ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ
FAKULTA ELEKTROTECHNICKÁ
KATEDRA ŘÍDICÍ TECHNIKY

DIPLOMOVÁ PRÁCE

EtherNet/IP a ControlNet

2003 Jakub Kožnar

Prohlášení

Prohlašuji, že sem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW, atd.) uvedené v přiloženém 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 22.1.03 Jakub Kožnar

České vysoké učení technické v Praze – Fakulta elektrotechnická

Katedra řídicí techniky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: Jakub Kožnar

Obor: Technická kybernetika

Název tématu: EtherNet/IP a ControlNet

Zásady pro vypracování:

1. Seznamte se s automaty PLC5, SLC500 a ControlLogix a jejich komunikačními linkami, speciálně se zaměřte na vlastnosti linek ControlNet a EtherNet/IP. Porovnejte možnosti jejich využití a otestujte jejich chování v reálných aplikacích.
2. Podílejte se též na dobudování modelů v laboratoři Allen – Bradley na katedře řídicí techniky a zpracování návodů pro výuku se zaměřením na využití zmíněných komunikací
3. Zpracujte základní výukovou dokumentaci i v angličtině včetně webové prezentace. Podílejte se na zpracování české a anglické dokumentace k dalším úlohám a jejich prezentaci pomocí webu.

Seznam odborné literatury: Dodá vedoucí práce.

Vedoucí diplomové práce: Ing. Jindřich Fuka

Datum zadání diplomové práce: prosinec 2001

Termín odevzdání diplomové práce: leden 2003

Abstrakt

Společností Allen – Bradley vyvinutá síťová architektura NetLinx se skládá ze tří průmyslových sítí. Nejnižší vrstvu tvoří síť DeviceNet, která nahrazuje starší síť Remote I/O. Střední vrstvu tvoří moderní síť ControlNet, která plní funkce starších sítí DH+ a DH485 a zajišťuje deterministický přenos dat. Nejvyšší vrstvu tvoří síť EtherNet/IP. Sítě DeviceNet, ControlNet a EtherNet/IP používají stejný komunikační protokol – Control and Information Protocol (CIP). Tato práce se v první části zabývá popisem implementace protokolu CIP pro standardní síť ethernet a popisem fyzické vrstvy sítě EtherNet/IP vhodné pro použití v průmyslovém prostředí. Dále je zde uvedena analýza některých typů komunikace na síti EtherNet/IP pro demonstraci předcházejících kapitol. Součástí této práce je také porovnání sítí EtherNet/IP a ControlNet, možnostmi jejich využití a zvláště srovnáním a ověřením jejich výkonnosti.

Abstract

Rockwell Automation (Allen-Bradley) developed NetLinx network architecture, which consists of three industrial networks. The DeviceNet network is found in the lowest layer; it has replaced the older industrial network Remote I/O. ControlNet forms the middle layer of this architecture by substitution for older DH+ and DH485 networks; ControlNet is a deterministic network for reliable data transfers. Finally the highest layer consists of an EtherNet/IP network. These three networks use a common communication protocol – Control and Information Protocol (CIP). The first part of this diploma thesis comprises a description of CIP implementation for a standard ethernet network and a description of the physical layer of the EtherNet/IP network suitable for use in an industrial environment. Additionally, there is an analysis of a few types of communication used on EtherNet/IP networks to assist in demonstrating claims in preceeding chapters. This thesis also considers EtherNet/IP and ControlNet networks with a focus on performance comparison and verification.

Poděkování

Na tomto místě bych rád vyjádřil díky vedoucímu diplomové práce Ing. J. Fukovi za jeho rady, které mě při práci inspirovaly. Dále děkuji panu Ing. Z. Blažkovi, CSc. za zapůjčení výpočetní techniky, bez níž by nebylo možné tuto práci realizovat.

Obsah

Abstrakt ..... **iv**

Abstract ..... **iv**

Poděkování ..... **v**

Obsah ..... **vi**

Seznam obrázků ..... **viii**

Seznam rovnic ..... **viii**

Seznam tabulek ..... **ix**

Úvod **1**

1 Architektura NetLinx **2**
1.1 Ethernet/IP 2
1.2 ControlNet 3
1.3 DeviceNet 6

2 Základní popis protokolu CIP **9**
2.1 Úvod 9
2.2 Objektový popis 9
2.3 Elektronické datové listy (Electronic data sheets) 13
2.4 Typy spojení 14
2.5 Transportní třídy protokolu CIP 17
2.6 Objekt manažer spojení (Connection Manager Object) 17

3 Popis sítě EtherNet/IP **19**
3.1 Úvod 19
3.2 Síťové modely ISO-OSI a TCP/IP 20
3.3 Porovnání síťových modelů ControlNet a EtherNet/IP 22
3.4 Zapouzdření protokolu CIP do TCP/UDP paketů 22
3.5 Zapouzdření paketů protokolu CIP 26
3.6 Popis příkazů 29
3.7 Management spojení (session management) 35
3.8 Common Packet Format (CPF – společný formát paketů) 36
3.9 Mapování explicitních a implicitních (I/O) zpráv na TCP/IP 38
3.10 Síťový identifikátor spojení 40
3.11 Založení spojení ve transportní třídě 0 a 1 41
3.12 Connected data v transportní třídě 0 a 1 43
3.13 Třídění přicházejících I/O dat (connected data) 44
3.14 Dočasný multicastový obzor (multicast scoping strategy) 44
3.15 Dočasná alokační strategie 44
3.16 Závěr 45

4 Fyzická vrstva sítě EtherNet/IP **46**
4.1 Úvod 46

4.2 Komerčně používaná média 46
4.3 Průmyslové médium EtherNet/IP 46
4.4 Průmyslové konektory RJ-45 47
4.5 Propojení stínění se zemí 49
4.6 Použití optických kabelů 50
4.7 Závěr 50

5 Analýza některých typů komunikace na síti EtherNet/IP 52
5.1 Úvod do problematiky monitorování sítě 52
5.2 Program CIPAlyzer 3.0 53
5.3 Program Sniffer Pro 4.7 54
5.4 Monitorování provozu na síti EtherNet/IP 55

6 EtherNet/IP a ControlNet 61
6.1 Navrhování sítě EtherNet/IP s ohledem na maximální šířku pásma 61
6.2 Porovnání sítí EtherNet/IP a ControlNet 62

7 Závěr a zhodnocení 67

Seznam zkratek a definic 68

Seznam použité literatury 71

Seznam použitého software 72

Příloha A 73
A.1 Detailní rozbor explicitní komunikace 73
A.2 Detailní rozbor implicitní komunikace 83

Příloha B 96
B.1 Požadavek UCMM 96
B.2 Odezva UCMM 97

Příloha C 98
C.1 Elektromechanické vlastnosti průmyslových EtherNet/IP kabelů 98

Seznam obrázků

Obrázek 1.1 Síťová architektura NetLinx 2
Obrázek 1.2 Mechanismus přístupu na sběrnici ControlNet 4
Obrázek 1.3 Plánované přenosy na síti ControlNet 5
Obrázek 1.4 Neplánované přenosy na síti ControlNet 5
Obrázek 1.5 Závislost přenosové rychlosti sběrnice CAN na její délce 6
Obrázek 1.6 Přístup na sběrnici CAN 7
Obrázek 1.7 Úrovně na sběrnici CAN 7
Obrázek 2.1 Komunikační síť založená na protokolu CIP 9
Obrázek 2.2 Model skutečného zařízení (motoru) 10
Obrázek 2.3 Adresování v protokolu CIP 11
Obrázek 2.4 Objektový model EtherNet/IP zařízení 13
Obrázek 2.5 Objektový model komunikačního CIP modulu 16
Obrázek 3.1 Základní přehled o protokolu CIP 19
Obrázek 3.2 ISO-OSI síťový model 20
Obrázek 3.3 Porovnání síťových modelů TCP/IP a ISO-OSI 21
Obrázek 4.1 Útlum průmyslové modifikace UTP kabelu 47
Obrázek 4.2 Upravená zásuvka RJ45 pro průmyslové prostředí 48
Obrázek 4.3 Upravený konektor RJ45 pro průmyslové prostředí 48
Obrázek 4.4 Schéma spojení stínění a zemnění 49
Obrázek 4.5 Komunikační deska 49
Obrázek 4.6 Schéma připojení stínění STP kabelu 50
Obrázek 5.1 Schéma sítě 52
Obrázek 5.2 Okno programu CIPAlyzer 54
Obrázek 5.3 Okno programu Sniffer Pro 55
Obrázek 5.4 Nastavení instrukce MSG 56
Obrázek 5.5 Komunikační cesta v instrukci MSG 57
Obrázek 5.6 Komunikace při zaslání zprávy CIP Data Table Read 58
Obrázek 5.7 I/O konfigurace pro implicitní komunikaci 59
Obrázek 5.8 Konfigurace lokálního modulu 1756-ENET/B 59
Obrázek 5.9 Vytvoření přímého I/O spojení a následná I/O komunikace 60
Obrázek 6.1 Měření zatížení programem Sniffer Pro 63
Obrázek 6.2 Testovaná I/O konfigurace 63
Obrázek 6.3 Chybové hlášení při překročení maximální šířky pásma modulu 64
Obrázek 6.4 Konfigurace sítě ControlNet 65

Seznam rovnic

Rovnice 4.1 Útlum UTP kabelu 47
Rovnice 6.1 Výpočet šířky pásma při připojení vzdálených analogových vstupů a výstupů 62

Seznam tabulek

Tabulka 1.1 Přenosové rychlosti sběrnice CAN v závislost na délce sběrnice 6
Tabulka 2.1 Příklad veřejně definovaného objektu 10
Tabulka 2.2 Transportní třídy v protokolu CIP 17
Tabulka 3.1 Začlenění protokolu CIP do ISO-OSI síťového modelu 21
Tabulka 3.2 Porovnání síťových modelů ControlNet a EtherNet/IP 22
Tabulka 3.3 Struktura IP datagramu (verze 4) 23
Tabulka 3.4 Struktura IP datagramu (verze 6) 24
Tabulka 3.5 IGMP paket 24
Tabulka 3.6 Zapouzdření TCP v IP datagramu na fyzické síti 25
Tabulka 3.7 Hlavička TCP paketu 25
Tabulka 3.8 Zapouzdření UDP v IP datagramu na fyzické síti 26
Tabulka 3.9 Hlavička UDP paketu 26
Tabulka 3.10 Struktura zapouzdřeného CIP paketu 27
Tabulka 3.11 Kódování pole příkaz 28
Tabulka 3.12 Kódování pole Status 29
Tabulka 3.13 Příkaz ListIdentity (požadavek) 30
Tabulka 3.14 Příkaz ListIdentity (odpověď) 30
Tabulka 3.15 Příkaz ListInterfaces (požadavek) 30
Tabulka 3.16 Příkaz ListInterfaces (odpověď) 31
Tabulka 3.17 Příkaz RegisterSession (požadavek) 31
Tabulka 3.18 Příkaz RegisterSession (odpověď) 31
Tabulka 3.19 Příkaz UnRegisterSession (požadavek) 32
Tabulka 3.20 Příkaz ListServices (požadavek) 32
Tabulka 3.21 Příkaz ListServices (odpověď) 32
Tabulka 3.22 Kódování příznaků Capability flags 33
Tabulka 3.23 Příkaz SendRRData (požadavek) 33
Tabulka 3.24 Příkaz SendRRData (odpověď) 34
Tabulka 3.25 Příkaz SendUnitData 34
Tabulka 3.26 Základní pohled na Common Packet Format 36
Tabulka 3.27 Struktura polí Address a Data item 36
Tabulka 3.28 Kódování pole Item ID 37
Tabulka 3.29 Pole adresa pro connected messages 37
Tabulka 3.30 Sequenced address 37
Tabulka 3.31 Formát pole dat (explicitní zpráva) 38
Tabulka 3.32 Formát pole dat (implicitní zpráva) 38
Tabulka 3.33 Kódování položky Sockaddr_info 38
Tabulka 3.34 Datový formát transportních tříd 0 a 1 43
Tabulka 5.1 Komunikační cesta v instrukci MSG 56
Tabulka 6.1 Výpočet šířky přenosového pásma pro jednotlivé typy komunikace 61
Tabulka 6.2 Výsledky testování sítí EtherNet/IP a ControlNet 65
Tabulka 7.1 Navázání přímého I/O spojení 83
Tabulka 7.2 Implicitní I/O komunikace 83
Tabulka B.1 Požadavek UCMM 96
Tabulka B.2 Odpvěd UCMM 97
Tabulka C.1 Elektromechanické vlastnosti EtherNet/IP kabelů pro průmysl 98
Tabulka C.2 Elektrické vlastnosti EtherNet/IP kabelů pro průmysl 99

Úvod

V současné době je v oboru průmyslové automatizace zřetelný trend využívat v distribuovaných řídicích systémech komunikační sítě založené na tradiční síti ethernet. Průmyslová síť EtherNet/IP představená společností ODVA/CI je implementací protokolu CIP (Control and Information Protocol) pro standardní síť ethernet běžně používanou v kancelářských prostředích.

V první kapitole je uveden popis síťové architektury NetLinx se zaměřením na jednotlivé části této architektury – sítě EtherNet/IP, ControlNet a DeviceNet. U každé z těchto sítí je uvedena stručná charakteristika včetně popisu typů přenosů a metod přístupu na sběrnici. Druhá kapitola popisuje základy objektově orientovaného protokolu CIP, který je používán síťovou architekturou NetLinx. Jsou zde vysvětleny pojmy explicitní a implicitní komunikace. Kapitola třetí se zabývá v první řadě síťovými modely ISO-OSI a TCP/IP a stručným popisem protokolů používaných při zapouzdření protokolu CIP do protokolů TCP/IP. Další část této kapitoly popisuje způsob, jakým jsou pakety protokolu CIP zapouzdřovány do protokolů nižších vrstev. Nakonec se zde nachází popis problematiky IP multicastu, která je využívaná přímými I/O spojeními. Čtvrtá kapitola popisuje fyzickou vrstvu sítě EtherNet/IP vhodnou pro použití v průmyslovém prostředí včetně popisu rozdílů vůči standardnímu ethernetu používanému v kancelářském prostředí.

Pro demonstraci faktů uvedených v kapitole třetí je v páté kapitole popis provedené analýzy některých typů komunikace na síti EtherNet/IP. Mimo samotný rozbor některých typů komunikace je zde popsána problematika monitorování sítí a popis použitých aplikací. V kapitole šesté jsou srovnány sítě ControlNet a EtherNet/IP se zaměřením na jejich výkonnost a zatížitelnost. Tato kapitola obsahuje část zaměřenou na navrhování sítí EtherNet/IP s ohledem na maximální šířku potřebného pásma pro jednotlivé typy implicitních spojení. Na závěr je provedeno zhodnocení testovacích experimentů pro porovnání zatížitelnosti sítí ControlNet a EtherNet/IP. Takovéto porovnání nebylo dosud publikováno.

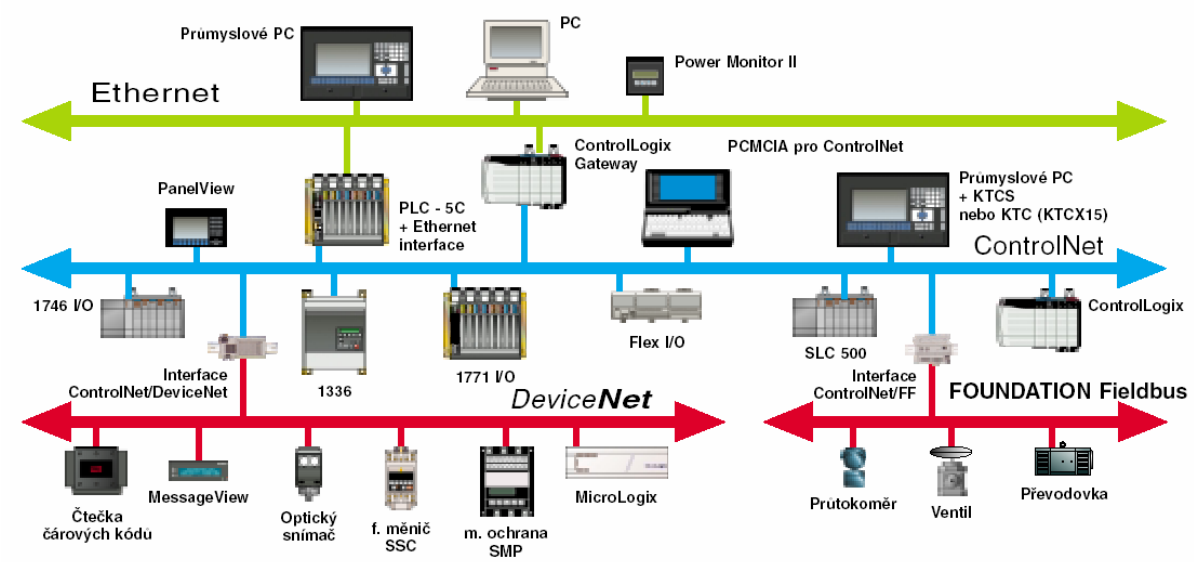
Jedním z velice důležitých bodů při plánování sítě EtherNet/IP je oddělení podsítí výrobního procesu (kde je síť ethernet použita pro řídicí účely, *plant floor*) a managementu (kde je síť ethernet použita pro dohled nad výrobou a sběr informací z výroby). Toto oddělení je nutné z bezpečnostních důvodů. Při některých operacích údržby sítě ethernet prováděných personálem IT by totiž mohlo dojít např. ke chvilkovému přetížení sítě (např. šířením požadavků vysílaných všesměrovým vysíláním) a tím k ohrožení přenosu real-time dat. Třetí kapitola obsahuje informace, které mohou být využity právě při oddělení těchto dvou podsítí pomocí zařízení typu *firewall*.

U některých výrazů, které se v textu vyskytují, bylo pro větší přehlednost vhodnější je ponechat v původním anglickém znění. Tyto výrazy jsou označeny kurzívou. Seznam zkratek a definic je uveden za poslední kapitolou.

1 Architektura NetLinx

Společností Rockwell Automation (Allen-Bradley) vyvinutá síťová architektura NetLinx se skládá z několika průmyslových sítí. Nejnižší vrstvu tvoří síť DeviceNet, která nahrazuje síť Remote I/O. Střední vrstvu tvoří moderní síť ControlNet, která podstatně rozšiřuje funkci starších sítí DH+ a DH485 a zajišťuje deterministický a uživatelem plánovatelný přenos dat. Nejvyšší vrstvu tvoří síť Ethernet/IP. Sítě DeviceNet, ControlNet a EtherNet/IP používají stejný komunikační protokol – Control and Information Protocol.

Sítě Remote I/O, DH+ a DH485 jsou původní sítě vyvinuté společností Allen – Bradley. Komunikační sítě DeviceNet a ControlNet byly vyvinuty firmou Rockwell Automation, nyní jsou však vlastněny a spravovány organizacemi ODVA (Open DeviceNet Vendor Association) a CI (ControlNet International). Tyto organizace také představily síť EtherNet/IP, která je otevřeným standardem – její použití je bezplatné.



Obrázek 1.1 Síťová architektura NetLinx

1.1 Ethernet/IP

1.1.1 Úvod

Ethernet Industrial Protocol (Ethernet/IP) je ***otevřený síťový standard***, podporující implicitní (skryté) zprávy (real-time I/O zprávy), explicitní (otevřené) zprávy nebo obojí a současně využívá komerčních komunikačních čipů a fyzických médií. Technologie sítě ethernet je užívána od poloviny sedmdesátých let a je široce akceptována v celém světě. Produkty pro tuto síť dodává velké množství výrobců a jsou snadno dostupné. Využití ethernetu v průmyslové oblasti není jen sledování dnešních technologických trendů, ale dává uživateli možnost používat důvěrně známou technologii i získávat potřebná data

prostřednictvím internetu. Ethernet/IP byl vyvinut z důvodu zvyšujících se požadavků na užívání ethernet sítí v řídicích aplikacích.

Ethernet/IP je otevřená síť užívající:

- IEEE 802.3 fyzický a datový standard
- ethernet TCP/IP a UDP/IP protokol
- řídicí a informační protokol (CIP – Control and Information Protocol), který poskytuje real-time I/O komunikaci a informační komunikaci.

1.1.2 Typy přenosů v sítích Ethernet/IP

Řídicí vrstva CIP se využívá pro posílání I/O zpráv v reálném čase nebo pro implicitní komunikaci.

Informační vrstva CIP se využívá pro výměnu datových zpráv nebo explicitní komunikaci.

Informační: Přenos časově nekritických dat. Typické jsou velké pakety. Informační datové výměny jsou většinou krátce žijící explicitní spojení mezi jedním tvůrcem a jedním cílovým zařízením (kanál může zůstat otevřený). Informační datové pakety využívají TCP/IP (Transmission Control Protocol / Internet Protocol – více viz kapitola 3.4) protokol a výhod TCP datových vlastností.

I/O data: Přenos časově kritických dat. Typické jsou malé pakety. I/O datové výměny jsou dlouhotrvající implicitní spojení mezi jedním tvůrcem a více cílovými zařízeními. I/O datové pakety využívají UDP/IP (User Datagram Protocol / Internet Protocol – více viz kapitola 3.4) protokol a výhod výkonu a rychlosti UDP.

Přenášení dat v reálném čase mezi procesory: Cyklická datová synchronizace mezi jedním produkujícím procesorem a více konzumujícími procesory. Data přenášená v reálném čase využívají UDP/IP protokol a výhod výkonu a rychlosti UDP.

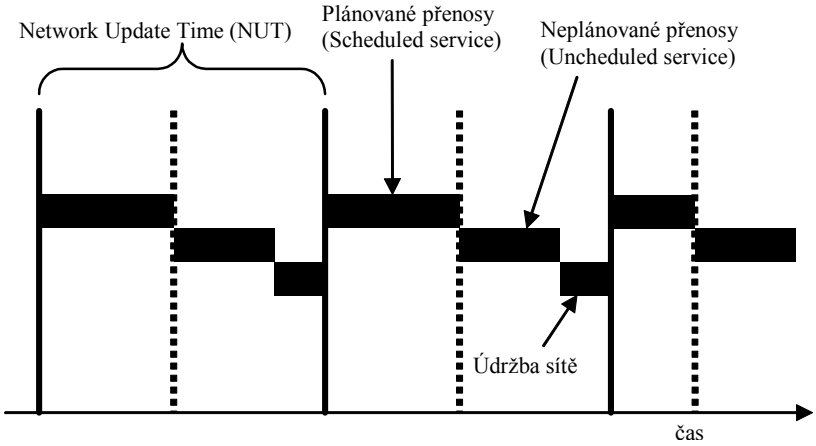
1.2 ControlNet

1.2.1 Úvod

ControlNet je moderní řídicí sběrnice pro průmyslové aplikace. Je to relativně velice rychlá – 5Mb/s (například oproti síti DeviceNet), deterministická a opakovatelná síť, kombinující úkoly dříve existujících sítí DH+ a Remote I/O. Oproti těmto starším sítí přináší ControlNet zlepšení spočívající především v možnosti přesného plánování přenosů. Je určena jak k výměně dat mezi procesory, případně k jejich programování, tak ke sběru dat ze vzdálených I/O zařízení, a to v rozsáhlých a rychlých systémech. ControlNet pracuje na principu producent/konzument a ke koordinaci periodického přístupu všech zařízení k lince je využita metoda CTDMA (Concurrent Time Domain Multiple Access – *současný vícenásobný přístup v definovaném intervalu*). Veškeré komunikace na síti se dějí v uživatelem definovaném, pravidelně se opakujícím časovém úseku – NUT (Network Update Time – čas

obnovy sítě). V něm mají trvale vyhrazený prostor časově kritická data (I/O data, interlock zprávy) jednotlivých stanic – tedy plánované přenosy s uživatelsky definovatelnou četností. Časově nekritická data (zprávy, programování), která představují neplánované přenosy, obsazují zbylý prostor v NUT.

Zmíněný model producent/konzument umožňuje na síti sdílení jednou vyprodukovaných dat více konzumenty v jediném okamžiku. ControlNet je také systémově redundantní síť, umožňující připojit až 99 stanic. Vzdálenost mezi dvěmi stanicemi (bez opakovačů) může být až 1 km. Pomocí opakovačů je možné tuto vzdálenost zvětšit a realizovat jakékoliv topologie (sběrnice, hvězda, strom). Standardním přenosovým médiem je koaxiální kabel RG-6 (75 Ω), provoz je možný i na optických vláknech a stíněné kroucené dvoulince (dva kroucené páry, STP – *shielded twisted pair*).



Obrázek 1.2 Mechanismus přístupu na sběrnici ControlNet

1.2.2 Mechanismus přístupu na sběrnici CTDMA

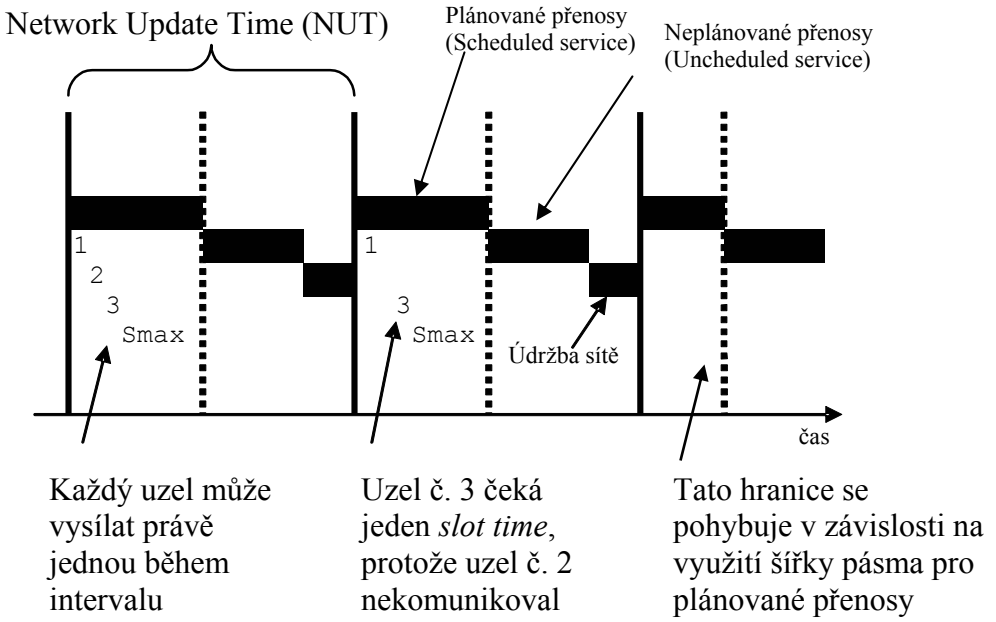
Časová osa je dělena na rovnoměrné intervaly zvané Network Update Time (NUT). Každý interval NUT je rozdělen na 3 podintervaly: ***Plánované přenosy*** *(scheduled service),* ***Neplánované přenosy*** *(unscheduled service)* a ***údržba sítě*** *(network maintenance)*.

Každý uzel (1 až Smax), který se účastní naplánované komunikace, má možnost zaslat zprávu v podintervalu *plánované přenosy*. Jestliže některý uzel nemá k poslání žádná data, zašle i přesto krátký rámec k indikaci svého stavu. Pokud se uzlu nepodaří rámec poslat, další uzel (s vyšším číslem uzlu) počká jeden *slot time*, než zašle vlastní zprávu. Tím je zaručeno, že chyby nebo selhání jednoho uzlu nenaruší cyklus NUT.

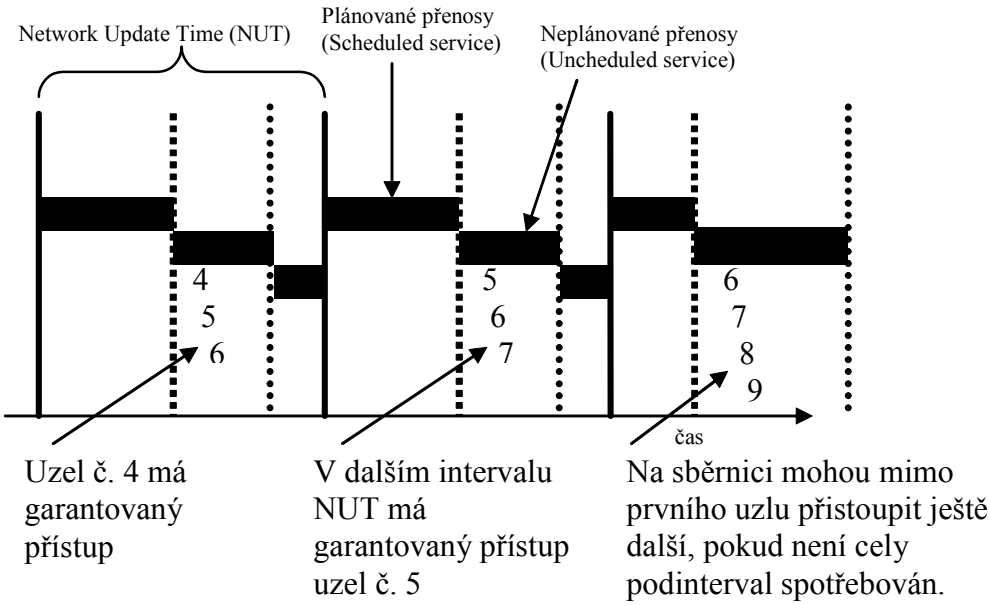
Služba neplánovaných přenosů je určena pro časově nekritickou komunikaci. V tomto podintervalu má pouze jeden uzel garantovaný přístup na sběrnici. Pokud tento uzel nespotřebuje celý podinterval, mohou komunikovat i další uzly (s vyšším číslem uzlu). Číslo uzlu, který může v tomto podintervalu komunikovat jako první, se s každým dalším NUT zvyšuje. Tím je zaručeno, že každý uzel má stejnou šanci přistoupit na sběrnici.

Kombinace těchto dvou metod zaručuje determinismus a opakovatelnost, přičemž je dostatek volnosti pro neplánovanou komunikaci. Rámce zaslané konkrétním uzlem (M

pakety) mohou obsahovat více zpráv (L paketů). Tato vlastnost dovoluje doručování zpráv více příjemcům (multicasting).



Obrázek 1.3 Plánované přenosy na síti ControlNet

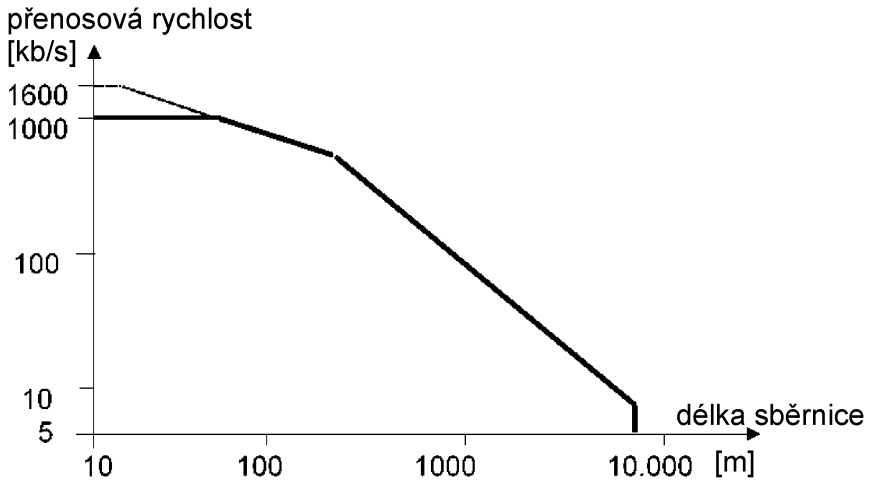


Obrázek 1.4 Neplánované přenosy na síti ControlNet

1.3 DeviceNet

DeviceNet je otevřený síťový standard pro přenos dat v průmyslových systémech. DeviceNet byla vůbec první implementací protokolu CIP; tato síť byla představena v roce 1994. Podporuje až 64 zařízení, přenosová rychlost je volitelně 125 kb/s, 250 kb/s nebo 500 kb/s v závislosti na rozsahu sítě. Tato síť umožňuje připojovat a odpojovat zařízení za chodu. Přenosovým médiem je speciální čtyř žilový stíněný kabel (2 vodiče datové, 2 napájecí a stínění), který sdružuje rozvod napájení a přenos dat, má také vysokou odolnost proti rušení.

Základem je technologie CAN (Controller Area Network), která byla původně vyvinuta pro evropský automobilový průmysl. CAN ve své typické formě (ISO 11898) definuje pouze první a druhou vrstvu ISO-OSI modelu (viz kapitola 3), zbylých horních pět vrstev je definováno protokolem DeviceNet.



