragmentu IP paketu včetně hlavičky. Pokud je paket nebo fragment větší než hodnota MTU a není nastaven bit DF, je rozdělen na více fragmentů. MTU je dáno možnostmi nižších síťových vrstev. Ethernet má maximální velikost datové části 1500 bajtů, proto je většinou MTU nastaveno na tuto hodnotu.

Pro podporu protokolu IP a konfiguraci zařízení připojených k síti slouží řada protokolů. V této práci je implementován protokol ARP a DHCP. Významným protokolem je ICMP (Internet Control Message Protocol). Jednou z jeho hlavních funkcí je reportovat chybové stavy při přenosu paketu, např. pokud hodnota TTL dosáhne 0 nebo cíl není dostupný. V takové případě odešle odesílateli zprávu o zahozeném paketu s důvodem jeho zahození. Další funkcí je Echo, tj. ověření dostupnosti adresáta. Protokol ICMP není v této diplomové práci implementován.

2.6.3 Protokol UDP

Protokol UDP (User Datagram Protocol) popisuje RFC 768 [7]. Protokol nepřidává k IP žádné zajištění přenosu paketu, má tedy v tomto směru stejné vlastnosti. Je jednoduchý a bezstavový, využívají ho jednodušší protokoly typu dotaz – odpověď jako je NTP a DNS, nebo má využití u přenosu obrazu a zvuku. Přidává k adresování IP protokolu

čísla portů pro rozlišení více aplikací běžících na jednom počítači. Čísla portů jsou 16 bitové hodnoty a ty jsou rozděleny dále do několika intervalů.

- 0 je rezervovaná hodnota. Lze ji použít jako zdrojový port v případě, že není očekávána odpověď na takovýto paket.
- 1 až 1023 jsou porty dobře známých služeb, pro spuštění aplikace přijímající pakety na těchto portech je obvykle potřeba mít administrátorská práva.
- 1024 až 49151 jsou porty zaregistrované u IANA.
- 49152 až 65535 se používají jako zdrojové porty při komunikaci klienta se serverem.

Offset	0	15	16	31
0	Source Port		Destination Port	
32	Length		Checksum	

Tab. 2.8 - Formát UDP hlavičky

- Source Port identifikuje port odesílatele. Příjemce použije toto číslo k odeslání odpovědi.
- Destination Port identifikuje port příjemce.
- Length určuje celkovou délku UDP. To znamená velikost UDP hlavičky a datové části.
- Checksum je volitelný kontrolní součet. Pokud není použit, je zde 0.

Datová část paketu následuje ihned za hlavičkou.

Kontrolní součet UDP se počítá z obsahu UDP hlavičky a datové části a takzvané pseudohlavičky IP. Je to struktura vzniklá spojením částí hlavičky IP s UDP hlavičkou. Způsob výpočtu je stejný jako u kontrolního součtu IP.

Offset	0	7	8	15	16	23	24	31
0	Source IP Address							
32	Destination IP Address							
64	0	Protocol			UDP Length			
96	Source Port				Destination Port			
128	Length				Checksum			

Tab. 2.9 - Formát IP pseudohlavičky

Pole UDP Length obsahuje součet velikosti UDP hlavičky a datové části.

2.6.4 Protokol ARP

Address Resolution Protocol (ARP) je podpůrný protokol IP. Slouží ke zjištění linkové adresy adresáta na základě znalosti jeho IP adresy. Při komunikaci protokolem IP je adresát identifikován svou IP adresou. Linková vrstva síťové komunikace má svoji vlastní adresaci a neřídí se adresou vyššího protokolu. Proto je nutné zjistit linkovou adresu při odesílání IP paketu. Specifikace ARP je v RFC 826 [8].

Offset	0	1
0	Hardware Type (HTYPE)	
2	Protocol Type (PTYPE)	
4	Hardware Address Length (HLEN)	Protocol Address Length (PLEN)
6	Operation (OPER)	
8	Sender Hardware Address (SHA)	
	Sender protocol Address (SPA)	
	Target Hardware Address (THA)	
	Target protocol Address (TPA)	

Tab. 2.10 - Formát ARP paketu

- Hardware Type označuje typ protokolu linkové vrstvy. Ethernet má přiřazenu hodnotu 1.
- Protocol Type označuje typ protokolu síťové vrstvy. IP má přiřazenu hodnotu 0800h.
- Hardware Address Length určuje délku adresy linkové vrstvy v bajtech. Velikost linkové adresy u Ethernetu je 6 bajtů.
- Protocol Address Length určuje délku adresy síťové vrstvy v bajtech. Velikost IPv4 adresy jsou 4 bajty.
- Operation obsahuje požadovanou operaci. 1 je pro žádost o sdělení linkové adresy. 2 je pro odpověď. Pole může obsahovat i jiné kódy operací, které ARP podporuje, ale ty nejsou podstatné pro tvorbu ovladače v této diplomové práci.
- Následují linkové a síťové adresy odesílatele a cíle.

Paket ARP žádosti se odesílá jako broadcast (linková adresa ff:ff:ff:ff:ff:ff). Obdrží jej tedy všichni na stejném segmentu sítě. Při odpovědi jsou vyplněné všechny adresy a ARP paket se posílá jen žadateli. Aby se před každým odesláním IP paketu nemusela zjišťovat linková adresa od adresáta, používá se ARP tabulka pro uchování již známých dvojic adres. Záznamy v této tabulce mohou být statické nebo dynamické. Statické záznamy jsou zadány při konfiguraci sítě nebo adaptéru. Dynamické záznamy se přidávají do tabulky na

základě ARP dotazů a mají časově omezenou platnost, aby se projevily změny v nastavení IP adres v síti.

2.6.5 Protokol DHCP

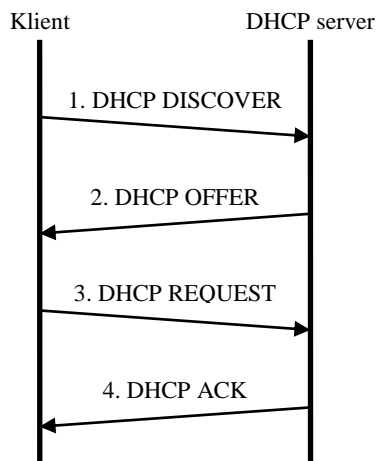
Dynamic Host Configuration Protocol (DHCP) je protokol, který slouží k distribuci konfiguračních parametrů sítě. Pomocí tohoto protokolu zjistí počítač nastavení sítě z DHCP serveru. Tímto je odstraněna nutnost manuálního nastavení sítě a přidává výhody centrální správy konfigurace sítě. Protože se konfigurace sítě může v čase měnit, mají obdržaná data omezenou platnost a musí se periodicky obnovovat. Základními parametry konfigurace sítě jsou IP adresa, maska sítě, adresa výchozí brány, adresy DNS serverů a časová platnost těchto záznamů. Možnosti DHCP jsou však mnohem větší. DHCP je protokol aplikační vrstvy a pro přenos dat využívá UDP. DHCP je specifikován v RFC 2131 [9]. Přidělení IP adres DHCP serverem žádajícímu zařízení může být realizováno několika způsoby:

- Statická alokace – DHCP server má tabulku se zadaným mapováním MAC adresy na IP adresu a na jejím základě odpovídá.
- Dynamická alokace – DHCP server vybírá z rozsahu adres. Žádající zařízení si „půjčuje“ adresu z tohoto rozsahu. DHCP server si udržuje informace, která adresa je použita a kdy vyprší platnost její výpůjčky.
- Automatická alokace – funguje podobně jako dynamická alokace s tím, že si DHCP server pamatuje výpůjčky adres i po vypršení její platnosti. Zařízení při opětovné žádosti dostane stejnou IP adresu.

Komunikace zařízení s DHCP serverem probíhá pomocí posílání zpráv na UDP port 67. DHCP server odpovídá zařízení zprávou na UDP port 68. Popis komunikace při získávání IP adresy klientem je znázorněn na Obr. 2.3.

- DHCP DISCOVER – používá klient ke zjištění dostupných DHCP serverů.
- DHCP OFFER – odpověď na předchozí zprávu. DHCP server v ní nabízí konfigurační parametry klientovi.
- DHCP REQUEST – žádost klienta o přidělení nabízených konfiguračních parametrů. Případně žádost o prodloužení výpůjčky IP adresy.
- DHCP ACK – DHCP server potvrzuje klientovi konfigurační parametry a platnost výpůjčky IP adresy, pokud bylo o ni žádáno.

- DHCP NAK – DHCP server oznamuje klientovi, že DHCP REQUEST nebyl akceptován. Například proto, že IP adresu už používá jiný klient.
- DHCP DECLINE – klient informuje DHCP server, že jemu přidělenou adresu používá někdo jiný. Může to zjistit například podle přijatých ARP paketů. Pokud toto detekuje, musí klient odeslat DHCP DECLINE a požádat znovu o IP adresu.
- DHCP RELEASE – klient ukončuje výpůjčku IP adresy. Například při vypínání počítače.
- DHCP INFORM – dotaz klienta na parametry sítě. Například se může webový prohlížeč informovat na adresu proxy serveru. DHCP server odpovídá na tuto zprávu zprávou DHCP ACK.



