ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA ELEKTROTECHNICKÁ

DIPLOMOVÁ PRÁCE

EtherNet/IP pro průmyslové řízení

Praha, 2007 Autor: Zdeněk Prokůpek

Prohlášení

Prohlašuji, že jsem 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 platném znění.

V Praze dne _____ _____

podpis

Poděkování

Rád bych poděkoval vedoucímu mé diplomové práce ing. Jindřichu Fukovi, bez jehož pomoci, rad a odborných připomínek by tato práce nemohla vzniknout. Můj vděk také patří všem, kteří mě při práci podporovali.

Abstrakt

EtherNet/IP pro průmyslové řízení

Práce se zabývá komunikačními linkami architektury NetLinx společnosti Rockwell Automation se zaměřením na ControlNet a EtherNet/IP. Realizace sítě pro testování komunikace EtherNet/IP je v laboratoři provedena prostřednictvím přepínače s managementem Cisco Catalyst 2960. Pro určení způsobu komunikace mezi automaty prostřednictvím EtherNet/IP se používá vlastnosti přepínače zrcadlení portů a programu Wireshark, určeného k odposlechu a analýze komunikace na sítí Ethernet. Pro určení zatížení sítě se používají vytvořené Flash aplikace, které získávají data prostřednictvím SNMP protokolu. Roli SNMP managera plní PHP skript periodicky spouštěný na serveru a roli SNMP agentů plní komunikační moduly Allen-Bradley a přepínač Cisco.

Kvůli prezentaci laboratoře byly vytvořeny webové stránky, jejichž obsah je věnován zejména popisu zařízení Allen-Bradley a modelů nacházejících se v laboratoři. Tyto webové stránky jsou navrženy tak, aby umožnili snadnou aktualizaci obsahu. Pro model dvouosého soustruhu byla vytvořena Flash aplikace umožňující výhradní řízení tohoto modelu a pro zajištění výhradního řízení je v automatu ControlLogix vytvořen jednoduchý rezervační systém.

Abstract

EtherNet/IP in Industrial Control

This dissertation deals with the communications lines of NetLinx architecture from the company Rockwell Automation, sight on ControlNet and EtherNet/IP networks. In the test room nets realization for testing communicating is carried out through a switch with management Cisco Catalyst 2960. Feature of the switch, called port minoring, and the program Wireshark intended for monitoring and analysing of the communication on the Ethernet network, are used to determine of a way of the communication among automatics through EtherNet/IP. Flash applications that attract data via SNMP protocol are used to determine net load. Then PHP script, which is periodically triggered on the server performs the role of SNMP manager and the communications modules Allen-Bradley and the switch Cisco perform the role of the SNMP agents.

Because of the laboratory presentation new web pages were created, whereas the content of the web is dedicated to description of the establishment Allen-Bradley especially and the models placed at the test room. The web pages are established to provide easy up-dating of the content. For the model of two axis sticker machine a Flash application enabling exclusive control of this model was also created and a simple reservation system in the ControlLogix automat was integrated to provide the exclusive control.

Obsah

1 ÚVOD ... 10

2 KOMUNIKAČNÍ LINKY ROCKWELL AUTOMATION ... 11

2.1 ÚVOD ... 11
2.2 OSI MODEL (OPEN SYSTEM INTERCONECT) ... 12
2.3 CIP (COMMON INDUSTRIAL PROTOCOL) ... 13
2.3.1 Definice pojmů ... 13
2.3.2 ControlNet ... 14
2.3.2.1 Fyzická vrstva ... 15
2.3.2.2 Topologie ... 16
2.3.2.3 Přístup na médium CTDMA ... 16
2.3.2.4 Popis komunikačního rámce ... 18
2.3.3 EtherNet/IP ... 18
2.3.3.1 Fyzická vrstva ... 19
2.3.3.2 Topologie ... 20
2.3.3.3 Popis komunikačního rámce ... 20
2.3.3.3.1 EtherNet/IP zapouzdřování (Encapsulation) ... 20
2.3.3.3.1.1 Zapouzdřené příkazy, příklady a jejich význam ... 21
2.3.3.3.1.2 Zapouzdřená data odeslaná jako explicitní zpráva ... 22
2.3.3.3.1.3 Zapouzdřená data odeslaná jako implicitní zpráva ... 23
2.3.4 Shrnutí ... 23
2.3.5 Porovnání komunikačních linek architektury NetLinx ... 25
2.4 ETHERNET ... 25
2.4.1 Fyzická vrstva ... 26
2.4.2 Topologie ... 26
2.4.3 Přístup na médium CSMA/CD ... 27
2.4.4 Popis základního komunikačního rámce Ethernet ... 28

3 POROVNÁNÍ KOMUNIKAČNÍCH LINEK ROCKWELL AUTOMATION S DALŠÍMI LINKAMI PŘEDNÍCH SVĚTOVÝCH VÝROBCŮ ... 29

4 WEBOVÉ STRÁNKY LABORATOŘE ALLEN-BRADLEY K23 ... 32

4.1 POPIS FUNKCÍ ... 32
4.2 WEBOVÉ ŘÍZENÍ SOUSTRUHU ... 33
4.2.1 Úvod ... 33
4.2.2 Rozšíření původního řídicího systému ... 34
4.2.3 Princip komunikace internetový prohlížeč – server – procesor ... 34
4.2.3.1 Detailní popis komunikace ... 34
4.2.3.2 Omezení komunikace ... 35
4.2.4 Princip rezervace ... 35
4.2.5 Vlastní řízení soustruhu ActionScriptem 2.0 (Flash) ... 36
4.3 STATISTIKA ZATÍŽENÍ SÍTĚ ... 37
4.3.1 Úvod ... 37
4.3.2 Princip funkce ... 37
4.3.3 Určení zatížení v kBajtech/s ... 40
4.3.4 Omezení vytvořených programů ... 40
4.3.5 Statistika zatížení ethernetových komunikačních modulů Allen-Bradley ... 41
4.3.6 Statistika zatížení jednotlivých portů přepínače Cisco Catalyst 2960 ... 42
4.3.6.1 Nastavení SNMP a RMON na přepínači ... 42

5 TESTOVÁNÍ PROPUSTNOSTI ETHERNET/IP ... 44

5.1 ZAPOJENÍ SÍTĚ V LABORATOŘI ... 44
5.1.1 Úvod ... 44
5.1.2 Konfigurace přepínače ... 45
5.1.2.1 Nastavení synchronizace času pomocí NTP ... 46
5.1.2.2 Nastavení DHCP snooping ... 46
5.1.2.3 IGMP protocol ... 47
5.1.2.3.1 IGMPv1 ... 47
5.1.2.3.2 IGMPv2 ... 47
5.1.2.3.3 IGMP snooping ... 47

5.2 IMPLICITNÍ KOMUNIKACE ... 49
5.2.1 Navázání a ukončení komunikace typu producent/konzument ... 49
5.2.2 Způsob navázaní dalšího konzumenta při komunikaci typu producent/konzument ... 53
5.2.3 Přenášení dvou digitálních vstupů rámově optimalizovaným spojením (Rack Optimization) ... 56
5.2.4 Přímé spojení (Direct Connection) ... 57
5.3 VÝPOČET ZATÍŽENÍ KOMUNIKAČNÍCH MODULŮ ... 57
5.3.1 Rámově optimalizované spojení ... 58
5.3.1.1 Přenášení pouze digitálních vstupů, režim multiplekeeper ... 58
5.3.1.2 Přenášení digitálních vstupů společně s digitálními výstupy ... 59
5.3.2 Přímé spojení ... 61
5.3.2.1 Určení zatížení při přenášení digitálních vstupů 1756-IB16 ... 61
5.3.2.2 Určení zatížení při přenášení izolovaných digitálních vstupů 1756-IB16I a digitálních výstupů 1756-OB16 ... 63
5.3.3 Produkovaní a konzumovaní tagů ... 64
5.4 PŘÍKLAD VÝPOČTU ZATÍŽENÍ KOMUNIKAČNÍCH MODULŮ 1756-ENBT/A ... 65
5.4.1 Určení zatížení 1756-ENBT/A komunikačního modulu při komunikaci producent/konzument ... 65
5.4.2 Určení zatížení 1756-ENBT/A komunikačních modulů při přímém spojení ... 67
5.5 MAXIMÁLNÍ KOMUNIKAČNÍ RYCHLOST MODULŮ 1756-ENBT/A REVIZE 3.6 ... 69
5.6 VLIV IGMP SNOOPINGU NA ZATÍŽENÍ NEVYUŽÍVANÝCH PORTŮ PŘEPÍNAČE PŘI IMPLICITNÍ KOMUNIKACI ... 70

6 SROVNÁNÍ SÍTĚ ETHERNET/IP A CONTROLNET ... 71

6.1 CONTROLNET ... 71
6.2 ETHERNET/IP ... 72

7 ZÁVĚR ... 73

8 LITERATURA ... 75

9 SEZNAM SOFTWARE ... 78

PŘÍLOHA A KOMUNIKACE POMOCÍ MODULŮ 1756-ENBT/A A 1756-EWEB/A ... 79

A.1 WEBOVÉ ŘÍZENÍ POMOCÍ 1756-EWEB, PODROBNÝ NÁVOD ... 79
A.2 KOMUNIKACE PRODUCENT/KONZUMENT 1756-ENBT, PŘENÁŠENÍ PROMĚNNÉ ... 85
A.3 KOMUNIKACE PRODUCENT/KONZUMENT 1756-ENBT, PŘENÁŠENÍ DIGITÁLNÍCH I/O ... 99

PŘÍLOHA B WEBOVÉ STRÁNKY LABORATOŘE ALLEN-BRADLEY K23 ... 109

PŘÍLOHA C ADMINISTRATIVNÍ ROZHRANÍ STRÁNEK ... 110

C.1 ADMINISTRACE WEBOVÝCH STRÁNEK ... 110
C.2 ADMINISTRACE MENU WEBOVÝCH STRÁNEK ... 110

PŘÍLOHA D ZPROVOZNĚNÍ PŘEPÍNAČE CISCO CATALYST 2960 ... 111

D.1 BOOTOVACÍ PROCES ... 111
D.2 ZPROVOZNĚNÍ PŘEPÍNAČE CISCO CATALYST 2960 ... 111

PŘÍLOHA E ZÁKLADNÍ POPIS FUNKCÍ PŘEPÍNAČE CISCO ... 113

PŘÍLOHA F ROZBOR PŘENÁŠENÝCH DAT PŘI RACK OPITIMIZATION KOMUNIKACI ... 118

PŘÍLOHA G MULTICASTOVÉ SKUPINY VYUŽITÉ PRO KOMUNIKACI ... 119

Seznam obrázků

OBR. 2.1 ARCHITEKTURA NETLINX, KOMPLETNÍ ŘEŠENÍ PRŮMYSLOVÉ SÍTĚ 11
OBR. 2.2 SPOJITOST CIP S OSI MODELEM 13
OBR. 2.3 CIP EXPLICITNÍ POSÍLÁNÍ ZPRÁV 14
OBR. 2.4 CIP MULTICASTOVÁ KOMUNIKACE I/O SPOJENÍ (TYP PRODUCENT/KONZUMENT) 14
OBR. 2.5 SPOJITOST OSI MODELU A SÍTĚ CONTROLNET 15
OBR. 2.6 ZÁVISLOST DÉLKY SEGMENTU NA POČTU ODBOČOVAČŮ 15
OBR. 2.7 PŘÍKLADY TOPOLOGIE ZAPOJENÍ SÍTĚ CONTROLNET 16
OBR. 2.8 PŘÍSTUP NA MÉDIUM CTDMA 16
OBR. 2.9 PLÁNOVANÝ PŘENOS 17
OBR. 2.10 NEPLÁNOVANÝ PŘENOS 17
OBR. 2.11 FORMÁT KOMUNIKAČNÍHO RÁMCE CONTROLNET 18
OBR. 2.12 SPOJITOST OSI MODELU A ETHERNET/IP 19
OBR. 2.13 PŘÍKLAD MOŽNÉ TOPOLOGIE ETHERNET/IP 20
OBR. 2.14 ETHERNET RÁMEC, DO KTERÉHO SE ZAPOUZDŘUJE CIP ZPRÁVA 21
OBR. 2.15 ZAPOUZDŘENÍ EXPLICITNÍ ZPRÁVY UCCM REQUEST 22
OBR. 2.16 ZAPOUZDŘENÍ IMPLICITNÍ ZPRÁVY I/O 23
OBR. 2.17 TYPICKÉ ZAPOJENÍ ZAŘÍZENÍ POMOCÍ ETHERNET/IP 23
OBR. 2.18 TYPICKÉ ZAPOJENÍ ZAŘÍZENÍ POMOCÍ CONTROLNET 24
OBR. 2.19 TYPICKÉ ZAPOJENÍ ZAŘÍZENÍ POMOCÍ DEVICENET 24
OBR. 2.20 SOUVISLOST OSI MODELU A SÍTĚ ETHERNET 26
OBR. 2.21 PŘÍKLAD MOŽNÝCH TOPOLOGIÍ SÍTĚ ETHERNET 27
OBR. 2.22 ZÁKLADNÍ KOMUNIKAČNÍ RÁMEC 28
OBR. 3.2 PŘÍKLAD ŘÍZENÍ VÝROBY NA VŠECH ÚROVNÍCH NETLINX. SPOJITOST S OSI MODELEM 29
OBR. 3.1 PŘÍKLAD ŘÍZENÍ VÝROBY NA VŠECH ÚROVNÍCH PROFINET. SPOJITOST S OSI MODELEM 30
OBR. 4.1 STRUKTURA WEBOVÝCH STRÁNEK 32
OBR. 4.2 ZÁKLADNÍ PRINCIP WEBOVÉHO ŘÍZENÍ SOUSTRUHU 33
OBR. 4.3 WEBOVÉ ŘÍZENÍ SOUSTRUHU ACTIONSCRIPTEM (FLASH) 36
OBR. 4.4 STROMOVÁ STRUKTURA MIB 37
OBR. 4.5 KOMUNIKACE SNMP 38
OBR. 4.6 PRINCIP FUNKCE WEBOVÝCH STRÁNEK 38
OBR. 4.7 DEFINICE XML SOUBORU NEZBYTNÁ PRO ZÍSKÁNÍ INFORMACÍ O STAVU VYTÍŽENÍ SÍTĚ Z JEDNOTLIVÝCH ZAŘÍZENÍ 39
OBR. 4.8 STATISTIKA VYTÍŽENÍ ETHERNETOVÝCH MODULŮ ALLEN-BRADLEY 42
OBR. 4.9 STATISTIKA VYTÍŽENÍ JEDNOTLIVÝCH PORTŮ PŘEPÍNAČE CISCO CATALYST 2960 43
OBR. 5.1 SCHÉMA PŘIPOJENÍ AUTOMATŮ V LABORATOŘI 44
OBR. 5.2 KOMUNIKACE MEZI MODULY PŘI IMPLICITNÍM SPOJENÍ TYPU PRODUCENT/KONZUMENT 50
OBR. 5.3 PŘIPOJENÍ DALŠÍHO KONZUMENTA 54
OBR. 5.4 RÁMOVĚ OPTIMALIZOVANÝ ZPŮSOB KOMUNIKACE 56
OBR. 5.5 CHYBA SPOJENÍ PŘI NEVHODNĚ ZVOLENÝCH RPI 58
OBR. 5.6 PRINCIP KOMUNIKACE RÁMOVĚ OPTIMALIZOVANÉHO SPOJENÍ PRO PŘENOS DIGITÁLNÍCH VSTUPŮ 59
OBR. 5.7 PRINCIP KOMUNIKACE RÁMOVĚ OPTIMALIZOVANÉHO SPOJENÍ PŘI ZAČLENĚNÍ DIGITÁLNÍCH VÝSTUPŮ . 60
OBR. 5.8 PŘÍMÉ SPOJENÍ A VÝPOČET ZATÍŽENÍ SÍTĚ PŘI PŘENÁŠENÍ DIGITÁLNÍCH VSTUPŮ, MODUL 1756-IB16 62
OBR. 5.9 PŘÍMÉ SPOJENÍ (DIRECT CONNECTION) A VÝPOČET ZATÍŽENÍ SÍTĚ PŘI PŘENÁŠENÍ IZOLOVANÝCH DIGITÁLNÍCH VSTUPŮ (MODUL 1756-IB16) A DIGITÁLNÍCH VÝSTUPŮ 1756-OB16 64
OBR. 5.10 PRINCIP KOMUNIKACE PŘI SPOJENÍ TYPU PRODUCENT/KONZUMENT 65
OBR. 5.11 ZATÍŽENÍ KOMUNIKAČNÍCH MODULŮ, KOMUNIKACE PRODUCENT/KONZUMENT. NENÍ ZAPNUT IGMP SNOOPING 66
OBR. 5.12 ZATÍŽENÍ KOMUNIKAČNÍCH MODULŮ, PŘÍMÁ KOMUNIKACE 68
OBR. 5.13 PŘEKROČENÍ MAXIMÁLNÍ KOMUNIKAČNÍ RYCHLOSTI 70
OBR. 5.14 VLIV IGMP SNOOPING NA ZATÍŽENÍ PORTŮ PŘEPÍNAČE 70
OBR. A.1 RSLINX ZJIŠTĚNÍ HARDWAROVÉ KONFIGURACE 79
OBR. A.2 ZALOŽENÍ NOVÉHO PROJEKTU 80
OBR. A.3 INFORMACE O VERZI SPUŠTĚNÉHO PROGRAMU 80
OBR. A.4 PŘIDÁNÍ ZVOLENÉHO MODULU DO I/O KONFIGURACE 81
OBR. A.5 VYTVOŘENÍ NOVÉ PROMĚNNÉ 82
OBR. A.6 NASTAVENÍ CESTY PROJEKTU 82
OBR. A.7 WEBOÉ ROZHRANÍ SERVER 1756-EWEB 84

OBR. A.8 RSLINX, ZJIŠTĚNÍ HARDWAROVÉ KONFIGURACE 85
OBR. A.9 ZALOŽENÍ NOVÉHO PROJEKTU 86
OBR. A.10 OVĚŘENÍ SPUŠTĚNÍ RSLOGIX V ODPOVÍDAJÍCÍ VERZI 86
OBR. A.11 ZAČLENĚNÍ KOMUNIKAČNÍHO MODULU DO I/O KONFIGURACE 87
OBR. A.12 VOLBA HLAVNÍ REVIZE PRO ENBT MODUL 87
OBR. A.13 NASTAVENÍ PARAMETRŮ MODULU 1756-ENBT/A 88
OBR. A.14 NASTAVENÍ RPI PRO PRODUCENTA 88
OBR. A.15 VYTVOŘENÍ PRODUKOVANÉ PROMĚNNÉ 89
OBR. A.16 NASTAVENÍ CESTY PROJEKTU 90
OBR. A.17 ZJIŠTĚNÍ HARDWAROVÉ KONFIGURACE 91
OBR. A.18 ZALOŽENÍ NOVÉHO PROJEKTU 91
OBR. A.19 OVĚŘENÍ SPUŠTĚNÍ RSLOGIX V ODPOVÍDAJÍCÍ VERZI 92
OBR. A.20 PŘIDÁNÍ KOMUNIKAČNÍHO MODULU DI I/O KONFIGURACE 92
OBR. A.21 NASTAVENÍ HLAVNÍ REVIZE 93
OBR. A.22 NASTAVENÍ PARAMETRŮ PŘIDÁVANÉHO MODULU 93
OBR. A.23 PŘIDÁNÍ VZDÁLENÉHO KOMUNIKAČNÍHO MODULU 94
OBR. A.24 VOLBA VZDÁLENÉHO KOMUNIKAČNÍHO MODULU 94
OBR. A.25 VOLBA HLAVNI REVIZE 94
OBR. A.26 NASTAVENÍ PARAMETRŮ KOMUNIKAČNÍHO MODULU 95
OBR. A.27 ZPŘÍSTUPNĚNÍ PROCESORU, ZE KTERÉHO BUDEME ČÍST PRODUKOVANOU PROMĚNNOU 95
OBR. A.28 ZVOLENÍ ODPOVÍDAJÍCÍHO TYPU PROCESORU 96
OBR. A.29 VOLBA HLAVNÍ REVIZE PŘIDÁVANÉHO PROCESORU 96
OBR. A.30 NASTAVENÍ PARAMETRŮ PŘIDÁVANÉHO PROCESORU 97
OBR. A.31 VYTVOŘENÍ NOVÉ PROMĚNNÉ TYPU KONZMENT 97
OBR. A.32 NAPOJENÍ VYTVOŘENÉHO TAHU NA VZDÁLENÝ PRODUKOVANÝ TAG 98
OBR. A.33 NASTAVENÍ CESTY PROJEKTU 98
OBR. A.34 ZALOŽENÍ NOVÉHO PROJEKTU 99
OBR. A.35 PŘIDANÍ DIGITÁLNÍCH VSTUPŮ DO I/O KONFIGURACE 100
OBR. A.36 I/O KONFIGURACE PO PŘIDÁNÍ DIGITÁLNÍCH VSTUPŮ 100
OBR. A.37 PŘIDÁNÍ DALŠÍCH DIGITÁLNÍCH VSTUPŮ 100
OBR. A.38 I/O KONFIGURACE 101
OBR. A.39 PŘIDÁNÍ DIGITÁLNÍCH VÝSTUPŮ DO I/O KONFIGURACE 101
OBR. A.40 I/O KONFIGURACE 101
OBR. A.41 NASTAVENÍ CESTY 102
OBR. A.42 VYTVOŘENÍ NOVÉHO PROJEKTU 102
OBR. A.43 I/O KONFIGURACE 103
OBR. A.44 PŘIDÁNÍ KOMUNIKAČNÍHO MODULU 103
OBR. A.45 I/O KONFIGURACE PO PŘIDÁNÍ KOMUNIKAČNÍHO MODULU 103
OBR. A.46 PŘIDÁNÍ KOMUNIKAČNÍHO MODULU 104
OBR. A.47 I/O KONFIGURACE 104
OBR. A.48 PŘIDÁNÍ DIGITÁLNÍCH VSTUPŮ 105
OBR. A.49 I/O KONFIGURACE PO PŘIDÁNÍ DIGITÁLNÍCH VSTUPŮ 105
OBR. A.50 PŘIDÁNÍ DALŠÍCH DIGITÁLNÍCH VSTUPŮ 105
OBR. A.51 I/O KONFIGURACE 106
OBR. A.52 PŘIDÁNÍ DIGITÁLNÍCH VÝSTUPŮ 106
OBR. A.53 KOMPLETNÍ I/O KONFIGURACE 107
OBR. A.54 ZADÁNÍ CESTY PRO NAHRÁNÍ PROJEKTU 107
OBR. A.55 ZADÁNÍ CESTY PRO NAHRÁNÍ PROJEKTU 108
OBR. B.1 VZHLED STRÁNEK LABORATOŘE 109
OBR. C.1 NÁSTROJ NA TVOŘENÍ WEBOVÝCH STRÁNEK 110
OBR. C.2 NÁSTROJ NA ÚPRAVU MENU 110
OBR. D.1 PŘEDNÍ PANEL PŘEPÍNAČE 111
OBR. D.2 EXPRESS SETUP PRO ZÁKLADNÍ NASTAVENÍ PŘEPÍNAČE 112