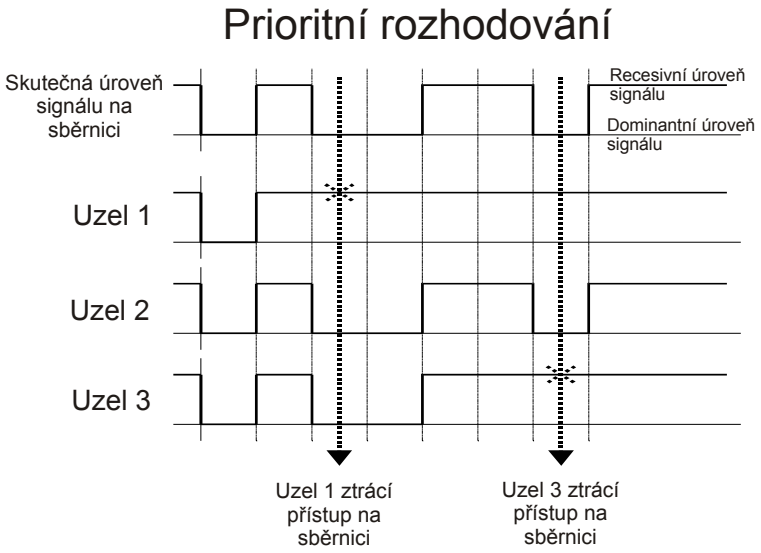
Obrázek 1.5 Závislost přenosové rychlosti sběrnice CAN na její délce

Přenosová rychlost	Délka sběrnice	Bit-time
1 Mb/s	30 m	1 μs
800 kb/s	50 m	1,25 μs
500 kb/s	100 m	2 μs
250 kb/s	250 m	4 μs
125 kb/s	500 m	8 μs
62,5 kb/s	1000 m	20 μs
20 kb/s	2500 m	50 μs
10 kb/s	5000 m	100 μs

Tabulka 1.1 Přenosové rychlosti sběrnice CAN v závislost na délce sběrnice

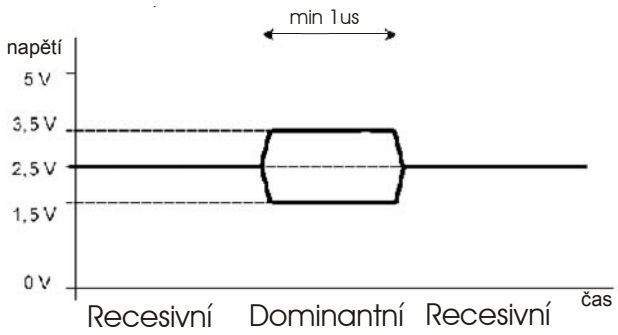
Pro přístup na sběrnici CAN je použito metody CSMA/DCR (Carrier Sense Media Access with Deterministic Collision Resolution) – jedná se o nedestruktivní bitové rozhodování podle priority. Dalším názvem této metody je CSMA/DCR + AMP (Arbitration

on Message Priority). Toto rozhodování je umožněno tím, že fyzická vrstva podporuje reprezentaci dominantní a recesivní úrovně tak, aby sběrnice fungovala jako „wired AND“ – fyzicky vytvořený logický součin (viz obrázek 1.6). Datový rámec na sběrnici CAN obsahuje po uvozujícím jednobitovém SOF (*Start Of Frame* – začátek rámce) tzv. *Arbitration Field* (rozhodovací pole). To je dlouhé 12 až 32 bitů dle nastavené priority uzlu. Každý uzel, který chce vysílat data, vyšle uvozující bity datového rámce a monitoruje úroveň sběrnice. Vysílat může dále ten uzel, jehož rozhodovací pole obsahovalo nejvíce dominantních bitů (nižší binární hodnoty).



Obrázek 1.6 Přístup na sběrnici CAN

Nominální napětí pro recesivní a dominantní úrovně na sběrnici jsou zobrazeny na obrázku 1.7. Díky diferenčnímu charakteru přenosu je CAN necitlivý na elektromagnetické rušení působící shodně na oba vodiče.



Obrázek 1.7 Úrovně na sběrnici CAN

Detekce chyb je uskutečněna pomocí CRC kódu. Ten je 15-ti bitový s Hammingovou vzdáleností 6. Je tedy schopen odhalit až 5 chyb v náhodně rozložených bitech nebo shlukové chyby až do délky řetězce 15 bitů.

Otevřená síť DeviceNet je navržena tak, aby umožňovala dvě základní funkce, potřebné pro jednoduché inteligentní zařízení:

- přenos dat v reálném čase, orientovaný na řízení použitím I/O zpráv
- přenos informací nižší priority, například parametrů konfigurace zařízení nebo ladicích dat, použitím explicitních zpráv.

Standard DeviceNet definuje protokol aplikační a fyzické vrstvy. Zbytek je definován specifikací CAN, která obsahuje definici fyzické a linkové vrstvy ISO-OSI modelu. Specifikace CAN nedefinuje protokol aplikační vrstvy, definici přenosového média a způsob připojení k médiu.

Výhodou sítě DeviceNet je

- vysoká spolehlivost (Hammingova vzdálenost je 6)
- rozsáhlá podpora výrobců hardwaru (Philips, Motorola, Intel, Siemens, atd.) – více než 300 výrobců z nich je členy organizace ODVA.
- nízké náklady na výrobu komunikačních zařízení i na instalaci sítě.

2 Základní popis protokolu CIP

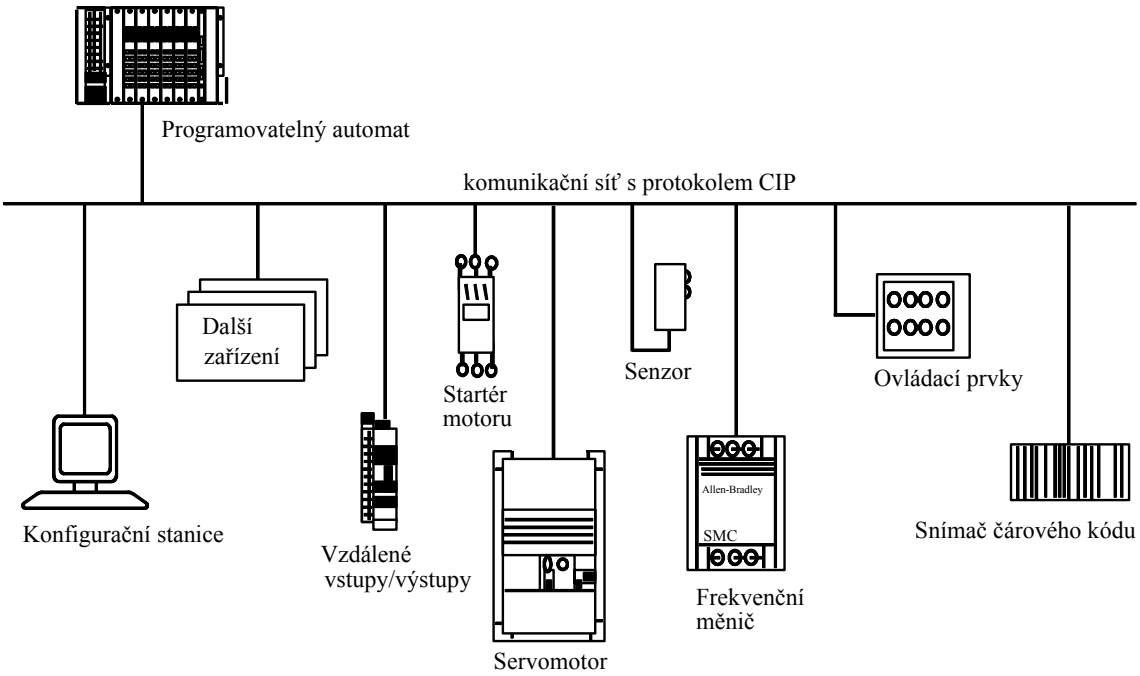
2.1 Úvod

Control and Information Protokol (CIP) je objektově orientovaný protokol, který zajišťuje spojení mezi průmyslovými zařízeními, jako jsou například senzory a regulátory, a zařízeními vyšší úrovně – programovatelnými automaty, panely operátorů.

CIP je nezávislý na fyzické a linkové vrstvě, zahrnuje v sobě relační, prezentační a aplikační vrstvu ISO-OSI modelu. Obsahuje tedy společné spojovací principy pro jednotlivé síťové standardy architektury NetLinx.

Hlavními úkoly protokolu CIP jsou:

- přenos časově kritických dat mezi I/O zařízeními
- přenos ostatních, časově nekritických dat jako například konfiguračních a diagnostických dat



Obrázek 2.1 Komunikační síť založená na protokolu CIP

2.2 Objektový popis

Protokol CIP je objektově orientovaný protokol – každý uzel sítě je popsán jako množina objektů. Objekt je abstraktní reprezentace jednotlivých komponent zařízení. Realizace tohoto abstraktního modelu je závislá na konkrétní implementaci.

Komunikační protokol CIP obsahuje velký soubor obecně definovaných objektů (v současnosti se jedná o 46 tříd objektů). Pouze několik tříd objektů je však specifických

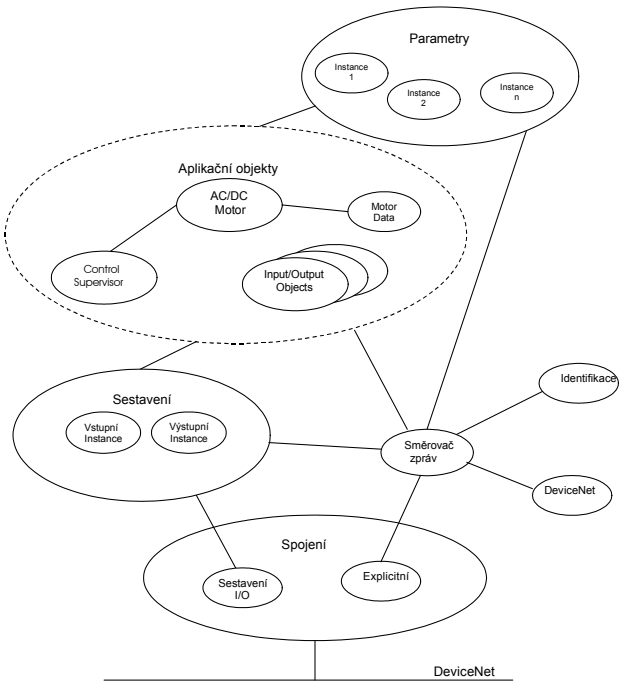
jednotlivým linkovým vrstvám: jedna třída pro DeviceNet, tři třídy pro ControlNet a dvě třídy pro linkovou vrstvu EtherNet/IP. Všechny ostatní objekty jsou obecné a mohou být použity ve všech třech sítích.

Další objekty jsou přítomny v závislosti na typu zařízení. To umožňuje velice dobrou škálovatelnost zařízení, například senzor na síti DeviceNet má velice jednoduchý objektový model a není tak zbytečně složitý. Vývojáři nových zařízení obvykle používají veřejně definované objekty, ale mohou také vytvářet svoje vlastní objekty v adresovém prostoru, který je přiřazen dané organizaci.

Příkladem povinného veřejně definovaného objektu je např. ***identity object*** (identifikační objekt). Zařízení obvykle nemění svoji identitu, takže všechny povinné atributy jsou přístupné pouze pro čtení.

Identity object	
Povinné atributy	Volitelné atributy
ID dodavatele Typ zařízení Kód produktu Revize Status Sériové číslo Název produktu	Stav *Configuration consistency value* (ocenění bezespornosti konfigurace) *Heartbeat interval* (tep sítě)

Tabulka 2.1 Příklad veřejně definovaného objektu



Pro příklad modelu skutečného zařízení uvádím objektový model motoru (až již se střídavým nebo stejnosměrným napájením). V modelu tohoto zařízení je zahrnuto řízení otáček v otevřené i uzavřené smyčce a řízení kroutivého momentu. Tento model je zobrazen na obrázku 2.2.

Obrázek 2.2 Model skutečného zařízení (motoru)

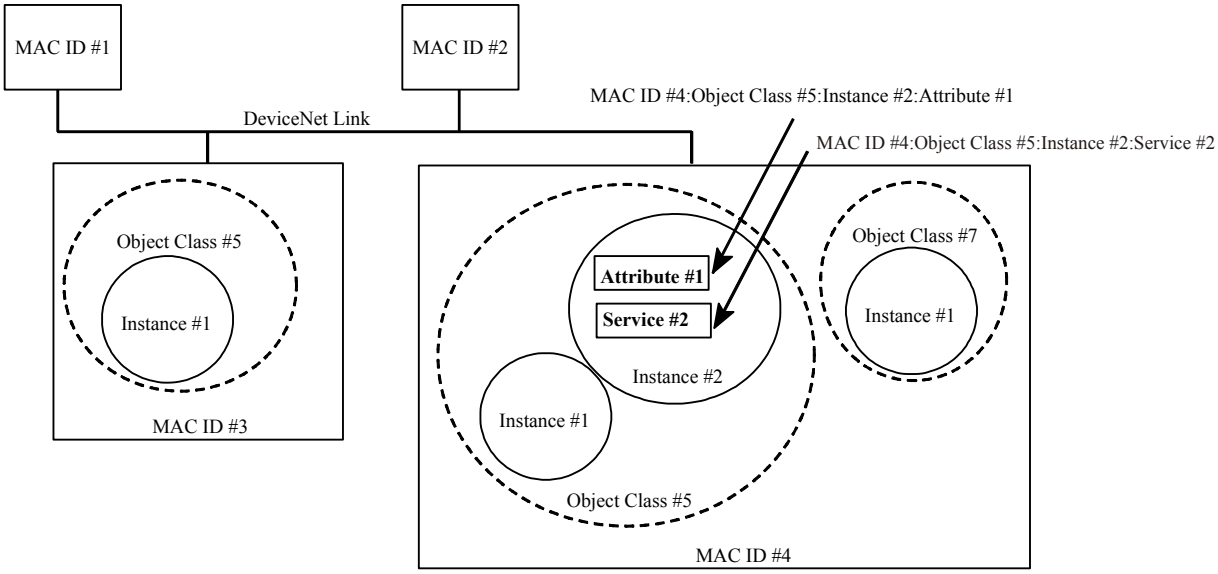
Fyzicky samostatné části se v protokolu CIP logicky adresují. Adresa se skládá ze čtyř částí:

- ***MAC ID*** (Media Access Control ID – identifikátor přístupu na médium) – číselná hodnota typu integer přiřazená každému uzlu sítě CIP, musí být jedinečná. Tato adresa je přiřazena výrobcem z adresového rozsahu přidělenému danému výrobci.
- ***Class ID*** – číselná hodnota typu integer přiřazená každé třídě objektů přístupných ze sítě.
- ***Instance ID*** – číselná hodnota typu integer přiřazená každé instanci (realizaci) objektu, reprezentuje instanci uvnitř třídy objektů. Hodnota je jedinečná ve dvojici MAC ID:Class ID

Čtvrtou částí adresy je buď:

- ***Atribute ID*** – hodnota typu integer označující adresu příslušného atributu, který náleží třídě a/nebo identifikátoru instance.
- ***Service*** – hodnota typu integer označující jednotlivé funkce (služby) instancí objektů

Pro přehlednost je na obrázku 2.3 zobrazeno toto adresové schema.



Obrázek 2.3 Adresování v protokolu CIP

Každé zařízení připojené k síti, které využívá služeb protokolu CIP, se skládá z objektů. První skupinu tvoří objekty, které se povinně vyskytují u všech zařízení. Druhou skupinu tvoří objekty, jejichž výskyt je závislý na konkrétní realizaci zařízení. Třetí skupinou

jsou objekty specifické pro danou síť (DeviceNet, ControlNet, EtherNet/IP). Povinně vyskytující objekty u všech zařízení jsou:

- *Identity object* (objekt totožnosti) – tento objekt poskytuje identifikaci a základní informace o zařízení. Jesliže se zařízení skládá z několika samostatných částí, existuje pro každou část zařízení jedna instance objektu totožnosti.
- *Assembly object* (objekt sestavení) – tento objekt spojuje atributy více objektů, čímž dovoluje, aby data do/z každého objektu mohla být zaslána nebo přijata pouze pomocí jednoho spojení.
- *Connection object* (objekt připojení) – tento objekt je použit pro komunikaci uskutečněnou skrze vytvořené spojení i pro komunikaci bez vytvořeného spojení včetně vytvoření spojení přes několik podsítí.
- *Message router object* (objekt směrovač zpráv) – tento objekt poskytuje spojovací bod, skrze který může klient adresovat jakoukoliv službu třídy objektů nebo instanci uvnitř fyzického zařízení.
- *Application object* (aplikační objekt) – popisuje konkrétní zařízení.
- *Parameter object* (objekt parametrů) – poskytuje rozhraní ke konfiguraci zařízení. Dále tento objekt poskytuje všechny informace nutné k definování a popsání všech konfigurovatelných parametrů zařízení.

Objekty specifické pro danou síť jsou

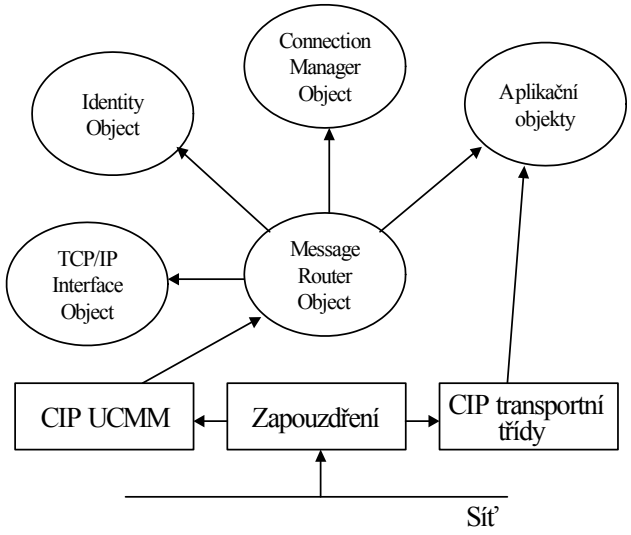
- *DeviceNet object* (objekt DeviceNet)
- *ControlNet object* (objekt ControlNet)
- *ControlNet Keeper object* (objekt „opatrovník“ sítě ControlNet)
- *ControlNet Scheduling object* (objekt rozvrhování sítě ControlNet)
- *TCP/IP Interface object* (objekt rozhraní TCP/IP)
- *Ethernet Link object* (objekt linkové vrstvy sítě ethernet)

Pro síť EtherNet/IP jsou některé objekty specifické. Každé EtherNet/IP zařízení musí implementovat alespoň jednu instanci těchto objektů:

- *Identity Object* (kód třídy 0x01)
- *Message Router Object* (kód třídy 0x02)
- *Connection Manager Objekt* (kód třídy 0x06)
- *TCP/IP Interface Object* (objekt rozhraní TCP/IP, kód třídy 0xF5)
- *Ethernet Link Objekt* (objekt linkové vrstvy sítě ethernet, kód třídy 0xF6), je-li použito médium ethernet[1].

[1] I když síť EtherNet/IP má ve svém názvu ethernet, technicky vzato není ethernet vyžadován. Linková vrstva sítě EtherNet/IP může být použita kterákoliv; požadavkem je podpora IP protokolu. Například se může jednat o linky FDDI, modemové linky (PPP nebo SLIP), ATM a podobně. Každé médium samozřejmě musí být použito v souladu s platnou normou – např. při použití sítě ethernet je to IEEE 802.3.

Každé zařízení musí dále implementovat *CIP Unconnected Message Manager* (UCMM – manažer zpráv zasílaných bez vytvořeného spojení), ačkoliv nemá kód třídy. Pro názornost, které objekty jsou vyžadovány v síti EtherNet/IP je na obrázku 2.4 základní objektový model zařízení:



Obrázek 2.4 Objektový model EtherNet/IP zařízení

2.3 Elektronické datové listy (Electronic data sheets)

Samotný konzistentní objektový model je k ničemu, jestliže není dispozici mechanismus pro identifikaci objektů, které jsou v zařízení implementovány. Protokol CIP poskytuje několik možností, jak zařízení konfigurovat:

- Pomocí datového listu
- *Parameter objects* (objekt parametrů) a *Parameter object stubs* (výsek objektu parametrů)
- Electronic data sheet (EDS)
- Kombinace EDS a *Parameter object stubs*
- *Configuration assembly* a některá z předchozích metod

V případě konfiguračních informací na vytištěném datovém listu je možno použít konfigurační nástroje pouze k přenesení zadaných informací do zařízení. Toto řešení je nejméně vhodné, protože nerozlišuje kontext, obsah a formát dat.

Parameter object poskytuje úplný popis všech konfigurovatelných dat v zařízení. To umožňuje konfiguračnímu nástroji přístup ke všem parametrům a poskytovat uživatelsky příjemné rozhraní. Atributy obsahují informace o datech jako je typ, jednotky, minimální, maximální a obvyklé hodnoty, měřítko a podobně. Tato metoda však jednoduchá zařízení, např. pro síť DeviceNet, zatěžuje příliš mnoha informacemi o parametrech.

Pro tato zařízení je možné použít zjednodušenou verzi *Parameter object*, která se nazývá *Parameter object stub* (v doslovném překladu znamená útržek objektu parametrů). Tato verze poskytuje přístup pouze k parametrům, nepopisuje však jejich význam. EDS poskytuje informace, které jsou rozdílem mezi množinou informací, kterou poskytuje *Parameter object* a množinou informací, kterou poskytuje *Parameter object stub*. Kombinace EDS a *Parameter object stub* tak poskytuje úplnou funkčnost a jednoduchost použití bez přetěžování jednotlivých zařízení.

2.4 Typy spojení

Jak je vidět z obrázku 2.5, přístup k vnitřním objektům modelu kteréhokoliv zařízení je uskutečněn pomocí jednoho z těchto objektů:

- *Unconnected message manager* (manažer zpráv bez vytvořeného spojení)
- *Connection manager* (manažer zpráv s vytvořeným spojením)

Většina zpráv na síti s protokolem CIP je zaslána pomocí vytvořeného spojení. Musí být tedy definován proces, který vytváří spojení mezi zařízeními, mezi nimiž ještě spojení neexistuje. Tímto procesem je *Unconnected message manager* (UCMM), který zpracovává požadavky na spojení. Po vytvoření spojení jsou vyhrazeny všechny komunikační prostředky v zařízení, včetně dalších mostů a směrovačů.

Všechna spojení v síti CIP lze rozdělit do dvou skupin: První skupinou jsou ***explicitní*** spojení a druhou skupinou ***implicitní*** spojení.

2.4.1 Zasílání explicitních zpráv

Zasílání explicitních zpráv poskytuje víceúčelovou komunikační cestu mezi dvěmi zařízeními. Zasílání explicitních zpráv je typická komunikace typu požadavek/odezva, je vždy uskutečněna pomocí objektu směrovače zpráv (*message router*). Požadavek obsahuje explicitní informaci, kterou příjemce dekóduje, na zprávu odpovídajícím způsobem zareaguje a zašle odpovídající odezvu. Pro přímé spojení mezi dvěmi zařízeními je třeba zdrojová adresa, cílová adresa a identifikace připojení (*Connection ID*) pro každý směr komunikace. Adresou na síti DeviceNet a ControlNet je číslo uzlu (*node number*), adresou na síti EtherNet/IP je IP adresa. Explicitní zprávy jsou spouštěny externími událostmi

2.4.2 Zasílání implicitních zpráv

Zasílání implicitních zpráv poskytuje speciální komunikační cestu mezi objektem produkující aplikace a jedním nebo více objekty konzumujících aplikací. Tento typ zpráv je určen pro I/O data – tento typ zpráv se také často nazývá I/O zprávy. Spojení se nazývá implicitní proto, že význam přenášených dat je implicitně definován pomocí identifikace připojení (*Connection ID*).

Tato výměna dat je pro uživatele transparentní a nachází se v aplikační vrstvě protokolu s tím, že producent i konzument „vědí“, jaký typ dat zpráva obsahuje a jaký je její význam (např. data z analogového vstupního modulu) ještě před samotným přenosem.

Implicitní komunikace se běžně používá pro I/O zprávy; tento přístup plně využívá modelu producent/konzument a je též používán i pro naplánovanou komunikaci mezi automaty (tzv. produkované / konzumované tagy).

Principiálně jsou čtyři typy implicitních zpráv (dané specifikací protokolu CIP):

- *Polled* (dotazované)
- *Change of State* (změna stavu)
- *Cyclic* (cyklické)
- *Strobed* (snímané)

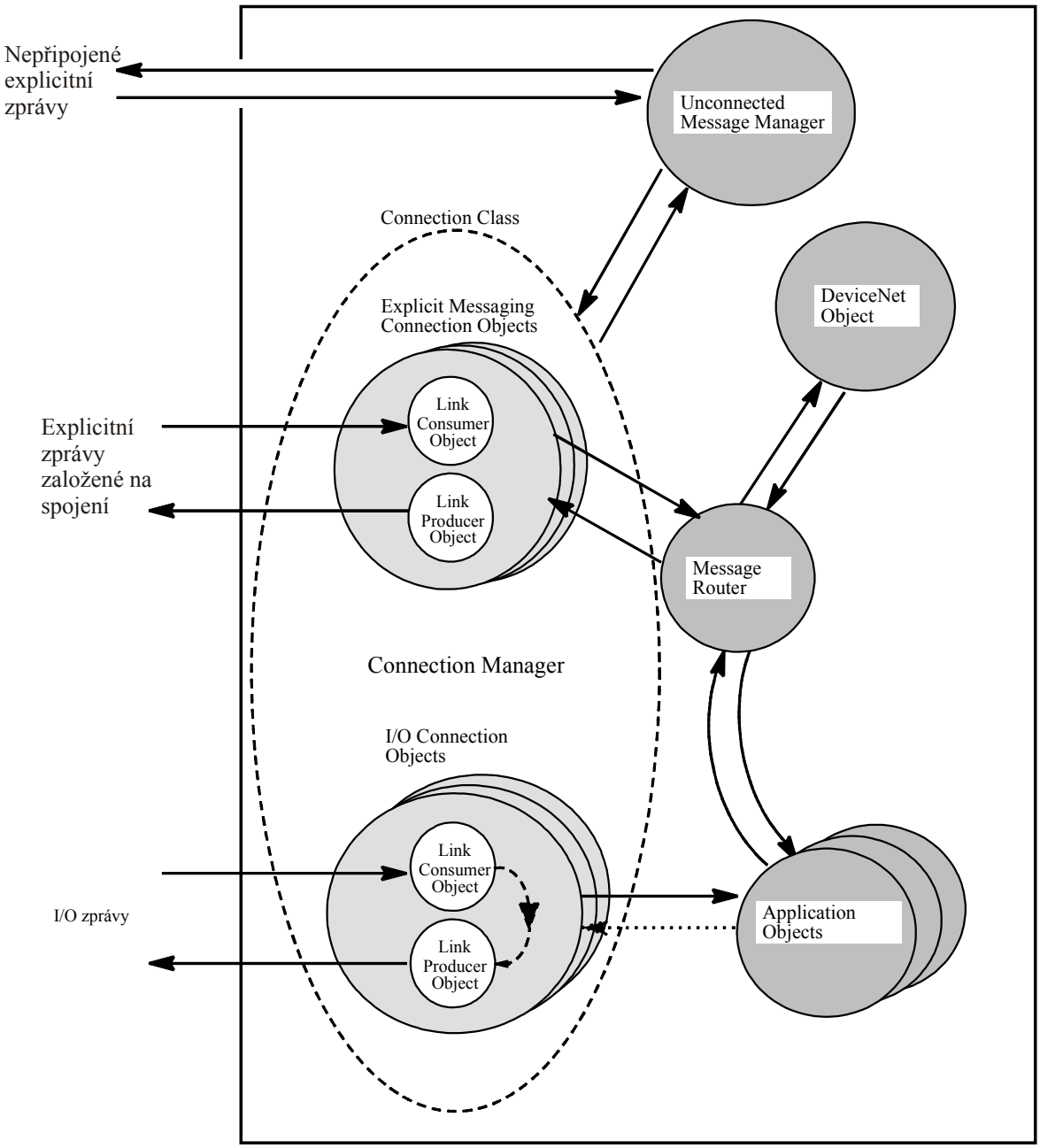
Polled messages jsou zprávy s typem komunikace používaným v dřívějších I/O sítích: Zařízení typu *master* postupně žádá zařízení typu *slave* o data. V žádosti mohou být obsažena výstupní data pro cílové zařízení (je-li to relevantní) a odpověď zařízení obsahuje informace o vstupních datech tohoto zařízení (je-li to relevantní).

Strobed je speciální případ předchozího případu, kdy master zasílá jeden požadavek typu multicast a zařízení typu slave postupně odpovídají zprávou se svými daty. Žádné další zprávy nejsou od mastera vyžadovány.

Cyclic messages jsou produkovány zařízením v určitém, pravidelně se opakujícím časovém intervalu. Zprávě je přiřazeno Connection ID. Jakékoliv další zařízení, které potřebuje data od produkujícího zařízení musí znát Connection ID produkujícího zařízení a přijímá všechny pakety na síti s tímto Connection ID.

Change of State je podobné cyklickým zprávám kromě toho, že data jsou produkována pouze při jejich změně. Změna stavu také zahrnuje cyklické zprávy *heartbeat* (v doslovném překladu znamená tlukot srdce), takže konzumující aplikace je schopna zjistit, zda-li je zařízení stále funkční a on-line.

Upřednostňovaným typem implicitní komunikace na síti EtherNet/IP jsou cyklické zprávy (*cyclic messages*). Tato komunikace je optimální z hlediska integrity dat a zatížení sítě. Protože se implicitní zprávy přenášejí od jednoho producenta obecně více konzumentům, není možné použít protokolu TCP (Transmission Control Protocol, více viz kapitola 3.4.3), který je pouze pro spojení typu bod-bod. Používá se tedy protokolů UDP/IP – protokol UDP (User Datagram Protocol, více viz kapitola 3.4.4) je mnohem jednodušší než-li TCP a je snadněji zpracovatelný, což je výhodné v malých (jednoduchých) zařízeních. Více se o protokolech použitých pro přenos implicitních zpráv zmiňuji v kapitole 3.4 Zapouzdření protokolu CIP do TCP/UDP paketů.



Obrázek 2.5 Objektový model komunikačního CIP modulu

2.5 Transportní třídy protokolu CIP

Prokol CIP rozlišuje sedm transportních tříd v závislosti na tom, zda-li koncový bod spojení může data pouze produkovat, pouze přijímat nebo obojí. V následující tabulce jsou tyto transportní třídy v krátokosti pospsány.

Transportní třída	Popis
Transportní třída 0	Koncový bod spojení typu klient může data pouze produkovat a koncový bod spojení typu server může data jenom přijímat.
Transportní třída 1	Koncový bod spojení typu klient produkuje data, která jsou opatřena počítadlem sekvence. To slouží k ověření duplicity doručených paketů v koncovém bodě spojení typu server.
Transportní třída 2	Koncový bod spojení může data produkovat i konzumovat. Koncový bod spojení typu klient odešle zprávu do koncového bodu spojení typu server, který data zpracuje, ověří zda doručená zpráva není duplicitní a odešle odpověď.
Transportní třída 3	Koncový bod spojení může data produkovat i konzumovat. Koncový bod spojení typu klient odešle vyprodukovaná data do koncového bodu spojení typu server, který data zpracuje a předá aplikačnímu objektu. Jsou-li přijadá data platná, aplikační objekt zašle odpověď.
Transportní třída 4	Neblokující
Transportní třída 5	Neblokující s možností fragmentace zprávy
Transportní třída 6	Multicast s možností fragmentace zprávy
Rezervováno	

Tabulka 2.2 Transportní třídy v protokolu CIP

2.6 Objekt manažer spojení (Connection Manager Object)

V této části jsou popsány parametry objektu Connection Manager, které mají speciální vztah k TCP/IP zapouzdření.

2.6.1 Parametry spojení (Connection parameters)

Connection Type (typ spojení)

CIP spojení mohou být NULL (žádné), MULTICAST (jeden odesílatel → více příjemců), nebo POINT2POINT (bod-bod). Spojení typu MULTICAST je podporováno pouze pro CIP transportní třídy 0 a 1. Naopak pro transportní třídy 2 a 3 je možné použít

pouze spojení typu bod-bod (POINT2POINT). Důvodem je využití protokolu TCP, který jiný typ spojení neumožňuje.

Priority (priorita)

CIP priority jsou LOW (nízká) a HIGH (vysoká). Priorita může být nastavena i na hodnotu SCHEDULED (plánovaná), která má v současnosti stejný význam jako HIGH. Předpokládá se, že v budoucnosti bude typ priority SCHEDULED dodefinován.

Trigger Type (typ spouštěče)

CIP Trigger může být buď CYCLIC (cyklický), CHANGE_OF_STATE (změna stavu), nebo APPLICATION (řízen aplikací).

Connection Size (velikost)

Connection Size nemá být větší než-li 65511 bytů.

Connection request time-out (časový limit požadavku na spojení)

Pro spolehlivé založení CIP spojení, které je na TCP/IP lince, musí být časový limit požadavku na spojení dostatečně velký – což v sobě může zahrnovat například překlad doménového jména nebo průchod několika směrovači (*routery*). Z tohoto důvodu (mnoho možností ve zpracování požadavku na spojení) CIP směrovače nezmenšují časový limit požadavku na spojení (narozdíl od směrovačů používaných v sítí založených na IP protokolu, kdy je časový limit (tzv. TTL) při každém průchodu směrovačem snížen).

Connection path (cesta spojení)

Linková část adresy TCP/IP spojení se kóduje jako řetězec ASCII znaků. Podporovány jsou tyto notace:

- IP adresa v tečkové notaci, např. 147.32.87.231 (viz RFC 1117)
- IP adresa v tečkové notaci následovaná dvojtečkou a číslem TCP portu
- Hostname (jméno hostitele), například controllogix.felk.cvut.cz. Hostname je přeloženo pomocí DNS požadavku na jmenný (DNS) server.
- Hostname následované dvojtečkou a číslem TCP portu (dekadicky či šestnáctkově – 0x...)

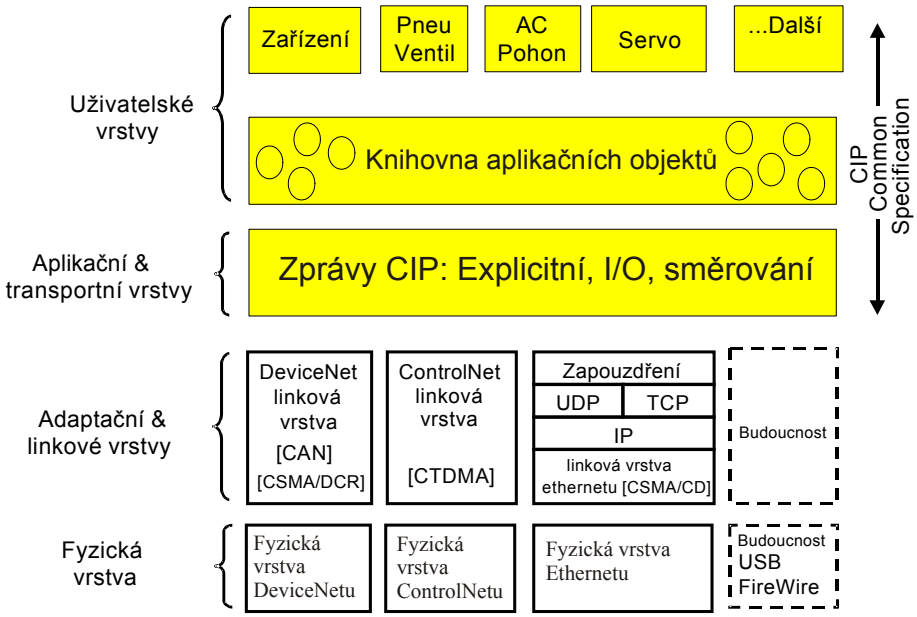
3 Popis sítě EtherNet/IP

3.1 Úvod

EtherNet/IP je komunikační systém vhodný pro použití v průmyslovém prostředí. Umožňuje výměnu časově kritických dat mezi průmyslovými zařízeními od senzorů až po programovatelné automaty. EtherNet/IP používá Control and Information Protocol (CIP), který tvoří síťovou, transportní a aplikační vrstvu ISO-OSI modelu. Control and Information Protocol je též používán sítěmi ControlNet a DeviceNet.