Obr. 2.3 - DHCP relace – získání IP adresy

Pokud klient nezná IP adresu DHCP serveru, začíná s ním komunikaci broadcastem na UDP port 67 se zprávou DHCP DISCOVER. Další komunikace již probíhá jen mezi klientem a serverem dle Obr. 2.3. Při žádosti o obnovení výpůjčky adresy, klient odešle DHCP REQUEST a DHCP server mu tuto žádost potvrdí, případně zamítne.

Všechny zprávy mají jednotný formát. Konfigurace a požadavky jsou umístěny na konci zprávy v poli Options. Toto pole nemá pevnou velikost. Seznam možných položek v poli Options začínají tzv. Magic Cookie, což je hodnota 63825363h, pak následuje typ zprávy. Konec pole Options je dán jednobajtovou zarážkou o hodnotě ffh.

Offset	0	7	8	15	16	23	24	31
0	Op		HType		HLen		Hops	
4	XId							
8	Secs				Flags			
12	CIAddr							
16	YIAddr							
20	SIAddr							
128	CHAddr							
132	GIAddr							
148	SName							
212	File							
340	Options							

Tab. 2.11 - Formát DHCP zprávy

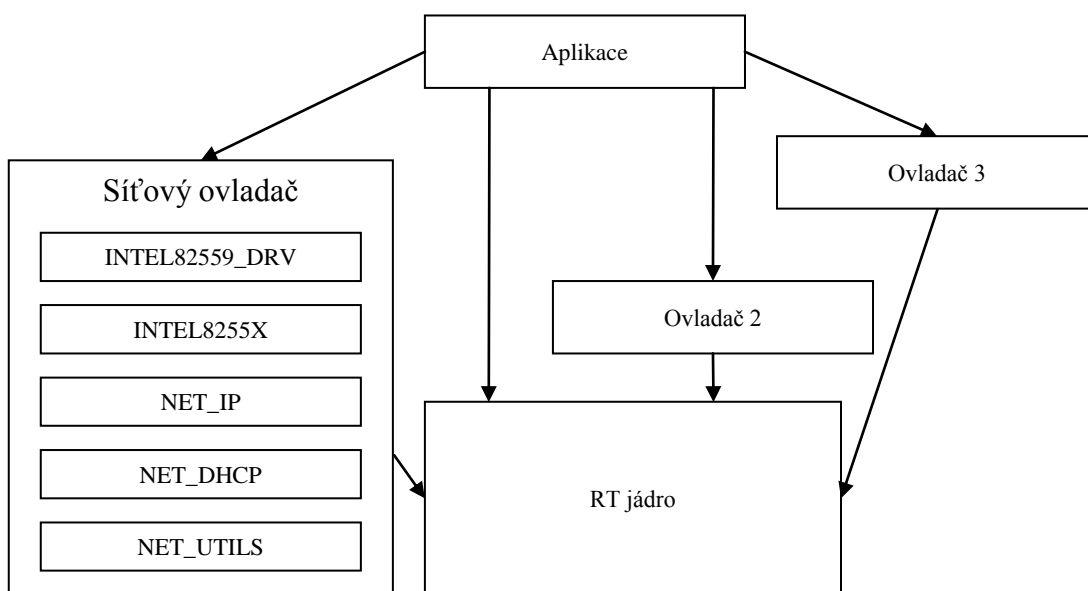
- Op rozlišuje mezi žádostí, hodnota 1, a odpovědí, hodnota 2.
- HType a HLen určují typ a velikost adresy linkové vrstvy. Pro Ethernet je to typ 1 a velikost 6. Typy adres přiděluje IANA a jsou v sekci ARP.
- XId je id transakce. Číslo náhodně vygenerované klientem. Slouží k propojení žádosti s odpovědí.
- Secs naplní klient. Obsahuje počet sekund od začátku relace s DHCP serverem.
- Flags obsahuje v nejvyšším bitu příznak broadcast. Tento bit nastaví klient, pokud chce, aby přišla odpověď jako broadcast. Ostatní bity jsou nulové.
- CIAddr naplní klient svou IP adresou, pokud ji už má přiřazenou.
- YIAddr obsahuje IP adresu klienta.
- CHAddr obsahuje adresu linkové vrstvy klienta. Zbytek pole je doplněn nulami.
- SName naplní klient svým jménem. Toto pole není povinné.
- Options obsahuje parametry, které nejsou obsažené ve struktuře DHCP zprávy. Jedná se například o typ požadovaných informací od DHCP serveru. Obsah tohoto pole specifikuje IANA.
- Ostatní pole se v této diplomové práci nevyužívají a jsou nulové.

3 Realizace

Ovladač síťového adaptéru je rozdělen do několika logických částí podle jejich funkce. Je tak možné v budoucnu rozšířit ovladač o podporu jiného síťového adaptéru nebo protokolu, a to bez větších zásahů do stávajícího ovladače.

Základní moduly síťového ovladače jsou:

- INTEL82559_DRV - Modul rozhraní komunikace s aplikací. Zde se nachází výkonná funkce ovladače, kterou periodicky volá RT jádro.
- INTEL8255X - Modul pro práci s hardware síťového adaptéru.
- NET_IP - Modul pro práci se síťovými protokoly IP, UDP a ARP.
- NET_DHCP - Modul DHCP klient.
- NET_UTILS - Modul podpůrných funkcí pro činnost ovladače ovladač.



Obr. 3.1 - Začlenění síťového ovladače v systému

V systému může být různý počet ovladačů. Tato diplomová práce obsahuje dva ovladače.

Všechny výše uvedené moduly síťového ovladače jsou součástí jedné binární knihovny. Spolu s touto knihovnou se distribuuje hlavičkový soubor s deklarací funkcí a datových typů použitých v rozhraní ovladače.

Součástí této diplomové práce je také speciální ovladač obsahující modul pro ověření funkčnosti síťového ovladače pracující pod RT jádrem a ten je doplněn dále testovací aplikací běžící pod OS Windows, která bude s tímto modulem v rámci testování komunikovat.

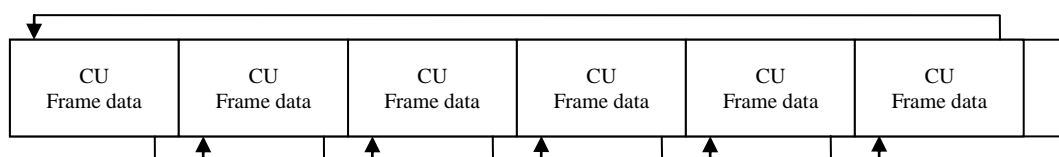
3.1 Modul INTEL8255X

Tato část síťového ovladače komunikuje přímo s hardware Intel 82559-ER. Poskytuje funkce pro inicializaci adaptéru, zjištění jeho stavu a příjem a odeslání Ethernetového rámce. Protože, jsou rozdíly v síťových adaptérech Intel 82557 až 82559 malé, je možné tento snadno upravit pro podporu více typů adaptérů. Proto má jméno tohoto modulu na konci písmeno x. V této diplomové práci je však implementován jen ovladač pro síťovou kartu nacházející se na průmyslovém počítači CML147786CX650HR-128.

Hlavní funkce adaptéru jsou příjem a odesílání Ethernetových rámců. Modul obsahuje dvě kruhové fronty, jednu pro přijímané a druhou pro odesílané rámce. Při inicializaci ovladače se modulu předají odkazy na paměť rezervovanou pro tyto fronty.

V inicializační funkci modulu se ve frontě pro příjem rámců vytvoří struktury RFD a vytvoří se z nich kruhový spojový seznam. Za RFD je vyhrazené místo pro přijímané rámce. Z jednotlivých RFD je vytvořen spojový seznam.

Ve frontě pro odesílání rámců se vytvoří struktury CB. Za samotným CB je vyhrazený prostor pro uložení dat odesílaného rámce. Jako u fronty pro příjem i zde je vytvořen spojový seznam z jednotlivých CB.



Obr. 3.2 - Fronta odchozích rámců

Počet CB a RFD je dán velikostí přidělené paměti při inicializaci. Každá položka ve frontě pevně danou velikost určenou maximální velikostí Ethernetového rámce a struktury RFD,

resp. CB. Protože se předává inicializační funkci velikost každé fronty v bajtech, může na jejím konci vzniknout nevyužívaná oblast.

V inicializační funkci dochází kromě nastavení front i k samotné inicializaci adaptéru. Ta spočívá v nalezení bazových adres adaptéru v paměťovém prostoru a provedení jeho resetu. Po vykonaném resetu se nastaví PHY na auto-negotiation, tj. automatická detekce přenosové rychlosti a duplexního režimu. Dále se přečte z EEPROM paměti MAC adresa adaptéru. Tato adresa se nastaví jako filtr. Adaptér pak přijímá jen rámce jemu určené. Nakonec se nastaví samotný adaptér a adresy CBL a RFA. Po této inicializaci je adaptér připraven na odesílání rámců a spuštění jejich příjmu.

Nastavení PHY se provádí jen při inicializaci a modul neobsahuje funkce pro změny v PHY. Komunikace s PHY probíhá po jednotlivých bitech a je časově náročná. Ovládají se výstupy elektricky připojené přímo na sériové rozhraní PHY. Pro odeslání každého bitu se musí nastavit jeho hodnota na výstup a provést jeden cyklus hodin, tj. nastavit bit hodinového signálu na jedničku a po prodlevě zpět na nulu. Čtení probíhá analogicky s tím, že se čtou jednotlivé datové bity a hodinový signál se řídí jako při zápisu.