1 Úvod

Tato práce je v návaznosti na zadání rozdělena do několika částí. První část je věnována popisu komunikačních linek architektury NetLinx společnosti Rockwell Automation a Ethernetu, který se používá pro přenos informací zapouzdřených EtherNet/IP protokolem. Pro zajištění vzájemné spolupráce jednotlivých linek se používá CIP protokol, jehož příkazy a data jsou zapouzdřovány do paketů vhodných pro přenos prostřednictvím sítí ControlNet, EtherNet/IP a DeviceNet, tedy linek architektury NetLinx.

Další část se zabývá srovnáním komunikačních linek Rockwell Automation a linek dalších předních světových výrobců zaměřených na popis real-time komunikace přes Ethernet.

Poslední část diplomové práce se pak věnuje popisu webových stránek laboratoře Allen-Bradley, které jsem v rámci diplomové práce vytvořil. Pro prezentaci Laboratoře Allen-Bradley K23 jsem vytvořil webové stránky, jejichž obsah je věnován zejména popisu průmyslových automatů Allen-Bradley a modelů. Na těchto stránkách je umístěna vytvořená Flash aplikace webového řízení modelu dvouosého soustruhu a vytvořené Flash aplikace pro měření a zobrazování zatížení sítě v laboratoři.

Za účelem určení způsobu komunikace a určení zatížení při komunikaci pomocí EtherNet/IP protokolu mezi moduly Allen-Bradley byl zakoupen přepínač s managementem Cisco Catalyst 2960. Pomocí tohoto přepínače byla následně v laboratoři realizována podsíť, jejímiž účastníky jsou průmyslové automaty. Přepínač byl nakonfigurován tak, aby byla zaručena bezproblémová a optimalizovaná funkce sítě při komunikaci mezi automaty.

Experimentálně odposlechem na síti byly stanoveny rovnice pro predikování zatížení sítě při vzájemné komunikaci automatů. Z experimentů rovněž byly vytvořeny určitá doporučení pro vlastní realizaci sítě.

2 Komunikační linky Rockwell Automation

2.1 Úvod

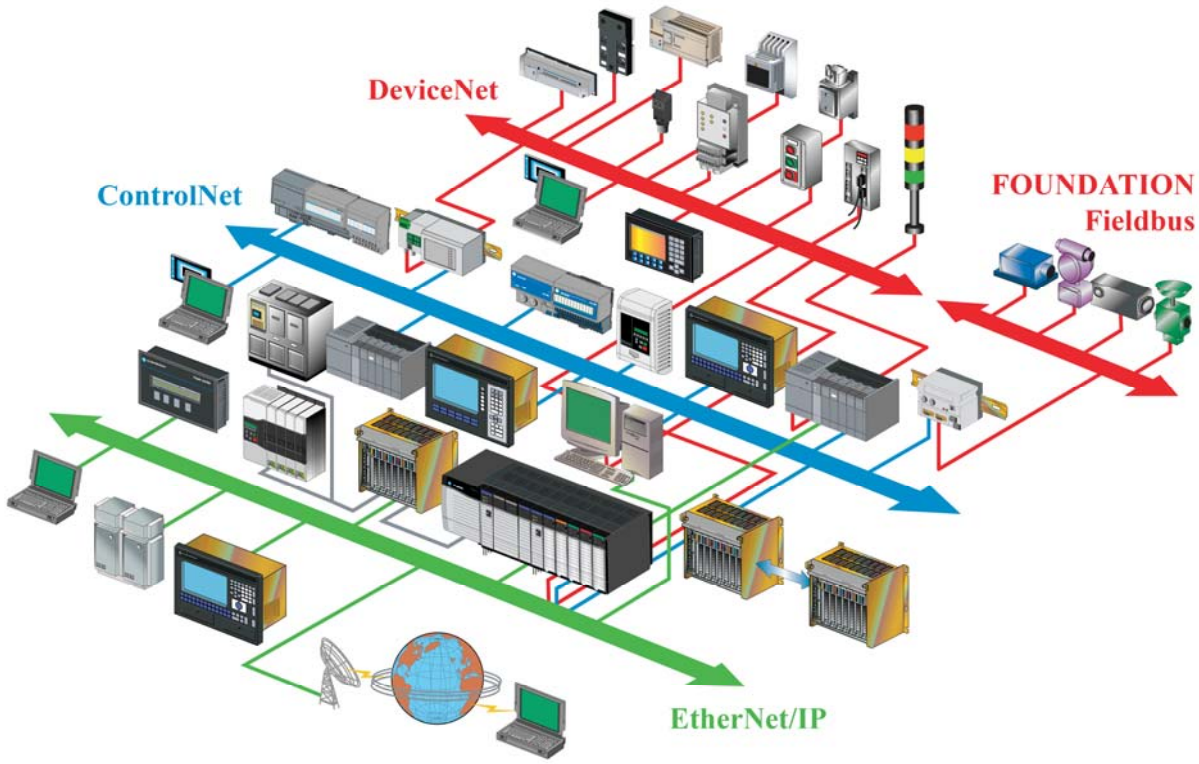
Firma Rockwell Automation vyvinula síťovou architekturu označovanou jako NetLinx Architecture. NetLinx zahrnuje průmyslové sítě EtherNet/IP, ControlNet, DeviceNet a podporuje FOUNDATION Fieldbus.

ControlNet, vyvinutý společností Rockwell Automation, rozšiřuje funkce starších sítí DH+/RIO a DH485 společnosti Allen-Bradley. Zajišťuje deterministický, opakovatelný a uživatelem plánovaný přenos dat. Nyní je vlastněn a spravován organizací CI (ControlNet International).

Síť DeviceNet byla vyvinuta rovněž společností Rockwell Automation a nahrazuje síť Remote I/O společnosti Allen-Bradley. Nyní je vlastněna a spravována organizací ODVA (Open DeviceNet Vendor Association). Používá se pro řízení na nejnižší úrovni (akční členy, senzory).

Síť EtherNet/IP je otevřený standard, určený pro konfiguraci, řízení a sběr dat . Tento standard představily organizace ODVA, CI a jeho použití je bezplatné.

NetLinx začleněním a spoluprací s technologiemi OLE (Object Linking and Embedding), OPC (OLE for Process Conrol), RSLinx a IOLinx, umožňuje snadné používání s komerčně běžně využívanými softwarovými a hardwarovými součástmi [1].



Obr. 2.1 Architektura NetLinx, kompletní řešení průmyslové sítě

2.2 OSI model (Open System Interconect)

Je standart, který popisuje síťové komunikace a umožňuje komunikaci mezi odlišnými síťovými zařízeními, které však dodržují protokoly (vrstvy) obsažené v modelu OSI modelu (Obr. 2.2). Každá vrstva komunikuje pouze s bezprostředně sousedními vrstvami definovanými v OSI modelu [3].

Aplikační vrstva

Je vrstva přístupná uživateli a obsahuje data, která uživatel odesílá na jiné zařízení. Aplikace pracující na úrovni této vrstvy převede uživatelská data do vhodné podoby a přidá hlavičku aplikační vrstvy. Ta obsahuje informace identifikující vysílající a přijímající zařízení.

Prezentační vrstva

Tato vrstva provádí převod dat z aplikační vrstvy do formy vhodné pro přenos cílovému zařízení (překódování, komprese atd.). Po převodu dat z aplikační vrstvy se přidá hlavička prezentační vrstvy identifikující provedené operace.

Relační vrstva

Zajišťuje vlastní spojení mezi síťovými stanicemi, ošetření chyb při komunikaci (např. přerušení komunikace) a určení způsobu komunikace. Komunikace může probíhat současně oběma směry (full-duplex) nebo střídavě v jednom směru (half-duplex - stanice se střídají ve vysílání dat). Podrobnosti nastavení jsou uloženy opět v hlavičce, která se přidá k předávaným datům.

Transportní vrstva

Provádí ochranu přenášených dat a rozděluje data do menších částí tzv. paketů. Provádí výpočty kontrolních součtů, kterými se ověřuje správnost rozložení, přenesení a opětovné složení dat. Pro případ neúspěšného přenosu dat se vytváří kopie těchto dat, které se uchovávají dokud není přenos úspěšně ověřený správností kontrolního součtu. Opět se přikládá hlavička vrstvy

Síťová vrstva

Zajišťuje výběr optimální cesty přenášených dat k cílové stanici. Jestliže naše síť neobsahuje podsítě a není dále strukturovaná, tato vrstva se může vynechat. Pro případ, že vrstva není vynechána, přikládá se opět hlavička vrstvy identifikující volbu cesty.

Linková vrstva

Tato vrstva zajišťuje vlastní přenos paketů, ze kterých vytváří rámce (frames). Rámce jsou doplněny informacemi potřebnými pro synchronizaci hodin síťových adaptérů spolupracujících stanic. Rovněž se provádí výpočet kontrolních součtů paketů. Pro případ neodpovídajícího kontrolního součtu, se uchovávají kopie paketů, které lze opětovně odeslat.

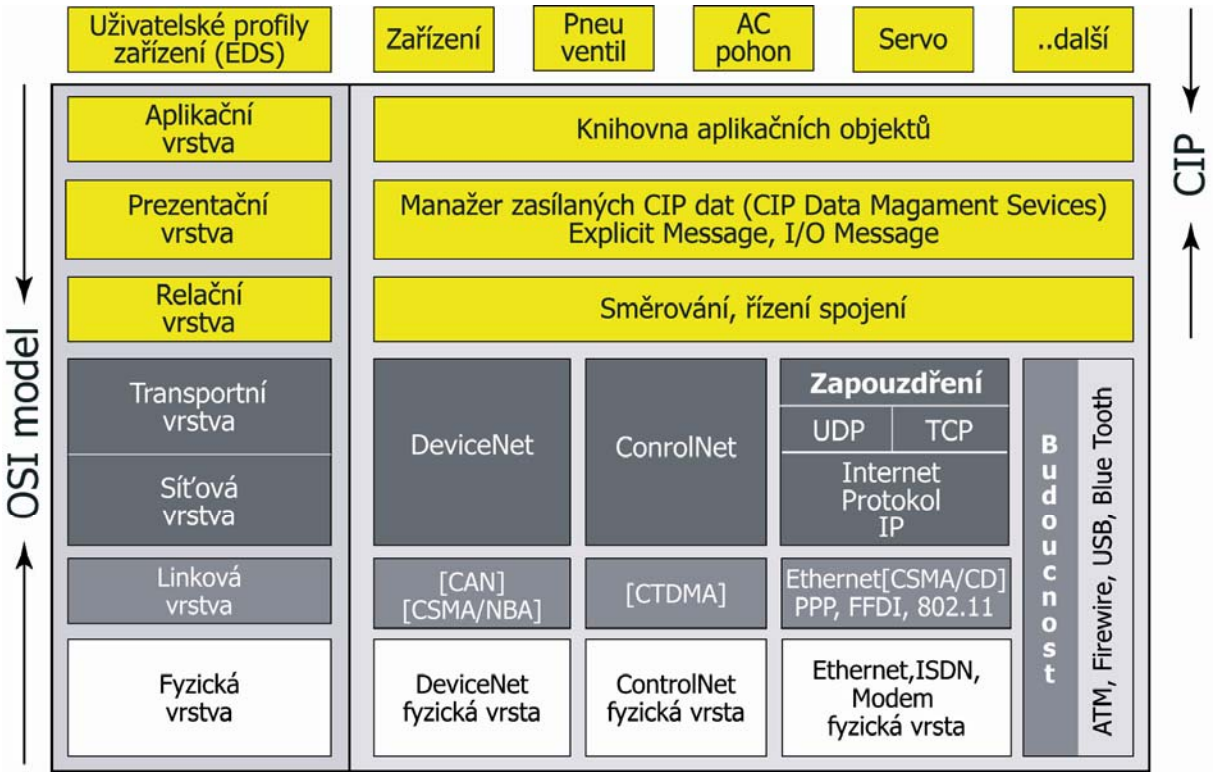
Fyzická vrstva

Zajišťuje zakódování rámců do podoby elektrických signálu a jejich následný přenos po médiu.

Pro obsloužení linkové a fyzické vrstvy je nutné odpovídají hardwarové síťové vybavení. Pro obsloužení vrstev vyšších (síťová až aplikační) bývá obvykle nutné pouze softwarové vybavení.

2.3 CIP (Common Industrial Protocol)

CIP je objektově orientovaný protokol, který zajišťuje spojení mezi průmyslovými zařízeními prostřednictvím vhodných komunikačních protokolů. Dříve se pro CIP používalo označení Control and Information Protocol, ale v současné době se častěji používá označení Common Industrial Protocol.



Obr. 2.2 Spojitost CIP s OSI modelem [1, 2, 3, 4, 5]

CIP umožňuje jednotný přístup a snadnou spolupráci resp. komunikaci mezi průmyslovými částmi nezávislou na použitém typu média. CIP obsahuje aplikační, prezentační a relační vrstvu OSI modelu, které obsahují společné komunikační principy využívané síťovou architekturou NetLinx (Obr. 2.1).

2.3.1 Definice pojmů

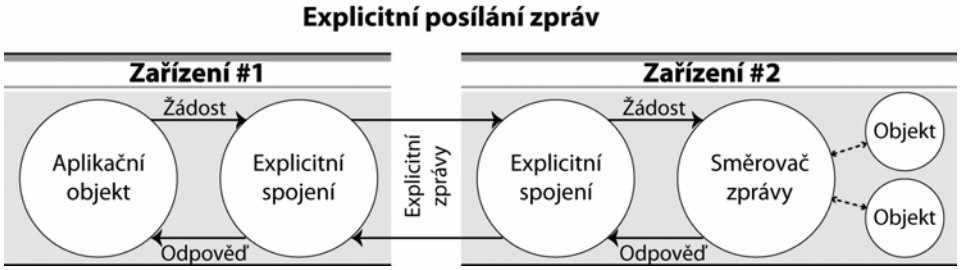
Klient je zařízení, které odesílá požadavek serveru.

Server je zařízení, které přijímá žádosti od klienta. Od serveru se očekává odpovědět klientovi.

Producent je zařízení, které umisťuje data na síť jako zprávu pro využití jedním nebo více konzumenty.

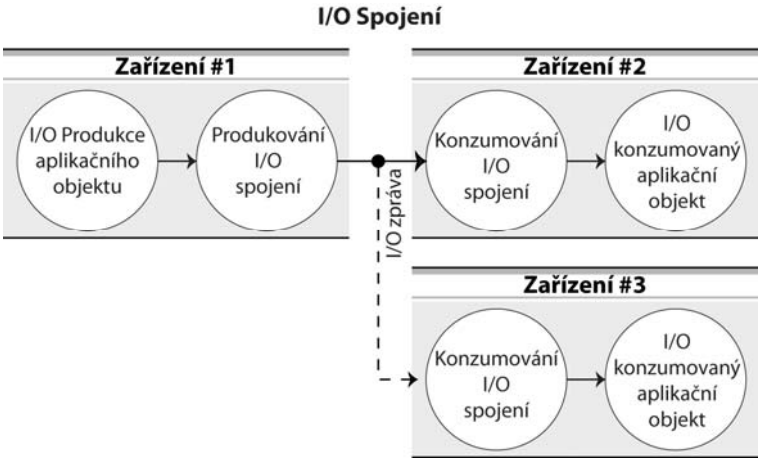
Konzument je jedno z mnoha zařízení, které přijímají zprávu umístěnou na síť producentem.

Manažer zpráv bez vytvořeného spojení (UnConnected Message Manager, UCMM) se používá pro vytvoření spojení mezi zařízeními. Zpracovává požadavky na spojení.
Manažer zpráv s vytvořeným spojením (Connection Manager) po vytvoření spojení mezi zařízeními pomocí UCMM, přebírá řízení komunikace.
Explicitní zpráva (Explicit Message) je komunikace typu požadavek/odezva. Tato zpráva je vždy uskutečněna pomocí směrovače zpráv (Message Router). Požadavek obsahuje informaci, která vyzve přijímající zařízení k vykonání určité akce. Po vykonání požadované akce, zařízení zasílá odpověď (odezvu). Pro přímé spojení explicitní zprávou (Direct Connection) mezi dvěma uzly je nutná zdrojová adresa, cílová adresa a identifikátor spojení (Connection ID) pro každý směr komunikace. Cílová adresa a zdrojová adresa je pro ControlNet a DeviceNet číslo uzlu (Node Address), pro EtherNet/IP je to IP adresa.



Obr. 2.3 CIP explicitní posílání zpráv

Implicitní zpráva (I/O Implicit Message) je zpráva, která nenese adresu a číslo služby (Service Information). Konzument již ví, co má dělat s daty podle identifikátoru spojení (Connection ID), které bylo přiřazeno při navazování spojení pomocí UCMM.

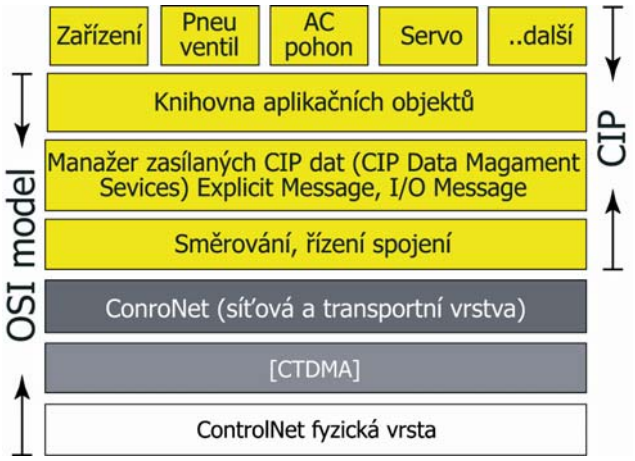


Obr. 2.4 CIP multicastová komunikace I/O spojení (typ producent/konzument)

2.3.2 ControlNet

ControlNet je deterministická digitální síť, která poskytuje vysokorychlostní přenos časově kritických I/O dat (implicitní zprávy). Rovněž umožňuje nahrávání a stahování (Download/Upload) řídicího programu do řídicí jednotky (Controller) automatu pomocí zasílání CIP zpráv (UCCM CIP Multiple Message). Podporuje rovněž typ komunikace producent/konzument a priorizaci zpráv. Deterministický a plánovaný přístup na médium

(prostý od kolizí) síťovým zařízením poskytuje plánované vysílání zpráv [6]. Pro přístup na médium se používá metoda CTDMA (Concurrent Time Domain Multiple Access – současný vícenásobný přístup v definovaném intervalu). Základní periodický interval, ve kterém mohou být data přenášena z jednotlivých zařízení definuje uživatel jako NUT (Network Update Time – čas obnovy sítě). Jednotlivá zařízení jsou na síti reprezentovány jako uzly sítě (Node). NUT se volí v intervalu 2 až 100 milisekund. Doporučovaná hodnota NUT je 5 milisekund.