EtherNet/IP poskytuje model producent/konzument pro výměnu časově kritických dat. Ten dovoluje výměnu dat mezi jedním data produkujícím zařízením a více data přijímajícími zařízeními (pomocí zasílání adresných oběžníků – IP multicast).

EtherNet/IP využívá standardu IEEE 802.3 bez dalších nestandardních doplňků pro zvýšení determinismu. Samozřejmostí je použití přepínané sítě 100 Mb/s.



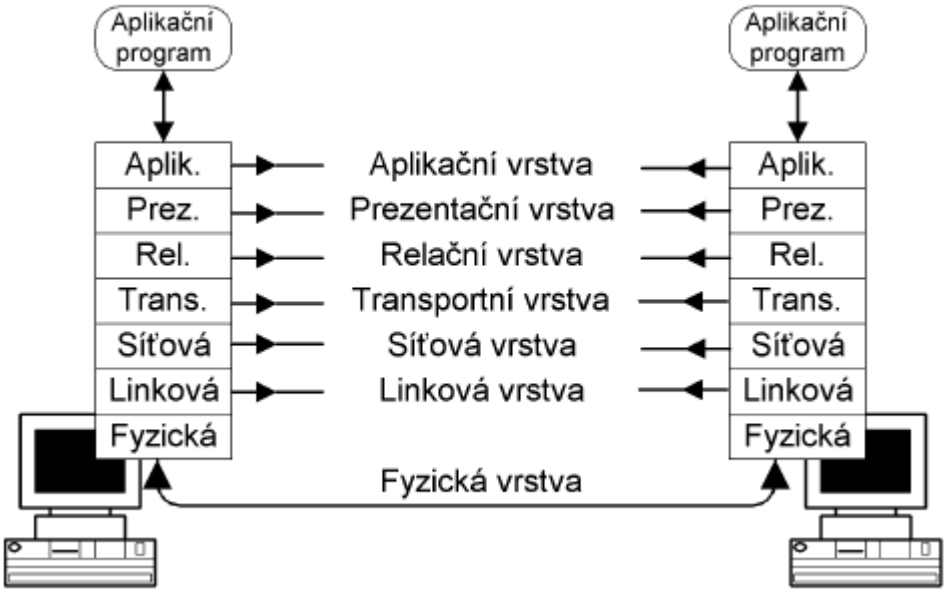
Obrázek 3.1 Základní přehled o protokolu CIP

Síť EtherNet/IP využívá zapouzdření paketů protokolu CIP do protokolu TCP/IP a UDP/IP používaném na běžných lokálních sítích ethernet (LAN). Proto se zde zmíním o základních vlastnostech protokolů IP (Internet Protocol), TCP (Transmission Control Protocol) a UDP (User Datagram Protocol) a dalších protokolů, které síť EtherNet/IP využívá. V neposlední řadě také o linkové vrstvě sítě ethernet.

Další termín, který budu v dalším textu používat, je síťový model ISO - OSI.

3.2 Síťové modely ISO-OSI a TCP/IP

Mezinárodní normalizační organizace (ISO – International Standard Organization) normalizovala soustavu protokolů označovaných jako ISO - OSI. Komunikace mezi dvěma počítači (zařízeními) je schématický znázorněna na následujícím obrázku:



Obrázek 3.2 ISO-OSI síťový model

Fyzická vrstva popisuje elektrické či optické signály používané při komunikaci mezi počítači. Na fyzické vrstvě je vytvořen tzv. fyzický okruh.

Linková vrstva zajišťuje přístup ke sdílenému médiu a adresaci na fyzickém připojení v jednom síťovém segmentu. Datové jednotky přenášené linkovou vrstvou jsou rámce (*frame*).

Síťová vrstva zajišťuje adresaci v rámci síťového prostředí s více fyzickými segmenty. Používá logické adresy a prostřednictvím nich zajišťuje přenos dat z jednoho zařízení na druhé i z jedné sítě do jiné. Datové jednotky přenášené síťovou vrstvou jsou pakety (*packet*).

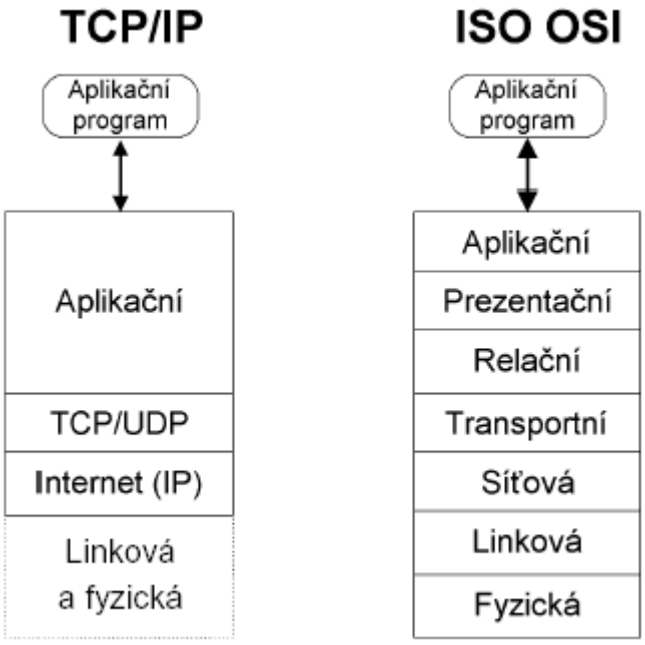
Transportní (přenosová) vrstva zajišťuje spolehlivost a kvalitu přenosu, jakou požadují vyšší vrstvy. Tato vrstva nabízí nespojové (služby bez vytvořeého spojení) a spojově orientované služby (s vytvořeným spojením). Datové jednotky přenášené přenosovou vrstvou jsou TPDU (*Transport Layer Protocol Data Unit* – datová jednotka protokolu transportní vrstvy).

Relační (spojová) vrstva zajišťuje pravidla pro navazování a ukončování datových přenosů mezi uzly na síti. Dále zajišťuje služby typu překladu jmen na adresy nebo bezpečnost přístupu. Jednou z funkcí této vrstvy je také synchronizace přenosů, která je zajištěna značkami vytvářenými touto vrstvou. Datové jednotky přenášené spojovou vrstvou jsou SPDU (*Session Layer Protocol Data Unit*- datová jednotka protokolu relační vrstvy).

Prezentační vrstva je zodpovědná za formátování a syntaxi dat. Další funkcí je konverze přenášených dat do formátu srozumitelného pro přijímající zařízení, jedná se

například o šifrování nebo komprimaci. Datové jednotky přenášené prezentační vrstvou jsou PPDU (*Presentation Layer Protocol Data Unit* – datová jednotka protokolu prezentační vrstvy).

Aplikační vrstva Aplikační vrstva představuje okno, prostřednictvím kterého mohou uživatelé nebo aplikace vidět výsledky služeb zajišťovaných všemi předcházejícími vrstvami. Datové jednotky přenášené aplikační vrstvou jsou APDU (*Application Layer Protocol Data Unit* – datová jednotka protokolu aplikační vrstvy).



Aplikační protokoly většinou odpovídají několika vrstvám ISO-OSI modelu. Relační, prezentační a aplikační vrstva ISO-OSI je zredukována do jedné aplikační vrstvy TCP/IP modelu. To je zobrazeno na následujících obrázcích:

V tabulce 3.1 je zobrazeno, které vrstvy ISO-OSI modelu pokrývá protokol CIP. V ISO-OSI modelu jsou to vrstvy relační, prezentační a aplikační, v TCP/IP modelu je to aplikační vrstva.

Obrázek 3.3 Porovnání síťových modelů TCP/IP a ISO-OSI

Aplikační	Protokoly a služby pro aplikace Výběr typu dialogu Identifikace a autorizace	7	FTP, SNMP SMTP, HTTP Telnet, **CIP**
Prezentační	Reprezentace dat Definice kódování Definice použitých znaků	6	
Relační	Řízení dialogu Synchronizace komunikace	5	
Transportní	Vytváření sekvence z dat Řízení začátku a konce přenosu Detekce a oprava chyb	4	UDP/TCP
Síťová	Směrování Upřednostňování Vytvoření a uzavření spojení	3	IP
Linková	Vytváření rámců Řízení sekvencí a toku	2	ethernet
Fyzická	Přenos bitů Kódování a synchronizace	1	

Tabulka 3.1 Začlenění protokolu CIP do ISO-OSI síťového modelu

3.3 Porovnání síťových modelů ControlNet a EtherNet/IP

Porovnáme-li síťové modely ControlNet a EtherNet/IP, je zřejmé, že není rozdílu v aplikační vrstvě. Protokol aplikační vrstvy (Control and Information Protocol) je v případě EtherNet/IP zapouzdřován do protokolů TCP/UDP/IP. Přístup na fyzické médium je u sítě ControlNet řešen metodou CTDMA (současný přístup na médium v definovaném intervalu), který vylučuje kolize. Naproti tomu může při nevhodné volbě prvků (a jejich konfiguraci) fyzické vrstvy (přepínače, směrovače, případně *huby* – rozbočovače) metoda přístupu na sběrnici CSMA/CD (přístup na médium s příposlechem nosné / s detekcí kolizí) způsobit naprostou nepoužitelnost sítě EtherNet/IP pro řídicí účely. Síť EtherNet/IP se vyznačuje použitím aktivních síťových prvků, které jsou nutné pro vyloučení kolizí (nikoliv však stoprocentní). Kolize jsou díky použití přepínané 100 Mb/s sítě (Fast Ethernet) téměř vyloučeny – další zvýšení determinismu je třeba provádět na aplikační úrovni.

V tabulce 3.2 je názorně zobrazeno porovnání těchto dvou síťových modelů.

	Společná aplikační vrstva	**Společná aplikační vrstva**	
		TCP/UDP IP	
Plánované přenosy I/O Vysoký determinismus	**CTDMA** Concurrent Time Domain Media Access	**CSMA/CD** Carrier Sense Media Access with Collision Detection	Kolize v síti ethernet jsou prakticky vyloučené použitím přepínačů a 100 Mb/s full-duplexních zařízeních
Pasivní prvky Trunkline topologie Redundantní prvky Vnitřní bezpečnost	**Fyzická vrstva**	**Fyzická vrstva**	Aktivní prvky Hvězdicová topologie Standardní RJ45 síťové produkty.
	ControlNet	**EtherNet/IP**	

Tabulka 3.2 Porovnání síťových modelů ControlNet a EtherNet/IP

3.4 Zapouzdření protokolu CIP do TCP/UDP paketů

Ve vztahu k ISO-OSI modelu je zapouzdření (*encapsulation*) CIP paketů do TCP/IP sítě provedeno v transportní vrstvě. Pro bližší porozumění principu zapouzdření CIP paketů do TCP/UDP paketů v následujících odstavcích stručně popíšu protokoly použité při zapouzdření.

3.4.1 Protokol IP (InterNet Protocol)

IP protokol je protokol, umožňující spojit jednotlivé lokální sítě do celosvětového internetu. IP protokol tedy přepravuje data mezi libovolnými počítači (zařízeními) v internetu. Od protokolu IP dostal také internet své jméno.

IP-protokol je tvořen několika dílčími protokoly:

- Vlastním protokolem IP.
- Služebním protokolem ICMP sloužícím zejména k signalizaci mimořádných stavů.
- Služebním protokolem IGMP sloužícím při zajišťování dopravy adresných oběžníků (*multicast*).
- Služebními protokoly ARP a RARP, které jsou často vyčleňovány jako samostatné, na IP nezávislé protokoly, protože jejich rámce nejsou předcházeny IP-záhlavím.

V IP protokolu má každé síťové rozhraní alespoň jednu IP adresu, která je v případě IP protokolu veze 4 čtyřbajtová, a v případě IP protokolu verze 6 šestnáctibajtová.

Základní jednotkou přenášených dat je IP datagram

IP-datagram verze 4:

<table>
<tr><td colspan="2">0</td><td>8</td><td colspan="2">16</td><td>24</td></tr>
<tr><td>Verze IP</td><td>Délka záhlaví</td><td>Typ služby</td><td colspan="3">Celková délka IP datagramu</td></tr>
<tr><td colspan="3">Identifikace IP datagramu</td><td>Příznaky</td><td colspan="2">Posunutí fragmentu od počátku</td></tr>
<tr><td colspan="2">Doba života datagramu (TTL)</td><td>Protokol vyšší vrstvy</td><td colspan="3">Kontrolní součet z IP záhlaví</td></tr>
<tr><td colspan="6">IP adresa odesílatele 32 bitů</td></tr>
<tr><td colspan="6">IP adresa příjemce 32 bitů</td></tr>
<tr><td colspan="6">Volitelné položky záhlaví</td></tr>
<tr><td colspan="6">Přenášená data (nepovinné)</td></tr>
</table>

Tabulka 3.3 Struktura IP datagramu (verze 4)

Protokol vyšší vrstvy obsahuje číselnou identifikaci protokolu vyšší vrstvy, který využívá IP-datagram ke svému transportu. V praxi se nesetkáváme s případem, že by se komunikovalo přímo IP-protokolem. Vždy je použit protokol vyšší vrstvy (TCP nebo UDP) nebo jeden ze služebních protokolů ICMP či IGMP (viz dále). Protokoly ICMP a IGMP jsou sice formálně součástí protokolu IP, avšak chovají se jako protokoly vyšší vrstvy, tj. v přenášeném paketu je záhlaví IP-protokolu následováno záhlavím protokolu ICMP (resp. IGMP).

IP-datagram verze 6

0		8	16	24
Verze IP	Třída dat	Identifikace toku dat		
Délka dat			Další hlavička	Doba života datagramu (TTL)
IP adresa odesílatele 128 bitů				
IP adresa příjemce 128 bitů				

Tabulka 3.4 Struktura IP datagramu (verze 6)

Další informace o protokolu IP je možno získat například v [4].

3.4.2 Protokol IGMP (Internet Group Management Protocol)

Protokol IGMP je stejně jako ICMP (Internet Control Message Protocol) služebním protokolem pro protokol IP. Zatímco protokol ICMP slouží k signalizování mimořádných situací v sítích postavených na IP, protokol IGMP slouží pro komunikaci mezi počítačem (zařízením) a směrovačem v situaci, kdy se chce počítač přihlásit do multicastové skupiny. Tehdy je potřeba dát vědět okolním směrovačům, ať mu zasílají datagramy pro tuto skupinu. Směrovače pak dále kontaktují další směrovače a ustanoví cestu pro multicastové datagramy. Současná verze protokolu IGMP je 2.

IGMP pracuje ve dvou fázích:

1) Když počítač vstupuje do jisté multicastové skupiny, zašle IGMP zprávu na adresu 224.0.0.2 deklarující jeho členství. Lokální směrovače zprávu přijmou a ustanoví příslušnou cestu k hostiteli dalším propagováním jeho členství.
2) Protože členství je dynamické, směrovač pravidelně počítá počítače na lokálních sítích, aby zjistil, zda ještě k dané skupině někdo patří. Jestli počítač odpovídá kladně, směrovač udržuje informace o navázaných cestách, jinak po několika nezodpovězených dotazech přestane propagovat členství ve skupině do okolí.

IP záhlaví	Typ 1 bajt	MRT 1 bajt	Kontrolní součet 2 bajty	IP adresa adresného oběžníku 4 bajty

Tabulka 3.5 IGMP paket

IGMP zpráva má osm oktetů (*octets* – bytů), viz. obrázek 3.5. Nejdůležitější je pole Typ, které určuje typ zprávy a může nabývat těchto hodnot:

Hodnota	Význam
0x11	Dotaz směřovače: Jsou na LAN ještě nějací členové?
0x16	Požadavek na členství ve skupině
0x17	Opuštění skupiny
0x12	Požadavek IGMP verze 1 na členství ve skupině

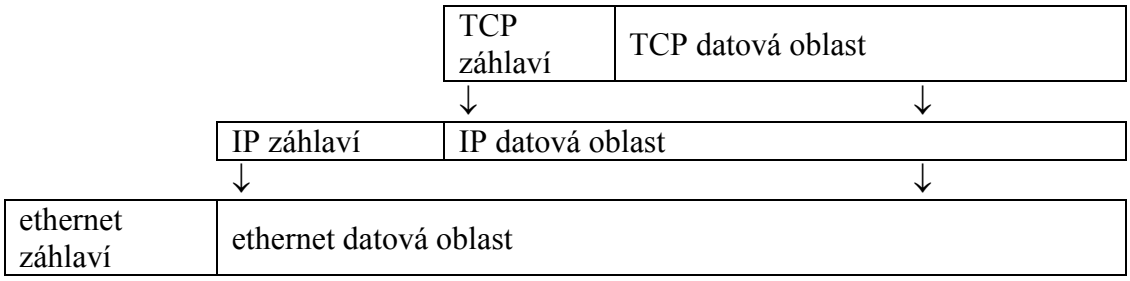
V políčku MRT (Maximum Response Time) je případná maximální povolená odezva, v poli Kontrolní součet je kontrolní součet a poslední čtyři bajty jsou vyhrazeny pro IP adresu skupiny. Více informací o protokolech ICMP a IGMP se lze dozvědět v [4] a [6].

3.4.3 Protokol TCP (Transmission Control Protocol)

Protokol TCP je proti protokolu IP protokolem vyšší vrstvy. Zatímco protokol IP přepravuje data mezi libovolnými počítači v Internetu, tak protokol TCP dopravuje data mezi dvěma konkrétními aplikacemi běžícími na těchto počítačích. Pro dopravu dat mezi počítači se využívá protokol IP. Protokol IP adresuje IP-adresou pouze síťové rozhraní počítače.

Protokol TCP je spojovanou službou (*connection oriented*), tj. službou která mezi dvěma aplikacemi naváže spojení – vytvoří na dobu spojení virtuální okruh. Tento okruh je plně duplexní (data se přenášejí současně nezávisle na sobě oběma směry). Přenášené bajty jsou číslovány. Ztracená nebo poškozená data jsou znovu vyžádána. Integrita přenášených dat je zabezpečena kontrolním součtem.

Konce spojení („odesílatel" a „adresát") jsou určeny tzv. číslem portu. Toto číslo je dvoubajtové, takže může nabývat hodnot 0 až 65535. U čísel portů se často vyjadřuje okolnost, že se jedná o porty protokolu TCP tím, že se za číslo napíše lomítko a název protokolu (tcp).



Tabulka 3.6 Zapouzdření TCP v IP datagramu na fyzické síti

0	4	10	16	31
Zdrojový port			Cílový port	
Sekvenční číslo				
Číslo oznámení				
délka	reservováno	Kódové bity	okno	
Kontrolní součet			urgent pointer	
options (volitelně)				doplnění

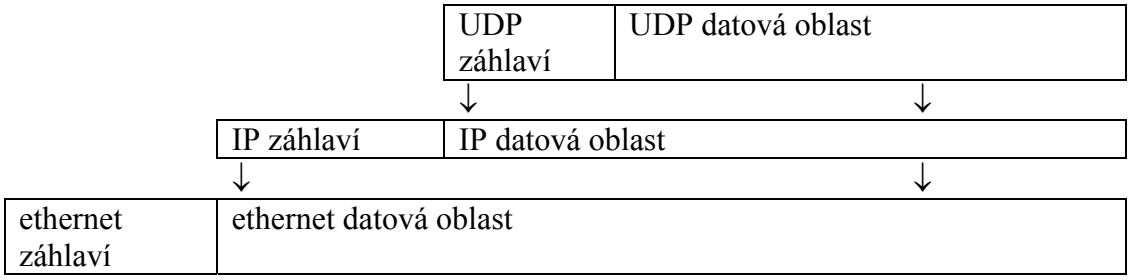
Tabulka 3.7 Hlavička TCP paketu

Více informací o protokolu TCP se lze dozvědět např. v [4].

3.4.4 Protokol UDP (User Datagram Protocol)

Protokol UDP je jednodušší alternativou protokolu TCP. Protokol UDP je nespojovaná služba (na rozdíl od protokolu TCP), tj. nenavazuje spojení.

UDP je jeden z jednoduchých transportních protokolů. Poskytuje tzv. „nespojovanou“ a nezabezpečenou službu dodávání uživatelských datagramů s použitím IP datagramů. Oproti ryzím IP datagramům má navíc schopnost rozlišit mezi různými procesy na cílovém počítači. Aplikace používající UDP je plně odpovědná za zabezpečení přenosu, včetně ztráty datagramů, jejich duplikace, příchod datagramu mimo pořadí a ztrátu kontaktu s cílem. Pro protokol UDP je jiná sada portů než pro protokol TCP (též 0 až 65535).



Tabulka 3.8 Zapouzdření UDP v IP datagramu na fyzické síti

0	16 31
UDP zdrojový port	UDP cílový port
UDP délka zprávy	UDP kontrolní součet

Tabulka 3.9 Hlavička UDP paketu

Položky PORT se používají k rozlišení výpočetních procesů čekajících na cílovém stroji na UDP datagramy. Položka zdrojový PORT je nepovinná; není-li použita, musí být 0. Jinak označuje číslo *portu*, na nějž má být zaslána případná odpověď.

Více informací o protokolu UDP se lze dozvědět např. v [4].

3.5 Zapouzdření paketů protokolu CIP

Zapouzdřující protokol definuje TCP (a také UDP) port, který podporují všechny EtherNet/IP zařízení. Každé zařízení má akceptovat nejméně 2 TCP připojení na port 44818 (0xAF12) a přijímat UDP pakety na UDP portu 44818. CIP paket přenášený pomocí protokolu TCP může být rozdělen do několika TCP paketů, v případě použití UDP však celá zpráva musí být v jediném UDP paketu, jelikož protokol UDP nemá možnost přeuspořádat došlé pakety (viz dále).

Příklad zapouzdřené zprávy:

ethernet záhlaví (14 bytů)	IP záhlaví (20 bytů)	TCP záhlaví (20 bytů)	Zapouzdřená zpráva č.1	Zapouzdřená zpráva č.2	CRC

Nebo

ethernet záhlaví (14 bytů)	IP záhlaví (20 bytů)	TCP záhlaví (20 bytů)	Zapouzdřená zpráva č.1	CRC

CRC je kontrolní součet rámce. Celková délka zapouzdřené zprávy **včetně hlavičky** je limitována na 65535 bytů s následující strukturou:

Struktura zapouzdřeného CIP paketu:

Struktura	Jméno pole	Datový typ	Hodnota pole
Záhlaví	Příkaz	UINT	
	Délka	UINT	
	Session handle	UDINT	
	Status	UDINT	
	Sender Context	Pole oktetů	
	Options (možnosti)	UDINT	
Datová část (struktura je závislá na typu příkazu)	Encapsulated data (zapouzdřená data)	Pole 0..65011 USINT	

Tabulka 3.10 Struktura zapouzdřeného CIP paketu

Délka zapouzdřené zprávy nemá překračovat maximální délku specifikovanou protokolem (TCP/UDP).

Více bytové zprávy se přenáší ve tvaru *little-endian*, na rozdíl od standardních síťových protokolů, které používají kódování *big-endian* (popis těchto kódování je uveden v seznamu zkratek a definic; jedná se o pořadí bytů (bitů) ve více bytových (bitových) zprávách).

Hlavička neobsahuje žádnou explicitní informaci o tom, zdali je zpráva požadavek či odpověď. Tatu informaci je možné zjistit buď implicitně (z kontextu, v jakém je zpráva generována) nebo explicitně (z obsahu zapouzdřeného paketu)

Kódování pole ***Příkaz***:

Kód	Název příkazu	Poznámka
0x0000	NOP	Může být zaslán pouze použitím TCP
0x0001	Vyhrazeno pro zpětnou kompatibilitu (legacy)	
0x0002 a 0x0003	Vyhrazeno pro zpětnou kompatibilitu (legacy)	
0x0004	ListServices (seznam služeb)	Může být zaslán použitím buď UDP nebo TCP
0x0005	Vyhrazeno pro zpětnou kompatibilitu (legacy)	
0x0006 až 0x0062	Rezerva pro budoucí rozšíření specifikace	
0x0063	ListIdentity (seznam identit)	Může být zaslán použitím buď UDP nebo TCP
0x0064	ListInterfaces (seznam rozhraní)	Volitelně (Může být zaslán použitím buď UDP nebo TCP)
0x0065	RegisterSession (registrace spojení)	Může být zaslán pouze použitím TCP
0x0066	UnRegisterSession (zrušení spojení)	Může být zaslán pouze použitím TCP
0x0067 až 0x006E	Vyhrazeno pro zpětnou kompatibilitu (legacy)	
0x006F	SendRRData	Může být zaslán pouze použitím TCP
0x0070	SendUnitData	Může být zaslán pouze použitím TCP
0x0071	Vyhrazeno pro zpětnou kompatibilitu (legacy)	
0x0072	IndicateStatus (indikuj status)	Volitelně (Může být zaslán pouze použitím TCP)
0x0073	Cancel (zrušit)	Volitelně (Může být zaslán pouze použitím TCP)
0x0074 až 0x00C7	Vyhrazeno pro zpětnou kompatibilitu (legacy)	
0x00C8 až 0xFFFF	Rezerva pro budoucí rozšíření specifikace	

Tabulka 3.11 Kódování pole příkaz

Pole Délka specifikuje délku dat ve zprávě v bytech. Je nulové pro zprávy bez dat. Celková délka zprávy je součet 24 bytové hlavičky a čísla v tomto poli.

Cíl generuje ***Session Handle*** v odpovědi příjemci na požadavek RegisterSession. Původce do tohoto pole vloží všechny předchozí zapouzdřené pakety zaslané tomuto cíli.

Hodnota v poli ***Status field*** udává, zda-li se cíli podařilo vykonat požadovaný zapouzdřený příkaz (přičemž nulová hodnota indikuje úspěšné vykonání příkazu). Ve všech požadavcích zaslaných původcem je pole *status field* nastaveno na nulu. V případě jiných hodnot je požadavek ignorován.

Pole Status field nabývá těchto hodnot:

Status kód	Popis
0x0000	Úspěšné vykonání příkazu
0x0001	Odesílatel vydal neplatný či nepodporovaný příkaz
0x0002	Nedostatek paměti příjemce pro vykonání příkazu
0x0003	Chybný formát nebo neplatná data
0x0004 až 0x0063	Vyhrazeno pro zpětnou kompatibilitu (legacy)
0x0064	Původce použil neplatný session handle pro zaslání zprávy
0x0065	Cíl přijal zprávu neplatné délky
0x0066 až 0x0068	Vyhrazeno pro zpětnou kompatibilitu (legacy)
0x0069	Nepodporovaná revize zapouzdřujícího protokolu
0x006A až 0xFFFF	Rezerva pro budoucí rozšíření

Tabulka 3.12 Kódování pole Status

Hodnota pole ***Sender context*** je přiřazena odesílatelem. Příjemce tuto hodnotu použije bez modifikace v odpovědi na požadavek.

Hodnota pole ***Options*** je odesílatelem nastavena na nulu. Příjemce kontroluje, zda-li je hodnota nulová; v jiných případech je paket zahozen. *Žádné speciální využití dosud nebylo specifikováno.*

Pole ***Command Specific Data*** má strukturu závislou na příkaze. Většina příkazů používá buď pevnou strukturu, nebo *common packet format*. Tento formát bude popsán později.

3.6 Popis příkazů

NOP

Příkaz NOP (No OPeration) slouží původci či cíli ke zjištění, zda-li je TCP spojení stále otevřené. Na tento příkaz není generována žádná odpověď. Příjemce tohoto příkazu ignoruje jakákoliv data obsažená v zaslané zprávě.

ListIdentity (seznam identit)

Původce spojení může použít tento příkaz k nalezení a identifikování potenciálních cílů. Tento příkaz se zasílá brodcastem pomocí UDP a nepotřebuje tedy vytvořené spojení.

Požadavek:

Struktura	Jméno pole	Datový typ	Hodnota pole
Header	Příkaz	UINT	ListIdentity (0x0063)
	Délka	UINT	0
	Session handle	UDINT	ignorováno
	Status	UDINT	0
	Sender Context	Pole oktetů	0
	Options	UDINT	0

Tabulka 3.13 Příkaz ListIdentity (požadavek)

Každý příjemce příkazu List Identity odpoví standardní hlavičkou a daty, datová část zprávy obsahuje informace o identitě cíle. Odpověď je zaslána na IP adresu, ze které byl broadcastový požadavek přijat. Datová část zprávy je v *common packet format*.

Odpověď:

Struktura	Jméno pole	Datový typ	Hodnota pole
Header	Příkaz	UINT	ListIdentity (0x0063)
	Délka	UINT	0
	Session handle	UDINT	Ignorováno
	Status	UDINT	0
	Sender Context	Pole oktetů	0
	Options	UDINT	0
Command specific data	Item Count	UINT	Počet následujících položek
		Struct of	Informace o rozhraní
		UINT	Kód typu
		UINT	Délka
		Array of byte	Data

Tabulka 3.14 Příkaz ListIdentity (odpověď)

ListInterfaces (seznam rozhraní)

Příkaz využívá původce spojení k identifikaci možného non-CIP komunikačního zařízení asociovaného s cílem. Není nutné vytvořené spojení (používá se UDP protokol).

Požadavek:

Struktura	Jméno pole	Datový typ	Hodnota pole
Header	Příkaz	UINT	ListInterfaces (0x0064)
	Délka	UINT	0
	Session handle	UDINT	ignorováno
	Status	UDINT	0
	Sender Context	Pole oktetů	0
	Options	UDINT	0

Tabulka 3.15 Příkaz ListInterfaces (požadavek)

Příjemce příkazu odpoví standardní hlavičkou a daty, datová část zprávy je ve formátu *common packet format* a obsahuje informace o non-CIP rozhraní asociovaném s cílem –

veřejně však nejsou definovány žádné položky odpovědi. Minimálně je vrácena položka Interface Handle (32 bitů), která je použita dalšími zapouzdřenými příkazy, například SendRRData.

Odpověď:

Struktura	Jméno pole	Datový typ	Hodnota pole
Header	Příkaz	UINT	List Interfaces (0x0064)
	Délka	UINT	0
	Session handle	UDINT	Ignorováno
	Status	UDINT	0
	Sender Context	Pole oktetů	0
	Options	UDINT	0
Command specific data	Item Count	UINT	Počet následujících položek
		Struct of	Informace o rozhraní
		UINT	Kód typu
		UINT	Délka
		Array of byte	Data

Tabulka 3.16 Příkaz ListInterfaces (odpověď)

RegisterSession (registrace spojení)

Když chce původce vytvořit nové spojení, zašle cíli příkaz RegisterSession.

Požadavek:

Struktura	Jméno pole	Datový typ	Hodnota pole
Header	Příkaz	UINT	Register Session (0x0065)
	Délka	UINT	4 byty
	Session handle	UDINT	0
	Status	UDINT	0
	Sender Context	Pole oktetů	Lib. context délky 8
	Options	UDINT	0
Command specific data	Protocol version	UINT	1
	Options flags	UINT	Session options jsou nastaveny na 0

Tabulka 3.17 Příkaz RegisterSession (požadavek)

Aby cíl indikoval, že zaregistroval původce spojení, zašle odpověď:

Struktura	Jméno pole	Datový typ	Hodnota pole
Header	Příkaz	UINT	Register Session (0x0065)
	Délka	UINT	4 byty
	Session handle	UDINT	Handle vygenerovaný cílem
	Status	UDINT	0
	Sender Context	Pole oktetů	Zachováno z požadavku
	Options	UDINT	0
Command specific data	Protocol version	UINT	1
	Options flags	UINT	Session options jsou nastaveny na 0

Tabulka 3.18 Příkaz RegisterSession (odpověď)

UnRegisterSession (zrušení spojení)

Pro ukončení spojení zašle původce či cíl tento příkaz. Po přijetí příkazu příjemce uzavře zapouzdřující TCP/IP spojení. Na příkaz není odpověď.

Požadavek:

Struktura	Jméno pole	Datový typ	Hodnota pole
Header	Příkaz	UINT	UnRegisterSession (0x0066)
	Délka	UINT	0
	Session handle	UDINT	Handle z odpovědi na RegisterSession
	Status	UDINT	0
	Sender Context	Pole oktetů	Jakýkoliv kontext
	Options	UDINT	0

Tabulka 3.19 Příkaz UnRegisterSession (požadavek)

ListServices (seznam služeb)

Příkaz ListServices se používá k rozpoznání, které třídy služeb cílové zařízení podporuje. Požadavek:

Struktura	Jméno pole	Datový typ	Hodnota pole
Header	Příkaz	UINT	ListServices (0x0004)
	Délka	UINT	0
	Session handle	UDINT	Ignored
	Status	UDINT	0
	Sender Context	Pole oktetů	Jakýkoliv kontext
	Options	UDINT	0

Tabulka 3.20 Příkaz ListServices (požadavek)

Odpovědí je standardní zapouzdřená zpráva složená z hlavičky a dat. Datová část obsahuje informace o podporovaných službách:

Struktura	Jméno pole	Datový typ	Hodnota pole
Header	Příkaz	UINT	ListServices (0x0004)
	Délka	UINT	0
	Session handle	UDINT	ignored
	Status	UDINT	0
	Sender Context	Pole oktetů	Zachováno z požadavku
	Options	UDINT	0
Command specific data	Item count	UINT	Počet následujících položek
	Target Items	STRUCT of	Informace o rozhraní
		UINT	Kód typu
		UINT	Délka
		UINT	Verze protokolu
		UINT	Capability flags (příznaky)
		Pole 16ti USINT	Jméno služby

Tabulka 3.21 Příkaz ListServices (odpověď)

Každá služba má jinou množinu příznaků *capability flags*:

Hodnota příznaku	Popis
Bity 0 až 4	Vyhrazeno pro zpětnou kompatibilitu (legacy)
Bit 5	Jestliže zařízení podporuje zapouzdření CIP paketů do TCP, je tento bit nastaven na 1, v jiných případech 0
Bity 6 až 7	Vyhrazeno pro zpětnou kompatibilitu (legacy)
Bit 8	Podpora CIP Class 0 nebo 1 UDP spojení
Bit 9 až 15	Rezerva pro budoucí rozšíření

Tabulka 3.22 Kódování příznaků Capability flags

Pole jméno služby je 16ti bytový, NULL zakončený ASCII řetězec, slouží pouze pro účely popisu.