Pro detekci přijatého rámce slouží funkce *Intel8255X_GetFrame*. Pokud vrátí *True*, je přijat alespoň jeden rámeček. Funkce má jeden parametr *skb*, ve kterém je v případě přijatého rámce vrácen ukazatel na data rámce a jeho velikost. Po vyzvednutí rámce z příjmové fronty se zavolá funkce *Intel8255X_FrameReceived*, která uvolní dané RFD pro příjem dalšího rámce. Pokud došlo k vyčerpání volných RFD, tak síťový adaptér zastavil příjem rámců. V takovém případě dojde ve funkci *Intel8255X_FrameReceived* po uvolnění RFD ke spuštění příjmu.

Parametr *skb* je struktura obsahující položky:

- *data* – obsahuje ukazatel na RFD
- *frame_data* – obsahuje ukazatel na data přijatého Ethernetového rámce.
- *len* – obsahuje velikost přijatého rámce v bajtech.

Odesílání rámců funguje analogicky k příjmu. Funkcí *Intel8255X_PrepareTransmitFrame* se zjistí adresa pro uložení odesílaného rámce. Po přenesení obsahu rámce na tuto adresu, funkce *Intel8255X_TransmitFrame* zajistí jeho odeslání. Pokud už jsou ve frontě jiné rámce k odeslání, tak se jen nastaví CB. Jinak se navíc spustí CU.

Modul obsahuje funkce pro přečtení stavu linky, statistických čítačů, MAC adresy a řízení příjmu rámců.

3.2 Modul NET_IP

Modul NET_IP se skládá ze dvou částí. První zpracovává IP a UDP pakety, druhá část obsluhuje ARP protokol. Pro přijetí Ethernetového rámce ze síťového adaptéru se určí protokol, kterým jsou data obsažená v Ethernetovém rámci přenášena. Na základě protokolu se tento rámec předá buď APR části NET_IP, anebo do IP části. Pokud má rámec jiný typ protokolu než ARP nebo IP, tak je zahozen.

3.2.1 Modul NET_IP – ARP

Zpracování přijatého ARP paketu se řídí doporučením v RFC 826 [8]. Je implementována jen podpora nezbytná pro tuto diplomovou práci, tj. protokol IPv4 a Ethernetové rámce. ARP část modulu obsahuje ARP tabulku. Tato tabulka má v každém řádku uloženou adresu linkové vrstvy (MAC adresa), IP adresu a čas poslední aktualizace. Příjem ARP paketu probíhá podle algoritmu, který reprezentuje následující pseudokód.

```

if "hardwarový typ == Ethernet"
    if "protokol == IPv4"
        merge := false
        if "IP adresa odesílatele je v tabulce"
            aktualizace tabulky
            merge := true

        if "IP adresa adresáta je moje"
            if "merge == false"
                přidání IP a MAC do tabulky

        if "je to ARP žádost"
            vytvoření ARP odpovědi a její odeslání

```

Alg. 3.1 - Zpracování přijatého ARP paketu

Při odesílání IP paketu je potřeba zjistit linkovou adresu. Toto se provede dotazem do ARP tabulky. Pokud se v ARP tabulce požadovaný záznam nachází a není zastaralý, tak se použije tato MAC adresa. Jinak se vytvoří ARP paket s dotazem na MAC adresu příslušnou k dané IP adrese a odešle se na síťový adaptér.

ARP tabulka má fixní velikost danou při kompilaci. Pokud dojde k jejímu zaplnění a je potřeba přidat nový záznam, tak se přepíše ten nejstarší.

3.2.2 Modul NET_IP - IP

Tato část modulu zpracovává IP a UDP pakety. Tento modul komunikuje pomocí *SYS_MAIL* (funkce jádra) s ostatními částmi systému. Zajišťuje rozdělení IP paketů na fragmenty, a pokud je to potřeba, i sestavení IP paketu z jeho fragmentů. Obsahuje dvě paměťové oblasti, jednu pro příchozí a druhou pro odchozí pakety. Tyto paměťové oblasti jsou organizovány jako kruhové fronty.

3.2.2.1 Příjem IP paketů

Při přijetí Ethernetového rámce a rozpoznání IPv4 protokolu se nejprve provede kontrola IP hlavičky. Pokud nesouhlasí kontrolní součet IP hlavičky, nebo se IP adresa adresáta neshoduje s nastavenou IP adresou v ovladači, tak se paket zahodí. Dále se provede kontrola protokolu nastaveného v IP hlavičce. Jiné než UDP pakety se zahodí. Když paket projde testy v IP hlavičce, pokračuje ve zpracování paketu. Projdou se příchozí pakety a hledá se paket se stejným odesílatelem a identifikací (pole Identification IP hlavičky). Počet prohlédnutých paketů je omezen, když se nenalezne hledaný paket a dosáhne se limitu, pokračuje se dál až při novém zavolání funkce ovladače RT jádrem. Pokud takový paket není nalezen, vytvoří se nová položka paketu a uloží se. Protože jako první nemusí být přijat první fragment IP paketu, nemusí být tedy ani známa jeho celková velikost. V takovém případě se pro paket rezervuje místo o maximální velikosti IP paketu. Když je přijat počáteční fragment IP paketu jako první, rezervuje se jen místo právě potřebné pro uložení konkrétního paketu. Při procházení paketů se kontroluje jejich stáří ve frontě, a pokud překročí hodnotu v TTL, je paket odstraněn. Protože z fragmentů IP paketu s výjimkou prvního není zjistitelný jeho UDP port, ukládají se všechny příchozí fragmenty. Teprve až po přijetí kompletního paketu se na základě UDP portu paket buď zahodí, nebo se pošle prostřednictvím *SYS_MAIL* aplikaci.

Hlavička záznamu s paketem obsahuje informace urychlující procházení přijatých paketů a jejich zprávu. Hlavička obsahuje tyto pole:

- *next* – Ukazatel na následující paket (hlavičku záznamu).
- *max_age* – Maximální stáří paketu. Hodnota vznikne součtem aktuálního času s TTL fragmentu paketu, který je přijat jako první.
- *first_hole* – Pořadové číslo bajtu první prázdné oblasti v paketu. Tj. takové oblasti, pro kterou ještě nebyly přijata data. Daný fragment zatím nebyl přijat.
- *identification* – Pole identification z IP hlavičky.
- *hdr_size* – Velikost IP a hlavičky v bajtech.

- *state* – Stav paketu. Indikuje, zda je paket kompletně přijatý, zpracovaný nebo paket není určen pro aplikaci a po přijetí bude zahozen.

Oblast v paměti, kde se ukládají přijaté pakety je realizována jako kruhová fronta.

Sestavování IP paketů z jejich fragmentů vychází z RFC 815 [10]. Algoritmus používá ke svojí činnosti strukturu *hole_descriptor* obsahující položky *first* a *last* obsahující první a poslední bajt v prázdné oblasti v paketu a položku *next*, kde je uložena adresa následující struktury *hole_descriptor*. Tyto struktury tvoří spojový seznam. Odkaz na první *hole_descriptor* je uložen v hlavičce záznamu. Na začátku je *first* nastaveno na 0 a *last* na maximální velikost, protože nejsou přijata žádná data z paketu. Podobně jsou u IP fragmentu zavedeny hodnoty *first* a *last* obsahující pozici prvního a posledního bajtu z fragmentu vzhledem k paketu. Při přijetí fragmentu se provádí algoritmus Alg. 3.2.

1. *if* “jsem na koci seznamu *hole_descriptor*”
 goto krok 9
2. *vyber* další *hole_descriptor* ze seznamu
3. *if* “*fragment.first* > *hole.last*”
 goto krok 1
4. *if* “*fragment.last* < *hole.first*”
 goto krok 1
5. *smaž* aktuální *hole_descriptor*
6. *if* “*fragment.first* > *hole.first*”
 vytvoř nový *hole_descriptor* s *first=hole.first* a *last=(fragment.first-1)*
7. *if* “*fragment.last* < *hole.last* a *fragment* paketu má nastaven MF (*more fragments*)”
 vytvoř nový *hole_descriptor* s *first=(fragment.last+1)* a *last= hole.last*
8. *goto* krok 1
9. *if* “seznam s *hole_descriptor* je prázdný”
 paket kompletně přijat
10. *return*. Čekání na další *fragment*.

Alg. 3.2 - Sestavení IP paketu z fragmentů

Protože pole *fragment offset* z IP hlavičky je v násobcích 8 bajtů, je tímto určena hranice, kde jsou rozděleny fragmenty, a je tak minimální velikost fragmentu. Tím vzniká místo pro uložení *hole_descriptor* bez použití další paměti. Struktura *hole_descriptor* je uložena na začátku každé prázdné oblasti.

Poté co je UDP paket kompletně přijat, je na základě UDP portu rozhodnuto, kdo je jeho příjemce. Buď to může být DHCP klient, pokud je zapnutý a číslo portu je 68. Jinak se odešle zpráva pomocí *SYS_MAIL*, že byl přijat paket.

Zpráva o dalším doručeném paketu se odesílá až po doručení potvrzení o zpracování předchozí zprávy.