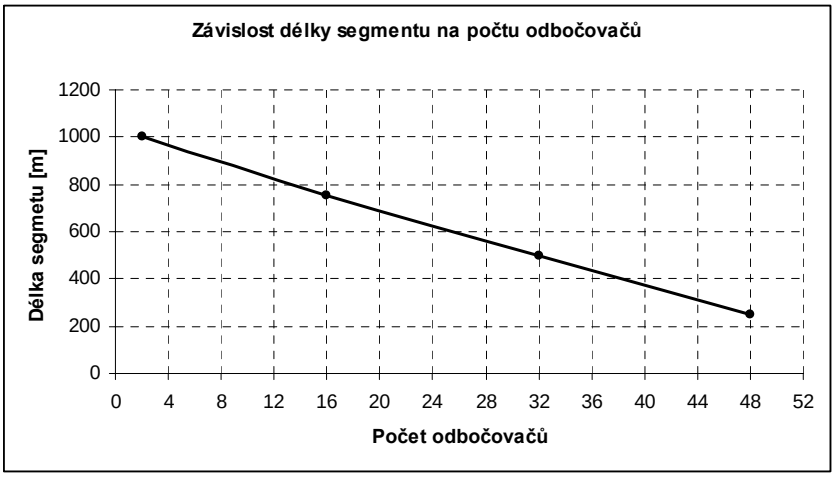
Obr. 2.5 Spojitost OSI modelu a sítě ControlNet

2.3.2.1 Fyzická vrstva

Přenosovým médiem je 75 Ω koaxiální páteřní kabel (typicky RG-6, možné je i použití optických vláken a stíněné kroucené dvoulinky STP) zakončený na obou koncích ukončovacími 75 Ω odpory. Zařízení se připojují pomocí speciálních odbočovačů (Taps) se specifickou délkou 1m (Dropline). Není definováno žádné omezení na minimální vzdálenost mezi odbočovači. Zařazení odbočovače však snižuje maximální použitelnou délku páteřního kabelu (Trunkline).

Výpočet maximální délky segmentu:

$$max_délka_segmetnu = 1000m - 16,3m \times (počet_odbočovačů - 2) \quad \textbf{(2.1)}$$

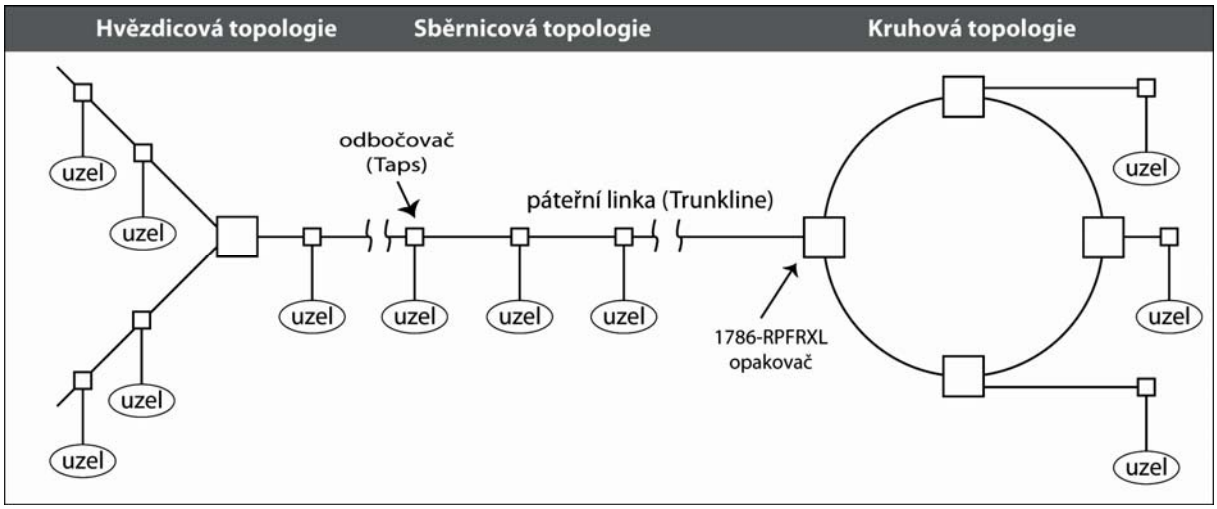


Obr. 2.6 Závislost délky segmentu na počtu odbočovačů

Pokud potřebujeme překročit maximální délku segmentu, musíme použít opakovač (Repeater). Ten umožní zvýšení délky segmentu bez snížení rychlosti komunikace. Dříve bylo možné mít mezi dvěma uzly maximálně 5 opakovačů, v současné době je to až 20 opakovačů. Maximální délka fyzické vrstvy s použitím 20 opakovačů je 20 kilometrů.

2.3.2.2 Topologie

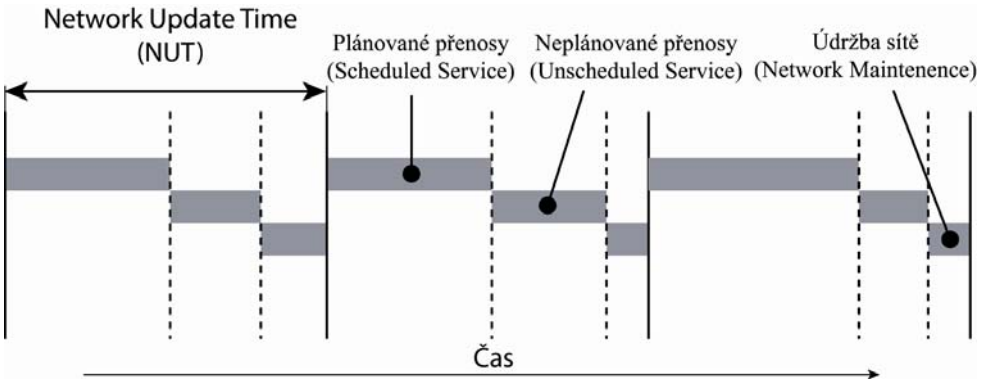
Jsou možné různé typy topologie zapojení sítě ControlNet. Například můžeme zařízení zapojit do stromové, hvězdicové či sběrnicové topologie. Speciálním typem je topologie kruhová, pro jejíž zapojení, však musíme použít speciální druh opakovačů (1786-RPFRXL).



Obr. 2.7 Příklady topologie zapojení sítě ControlNet

2.3.2.3 Přístup na médium CTDMA

Na médium se vysílá v uživatelem definovaných intervalech NUT (Network Update Time – čas obnovy sítě). Tento interval je rozdělen do tří částí. První část intervalu je vyhrazena pro plánované přenosy (Schedule Service). Neplánované přenosy (Unscheduled Service) a informace nutné pro údržbu sítě (Network Maintenence, někdy označované jako Guardband Time) zaujímají zbývající část.



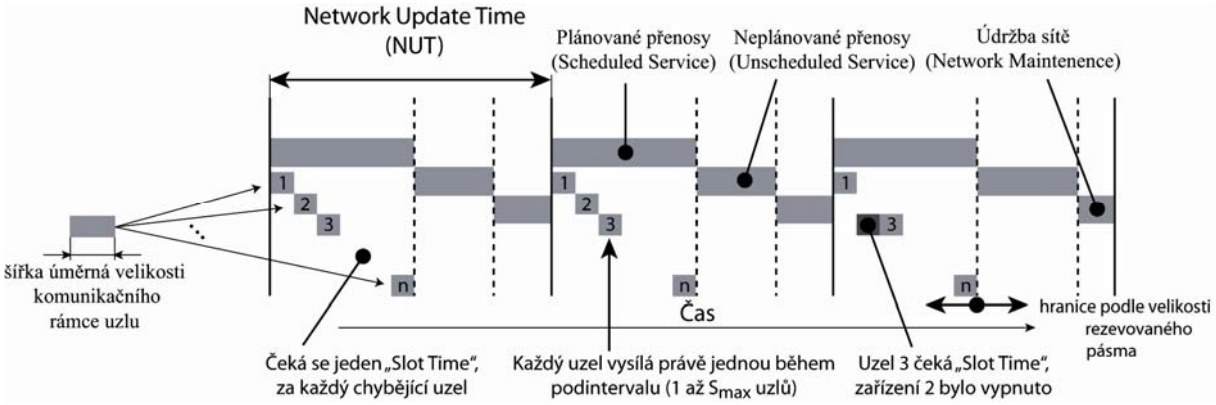
Obr. 2.8 Přístup na médium CTDMA

Plánované přenosy

Každý uzel od prvního až do maximálního plánovaného (S_{max}) má v této části NUT vyhrazený prostor pro odeslání zprávy. Jestliže některý z uzlů nemá data, která by odeslal,

vloží do své časti krátkou zprávu. Tím ostatním zařízením sdělí, že přesto, že nemá data k poslání, je stále „živý“. V případě, že se zařízení nepodaří vložit ani rámec (Frame) s daty, ani rámec s indikací, že je stále naživu, první zařízení s vyšším číslem uzlu čeká předem definovanou dobu (Slot Time), než vloží vlastní zprávu.

Velikost podintervalu pro plánované přenosy v intervalu NUT je závislá na plánovaném počtu uzlů. Tento podinterval nemůže nikdy překročit velikost NUT. Může však zabrat celé pásmo pro neplánované přenosy a část pásma pro údržbu sítě.

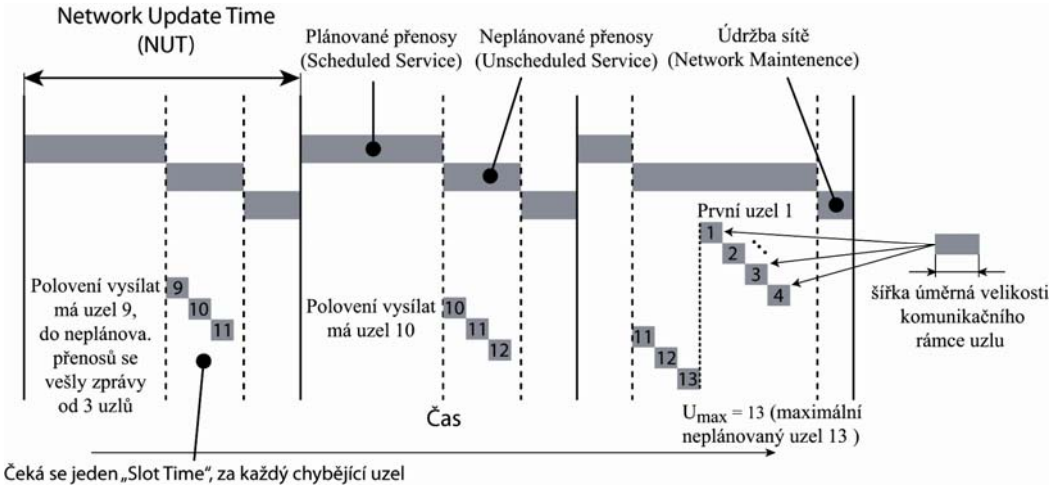


Obr. 2.9 Plánovaný přenos

Neplánované přenosy

Jsou určeny k přenášení časově nekritických zpráv a pouze jednomu uzlu je garantován přístup na médium během intervalu NUT. Pokud zpráva není natolik velká, aby zabrala celé pásmo neplánovaného přenosu, je povolen přístup dalšímu uzlu s vyšším pořadovým číslem.

Pokud dojde k selhání během přenosu, nebo se z jakýchkoliv příčin uzlu, který má garantovaný přístup na médiu, nepodaří provést odeslání zprávy, pokračuje v zápisu uzel s vyšším pořadovým číslem, přičemž čeká „Slot Time“. Číslo uzlu, který má garantovaný přístup na médium, je po každém proběhnutí NUT navýšeno o jedna. Tím se zajišťuje postupný přístup všech uzlů na médium a odhadnutelnost maximální doby přístupu.

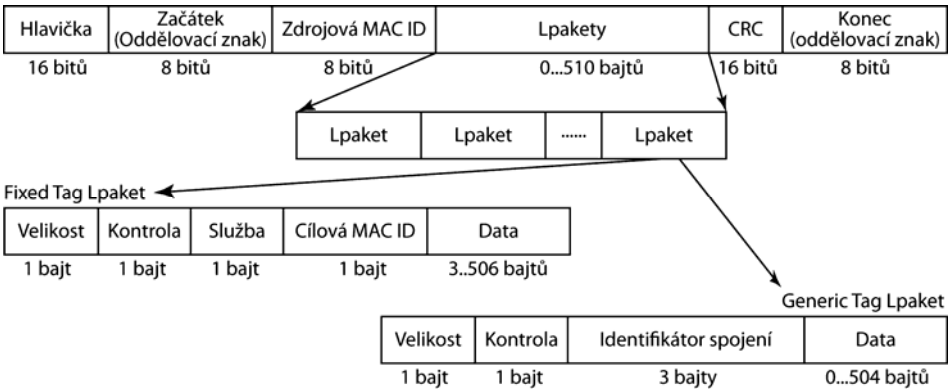


Obr. 2.10 Neplánovaný přenos

2.3.2.4 Popis komunikačního rámce

V každém MAC (Media Access Control) rámci je 510 bajtů dostupných pro přenášená data a zprávy. Těchto 510 bajtů může být složeno z několika Lpaketů (Link Pakets). Lpakety dělíme do dvou skupin, na pakety určené pro přenos nespojovaných zpráv a administraci sítě (Fixed Tag Lpackets) a pakety určené pro přenos implicitních a explicitních zpráv spojovaným typem komunikace (Generic Tag Lpackets). Speciální typ Lpaketu je určen pro údržbu sítě.

Všechny uzly zapojené k médiu při jakémkoli typu komunikace poslouchají přenášené MAC rámce. To je výhodné zejména při typu komunikace producent/konzument.



Obr. 2.11 Formát komunikačního rámce ControlNet

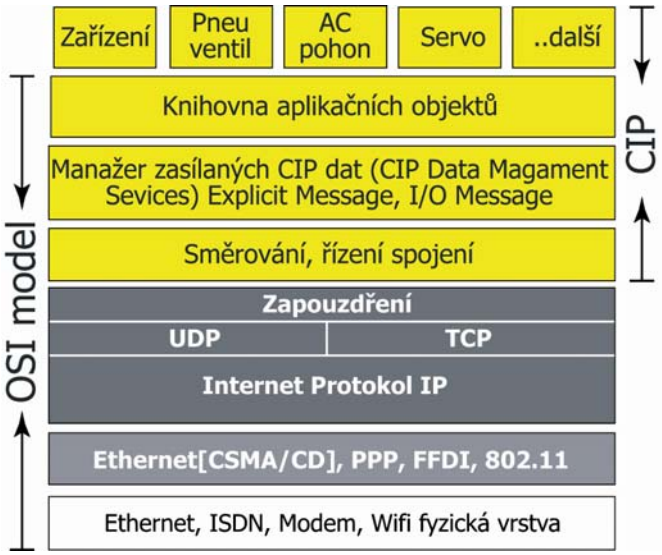
2.3.3 EtherNet/IP

Ethernet/IP je technologie podporovaná jak sdružením ODVA (Open DeviceNet Vendor Association), tak sdružením CI (ControlNet International). Je to nejnovější člen rodiny CIP protokolů. Síť EtherNet/IP rozšiřuje Ethernet pro použití v průmyslu. EtherNet/IP může být použit na jakýchkoliv sítích podporujících transportní vrstvu TCP/UDP a lze jej součastně použít i s dalšími CIP protokoly.

Význam slova EtherNet/IP bývá vykládán buď jako Ethernet Industrial Protocol, nebo jako Ethernet + CIP. Tyto názvy jsou trochu zavádějící, protože jako linková vrstva OSI modelu nemusí být použit standart Ethernet, ale může být použit libovolný standart, který spolupracuje s transportní vrstvou TCP/UDP.

TCP/IP je používán pro zapouzdřování (Encapsulation) explicitního spojení, zatímco UDP/IP je používán pro zapouzdřování implicitního spojení.

EtherNet/IP byl původně určen pro sběr dat o výrobním procesu a následné poskytnutí těchto dat manažerským systémům. Původně nebyl vhodným kandidátem pro řízení výrobního procesu, který vyžaduje pro své řízení deterministickou síť. Avšak použitím krátkých intervalů přenosu dat, plně duplexních linek, 100 Mbitového Ehternetu a rozšířením standartu CIP např. o CIP Sync (umožňuje priorizaci zpráv) a CIP Motion [6], je nyní EtherNet/IP využitelný i jako deterministická síť vhodná pro real-time řízení výrobního procesu.



Obr. 2.12 Spojitost OSI modelu a Ethernet/IP

2.3.3.1 Fyzická vrstva

Protože EtherNet/IP je určen pro nasazení v průmyslu, jsou na jeho fyzickou vrstvu kladeny vyšší nároky. Z toho vyplývají určitá omezení, které budou popsána níže.

V současné době jsou ve specifikaci EtherNet/IP [7] a specifikaci CIP [6] popsána omezení pro využití fyzické vrstvy Ethernetu v průmyslu. Pro jiný typ fyzických vrstev, zatím ve specifikacích, žádná omezení určena nejsou.

Jsou stanoveny dvě úrovně omezení. Omezení s nižšími nároky je označováno jako The COTS EtherNet/IP level a omezení s vyššími nároky jako Industrial EtherNet/IP level:

The COTS (Commercial Off The Shelf) EtherNet/IP Level – přechod mezi kancelářskou a tovární úrovní

- konektory RJ-45 s omezenou délkou kabelu na 100m
- na kabeláž je kladena podmínka splnění standardů definovaných IEEE, ANSI/TIA/EIA
- splnění ochrany IP20 [8]

The Industrial EtherNet/IP – průmyslová úroveň

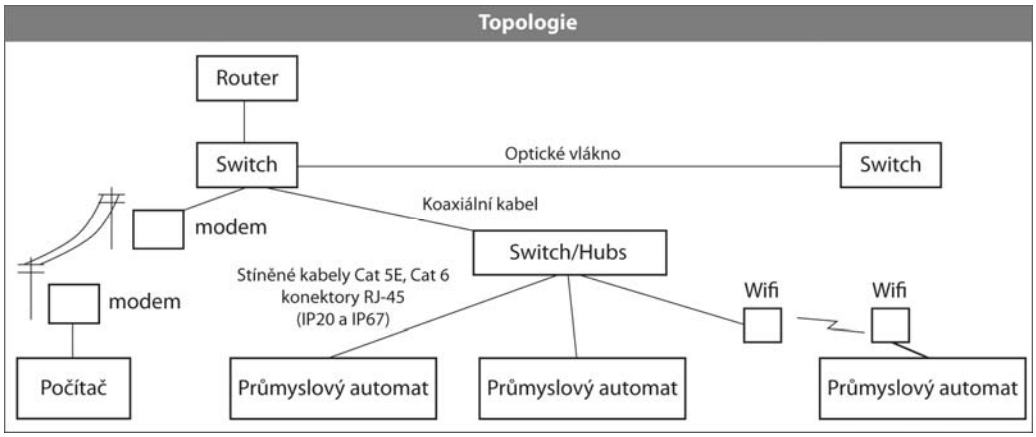
- vyžaduje minimálně splnění požadavků na konektory a kabeláž kladené standarty IEC, ANSI/TIA/EIA
- doporučené konektory jsou vodě odolný RJ-45, D-Coded M12-4
- splnění ochrany až IP67 [8]

Pro spojování jsou doporučeny stíněné a nestíněné kabely Cat 5E a Cat 6. Použití stíněných kabelů je doporučováno tam, kde hrozí ovlivnění okolním prostředím (napájecí kabely, napájecí zdroje atd.). Délka těchto kabelů, by neměla být delší jak 100 metrů. Optické

vlákna jsou doporučovány pro přenos na velké vzdálenosti a tam, kde jsou silná elektromagnetická pole nebo vysoký napěťový potenciál mezi dvěma zařízeními.