SendRRData

Příkaz SendRRData přenáší zapouzdřený paket požadavku/odpovědi mezi původcem a cílem, příkaz zasílá původce. Pakety požadavku/odpovědi jsou zapouzdřeny datové části zprávy. SendRRData požadavek a odpověď se používá k zasílání UCMM zpráv.

Požadavek:

Struktura	Jméno pole	Datový typ	Hodnota pole
Header	Příkaz	UINT	SendRRData (0x006F)
	Délka	UINT	Délka datové části
	Session handle	UDINT	Handle vygenerovaný cílem
	Status	UDINT	0
	Sender Context	Pole oktetů	Lib. kontext, délky 8
	Options	UDINT	0
Command specific data	Interface handle	UINT	0 pro CIP
	Timeout	UINT	Timeout (časový limit)
	Zapouzdřený paket		...Common packet format

Tabulka 3.23 Příkaz SendRRData (požadavek)

Interface handle identifikuje komunikační interface, na který je požadavek směrován, je 0 pro CIP.

Při zapouzdřování CIP paketů se obvykle pole Timeout nastavuje na 0, jelikož CIP má vlastní timeout mechanizmus pro zprávy přenášené pomocí vytvořeného spojení (*connected messages*).

Odpověď:

Struktura	Jméno pole	Datový typ	Hodnota pole
Header	Příkaz	UINT	SendRRData (0x006F)
	Délka	UINT	Délka datové části
	Session handle	UDINT	Handle vygenerovaný cílem
	Status	UDINT	0
	Sender Context	Pole oktetů	Kontext odpovídající požadavku SendRRData
	Options	UDINT	0
Command specific data	Interface handle	UINT	0 pro CIP
	Timeout	UINT	Timeout (časový limit)
	Zapouzdřený paket		...Common packet format

Tabulka 3.24 Příkaz SendRRData (odpověď)

SendUnitData

Příkaz SendUnitData zasílá zapouzdřené zprávy, pro které již bylo vytvořeno spojení. Tento příkaz je možno použít, jestliže zapouzdřující protokol má svůj vlastní transportní mechanizmus typu bod-bod. Na tento příkaz se nezasílá odpověď, může být zaslán oběma konci TCP spojení.

Formát příkazu:

Struktura	Jméno pole	Datový typ	Hodnota pole
Header	Příkaz	UINT	SendUnitData (0x0070)
	Délka	UINT	Délka datové části
	Session handle	UDINT	Handle vygenerovaný cílem
	Status	UDINT	0
	Sender Context	Pole oktetů	Lib. kontext délky 8
	Options	UDINT	0
Command specific data	Interface handle	UINT	0
	Timeout	UINT	0
	Zapouzdřený paket		...Common packet format

Tabulka 3.25 Příkaz SendUnitData

3.7 Management spojení (session management)

Zapouzdřené spojení se skládá ze tří částí:

- navázání spojení
- udržování navázaného spojení
- uzavření spojení

Při navazování spojení se postupuje podle těchto kroků:

- původce otevře TCP/IP spojení s cílem na portu 0xAF12, případně na jiném definovaném portu)
- původce zašle příkaz RegisterSession cíli
- cíl zkontroluje verzi protokolu ve zprávě příkazu; v případě že cíl nepodporuje verzi protokolu původce, odpoví příkaze RegisterSession s odpovídajícím polem Status
- cíl přiřadí unikátní SessionID (identifikátor relace) a zašle RegisterSession původci

Spojení může ukončit jak původce, tak i cíl. Ukončení může proběhnou dvěma způsoby:

- Původce (či cíl) uzavře TCP spojení. Druhý konec spojení zjistí ztrátu TCP spojení a uzavře svoji stranu spojení.
- Původce (či cíl) zašle příkaz UnRegisterSession a čeká na uzavření TCP spojení. Odpovídající konec spojení potom uzavře svoji stranu TCP spojení. Odesílatel příkazu UnRegisterSession zjistí ztrátu TCP spojení a pak uzavře svoji stranu spojení.

Navázané spojení přetrvává, dokud nenastane jedna z těchto událostí:

- původce či cíl uzavře TCP spojení
- původce či cíl zašle příkaz UnRegisterSession
- TCP spojení je přerušeno

3.7.1 Vlastnosti TCP (informativně)

Transmission Control Protocol (TCP) je spolehlivý protokol orientovaný na spojení. Jestliže proces na kterékoliv straně spojení ukončí spojení, je TCP na druhé straně okamžitě informováno. Jestliže zpráva nemůže být doručena v přiměřeném čase, spojení se považuje za přerušené a odesílateli je oznámena chyba.

Jestliže původce spojení detekuje, že cíl uzavřel spojení nebo že spojení bylo přerušeno, považuje spojení s cílem za přerušené a uzavře spojení s cílem. Potom je navázáno nové spojení, pomocí kterého se pokračuje v komunikaci s cílem.

Ačkoli původce je informován, když druhá strana spojení uzavře, přerušené spojení je detekováno pouze při pokusu zaslat zprávu pomocí spojení. Ve většině případů je to zjištěno

díky rychlé výměně zpráv velmi rychle. Je však možné, že ani jedna strana spojení nemá žádnou zprávu k poslání po relativně dlouhou dobu.

TCP protokol podporuje tzv. *keep-alive processing*. Aplikace se může zeptat TCP, zda-li spojení zůstává navázáno během doby, kdy nejsou žádné zprávy k poslání. Je-li tato vlastnost povolena, TCP posílá keep-alive zprávy během doby, kdy je spojení neaktivní. Jestliže TCP zašle několik keep-alive zpráv a neobdrží odpověď, je spojení považováno za přerušené a aplikace je informována ihned po časovém vypršení aktuální zprávy.

Většina implementací TCP/IP retry/timeout processing nedeklaruje chybu spojení pokud zůstává nepoužito po dobu několika minut. To je vlastnost TCP protokolu, kterou zapnutí keep-alive processing nemodifikuje.

3.8 Common Packet Format (CPF – společný formát paketů)

Common Packet Format definuje standardní formát pro pakety protokolu, které jsou přenášeny zapouzdřujícím protokolem. Základním smyslem je vytvoření formátu, který bude vyhovovat budoucím typům paketů či adres.

Common Packet Format se skládá z pole *Item Count*, udávajícím počet následujících polí, *Address Item* a *Data Item*, případně mohou následovat další volitelné pole.

Jméno pole	Datový typ	Popis
Item count	UINT	Počet následujících polí
Address item	Item struct	Informace o adrese pro zapouzdřující protokol
Data item	Item struct	Zapouzdřený datový paket
Další volitelná pole		

Tabulka 3.26 Základní pohled na Common Packet Format

Pole *Address item* a *Data item* mají tuto strukturu:

Jméno pole	Datový typ	Popis
Type ID	UINT	Typ zapouzdřené položky
Délka	UINT	Délka dat v bytech
Data	proměnný	data

Tabulka 3.27 Struktura polí Address a Data item

Item ID jsou následující:

Číslo Item ID	Item type	Popis
0x0000	adresa	Null (použito pro UCMM zprávy) Indikuje, že není třeba směrování. Cíl je buď místní (ethernet), nebo jsou směrovací informace uloženy v položce *Data Item*
0x0001 – 0x000B		Vyhrazeno pro zpětnou kompatibilitu (legacy)
0x000C		ListIdentity odpověď

0x000D – 0x0083		Vyhrazeno pro zpětnou kompatibilitu (legacy)
0x0084 – 0x0090		Rezerva pro budoucí rozšíření
0x0091		Vyhrazeno pro zpětnou kompatibilitu (legacy)
0x0092 – 0x00A0		Rezerva pro budoucí rozšíření
0x00A1	adresa	Connection based (použito pro *connected messages*)
0x00A2 – 0x00A4		Vyhrazeno pro zpětnou kompatibilitu (legacy)
0x00A5 – 0x00B0		Rezerva pro budoucí rozšíření
0x00B1	Data	Connected Transport packet
0x00B2	Data	Unconnected message
0x00B3 – 0x00FF		Rezerva pro budoucí rozšíření
0x0100		ListServices odpověď
0x0101 – 0x010F		Vyhrazeno pro zpětnou kompatibilitu (legacy)
0x0110 – 0x7FFF		Rezerva pro budoucí rozšíření
0x8000	Data	Sockaddr_Info, původce → cíl
0x8001	Data	Sockaddr_Info, cíl → původce
0x8002		Sequenced Address item
0x8003 – 0xFFFF		Rezerva pro budoucí rozšíření

Tabulka 3.28 Kódování pole Item ID

Jeli *item ID* nulové, nenásledují za polem Délka žádná data. Nulová adresa dále neobsahuje žádné směrovací informace, používá se tehdy, obsahuje-li sám paket nezbytné směrovací informace. Nulová adresa se používá pro zprávy zasílané bez vytvořeného spojení (*unconnected messages*). Pole Délka je nastaveno na nulu.

Pro zprávy zasílané pomocí vytvořeného spojení je pole adresa nastaveno takto:

Jméno pole	Datový typ	Hodnota
Type ID	UINT	0x00A1
Délka	UINT	4
Data	UDINT	Identifikátor spojení

Tabulka 3.29 Pole adresa pro connected messages

Pro CIP transportní třídy 0 a 1 se používá *sequenced address*. Data obsahují identifikátor spojení a číslo sekvence:

Jméno pole	Datový typ	Hodnota
Type ID	UINT	0x8002
Délka	UINT	8
Data	UDINT	Identifikátor spojení
	UDINT	Číslo sekvence

Tabulka 3.30 Sequenced address

Formát pole Data je závislý na zapouzdřovaném protokolu. Formát pole data v případě protokolu CIP odpovídá požadavku a odpovědi objektu *Message Router* (pro *unconnected data*). Formát pole dat v případě zpráv zasílaných pomocí vytvořeného spojení je daný dotyčným paketem.

Unconnected data:

Jméno pole	Datový typ	Hodnota
Type ID	UINT	0x00B2
Délka	UINT	Délka unconnected message v bytech
Data	proměnný	Vlastní zpráva

Tabulka 3.31 Formát pole dat (explicitní zpráva)

Connected data:

Jméno pole	Datový typ	Hodnota
Type ID	UINT	0x00B1
Délka	UINT	Délka connected message v bytech
Data	proměnný	Přenášený paket

Tabulka 3.32 Formát pole dat (implicitní zpráva)

Sockaddr_info se používá pro zapouzdření informace o *socket address* nutné pro zaslání datagramů (*connected data*) mezi cílem a původce. Jsou zde oddělené položky pro každý směr komunikace.

Jméno pole	Datový typ	Hodnota
Type ID	UINT	0x8000 pro směr původce → cíl 0x8001 pro směr cíl → původce
Délka	UINT	16
Sin_family	INT	AF_INET = 2, big endian
Sin_port	UINT	Číslo TCP nebo UDP portu
Sin_addr	UDINT	IP adresa
Sin_zero	Pole USINT	0

Tabulka 3.33 Kódování položky Sockaddr_info

Struktura položky *Sockaddr item* je odvozena od specifikace Winsock verze 1.1.

3.9 Mapování explicitních a implicitních (I/O) zpráv na TCP/IP

Jestliže CIP paket prochází přes ethernet-TCP/IP síť, je potřeba tento CIP paket zapouzdřit do protokolu TCP/IP – viz kapitola 3.5. Zprávy na síti EtherNet/IP je možné rozdělit na zprávy bez vytvořeného spojení (***unconnected***) a na zprávy s vytvořeným spojením (***connected***). Formát zpráv bez vytvořeného spojení je popsán v příloze B, v dalším textu této kapitoly je popsáno, jakým způsobem se přenášejí jednotlivé transportní třídy s využitím protokolů TCP/UDP/IP.

3.9.1 Transportní třída CIP 0 a 1

Použití transportní třídy 0 a 1 a jejich definice je popsána v [1].

Pakety transportní třídy 0 a 1 se přenášejí pomocí UDP a Common Packet Format popsaným v podkapitolách 3.5 a 3.8. Pakety pro multicast spojení se přenášejí IP multicastem.

3.9.2 Vlastnosti spojení třídy 0 a třídy 1

Jelikož ethernet nemá mechanismus pro zasílání *rozvrhovaných zpráv*, je třeba mít na zřeteli několik důležitých vlastností třídy 0 a třídy 1:

Na síti ethernet je možné, například díky nadměrným kolizím, že je paket třídy 0 nebo 1 ztracen (např. poškozen při kolizi). Dle definice třída 0 a 1 negarantují doručení každého paketu. Proto producent zasílá data určitou rychlostí, tzv. API (*Application Packet Interval* – aplikační interval paketu). Jestliže je ztracen paket třídy 0 nebo 1, konzument obdrží další ze zaslaných paketů od producenta.

Kolik paketů se může ztratit je specifikováno na úrovni aplikace. Ethernet není vhodný pro aplikace, kde není možné tolerovat ztrátu ani jediného paketu.

Ztrátu paketu je možno detekovat v transportní třídě 1 pomocí *CIP sequence number.* Ve třídě 0 aplikace nemůže zjistit ztrátu paketu, jelikož třída 0 nevyužívá sequence number.

3.9.3 Connection Timeout (časový limit spojení)

Když se ztrácí příliš mnoho paketů, je mechanizmus časového limitu spojení zpětnou vazbou pro aplikaci. *Connection timeout* se odvozuje na základě požadovaného RPI (Requested Packet Interval – požadovaný interval paketu) a *Connection Timeout Multiplier* (násobitel). Jestliže není paket doručen v čase RPI krát násobitel, spojení je přerušené. Například je-li RPI 50ms a Connection Timeout Multiplier je 4, spojení je přerušeno jestliže není obdržen paket do 200 ms.

3.9.4 Transportní třída CIP 2 a 3

Transportní třída CIP 2 a 3 se přenáší přes TCP spojení použitím zapouzdřujícího protokolu popsaného v kapitole 3.5.

Pomocí jediného TCP spojení může být přenášeno mnoho CIP spojení. V implementaci není třeba specifikovat počet CIP spojení na jedno TCP spojení, je ale možné definovat horní mez. Díky tomu, že jednou ze základních vlastností TCP je full-duplex, CIP spojení ve směru původce→cíl a cíl→původce používají stejné TCP spojení.

3.9.5 Transportní třída CIP 4 až 6

Zapouzdřující protokol popsaný v kapitole 3.5 se pro transportní třídy 4 a 6 nepoužívá.

3.10 Síťový identifikátor spojení

Pro ethernet TCP/IP spojení je síťový identifikátor spojení (*Network connection ID*) 32bitový identifikátor, který identifikuje komunikující zařízení. *Network Connection ID* není děleno do více polí.

Network Connection ID musí být unikátní – stejné ID lze použít znovu jen jestliže spojení s tímto ID vypršelo (timed-out) nebo bylo uzavřeno. Stejné Network Connection ID nelze znovu použít pokud je pravděpodobnost, že pakety s tímto Network Connection ID jsou stále přítomny na síti.

Existují dvě metody, jak zajistit unikátní NC ID.

3.10.1 Použití incarnation ID

Network Connection ID	
Incarnation ID (16 bitů)	Connection Number (16 bitů)

Incarnation ID je 16ti bitový identifikátor, který každé zařízení generuje při vytvoření spojení. Po dobu, kdy je zařízení zapnuto, zůstává Incarnation ID stejné. Při každém zapnutí zařízení je třeba Incarnation ID vygenerovat znovu.

Jednou z možností jak generovat Incarnation ID je použití paměti, ve které je Incarnation ID uloženo. Po zapnutí zařízení odtud přečte svoje Incarnation ID a hodnotu uloženou v paměti inkrementuje pro použití při dalším zapnutí.

Další možností je generovat Incarnation ID jako pseudo-náhodné číslo při zapnutí zařízení. Tento přístup však vyžaduje opatrnost: Podle definice pseudo-náhodného čísla je nenulová pravděpodobnost, že vygenerované číslo bude stejné jako některé předchozí. Nicméně při rozumném přístupu je tato pravděpodobnost dostatečně malá.

Zařízení jako například stolní PC mohou použít jako Incarnation ID hodnotu systémových hodin – doba, po kterou jsou jednotlivá zařízení zapnuta se dostatečně liší. U vestavěných zařízení to však není příliš rozumné, jelikož firmware obecně spouští přesně stejnou sekvenci instrukcí při každém startu.

3.10.2 Pseudo-náhodné connection ID

Network Connection ID	
Pseudo-náhodné číslo (16 bitů)	Connection Number (16 bitů)

Pseudo-náhodné číslo je 16ti bitové číslo generované použitím příslušného generátoru pseudo-náhodných čísel.

S tímto přístupem každé zařízení generuje pseudo-náhodné číslo vždy, když je NC ID potřeba. Generátor pseudo-náhodných čísel by měl využívat *strong mixing function*, jako

například MD5[2]. Pro zajištění jedinečnosti Connection ID je doporučené využít jako vstup MD5 tyto hodnoty:

- Vendor ID, sériové číslo
- Obsah struktury sockaddr_info
- Hodnotu systémových hodin

3.11 Založení spojení ve transportní třídě 0 a 1

K založení spojení je určena služba Forward_open, která je přenášena zapouzdřeným příkazem SendRRData. Producent a konzument si sdělí čísla UDP portů a multicastovou IP adresu (pro multicast spojení), která jsou nutná pro zaslání dat v CIP transportní třídě 0 a 1. Položka sockaddr_info se používá k zakódování čísla UDP portu a IP multicast adresy. Použití této položky však záleží na tom, zda-li jde o spojení bod-bod nebo multicast, a také na tom, které zařízení je producentem.

3.11.1 Spojení multicast

Producent zvolí multicastovou IP adresu, na kterou bude zasílat *connected data*. IP adresa a číslo UDP portu (pouze port registrovaný IANA – 0x08AE) je zakódováno do položky Sockaddr_info a ta je zaslána službou Forward_open nebo Forward_open_reply.

3.11.2 Spojení bod-bod

Konzument zvolí UDP port na který budou zaslána *connected data*. Číslo portu může být registrované (0x08AE), nebo jiné. Číslo portu je zakódováno do položky Sockaddr_info a zasláno službou Forward_open nebo Forward_open_reply.

Položka Sockaddr_info je umístěna za polem dat v příkazu SendRRData.

3.11.3 Úvod do problematiky multicastingu

Pro snazší popis použití IP multicastu v dalších odstavcích zde uvádím několik faktů o multicastingu.

Na linkové vrstvě je možno multicasting využívat. Např. ethernet má polovinu adresového prostoru vyhrazenou pro multicastové skupiny. Jedná se o adresy s nastaveným nejnižším bitem v prvním bajtu MAC adresy. Multicastová adresa pak představuje číslo multicastové skupiny. Síťové zařízení je možné nastavit tak, aby přijímalo rámce s cílovou adresou jistých skupin a od té doby se budou tyto rámce přijímat a zpracovávat. IP multicasting je abstrakcí multicastingu na linkové vrstvě – síťové rozhraní má možnost se

[2] MD5 je jednocestná hešovací funkce vytvořená profesorem Ronaldem L. Rivestem v roce 1991. Algoritmus MD5 vytváří z libovolně dlouhého vstupu 128 bitů dlouhý mesage digest (výtah). Je odhadováno, že je prakticky nemožné vytvořit dvě zprávy, které by měly stejný message digest nebo vytvořit zpětně zprávu ze známého digestu. Smyslem algoritmu MD5 je vytvoření elektronického podpisu (ověření integrity dat).

začlenit do multicastové skupiny a poté bude dostávat datagramy určené pro tuto skupinu. IP multicasting má tyto vlastnosti:

1) Skupinová adresa je třídy D[3]. Několik multicastových adres je vyhrazeno pro konkrétní účely, další jsou volné k osobnímu použití. Adres je 2^{28}, není tedy třeba se obávat jejich nedostatku.
2) Hostitel se může do skupiny libovolně přihlašovat a ze skupiny odhlašovat. Může být také členem více skupin.
3) Pokud použitá linková vrstva nepodporuje multicasting, je třeba IP multicast simulovat pomocí broadcastu nebo unicastu.

Multicastové adresové schéma

Pro multicastové účely jsou vyhrazeny IP adresy třídy D (224.0.0.0 až 239.255.255.255). Část adresového prostoru třídy D jsou tzv. *dobře známé skupiny*. Ty jsou permanentně přiřazeny k jistým účelům a existují, i když do nich není přihlášen zrovna žádný člen. Zbytek je volný k použití pro tzv. *přechodné multicastové skupiny*, které se vytvoří, když se do nich někdo přihlásí a zaniknou, když se odhlásí poslední člen.

Permanentně přiřazené známé skupiny jsou 224.0.0.0/23 pro směrování a údržbu skupin.

239.192.0.0 do 239.251.255.255 je privátní prostor adres pro jednu organizaci.

239.252.0.0 do 239.255.255.255 je privátní prostor adres pro jednu lokalitu

Speciální významné adresy jsou: 224.0.0.1 – všechny systémy na lokální síti a 224.0.0.2 – všechny směrovače na lokální síti.

Multicastová IP adresa je určena pouze pro cíl, nikdy se tedy neobjeví jako zdrojová adresa.

Multicastový obzor (IP multicast scoping) slouží k určení hranic skupiny; znamená jak dalece je daný multicastový požadavek šířen sítí. IP používá dvě techniky ke kontrole hranic. Jedna z nich je TTL bajt v hlavičce IP datagramu (tzv. *TTL scoping*):

Tato metoda je založena na postupné dekrementaci pole TTL v hlavičce IP paketu. Při každém „hopu“ (průchodu směrovačem na síti s protokolem IP) je pole sníženo o 1; směrovače jsou konfigurovány tak, aby nepropouštěli dál pakety s definovaným a nižším polem TTL.

Druhá metoda je známá jako *administrativní obzor*. Ta zakazuje internetovým směrovačům přeposílat ty pakety, které mají cílovou adresu organizace (239.192.0.0 až 239.251.255.255) nebo lokality (239.252.0.0 až 239.255.255.255).

---

[3] IP adresy byly rozděleny do pěti tříd v závislosti na typu použití a dle velikosti složek IP adresy, které udávají tzv. *netid* (číslo sítě) a *hostid* (číslo stroje uvnitř této sítě) takto:

Třída A: **1.0.0.0 až 127.255.255.255**
Třída B: **128.0.0.0 až 191.255.255.255**
Třída C: **192.0.0.0 až 223.255.255.255**
Třída D: **224.0.0.0 až 239.255.255.255** multicastové adresy
Třída E: **240.0.0.0 až 247.255.255.255** rezerva pro budoucí použití

Přiřazování multicastových IP adres

Adresový prostor multicastových IP adres je administrován organizací IANA. Ta ale pro EtherNet/IP nevyhradila žádné speciální adresy. V současnosti proto existuje jediná možnost – aplikace si vybere adresu z adresového prostoru vyhrazeným IANA s ohledem na možné kolize s jiným aplikacemi. Dále se v současnosti vyvíjí několik další protokolů a metod přidělování multicast IP adres, jednou z nich je např. MDHCP (Multicast Address Allocation Based on the Dynamic Host Configuration), který je podobný známému DHCP.

3.11.4 Přiřazování multicastových IP adres a spojení

Je doporučeno [2], aby producenti používali unikátní multicastovou IP adresu pro každé aktivní multicastové spojení. V závislosti na implementaci může toto doporučení snížit počet operací třídění na straně konzumenta. Další výhodou může být vyrovnanější obsloužení příchozích dat z několika spojení konzumentem.

Jelikož pro multicast spojení není třeba unikátní multicastová IP adresa, konzument musí být schopen zvládnout situaci, kdy přicházejí pakety z několika multicastových spojení na stejnou multicastovou IP adresu. Konzument musí tedy být schopen pakety třídit na základě Connection ID a zdrojové IP adresy.

Po přijetí odpovědi Forward_open_reply se konzumující zařízení přihlásí do vybrané multicastové skupiny, aby mohlo dostávat datagramy určené pro tuto skupinu.

3.12 Connected data v transportní třídě 0 a 1

Podle definice (CIP Common Specification – [1] a [2]) transportních tříd 0 a 1 tyto třídy nedetekují pakety doručené mimo pořadí. V třídě 0 je každý paket brán jako nová data. Ve třídě 1 jsou detekovány pouze duplicitní pakety. Jestliže přijde nový paket a sekvenční číslo paketu je odlišné od předchozího, je nový paket brán jako nová data i když je sekvenční číslo menší než předchozí (pakety přišly přeházené).

V transportních třídách 0 a 1 se pro přenos *connected data* používá protokol UDP. Zde není žádná garance, že pakety dorazí ve stejném pořadí. Jestliže se odesílatel a příjemce nachází na stejné podsíti, pakety většinou přijdou ve správném pořadí. Prochází-li však paket několika směrovači, je možné, že každý paket půjde jinou cestou a pakety tak přijdou v nesprávném pořadí.

Datový formát pro transportní třídy 0 a 1 je následující:

Název pole	Typ	Hodnota
Item Count	UINT	2
Type ID	UINT	0x8002
Délka	UINT	8
Address Data	UDINT	Connection ID (z Forward_open reply)
	UDINT	Sequence number
Type ID	UINT	0x00B1 (connected data type)
Length	UINT	Počet bytů následujícího paketu
Data		Paket třídy 0 nebo 1

Tabulka 3.34 Datový formát transportních tříd 0 a 1

3.13 Třídění přicházejících I/O dat (connected data)

Zařízení, která přijímají *connected data* třídy 0 a 1 musí třídit pakety na základě Network Connection ID a IP adresy. Je to nutné z těchto důvodů:

- Není žádný garantovaný mechanismus k potlačení používání stejné multicastové IP adresy více zařízeními. Následkem toho může zařízení přijmout multicast *connected data* ze zařízení, které nemá vytvořené spojení.
- Zařízení může použít stejnou multicastovou IP adresu pro více multicast spojení třídy 0 a 1.
- Předcházení konfliktům Network Connection ID.

Po vytvoření spojení třídy 0 a 1 si původce a cíl uloží Network Connection ID, na kterém cíl bude přijímat *connected data* společně s IP adresou zařízení na druhém konci spojení. Když přijme zařízení *connected data*, ověří, zda-li Network Connection ID je platné pro IP adresu odesílatele. Jestliže ne, paket je zahozen.

3.14 Dočasný multicastový obzor (multicast scoping strategy)

Do doby, než bude vyvinuta architektura administrativního obzoru a přijata za standard, budou zařízení zasílající data ve třídě 0 a 1 používat TTL scoping s limitem šíření multicast datagramů na jednu podsíť. Proto pro komunikaci zařízení z různých podsítí je třeba spojení typu bod-bod. To fakticky znamená, že nelze použít přímou I/O komunikaci (*direct I/O communication*) mezi zařízeními v různých podsítích (tj. komunikace přes směrovač).

3.15 Dočasná alokační strategie

Pro multicast spojení je doporučena síť 239.192.0.0/14 (lomítkem a číslem za IP adresou se zkráceně označuje maska podsítě; číslo určuje počet jedniček v masce, tj. maska podsítě je 255.252.0.0 – jde tedy o privátní adresový prostor pro jednu organizaci, jak bylo zmíněno dříve). Prvních 256 adres je použito pro relativní přiřazení. Zařízení zasílající multicast data má multicast IP adresy buď nakonfigurované uživatelem, nebo si vybere multicast IP adresy podle následujícího algoritmu:

Každé zařízení smí použít 32 multicast adres v závislosti na své IP adrese: Zařízení 1 použije 239.192.1.0 až 239.192.1.31, zařízení 2 použije 239.192.1.32 až 239.192.1.63 a tak dále. Pro zajištění, že multicast IP adresa je v IP rozsahu organizace, použije se pouze 10 nejnižších bitů z adresy zařízení, jestliže maska podsítě tvoří víc než 10 bitů. Nepoužívá-li se maska podsítě, použije se část IP adresy, která tvoří adresu počítače. Právě u zařízení s IP adresami z třídy A a B (kde je použito právě jen 10 spodních bitů) je možnost, že jiné zařízení použije stejnou multicast IP adresu – proto je třeba přijaté multicast pakety třídit, jak bylo popsáno výše.

3.16 Závěr

Pro správnou funkci sítě EtherNet/IP pro řídicí účely v průmyslu je také důležitý určitý stupeň izolace zóny sítě používané pro řízení a zóny sítě ethernet využívané pro informační účely. V této kapitole byly popsány používané protokoly a jejich využití protokolem CIP. Informace zde uvedené je tedy možné využít právě pro konfiguraci zařízení zajišťujícího izolaci zóny sítě ethernet používané pro řízení a zóny pro ostatní informační komunikaci v průmyslovém podniku.

Zařízení zajišťující toto oddělení se většinou nazývá *firewall* (požární ochranná zeď). Konfigurace a správa takovýchto zařízení je většinou poměrně komplikovaná a vyžaduje kvalifikovaný a vyškolený personál. *Firewall* může být realizován buď standardním počítačem PC s více ethernet rozhraními, nebo se využívá specializovaného hardwaru určeného pro tyto účely. Při konfiguraci *firewall*u je třeba zadat minimálně IP adresy zařízení, které mají povolenu komunikaci, které zařízení ve které zóně může být inciátorem komunikace se zařízením v jiné zóně a jaké cílové UDP a TCP porty jsou povoleny při komunikaci mezi zónami. Dalším parametrem konfigurace může být zákaz určitých komunikačních protokolů; často bývá zakázán protokol ICMP, pomocí kterého je možno zjistit existenci a stav zařízení na síti.

4 Fyzická vrstva sítě EtherNet/IP

4.1 Úvod

V průmyslovém prostředí se požadavky na fyzické médium a instalace sítě EtherNet/IP mohou lišit od požadavků v kancelářském prostředí. Požadavky na odolnost média a instalace sítě EtherNet/IP v průmyslovém prostředí mohou zahrnovat: odolnost proti zvýšenému elektromagnetickému rušení, izolace, chemická odolnost, otřesy, vibrace a velký rozsah pracovních teplot.

Pro instalaci sítě EtherNet/IP je v zásadě možné použít dvě skupiny fyzických médii:

- Komerčně používaná média (UTP/STP, RJ 45, koaxial)
- Průmyslové médium EtherNet/IP

Komerčně používaná média byla primárně vyvinuta pro informační systémy. Ve většině případů tyto média spolehlivě postačují pro provoz při rychlosti přenosu 10Mb/s. Testování společnosti ODVA/CI však ukázalo, že pro spolehlivý provoz při rychlosti 100Mb/s v zašuměném a na vnější podmínky náročném průmyslovém prostředí je třeba komerčně používaná média „vylepšit“ – tímto vylepšeným médiem je průmyslové médium EtherNet/IP (Industrial EtherNet/IP Media).

4.2 Komerčně používaná média

Ačkoliv je možno tyto média použít, může být jejich nasazení příčinou nevyhovujícího výkonu v aplikacích průmyslového řízení.

Standardní komerčně používaná média musí splňovat tyto podmínky:

- Technologie stíněné nebo nestíněné kroucené dvoulinky (TP – twisted pair), které splňují standard ANSI/TIA/EIA-568 B.2.
- Popis elektrických signálů je uveden v normě IEEE 802.3u/TP-PMD.
- Konektory se používají typu RJ45, které jsou de facto standardem. Ty jsou popsány v stejné normě jako TP medium.
- Celková délka kabelu je limitována na 90m (100m včetně patch kabelů).

4.3 Průmyslové médium EtherNet/IP

Průmyslové médium EtherNet/IP se od standardních komerčně používaných médií liší především v izolaci a provedení konektorů RJ-45. Elektromechanické požadavky na průmyslové kabely pro síť EtherNet/IP jsou uvedeny v příloze C.1. Dalším rozdílem je omezení šířky pásma kabelu.