3.2.2.2 Odesílání IP paketů

Síťový ovladač periodicky kontroluje stav fronty odchozích paketů. Pokud se v ní nachází paket, dojde nejprve k doplnění chybějících informací IP hlavičky, pak k samotnému odeslání paketu.

Odesílání se řídí parametrem MTU. Pokud je celkový počet bajtů vyšší než toto číslo, dochází k fragmentaci paketu. Jednotlivé fragmenty jsou předávány do modulu INTEL8255X, který si je ukládá do výstupní fronty. Předávání fragmentů probíhá postupně tak, aby se nepřekročil maximální čas běhu funkce ovladače.

Při odesílání paketu se doplňuje MAC adresa adresáta do Ethernetového rámce. Může nastat situace, že tato adresa není známa. Pak se vygeneruje ARP paket s dotazem na MAC adresu a čeká se na odpověď. Pokud se nepodaří zjistit MAC adresu, je paket zahozen, protože jej nelze odeslat.

Při dotazu na MAC adresu se kontroluje, jestli je IP adresa cíle ve stejné síti jako IP adresa odesílatele. Pokud tomu tak není, hledá se MAC adresa výchozí brány. Když v tomto případě není výchozí brána nastavena, je paket zahozen.

3.2.2.3 Komunikace s aplikací

UDP pakety jsou předávány aplikaci pomocí služby *SYS_MAIL* RT jádra. Tato služba rozesílá zaregistrovaným příjemcům zprávy obsahující její identifikátor a ukazatel na data. Identifikátory zpráv jsou zadané při inicializaci ovladače. Pro komunikaci aplikace s ovladačem síťové karty jsou definovány čtyři zprávy:

- `receivedUdp` – Zprávu odesílá síťový ovladač pro přijetí UDP paketu. Po obdržení této zprávy její příjemce vyhodnotí, případně zkopíruje, obsah UDP paketu a potvrdí síťovému ovladači jeho přijetí.

- `processedUdp` – Zprávu odesílá příjemce UDP paketu. Signalizuje síťovému ovladači, že může danou zprávu odstranit z fronty příchozích paketů.
- `sendingUdp` – Odesílá aplikace síťovému ovladači. Ovladač po jejím přijetí zařadí požadovaný UDP paket do odchozí fronty. Poté odpoví aplikaci, že může uvolnit paměť s UDP paketem.
- `sentUdp` – Odesílá síťový ovladač jako potvrzení přijetí UDP paketu k odeslání.

V systému může být v jednom okamžiku jen jedna nepotvrzená zpráva o přijetí, resp. požadavek na odeslání UDP paketu. Pokud aplikace odešle více zpráv `sendingUdp`, nebo by ovladač poslal vícekrát `receivedUdp`, tak by se zpracovala jen první zpráva a ostatní zprávy by se ignorovaly.

Ve zprávě je uložen ukazatel na strukturu s UDP paketem. Tato struktura se skládá z IP a UDP hlavičky (převzatých z prvního fragmentu paketu) a datové části.

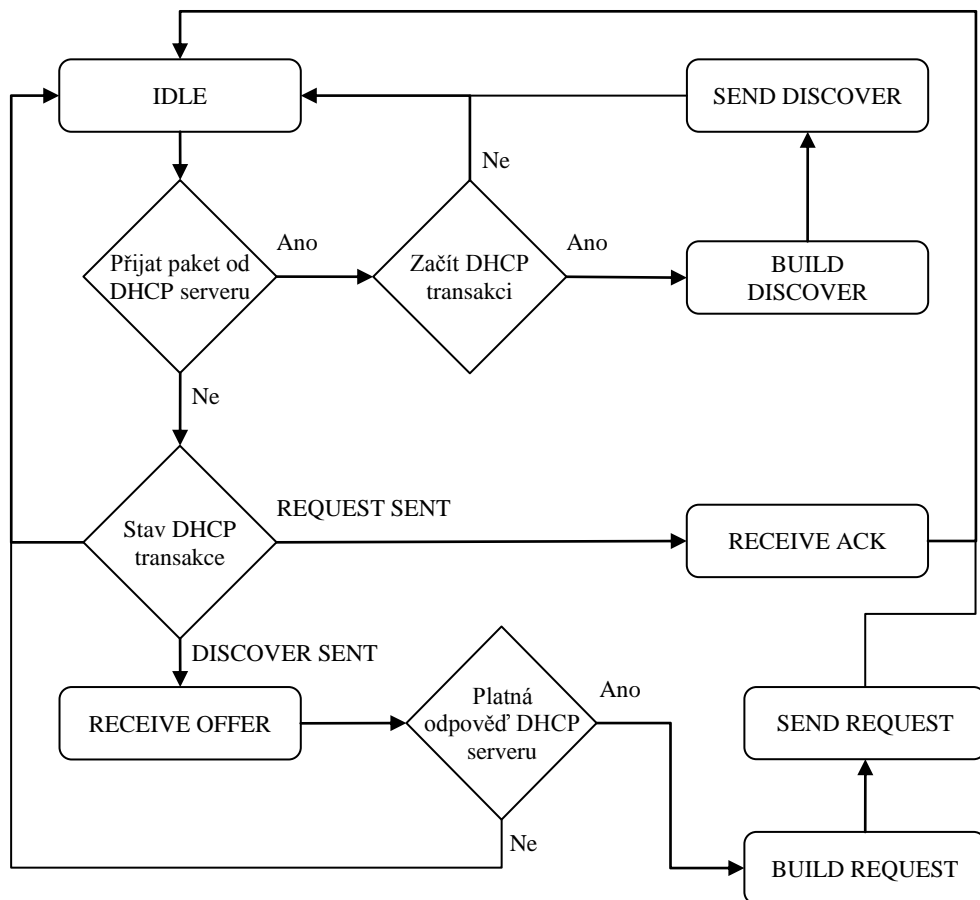
Před odesláním zprávy `sendingUdp` musí aplikace zadat hlavičky IP a UDP. Ovladač doplňuje pouze kontrolní součet hlaviček, délku fragmentů IP paketů a pole `identification`. K tomuto účelu jsou v síťovém ovladači vytvořené pomocné funkce. Pokud není zadána IP a MAC adresa odesílatele, tj. obě pole jsou nulové, doplní ji ovladač podle hodnot nastavených při inicializaci ovladače.

V případě, že není nastavena IP adresa počítače, nedochází k odeslání UDP paketů. Jen se v takovém případě potvrdí jejich převzetí a dané pakety se zahodí. Tato situace může nastat, pokud se nepodaří zjistit IP adresu z DHCP serveru.

3.3 Modul NET_DHCP

Modul DHCP se používá jen, pokud je povolena funkce DHCP klient při inicializaci ovladače. Modul má za úkol zajistit přidělení IP adresy, zjištění masky sítě, výchozí brány a hodnoty MTU od DHCP serveru. Po přidělení IP adresy musí tento modul periodicky obnovovat výpůjčku přidělené IP adresy. Žádost o obnovení výpůjčky se odesílá po překročení tří čtvrtin času výpůjčky. Při korektním ukončení práce programu a tím i ovladače, dojde k ukončení výpůjčky u DHCP serveru.

Žádné funkce tohoto modulu nejsou zpřístupněné pro aplikaci běžící pod RT jádrem. Aplikace může jen zjistit, zda byla IP adresa přidělena a síťový ovladač je schopen přijímat a odesílat UDP pakety.



Obr. 3.3 - Vývojový diagram DHCP klienta

Komunikace s DHCP serverem tvoří transakce, které mají svůj číselný identifikátor. Základní stav DHCP klienta je *IDLE*. Pokud zatím není přidělena IP adresa nebo se blíží konec její výpůjčky, začíná se nová transakce.

Odesílání paketů DHCP serveru je rozděleno do dvou částí *BUILD* a *SEND*. V části *BUILD* se vytváří UDP paket s požadavkem, a v části *SEND* se tento požadavek odesílá. Po odeslání se v části *SEND* upravuje stav DHCP transakce, na jehož základě se zpracovávají příchozí UDP pakety od DHCP severu.

Příchozí UDP pakety se přijímají, jen pokud jsou očekávané. Jinak se zahodí. Při příjmu se ověří identifikátor transakce, identifikátor DHCP severu a typ zprávy. Pokud údaje souhlasí, aktualizuje se stav transakce a DHCP klienta.

3.4 Modul NET_UTILS

V modulu NET_UTILS jsou podpůrné funkce pro činnost ovladače. Modul využívá knihovny pcilib vytvořené firmou RTD Embedded Technologies, Inc. Tato knihovna obsahuje funkce pro detekci síťového adaptéru v počítači a umožňuje přístup ke konfiguračním registrům PCI. Modul obsahuje také časový subsystém ovladače a definice struktur, které nespadají pod žádný z ostatních modulů.

3.4.1 Čas v síťovém ovladači

Modul obsahuje funkce pro zjištění aktuálního času v počítači. Jednak funkce čítá procesoru TSC a za druhé funkce pracující se systémovým časem. Pro čtení TSC (Time Stamp Counter), jsou v modulu dvě funkce:

- GetTSC64S – Funkce vrací 64 bitovou hodnotu čítače TSC za použití instrukce RDTSC. Před přečtením TSC provede serializaci pomocí instrukce CPUID. Tím se zaručí, že je čítač TSC přečtený až po vykonání všech instrukcí, umístěných v programu před touto funkcí. Serializace eliminuje vliv toho, že procesor může vykonávat instrukce mimo pořadí, než v jakém jsou v programu. U měření malých časů, jednotky až desítky instrukcí, by bez serializace docházelo k velkým nepřesnostem.
- GetTSC64N – Funkce vrací 64 hodnotu čítače TSC, ale neprovádí serializaci, před čtením čítače. Serializace má nenulovou režii a proto se neprovádí v případech, kdy se měří větší časový úsek. Například u systémového času.