2.3.3.2 Topologie

Příklad způsobu zapojení zařízení komunikujících prostřednictvím EtherNet/IP viz Obr. 2.13.



Obr. 2.13 Příklad možné topologie EtherNet/IP

2.3.3.3 Popis komunikačního rámce

EherNet/IP používá standardní Ethernet TCP/IP a UDP/IP rámce definované mezinárodním standartem [9, 10, 11, 12, 13, 14] bez jakýchkoliv změn. V této časti se zaměřím na popis EtherNet/IP zapouzdřování (Encapsulation), vytváření části rámce specifické pro EtherNet/IP a popis dat, které může rámec obsahovat.

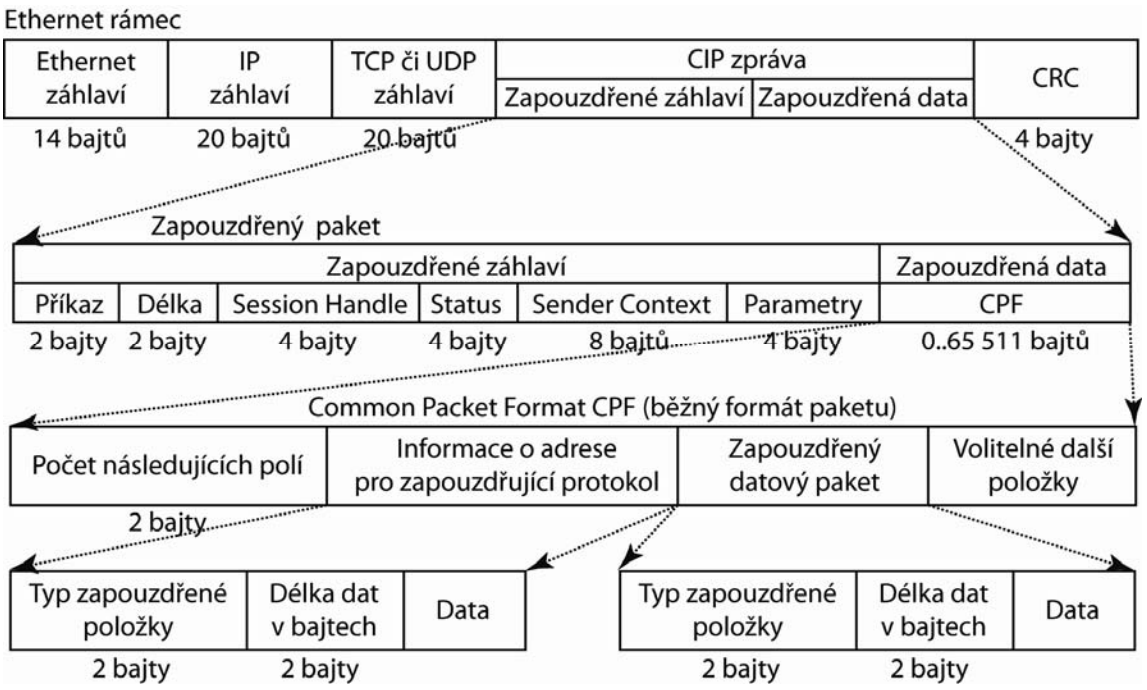
2.3.3.3.1 EtherNet/IP zapouzdřování (Encapsulation)

Zapouzdřovací protokol definuje dva rezervované porty 0xAF12 a 0x08AE. Všechna zařízení podporující EtherNet/IP přijmou nejméně dvě TCP spojení na TCP portu 0xAF12 (44818). Tento TCP port je používán pro zasílání explicitních zpráv, ať už se jedná o spojovanou či nespojovanou komunikaci a port se rovněž používá pro nastavení komunikace mezi dvěma uzly. Některé příkazy mohou být rovněž odeslány přes tento port pomocí UDP datagramů.

Port 0x08AE (2222) používají zařízení pro implicitní I/O komunikaci. Tento typ komunikace probíhá přes UDP datagramy, a to z důvodů, aby se využilo výhod multicastového vysílání.

Na rozdíl od TCP, UDP nemá definovanou možnost přeuspořádání paketů. Z toho vyplývá, že v jednom UDP paketu je zapouzdřena pouze jedna zpráva. U TCP tomu tak být nemusí, neboť zpráva reps. několik zpráv může být rozděleno do několika TCP paketů.

CIP zpráva se skládá ze záhlaví a dat jejichž složení je patrno z Obr. 2.14, který zachycuje způsob zapouzdřovaní CIP zprávy do Ethernet rámce. Datová část CIP zprávy dodržuje formát CPF (Common Packet Format), který definuje složení této části.



Obr. 2.14 Ethernet rámec, do kterého se zapouzdřuje CIP zpráva

CPF, pokud to lze provést pro daný typ zprávy, umožňuje zapouzdření více položek do jednoho rámce. Jako typický příklad odeslání vícenásobné položky, lze uvést nahrávání programu do řídicí jednotky. Data jsou do řídicí jednotky (Controller) odeslána jako UCCM CIP Multiple Message. To je explicitní typ zprávy, kdy je v jednom rámci, nespojovaným typem komunikace odeslána zapouzdřená násobná zpráva.

2.3.3.3.1.1 Zapouzdřené příkazy, příklady a jejich význam

V této části jsou vysvětleny nejdůležitější příkazy, které mohou být obsaženy v záhlaví CIP zprávy (viz Obr. 2.14). Příklad užití některých z těchto příkazů je ukázán v Obr. 5.3.

No Operation (NOP)

Používá se ke zjištění zda je TCP spojení stále otevřené. Jakákoliv zapouzdřená data, která tento rámec obsahuje, jsou ignorována a přijímající zařízení na tuto zprávu negeneruje žádnou odpověď.

ListIdentity (seznam identit – 0x0063)

Používá se k nalezení a identifikování potenciálních EtherNet/IP zařízení. Tento příkaz bývá odeslán jako UDP broadcast rámec (obdrží jej všechna zařízení s odpovídající maskou podsítě). Typicky jej používají softwarové nástroje k získání informací o přítomných zařízeních na síti.

ListInterface (seznam rozhraní – 0x0064)

Používá se k identifikaci možného non-CIP komunikačního rozhraní asociovaného s cílem. Pro odeslání se využívá UDP rámec. Zařízení přijímající tento příkaz generuje

odpověď obsahující minimálně informaci Interface Handle (identifikátor rozhraní, 32 bitů) (Obr. 2.15).

RegisterSession/UnRegisterSession (registrování 0x0065 / odregistrování relace 0x0066)

Tyto příkazy se používají k otevření či uzavření relace mezi dvěma zařízeními. Jakmile proběhne registrace relací, může proběhnout výměna zpráv. Lze vytvořit i více relací mezi dvěma zařízeními, ale není to obvyklé. Zařízení odesílající příkaz RegisterSession vytváří hodnotu Sender Context (viz Obr. 2.14) a zařízení přijímající tento požadavek vytváří Session Handle. Tyto dvě hodnoty jsou následně používány k identifikaci zpráv mezi těmito dvěma zařízeními.

SendRRData/SendUnitData (0x006F / 0x0070)

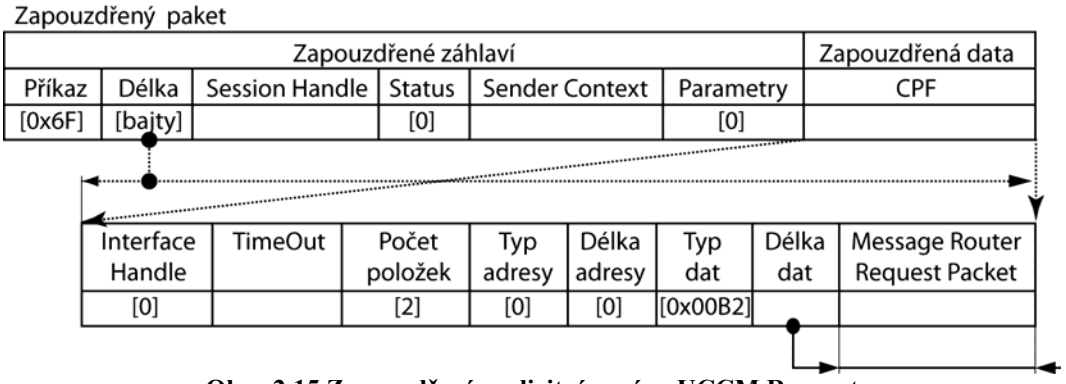
SendRRData se používá pro odeslání zpráv nespojovaným typem komunikace UCCM a přenáší zapouzdřený paket požadavku/odpovědi. Požadavek a odpověď jsou zapouzdřeny do datové části zprávy. SendUnitData je používán pro spojované explicitní zprávy a používá se, pokud zapouzdřující protokol má vlastní transportní mechanismus [2].

2.3.3.3.1.2 Zapouzdřená data odeslaná jako explicitní zpráva

Na EtherNetu/IP může být explicitní zpráva odeslána jako spojovaná či nespojovaná (Connected nebo UnConnected). Spojovaný typ explicitních zpráv vyžaduje nejdříve vytvoření spojením, což znamená, že se rezervují prostředky potřebné pro vytvářené spojení a tyto prostředky jsou rezervovány až do ukončení spojení. Tento typ spojení je vlivem rezervovaných prostředků schopen včas odpovídat na přijaté dotazy a je proto vhodný pro periodické explicitní dotazy.

Většina explicitních zpráv může být rovněž zaslána jako nespojovaná, což lze použít v případech, kdy nepožadujeme včasné odpovědi v pravidelných intervalech. Při tomto typu spojení nejsou totiž rezervované prostředky, které zajistí včasnou odpověď.

Explicitní zprávy jsou na EtherNet/IP odesílány s TCP/IP záhlavím a používá se SendUnitData Encapsulation Command (nespojovaný typ) a SendRRData Encapsulation Command (spojovaný typ).



Obr. 2.15 Zapouzdření explicitní zprávy UCCM Request
(Message Router Request – dotaz na objekty viz Obr. 2.3)

2.3.3.3.1.3 Zapouzdřená data odeslaná jako implicitní zpráva (I/O Message)

Implicitní I/O zpráva se odesílá s UDP/IP záhlavím. Není sice vyžadováno zapouzdřovací záhlaví, ale přesto se komunikace drží CPF formátu.

Poč. položek	Typ adresy	Délka adresy	Identifikátor spojení	Pořadové číslo	Typ dat	Délka dat	Data
[2]	[0x8002]	[8]			[0x00B1]		

16 bit. pořad. číslo paketu	Real-Time záhlaví	I/O data

Obr. 2.16 Zapouzdření implicitní zprávy I/O

Datové pole obsahuje prefix s 16 bitovým pořadovým číslem paketu., který slouží k identifikaci dat. Real-time záhlaví slouží k identifikaci stavu Run/Idle (běh/pauza), pro identifikaci tohoto stavu se ale rovněž používají speciální datové pakety (Run/Idle). Metoda se volí podle nastaveného parametru v EDS (Electronic Data Sheet) souboru v části Connection Manager.

Zdroj I/O zpráv přenáší data k cíli pomocí multicast (určen skupině uživatelů) UDP rámců. Cíle odpovídají zdroji pomocí unicast (určen jednomu uživateli) UDP rámců.

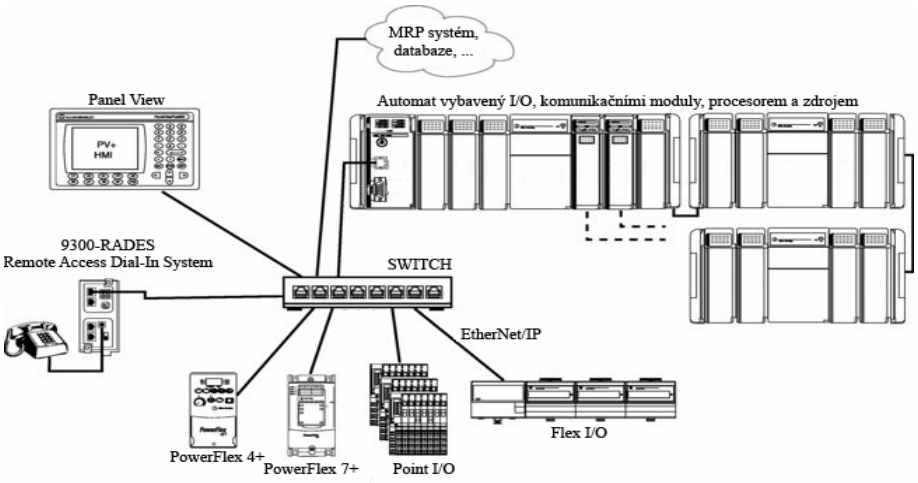
2.3.4 Shrnutí

V této části jsou shrnuty vlastnosti komunikačních linek architektury NetLinx (sítě EtherNet/IP a ControlNet). Pro úplnost popisu této architekrury je doplněna i poslední z linek DeviceNet.

EtherNet/IP

- schopnost komunikovat s manažerskými systémy
- konfigurace zařízení, sběr dat, řízení
- časově kritické aplikace bez podpory plánovaných přenosů (v současné době je však rozšiřován o tuto možnost, CIP Sync a CIP Motion [6])
- data odesílána pravidelně
- komunikace internet/intranet

Typické použití

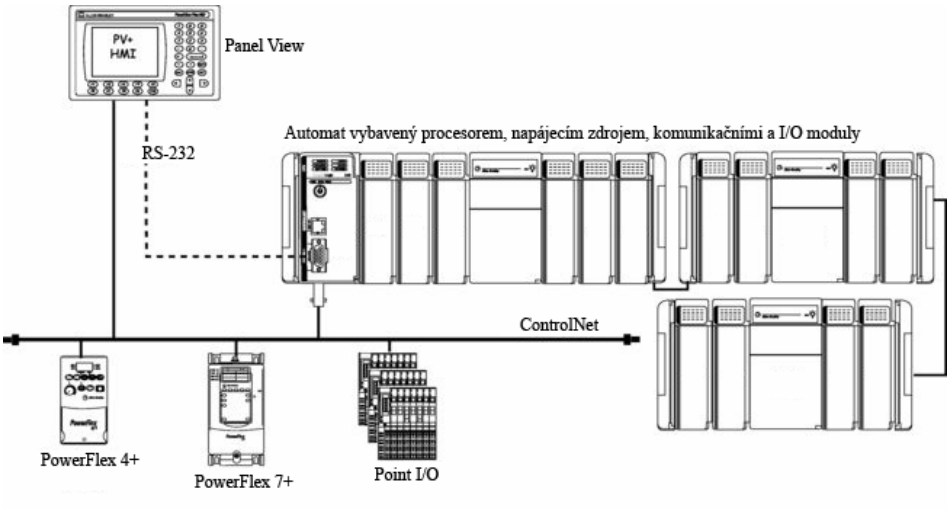


Obr. 2.17 Typické zapojení zařízení pomocí EtherNet/IP

ControlNet

- vysokorychlostní přenos časově kritických dat mezi automaty a I/O zařízeními
- deterministické a opakovatelné přijímání a odesílání dat
- redundance média
- jiskrová bezpečnost
- redundantní zapojení automatů (procesorů)

Typické použití

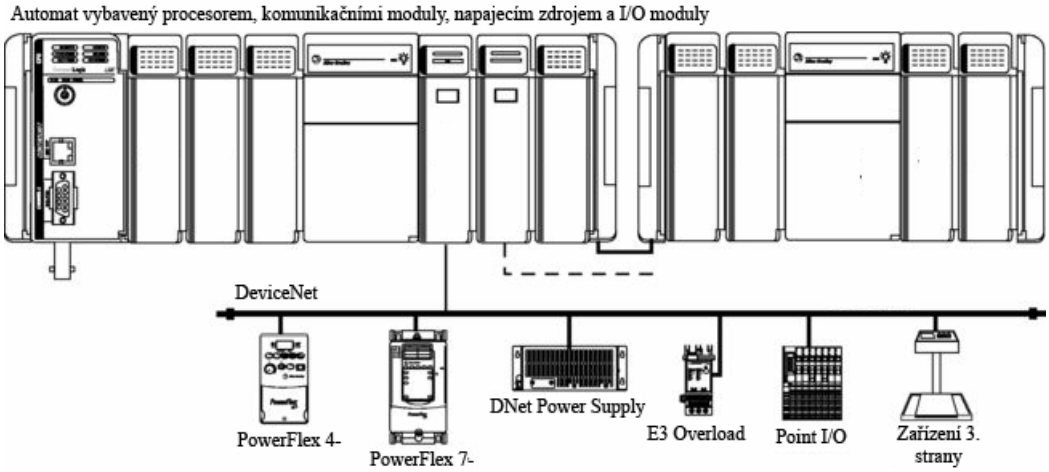


Obr. 2.18 Typické zapojení zařízení pomocí ControlNet

DeviceNet

- spojení se zařízeními na nejnižší úrovní (akční členy) bez použití rozhraní I/O modulů
- vysílání dat podle potřeby
- vyšší úroveň diagnostiky pro sběr dat a detekci chyb

Typické použití



Obr. 2.19 Typické zapojení zařízení pomocí DeviceNet

2.3.5 Porovnání komunikačních linek architektury NetLinx

Komunikační linka ControlNet se z pohledu nabízených možností různých typů komunikací a parametrů, jeví jako nejvýhodnější varianta. Jako druhou bych umístil linku EtherNet/IP a na posledním místě DevicNet.

Nicméně každá z těchto linek je navržena pro určitou specifickou oblast řízení a volit některou z těchto linek pro celou oblast řízení je nevhodné. Optimální řešení pro řízení výrobního procesu až po sběr dat, je tak možné dosáhnout pouze kombinací všech uvedených možností.

Společným využitím všech dostupných možností lze rovněž zajistit optimální výkon celého systému a snížit náklady na vytvoření průmyslové infrastruktury. Díky CIP protokolu, který všechny tyto linky podporují, je velmi snadné zajistit jejich vzájemnou spolupráci.

Parametr	EtherNet/IP	ControlNet	DeviceNet
Maximální délka segmentu	10 km	1 km (koaxiální kabel) 3 km (optický kabel)	100 m, 250 m, 500 m
Maximální délka	neomezená	20 km (koax., optický kabel nebo obojí)	4 km s opakovači
Maximální počet uzlů	1024 uzlů	99 uzlů	64 uzlů
Maximální rychlost komunikace	10 Mbit/s, 100 Mbit/s	5 Mbit/s	125 kbit/s, 250 kbit/s, 500 kbit/s
Použitelné médiu	koaxiální kabel, křížená dvou-linka, bez drátová komunikace	koaxiální kabel nebo optický	Kroucený svolila(dva páry, T2), plochý drát
Redundance média	Ano	Ano	Ne
Jiskrová bezpečnost	Ne	Ano	Ne
Napájení zařízení	z samostatného zdroje	z samostatného zdroje	24VDC/84AMPS
Standart	IEEE 802.3	IEC 61158	CENELEC EN 50325
Organizace	ControlNet International Open DeviceNet Vendor Associacion	ControlNet International	Opend DevicNet Vendors Associacion

Tab. 2.1 Porovnání linek NetLinx podle parametrů

Požadavek	EtherNet/IP	ControlNet	DeviceNet
Řízení I/O	Dobrá	Nejlepší	Dobrá (pro malý počet)
Konfigurace zařízení	Nejlepší*	Nejlepší*	Nejlepší*
Sběr dat	Nejlepší	Lepší	Dobrá
Zasílání zpráv automat-automat	Nejlepší *	Nejlepší *	-
Synchronizace automat-automat	Lepší	Nejlepší	-

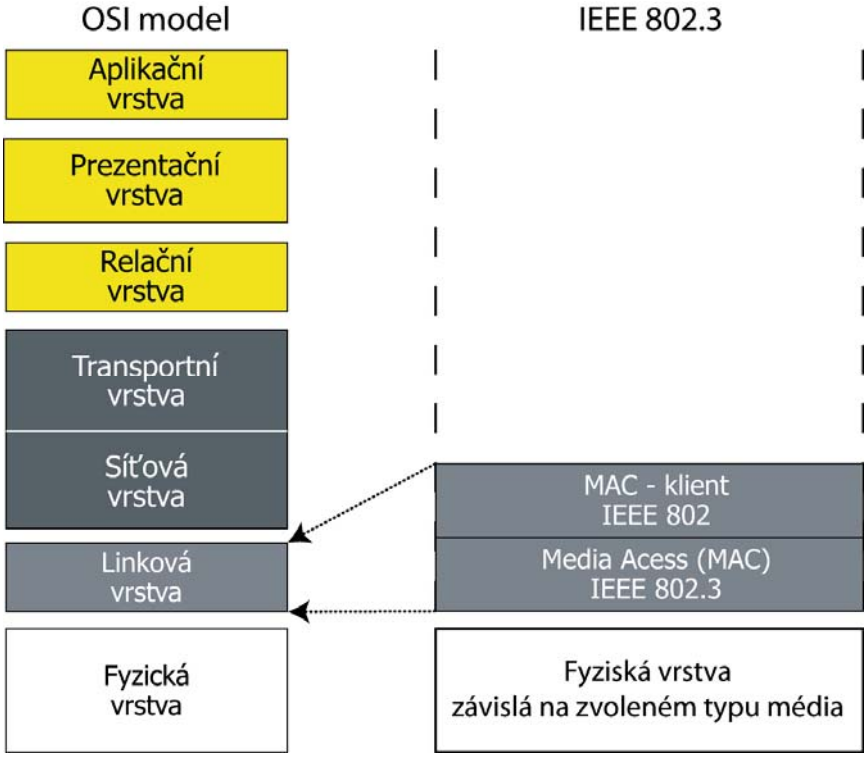
Tab. 2.2 Porovnání linek NetLinx podle typu komunikace (* stejný princip)

2.4 Ethernet

Ethernet byl vyvinut v roce 1970 společností Xerox Corporation jako experimentální sít, která byla spojovaná koaxiálními kabely. Přenosová rychlost byla 3 Mbit za sekundu a pro přenos se používal CSMA/CD (Carrier Sense Multiple Access Collision Detect) protokol. Úspěch tohoto projektu vedl v roce 1980 k vývoji 10 Mbitové verzi Ethernet Version 1.0

vytvořené sdružením tří společností Digital Equipment Corporation, Intel Corporation a Xerox Corporation.