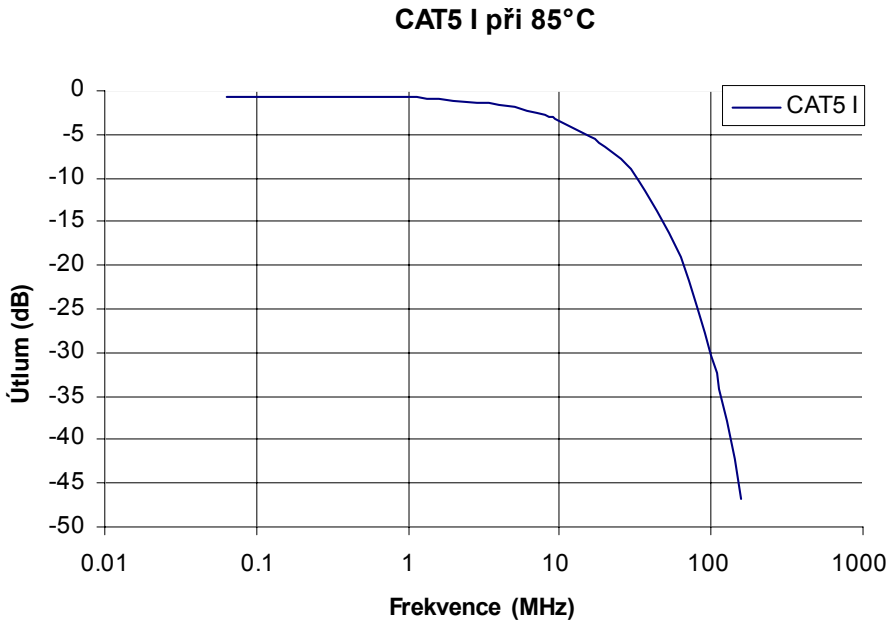
Pro zvýšení výkonu v prostředí s vysokým elektromagnetickým rušením je výhodné, aby kabel měl omezenou šířku přenosového pásma. Vhodnou metodou pro omezení šířky pásma šumu je zvýšení útlumu v pásmu nad 65,5MHz. Útlum stíněného i nestíněného TP kabelu je dán rovnicí:

$$Útlum_{UTP}(f) = \frac{-\left(K_0\sqrt{f} + K_1 f + \frac{K_2}{\sqrt{f}}\right)}{100}$$

Rovnice 4.1 Útlum UTP kabelu

Konstanty K_0 až K_2 jsou pro standardní CAT 5E kabel [K_0 K_1 K_2] = [1.967 0.023 0.050], pro průmyslovou variantu CAT 5I [K_0 K_1 K_2] = [0 0.3 0.1].

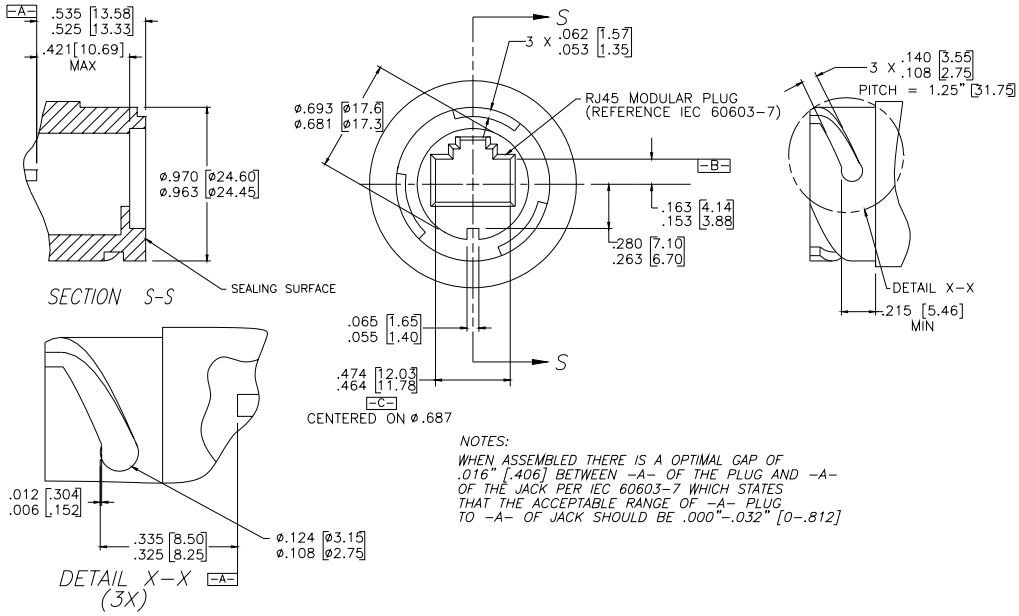
Průběh této funkce je zobrazen na obrázku:



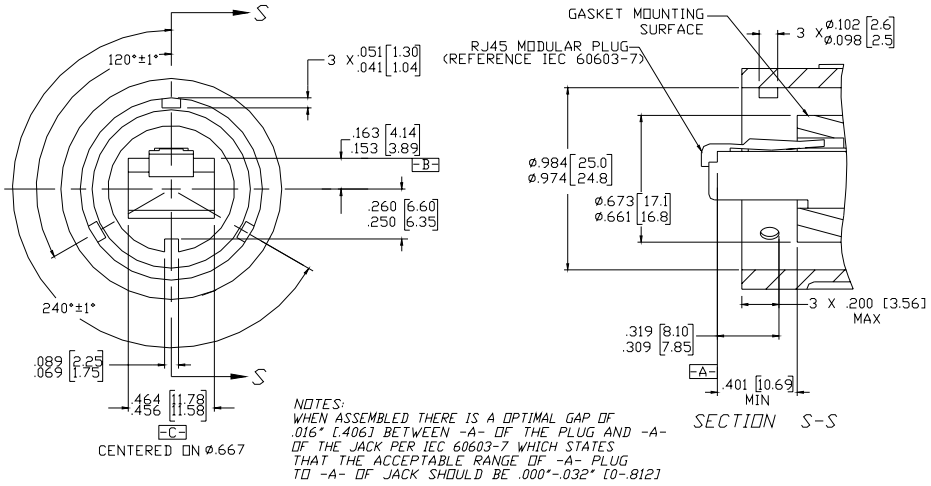
Obrázek 4.1 Útlum průmyslové modifikace UTP kabelu

4.4 Průmyslové konektory RJ-45

Konektory se používají buď netěsněné, nebo těsněné s krytím IP67. Těsněný konektor i zásuvka jsou kompatibilní se standardními konektory RJ45, jediným rozdílem je mechanismus zámku, který zvyšuje bezpečnost instalace.



Obrázek 4.2 Upravená zásuvka RJ45 pro průmyslové prostředí



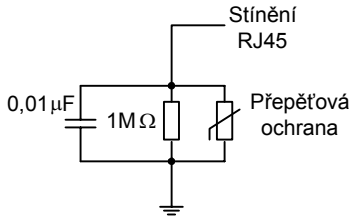
Obrázek 4.3 Upravený konektor RJ45 pro průmyslové prostředí

4.5 Propojení stínění se zemí

4.5.1 Síťové prvky

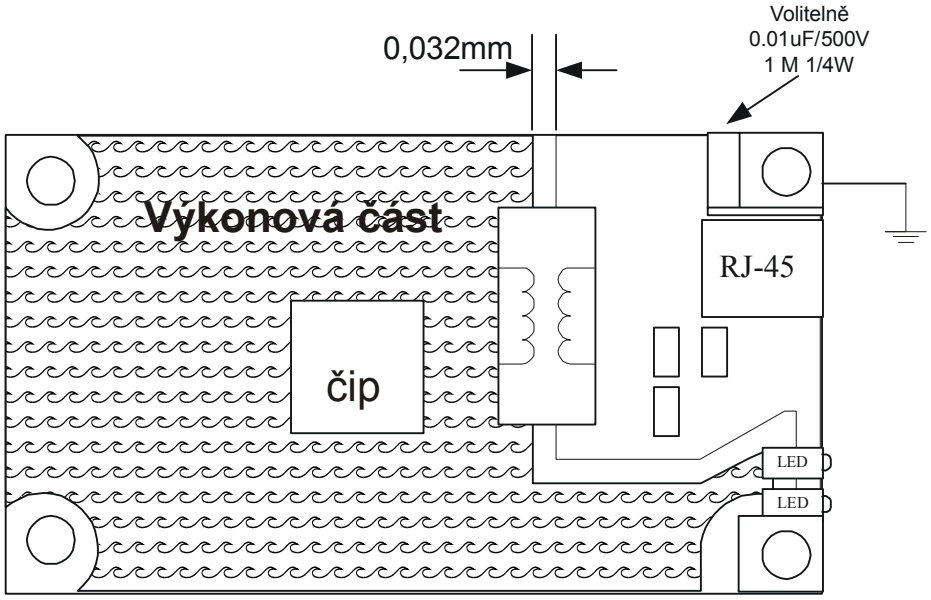
Stínění kabelů na straně síťových zařízení (přepínače, huby, mosty, směrovače) musí být ukončeno takto:

1) Stínění komunikačního kabelu musí být ukončeno přímým spojením se zemí pouze na jedné straně kabelu (a to na straně aktivního síťového prvku).
2) Pokud je z jakéhokoliv důvodu propojení stínění se zemí i na straně EtherNet/IP zařízení, je tak třeba učinit přes paralelní kombinaci RC zobrazené na obrázku 4.4.



Obrázek 4.4 Schéma spojení stínění a zemnění

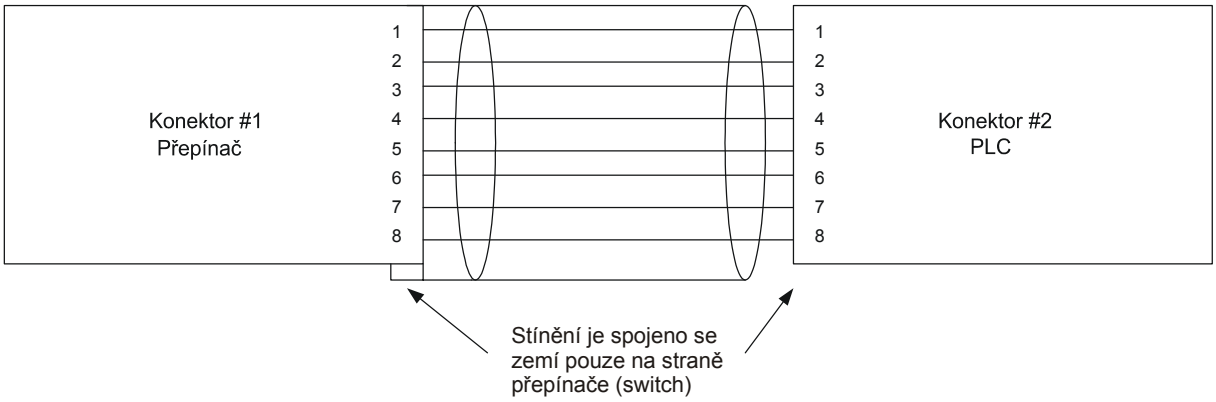
V pohledu na nákres celé komunikační desky se toto spojení nachází v místě označeném šipkou.



Obrázek 4.5 Komunikační deska

4.5.2 Řídicí zařízení

Aby nevznikaly zemnící smyčky[4] je třeba, aby stínění bylo spojeno se zemí pouze na jedné straně, a to na straně aktivních síťových prvků (přepínače, směrovače atd.). Spojení stínění se zemí na straně řídicích zařízení (PLC, senzory a pod.) je možné jen pomocí paralelní kombinace RC. Pokud je konektor (zásuvka) RJ-45 spojen přímo se zemí (díky konstrukci zařízení), nesmí být stínění na zástrčce RJ-45 připojeno.



Obrázek 4.6 Schéma připojení stínění STP kabelu

4.6 Použití optických kabelů

Pro použití komunikační sítě EtherNet/IP v prostředích, kde hrozí výbuch – například v oblasti zpracování plynu nebo chemickém průmyslu, je možno použít namísto tradičních metalických médií optických kabelů.

Konektory je možno použít typu SC, ST nebo MT-RJ vyhovující normě ANSI/TIA/EIA 568-B.3. Optický kabel může být jednovidový nebo vícevidový.

4.7 Závěr

Pro použití sítě EtherNet/IP (a ethernetu vůbec) v průmyslovém prostředí je třeba jiná, a to vyšší úroveň jak samotných komunikačních zařízení, tak i použitých kabelů. Výskyt linkové kolize při použití ethernetu v kancelářském prostředí je téměř nepostřehnutelný, avšak v některých řídicích systémech může být tato chyba neakceptovatelná. Pro zajištění výkonu srovnatelného s deterministickými sítěmi (např. ControlNet) je na síti ethernet třeba minimalizovat síťový provoz. Minimalizace provozu vyžaduje optimalizaci komunikační sítě, která je provázena správnou instalací, výběrem vhodných materiálů, minimalizací vnějšího elektromagnetického rušení a zlepšením elektrických vlastností systému kabeláže.

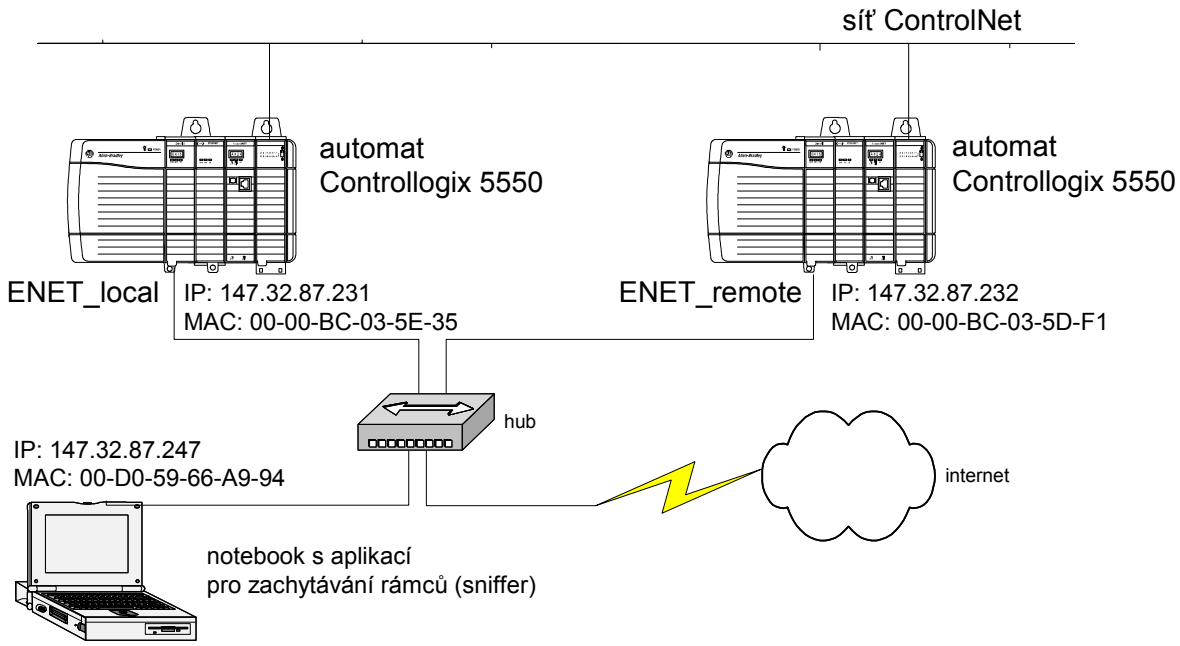
[4] Zemnící smyčky vznikají spojením uzemňovacích bodů, například stínění, na více bodech se zemí. Pak může nastat situace, kdy stíněním protéka určitý elektrický proud, což má nepříznivý vliv na vlastní funkci stínění.

Minimalizace elektromagnetického rušení je důležitá zejména při použití sítě ethernet při rychlosti 100 Mb/s. Více informací o použití ethernetu v průmyslu a zvláště o požadavcích na fyzickou vrstvu je možné se dozvědět v [8].

5 Analýza některých typů komunikace na síti EtherNet/IP

V této části popisuji testovací aplikace, které jsem použil pro analýzu některých typů komunikace na síti EtherNet/IP. Testovací systém se skládal ze dvou automatů ControlLogix 5550 se třinácti-slotovým šasi pro rozšiřující moduly. Každé šasi obsahovalo mimo jiných modulů dva komunikační moduly: jeden modul pro síť ControlNet – ControlNet Interface Module 1756-CNBR a druhý modul pro síť EtherNet/IP – ControlLogix Ethernet Bridge Module 1756-ENET/B.

Síť ControlNet, ke které byla připojena jedna stanice PC s konfiguračním software RSLogix5000, byla použita pouze pro konfiguraci a ovládání systému. Pro přenos aplikačních dat mezi oběma automaty byla použita síť EtherNet/IP. Moduly pro síť ethernet byly pomocí UTP kabelů spojeny do hvězdicové topologie, jedním z vrcholů hvězdy byl přenosný počítač, na kterém byla nainstalována aplikace pro zachytávání rámců sítě ethernet. Jako síťový koncentrátor (centrální prvek) byl použit hub (SMC). Pomocí tohoto prvku byl celý systém připojen k počítačové síti katedry řídicí techniky a do celosvětové sítě Internet. Schéma sítě je zobrazeno na obrázku 5.1.



Obrázek 5.1 Schéma sítě

5.1 Úvod do problematiky monitorování sítě

Monitorováním sítě se rozumí zachytávání linkových rámců v síti a jejich ukládání do paměti. Potom je možné prohlížet obsah jednotlivých rámců. Moderní programy pro monitorování sítě mají zobrazování rámců dosti propracované; jejich hlavní předností je

schopnost oddělit a rozpitvat jednotlivé položky záhlaví síťových protokolů – ethernet, IP, TCP, UDP a další. V operačním systému Windows NT/2000/XP je možno použít širokou škálu komerčně dostupných programů, nejznámějším je asi MS Network Monitor, který je v několika modifikacích distribuován spolu s operačními systémy firmy Microsoft. Velmi známým názvem pro tento druh programů je *sniffer*. Toto slovo, odvozeného od anglického slova *sniff* – čenichat, vyjadřuje možnosti těchto programů. Těmi jsou zjištění nezabezpečených hesel uživatelů sítě (jedná se například o použití programů ftp nebo telnet).

Na lokálních sítích (LAN) mohou být zachycovány buď pouze rámce adresované stanici, na které běží *sniffer*, nebo veškerý provoz segmentu lokální sítě. V druhém případě je nutné, aby byla síťová karta přepnuta do tzv. promiskuitního módu. Tento mód způsobuje, že síťová karta neakceptuje pouze rámce jí adresované, ale všechny rámce na segmentu.

Podstatné při monitorování sítě je to, že je možné zachytit ***pouze*** veškerý síťový provoz ***segmentu*** lokální sítě. Segment lokální sítě přepínaných sítí je omezen pouze na spojení stanice – přepínač (*switch*). Proto při monitorování přepínané sítě musí být na přepínači vyhrazen jeden port (fyzický), který je speciálně nastaven – je na něj kopírován provoz i z portu, na kterém chceme monitorovat provoz (tzv. *mirroring*). K tomuto portu je pak připojena stanice pro monitorování sítě. Pokud je jako síťový koncentrátor použit hub, je na všechny jeho porty opakován provoz z ostatních portů a lze tedy odposlouchávat provoz mezi ostatními stanicemi připojenými k tomuto hubu.

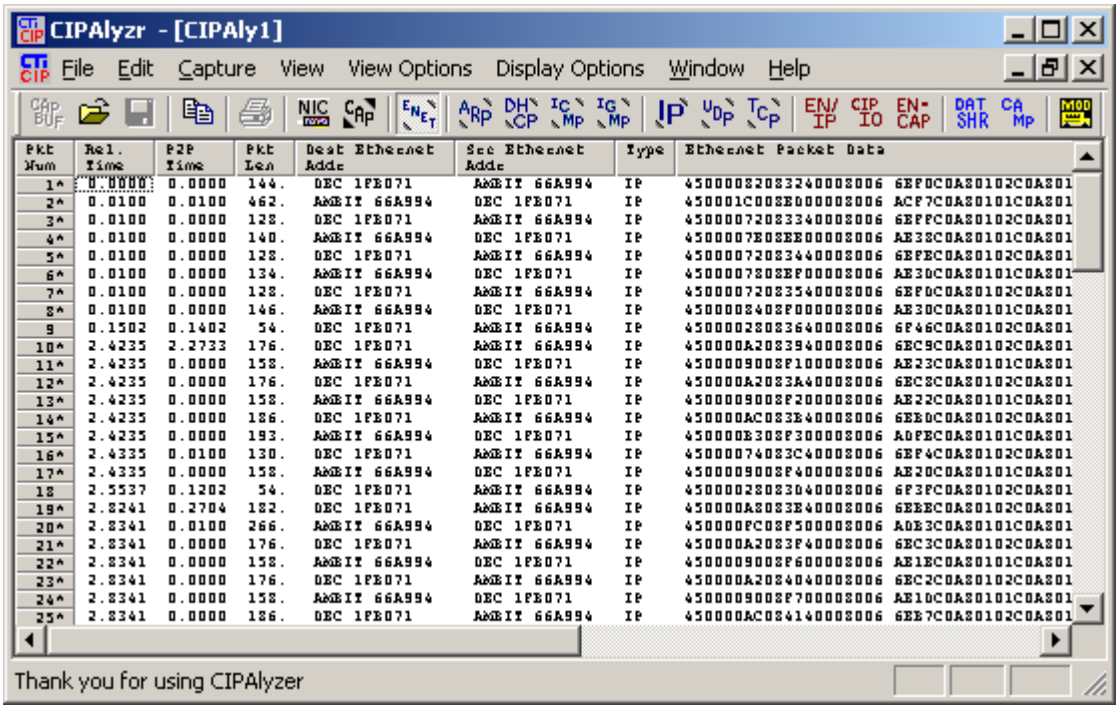
5.2 Program CIPAlyzer 3.0

CTI Industrial Protocol Analyzer (CIPAlyzer) je software k monitorování sítí vyvinutý speciálně pro sítě ethernet, na kterých se používá průmyslových protokolů. Tradiční nástroje pro monitorování se zaměřují na protokoly nižších vrstev, jak bylo zmíněno v předchozím odstavci. Protokoly aplikačních vrstev, jako je například EtherNet/IP, je nutno dekódovat ručně pouze z hexadecimálního pohledu na data přenášená protokolem nižší vrstvy (TCP, UDP). CIPAlyzer je navržen pro dekódování a zobrazení právě těchto protokolů.

CIPAlyzer má jednoduché grafické rozhraní, které umožňuje přehledně zobrazovat zachycené dekódované pakety. V tlačítkové liště je čtrnáct tlačítek, pomocí kterých je možné snadno změnit pohled na zachycená data – od samotných rámců ethernet až po protokoly aplikačních vrstev jako je EtherNet/IP nebo MODBUS/TCP. Pro příklad uvádím základní obrazovku této aplikace na obrázku 5.2. Program dokáže zobrazit (a dekódovat) protokoly služební protokoly ARP, DHCP, ICMP a IGMP, IP protokol v síťové vrstvě. V transportní vrstvě jsou to protokoly TCP a UDP. Z protokolů aplikační vrstvy jsou podporovány EtherNet/IP, CIP I/O, DataShare, CAMP (Common ASCII Messaging Protokol) a MODBUS/TCP.

Jelikož jsem měl k dispozici pouze verzi programu s omezenou funkčností, nebylo možné využít všech možností programu. Největším omezením bylo nemožnost ukládat zachycené pakety, z dalších omezení například nemožnost změnit font zobrazení zachycených paketů, který je příliš malý. Nicméně i přes tato omezení je tento program velice užitečným pomocníkem při monitorování a analyzování komunikace na síti EtherNet/IP.

Tento software byl vyvinut a je prodáván firmou Control Technology Inc., USA. Volně šířitelná verze s omezenou funkčností je distribuována společností ODVA/CI jako část balíčku EtherNet/IP Developer's Toolkit.



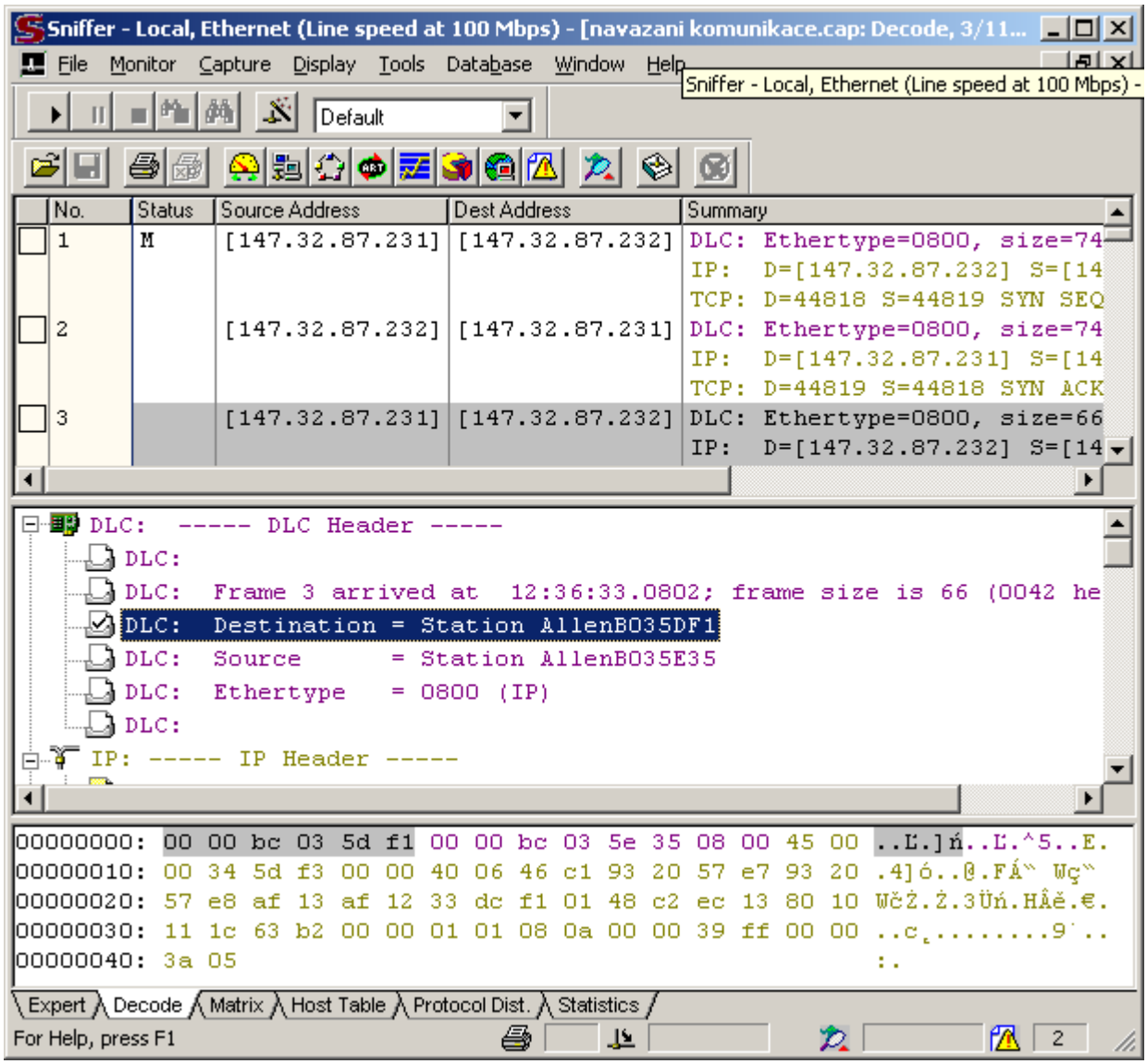
Obrázek 5.2 Okno programu CIPAlyzer

5.3 *Program Sniffer Pro 4.7*

Sniffer Pro je komerční program pro kompletní monitorování sítě dodávaný firmou Network Associates. Tento program je možné použít k monitorování síťových segmentů od ethernetu přes Wireless LAN, ATM až po sítě WAN se synchronním přenosem. Kromě jiného umožňuje zachytávat provoz na síti pro detailní analýzu, měřit dobu odezvy, počítat hopy (skoky) nebo monitorovat síť v reálném čase.

I když je program Sniffer Pro jedním z nejlepších produktů v této oblasti na trhu, není primárně určen pro monitorování průmyslových protokolů – není tedy schopen dekódovat protokol EtherNet/IP. Toto dekódování je třeba provést ručně z hexadecimálního výpisu datové části UDP nebo TCP paketu.

Tento program jsem použil k detailnímu dekódování zachycených paketů při komunikaci mezi automaty ControlLogix. Program CIPAlyzer umožňuje rychlý a názorný pohled na zachycená data; pro detailnější pohled je však vhodnější tento program. Okno programu Sniffer Pro s dekódovanými pakety je zobrazeno na obrázku 5.3. V horní části okna je možno vybrat požadovaný rámec; je zde také zobrazena stručné shrnutí informací o rámci. Ve střední části okna jsou podrobně rozebrána jednotlivá síťová záhlaví jednotlivých protokolů. Dolní část okna zobrazuje hexadecimální pohled na zachycená data.



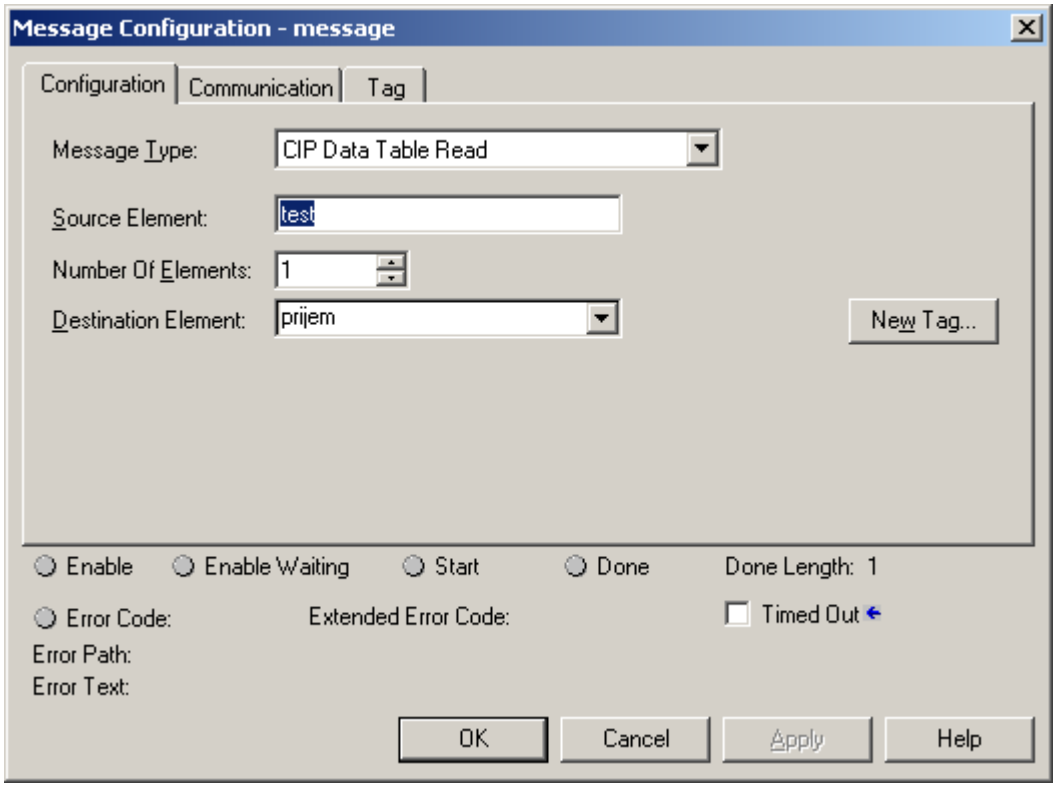
Obrázek 5.3 Okno programu Sniffer Pro

5.4 *Monitorování provozu na síti EtherNet/IP*

Pro ukázku provozu na síti EtherNet/IP jsem zvolil explicitní i implicitní komunikaci, aby bylo zřetelně vidět použití obou protokolů – TCP i UDP. V části s explicitní komunikací se jedná o zaslání zprávy s žádostí o přečtení tagu ve vzdáleném procesoru ControlLogix. V části s implicitní komunikací je ukázka navázání přímého spojení pro komunikaci se vzdálenými analogovými moduly (*direct I/O communication*).

5.4.1 Explicitní komunikace

Explicitní komunikaci demonstruji na použití instrukce MSG. Pomocí této instrukce je procesor ControlLogix 5550 schopen přenášet data mezi procesory. Tato instrukce podporuje komunikaci s procesory stejné řady, ale i s procesory řady PLC 5, PLC 2, FlexLogix a SLC. Pro instrukci MSG mezi procesory řady ControlLogix a FlexLogix musí být uveden v konfiguračním okně instrukce typ *CIP Data Table Read* (CIP čtení) nebo *CIP Data Table Write* (CIP zápis). Zde je nutno také zadat zdrojový a cílový tag.



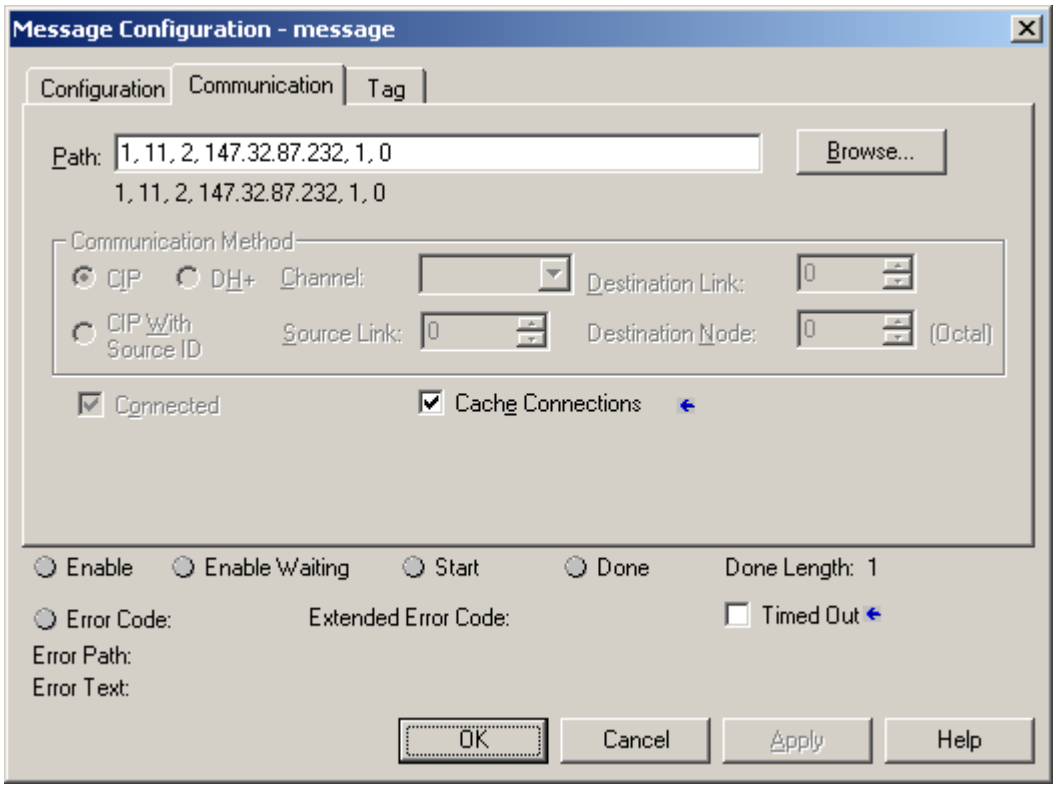
Obrázek 5.4 Nastavení instrukce MSG

V záložce Communication je třeba nastavit příslušnou komunikační cestu. Ta je uvedena buď symbolicky (pokud se vzdálený procesor nachází v I/O konfiguraci prostředí RSLogix 5000) nebo pomocí číslic oddělených čárkami:

Číslice	Význam pole při tvorbě cesty
1	Lokální Sběrnice ControlBus
11	Číslo slotu, kde je umístěn komunikační modul
2	Port komunikačního modulu 1756-ENET
147.32.87.232	Adresa vzdáleného komunikačního modulu v síti ethernet
1	Vzdálené šasi, ve kterém je umístěn komunikační modul
0	Číslo slotu, ve kterém je zasunut vzdálený procesor

Tabulka 5.1 Komunikační cesta v instrukci MSG

Typ komunikační metody musí být v síti EtherNet/IP nastaven na hodnotu CIP. Je možné též povolit možnost *Cache connections*. Povolení této možnosti způsobí, že jednou vytvořené spojení zůstane stále otevřené, i když se nepoužívá. To je výhodné tehdy, používáme-li instrukce MSG často; výhodou je zkrácení doby vykonání instrukce.