Systémový čas není reálný čas, protože tento není k dispozici na tomto počítači. Počítač neobsahuje baterii pro zálohování RTC (Real Time Clock) a po startu počítače jsou hodiny vždy inicializované na stejnou hodnotu. Systémový čas se odvozuje od čítače TSC a je závislý na správném nastavení hodnoty frekvence procesoru při inicializaci ovladače.

Systémový čas se používá proto, že je to 32 bitová hodnota a jeho jednotka je vždy stejná. Jako jednotka času byla zvolena desetina sekundy. Hrubější dělení na jednotky sekundy by nemuselo vyhovovat při nižších hodnotách TTL příchozích fragmentů paketů. Nižší hodnota jednotky systémového času není potřeba pro činnost. Také by se zkrátila perioda, za kterou dojde k přetečení 32 bitového čítače, která je při jednotce času desetina sekundy přibližně 13,6 let.

Časové údaje jsou potřebné pro správnou činnost DHCP klienta, ARP tabulky a vyřazování neúplných IP paketů z fronty příchozích paketů.

3.5 Modul INTEL82559_DRV

Modul INTEL82559_DRV tvoří rozraní síťového ovladače a řídí jeho části. V tomto modulu je umístěna výkonná funkce ovladače, kterou periodicky volá RT jádro. Obsahuje také podpůrné funkce pro práci s UDP pakety, a funkce pro řízení a zjištění stavu adaptéru.

3.5.1 Inicializace ovladače

Před použitím ovladače se musí provést jeho inicializace. Při ní se předávají ovladači konfigurační parametry a ukazatele na paměť pro fronty používané ovladačem, uložené ve struktuře *I82559_DVR_INIT_T*:

- `frameBuff` - Ukazatele na paměti pro fronty pro Ethernetové rámce v modulu INTEL8255X včetně jejich velikostí. Velikost paměti pro rámce nesmí být moc malá, minimálně by se do ní měl vejít UDP paket o maximální velikosti.
- `udpBuff` - Ukazatele na paměti pro fronty pro UDP pakety v modulu NET_IP včetně jejich velikostí. Velikosti těchto oblastí musí být větší než maximální velikost UDP paketu.
- `dhcpClientEnable` – Povoluje funkci DHCP klient v síťovém ovladači. Jinak se použijí síťové parametry zadané dále v této struktuře.
- `inetAddr` – IP adresa počítače. Pro zadání této adresy slouží makro *IP4_ADDR*.
- `inetNetMask` – Maska sítě. Pro její zadání lze použít makro jako u `inetAddr`.
- `defGateway` – Adresa výchozí brány. Pokud není zadána, je zde nastavena 0.
- `mtu` – Maximální velikost dat v Ethernetovém rámci. Standardní hodnota je 1500 bajtů.
- `ttl` – Maximální životnost paketu v sekundách, resp. maximální počet průchodů paketu směrovačem. Tato hodnota se doplňuje do IP hlavičky.
- `macAddr` – MAC adresa síťového adaptéru. Pokud jsou zde zadány samé 0, použije se MAC adresa uložená v EEPROM síťového adaptéru.
- `fCpu` – Frekvence procesoru. Slouží k přepočtu hodnoty vrácené instrukcí RDTSC na sekundy.

- `displayNum` – Číslo servisní obrazovky. Pokud je zadána záporná hodnota, servisní obrazovka se nepoužívá.
- `mailId` – Číselné identifikátory zpráv, se kterými komunikuje ovladač s ostatními ovladači a aplikací prostřednictvím `SYS_MAIL`.

Během inicializace ovladače funkcí `INTEL82559_DRV_Init` se inicializují všechny jeho součásti a zaregistrují se příjemci zpráv z modulu jádra `SYS_MAIL`. Pak je ovladač připraven na zaregistrování v RT jádru pomocí funkce `INTEL82559_DRV_Register`. Po registraci je ovladač připraven na spuštění aplikace pod RT jádrem.

3.5.2 Řídící a stavové funkce ovladače

Následující funkce, vyjma inicializační, lze používat až po inicializaci ovladače. Deklarace publikované mimo ovladač mají předponu `INTEL82559_DRV_` u funkcí a `I82559_DRV_` u datových typů, aby jejich jména nekolidovaly s deklaracemi jiných ovladačů. Tato předpona není v následujícím textu uváděna.

*`bool_t Init (INIT_T *netParams)`*

Funkce provede inicializaci ovladače, tj. inicializuje všechny jeho moduly. Jako parametr je uvedena struktura popsaná výše v kapitole 3.5.1. Funkce vrací `TRUE`, pokud se inicializace zdařila. V opačném případě se inicializace nezdařila, například z důvodu absence síťového adaptéru v počítači.

`void Uninit()`

Funkce ukončí činnost síťového ovladače. Hardware síťového adaptéru je uveden do výchozího stavu.

`bool_t Register()`

Zaregistruje ovladač v RT jádru. Při registraci ovladače se v jádru zaregistruje výkonná funkce ovladače, kterou jádro periodicky volá.

*`bool_t SetFilter(uint8_t index, FILTER_T *filter)`*

Nastaví filtr přijímaných UDP paketů na základě zdrojové IP adresy, a cílové IP adresy a UDP portu. Lze uložit více filtrů, každý má svoji pozici danou parametrem `index`. Filtry jsou číslovány od 0. Pro zrušení filtru na dané pozici se funkce zavolá s parametrem `filter` rovno `NULL`. Pokud se aplikace pokusí nastavit filtr mimo rozsah 0 až `MAX_FLT_INDEX`, funkce vrátí `FALSE`.

STATE_T GetState()

Funkce vrací stav ovladače síťového adaptéru. Návrátové hodnoty mohou být:

- LINK_DOWN – Síťový adaptér nedetekuje linkový signál. Pravděpodobně adaptér není zapojen do sítě.
- READY – Ovladač má přidělenou IP adresu a je připraven na příjem a odesílání UDP paketů.
- NO_IP_ADDR – Nepodařilo se získat IP adresu od DHCP serveru, v tomto stavu ovladač nepřijímá ani neodesílá UDP pakety. Může jít jen o dočasný stav, než se IP adresu podaří získat.

3.5.3 Funkce pro práci s UDP pakety

Tyto funkce používá aplikace pro usnadnění práce s UDP pakety. Funkce pracují nad strukturou používanou při zasílání zpráv mezi ovladačem a aplikací.

*int32_t GetPacketLength(void *packet)*

Vrací celkovou velikost UDP paketu v bajtech, tj. velikost UDP hlavičky a dat.

*int32_t GetUDPDataLength(void *packet)*

Vrací velikost dat v UDP paketu v bajtech.

void GetUDPDataPtr(void *packet)*

Funkce vrací ukazatel na první bajt dat v UDP paketu.

*void ExchangeSrcDst(void *packet, uint16_t dataLength)*

Funkce pracuje nad přijatým paketem. Provede záměnu odesílatele a adresáta u MAC adres, IP adres a UDP portů. Zároveň nastaví velikost datové části UDP paketu.

*void SetDontFragment (void *packet)*

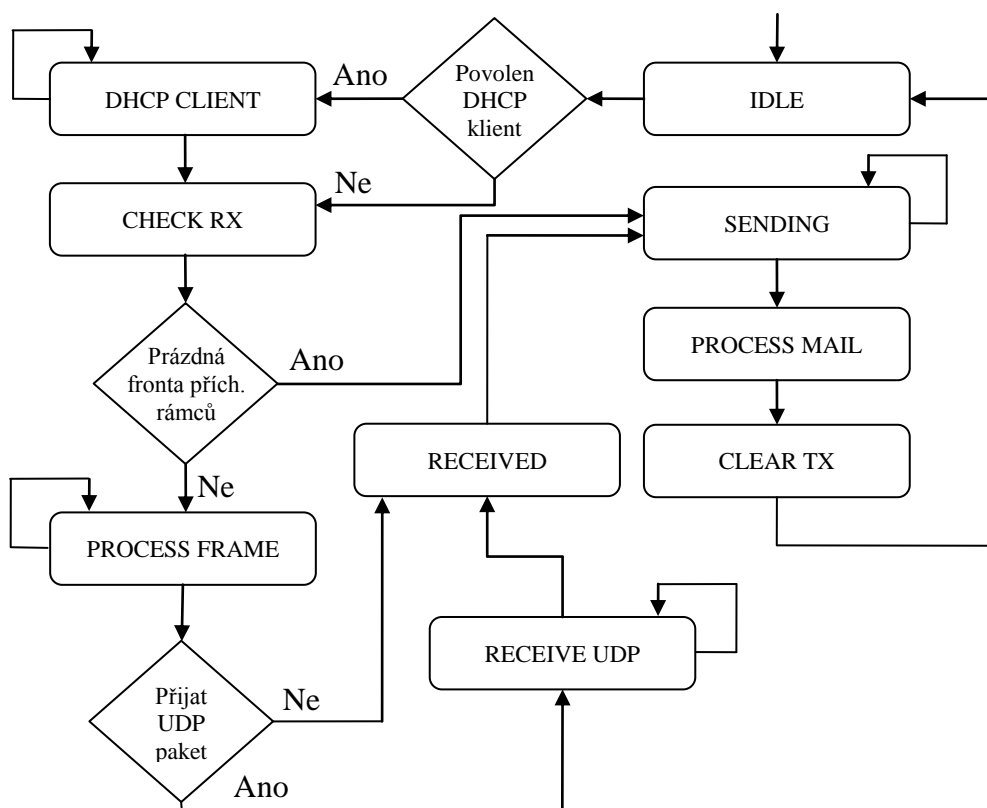
Funkce nastaví příznak nefragmentovat u IP hlavičky. Pokud je celková velikost IP paketu větší než nejnižší MTU na cestě paketu sítě, není takový paket doručen adresátovi. Na druhou stranu zaručuje, že paket nebude rozdělen do více fragmentů.