Následně byl vytvořen IEEE 802.3 standart, který je velmi podobný Ethernet Version 1.0. Vylepšená verze tohoto standartu byla publikována jako oficiální standart v roce 1985 (ANSI/IEEE Std. 802.3 - 985). Pojem Ethernet bude dále využíván výhradně v souvislosti se standardem IEEE 802.3.



Obr. 2.20 Souvislost OSI modelu a sítě Ethernet

2.4.1 Fyzická vrstva

Protože Ethernet zařízení implementují pouze spodní dvě vrstvy OSI modelu, bývají typicky implementovány jako síťové karty (Network Interface Cards - NIC), které se vkládají do požadovaného zařízení, nebo jsou integrovány přímo do základních desek zařízení. Rozlišujeme tři základní druhy NIC založené na použitém přenosovém médiu

10Base-T = 10 Mbit/s, spojení kroucenou dvoulinkou (2 páry vodičů)

100Base-T2 = 100 Mbit/s, spojení kroucenou dvoulinkou (2 páry vodičů)

-T4 = 100 Mbit/s, spojení kroucenou dvoulinkou (4 páry vodičů)

1000Base-LX = 1 Gbit/s, optický kabel

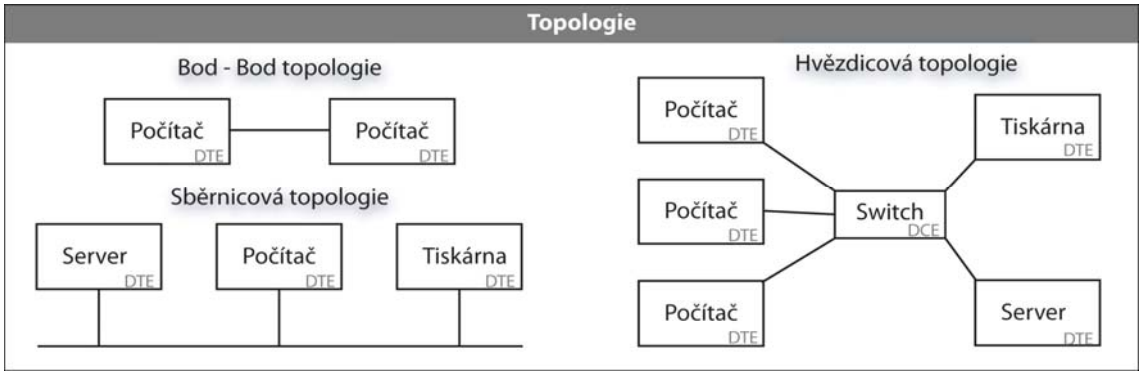
Podrobnější informaci viz. [15, 16, 17].

2.4.2 Topologie

Uzly sítě se dělí na dva základní typy:

Data terminal equipment (DTE) označuje zařízení, které jsou zdrojem nebo cílem vysílaných rámců. Typicky to bývají zařízení jako PC, tiskové servery, souborové servery (File Servers).

Data comunication equipment (DCE) se označují zařízení tvořící mezistupně, které přijímají a přeposílají rámce po sítí. DCE mohou být samostatně fungující zařízení jako jsou opakovače (Repeaters), přepínače (Switches) a směrovače (Routers) nebo komunikační rozhraní typu různých karet či modemů.



Obr. 2.21 Příklad možných topologií sítě Ethernet

2.4.3 Přístup na médium CSMA/CD

Ethernet používá pro přenos rámců sdílené médium. Pokud uzel vyšle na médiu rámec, tento rámec přijmou všechny uzly zapojené k médiu. Podle cílové MAC (Medium Access Control) adresy obsažené v rámci síťová zařízení poznají, zda je vyslaný rámec určen právě jim. Zařízení, kterým rámec není určen, s ním neprovádí další operace.

CSMA (Carrier Sense Multiple Access)

Jakmile má NIC data, která potřebuje vyslat, nejdříve poslouchá, zda na médium nevysílá jiný NIC. V případě, že data nejsou vysílána a médium není využíváno, začne NIC, který má data k odeslání, vysílat. Ostatní NIC poslouchají, protože nemají data, která by odeslala. Může však nastat případ kolize, kdy na médiu začne vysílat více NIC, ať už vlivem dopravního zpoždění nebo současným začátkem vysílání. V případě kolize bude narušen vyslaný rámec, což přijímající zařízení snadno pozná podle nesprávného 32 bitového kontrolního součtu (MAC CRC) nebo podle rušící sekvence vyslané zařízením, které kolizi detekovalo.

CD (Collision Detection)

Každý NIC provádí monitorování a kontrolu vyslaných dat. Jestliže zjistí kolizi na médiu (kontrola MAC CRC), odešlou 32 bitovou rušící sekvenci. Poté se všechny stanice odmlčí a o nové vysílání se pokusí později. Stanice mezi pokusy o vysílání čeká náhodnou dobu a čekací interval se mezi deseti pokusy vždy zdvojnásobuje. Tím se zvyšuje pravděpodobnost, že se stanice o médium lépe podělí a nedojde ke kolizi. Pokud se nepodaří rámec odeslat během šestnácti pokusů, informace o neúspěchu se hlásí nadřízené vrstvě.

Ke kolizi signálu přitom může dojít jen v tzv. kolizním okénku, což je doba, než vyslaný signál obsadí celé mediu a ostatní NIC jsou schopny rozpoznat, že je médium používáno. Kolizní okénko musí být kratší než vyslání nejmenšího rámce. Minimální velikost rámce je závislá na délce segmentu, typu média a počtu opakovačů, kterými signál musí

projít k nejvzdálenější části. Tyto parametry určují tzv. Ethernet Slot Time odpovídající např. pro 10 MBit a 100 Mbit síť 512 bitům (64 bajtů velký rámec). Pro CSMA/CD je vyšší efektivita pro větší rámce, protože je výhodnější poměr mezi trváním kolizního okénka a délkou vyslání dat. Podrobnější informace o CSMA/CD viz [18].

2.4.4 Popis základního komunikačního rámce Ethernet

IEEE 802.3 standart definuje základní datový formát rámce vyžadovaný MAC implementací s několika dalšími rozšířenými formáty. V této části se zaměřím na popis základního rámce (Obr. 2.22), který je pro vysvětlení skladby rámce dostačující.

PRE	SFD	DA	SA	LENGHT/TYPE	DATA	PAD	FCS
7 bajtů	1 bajt	6 bajtů	6 bajtů	4 bajty	46 -1500 bajtů		4 bajty

Obr. 2.22 Základní komunikační rámec

Preamble (PRE) označuje záhlaví, zabírá 7 bajtů a obsahuje střídající se jedničky a nuly, označuje začátek rámce a slouží k synchronizaci.

Start-of-Frame Delimiter (SFD), neboli omezovač počátku rámce, zabírá 1 bajt a obsahuje střídající se jedničky a nuly na konci ukončený dvěma jedničkovými bity označujícími místo, kde začíná adresa cíle.

Destination address (DA) označuje adresu cíle, zabírá 6 bajtů a identifikuje stanici, které má být rámec doručen. Nejvyšší bit v DA určuje, jestli se jedná o jednu adresu (určeno nulou), nebo jestli se jedná o skupinu adres (určeno jedničkou). Druhý nejvyšší bit určuje, zda se jedná o globální skupinu (určeno nulou) nebo zda se jedná lokální skupinu (určeno jedničkou). Zbylých 46 bitů je unikátní přidělená hodnota identifikující buď jednu stanici, definovanou skupinu stanic, nebo všechny stanice na síti.

Source address (SA) označuje adresu zdroje, zabírá 6 bajtů a identifikuje odesílající stanici. SA je vždy individuální stanice a nejvyšší bit je vždy roven nule.

Lenght/Type určuje délku případně typ. Zabírá 4 bajty. Pro Ethernet II. (upravená verze původního Ethernetu od Xerox Corporation) určuje typ vyššího protokolu, pro IEE 802.3 toto pole udává délku dat.

Data označují sekvenci *n* bajtů libovolné hodnoty. N musí být větší než 46 bajtů a menší nebo rovno 1500 bajtů. Pokud je hodnota menší než 46 bajtů, je doplněna o PAD na velikost 46 bajtů.

Frame Check Sequence (FCS) označuje kontrolní součet, zabírá 4 bajty a jedná se o sekvenci 32 bitového kontrolního součtu CRC (Cyclic Redundancy Check). Tato sekvence je vytvořena při odeslání rámce a přepočtena a ověřena při příchodu rámce. Takto určíme poškození rámce při přenosu. FCS je generováno z DA, SA, Lenght/Type a Data položek.

3 Porovnání komunikačních linek Rockwell Automation s dalšími linkami předních světových výrobců

Vlivem velké konkurence mezi světovými výrobci (např. Rocwell Automation, Siemens, Beckhoff, Honeywell, Bernecker + Rainer Industrie), jsou nabízené produkty a jejich využití velmi podobné. Takřka každý významný výrobce, aby se vůbec mohl v této konkurenci prosadit, nabízí kompletní sadu produktů, pomocí nichž lze zajistit automatizaci výroby na všech úrovních. Při volbě konkrétního dodavatele řídicího systému, se tak klade důraz na renomé společnosti a otevřenost využívaných komunikačních standardů.

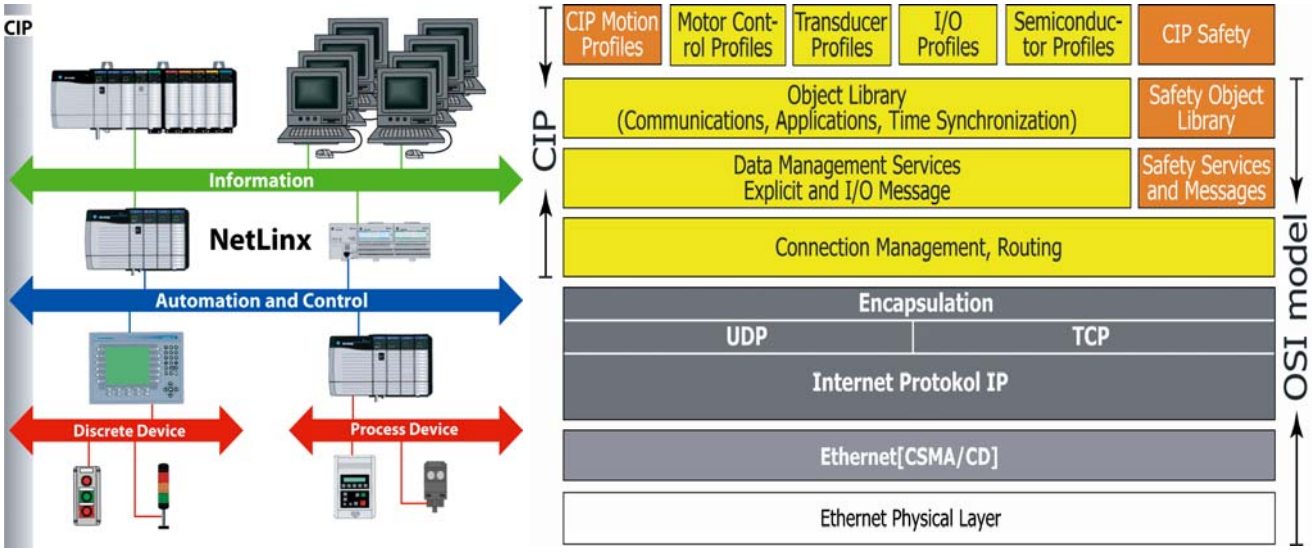
Domnívám se, že v budoucnu bude nejdůležitější komunikačním kanálem využívaným v průmyslu Ethernet. Ten je již dnes v průmyslu značně rozšířen kvůli rychlé komunikaci a nízkým nákladům na vybudování infrastruktury. V současnosti je kladen velký důraz na možnost využití Ethernetu pro real-time řízení, a proto se zaměřím právě na popis a srovnání této komunikace.

Rockwell Automation – EtherNet/IP

- standardní komunikační rozhranní
- synchronizace IEEE 1588
- následník DeviceNet (CAN) a ControlNet
- Organizace Open DeviceNet Vendor Association (ODVA)

Vlastnosti

- Soft Real-Time (s/bez hardwarové podpory) kolísání velikosti zpoždění paketu při průchodu sítí 5-10 μs
- Hard Real-Time (s hardwarovou podporou) kolísání velikosti zpoždění paketu při průchodu sítí < 1μs
- ruzné typy topologií sběrnicová, hvězdicová, kruhová
- komplexní konfigurace (nastavení priorit atd.)



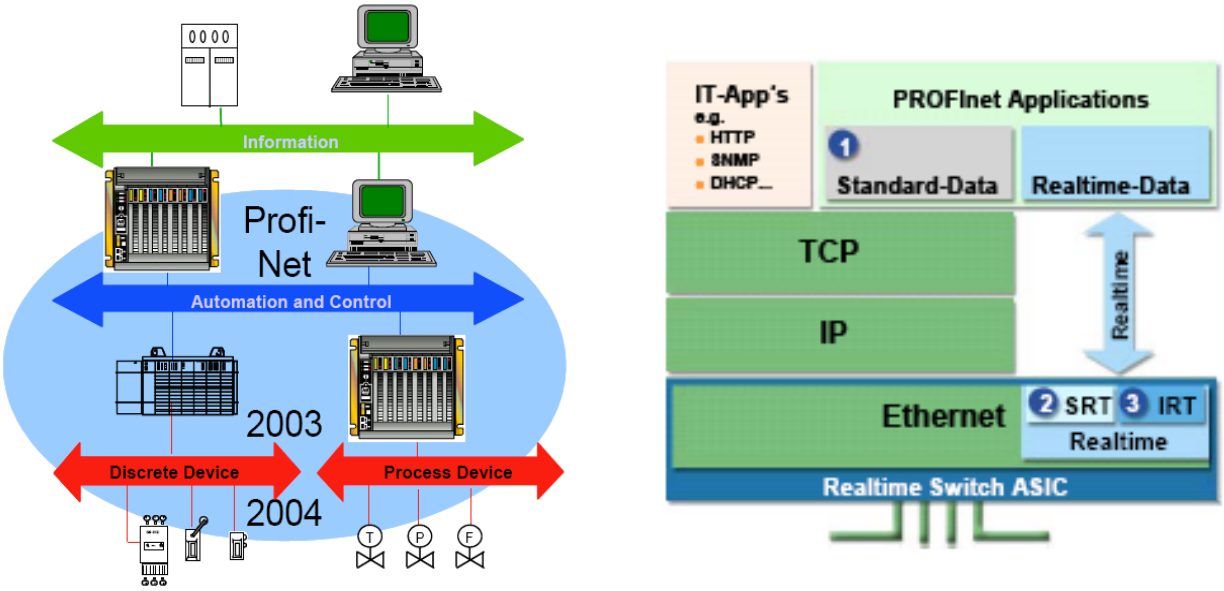
Obr. 3.1 Příklad řízení výroby na všech úrovních NetLinx (vlevo). Spojitost s OSI modelem (vpravo).

Siemens – PROFInet

- nezbytné speciální ASIC (Application-Specific Integrated Circuit) rozhraní, speciální rozhraní pro konkrétní aplikaci
- pro komunikaci se nevyužívá standardní ethernetovský rámec
- synchronizace IEEE 1588
- nástupce PROFIbus
- organizace Profibus Nutzer Organisation (PNO)

Vlastnosti

- Soft Real-Time kolísání velikosti zpoždění paketu při průchodu sítí (jitter) > 10µs (závisí na zvoleném času opakování vysílání paketu)
- Hard Real-Time (s podporou hardware) kolísání velikosti zpoždění paketu při průchodu sítí (jitter) < 1µs
- topologie omezena na hvězdicovou a sběrnicovou
- komplexní konfigurace (nastavení priorit atd.)



Obr. 3.2 Příklad řízení výroby na všech úrovních ProfiNet (vlevo). Spojitost s OSI modelem (vpravo).

Bosch Rexroth – SERCOS III

- specifikace dokončena v prosinci 2004 a předána IEC jako PAS (Publicy Available Specification)
- v roce 2007 přidělené číslo PAS 62410 (titulek: Real-Time Ethernet SERCOS III.)
- specifikace stojí 279 dolarů, proto bez dalších podrobností

Scheiner Electric – IDA, Modus TCP

- nezpůsobilý real-time komunikace

Beckhoff – EtherCAT

- nezbytné speciální ASIC rozhraní
- pro komunikaci se nevyužívá standardní ethernetovský rámec
- součást Interbus-S a CANopen
- organizace EtherCAT Technology Group (ETG)

Vlastnosti

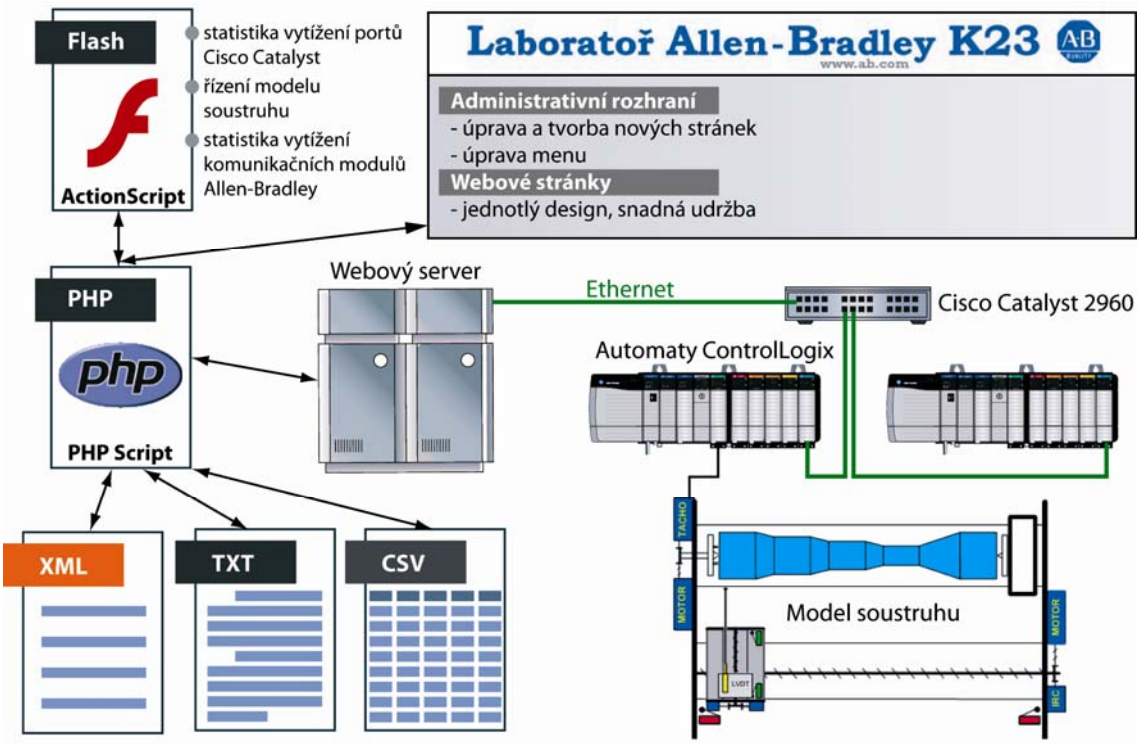
- synchronizační nestabilita > 1μs
- omezená volba topologie, pouze kruhová
- Plug-in-Play konfigurace
- standardní IP pouze tunelováním

4 Webové stránky Laboratoře Allen-Bradley K23

4.1 Popis funkcí

Webové stránky laboratoře (Obr. B.1) jsou navrženy tak, aby umožňovali snadnou údržbu (viz Příloha C) a jednotný design při vytváření a aktualizování obsahu. Nabídkové menu je dostupné ve dvou jazykových verzích a obsah je generován podle definic v xml souborech, které reprezentují menu. Menu může mít až čtyři úrovně. Podle xml souborů popisujících menu, se rovněž generuje navigační pruh, který nás informuje o pozici vybrané položky v menu.