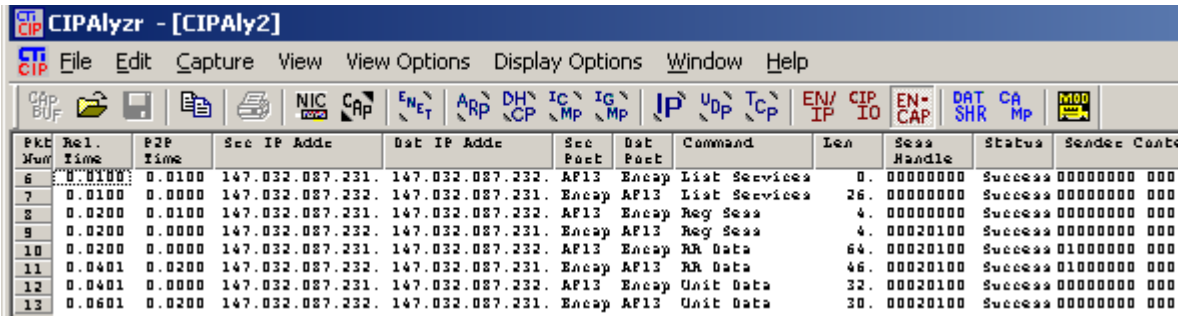


Obrázek 5.5 Komunikační cesta v instrukci MSG

V případě povolení možnosti *Cache Connections* probíhá na síti ethernet tato komunikace:

1) Zaslání požadavku ListServices stanicí, na které je spouštěna instrukce MSG
2) Odpověď na požadavek ListServices
3) Zaslání požadavku RegisterSession
4) Odpověď na požadavek RegisterSession
5) Zaslání zapouzdřeného příkazu Forward_Open pro vytvoření CIP spojení třídy 3 pomocí příkazu SendRRData
6) Odpověď na požadavek Forward_Open
7) Zaslání požadavku CIP Data Table Read pomocí příkazu SendUnitData
8) Odpověď na požadavek CIP Data Table Read

Pokud by nebyla povolena možnost *Cache Connections*, následovalo by uzavření CIP spojení pomocí Forward_Close a uzavření TCP spojení. Komunikace zachycená pomocí programu CIPAlyzer 3.0 je zobrazena na obrázku 5.6.



Pkt Num	Rel. Time	P2P Time	Sec IP Addr	Dst IP Addr	Sec Port	Dst Port	Command	Len	Sess Handle	Status	Sender Cont
6	0.0100	0.0100	147.032.087.231.	147.032.087.232.	AF13	Encap	List Services	0.	00000000	Success	00000000 000
7	0.0100	0.0000	147.032.087.232.	147.032.087.231.	Encap	AF13	List Services	26.	00000000	Success	00000000 000
8	0.0200	0.0100	147.032.087.231.	147.032.087.232.	AF13	Encap	Reg Sess	4.	00000000	Success	00000000 000
9	0.0200	0.0000	147.032.087.232.	147.032.087.231.	Encap	AF13	Reg Sess	4.	00020100	Success	00000000 000
10	0.0200	0.0000	147.032.087.231.	147.032.087.232.	AF13	Encap	RR Data	64.	00020100	Success	01000000 000
11	0.0401	0.0200	147.032.087.232.	147.032.087.231.	Encap	AF13	RR Data	46.	00020100	Success	01000000 000
12	0.0401	0.0000	147.032.087.231.	147.032.087.232.	AF13	Encap	Unit Data	32.	00020100	Success	00000000 000
13	0.0601	0.0200	147.032.087.232.	147.032.087.231.	Encap	AF13	Unit Data	30.	00020100	Success	00000000 000

Obrázek 5.6 Komunikace při zaslání zprávy CIP Data Table Read

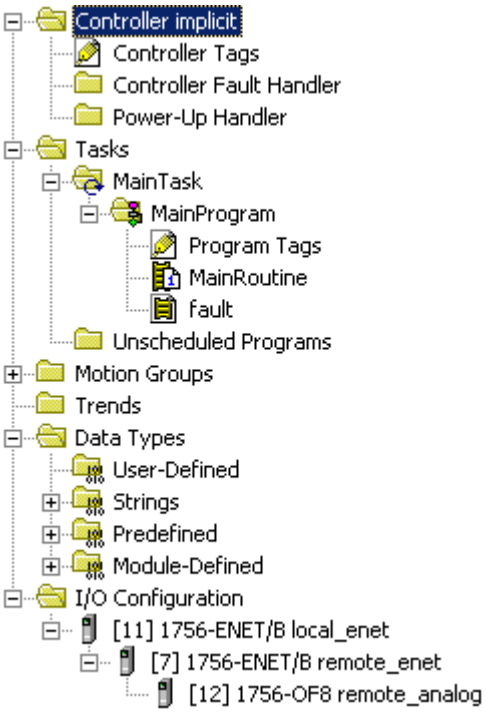
Komunikaci iniciuje modul pro síť ethernet 1756-ENET/B s IP adresou 147.32.87.231 a zdrojovým portem 0xAF13/tcp. Cílovým zařízením je modul pro síť ethernet s IP adresou 147.32.87.232. Cílový port je 0xAF12/tcp, jak bylo již zmíněno v kapitole 3.5.

Detailnější rozbor komunikace je uveden v příloze A.1.

5.4.2 Implicitní komunikace

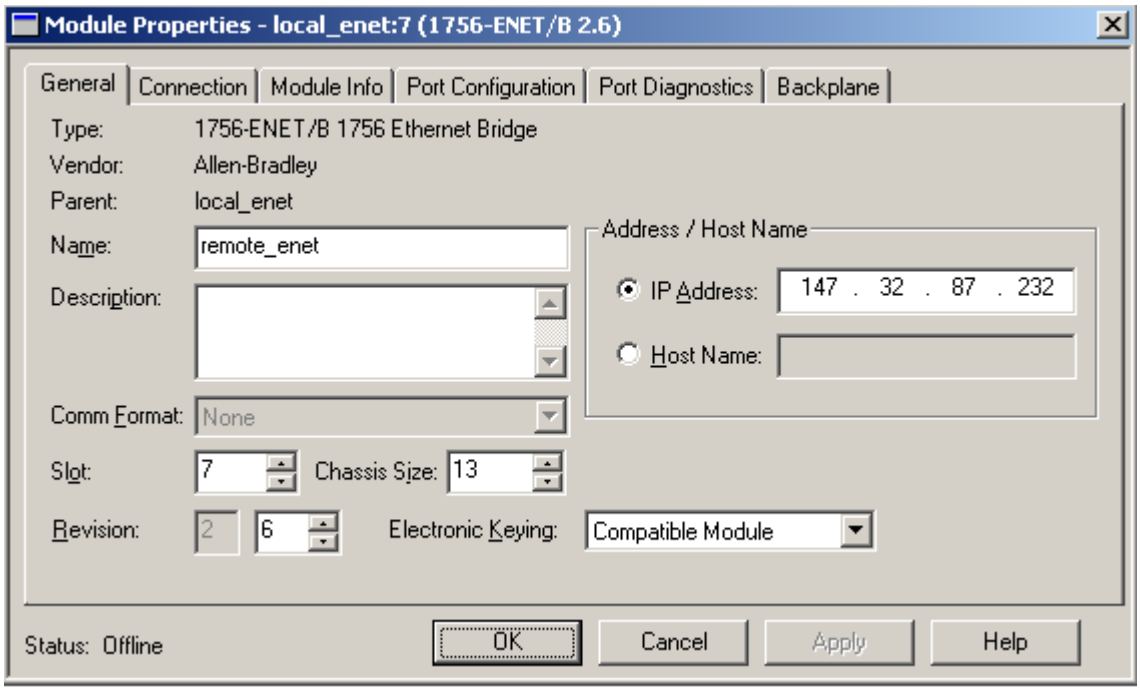
Pro ukázku implicitní komunikace jsem vybral vytvoření I/O spojení a následnou komunikaci se vzdáleným modulem analogových výstupů. Jedná se tedy o vytvoření následující komunikační cesty: procesor ControlLogix 5550 – sběrnice ControlBus – modul pro síť ethernet 1756-ENET/B – modul pro síť ethernet 1756-ENET/B – sběrnice ControlBus – modul analogových výstupů 1756-OF8.

Pro nastavení této komunikace je v prostředí RSLogix 5000 třeba přidat do *I/O configuration* lokální modul 1756-ENET/B, vzdálený modul 1756-ENET/B jako jeho potomka a modul analogových výstupů 1756-OF8 jako potomka vzdáleného modulu pro ethernet.



Obrázek 5.7 I/O konfigurace pro implicitní komunikaci

Při vkládání vzdáleného modulu 1756-ENET/B do I/O konfigurace je třeba zvolit kromě jiného také komunikační formát (*Comm. Format*). Ten má buď hodnotu *Rack Optimization* nebo *None*. V prvním případě se v záložce Communication uvede RPI (Requested Packet Interval) společný pro všechny moduly, které využívají tohoto komunikačního modulu. V druhém případě je možné u každého I/O modulu uvést RPI zvlášť.



Obrázek 5.8 Konfigurace lokálního modulu 1756-ENET/B

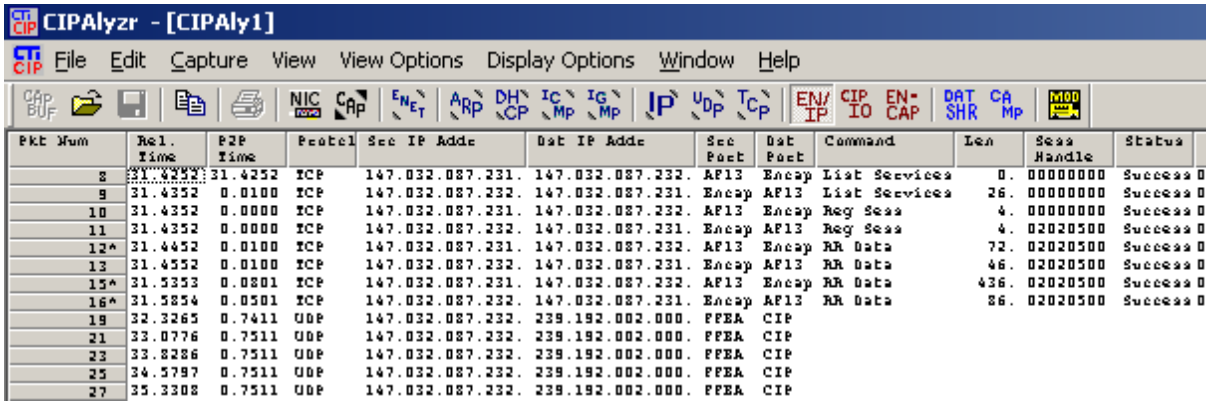
U modulu analogových výstupů byl parametr RPI nastaven na hodnotu 750 ms, což je maximum. Tato hodnota vyjadřuje, jak často jsou přenášena data mezi procesorem a tímto modulem. Největší možnou hodnotu jsem použil proto, aby se nepřenášelo zbytečně mnoho paketů po síti ethernet, což by pro tento demonstrační účel nebylo žádoucí.

Implicitní komunikaci můžeme rozdělit na dvě části: Vytvoření samotného CIP spojení a následná I/O komunikace. Na síti ethernet probíhá tedy tato komunikace:

1) Zaslání požadavku ListServices stanicí, na které je nakonfigurováno I/O.
2) Odpověď na požadavek ListServices
3) Zaslání požadavku RegisterSession
4) Odpověď na požadavek RegisterSession
5) Zaslání zapouzdřeného příkazu Forward_Open pro vytvoření CIP spojení třídy 1 pomocí požadavku SendRRData
6) Odpověď na požadavek Forward_Open

Součástí vytvoření spojení je přihlášení lokálního ethernet modulu do multicastové skupiny, poněvadž komunikace vzdálený modul – lokální procesor ControlLogix je typu multicast.

Následně jsou každých 750 ms (jak bylo definováno pomocí položky RPI) čtena/zapisována data do modulu.



Pkt Num	Rel. Time	P2P Time	Protocl	Sec IP Adde	Dst IP Adde	Sec Port	Dst Port	Command	Len	Sess Handle	Status
8	31.4252	31.4252	TCP	147.032.087.231.	147.032.087.232.	AF13	Encap	List Services	0.	00000000	Success 0
9	31.4352	0.0100	TCP	147.032.087.232.	147.032.087.231.	Encap	AF13	List Services	26.	00000000	Success 0
10	31.4352	0.0000	TCP	147.032.087.231.	147.032.087.232.	AF13	Encap	Reg Sess	4.	00000000	Success 0
11	31.4352	0.0000	TCP	147.032.087.232.	147.032.087.231.	Encap	AF13	Reg Sess	4.	02020500	Success 0
12^	31.4452	0.0100	TCP	147.032.087.231.	147.032.087.232.	AF13	Encap	RR Data	72.	02020500	Success 0
13	31.4552	0.0100	TCP	147.032.087.232.	147.032.087.231.	Encap	AF13	RR Data	46.	02020500	Success 0
15^	31.5353	0.0801	TCP	147.032.087.231.	147.032.087.232.	AF13	Encap	RR Data	436.	02020500	Success 0
16^	31.5854	0.0501	TCP	147.032.087.232.	147.032.087.231.	Encap	AF13	RR Data	86.	02020500	Success 0
19	32.3265	0.7411	UDP	147.032.087.232.	239.192.002.000.	FFEA	CIP				
21	33.0776	0.7511	UDP	147.032.087.232.	239.192.002.000.	FFEA	CIP				
23	33.8286	0.7511	UDP	147.032.087.232.	239.192.002.000.	FFEA	CIP				
25	34.5797	0.7511	UDP	147.032.087.232.	239.192.002.000.	FFEA	CIP				
27	35.3308	0.7511	UDP	147.032.087.232.	239.192.002.000.	FFEA	CIP				

Obrázek 5.9 Vytvoření přímého I/O spojení a následná I/O komunikace

Detailní rozbor vytvoření I/O spojení a následný rozbor I/O komunikace je uveden v příloze.

6 EtherNet/IP a ControlNet

V této kapitole popisuji nejdříve výpočet šířky pásma pro jednotlivé typy implicitní komunikace na síti EtherNet/IP a dále jak je možné vypočítat šířku přenosového pásma sítě EtherNet/IP při navrhování sítě. Dále je zde srovnání sítí EtherNet/IP a ControlNet včetně popisu ověřovacích experimentů.

6.1 Navrhování sítě EtherNet/IP s ohledem na maximální šířku pásma

Modul 1756-ENET/B, který je určen pro komunikaci na sítích ethernet s rychlostí 10Mb/s, má šířku pásma limitovanou na 900 rámců za sekundu. Toto pásmo je využíváno dvěma typy komunikace – implicitní (I/O, rack optimized connections, direct connections, produkované tagy) a explicitní (zasílání zpráv, komunikace se softwarem RS, HTTP). Zmíněnou šířku pásma modulu ENET/B je doporučené [7] dělit takto: Pro explicitní komunikaci rezervovat 10% šířky pásma modulu ENET/B (90 rámců za sekundu), celková implicitní komunikace by neměla zabírat více než 90% šířky pásma modulu (810 rámců za sekundu).

Pokud by nebyla dodržena zmíněná rezerva deseti procent šířky pásma pro explicitní komunikaci, může být omezen přístup na web server modulu, případně by nebylo možné přepnutí do online módu v RSLogix 5000. Explicitní komunikace je však závislá také na dostupnosti sítě a dostupnosti cíle komunikace. Proto vyhrazení zmíněných deseti procent šířky pásma modulu negarantuje průchodnost sítě pro explicitní komunikaci.

Každé CIP spojení na síti EtherNet/IP je duplexní, tzn. že je třeba přenosu nejméně dvou rámců na jeden RPI (*requested packet interval*). Můžeme tedy pro všechna plánovaná spojení vypočítat zabírané šířky pásma (BW – *bandwidth*) takto:

Typ komunikace		BW [rámců za sekundu][5]
Rack Optimized		(2 × počet spojení)/RPI
Direct Connect		(2 × počet spojení)/RPI
Produkovaný tag		
	U producenta	(1 + počet spojení)/RPI pro každý produkovaný tag
	U konzumenta	2/RPI pro každý konzumovaný tag

Tabulka 6.1 Výpočet šířky přenosového pásma pro jednotlivé typy komunikace

Celková šířka pásma pro jednotlivý ENET/B modul je pak součtem jednotlivých šířek pásma pro navržené typy komunikace. Ta by neměla přesáhnou doporučený limit 810 rámců za sekundu.

[5] Jedná se o transportní rámce – pakety protokolu UDP.

Je třeba si uvědomit, že vypočtené šířky pásma pro navržené typy komunikace jsou ***minimální*** a mohou tedy být větší. Proto je třeba brát tyto výpočty pouze jako první vodítko při návrhu sítě.

6.2 Porovnání sítí EtherNet/IP a ControlNet

6.2.1 Úvod

Síť ControlNet je navržena především pro předvídatelný a opakovatelný přenos rozvrhovaných (*scheduled*) dat. Tato síť poskytuje také neplánovanou komunikaci, ale díky mechanismu přenosu neplánovaných zpráv (více viz kapitola 1.2.2) je obtížné odhadnout výkonnost této neplánované komunikace.

ControlNet a EtherNet/IP nejsou ekvivalentní sítě a jsou navrženy pro rozdílné účely. Pokud jsou tyto sítě použity korektně, jsou komplementární, nikoliv však konkurenceschopné. Je proto důležité si být vědom toho, že při použití sítě ControlNet pro přenos především neplánovaných zpráv nemusí být výkonnost optimální nebo dokonce nemusí splnit naše požadavky.

Přenos dat na síti ethernet je koncipován stylem *„kdo dřív přijde, ten dřív komunikuje“*. Proto doručení zprávy na nezatížené síti ethernet může být rychlejší než-li na síti ControlNet. Pro porovnání jedna zpráva na síti ethernet s maximální délkou 750 slov odpovídá zaslání sedmi neplánovaných zpráv na síti ControlNet.

Pokud očekáváme neplánované přenosy s délkou do 100 slov a nízký síťový provoz (neplánovaný), je možné použít ControlNet nebo EtherNet/IP. Pokud očekáváme větší pakety a silný provoz, je vhodnější použít síť ethernet.

6.2.2 Porovnání výkonnosti

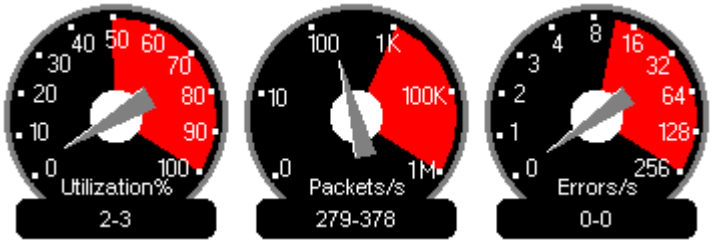
Pro porovnání výkonnosti jsem vybral 2 typy implicitní komunikace. V prvním případě to byla komunikace při připojení vzdálených analogových vstupů nebo výstupů, ve druhém případě komunikace při využití produkovaných/konzumovaných tagů.

Při připojení vzdálených analogových vstupů a výstupů (jednalo se modul analogových výstupů 1756-IF6I a modul analogových vstupů 1756-OF8, viz obrázek 6.2) byla u každého modulu nastavena zvlášť nejmenší možná hodnota RPI. Ta v případě výstupního modulu byla 25ms a v případě vstupního modulu 10ms. Jednalo se tedy o dvě přímá I/O spojení s celkovou šířkou pásma 280 rámců za sekundu (výpočet dle tabulky 6.1):

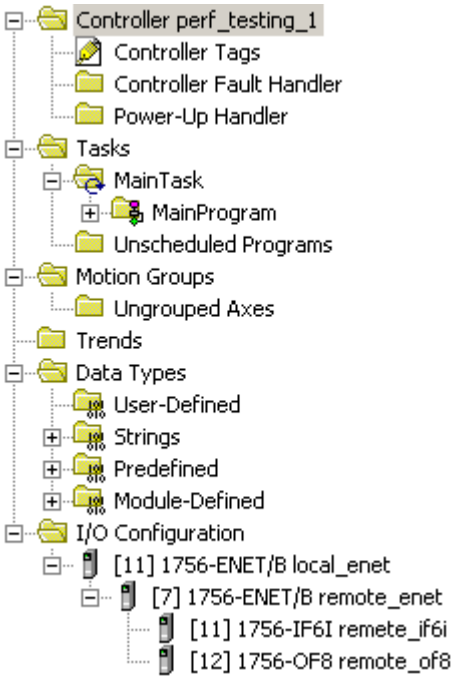
$$BW = \frac{2 \cdot 1}{0.010\,\text{sec.}} + \frac{2 \cdot 1}{0.025\,\text{sec.}} = 280 \textit{rámců} / \text{sec.}$$

Rovnice 6.1 Výpočet šířky pásma při připojení vzdálených analogových vstupů a výstupů

Hodnota 280 rámců/sec. je menší nežli maximum 810 rámců/sec., takže je zaručena funkčnost a spolehlivost komunikace. Pro ověření této hodnoty jsem použil software Sniffer Pro 4.7. Naměřená hodnota zatížení sítě skutečně kolísala (±2 rámce/sec.) okolo hodnoty 280 rámců/sec, viz obrázek 6.1. Pozorované kolísání je způsobené nepřesností měření (chyba způsobená zaokrouhlením).



Obrázek 6.1 Měření zatížení programem Sniffer Pro



Obrázek 6.2 Testovaná I/O konfigurace

Takovéto zatížení bylo naměřeno v situaci, kdy moduly komunikovaly, avšak žádná data nebyla do výstupního modulu zapisována. Pro otestování i této situace jsem zapisoval do všech osmi kanálů modulu analogových výstupů 1756-OF8 hodnotu 0 ÷ 8000 danou čítačem. V tomto případě bylo zatížení sítě přibližně 340 rámců/sec.

Pro přehlednost uvádím, že se jednalo o tuto komunikaci:

Analogový výstup: 147.32.87.231 → 147.32.87.232
Port 65516/udp → 65497/udp

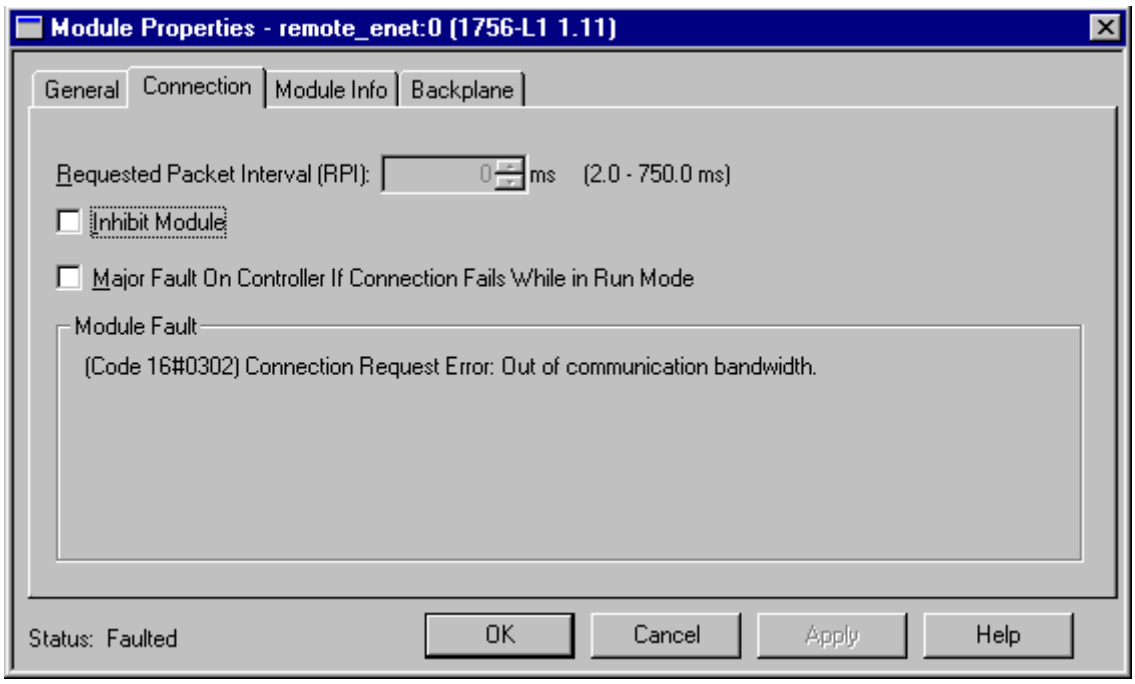
Analogový vstup: 147.32.87.232 → 239.192.2.1
Port 65496/udp → 2222/udp

Je tedy vidět, že vypočtená hodnota šířky pásma je minimální možná, při zápisu dat do vzdáleného I/O modulu je potřebná šířka pásma vyšší přibližně o 22%.

Jelikož připojení dvou vzdálených analogových I/O modulů využívá méně než polovinu doporučené maximální šířky pásma pro implicitní komunikaci (a více těchto modulů nebylo k dispozici), rozhodl jsem se otestovat zatížitelnost sítě EtherNet/IP pomocí komunikace při produkci / konzumování tagů.

Pro každý produkovaný tag je dle tabulky 6.1 třeba šířka pásma 200 rámců/sec. při RPI=10ms. Modul 1756-ENET/B lze tedy použít při přenosu maximálně 4 tagů při zmíněné hodnotě RPI. Na první pohled je to velice málo, ale stejnou šířku pásma zabere i přenos celého pole tagů až do velikosti 125 prvků. Je tedy velice výhodné seskupovat proměnné, u kterých předpokládáme přenos, do polí.

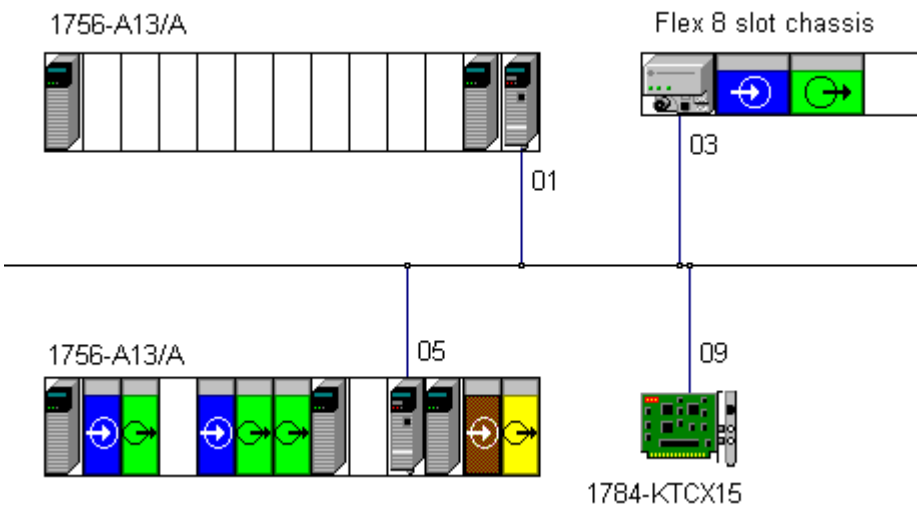
Pokud překročíme dovolenou hranici 900 rámců/sec., modul je přetížen a zobrazí se následující chybové hlášení:



Obrázek 6.3 Chybové hlášení při překročení maximální šířky pásma modulu

Pomocí provedeného měření zatížení sítě jsem ověřil platnost vztahu pro výpočet požadované šířky pásma pro komunikaci v případě produkovaných / konzumovaných tagů. Naproti tomu je třeba brát vztah v případě přímých I/O spojení s rezervou, udává totiž pouze minimální možnou hodnotu.

Pro srovnání jsem nastavil stejnou komunikaci jako v předchozích dvou případech i na síti ControlNet. Konfigurace sítě ControlNet je zobrazena na obrázku 6.4.



Obrázek 6.4 Konfigurace sítě ControlNet

NUT (*Network Update Time*) sítě ControlNet byl nastaven na 2ms, u vzdálených I/O modulů a konzumovaných tagů byla hodnota RPI stejná jako při použití sítě EtherNet/IP.

V případě vzdálených analogových I/O modulů bylo využito průměrně 2,93% pásma pro plánovanou (*scheduled*) komunikaci.

V případě produkovaných/konzumovaných tagů bylo pro přenos 4 tagů využito průměrně 78,60% pásma pro plánovanou komunikaci.

Měření průměrné hodnoty využití pásma pro plánovanou komunikaci na síti ControlNet bylo provede pomocí programu RSNetWorx for ControlNet. Tento program se dále používá především pro nastavení přenosů na síti ControlNet

Pro přehlednost uvádím výsledky testů v následující tabulce:

Typ komunikace	EtherNet/IP 10 Mb/s			ControlNet 5 Mb/s	Poznámka
	Vypočteno	Naměřeno		% BW pro plánované přenosy	
		Rámce/sec	% BW		
Direct I/O, 2 spojení	280	280	35%	2,93%	Bez zápisu
Direct I/O, 2 spojení	280	340	42%	2,93%	Se zápisem
Konzumované tagy Počet: 3 (RPI=10ms)	600	600	74%	53,21%	NUT = 2ms
Konzumované tagy Počet: 4 (RPI=10ms)	800	785	97%	78,60%	NUT = 2ms

Tabulka 6.2 Výsledky testování sítí EtherNet/IP a ControlNet

6.2.3 Závěr

Testování zatížitelnosti sítí EtherNet/IP s přenosovou rychlostí 10Mb/s a ControlNet ukázalo, že výpočet šířky potřebného pásma sítě ethernet je v případě produkovaných/konzumovaných tagů směrodatný a v praxi využitelný (zejména při návrhu sítě). Naproti tomu je tento výpočet pouze orientační v případě přímých I/O spojení.

Síť ControlNet je oproti EtherNet/IP výkonnější zejména při přenosu malých objemů dat, což jsou přímá I/O spojení (a spojení optimalizovaná pro rack). Zde byl konkrétně rozdíl v potřebě přibližně 10ti násobně větší šířky pásma sítě ethernet oproti síti ControlNet. Rozdíly ve výkonnosti sítí se zmenšují při přenosu velkým objemů dat, jako tomu je v případě produkovaných/konzumovaných tagů. Zde je náskok sítě ControlNet pouze malý, v testovaných případech přibližně 23% a 40%. Při tomto měření byl čas obnovy sítě NUT=2ms. To je nejnižší možná hodnota a následkem toho dojde k zaplnění pásma pro rozvrhovanou komunikaci poměrně rychle. Takto nízkou hodnotu NUT však bylo nutné nastavit proto, aby byl přenos čtyřech produkovaných tagů s RPI=10ms vůbec možný. Při vyšších hodnotách NUT má pásmo pro rozvrhovanou komunikaci vyšší kapacitu a výsledky jsou příznivější pro síť ControlNet. Například při přenosu 4 produkovaných tagů s RPI=20ms je na síti EtherNet/IP využito 49% využitelného pásma (BW=4·(2/0.020)=400 rámců za sekundu, což odpovídá 49%), ale při stejné komunikaci na síti ControlNet je využito pouze 17% pásma pro rozvrhovanou komunikaci. Toto srovnání však nepočítá s použitím ethernet sítě při rychlosti 100Mb/s, potom by výsledky byly příznivější opět pro EtherNet/IP.

Pro plánování a sledování provozu sítě EtherNet/IP byl firmou Rockwell Software vyvinut program RSNetWorx for EtherNet/IP; jeho otestování však nebylo možné, jelikož tento software nebyl doposud k dispozici.

7 Závěr a zhodnocení

Tato diplomová práce se zabývá popisem implementace protokolu CIP na standardní síť ethernet včetně popisu použití protokolů TCP a UDP. Vzhledem k tomu, že síť EtherNet/IP využívá stejnou infrastrukturu jako tradiční ethernet používaný v kancelářském prostředí, nabízí se možnost propojení zóny sítě využívané pro řídicí účely a zóny využívané pro ostatní informační komunikaci. Z bezpečnostních důvodů je však třeba určité oddělení těchto zón. Oddělujícím prvkem může být určitý typ zařízení zvaného *firewall*. Pro jeho správnou konfiguraci je třeba znát datové toky, které budou tímto zařízením procházet. To zahrnuje znalost používaných čísel TCP a UDP portů včetně iniciátorů komunikace. V kapitole třetí jsou informace, které lze využít i pro tyto účely.

Provedená analýza některých typů komunikace na síti EtherNet/IP v kapitole páté je přínosná zejména z výukového hlediska. Zachycenou komunikaci, která je podrobně rozepsaná v přílohách A.1 a A.2 , je možné využít při výuce komunikačních sítí používaných v distribuovaných řídicích systémech. Popis aplikace CIPAlyzer 3.0 je též využitelný při výuce předmětů se zaměřením na sítě s komunikačním protokolem CIP.