*void BuildUDPPacket(void *packet, INET_ADDR_T dstAddr, uint16_t dstPort, uint16_t dataLength)*

Funkce vytvoří hlavičky Ethernetového rámce, IP a UDP. Nastaví adresu cíle a velikost dat přenášených paketem.

3.5.4 Popis činnosti ovladače

Činnost ovladače je řízena jednou výkonnou funkcí, kterou periodicky volá RT jádro. Protože se musí minimalizovat čas strávený v této funkci, je funkce ovladače rozdělena do několika fází.



Obr. 3.4 - Vývojový diagram činnosti ovladače

Po inicializaci ovladače je výchozí stav IDLE. Pak se cyklicky prochází jednotlivými stavy v závislosti na stavu fronty příchozích Ethernetových rámců, rozpoznání protokolu v rámci a povolení funkce DHCP klient.

Protože je stavů v systému relativně hodně a u výkonné funkce se musí minimalizovat maximální čas jejího vykonání, je aktuální stav reprezentován jako odkaz na funkci, pro správu daného stavu. Každému stavu tedy odpovídá jedna funkce. Přejít mezi stavy je realizován změnou funkce, která se vykonává ve výkonné funkci ovladače. Tento princip poskytuje nejen konstantní časovou složitost v závislosti na počtu stavů, ale i jednodušší správu většího počtu stavů, než v případě použití větvení programu pomocí *switch*, protože je zde každý stav izolován ve své funkci.

- Stav IDLE – V tomto stavu se provádí aktualizace systémového času.
- Stav DHCP CLIENT – Zde se řídí komunikace s DHCP serverem. Realizuje odesílání a zpracování přijatých UDP paketů. Rozdělení na menší části je provedeno až v modulu DHCP klienta. Pro vytvoření, odeslání UDP paketu DHCP serveru anebo jeho přijetí je zapotřebí vícenásobného volání funkce v modulu NET_DHCP. Dokud není operace dokončena, tak ovladač zůstává v tomto stavu.
- Stav CHECK RX – Kontroluje, zda nejsou ve frontě příchozích Ethernetových rámců přijaté rámce.
- Stav PROCESS FRAME – Zjistí se typ protokolu přenášeného Ethernetovým rámcem. Pokud je to ARP, předá se ke zpracování ARP modulu. Pokud IP, ověří se IP hlavička a určí vyšší protokol, přenášený v paketu. Pokud rámec neobsahuje podporovaný protokol, je zahozen.
- Stav RECEIVE UDP – Voláním funkce *Ip_RecvUDPFragment*, se provede zařazení fragment IP paketu do fronty v modulu NET_IP. Funkce má vnitřní stav a může být vyžadováno opakované volání, než se podaří vložit IP fragment do fronty. V tomto stavu může ovladač zůstat po několik zavolání výkonné funkce.
- Stav RECEIVED – Provede uvolnění místa ve frontě příchozích Ethernetových rámců a připraví RFD na přijetí dalšího rámce.
- Stav SENDING – Volá se zde funkce *Ip_SendFrame* z modulu NET_IP. Tato funkce zajistí fragmentování a odesílání IP paketů umístěných ve frontě odchozích UDP paketů. Funkce se jako u stavu RECEIVE UDP volá opakovaně, dokud není operace dokončena. Pak teprve dochází k přesunu do dalšího stavu ovladače.
- Stav PROCESS MAIL – Zpracovává přijaté zprávy od *SYS_MAIL*. Jsou to zprávy o zpracování přijatého UDP paketu aplikací a požadavek na odeslání UDP paketu.

- Stav CLEAR TX – Aktualizuje stav fronty odchozích Ethernetových rámců podle reálně odeslaných rámců. Připraví místo nové místo pro odesílání dalších rámců.

3.6 Sestavení ovladače

Tvorba ovladače probíhá ve vývojovém prostředí Eclipse. Síťový ovladač tvoří samostatný projekt v Eclipse. Protože kompilátor Open Watcom není přímo podporován v Eclipse bylo nutné vytvořit *make* soubory, které popisují kompilátoru jak sestavit knihovnu ovladače. Samotná kompilace se spouští pomocí dávkového souboru *gen_drv.bat*, který lze spustit přímo z Eclipse. Výstup kompilátoru přebírá Eclipse a je v jeho prostředí je k dispozici jak celý výpis z kompilace, tak zformátované chybové zprávy.

Pro odstranění výsledků předchozích kompilací je vytvořen dávkový soubor *clean.bat*, který je také spustitelný přímo z Eclipse.

Součástí ovladače je informace o jeho verzi a čase sestavení. Tyto informace se předávají RT jádru při registraci ovladače a jsou pak dostupné na obrazovce při běhu systému s RT jádrem.

4 Testy ovladače

Pro testování síťového ovladače byly vytvořeny dvě pomocné aplikace. První aplikace má název NET_TEST a je to demonstrační aplikace běžící pod RT jádrem. Druhá aplikace s názvem UDPTest je pro OS Windows, a umožňuje odesílat a přijímat UDP pakety.

Při vývoji a pro ověření přenášených dat po síti byla použita aplikace Wireshark. Tato aplikace zaznamenává Ethernetové rámce přenášené zvoleným síťovým rozhraním. Umí pracovat s velkým množstvím síťových protokolů a umožňuje detailní analýzu přenášených rámců.

4.1 Demonstrační aplikace UDPTest

Aplikace komunikuje protokolem UDP s aplikací NET_TEST a používá pakety definované v Tab. 4.1. Je vytvořena ve vývojovém prostředí QtCreator verze 3.4.0. Toto prostředí bylo zvoleno proto, že díky knihovně Qt lze snadno vytvořit požadovanou aplikaci s grafickým uživatelským rozhraním. Knihovny Qt jsou nezávislé na operačním systému a lze tak aplikaci zkompileovat i pro jiné OS než Windows.

Offset	Popis
0	Pořadové číslo zprávy. Slouží ke spárování odeslaného a přijatého UDP paketu v aplikaci UDPTest.
4	Pozice textové zprávy v datech UDP paketu.
8	Délka textové zprávy v UDP paketu.
12	Výplň paketu před zprávou.
Pozice textové zprávy.	Textová zpráva.
	Výplň UDP paketu za zprávou.

Tab. 4.1 - Formát UDP paketu testovacích aplikací

Program UDPTest vytváří UDP pakety zadané délky a vkládá do nich zprávu zadanou uživatelem. Textová zpráva je vložena na náhodnou pozici v datech UDP paketu. Po odeslání UDP paketu se čeká na odpověď od aplikace NET_TEST. Když odpověď dorazí je zobrazen časový rozdíl mezi odesláním a přijetím UDP paketu. Odesílaný a přijatý UDP paket musí mít stejné pořadové číslo.

Vzhled a popis aplikace UDPTest je v příloze D.

4.2 Demonstrační aplikace NET_TEST

Demonstrační aplikace se skládá ze síťového ovladače a testovacího ovladače. Oba ovladače jsou zaregistrované v RT jádru a jádro periodicky volá jejich výkonné funkce. Aplikace očekává data v UDP paketu ve formátu podle Tab. 4.1.

Výplň před a za zprávou je v paketu obsažena, jen pokud je potřeba. Po přijetí UDP paketu je zobrazena zpráva a pořadové číslo paketu na obrazovce aplikace a UDP paket je pak odeslán zpět odesílateli.

Výkonné funkce v ovladačích se volají jednou za periodu simulace hlavní smyčky RT jádra. Tato perioda je v aplikaci nastavena na 500 μ s. Na vyzvednutí, zpracování a odeslání UDP paketu je potřeba vícero zavolání výkonných funkcí v závislosti na velikosti UDP paketu. Proto odeslání odpovědi trvá delší čas, než by bylo potřeba, kdyby mezi voláním výkonných funkcí nebyla časová prodleva.

Vzhled obrazovky aplikace NET_TEST je v příloze C.

4.3 Testy komunikace pomocí UDP

První test spočívá v příjmu a odeslání jednoho nefragmentovaného UDP paketu. Tj. takového, že velikost celého IP paketu nepřesáhne hodnotu MTU. UDP paket zachycený pomocí aplikace Wireshark je na obrázku Obr. 4.1.