Obsah stránek je generován z textových souborů, které obsahují xhtml kód. Při zaslání požadavku na zobrazení určité stránky uživatelem (kliknutí na odkaz ve stránce, či na položku v menu) skriptovací jazyk PHP doplní do základní definované struktury stránek obsah, který získá z textových souborů. Skriptovací jazyk rovněž provede kontrolu textu na výskyt nadpisů a v případě výskytu nadpisů H1, H2, H3 vygeneruje záložkové menu. Toto záložkové menu plní funkci osnovy a umožňuje odskakovat na pozice definované nadpisy.



Obr. 4.1 Struktura webových stránek

Pro snadnou údržbu obsahu je vytvořeno administrativní rozhraní s jednoduchým principem přihlašování (pouze jeden správce). Správce po přihlášením automaticky získává přístup na třicet minut, přičemž tuto dobu může libovolně prodlužovat podle potřeb. Administrativní rozhraní umožňuje upravovat nabídkové menu editací odpovídajícího xml souboru a dále umožňuje vytvářet a editovat textové soubory, které tvoří obsah stránek. Rovněž obsahuje možnost zaslání souboru na server. Zaslaný soubor se nahraje do správcem

zvoleného podadresáře v adresáři určeném pro upload souborů. Tento adresář obsahuje data, která se využívají pro vytváření obsahu stánek.

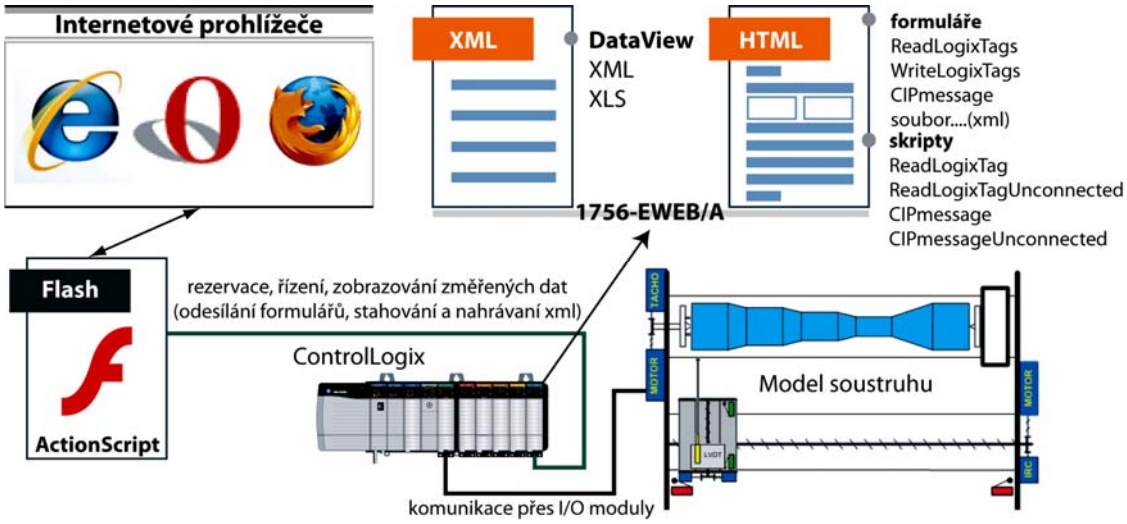
Administrativní rozhraní obsahuje jednoduchý editor, který umožňuje vkládat předem určené kusy xml či xhtml kódu. Chceme-li vložit např. tabulku, stačí kliknout na obrázek tabulky a do textového pole se vloží stylovaný xhtml kód popisující tabulku. V tomto kódu, správce vyplní pouze konkrétní obsahová místa. Ten samý princip je využit i při editaci xml souborů. Samozřejmě je rozhraní vybaveno i možností zobrazit si vytvořené stránky, či menu.

Webové stránky laboratoře K23 dodržují standart XHTML 1.0 a CSS. Jako skriptovací jazyk je použit PHP 5, a proto v současné době nejsou stránky umístěny na školním serveru. Funkční verze běží na soukromém serveru „zdenda.sh.cvut.cz/ab“. Jakmile v červenci 2007 školní FreeBSD server nahradí Linux s PHP 5, bude tato verze přemístěna a zprovozněna.

4.2 Webové řízení soustruhu

4.2.1 Úvod

Model dvouosého obráběcího stroje se nachází v Laboratoři Allen-Bradley. Byl zkonstruován a naprogramován Jiřím Novákem v rámci diplomové práce [22]. Neumožňuje provádět vlastní obrábění obrobku, ale měří tvar. Model po změření profilu dokáže kopírovat tento profil pohybem obráběcí části. Může být řízen ručně, automatem nebo pomocí Matlabu.



Obr. 4.2 Základní princip webového řízení soustruhu

První verzi webového řízení soustruhu jsem realizoval v rámci semestrální práce a semestrálního projektu. V této práci jsem využíval jazyky html, javascript, java applet a dvě základní ASP funkce, které poskytuje webový server 1756-EWEB/A. Java applet se používá pro zobrazení změřených dat a video signálu z webové kamery. Javascript se používá pro vytvoření dynamických stránek, kontrolu formulářů před odesláním, vytváření a čtení cookies. Rovněž se pomocí Javascriptu pohybuje obrázky, které vizualizují pohyb soustruhu. Stránky jsou uloženy ve webovém modulu automatu a jsou stále funkční [23, 24].

V rámci diplomové práce jsem toto řízení předělal do ActionScriptu (Obr. 4.2). Tento jazyk je pro webové řízení velmi vhodný. Umožňuje snadné vytvoření designově zajímavých aplikací a je podporován takřka všemi prohlížeči (stačí nainstalovat Flash Player). Rovněž umožňuje objektový přistup ke zpracování xml a velmi snadnou práci s formuláři. Dalšími výhodami Flash je možnost vytvořit binární spustitelný soubor (projekt) a nižší vytížení procesoru počítače při spuštění flashové aplikace oproti Java Appletům.

4.2.2 Rozšíření původního řídicího systému

Řídicí automat ControlLogix byl doplněn o webový server 1756-EWEB, který poskytuje http, xml a CIP rozhraní pro komunikaci s procesorem automatu. V našem případě se jedná o procesor 1756-L1/A LOGIX5550 firmy Allen-Bradley.

Řídicí program byl doplněn o řadu proměnných, které se využívají pro komunikaci procesoru s webovým serverem a webového serveru s uživatelem. Dále byl řídicí program doplněn o rezervační systém, který umožňuje výhradní přístup uživatele k modelu.

4.2.3 Princip komunikace internetový prohlížeč – server – procesor

Komunikace mezi modulem EWEB/A a procesorem L1/A probíhá po sběrnici ControlBus. Webový server se dotazuje procesoru na proměnné, které jsou určeny pro webové řízení. Jedinou podmínkou proto, aby EWEB dokázal přečíst určitou proměnnou z procesoru, je rozsah (Scope) dané proměnné. Vyžaduje se typ globální proměnné (Controller Tag).

4.2.3.1 Detailní popis komunikace

Komunikace mezi internetovými prohlížeči a EWEB modulem probíhá pomocí html, „ASP“ nebo xml. Internetový prohlížeč se dotazuje webového serveru pomocí zasílání formulářů metodou POST. Při dotazování webového serveru je nutné dodržet správný formát zaslaného formuláře (viz Příloha A.1). Pomocí formuláře se můžeme dotazovat na globální proměnnou, nebo můžeme požadovat od serveru odeslání CIP zprávy, spojované nebo nespojované.

Další způsob zjišťování informací z procesoru probíhá pomocí skriptů ASP. Tyto skripty musí být uloženy na serveru. Skripty obsahují kromě standardního html kódu (html nemusí být obsažen ve skriptu) tzv. ASP funkce. Tyto funkce jsou v kódu oddělené pomocí znaků „<% %>“ a mezi tyto znaky se pak vkládá aktivní obsah, tedy jména ASP funkcí, na jejichž místo server po zpracování skriptu vloží výsledek vykonané funkce. Server umožňuje uživateli využít tři základní ASP funkce ReadLogixTag (čtení proměnné z procesoru), CIPmessage (odeslání zprávy spojovaným typem komunikace), CIPmessageUnconnected (odeslání zprávy nespojovaným typem komunikace).

Pro zjišťování informací z procesoru lze použít i dotazovaní na xml soubory, tyto soubory se musí ale nejdříve vytvořit v administrativním rozhranní webového serveru. Po nadefinování xml souboru tento soubor získáváme pomocí ActiveX prvků ze serveru a

mužeme s ním dále pracovat. Pokud chceme nastavit nějakou hodnotu v procesoru pomocí xml, stačí provést editaci obsahu jednotlivých elementů resp. atributů elementů xml. Následně tento vytvořený obsah odešleme serveru a ten pokud je dodržena struktura xml odpovídající existujícímu xml souboru na serveru, provede aktualizaci hodnot. Tento způsob je vhodný pro přijímání a odesílání velkého počtu proměnných. Bohužel je však omezen užitím ActiveX (podporuje Opera a Internet Explorer, nepodporuje Mozilla Firefox).

4.2.3.2 Omezení komunikace

Při vytváření webového řízení jsem narazil na problém kešování (cache) webových stránek na straně prohlížeče a kešování na straně proxy serveru. Kešování způsobí, že se obsah xml souboru, nebo výsledek zpracování skriptu, uloží do keše a při opakovaných dotazech, se již bere daný soubor či skript z keše. Toto chování lze omezit vložením META tagů do kódu (použitelné pro skripty nikoliv pro xml soubory). Kešování na straně prohlížeče je vlastní zejména prohlížečům Internet Explorer a Opera. Mozilla při odeslání dotazu získá aktuální data.

Použitím META tagů se mi kešování nepodařilo vyrušit, a proto jsem zvolil jiný způsob pro získání aktuálních informací. Pro získávání dat používám odesílání formulářů metodou POST. K dotazu navíc přikládám „nevyužívaná“ data (skript je obdrží jako vstupní proměnnou $_POST a dále s nimi nijak nepracuje, [29]). Dokud jsem neprovedl přiložení „nevyužívaných“ dat, prohlížeče Opera a Internet Explorer stále užívali kešovaná stará data. Jakmile přiložíme jakákoliv nesmyslná data do formuláře a navíc je odešleme metodou POST, vynutíme si přijmutí aktuálních dat ze serveru ve všech zmiňovaných prohlížečích.

4.2.4 Princip rezervace

Cílem rezervace modelu je povolit řízení v daném čase pouze jednomu uživateli a ostatním umožnit pouze sledování vykonávaných akcí. Pro zajištění tohoto výhradního přístupu je v řídicím programu dopsána rezervační část.

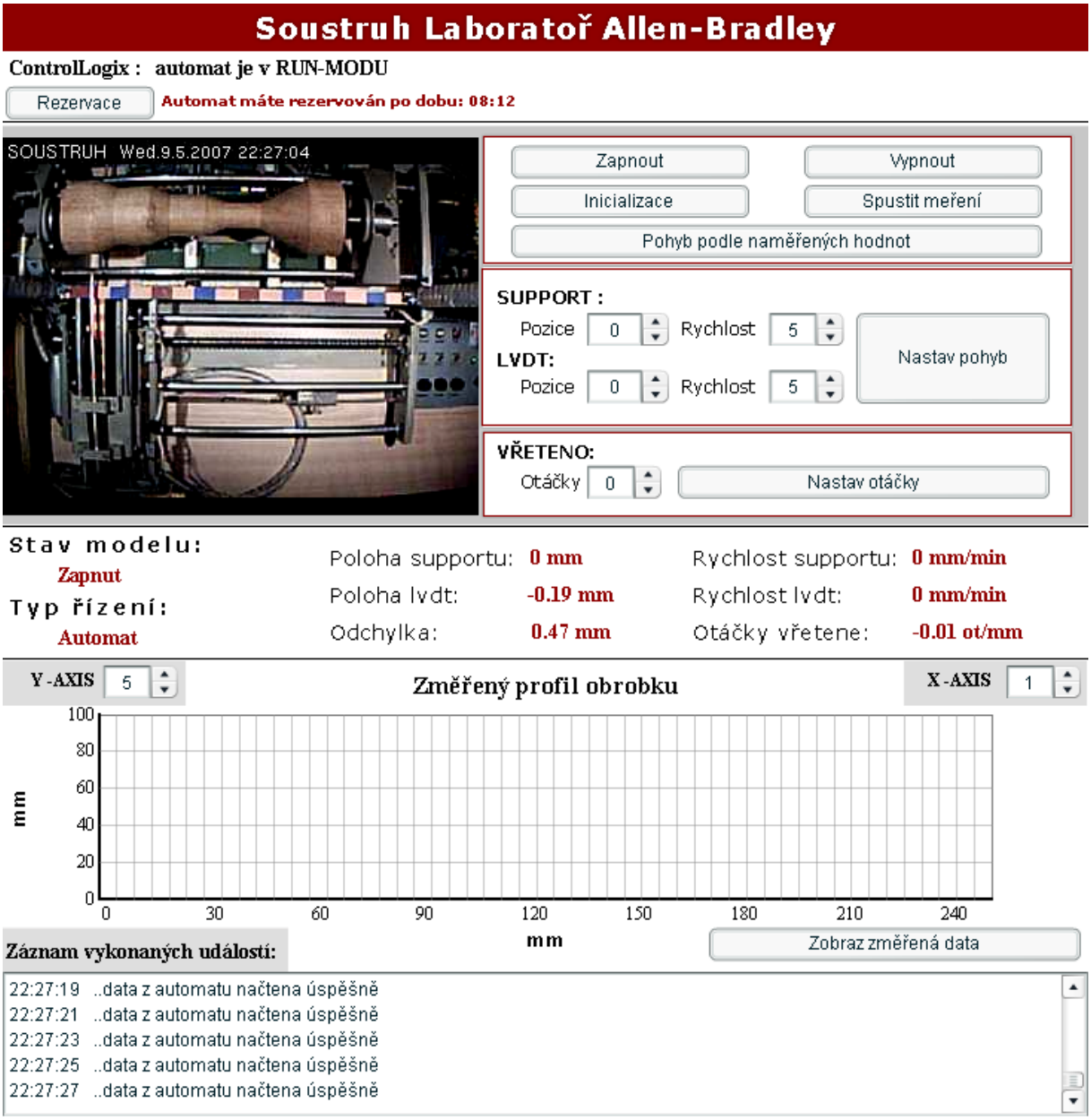
Uživatel kliknutím na tlačítko Rezervace odešle formulář obsahující jeho systémový čas, webový server zpracuje tento formulář a předá procesoru požadavek s nastavením hodnoty proměnné Web_cookie (obsahuje systémový čas). Původní hodnota WEB_cookie „Nerizeno“, je následně nahrazena zaslaným systémovým časem. Čas je uchován u uživatele, který zaslal formulář, a v procesoru automatu. Žádný další klient tento čas uchován nemá. Klient, který nemá uložený stejný čas jako je čas uložený v procesoru automatu, nemá přístupná tlačítka pro řízení automatu. Vyhodnocování shodnosti probíhá na straně klienta v pravidelných interval 1.5 sekundy. To je interval, ve kterém každý klient získává aktuální data ze serveru.

Automat v okamžiku nastavení hodnoty proměnné WEB_cookie, spustí časovač, který čítá po dobu deseti minut. Po uplynutí doby deseti minut je proměnná Web_cookie nastavena na svou původní hodnotu „Nerizeno“. Uživatel, který měl výhradní přistup o něj přichází,

neboť se neshoduje jeho uložený systémový čas s hodnotou proměnné WEB_cookie v automatu. Uživatel samozřejmě může dobu deseti minut prodloužit tak, že klikne opětovně na tlačítko Rezervace. Tím se provede resetování čítače a uživatel má opět přidělenu dobu deseti minut, po kterou může řídit model.

4.2.5 Vlastní řízení soustruhu ActionScriptem 2.0 (Flash)

Flash aplikace využívá pro získání dat ze serveru všechny typy komunikace (xml, formuláře, skripty). Příkazy jsou zasílány individuálně pomocí formulářů. Výsledek měření je získáván pomocí načtení xml, neboť měření obsahuje 255 hodnot. Informace o stavu automatu jsou získávány pomocí skriptu. Ten obsahuje přibližně patnáct položek, což převyšuje použitelnost formulářů, ale současně je „málo“ na použití xml.



Obr. 4.3 Webové řízení soustruhu ActionScriptem (Flash)

Řízení se uskutečňuje nastavením logické proměnné na určitou hodnotu. Tyto proměnné po nastavení spustí podprogramu (Rutine), který provede veškeré nezbytné operace.

Jak je vidět (Obr. 4.3), aplikace je doplněna o záznam vykonávaných akcí. Toto okno plní funkci diagnostiky komunikace klienta se serverem a jsou v něm zaznamenány veškeré prováděné akce.

4.3 Statistika zatížení sítě

4.3.1 Úvod

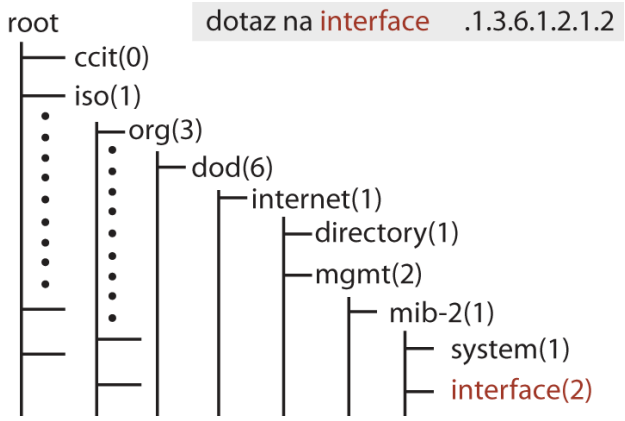
Pro měření statistik zatížení sítě, jsem zvažoval využití mnoha programů od MRTG (Multi Router Traffic Grapher), Adrem SNMP Manager po PRTG (Paessler Router Traffic Grapher). Jediný podléhající licenci GNU, a tedy „volně“ použitelný, je MRTG. Ten však nesplňoval očekávání na grafické zpracování výstupu a označení vzorků. Proto jsem vytvořil vlastní nástroje na měření statistik zatížení sítě. V těchto nástrojích zobrazuji pouze data, která jsem potřeboval pro analýzu komunikace mezi automaty.

4.3.2 Princip funkce

Získávání informací o zatížení sítě probíhá pomocí SNMP protokolu (Simple Network Management Protocol). Dotazy, kterými se získávají informace o stavu konkrétního zařízení, jsou vytvářeny a interpretovány pomocí MIB objektů (SNMP MIB databáze). Tyto objekty mohou být rozdílné pro různé typy síťových zařízení, ale vytvořené programy využívají pouze jádro MIB, které je společné pro všechna síťová zařízení.

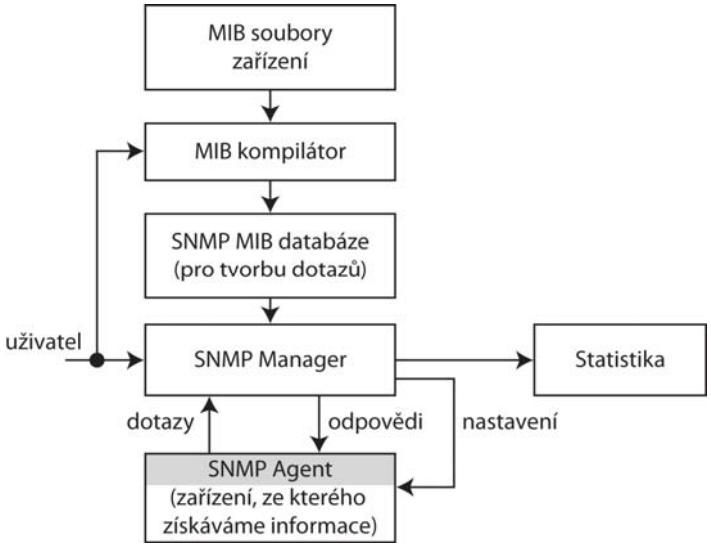
MIB (Management Information Base) je objektově orientovaná báze a data jsou v ní uložena jako objekty, které se sdružují do tříd. V těchto objektech se sdružují veškeré informace potřebné pro popis dané položky. MIB databáze je tvořena stromovou strukturou MIB objektů jakou můžeme vidět na Obr. 4.4.