Porovnání sítí EtherNet/IP a ControlNet provedené v kapitole šesté se zabývá dvěma typy implicitních přenosů: Přímá I/O spojení a komunikace při přenosu produkovaných / konzumovaných tagů. Výsledky testů ukázaly, že budou-li na plánované komunikační síti pro distribuovaný řídicí systém převládat přímá I/O spojení, je výhodnější použít síť ControlNet. Naopak při předpokladu přenosů větších objemů dat, jako například zmíněné produkované / konzumované tagy, je výhodnější použít síť EtherNet/IP. Toto srovnání nezohledňuje další rysy sítě ControlNet, kterými je naprostý determinismus a možnost redundantní kabeláže. Ani jeden z těchto rysů není na síti EtherNet/IP možný. Takovéto porovnání sítí ControlNet a EtherNet/IP nebylo dosud publikováno.

Při provedeném porovnání sítí EtherNet/IP a ControlNet bylo použito standardní sítě ethernet s přenosovou rychlostí 10 Mb/s. Aby takovéto srovnání bylo úplné, je třeba otestovat propustnost též sítě ethernet s rychlostí 100 Mb/s (*fast ethernet*). Avšak vzhledem k tomu, že *fast ethernet* moduly pro automaty ControlLogix nebyly k dispozici, nebylo toto testování možné.

Seznam zkratek a definic

ARP Address Resolution Protocol. Internetový protokol, který dynamicky mapuje internetové síťové IP adresy do fyzických (hardwarových) MAC adres. Protokol ARP je definován v RFC 826 a je omezen na sítě, které podporují všesměrové vysílání.

ATM Asynchronous transfer mode. Vysokorychlostní přenosová technologie, používající přepínání buněk pevné délky (53 bajtů). Pevná délka buněk umožňuje rychlé zpracování v hardware přepínačů při jejich přenosu ATM sítí.

Big endian Název pořadí, ve kterém jsou v paměti uložena vícebytové hodnoty: Big endian znamená, že významově vyšší byte je uložen v paměti na nižší adrese a významově nižší byte je uložen v paměti na vyšší adrese.

Broadcast Všesměrové vysílání - služba doručování paketů, kdy všechny uzly sítě obdrží kopii rámce, určeného pro toto všesměrové vysílání.

Datagram Samostatně nesená, nezávislá entita dat přenášející dostatek informací k tomu, aby mohla být směrována od zdrojového k cílovému počítači bez spoléhání na dřívější komunikaci zdroje a cíle a bez spoléhání na přenášející síť.

DHCP Dynamic Host Configuration Protocol. Mechanismus pro dynamické přidělování IP adres jednotlivým uzlům sítě.

DINT 32 bitové celé číslo. DINT znamená *double integer.*
UDINT 32 bitové celé číslo bez znaménka.

Ethernet 10-ti Mb/s standard pro LAN, původně vyvinutý firmou Xerox. Později byl přepracován firmami Digital, Intel a Xerox. Všechny stanice jsou připojeny na koaxiální kabel. Pro přístup na sběrnici se použita metoda CSMA/CD (přístup na sběrnici s příposlechem nosné s detekcí kolizí).

EtherNet/IP Produkty odpovídající specifikaci CIP Common Specification a EtherNet/IP Adaptation of CIP. EtherNet/IP znamená ethernet Industrial Protocol.

Fast Ethernet
Ethernet II 100 Mb/s standard pro LAN, který používá stejnou strukturu paketů, adresové schéma a CSMA/CD jako 10-ti Mb/s ethernet. Fast ethernet je definován IEEE 802.3u.

FDDI Fibber Distributed Data Interface. Síťová technologie vyvinutá ANSI (American National Standards Institute), výborem X3T9.5, s přenosovou rychlostí 100 Mb/s po optických vláknech. Používá přístupovou metodu token-passing a architekturu dvojitého kruhu pro zvýšení spolehlivosti s maximální vzdáleností 100 km.

FTP Protokol pro přenos souborů (File Transfer Protocol). FTP je také internetová aplikace, která používá spolehlivý přenos paketů protokolem TCP k přenášení souborů mezi různými uzly.

HTTP Hypertext Transfer Protocol. Protokol používaný WWW servery a WWW prohlížeči k přenosu souborů na Internetu.

Hub Rozbočovač (koncentrátor, *repeater*) propojující uzly ve hvězdicové síťové topologii bez ohledu na sledování a řízení toku dat v počítačové síti.

I/O Vstupy/výstupy

IEEE Institute of Electrical and Electronics Engineers. Největší profesní a standardizační organizace na světě, založená roku 1884, jejíž aktivity mimo pořádání konferencí a vydávání odborných časopisů zahrnují přípravu a vydávání komunikačních a síťových standardů. Pro počítačové sítě má největší význam standardizační orgán založený v rámci IEEE v únoru roku 1980 (a proto označovaný jako IEEE 802), který je specificky zaměřen na problematiku standardu lokálních sítí. Pro jednotlivé oblasti jsou pak vytvořeny pracovní skupiny.

INT 16-ti bitové celé číslo.
UINT UINT je 16-ti bitové celé číslo bez znaménka.

Interlock zprávy V kontextu automatizační techniky se jedná o zprávy zasílané mezi jednotlivými procesory distribuovaného řídicího systému.

IP multicast Směrovací technika umožňující vysílání IP paketů z jednoho zdroje (více zdrojů) více cílům. Paket není posílán separátně každému cíli, ale jeden paket je zaslán do tzv. multicast group, identifikované speciální IP adresou (IP destination group address).

ISO International Organisation for Standardization. Mezinárodní nevládní organizace založená v roce 1947, sdružuje národní standardizační instituce (např. ANSI, DIN atd.). Pokrývá širokou škálu technických problémů mj. globální standardy pro komunikace a výměnu informací. Nejznámější je koncepce tzv. referenčního modelu ISO-OSI (ISO Open Systems Interconnection). Standardy ISO nejsou volně dostupné (např. proti dokumentům RFC).

ISO-OSI model Open Systems Interconnection model. Sedmivrstvový model pro datové komunikace, vyvinutý organizací ISO a ITU-T. Model sestává ze sedmi vrstev, každá vrstva má specifickou funkci jako např. adresaci, řízení toku, kontrolu chyb a zajištění spolehlivosti přenosu. Nejnižší vrstva (fyzická) je svázána s přenosovým médiem a spolu s druhou vrstvou (spojová) je implementována jak v hardware tak v software, zatímco horních pět vrstev je implementováno pouze v software.

LAN Local area network. Skupina počítačů a dalších zařízení propojená komunikačním spojem na relativně malé geografické oblasti (do několika 1000 m), umožňující zařízením vzájemnou komunikaci. Standardy pro LAN specifikují kabeláž a signalizaci na fyzické a spojové vrstvě OSI modelu.

Little endian Název pořadí, ve kterém jsou v paměti uložena vícebytové hodnoty: Little endian znamená, že významově nižší byte je uložen v paměti na nižší adrese a významově vyšší byte je uložen v paměti na vyšší adrese.

Paket Jednotka přenesených dat

Point-to-point Spojení, které existuje jen mezi dvěma uzly (nody). Jiným typem spojení je například multicast.

Port Ve kontextu protokolů TCP a UDP je port hodnota, která demultiplexuje transportní vrstvu – každá aplikace má jedinečné číslo portu, pod kterým komunikuje na transportní vrstvě (protokoly TCP, UDP).

PPP Point-to-point protocol. Protokolový standard pro přenosy na synchronních a asynchronních linkách bod-bod (směrovač-směrovač, host-síť). Nahradil protokol SLIP možností práce i s dalšími protokoly síťové vrstvy (jiné než IP) a má zabudovány bezpečnostní mechanismy např. CHAP a PAP.

Rámec Jednotka dat přenesená po lince.

RARP Reverse Address Resolution Protocol. Jeden ze sady protokolů TCP/IP, mapující fyzickou MAC adresu do IP adresy, používaný např. stanicí pro zjištění své IP adresy při bootování.

RFC Request for comments – dokumenty spravované nezávislou organizací IETF (Internet Engineering Task Force). Tato organizace vystupuje jako standardizační organizace pro internetové protokoly.

RJ-45 Modulární 8-pinový konektor používaný v počítačových sítích pro čtyři kroucené 2-páry (UTP).

SINT 8-mi bitové celé číslo. SINT znamená *short integer.*
USINT USINT je 8-mi bitové celé číslo bez znaménka.

SLIP Serial Line Internet Protocol – Internetový protokol, používaný pro přenos protokolu IP po sériových linkách typu bod-bod. V dnešní době již většinou nahrazen protokolem PPP (Point-to-Point protocol).

SMTP Internetový standard pro přenos zpráv elektronické pošty (Simple Mail Transfer Protokol).

SNMP Simple Network Management Protokol - síťový protokol používaný pro správu počítačových sítí, převážně založených na protokolu TCP/IP. Umožňuje monitorování a řízení síťových zařízení (konfigurace, sběr statistických dat, monitorování výkonnosti atd.).

STP Stíněná kroucená dvoulinka, přenosové médium používané v sítích 10Base-T a 100Base-T.

UTP Nestíněná kroucená dvoulinka, přenosové médium používané v sítích 10Base-T a 100Base-T.

Zapouzdření Technika použitá vrstvenými protokoly, ve kterých každá vrstva přidá hlavičku (header) do datové části paketu z předchozí vrstvy. V internetové terminologii například paket obsahuje hlavičku z linkové vrstvy, následovanou hlavičkou ze síťové vrstvy (IP), následovanou hlavičkou transportní vrstvy (TCP), která je následovaná daty aplikačního protokolu.

Seznam použité literatury

[1] ODVA/CI, *EtherNet/IP Specification, volume 1: CIP Common Specification, Release 1.0*, ControlNet International and Open DeviceNet Vendor Association, June 5, 2001

[2] ODVA/CI, *EtherNet/IP Specification, volume 2: EtherNet/IP Adaptation of CIP, Release 1.0,* ControlNet International and Open DeviceNet Vendor Association, June 5, 2001

[3] Kment, M., *Průmyslové komunikace Allen-Bradley*, diplomová práce, České Vysoké Učení Technické, katedra řídicí techniky, Praha, 2001

[4] Dostálek L., Kabelová A., *Velký průvodce protokoly TCP/IP a systémem DNS*, 2. aktualizované vydání, Computer Press Praha, 2000

[5] Rockwell Automation, *Allen-Bradley Home Page*, [online], dostupné na http://www.ab.com, 2002

[6] Mojžíšek, P., *Multicasting na internetu*, [online], dostupné na http://www.isdn.cz, 2002

[7] Rockwell Automation, *EtherNet/IP Performance and Application Guide*, ENET-AP001B-EN-P, [online], dostupné na http://www.ab.com, 2001

[8] Lounsbury B., Westerman J., *Surviving the Manufacturing Industrial Environment White Paper*, [online], dostupné na http://www.ab.com, 2001

[9] Brooks P., *EtherNet/IP: Industrial Protocol White Paper*, [online], dostupné na http://www.ab.com, 2001

[10] Rockwell Automation, *ControlLogix Ethernet Bridge Module*, 1756-UM050A-EN-P, [online], dostupné na http://www.ab.com, 2001

[11] Rockwell Automation, *ControlLogix Communication Interface Module*, 1756-UM051B-EN-P, [online], dostupné na http://www.ab.com, 2001

[12] Rockwell Automation, *ControlLogix System: User Manual*, 1756-UM001E-EN-P, [online], dostupné na http://www.ab.com, 2002

[13] Rockwell Automation, *ControlLogix ControlNet Interface: Module User Manual*, [online], dostupné na http://www.ab.com, 1999

[14] Rockwell Software, *RSNetWorx For ControlNet: Getting Results Guide*, 9939-CNETGR-DEC00, [online], dostupné na http://www.ab.com, 2000

[15] Rockwell Software, *RSNetWorx For EtherNet/IP: Getting Results Guide*, ENET-GR001A-EN-P, [online], dostupné na http://www.ab.com, 2002

[16] Marshall, Perry S., *Industrial EtherNet: A pocket guide*, ISA – The Instrumentation, Systems, and Automation Society, U.S.A, 2002

[17] Filler, R., Leinonen G., *Programmable controllers using Allen-Bradley SLC 500 and ControlLogix*, Prentice Hall, 2002

[18] ControlNet International, *ControlNet International home page*, [online], dostupné na http://www.controlnet.org, 2002

[19] ODVA, *DeviceNet/ODVA Official Website*, [online], dostupné na http://www.odva.org, 2002

Seznam použitého software

[20] Rockwell Software, *RSLogix 5000 Enterprise Series*, V11.11.00
[21] Rockwell Software, *RSLinx*, v. 2.40.01
[22] Rockwell Software, *RSNetWorx for ControlNet*, v. 3.30
[23] Network Associates, *Sniffer Pro*, v. 4.70.04
[24] Control Technology Inc., *CIPAlyzer (EtherNet/IP Toolkit Version)*, r. 3.00

Příloha A

A.1 Detailní rozbor explicitní komunikace

V dalším textu budou použity pro zvýšení přehlednosti různé barvy textu. **Modře** je označeno záhlaví rámce ethernet, **zeleně** je označeno záhlaví IP protokolu, **černě** je označeno záhlaví protokolu transportní vrstvy (nebo protokolu IGMP). **Červeně** je označena datová oblast paketu TCP nebo UDP.

V hexadecimálním pohledu na data jsou významné části označeny tučně a příslušnou barvou, pod tímto pohledem je opět s použitím těchto barev napsaná legenda.

Vytvoření explicitního spojení pro zasílání zpráv pomocí instrukce MSG se skládá z následujících kroků:

Paket	Použitý protokol	Význam
1	TCP	Požadavek ListServices
2	TCP	Odpověď ListServices
3	TCP	Požadavek RegisterSession
4	TCP	Odpověď RegisterSession
5	TCP	Požadavek Forward_Open
6	TCP	Odpověď Forward_Open

Následná komunikace se skládá z těchto kroků:

Paket	Použitý protokol	Význam
7	TCP	Požadavek Data Table Read
8	TCP	Odpověď Data Table Read

Paket 1: Požadavek ListServices

```
Source          Destination  Bytes Rel Time
Delta Time Abs time           Summary
-----------------------------------------------------------------------------
[147.32.87.231] [147.32.87.232] 90 0:00:00.006
0.001.801 02.12.2002 14:46:03 TCP: D=44818 S=44819

TCP: ----- TCP header -----
TCP:
TCP: Source port = 44819
TCP: Destination port = 44818
TCP: Sequence number = 3623895464
TCP: Next expected Seq number= 3623895488
TCP: Acknowledgment number = 3623895464
TCP: Data offset = 32 bytes
TCP: Reserved Bits: Reserved for Future Use (Not shown in the Hex Dump)
TCP: Flags = 18
TCP: ..0. .... = (No urgent pointer)
TCP: ...1 .... = Acknowledgment
TCP: .... 1... = Push
TCP: .... .0.. = (No reset)
TCP: .... ..0. = (No SYN)
TCP: .... ...0 = (No FIN)
TCP: Window = 4380
TCP: Checksum = F848 (correct)
TCP: Urgent pointer = 0
TCP:
TCP: Options follow
TCP: No-Operation
TCP: No-Operation
TCP: Timestamp Option:
TCP: Timestamp value = 216
TCP: Timestamp echo reply = 216
TCP: [24 Bytes of data]
TCP:

ADDR  HEX                                                       ASCII
0000: 00 00 bc 03 5d f1 00 00 bc 03 5e 35 08 00 45 00 | ..Ľ.]ń..Ľ.^5..E.
0010: 00 4c 00 02 00 00 40 06 a4 9a 93 20 57 e7 93 20 | .L....@.¤š" Wç" 
0020: 57 e8 af 13 af 12 d8 00 41 a8 d8 00 41 a8 80 18 | WčŻ.Ż.Ř.A¨Ř.A¨€.
0030: 11 1c f8 48 00 00 01 01 08 0a 00 00 00 d8 00 00 | ..řH.........Ř..
0040: 00 d8 04 00 00 00 00 00 00 00 00 00 00 00 00 00 | .Ř..............
0050: 00 00 00 00 00 00 00 00 00 00                   | ..........
```

Zdrojová IP adresa: **93 20 57 e7**

Cílová IP adresa: **93 20 57 e8**

Zdrojový port: **af 13**

Cílový port: **af 12**

Požadavek ListServices: **04 00**

Paket 2: Odpověď na požadavek ListServices

```
Source          Destination   Bytes Rel Time
Delta Time Abs time           Summary
-----------------------------------------------------------------------------
[147.32.87.232] [147.32.87.231] 116 0:00:00.011
0.005.387 02.12.2002 14:46:03 TCP: D=44819 S=44818

TCP: ----- TCP header -----
TCP:
TCP: Source port = 44818
TCP: Destination port = 44819
TCP: Sequence number = 3623895464
TCP: Next expected Seq number= 3623895514
TCP: Acknowledgment number = 3623895488
TCP: Data offset = 32 bytes
TCP: Reserved Bits: Reserved for Future Use (Not shown in the Hex Dump)
TCP: Flags = 18
TCP: ..0. .... = (No urgent pointer)
TCP: ...1 .... = Acknowledgment
TCP: .... 1... = Push
TCP: .... .0.. = (No reset)
TCP: .... ..0. = (No SYN)
TCP: .... ...0 = (No FIN)
TCP: Window = 16360
TCP: Checksum = B042 (correct)
TCP: Urgent pointer = 0
TCP:
TCP: Options follow
TCP: No-Operation
TCP: No-Operation
TCP: Timestamp Option:
TCP: Timestamp value = 216
TCP: Timestamp echo reply = 216
TCP: [50 Bytes of data]
TCP:

ADDR  HEX                                                       ASCII
0000: 00 00 bc 03 5e 35 00 00 bc 03 5d f1 08 00 45 00 | ..Ľ.^5..Ľ.]ń..E.
0010: 00 66 00 01 00 00 40 06 a4 81 93 20 57 e8 93 20 | .f....@.¤_" Wč"
0020: 57 e7 af 12 af 13 d8 00 41 a8 d8 00 41 c0 80 18 | WçŻ.Ż.Ř.A¨Ř.AŕŁ.
0030: 3f e8 b0 42 00 00 01 01 08 0a 00 00 00 d8 00 00 | ?č°B.........Ř..
0040: 00 d8 04 00 1a 00 00 00 00 00 00 00 00 00 00 00 | .Ř..............
0050: 00 00 00 00 00 00 00 00 00 00 01 00 00 01 14 00 | ................
0060: 01 00 20 01 43 6f 6d 6d 75 6e 69 63 61 74 69 6f | .. .Communicatio
0070: 6e 73 00 00                                     | ns..
```

Zdrojová IP adresa: `93 20 57 e8`

Cílová IP adresa: `93 20 57 e7`

Zdrojový port: **`af 12`**

Cílový port: **`af 13`**

Odpověď na požadavek ListServices: `04 00`

Jméno služby: **`Communications`**

Paket 3: Požadavek RegisterSession

```
Source           Destination    Bytes  Rel Time
Delta Time Abs time             Summary
-----------------------------------------------------------------------------
8 [147.32.87.231] [147.32.87.232] 94 0:00:00.015
0.003.487 02.12.2002 14:46:03 TCP: D=44818 S=44819

TCP: ----- TCP header -----
TCP:
TCP: Source port = 44819
TCP: Destination port = 44818
TCP: Sequence number = 3623895488
TCP: Next expected Seq number= 3623895516
TCP: Acknowledgment number = 3623895514
TCP: Data offset = 32 bytes
TCP: Reserved Bits: Reserved for Future Use (Not shown in the Hex Dump)
TCP: Flags = 18
TCP: ..0. .... = (No urgent pointer)
TCP: ...1 .... = Acknowledgment
TCP: .... 1... = Push
TCP: .... .0.. = (No reset)
TCP: .... ..0. = (No SYN)
TCP: .... ...0 = (No FIN)
TCP: Window = 4380
TCP: Checksum = 91FA (correct)
TCP: Urgent pointer = 0
TCP:
TCP: Options follow
TCP: No-Operation
TCP: No-Operation
TCP: Timestamp Option:
TCP: Timestamp value = 216
TCP: Timestamp echo reply = 216
TCP: [28 Bytes of data]
TCP:

ADDR  HEX                                                       ASCII
0000: 00 00 bc 03 5d f1 00 00 bc 03 5e 35 08 00 45 00 | ..Ľ.]ń..Ľ.^5..E.
0010: 00 50 00 03 00 00 40 06 a4 95 93 20 57 e7 93 20 | .P....@.¤•" Wç" 
0020: 57 e8 af 13 af 12 d8 00 41 c0 d8 00 41 da 80 18 | WčŻ.Ż.Ř.AŕŘ.AÚ€.
0030: 11 1c 91 fa 00 00 01 01 08 0a 00 00 00 d8 00 00 | ..'ú.........Ř..
0040: 00 d8 65 00 04 00 00 00 00 00 00 00 00 00 00 00 | .Ře.............
0050: 00 00 00 00 00 00 00 00 00 00 01 00 00 00       | ..............
```

Zdrojová IP adresa: **93 20 57 e7**

Cílová IP adresa: **93 20 57 e8**

Zdrojový port: **af 13**

Cílový port: **af 12**

Požadavek RegisterSession: **04 00**

Paket 4: Odpověď na požadavek RegisterSession

```
Source          Destination  Bytes Rel Time
Delta Time Abs time           Summary
-------------------------------------------------------------------------
[147.32.87.232] [147.32.87.231] 94 0:00:00.018
0.003.312 02.12.2002 14:46:03 TCP: D=44819 S=44818

TCP: ----- TCP header -----
TCP:
TCP: Source port = 44818
TCP: Destination port = 44819
TCP: Sequence number = 3623895514
TCP: Next expected Seq number= 3623895542
TCP: Acknowledgment number = 3623895516
TCP: Data offset = 32 bytes
TCP: Reserved Bits: Reserved for Future Use (Not shown in the Hex Dump)
TCP: Flags = 18
TCP: ..0. .... = (No urgent pointer)
TCP: ...1 .... = Acknowledgment
TCP: .... 1... = Push
TCP: .... .0.. = (No reset)
TCP: .... ..0. = (No SYN)
TCP: .... ...0 = (No FIN)
TCP: Window = 16332
TCP: Checksum = 612D (correct)
TCP: Urgent pointer = 0
TCP:
TCP: Options follow
TCP: No-Operation
TCP: No-Operation
TCP: Timestamp Option:
TCP: Timestamp value = 216
TCP: Timestamp echo reply = 216
TCP: [28 Bytes of data]
TCP:

ADDR  HEX                                                       ASCII
0000: 00 00 bc 03 5e 35 00 00 bc 03 5d f1 08 00 45 00 | ..Ľ.^5..Ľ.]ń..E.
0010: 00 50 00 02 00 00 40 06 a4 96 93 20 57 e8 93 20 | .P....@.¤-" Wč"
0020: 57 e7 af 12 af 13 d8 00 41 da d8 00 41 dc 80 18 | WçŻ.Ż.Ř.AÚŘ.AÜ€.
0030: 3f cc 61 2d 00 00 01 01 08 0a 00 00 00 d8 00 00 | ?Ěa-.........Ř..
0040: 00 d8 65 00 04 00 00 01 02 00 00 00 00 00 00 00 | .Ře.............
0050: 00 00 00 00 00 00 00 00 00 00 01 00 00 00       | ..............
```

Zdrojová IP adresa: **93 20 57 e8**

Cílová IP adresa: **93 20 57 e7**

Zdrojový port: **af 12**

Cílový port: **af 13**

Odpověď na požadavek RegisterSession: **65 00**

Paket 5: Požadavek Forward_Open

```
Source          Destination   Bytes Rel Time
Delta Time Abs time           Summary
-----------------------------------------------------------------------------
[147.32.87.231] [147.32.87.232] 154 0:00:00.022
0.003.928 02.12.2002 14:46:03 TCP: D=44818 S=44819

TCP: ----- TCP header -----
TCP:
TCP: Source port = 44819
TCP: Destination port = 44818
TCP: Sequence number = 3623895516
TCP: Next expected Seq number= 3623895604
TCP: Acknowledgment number = 3623895542
TCP: Data offset = 32 bytes
TCP: Reserved Bits: Reserved for Future Use (Not shown in the Hex Dump)
TCP: Flags = 18
TCP: ..0. .... = (No urgent pointer)
TCP: ...1 .... = Acknowledgment
TCP: .... 1... = Push
TCP: .... .0.. = (No reset)
TCP: .... ..0. = (No SYN)
TCP: .... ...0 = (No FIN)
TCP: Window = 4380
TCP: Checksum = B308 (correct)
TCP: Urgent pointer = 0
TCP:
TCP: Options follow
TCP: No-Operation
TCP: No-Operation
TCP: Timestamp Option:
TCP: Timestamp value = 216
TCP: Timestamp echo reply = 216
TCP: [88 Bytes of data]
TCP:

ADDR  HEX                                                       ASCII
0000: 00 00 bc 03 5d f1 00 00 bc 03 5e 35 08 00 45 00 | ..Ľ.]ń..Ľ.^5..E.
0010: 00 8c 00 04 00 00 40 06 a4 58 93 20 57 e7 93 20 | .Ś....@.¤X" Wç"
0020: 57 e8 af 13 af 12 d8 00 41 dc d8 00 41 f6 80 18 | WčŻ.Ż.Ř.AÜŘ.Aö€.
0030: 11 1c b3 08 00 00 01 01 08 0a 00 00 00 d8 00 00 | ..ł..........Ř..
0040: 00 d8 6f 00 40 00 00 01 02 00 00 00 00 00 00 00 | .Řo.@...........
0050: 00 01 00 01 00 00 00 00 00 00 00 00 00 00 00 00 | ................
0060: 02 00 00 00 00 00 b2 00 30 00 54 02 20 06 24 01 | ......².0.T. .$.
0070: 07 e8 00 00 0b 80 00 01 00 00 00 20 01 00 88 75 | .č...€..... ..u
0080: 05 00 00 00 00 00 e0 70 72 00 f6 43 e0 70 72 00 | ......řpr.öCřpr.
0090: f6 43 a3 03 01 00 20 02 24 01                   | öCŁ... .$.
```

Požadavek SendRRData: `6f 00`

Následují dvě `02 00` pole: adresové a datové, ID datového typu je `b2 00` (UCMM)

Délka dat je `30 00`

Požadavek Forward_Open: `54`

Navržené Connection ID pro směr zdroj-cíl: `00 00 0b 80`

Connection ID pro opačný směr: `00 01 00 00`

Connection serial number: **00 20**

Komunikační cesta: **01 00 20 02 24 01** (backplane (ENET port 01), 1756-L1 in slot 01, MR 02, instance 01)

Paket 6: Odpověď na požadavek Forward_Open

```
Source          Destination   Bytes Rel Time
Delta Time Abs time           Summary
-----------------------------------------------------------------------------
[147.32.87.232] [147.32.87.231] 136 0:00:00.038
0.015.610 02.12.2002 14:46:03 TCP: D=44819 S=44818

TCP: ----- TCP header -----
TCP:
TCP: Source port = 44818
TCP: Destination port = 44819
TCP: Sequence number = 3623895542
TCP: Next expected Seq number= 3623895612
TCP: Acknowledgment number = 3623895604
TCP: Data offset = 32 bytes
TCP: Reserved Bits: Reserved for Future Use (Not shown in the Hex Dump)
TCP: Flags = 18
TCP: ..0. .... = (No urgent pointer)
TCP: ...1 .... = Acknowledgment
TCP: .... 1... = Push
TCP: .... .0.. = (No reset)
TCP: .... ..0. = (No SYN)
TCP: .... ...0 = (No FIN)
TCP: Window = 16244
TCP: Checksum = 5469 (correct)
TCP: Urgent pointer = 0
TCP:
TCP: Options follow
TCP: No-Operation
TCP: No-Operation
TCP: Timestamp Option:
TCP: Timestamp value = 216
TCP: Timestamp echo reply = 216
TCP: [70 Bytes of data]
TCP:

ADDR  HEX                                                       ASCII
0000: 00 00 bc 03 5e 35 00 00 bc 03 5d f1 08 00 45 00 | ..Ľ.^5..Ľ.]ń..E.
0010: 00 7a 00 03 00 00 40 06 a4 6b 93 20 57 e8 93 20 | .z....@.¤k" Wč"
0020: 57 e7 af 12 af 13 d8 00 41 f6 d8 00 42 34 80 18 | WçŻ.Ż.Ř.AöŘ.B4€.
0030: 3f 74 54 69 00 00 01 01 08 0a 00 00 00 d8 00 00 | ?tTi.........Ř..
0040: 00 d8 6f 00 2e 00 00 01 02 00 00 00 00 00 00 00 | .Řo.............
0050: 00 01 00 01 00 00 00 00 00 00 00 00 00 00 00 00 | ................
0060: 02 00 00 00 00 00 b2 00 1e 00 d4 00 00 00 00 01 | ......²...Ô.....
0070: 00 00 00 01 00 00 00 20 01 00 88 75 05 00 e0 70 | ....... ..^u..ŕp
0080: 72 00 e0 70 72 00 00 00                         | r.ŕpr...
```

Odpověď na požadavek SendRRData: `6f 00`

Vybrané Connection ID pro směr původce-cíl: `00 01 00 00`

Connection ID pro opačný směr: `00 01 00 00`

Connection serial number: `00 20`

Paket 7: Požadavek Data Table Read

```
Source          Destination   Bytes Rel Time
Delta Time Abs time           Summary
-----------------------------------------------------------------------------
[147.32.87.231] [147.32.87.232] 122 0:00:00.045
0.006.802 02.12.2002 14:46:03 TCP: D=44818 S=44819

TCP: ----- TCP header -----
TCP:
TCP: Source port = 44819
TCP: Destination port = 44818
TCP: Sequence number = 3623895604
TCP: Next expected Seq number= 3623895660
TCP: Acknowledgment number = 3623895612
TCP: Data offset = 32 bytes
TCP: Reserved Bits: Reserved for Future Use (Not shown in the Hex Dump)
TCP: Flags = 18
TCP: ..0. .... = (No urgent pointer)
TCP: ...1 .... = Acknowledgment
TCP: .... 1... = Push
TCP: .... .0.. = (No reset)
TCP: .... ..0. = (No SYN)
TCP: .... ...0 = (No FIN)
TCP: Window = 4380
TCP: Checksum = 3E23 (correct)
TCP: Urgent pointer = 0
TCP:
TCP: Options follow
TCP: No-Operation
TCP: No-Operation
TCP: Timestamp Option:
TCP: Timestamp value = 216
TCP: Timestamp echo reply = 216
TCP: [56 Bytes of data]
TCP:

ADDR  HEX                                                       ASCII
0000: 00 00 bc 03 5d f1 00 00 bc 03 5e 35 08 00 45 00 | ..Ľ.]ń..Ľ.^5..E.
0010: 00 6c 00 05 00 00 40 06 a4 77 93 20 57 e7 93 20 | .l....@.¤w" Wç"
0020: 57 e8 af 13 af 12 d8 00 42 34 d8 00 42 3c 80 18 | WčŻ.Ż.Ř.B4Ř.B<€.
0030: 11 1c 3e 23 00 00 01 01 08 0a 00 00 00 d8 00 00 | ..>#.........Ř..
0040: 00 d8 70 00 20 00 00 01 02 00 00 00 00 00 00 00 | .Řp. ...........
0050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
0060: 02 00 a1 00 04 00 00 01 00 00 b1 00 0c 00 01 00 | ..¡.......±.....
0070: 4c 03 91 04 74 65 73 74 01 00                   | L.`.test..
```

Požadavek SendUnitData: **70 00**

Connection ID pro směr původce-cíl: **00 01 00 00**

Číslo paketu: **01 00**

Požadavek Data Table Read: na symbolický segment **91** tagu **test**

Paket 8: Odpověď na požadavek Data Table Read

```
Source          Destination   Bytes Rel Time
Delta Time Abs time           Summary
-----------------------------------------------------------------------------
[147.32.87.232] [147.32.87.231] 120 0:00:00.060
0.015.208 02.12.2002 14:46:03 TCP: D=44819 S=44818

TCP: ----- TCP header -----
TCP:
TCP: Source port = 44818
TCP: Destination port = 44819
TCP: Sequence number = 3623895612
TCP: Next expected Seq number= 3623895666
TCP: Acknowledgment number = 3623895660
TCP: Data offset = 32 bytes
TCP: Reserved Bits: Reserved for Future Use (Not shown in the Hex Dump)
TCP: Flags = 18
TCP: ..0. .... = (No urgent pointer)
TCP: ...1 .... = Acknowledgment
TCP: .... 1... = Push
TCP: .... .0.. = (No reset)
TCP: .... ..0. = (No SYN)
TCP: .... ...0 = (No FIN)
TCP: Window = 16188
TCP: Checksum = 0AAD (correct)
TCP: Urgent pointer = 0
TCP:
TCP: Options follow
TCP: No-Operation
TCP: No-Operation
TCP: Timestamp Option:
TCP: Timestamp value = 216
TCP: Timestamp echo reply = 216
TCP: [54 Bytes of data]
TCP:

ADDR  HEX                                                       ASCII
0000: 00 00 bc 03 5e 35 00 00 bc 03 5d f1 08 00 45 00 | ..Ľ.^5..Ľ.]ń..E.
0010: 00 6a 00 04 00 00 40 06 a4 7a 93 20 57 e8 93 20 | .j....@.¤z" Wč"
0020: 57 e7 af 12 af 13 d8 00 42 3c d8 00 42 6c 80 18 | WçŻ.Ż.Ř.B<Ř.Bl€.
0030: 3f 3c 0a ad 00 00 01 01 08 0a 00 00 00 d8 00 00 | ?<.-.........Ř..
0040: 00 d8 70 00 1e 00 00 01 02 00 00 00 00 00 00 00 | .Řp.............
0050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
0060: 02 00 a1 00 04 00 00 01 00 00 b1 00 0a 00 01 00 | ..¡.......±.....
0070: cc 00 00 00 c3 00 40 00                         | Ě...Ă.@.
```

Odpověd na požadavek SendRRData: `70 00`

Connection ID pro směr cíl-původce: `00 01 00 00`

Číslo paketu: `01 00`

Odpověď na požadavek Data Table Read: `cc`

Typ integer: `c3 00` , hodnota `40 00` (0x0040)

A.2 Detailní rozbor implicitní komunikace

V dalším textu budou použity pro zvýšení přehlednosti různé barvy textu. **Modře** je označeno záhlaví rámce ethernet, **zeleně** je označeno záhláví IP protokolu, **černě** je označeno záhlaví protokolu transportní vrstvy (nebo protokolu IGMP). **Červeně** je označena datová oblast paketu TCP nebo UDP.