No.	Time	Source	Destination	Protocol	Length	Info
67	45.890492000	CadmusCo_41:4f:06	Broadcast	ARP	42	Who has 192.168.16.89? Tell 192.168.16.86
68	45.893805000	RtdEmbed_03:1f:2e	CadmusCo_41:4f:06	ARP	60	192.168.16.89 is at 00:d0:81:03:1f:2e
69	45.894052000	192.168.16.86	192.168.16.89	UDP	154	Source port: 12036 Destination port: 12035
70	45.907111000	192.168.16.89	192.168.16.86	UDP	154	Source port: 12035 Destination port: 12036

Obr. 4.1 - Nefragmentovaný UDP paket

Při odeslání prvního UDP paketu, nejprve proběhne ARP dotaz na MAC adresu patřící k IP adrese 192.168.16.89. Poté, co síťový ovladač odpoví, dochází k samotnému přenosu UDP paketu. V opačném směru už ARP dotaz není potřeba, protože MAC adresa je uložena v ARP tabulce. Síťový ovladač ji tam zapsal při příjmu UDP paketu.

Druhý test ukazuje příjem a odeslání UDP paketu, který bylo nutné fragmentovat. V aplikaci UDPTTest byla nastavena velikost paketu na 3500 bajtů. UDP paket byl při odesílání rozdělen na 3 IP fragmenty. MTU je u síťového rozhraní na straně programu UDPTTest nastaveno na 1500 bajtů. Na straně aplikace NET_TEST je v síťovém ovladači nastavena hodnota MTU na 1492 bajtů. Zaznamenané Ethernetové rámce z této komunikace jsou na Obr. 4.2.

Je zde vidět rozdělení UDP paketů v obou směrech na 3 fragmenty. Při odesílání má Ethernetový rámec maximální velikost 1514 bajtů. Tj. hodnota MTU zvětšená o velikost hlavičky Ethernetového rámce.

No.	Time	Source	Destination	Protocol	Length	Info
3	1.403469000	192.168.16.86	192.168.16.89	IPv4	1514	Fragmented IP protocol (proto=UDP 17, off=0, ID=28bf) [Reassembled in #5]
4	1.403556000	192.168.16.86	192.168.16.89	IPv4	1514	Fragmented IP protocol (proto=UDP 17, off=1480, ID=28bf) [Reassembled in #5]
5	1.403646000	192.168.16.86	192.168.16.89	UDP	594	Source port: 12036 Destination port: 12035
6	1.426856000	192.168.16.89	192.168.16.86	IPv4	1506	Fragmented IP protocol (proto=UDP 17, off=0, ID=0700) [Reassembled in #8]
7	1.427362000	192.168.16.89	192.168.16.86	IPv4	1506	Fragmented IP protocol (proto=UDP 17, off=1472, ID=0700) [Reassembled in #8]
8	1.427707000	192.168.16.89	192.168.16.86	UDP	610	Source port: 12035 Destination port: 12036


```

Internet Protocol Version 4, Src: 192.168.16.89 (192.168.16.89), Dst: 192.168.16.86 (192.168.16.86)
  Version: 4
  Header length: 20 bytes
  Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
  Total Length: 1492
  Identification: 0x0700 (1792)
  Flags: 0x01 (More Fragments)
  Fragment offset: 1472
  Time to live: 16
  Protocol: UDP (17)
  Header checksum: 0xdb61 [correct]
  Source: 192.168.16.89 (192.168.16.89)
  Destination: 192.168.16.86 (192.168.16.86)
  [Source GeoIP: Unknown]
  [Destination GeoIP: Unknown]
  Reassembled IPv4 in frame: 8
  Data (1472 bytes)

```

Obr. 4.2 - Fragmentovaný UDP paket

V opačném směru má zachycený rámec maximální velikost 1506 bajtů, což odpovídá nastavené hodnotě MTU.

O správném přenosu UDP paketů svědčí i správné zobrazení zprávy na obrazovce aplikace NET_TEST a zobrazení času přijetí odpovědi v aplikaci UDPTTest.

Na obrazovce NET_TEST je vidět i maximální čas strávený uvnitř výkonné funkce síťového ovladače. Tento čas se pohybuje v rozmezí 11 až 25 μ s, v závislosti na maximální velikosti přenášeného paketu.

Při zjišťování maximální časové náročnosti byl testován přístup k síťovému adaptéru pomocí I/O i registrů mapovaných do paměti. Nebyl zjištěn významný rozdíl mezi oběma přístupy, časová náročnost byla v průměrném případě u obou stejná.

4.4 Testy DHCP klienta

Pokud je v programu NET_TEST zapnuta funkce DHCP klient, tak se jeho startu pokouší získat IP adresu od DHCP serveru. Záznam takové komunikace je na Obr. 4.3. Je vidět, že probíhá podle specifikace, jak je uvedeno v kapitole 2.6.5 Protokol DHCP. Komunikace začíná tím, že síťový ovladač odešle DISCOVER, na tuto výzvu reaguje DHCP server odpovědí OFFER. Pak síťový ovladač zažádá o nabídnutou IP adresu zprávou REQUEST

a nakonec DHCP server potvrdí přidělení této adresy zprávou ACK. Dokud není IP adresa přidělena, odesílá síťový ovladač UDP pakety DHCP serveru jako broadcast.

No.	Time	Source	Destination	Protocol	Length	Info
15	15.870595000	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x567acf6a
16	15.871627000	192.168.16.1	192.168.16.89	DHCP	364	DHCP Offer - Transaction ID 0x567acf6a
17	15.884725000	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x567acf6a
18	15.885528000	192.168.16.1	192.168.16.89	DHCP	364	DHCP ACK - Transaction ID 0x567acf6a


```

Seconds elapsed: 0
▷ Bootp flags: 0x0000 (Unicast)
  Client IP address: 0.0.0.0 (0.0.0.0)
  Your (client) IP address: 0.0.0.0 (0.0.0.0)
  Next server IP address: 0.0.0.0 (0.0.0.0)
  Relay agent IP address: 0.0.0.0 (0.0.0.0)
  Client MAC address: RtdEmbed_03:1f:2e (00:d0:81:03:1f:2e)
  Client hardware address padding: 00000000000000000000
  Server host name not given
  Boot file name not given
  Magic cookie: DHCP
▷ Option: (53) DHCP Message Type
▷ Option: (54) DHCP Server Identifier
▽ Option: (50) Requested IP Address
  Length: 4
  Requested IP Address: 192.168.16.89 (192.168.16.89)
▷ Option: (55) Parameter Request List
▷ Option: (255) End
  Padding

```

Obr. 4.3 - Záznam DHCP komunikace

Po přidělení IP adresy je vidět v záznamech DHCP serveru MAC adresa 00:d0:81:03:1f:2e a k ní přidělená IP adresa 192.168.16.89. Což ukazuje, že přidělení IP adresy proběhlo opravdu správně.

Záznam z aplikace Wireshark ukazuje použité *Options* při odesílání zprávy REQUEST. Je zde uveden typ DHCP zprávy (REQUEST), identifikátor DHCP serveru (obsahuje IP adresu serveru), požadovanou IP adresu a seznam dalších požadovaných parametrů.

Žádost o obnovení výpůjčky IP adresy probíhá stejně jako žádost o její přidělení a proto zde není uveden záznam z této komunikace.

5 Závěr

Síťový ovladač v real-time systému je komplexní subsystém pracující s hardware síťové karty a síťovými protokoly. Proto, abych mohl vytvořit tento ovladač, jsem nastudoval specifiky tvorby software v real-time systémech, dokumentaci k síťovému adaptéru Intel 82559-ER, specifikace síťových protokolů a RT jádro vytvořené R. Martínkem.

Vytvořil jsem síťový ovladač, který umožňuje aplikaci pod RT jádrem komunikaci prostřednictvím protokolu UDP. V ovladači je možné zapnout funkci DHCP klient, pro automatické přidělení IP adresy od DHCP serveru, jinak musí být nastaveny parametry sítě při inicializaci ovladače. Pro snadnější diagnostiku a přehled o práci ovladače, může ovladač zobrazovat statistické údaje o přenesených Ethernetových rámcích a aktuálním nastavení IP adresy na obrazovku aplikace.

Rozdělením práce ovladače do menších úseků se podařilo dosáhnout maximální doby běhu výkonné funkce ovladače 25 μ s. Tím je zaručeno, že síťový ovladač neovlivní běh hard real-time úlohy pod RT jádrem, pokud tato úloha poskytne ovladači minimálně 25 μ s dlouhé časové intervaly pro jeho provedení. Činnost samotného síťového ovladače není hard real-time úlohou, přednost má aplikace spuštěná pod RT jádrem.

Síťový ovladač podporuje jen IP protokol verze 4. Podpora pro IPv6 není v ovladači implementována. Toto ale nepředstavuje problém, protože i když podporu IPv6 už obsahuje většina nových zařízení a počítačů, tak podpora pro IPv4 je u nich stále přítomná.

Zlepšení výkonnosti síťového ovladače by bylo možné dosáhnout úpravou RT jádra tak, aby se výkonná funkce síťového ovladače volala častěji. Například ve stejném bloku jako se volají servisní funkce jádra. Aktuálně se volá jednou za periodu RT smyčky a nedocíluje se tak velké datové prostupnosti a tak rychlé odezvy, jakých by se dalo dosáhnout v opačném případě.