Adresování položek probíhá pomocí číslic a je oddělováno tečkou. Tečka plní roli posunu do další větve a číslo určuje konkrétní větev.



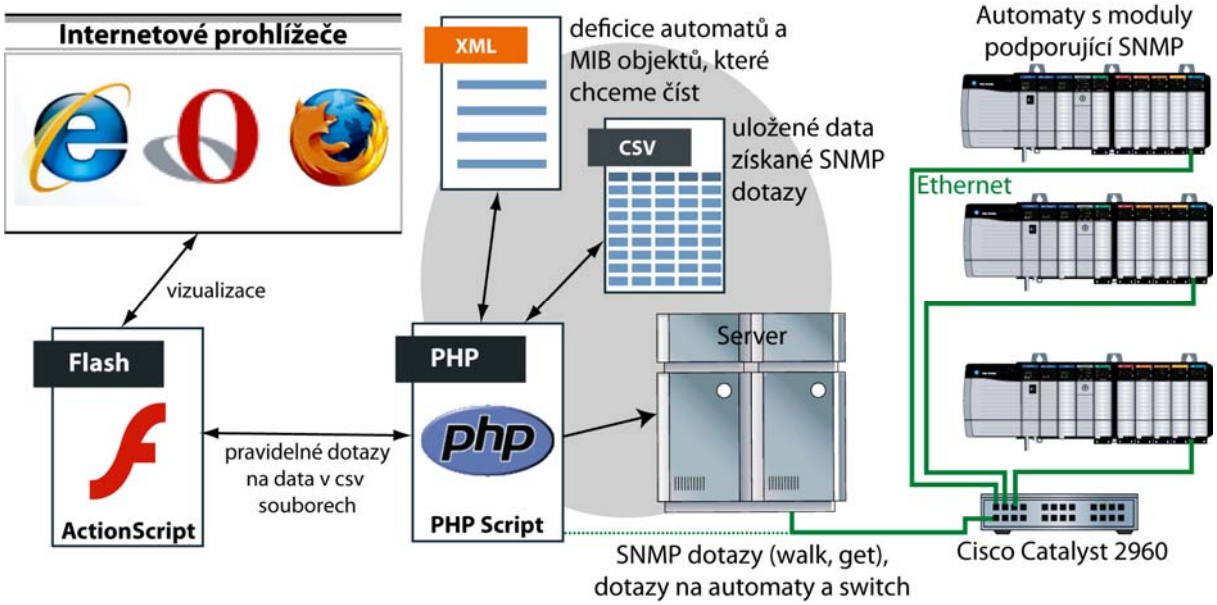
Obr. 4.4 Stromová struktura MIB

Nezbytným předpokladem funkčnosti je, aby zařízení, kterých se chceme dotazovat pomocí SNMP protokolu, měly v sobě naimplementovaného SNMP agenta. Dotazování provádí tzv. SNMP Manager, který se může dotazovat pouze na MIB objekty, které dokáže interpretovat díky vytvořené SNMP MIB databázi. Získaná data z agenta poskytují informace o stavu sítě.



Obr. 4.5 Komunikace SNMP

Roli SNMP manažera plní v mnou vytvořených programech PHP skript (Obr. 4.6). Ten se dotazuje jednotlivých zařízení (SNMP agentů) na MIB objekty (např. Interface, System, UDP, TCP, IP atd.). Seznam objektů, které nás zajímají, je nadefinován v xml souborech.

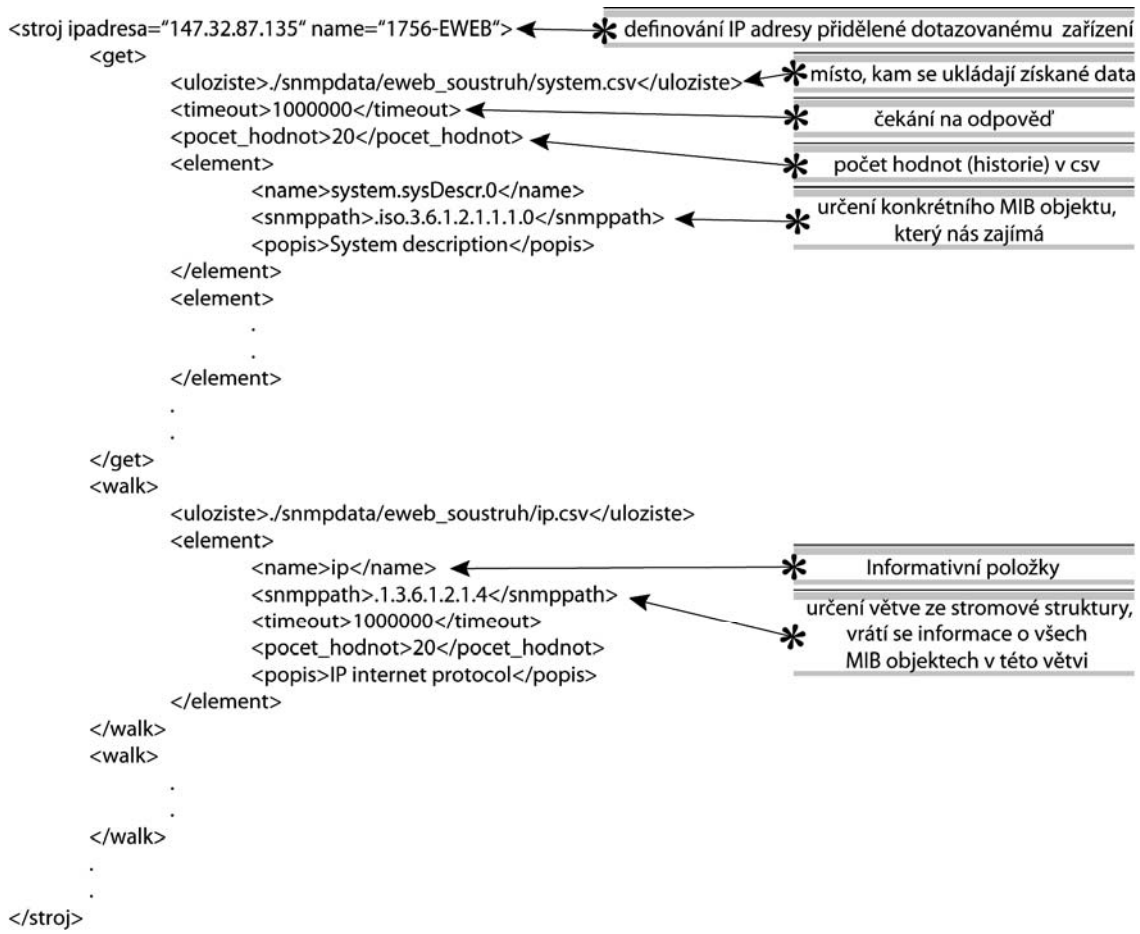


Obr. 4.6 Princip funkce webových stránek

Xml soubory obsahují dva druhy elementů. První se používá pro dotazování typu GET, kdy se získává informace pouze o jednom MIB objektu. Druhý se používá pro dotazování typu WALK, kdy se získávají informace o skupině MIB objektů. Tyto elementy

jsou dále větveny na podelementy, které je blíže specifikují. Podelementy určují kam se získaná data budou ukládat, jaký bude počet ukládaných položek, jaká je maximální prodleva mezi dotazem manažera a odpovědí agenta atd. (Obr. 4.7)

Získávání dat tedy provádí PHP skript podle definic obsažených v xml souborech. Výsledky dotazů ukládá do datových souborů typu csv (Comma-Separated Values). Csv soubory jsou vytvořeny s oddělovačem, který vyžaduje program Microsoft Excel. To umožňuje načíst tyto soubory v Excelu a dále pracovat se získanými daty (např. vytvářet grafy). PHP skript získávající data je spouštěn v plánovaných interval, které může obstarávat pod Linuxem například program Cron a pod Windows například Naplánované úlohy.



Obr. 4.7 Definice xml souboru nezbytná pro získání informací o stavu vytížení sítě z jednotlivých zařízení

Pro vizualizaci dat prostřednictvím Internetu se používá ActionScript (Flash). Ten se v pravidelných intervalech dotazuje speciálního PHP skriptu na data v csv souborech. ActionScript umožňuje dotazovat se přímo na csv soubory a následně je zpracovat. Kvůli prohlížečům Opera a Internet Explorer to není vhodné. Tyto prohlížeče přečtou csv soubor pouze jednou a uloží si jej do keše. Při opakovaných dotazech následně poskytují data z keše a ActionScript vizualizuje neustále stejná data. Použitím PHP skriptu, kterého se dotazujeme na data prostřednictvím formuláře, tomuto chování zamezíme.

Dalším důvodem, proč využít PHP skript pro předávání dat z csv souborů, jsou restriktivní bezpečnostní opatření ActionScriptu. Ten umožňuje lokální přístup k datům (data

musí být uložena na stejném místě jako samotný skript) nebo síťový přístup k datům (data získává prostřednictvím http protokolu). U síťového přístupu je navíc rozšíření bezpečnostních omezení. Ty povolují získávat data pouze v rámci jedné domény. Toto zabezpečení lze omezit, buď prostřednictvím parametrů při vytváření ActionScript projektu, nebo tak, že se budeme dotazovat pomocí formuláře PHP skriptu, který v odpovědi na dotaz zašle požadovaná data.

4.3.3 Určení zatížení v kBajtech/s

Určení průměrného zatížení v kBajtech/s během jednoho intervalu je určeno podle následující rovnice.

$$kBajty / s = \frac{současná_hodnota[Oktety] - minulá_hodnota[Oktety]}{1024 \cdot (délka_intervalu[Sekundy])} \tag{4.1}$$

Délka intervalu určuje čas mezi spuštěním PHP skriptu, který získává data ze SNMP agenta. Současná a minulá hodnota je údaj čítače (MIB objektu), který obsahuje informace o počtu přijatých oktetů.

4.3.4 Omezení vytvořených programů

Pro správnou funkci programu je nutné dodržet maximální délku mezi zjišťováním statistických údajů 343,6 sekundy (pro 100 Mbitové zařízení). Údaje o oktetech jsou v zařízeních uchovávány v proměnných typu Counter32. Jejich maximální velikost je $2^{32} - 1$ *(4294967295)* a v okamžiku přetečení je hodnota nastavena na *0*. Bude-li údaj čítače blízko maximální hodnotě, snadno při větším/menším toku dat, tato hodnota překročí svou maximální hodnotu a čítač přeteče. Rozdíl současné a minulé hodnoty z čítače, vlivem přetečení, bude záporný a tím bude i údaj kBajty/s (viz Rovnice 4.1) Tuto chybu odstraníme přičtením maximální hodnoty $2^{32} - 1$ k současné hodnotě zjištěné z čítače. Takto odstraníme vliv jednonásobného přetečení.

Bude-li však doba mezi vzorky nebo komunikace mezi zařízeními příliš velká, může čítač přetéct několikrát a údaj o skutečném zatížení sítě neurčíme. K určení zátěže v tomto případě bychom potřebovali znát počet přetečení mezi současnou a minulou hodnotou . Tento objekt se mi však v MIB objektech nepodařilo nalézt, a proto je nutné dodržet vypočtený interval mezi vzorky pro zajištění správné funkce programu.

Maximální propustnost 100 Mbitového zařízení 12207 kBajtů/s:

$$\begin{aligned} &\underline{100\ Mbitů / s} = 12500000\ Oktetů / s \quad (100000000\ bitů / 8) \\ &12500000\ Oktetů / s = \underline{\underline{12207\ kBajtů / s}} \quad (12500000\ Oktetů / 1024) \end{aligned} \tag{4.2}$$

Maximální interval mezi vzorky:

Uvažujeme maximální možnou zátěž 12207 kBajtů/s (viz Rovnice 4.2) a maximální možný nárůst hodnoty čítače $2^{32} - 1$.

$$12207\left[kBajtů / s\right] = \frac{\left(2^{32} - 1\right) \cdot čas\left[s\right]}{1024}$$

$$čas\left[s\right] = \frac{\left(2^{32} - 1\right)}{12207 \cdot 1024} \tag{4.3}$$

$$\underline{\underline{čas = 343{,}6\,s}}$$

Vypočítaná maximální bezpečná doba mezi vzorky je 343,6 sekundy. Čas 343,6 sekund určuje dobu, za kterou přeteče čítač typu Counter32, bude-li síť maximálně zatížena. Nebude-li počet kBajtů/s roven hodnotě maximální (12207 kBajtů/s), tato doba bude větší. Nedoporučuji však volit větší interval než interval vypočítaný, neboť by mohlo dojít k nesprávnému určení zatížení.

Nebude-li skript delší dobu spuštěn, získáme nepravdivé informace vlivem možného několikanásobného přetečení čítače. Tento mezní stav je ošetřen nahrazením vypočtené hodnoty nulovou hodnotu. Nulová hodnota v grafu indikuje vícenásobné přetečení nebo vypnutý model.

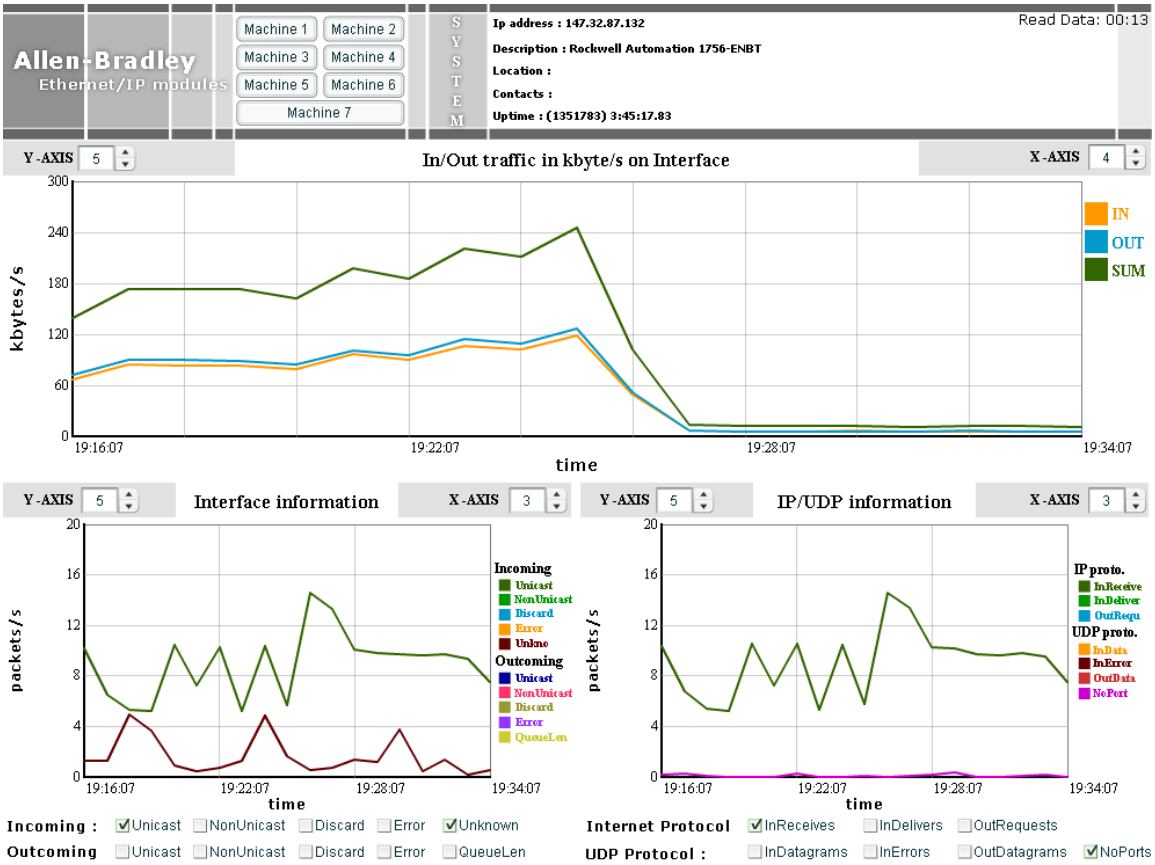
4.3.5 Statistika zatížení ethernetových komunikačních modulů Allen-Bradley

Tento program je vytvořen pro zobrazování statistik sedmi 100 MBitových ethernetových modulů (pro všechny moduly, které jsou fyzicky připojené do přepínače Cisco). Mezi tyto moduly patří webové servery 1756-EWEB/A, Ethernet/IP moduly 1756-ENBT/A a PLC automat 1785-L40E PLC5-40/E.

Program byl původně vytvořen pro zobrazování libovolného počtu zařízení, ale musel být předělán do současné podoby (Obr. 4.8). Při testování programu jsem zjistil, že u některých typů komunikačních modulů nejsou zobrazena „správná“ data (ověřování správnosti zobrazených dat proběhlo pomocí programu Wireshark a užitím vlastnosti zrcadlení portů přepínače Cisco). Nesouhlas dat získaných přes SNMP agenta v modulu s daty odposlechnutými na lince Ethernet byl dán rozdílným indexováním komunikačních rozhraní v jednotlivých modulech. U starších typů je rozhraní vysílající přes ControlBus registrováno pod číslem např. 1. Rozhraní vysílající přes Ethernet pod číslem např. 0 (moduly 1756-ENET/B, 1785-L40E PLC5-40/E). U nových modulů je toto značení opačné (moduly 1756-EWEB/A, 1756-ENBT/A).

Další objevenou chybou je nečítání čítače UDP paketů u ethernetových modulů. V závislosti na tom SNMP agent poskytuje programu nepravdivé údaje, podle kterých se vykresluje graf (ověření programem Wireshark a programem na zobrazování statistik vytížení jednotlivých portů přepínače Cisco viz kapitola 4.3.6).

Důležitý údaj o počtech přijatých a odeslaných oktetech je správný, a to je pro diagnostiku možného zátížení rozhodující parametr.



Obr. 4.8 Statistika vytížení ethernetových modulů Allen-Bradley

4.3.6 Statistika zatížení jednotlivých portů přepínače Cisco Catalyst 2960

Tento program nevyužívá pro dotazování SNMP agenta pouze jádro MIB objektů. Pro zobrazování statistik se používají MIB objekty RMON (Remote Monitoring). Aby mohl SNMP agent použít RMON, musí se nakonfigurovat. Jako první krok musíme povolit SNMP agenta na přepínači a nastavit způsob komunikace a ověřování.

Dále musíme nakonfigurovat RMON statistiky či RMON historii. RMON statistika poskytuje údaj o aktuálním stavu zařízení. RMON historie se používá pro výpočet a záznam průměrných hodnot v definovaných intervalech. Vytvořený program (Obr. 4.9) si vystačí s RMON statistikami, neboť uchovávání a výpočet průměrných hodnot zajišťuje sám.

4.3.6.1 Nastavení SNMP a RMON na přepínači

Jako první krok musíme zprovoznit agenta na přepínači. Pro zprovoznění agenta je nezbytné nastavit identifikátor snmp serveru (SNMP agenta) a vytvořit uživatele, pod kterým budeme získávat data. Standardně se vytváří uživatel „public“ s právem číst data. Rovněž můžeme provést nakonfigurovaní tzv. traps (zpráv). Ty se používají pro informování SNMP manažera v případě překročení mezních hodnot či při změnách konfigurace atd.