V hexadecimálním pohledu na data jsou významné části označeny tučně a příslušnou barvou, pod tímto pohledem je opět s použitím těchto barev napsaná legenda.

Vytvoření přímého spojení pro implicitní komunikaci (direct I/O communication) se skládá z těchto kroků:

Paket	Použitý protokol	Význam
1	TCP	Požadavek ListServices
2	TCP	Odpověď ListServices
3	TCP	Požadavek RegisterSession
4	TCP	Odpověď RegisterSession
5	TCP	Požadavek Forward_Open
6	TCP	Odpověď Forward_Open
7	TCP	Požadavek Forward_Open
8	TCP	Odpověď Forward_Open
9	IGMP	Zpráva o členství v multicastové skupině

Tabulka 7.1 Navázání přímého I/O spojení

Následná komunikace se skládá z těchto kroků:

Paket	Použitý protokol	Význam
10	UDP	požadavek na I/O data
11	UDP	I/O data

Tabulka 7.2 Implicitní I/O komunikace

Paket 1: Požadavek ListServices

```
Source          Destination  Bytes Rel Time
Delta Time Abs time           Summary
-----------------------------------------------------------------------------
[147.32.87.231] [147.32.87.232] 90 0:00:00.004
0.001.824 02.12.2002 12:36:33 TCP: D=44818 S=44819

TCP: ----- TCP header -----
TCP:
TCP: Source port = 44819
TCP: Destination port = 44818
TCP: Sequence number = 870117633
TCP: Next expected Seq number= 870117657
TCP: Acknowledgment number = 1220733971
TCP: Data offset = 32 bytes
TCP: Reserved Bits: Reserved for Future Use (Not shown in the Hex Dump)
TCP: Flags = 18
TCP: ..0. .... = (No urgent pointer)
TCP: ...1 .... = Acknowledgment
TCP: .... 1... = Push
TCP: .... .0.. = (No reset)
TCP: .... ..0. = (No SYN)
TCP: .... ...0 = (No FIN)
TCP: Window = 4380
TCP: Checksum = 5F92 (correct)
TCP: Urgent pointer = 0
TCP:
TCP: Options follow
TCP: No-Operation
TCP: No-Operation
TCP: Timestamp Option:
TCP: Timestamp value = 14847
TCP: Timestamp echo reply = 14853
TCP: [24 Bytes of data]
TCP:

ADDR  HEX                                                  ASCII
0000: 00 00 bc 03 5d f1 00 00 bc 03 5e 35 08 00 45 00 | ..Ľ.]ń..Ľ.^5..E.
0010: 00 4c 5d f4 00 00 40 06 46 a8 93 20 57 e7 93 20 | .L]ô..@.F¨" Wç" 
0020: 57 e8 af 13 af 12 33 dc f1 01 48 c2 ec 13 80 18 | WčŻ.Ż.3Üń.HÂě.€.
0030: 11 1c 5f 92 00 00 01 01 08 0a 00 00 39 ff 00 00 | .._'........9ÿ..
0040: 3a 05 04 00 00 00 00 00 00 00 00 00 00 00 00 00 | :...............
0050: 00 00 00 00 00 00 00 00 00 00                   | ..........
```

Cílová MAC adresa: `00 00 bc 03 5d f1` Zdrojová MAC adresa: `00 00 bc 03 5e 35`

Zdrojová IP adresa: `93 20 57 e7`, cílová IP adresa: `93 20 57 e8`

Zdrojový port: `af 13`, cílový port: `af 12`

Požadavek ListServices `04 00`

Paket 2: Odpověď ListServices

```
Source          Destination  Bytes Rel Time
Delta Time Abs time            Summary
-----------------------------------------------------------------------------
[147.32.87.232] [147.32.87.231] 116 0:00:00.009
0.005.347 02.12.2002 12:36:33 TCP: D=44819 S=44818

TCP: ----- TCP header -----
TCP:
TCP: Source port = 44818
TCP: Destination port = 44819
TCP: Sequence number = 1220733971
TCP: Next expected Seq number= 1220734021
TCP: Acknowledgment number = 870117657
TCP: Data offset = 32 bytes
TCP: Reserved Bits: Reserved for Future Use (Not shown in the Hex Dump)
TCP: Flags = 18
TCP: ..0. .... = (No urgent pointer)
TCP: ...1 .... = Acknowledgment
TCP: .... 1... = Push
TCP: .... .0.. = (No reset)
TCP: .... ..0. = (No SYN)
TCP: .... ...0 = (No FIN)
TCP: Window = 16360
TCP: Checksum = 178C (correct)
TCP: Urgent pointer = 0
TCP:
TCP: Options follow
TCP: No-Operation
TCP: No-Operation
TCP: Timestamp Option:
TCP: Timestamp value = 14853
TCP: Timestamp echo reply = 14847
TCP: [50 Bytes of data]
TCP:

ADDR  HEX                                                ASCII
0000: 00 00 bc 03 5e 35 00 00 bc 03 5d f1 08 00 45 00 | ..Ľ.^5..Ľ.]ń..E.
0010: 00 66 67 b1 00 00 40 06 3c d1 93 20 57 e8 93 20 | .fg±..@.<Ń" Wč"
0020: 57 e7 af 12 af 13 48 c2 ec 13 33 dc f1 19 80 18 | WçŻ.Ż.HÂě.3Üń.€.
0030: 3f e8 17 8c 00 00 01 01 08 0a 00 00 3a 05 00 00 | ?č.Ś........:...
0040: 39 ff 04 00 1a 00 00 00 00 00 00 00 00 00 00 00 | 9ÿ..............
0050: 00 00 00 00 00 00 00 00 00 00 01 00 00 01 14 00 | ................
0060: 01 00 20 01 43 6f 6d 6d 75 6e 69 63 61 74 69 6f | .. .Communicatio
0070: 6e 73 00 00                                     | ns..
```

Cílová MAC adresa: `00 00 bc 03 5e 35` Zdrojová MAC adresa: `00 00 bc 03 5d f1`

Zdrojová IP adresa: `93 20 57 e8` , cílová IP adresa: `93 20 57 e7`

Zdrojový port: `af 12` , cílový port: `af 13`

Odpověď ListServices `04 00`

Jméno služby: **Communications**

Paket 3: Požadavek RegisterSession

```
Source          Destination   Bytes  Rel Time
Delta Time Abs time            Summary
-------------------------------------------------------------
[147.32.87.231] [147.32.87.232] 94 0:00:00.013
0.003.502 02.12.2002 12:36:33 TCP: D=44818 S=44819

TCP: ----- TCP header -----
TCP:
TCP: Source port = 44819
TCP: Destination port = 44818
TCP: Sequence number = 870117657
TCP: Next expected Seq number= 870117685
TCP: Acknowledgment number = 1220734021
TCP: Data offset = 32 bytes
TCP: Reserved Bits: Reserved for Future Use (Not shown in the Hex Dump)
TCP: Flags = 18
TCP: ..0. .... = (No urgent pointer)
TCP: ...1 .... = Acknowledgment
TCP: .... 1... = Push
TCP: .... .0.. = (No reset)
TCP: .... ..0. = (No SYN)
TCP: .... ...0 = (No FIN)
TCP: Window = 4380
TCP: Checksum = F943 (correct)
TCP: Urgent pointer = 0
TCP:
TCP: Options follow
TCP: No-Operation
TCP: No-Operation
TCP: Timestamp Option:
TCP: Timestamp value = 14847
TCP: Timestamp echo reply = 14853
TCP: [28 Bytes of data]
TCP:

ADDR  HEX                                                  ASCII
0000: 00 00 bc 03 5d f1 00 00 bc 03 5e 35 08 00 45 00 | ..Ľ.]ń..Ľ.^5..E.
0010: 00 50 5d f5 00 00 40 06 46 a3 93 20 57 e7 93 20 | .P]ő..@.FŁ" Wç" 
0020: 57 e8 af 13 af 12 33 dc f1 19 48 c2 ec 45 80 18 | WčŻ.Ż.3Üń.HÂěE€.
0030: 11 1c f9 43 00 00 01 01 08 0a 00 00 39 ff 00 00 | ..ůC........9ÿ..
0040: 3a 05 65 00 04 00 00 00 00 00 00 00 00 00 00 00 | :.e.............
0050: 00 00 00 00 00 00 00 00 00 00 01 00 00 00       | ..............
```

Cílová MAC adresa: 00 00 bc 03 5d f1 Zdrojová MAC adresa: 00 00 bc 03 5e 35

Zdrojová IP adresa: 93 20 57 e7, cílová IP adresa: 93 20 57 e8

Zdrojový port: af 13, cílový port: af 12

Požadavek RegisterSession: 65 00

Paket 4: Odpověď RegisterSession

```
Source          Destination   Bytes  Rel Time
Delta Time Abs time            Summary
-----------------------------------------------------------------------------
[147.32.87.232] [147.32.87.231] 94 0:00:00.016
0.003.401 02.12.2002 12:36:33 TCP: D=44819 S=44818

TCP: ----- TCP header -----
TCP:
TCP: Source port = 44818
TCP: Destination port = 44819
TCP: Sequence number = 1220734021
TCP: Next expected Seq number= 1220734049
TCP: Acknowledgment number = 870117685
TCP: Data offset = 32 bytes
TCP: Reserved Bits: Reserved for Future Use (Not shown in the Hex Dump)
TCP: Flags = 18
TCP: ..0. .... = (No urgent pointer)
TCP: ...1 .... = Acknowledgment
TCP: .... 1... = Push
TCP: .... .0.. = (No reset)
TCP: .... ..0. = (No SYN)
TCP: .... ...0 = (No FIN)
TCP: Window = 16332
TCP: Checksum = C870 (correct)
TCP: Urgent pointer = 0
TCP:
TCP: Options follow
TCP: No-Operation
TCP: No-Operation
TCP: Timestamp Option:
TCP: Timestamp value = 14853
TCP: Timestamp echo reply = 14847
TCP: [28 Bytes of data]
TCP:

ADDR  HEX                                                       ASCII
0000: 00 00 bc 03 5e 35 00 00 bc 03 5d f1 08 00 45 00 | ..Ľ.^5..Ľ.]ń..E.
0010: 00 50 67 b2 00 00 40 06 3c e6 93 20 57 e8 93 20 | .Pg²..@.<ć" Wč"
0020: 57 e7 af 12 af 13 48 c2 ec 45 33 dc f1 35 80 18 | WçŻ.Ż.HÂěE3Üń5€.
0030: 3f cc c8 70 00 00 01 01 08 0a 00 00 3a 05 00 00 | ?ĚČp........:...
0040: 39 ff 65 00 04 00 00 05 02 02 00 00 00 00 00 00 | 9ÿe.............
0050: 00 00 00 00 00 00 00 00 00 00 01 00 00 00       | ..............
```

Cílová MAC adresa: `00 00 bc 03 5e 35` Zdrojová MAC adresa: `00 00 bc 03 5d f1`

Zdrojová IP adresa: `93 20 57 e8` , cílová IP adresa: `93 20 57 e7`

Zdrojový port: `af 12` , cílový port: `af 13`

Odpověď RegisterSession `65 00`

Paket 5: Požadavek Forward_Open (zasláno pomocí SendRRData)

```
Source          Destination   Bytes  Rel Time
Delta Time Abs time             Summary
-----------------------------------------------------------------------------
[147.32.87.231] [147.32.87.232] 162 0:00:00.021
0.004.180 02.12.2002 12:36:33 TCP: D=44818 S=44819

TCP: ----- TCP header -----
TCP:
TCP: Source port = 44819
TCP: Destination port = 44818
TCP: Sequence number = 870117685
TCP: Next expected Seq number= 870117781
TCP: Acknowledgment number = 1220734049
TCP: Data offset = 32 bytes
TCP: Reserved Bits: Reserved for Future Use (Not shown in the Hex Dump)
TCP: Flags = 18
TCP: ..0. .... = (No urgent pointer)
TCP: ...1 .... = Acknowledgment
TCP: .... 1... = Push
TCP: .... .0.. = (No reset)
TCP: .... ..0. = (No SYN)
TCP: .... ...0 = (No FIN)
TCP: Window = 4380
TCP: Checksum = F55B (correct)
TCP: Urgent pointer = 0
TCP:
TCP: Options follow
TCP: No-Operation
TCP: No-Operation
TCP: Timestamp Option:
TCP: Timestamp value = 14847
TCP: Timestamp echo reply = 14853
TCP: [96 Bytes of data]
TCP:

ADDR  HEX                                                       ASCII
0000: 00 00 bc 03 5d f1 00 00 bc 03 5e 35 08 00 45 00 | ..Ľ.]ń..Ľ.^5..E.
0010: 00 94 5d f6 00 00 40 06 46 5e 93 20 57 e7 93 20 | .”]ö..@.F^‟ Wç‟
0020: 57 e8 af 13 af 12 33 dc f1 35 48 c2 ec 61 80 18 | WčŻ.Ż.3Üń5HÂěa€.
0030: 11 1c f5 5b 00 00 01 01 08 0a 00 00 39 ff 00 00 | ..ő[........9ÿ..
0040: 3a 05 6f 00 48 00 00 05 02 02 00 00 00 00 00 00 | :.o.H...........
0050: 00 01 00 02 00 2c 00 00 00 00 00 00 00 00 00 00 | .....,..........
0060: 02 00 00 00 00 00 b2 00 38 00 54 02 20 06 24 01 | ......².8.T. .$.
0070: 05 99 00 00 00 00 00 00 00 00 00 00 01 00 88 75 | .™............_u
0080: 05 00 00 00 00 00 00 00 00 00 00 04 00 00 00 00 | ................
0090: 00 04 81 07 34 04 01 00 0c 00 14 00 82 06 20 01 | ..._.4.......,. .
00a0: 24 01                                           | $.
```

Požadavek SendRRData: `6f 00`

Následují `02 00` pole: 1 adresové a 1 datové

Délka paketu požadavku `38 00` (56 slov)

Požadavek Forward_Open: `54`

Paket 6: Odpověď na požadavek Forward_Open

```
Source          Destination    Bytes  Rel Time
Delta Time Abs time             Summary
-----------------------------------------------------------------------------
[147.32.87.232] [147.32.87.231] 136 0:00:00.029
0.008.401 02.12.2002 12:36:33 TCP: D=44819 S=44818

TCP: ----- TCP header -----
TCP:
TCP: Source port = 44818
TCP: Destination port = 44819
TCP: Sequence number = 1220734049
TCP: Next expected Seq number= 1220734119
TCP: Acknowledgment number = 870117781
TCP: Data offset = 32 bytes
TCP: Reserved Bits: Reserved for Future Use (Not shown in the Hex Dump)
TCP: Flags = 18
TCP: ..0. .... = (No urgent pointer)
TCP: ...1 .... = Acknowledgment
TCP: .... 1... = Push
TCP: .... .0.. = (No reset)
TCP: .... ..0. = (No SYN)
TCP: .... ...0 = (No FIN)
TCP: Window = 16236
TCP: Checksum = 6084 (correct)
TCP: Urgent pointer = 0
TCP:
TCP: Options follow
TCP: No-Operation
TCP: No-Operation
TCP: Timestamp Option:
TCP: Timestamp value = 14853
TCP: Timestamp echo reply = 14847
TCP: [70 Bytes of data]
TCP:

ADDR  HEX                                                       ASCII
0000: 00 00 bc 03 5e 35 00 00 bc 03 5d f1 08 00 45 00 | ..Ľ.^5..Ľ.]ń..E.
0010: 00 7a 67 b3 00 00 40 06 3c bb 93 20 57 e8 93 20 | .zgł..@.<»" Wč"
0020: 57 e7 af 12 af 13 48 c2 ec 61 33 dc f1 95 80 18 | WçŻ.Ż.HÂěa3Üń•€.
0030: 3f 6c 60 84 00 00 01 01 08 0a 00 00 3a 05 00 00 | ?l`„........:...
0040: 39 ff 6f 00 2e 00 00 05 02 02 00 00 00 00 00 00 | 9ÿo.............
0050: 00 01 00 02 00 2c 00 00 00 00 00 00 00 00 00 00 | .....,..........
0060: 02 00 00 00 00 00 b2 00 1e 00 d4 00 00 00 00 00 | ......²...Ô.....
0070: 00 00 00 00 00 00 00 00 01 00 88 75 05 00 00 00 | ..........._u....
0080: 00 00 00 00 00 00 00 00                         | ........
```

Odpověď SendRRData: `6f 00`

Následují `02 00` datová pole: 1 adresové a 1 datové

Délka paketu požadavku `1e 00`(30 slov)

Odpověď Forward_Open: `d4`

Paket 7: Požadavek Forward_Open

```
Source            Destination     Bytes Rel Time
Delta Time Abs time               Summary
-----------------------------------------------------------------------------
[147.32.87.231] [147.32.87.232] 526 0:00:00.110
0.051.451 02.12.2002 12:36:33 TCP: D=44818 S=44819

TCP: ----- TCP header -----
TCP:
TCP: Source port = 44819
TCP: Destination port = 44818
TCP: Sequence number = 870117781
TCP: Next expected Seq number= 870118241
TCP: Acknowledgment number = 1220734119
TCP: Data offset = 32 bytes
TCP: Reserved Bits: Reserved for Future Use (Not shown in the Hex Dump)
TCP: Flags = 18
TCP: ..0. .... = (No urgent pointer)
TCP: ...1 .... = Acknowledgment
TCP: .... 1... = Push
TCP: .... .0.. = (No reset)
TCP: .... ..0. = (No SYN)
TCP: .... ...0 = (No FIN)
TCP: Window = 4380
TCP: Checksum = B5E8 (correct)
TCP: Urgent pointer = 0
TCP:
TCP: Options follow
TCP: No-Operation
TCP: No-Operation
TCP: Timestamp Option:
TCP: Timestamp value = 14847
TCP: Timestamp echo reply = 14853
TCP: [460 Bytes of data]
TCP:

ADDR  HEX                                                       ASCII
0000: 00 00 bc 03 5d f1 00 00 bc 03 5e 35 08 00 45 00 | ..Ľ.]ń..Ľ.^5..E.
0010: 02 00 5d f8 00 00 40 06 44 f0 93 20 57 e7 93 20 | ..]ř..@.Dđ" Wç" 
0020: 57 e8 af 13 af 12 33 dc f1 95 48 c2 ec a7 80 18 | WčŻ.Ż.3Üń•HÂě§€.
0030: 11 1c b5 e8 00 00 01 01 08 0a 00 00 39 ff 00 00 | ..µč........9ÿ..
0040: 3a 05 6f 00 b4 01 00 05 02 02 00 00 00 00 00 00 | :.o.´...........
0050: 00 01 00 02 00 2d 00 00 00 00 00 00 00 00 00 00 | .....-..........
0060: 02 00 00 00 00 00 b2 00 a4 01 54 02 20 06 24 01 | ......².¤.T. .$.
0070: 05 99 00 00 00 00 00 00 00 00 21 00 01 00 88 75 | .™........!...^u
0080: 05 00 00 00 00 00 b0 71 0b 00 26 48 b0 71 0b 00 | ......°q..&H°q..
0090: 30 28 11 bd 01 0c 34 04 01 00 0a 00 09 00 81 05 | 0(.½..4.......^.
00a0: 20 04 24 22 2c 90 2c b4 80 b2 01 00 00 00 00 00 | .$",^,´€².....
00b0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
00c0: 20 c1 00 00 20 41 00 00 20 c1 00 00 20 41 00 00 | Á.. A.. Á.. A..
00d0: 20 c1 00 00 20 41 00 00 00 00 00 00 00 00 00 00 | Á.. A..........
00e0: 00 00 00 00 00 00 00 00 00 00 00 00 20 c1 00 00 | ............ Á..
00f0: 20 41 00 00 20 c1 00 00 20 41 00 00 20 c1 00 00 | A.. Á.. A.. Á..
0100: 20 41 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | A..............
0110: 00 00 00 00 00 00 00 00 20 c1 00 00 20 41 00 00 | ........ Á.. A..
0120: 20 c1 00 00 20 41 00 00 20 c1 00 00 20 41 00 00 | Á.. A.. Á.. A..
0130: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
0140: 00 00 00 00 20 c1 00 00 20 41 00 00 20 c1 00 00 | .... Á.. A.. Á..
```

```
0150: 20 41 00 00 20 c1 00 00 20 41 00 00 00 00 00 00 | A.. Á.. A......
0160: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
0170: 20 c1 00 00 20 41 00 00 20 c1 00 00 20 41 00 00 | Á.. A.. Á.. A..
0180: 20 c1 00 00 20 41 00 00 00 00 00 00 00 00 00 00 | Á.. A..........
0190: 00 00 00 00 00 00 00 00 00 00 00 00 20 c1 00 00 | ............ Á..
01a0: 20 41 00 00 20 c1 00 00 20 41 00 00 20 c1 00 00 | A.. Á.. A.. Á..
01b0: 20 41 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | A..............
01c0: 00 00 00 00 00 00 00 00 20 c1 00 00 20 41 00 00 | ........ Á.. A..
01d0: 20 c1 00 00 20 41 00 00 20 c1 00 00 20 41 00 00 | Á.. A.. Á.. A..
01e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
01f0: 00 00 00 00 20 c1 00 00 20 41 00 00 20 c1 00 00 | .... Á.. A.. Á..
0200: 20 41 00 00 20 c1 00 00 20 41 00 00 00 00       | A.. Á.. A....
```

Požadavek SendRRData: **6f 00**

Následují **02 00** datová pole: 1 adresové a 1 datové

Délka paketu požadavku **a4 01** (420 slov – pozor: little endian)

Požadavek Forward_Open: **54**

Paket 8: Odpověď na požadavek Forward_Open

```
Source          Destination     Bytes Rel Time
Delta Time Abs time             Summary
-----------------------------------------------------------------------------
[147.32.87.232] [147.32.87.231] 176 0:00:00.164
0.053.510 02.12.2002 12:36:33 TCP: D=44819 S=44818

TCP: ----- TCP header -----
TCP:
TCP: Source port = 44818
TCP: Destination port = 44819
TCP: Sequence number = 1220734119
TCP: Next expected Seq number= 1220734229
TCP: Acknowledgment number = 870118241
TCP: Data offset = 32 bytes
TCP: Reserved Bits: Reserved for Future Use (Not shown in the Hex Dump)
TCP: Flags = 18
TCP: ..0. .... = (No urgent pointer)
TCP: ...1 .... = Acknowledgment
TCP: .... 1... = Push
TCP: .... .0.. = (No reset)
TCP: .... ..0. = (No SYN)
TCP: .... ...0 = (No FIN)
TCP: Window = 15776
TCP: Checksum = 6F23 (correct)
TCP: Urgent pointer = 0
TCP:
TCP: Options follow
TCP: No-Operation
TCP: No-Operation
TCP: Timestamp Option:
TCP: Timestamp value = 14854
TCP: Timestamp echo reply = 14847
TCP: [110 Bytes of data]
TCP:

ADDR  HEX                                                       ASCII
0000: 00 00 bc 03 5e 35 00 00 bc 03 5d f1 08 00 45 00 | ..Ľ.^5..Ľ.]ń..E.
0010: 00 a2 67 b4 00 00 40 06 3c 92 93 20 57 e8 93 20 | .¢g´..@.<’" Wč"
0020: 57 e7 af 12 af 13 48 c2 ec a7 33 dc f3 61 80 18 | WçŻ.Ż.HÂě§3Üóa€.
0030: 3d a0 6f 23 00 00 01 01 08 0a 00 00 3a 06 00 00 | = o#........:...
0040: 39 ff 6f 00 56 00 00 05 02 02 00 00 00 00 00 00 | 9ÿo.V...........
0050: 00 01 00 02 00 2d 00 00 00 00 00 00 00 00 00 00 | .....-..........
0060: 04 00 00 00 00 00 b2 00 1e 00 d4 00 00 00 00 d2 | ......²...Ô....Ň
0070: 01 0a 0f ca 01 0a 21 00 01 00 88 75 05 00 b0 71 | ...Ę..!..._u..°q
0080: 0b 00 b0 71 0b 00 00 00 00 80 10 00 00 02 ff e9 | ..°q.....€....ÿé
0090: 00 00 00 00 00 00 00 00 00 00 00 00 01 80 10 00 | .............€..
00a0: 00 02 08 ae ef c0 02 00 00 00 00 00 00 00 00 00 | ...®ďŘ..........
```

Odpověď SendRRData: `6f 00`

Následují `04 00` datová pole, jedná se o zprávu od UCMM - `b2 00`

Délka paketu požadavku `1e 00` (30 slov)

Odpověď Forward_Open: `d4`

Druhá část zprávy obsahuje multicastovou adresu, pomocí které se bude uskutečňovat I/O komunikace: `ef c0 02 00` – 239.192.2.0

Paket 9: Zpráva o členství v multicastové skupině

```
Source          Destination Bytes Rel Time
Delta Time Abs time             Summary
-----------------------------------------------------------------------------
[147.32.87.231] [239.192.2.0] 60 0:00:00.170
0.006.011 02.12.2002 12:36:33 IGMP: Type 6, Ver2 Membership Report

IGMP: ----- IGMP header -----
IGMP:
IGMP: Version = 1
IGMP: Type = 6 (Ver2 Membership Report)
IGMP: Unused = 0x00
IGMP: Checksum = F83E (correct)
IGMP: Group Address = [239.192.2.0]
IGMP:

DLC: Frame padding= 18 bytes

ADDR  HEX                                                  ASCII
0000: 01 00 5e 40 02 00 00 00 bc 03 5e 35 08 00 45 00 | ..^@....Ľ.^5..E.
0010: 00 1c 5d f9 00 00 01 02 7f 1f 93 20 57 e7 ef c0 | ..]ů...._." WçďŘ
0020: 02 00 16 00 f8 3e ef c0 02 00 88 88 88 88 88 88 | ....ř>ďŘ.._______
0030: 88 88 88 88 88 88 88 88 88 88 88 88             | ____________
```

Uzel, který vytvořil spojení (147.32.87.231 - 93 20 57 e7), se přihlásil do multicastové skupiny 239.192.2.0 - ef c0 02 00. Na tuto adresu bude druhý uzel (147.32.87.232) zasílat I/O data pomocí UDP paketů.

Paket 10: Požadavek na I/O data

```
Source          Destination       Bytes Rel Time
Delta Time Abs time             Summary
-----------------------------------------------------------------------------
16 [147.32.87.231] [147.32.87.232] 98 0:00:00.919
0.013.937 02.12.2002 12:36:33 UDP: D=65513 S=65524 LEN=64

UDP: ----- UDP Header -----
UDP:
UDP: Source port = 65524 (Dynamic and/or Private)
UDP: Destination port = 65513 (Dynamic and/or Private)
UDP: Length = 64
UDP: Checksum = 4322 (correct)
UDP: [56 byte(s) of data]
UDP:

ADDR  HEX                                                 ASCII
0000: 00 00 bc 03 5d f1 00 00 bc 03 5e 35 08 00 45 00 | ..L.]ń..L.^5..E.
0010: 00 54 5d fb 00 00 40 11 46 8e 93 20 57 e7 93 20 | .T]ú..@.FŽ" Wç"
0020: 57 e8 ff f4 ff e9 00 40 43 22 02 00 02 80 08 00 | Wčÿôÿé.@C"...€..
0030: 00 d2 01 0a 00 00 00 00 b1 00 26 00 01 00 00 00 | .Ň......±.&.....
0040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
0050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
0060: 00 00                                           | ..
```

Paket 11: Zaslaná I/O data

```
Source          Destination   Bytes Rel Time
Delta Time Abs time             Summary
-----------------------------------------------------------------------------
[147.32.87.232] [239.192.2.0] 108 0:00:00.905
0.646.454  02.12.2002  12:36:33  Expert:  Time-to-live  expiring  UDP:  D=2222
S=65514

IP:
UDP: ----- UDP Header -----
UDP:
UDP: Source port = 65514 (Dynamic and/or Private)
UDP: Destination port = 2222
UDP: Length = 74
UDP: Checksum = 1DC3 (correct)
UDP: [66 byte(s) of data]
UDP:

ADDR  HEX                                                    ASCII
0000: 01 00 5e 40 02 00 00 00 bc 03 5d f1 08 00 45 00 | ..^@....Ľ.]ń..E.
0010: 00 5e 67 b5 00 00 01 11 75 11 93 20 57 e8 ef c0 | .^gµ....u.‟ WčďŘ
0020: 02 00 ff ea 08 ae 00 4a 1d c3 02 00 02 80 08 00 | ..ÿę.®.J.Ă...€..
0030: 0f ca 01 0a 00 00 00 00 b1 00 30 00 62 cf 00 00 | .Ę......±.0.bĎ..
0040: 00 00 00 00 00 00 00 00 00 00 20 ff 1f c1 00 00 | .......... ÿ.Á..
0050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
0060: 00 00 00 00 00 00 00 00 00 00 5a 50             | ..........ZP
```

Vzdálený modul I/O má IP adresou **93 20 57 e8** – 147.32.87.232

Tento modul zasílá I/O data na adresu **ef c0 02 00** – 239.192.2.0

Zdrojový port: **ff ea**

Cílový port: **08 ae**

Příloha B

B.1 Požadavek UCMM

Struktura	Jméno pole		Datový typ	Hodnota
Hlavička	Příkaz		UINT	SendRRData (0x6F)
	Délka		UINT	Délka datové části
	Session handle		UDINT	Handle vrácený příkazem RegisterSessing
	Status		UDINT	0
	Sender Context		Pole osmi USHORT	Lib. Kontext
	Options		UDINT	0
Data příslušná příkazu	Interface handle		UDINT	0 pro CIP
	Timeout		UINT	Timeout operace
	Zapouzdřený paket	Item count	UINT	2 (nasleduje 1 adresové a 1 datové pole)
		Address type ID	UINT	0 (pro UCMM)
		Délka adresy	UINT	0 (UCMM)
		Data type ID	UINT	0x00B2 (UCMM)
		Délka dat	UINT	Délka paketu požadvku objektu Message Router
		Paket požadavku MR	Pole USINT	Toto pole obsahuje paket CIP MR požadaveku.

Tabulka B.1 Požadavek UCMM

B.2 Odezva UCMM

Struktura	Jméno pole		Datový typ	Hodnota
Hlavička	Příkaz		UINT	SendRRData (0x6F)
	Délka		UINT	Délka datové části
	Session handle		UDINT	Handle vrácený příkazem RegisterSessing
	Status		UDINT	0
	Sender Context		Pole osmi USHORT	Stejný kontext z požadavku
	Options		UDINT	0
Data příslušná příkazu	Interface handle		UDINT	0 pro CIP
	Timeout		UINT	Timeout operace
	Zapouzdřený paket	Item count	UINT	2 (nasleduje 1 adresové a 1 datové pole)
		Address type ID	UINT	0 (pro UCMM)
		Délka adresy	UINT	0 (UCMM)
		Data type ID	UINT	0x00B2 (UCMM)
		Délka dat	UINT	Délka paketu odezvy objektu Message Router
		Paket požadavku MR	Pole USINT	Toto pole obsahuje paket CIP MR odezvy.

Tabulka B.2 Odpověd UCMM

Příloha C

C.1 Elektromechanické vlastnosti průmyslových EtherNet/IP kabelů

Průmyslová EtherNet/IP média (měděná i optická) musí splňovat tyto mechanické a elektrické požadavky:

Odolnost	Kriterium	Průmyslový standard
Vibrace		IEC 60068-2-6
Frekvenční rozsah	10 – 500 Hz	
Zrychlení	5 g	
Posunutí	0,3 mm	
Rázy		IEC 60068-2-27
Zrychlení (v provozu)	30 g	
Zrychlení (mimo provoz)	50 g	
Teplota		IEC 60068-2-1,2
Pracovní rozsah	0°C – +60°C	
Skladování	-40°C – +85°C	
Vlhkost		IEC 60068-2-30
	5 až 95% rel. vlhkost	
Krytí		IEC 60529
	Minimálně IP20	
Napěťová pevnost konektoru		IEC 60512-1
Kontakt/kontakt	1000 V	
Kontakt/panel	1500 V	

Tabulka C.1 Elektromechanické vlastnosti EtherNet/IP kabelů pro průmysl

Ve velmi zašuměných prostředích je kritickým místem při ovlivnění výkonu kabel. Průmyslová média pro síť EtherNet/IP musí dále splňovat mechanické, elektrické a výkonnostní požadavky uvedené v následující tabulce:.

Vlastnosti	Typ	
Elektrické	Stíněný	Nestíněný
Vodiče	2 nebo 4 páry + stínění	2 nebo 4 páry
Útlum	$\text{útlum}(f) \leq 1.967 \cdot \sqrt{f} + 0.023 \cdot f + \frac{0.05}{\sqrt{f}}$	
Impedance	95 - 110Ω 1 – 4 MHz 95 - 107Ω 4 – 100MHz	
RL	1 – 10MHz $20 + 6\log_{10} f$ 10 – 20MHz 26 20 – 100MHz $26 - 5\log_{10}(f/20)$	
NEXT Loss	$NEXT(f) \geq 64 - 15\log_{10} f$ dB	
Účinnost stínění	TBD	n/a
Cupp	≤150pF/100m	
DCR	9,38Ω/100m	
Common Mode Rejection	TBD	TBD
DCR Unbalance	5%	5%

Tabulka C.2 Elektrické vlastnosti EtherNet/IP kabelů pro průmysl