6 Literatura

- [1] Martínek, R.: RT rozšíření OS DOS pro platformu PC104. Diplomová práce ČVUT, Praha 2013
- [2] Hermann Kopetz: Real-Time Systems, 2011, ISBN 978-1-4419-8236-0
- [3] Intel: 82559ER Fast Ethernet PCI Controller Network Silicon Datasheet, Revision 1.8, October 2006
- [4] Intel: 8255x 10/100 Mbps Ethernet Controller Family, Open Source Software Development Manual, Revision 1.3, January 2006
- [5] RFC 791 – Internet Protocol, September 1981, [cit. 11.5.2015], <http://tools.ietf.org/html/rfc791>
- [6] Příspěvatelé Wikipedie: IPv4, [cit. 7.5.2015], <http://en.wikipedia.org/wiki/IPv4>
- [7] RFC 768 – User Datagram Protocol, 28 August 1980, [cit. 11.5.2015], <http://tools.ietf.org/html/rfc768>
- [8] RFC 826 – An Ethernet Address Resolution Protocol, November 1980, [cit. 11.5.2015], <http://tools.ietf.org/html/rfc826>
- [9] RFC 2131 – Dynamic Host Configuration Protocol, March 1997, [cit. 11.5.2015], <http://tools.ietf.org/html/rfc2131>
- [10] RFC 815 – IP Datagram Reassembly Algorithms, July 1982, [cit. 11.5.2015], <http://tools.ietf.org/html/rfc815>

A Seznam zkratek

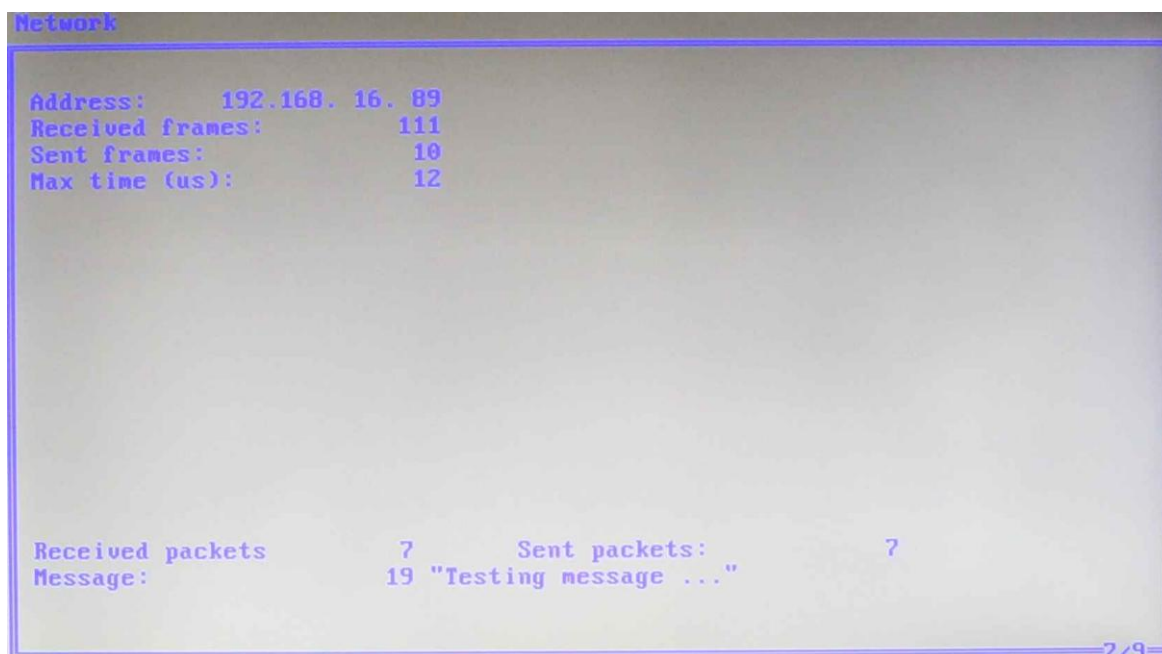
ARP-	Address Resolution Protocol
BIOS	Basic Input-Output System
CB	Command Block
CBL	Command Block List
CSR	Command / Status Register
CRC	Cyclic Redundancy Check
CU	Command Unit
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name Server
EEPROM	Electrically Erasable Programmable Read-Only Memory
FIFO	First In First Out
IANA	Internet Assigned Numbers Authority
RFC	Request for Comments
ICMP	Internet Control Message Protocol
IP	Internet Protocol
ISO/OSI	International Organization for Standardization / Open Systems Interconnection
MAC	Media Access Control
MII	Media Independent Interface
MTU	Maximum Transmission Unit
NTP	Network Time Protocol

OS	Operating System
PCI	Peripheral Component Interconnect
PHY	Physical Layer
RFA	Receive Frame Area
RFD	Receive Frame Descriptor
RT	Real Time
RTC	Real Time Clock
RU	Receive Unit
SCR	Status and Control Registers
SCB	System Control Block
TCP	Transmission Control Protocol
TSC	Time Stamp Counter
TTL	Time To Leave
UDP	User Datagram Protocol

B Obsah přiloženého CD

- readme.txt - popis obsahu CD
- text
 - dp_cvekm1.pdf – Text diplomové práce ve formátu PDF
 - dp_cvekm1.docx – Text diplomové práce ve formátu DOCX
- exe
 - UDPTest – Aplikace UDPTest.exe
 - NET_TEST.exe – Testovací aplikace
 - INTEL82559_DRV – Binární knihovna ovladače s hlavičkovým souborem
- src
 - UDPTest – Zdrojový kód aplikace UDPTest
 - INTEL82559 – Zdrojový kód testovací aplikace

C Obrazovka aplikace NET_TEST



```
Network
Address:      192.168. 16. 89
Received frames:  111
Sent frames:    10
Max time (us):   12

Received packets      7      Sent packets:      7
Message:             19 "Testing message ..."
```

The screenshot shows a window titled "Network" with a blue border. The text is displayed in a monospaced font. The top section shows network statistics: Address (192.168. 16. 89), Received frames (111), Sent frames (10), and Max time (us) (12). The bottom section shows packet statistics: Received packets (7), Sent packets (7), and a message (19 "Testing message ..."). A small "7/9" indicator is visible in the bottom right corner of the window.

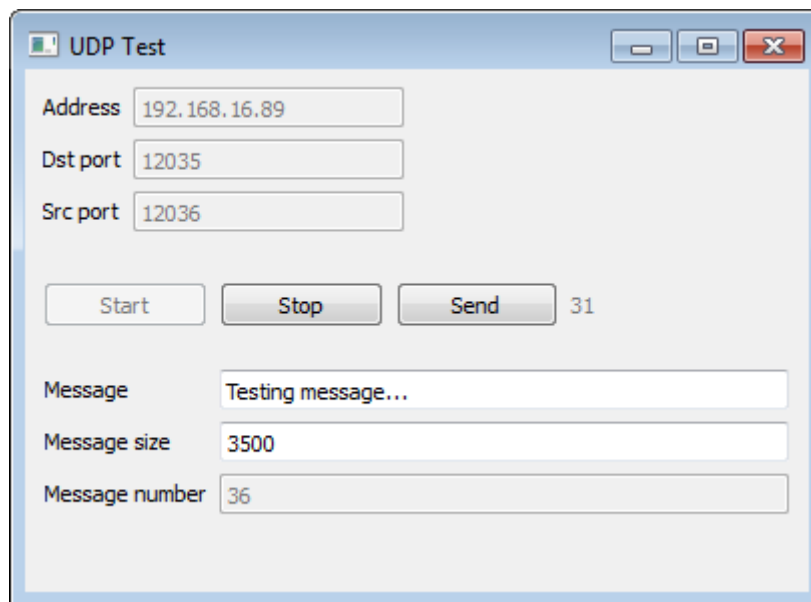
Obr. C.1 - Snímek aplikace NET_TEST

V horní části obrazovky je zobrazena aktuální IP adresa. Následují počty přijatých a odeslaných Ethernetových rámců. Včetně těch co se z nějakého důvodu zahodí.

V dolní části se zobrazují údaje od testovacího ovladače.

Je zde uveden počet přijatých a odeslaných UDP paketů a je zobrazena poslední přijatá zpráva, včetně jejího pořadového čísla.

D Aplikace UDPTTest



Obr. D.2 - Aplikace UDPTTest

V horní části aplikace se nastavuje IP adresa, zdrojový a cílový UDP portu. Změna těchto údajů je možná jen dokud se nestiskne Start a aplikace neotevře UDP port pro naslouchání.

Po stisku tlačítka Start lze odeslat zprávu, zadanou níže. Lze zadat celkovou velikost dat v UDP paketu. Textová zpráva se umístí na náhodnou pozici v datové oblasti paketu. Pokud je velikost paketu příliš malá, aplikace ji upraví tak, aby bylo možné odeslat textovou zprávu.

Tlačítkem Send se zpráva odesílá, vedle tlačítka se zobrazí čas mezi odesláním a přijetím dané zprávy v milisekundách.

Pro úpravu nastavení aplikace po stisku Start, lze použít tlačítko Stop, které opět zpřístupní nastavení aplikace.

V dolní části je uvedeno pořadové číslo poslední odeslané zprávy.