<u>Povolení SNMP agenta:</u>

```
1. telnet 147.32.87.158
2. CiscoK023# configure terminal
   // # značí přihlášení na úrovni 15, tzv. root oprávnění
3. CiscoK023(config)# snmp-server engineID local 10000000
```

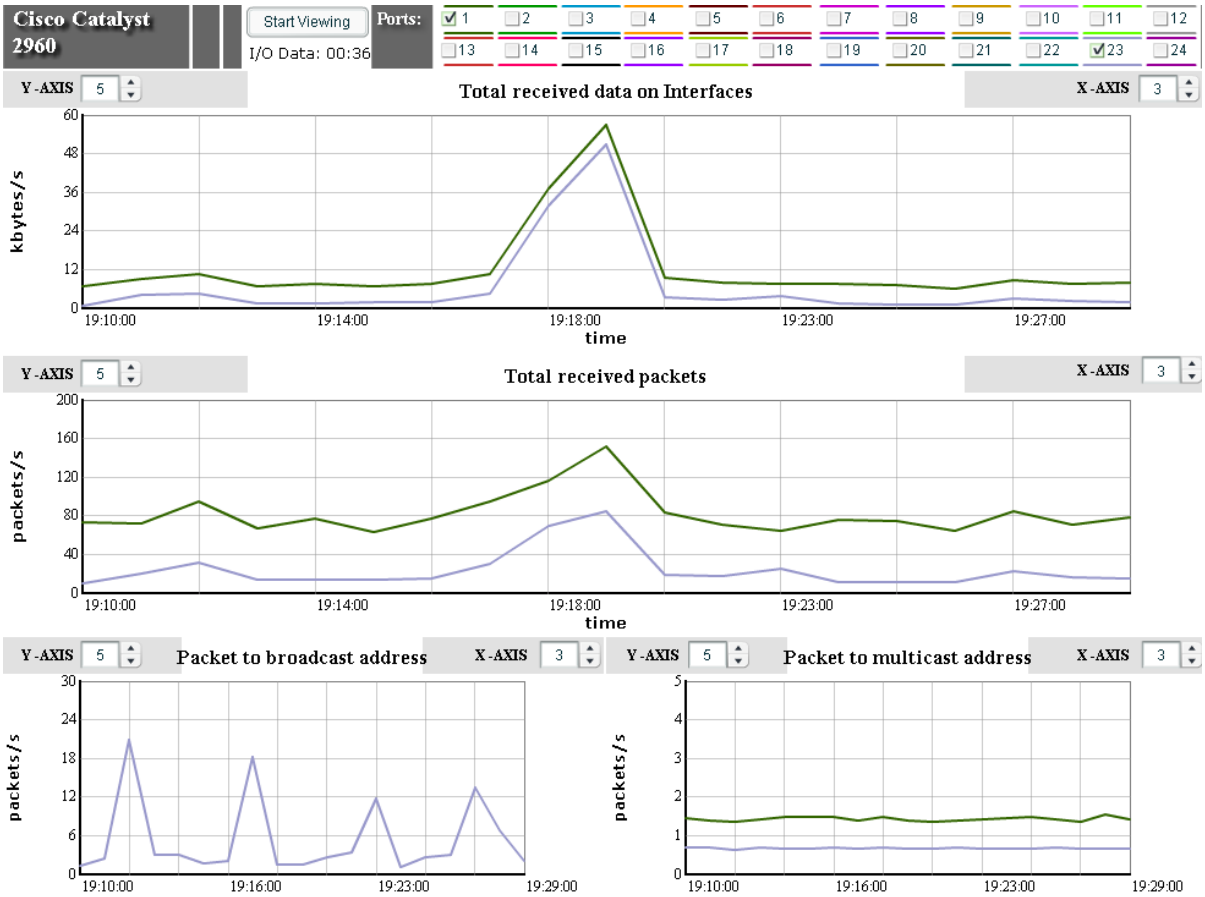
4. `CiscoK023(config)# snmp-server engineID remote 147.32.87.157 1100000000`
   // vytvoření autentifikace, ověřování na přepínači; IP 147.32.87.157 je adresa počítače, kde byl umístěn Adrem SNMP Manager, který přijímal traps (zprávy) od přepínače
5. `CiscoK023(config)# snmp-server community public ro`
   // povolení čtení (read-only) pod standardním SNMP uživatelským jménem public
6. `CiscoK023(config)# snmp-server community ..... rw 1`
   // čtení a zápis (read-write) pro ……. z IP adres povolených v přístupovém listu 1 (aceess-list 1)
7. `CiscoK023(config)# access-list 1 permit 147.32.87.157`
   // vytvoření přístupového listu označeného jako 1
8. `CiscoK023(config)# snmp-server contact .....`
9. `CiscoK023(config)# snmp-server location .....`
   // zadání parametrů SNMP serveru
10. `CiscoK023(config)# snmp-server enable traps cpu threshold`
11. `CiscoK023(config)# snmp-server enable traps syslog`
   // povolení zasílání upozornění v případě překročení mezních hodnot zatížení cpu a při změnách konfigurace přepínače
12. `CiscoK023(config)# snmp-server host 147.32.87.157 traps public`
   // nastavení adresáta zpráv při nastavených událostí

Povolení a nakonfigurování RMON statistik:

13. `CiscoK023(config)# interface FastEthernet 0/1`
14. `CiscoK023(config)# rmon collection stats 1 owner Fa0/1`
15. `CiscoK023(config)# interface FastEthernet 0/2`
16. `CiscoK023(config)# rmon collection stats 1 owner Fa0/2`

až

17. `CiscoK023(config)# interface FastEthernet 0/24`
18. `CiscoK023(config)# rmon collection stats 1 owner Fa0/24`
   // povolení statistik na jednotlivých portech



Obr. 4.9 Statistika vytížení jednotlivých portů přepínače Cisco Catalyst 2960

5 Testování propustnosti EtherNet/IP

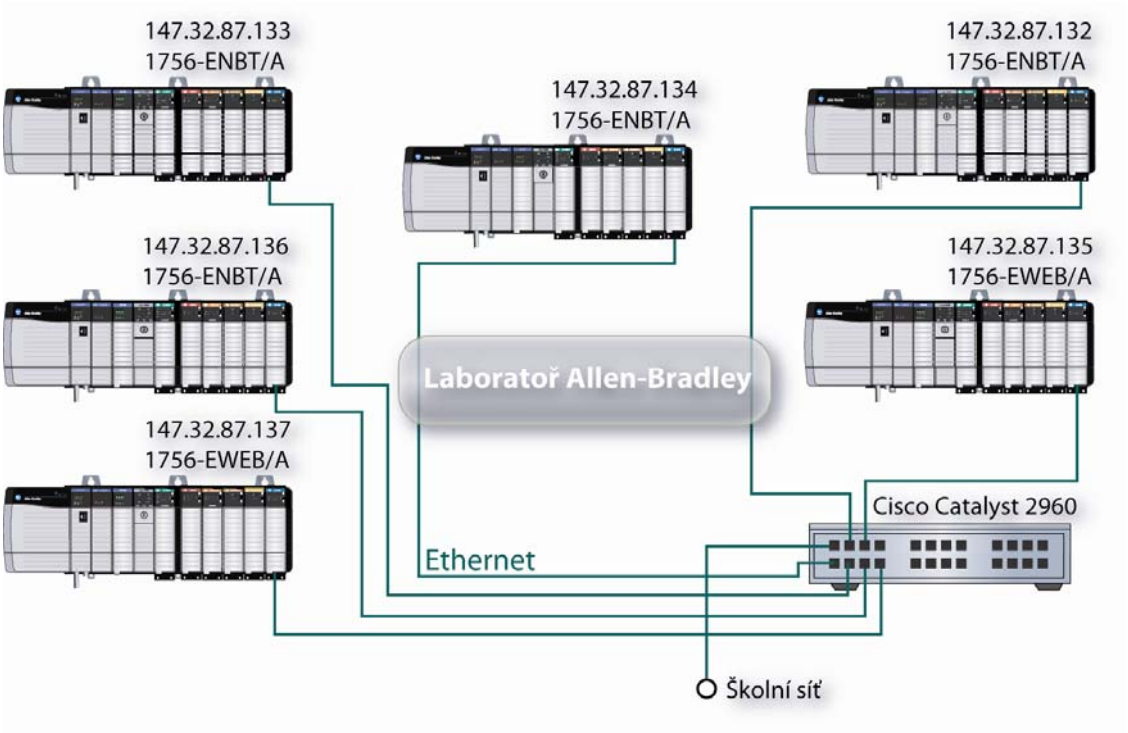
5.1 Zapojení sítě v laboratoři

5.1.1 Úvod

První krok byl výběr vhodného zařízení DCE (aktivní síťový prvek). Bylo nutné rozhodnout se mezi přepínačem s managementem nebo přepínačem bez managementu. Protože pro další nastavování a optimalizaci komunikace je nutné zařízení, které dovolí provádět nastavování parametrů sítě, byl zvolen přepínač s managementem. Konkrétní typ zařízení byl vybrán ze seznamu doporučovaných zařízení Rockwell Automation Encompass [30]. Byl zvolen produkt Cisco Catalyst 2960.

Dalším krokem po výběru přepínače bylo určení součastného způsobu zapojení síťových prvků v Laboratoři Allen-Bradley K23. Zapojení laboratoře je provedeno pomocí dvou SMC přepínačů bez managementu. Na těchto přepínačích jsem určil porty, na kterých jsou připojeny komunikační zařízení automatů a přepojil, společně s jedním počítačem, do přepínače Cisco. Nyní jsou počítače v laboratoři zapojeny pomocí SMC přepínačů a téměř všechny automaty prostřednictvím přepínače Cisco (Obr. 5.1).

Kvůli testování propustnosti musely být nahrazeny starší 10 Mbitové komunikační moduly 1756-ENET/B novými 100 Mbitovými 1756-ENBT/A moduly. Po přepojení či přidání nových modulů se vyskytovaly duplicitní IP adresy a nebyly číslované v logické posloupnosti. Proto jsem provedl nové přiřazení IP adres modulům prostřednictvím programu RSLinx.



Obr. 5.1 Schéma připojení automatů v laboratoři

5.1.2 Konfigurace přepínače

Základní nastavení a uvedení Cisco přepínače do provozu je popsáno viz Příloha E. Po provedení této konfigurace, však přepínač nešlo zapojit do školní sítě. Po připojení do sítě byl port, přes který se komunikovalo se školní sítí, automaticky uveden do blokovacího stavu. Zjistil jsem, že toto chování působilo nastavení přepínačů Cisco, které jsou v hierarchii sítě nad naším přepínačem Cisco.

Pro uvedení do funkčního stavu, bylo nezbytné pomocí CNA (Cisco Network Assistant) provést nastavení operačního módu portu (poskytujícího připojení do školní sítě) na 802.1Q Trunk – Nonnegotiate, nebo na 802.1Q Trunk.

Trunk označuje trunking protokol, a tímto označením se říká, že daný port je členem všech virtuálních sítí (VLAN) a komunikace z těchto VLAN je přenášena prostřednictvím tohoto portu. Každá VLAN komunikuje se školní sítí prostřednictvím portu označeného Trunk.

Pokud nastavíme operační mód 802.1Q-Trunk, musíme provést nastavení VTP (VLAN Trunking protocol) do transparentního módu. Jinak by komunikace nefungovala. VTP totiž umožňuje provádět centrální nastavení VLAN a jejich následné sdílení přes trunking protokol. Používá se zejména v rozsáhlých sítích, což není náš případ. Jinými slovy, pokud necháme VTP typu server či klient, musíme uvažovat nastavení ostatních Cisco přepínačů v síti. Pokud nastavíme transparentní mód, okolní Cisco přepínače nás neovlivňují.

Pokud nastavíme operační mód 802.1Q Trunk-Nonnegotiate, nemusíme se starat o VTP, neboť při této konfiguraci je na daném rozhraní zabráněno generovat DTD (Dynamic Trunk Protocol) rámec, který se používá pro dohodnutí VTP.

Po této konfiguraci by již měli zapojené zařízení na přepínači komunikovat se školní sítí prostřednictvím trunk portu. Může nastat případ, že tomu tak nebude. Máme-li zapnutou vlastnost STP (Spanning Tree Protocol), který se používá pro odstranění smyček v síti, může nastat situace, že přepínač nad námi zašle BPDU guard rámec. Ten způsobí uvedení portu do blokovacího stavu. K tomuto dojde, pokud je na portu zapnuta vlastnost Port Fast. Port Fast způsobí, že port je automaticky po připojení uveden do „forwarding“ (přeposílacího) stavu. Vynechá se stav naslouchání a učení, které se používají pro správné nastavení STP. Okamžité uvedení do „forwarding“ stavu způsobí, že se trunk port nastaví jako root (hlavní) port. STP přepínačů v hierarchii nad naším Cisco přepínačem rozpozná, že přepínač pod ním si neoprávněně přivlastnil označení root a odešle BPDU (Bridge Protocol Data Units) guard rámec.

Protože školní síť neobsahuje smyčky, doporučuji vypnout STP pro porty poskytující připojení do školní sítě, nebo vypnout vlastnost Port Fast. Na ostatních portech můžeme STP rovněž vypnout, neboť jsou na ně zapojeny moduly automatů a nehrozí vznik smyček, nestabilit.

Podrobnější popis vlastností a vysvětlení pojmů naleznete v části Příloha E.

5.1.2.1 Nastavení synchronizace času pomocí NTP

Časová synchronizace probíhá pomocí NTP (Network Time Protocol) protokolu. Aby přepínač podporoval tento protokol, musí se nakonfigurovat globální parametry NTP na přepínači.

Nastavení globálních parametrů:

5. `CiscoK023 # configure terminal`
6. `CiscoK023(config)# ntp server 207.46.130.100 prefer`
   // nastavení synchonizačního serveru jako preferovaného zdroje
7. `CiscoK023(config)# ntp peer 147.32.87.157`
   // nastavení dalšího zdroje pro synchronizaci času, tentokrát počítač v laboratoři
8. `CiscoK023(config)# end`
9. `CiscoK023 # copy runnig-config startup-config`

5.1.2.2 Nastavení DHCP snooping

Používá se jako ochrana před útočníkovým DHCP serverem. Princip funkce spočívá v rozdělení portů na přepínači na důvěryhodné (trusted) a nedůvěryhodné (untrusted). Pokud je na port připojen počítač nebo komunikační modul automatu, nastavíme port jako nedůvěryhodný. Pokud je na port připojen DHCP server nebo další přepínač, port označíme jako důvěryhodný (pokud je na port připojen přepínač, který nepodporuje DCHP snooping, musíme port nastavit jako nedůvěryhodný).

Pokud nějaký klient požádá o přidělení IP adresy útočníkův DHCP server, tento server sice odpoví, ale při příchodu odpovědi na přepínač proběhne kontrola, zda je port důvěryhodný. Pokud přepínač zjistí, že odpověď byla odeslána z nedůvěryhodného portu, tato odpověď je zahozena.

DHCP snooping vytváří tabulku, která obsahuje MAC adresu, IP adresu a port klienta, který získal údaje z DHCP serveru. V této tabulce je rovněž uvedena zbývající doba zapůjčení IP adresy a příslušnictví do VLAN.

Nastavení DHCP snooping:

1. `CiscoK023# configure terminal`
2. `CiscoK023(config)# ip dhcp snooping`
   // globální povolení DHCP snooping
3. `CiscoK023(config)# ip dhcp snooping vlan 5`
   // nastavení snooping pro VLAN 5
4. `CiscoK023(config)# interface range fastethernet0/2-24`
   // nastavení rozsahu rozhraní, pro které budeme provádět konfiguraci
5. `CiscoK023(config-if-range)# no ip dhcp snooping trust`
   // nedůvěryhodné porty 2-24
6. `CiscoK023(config-if-range)# exit`
7. `CiscoK023(config)# interface fastethernet0/1`
   // konfigurace pro jedno konkrétní rozhraní
8. `CiscoK023(config-if) # ip dhcp snooping trust`
   // nastavení rozhraní 1 jako důvěryhodné
9. `CiscoK023(config)# ip dhcp snooping verify mac-address`
   // ověří, že zdroj. MAC adresa odpovídá adrese obsažené v paketu
10. `CiscoK023(config)# exit`
11. `CiscoK023# copy runnig-config startup-config`

DHCP snooping zabezpečení konfigurujeme pro případ, aby uživatel, jehož osobní počítač je vybaven DCHP serverem, nezpůsobil přidělení nesprávných IP adres komunikačním modulům automatů. Tím by zapříčinil nefunkčnost programů RSLinx a RSLogix 5000 používaných pro výuku.

5.1.2.3 IGMP protocol

Základem pro IGMP (Internet Group Management Protocol) byl Host Membership Protokol navržený Steve Deering. Tento protokol používají multicastoví klienti, aby signalizovali směrovačům, přepínačům, že chtějí odebírat určitou multicastovou skupinu. Existují tři verze protokolu IGMPv1, IGMPv2, IGMPv3. Komunikační moduly Allen-Bradley podporují verze v1 a v2.

5.1.2.3.1 IGMPv1

Je vybaven pouze dvěma druhy IGMP zpráv, dotazem (Membership Query) a odpovědí (Membership Report). Zdroj dotazů, obvykle směrovač nebo přepínač odesílá pravidelně Membership Query na adresu 224.0.0.1 (přijmou ji všechna zařízení lokální sítě). Zařízení, která chtějí odebírat nějakou multicastovou skupinu, zašlou odpověď Membership Report. Tato zpráva obsahuje adresu multicastové skupiny, kterou chce zařízení odebírat.

5.1.2.3.2 IGMPv2

Vychází z předchozí verze, ale byla doplněna o další zprávu, která umožňuje snížit zatížení sítě. Je to zpráva odhlášení ze skupiny Leave Group. Jakmile se host skupiny přestane zajímat o přijímání skupiny, zašle odhlašovací zprávu. Při příchodu této zprávy směrovač nebo přepínač zjistí vysláním Membership Query, zda existuje ještě nějaký host, který má o danou skupinu zájem. Pokud nikdo zájem nemá, zruší přeposílání celé skupiny. V opačném případě odebere pouze zařízení, které poslalo odhlašovací zprávu.

5.1.2.3.3 IGMP snooping

Pomáhá omezit multicastovou komunikaci na jednotlivých portech přepínače. Zapnutím IGMP snooping řekneme přepínači, ať přeposílá určitou skupinu multicastu pouze na rozhraní asociované s touto skupinou. Aby přepínač dokázal směrovat určitou multicastovou skupinu na správný port, udržuje si směrovací tabulku. Tuto tabulku vytváří z IGMP zpráv, které chodí mezi směrovačem resp. přepínačem a zařízeními. Směrovač odesílá IGMP zprávu (Membership Query), na kterou automaty zařízení odpovídají IGMP zprávou obsahující IP adresu skupiny multicastu, o kterou žádají.

Tuto odpověď směrovač, resp. přepínač přečte a do směrovací tabulky uloží číslo portu a skupinu mutlicastu. Následně je přepínačem mutlicastová skupina poskytována asociovanému zařízení, dokud neobdrží Leave Group zprávu. Po přijetí odhlašovací zprávy přepínač odstraní patřičný záznam ze své směrovací tabulky. Směrovací tabulka přepínače je periodicky aktualizována a v případě, že zařízení neodpoví na periodickou IGMP

Membership Query ze směrovače či přepínače, je zařízení resp. odpovídající položka vymazána ze směrovací tabulky a multicastová skupina již není tomuto zařízení poskytována.

Dvě možnosti jak nastavit IGMP snooping:

a) staticky

V tomto případě se konfigurují multicastové skupiny manuálně a musí se přiřadit jednotlivým VLAN nebo konkrétním rozhraním. To je značně nepraktické. Při změnách IP adres nebo při změnách portů, do kterých jsou zařízení zapojeny, musíme provést překonfigurování staticky definovaných skupin.

Příklad konfigurace

```
1. CiscoK023(config)# ip igmp snooping vlan 5 static 239.192.1.96
   interface FastEthernet0/5
```

/ / povolení přenosu multicastové adresy 239.192.1.96 na rozhranní FastEthernet0/5, port číslo 5

b) dynamicky

Multicastové skupiny a příjemci těchto skupin jsou naučeni podle zasílaných IGMP zpráv. Protože laboratoř není vybavena směrovačem, musel jsem nakonfigurovat přepínač, aby plnil funkci směrovače pro multicastovou komunikaci. Tuto funkci může přepínač plnit, pokud je vybaven IGMP Snooping Querier módem.

Konfigurace dynamického IGMP snooping

```
1. CiscoK023(config)# ip igmp snooping
2. CiscoK023(config)# ip igmp snooping querier
3. CiscoK023(config)# ip igmp snooping querier address 147.32.87.158
4. CiscoK023(config)# ip igmp snooping querier  query-interval 20
5. CiscoK023(config)# ip igmp snooping querier timer expiry 60
6. CiscoK023(config)# ip igmp snooping querier version 2
7. CiscoK023(config)# no ip igmp snooping querier tcn query count
8. CiscoK023(config)# no ip igmp snooping querier tcn query interval
9. CiscoK023(config)# no ip igmp snooping report-suppression
10. CiscoK023(config)# ip igmp snooping vlan 5 immediate-leave
11. CiscoK023(config)# exit
12. CiscoK023 # copy running-config startup-config
```

IGMP snooping querier

Nastaví přepínač do režimu, kdy zajišťuje odesílaní IGMP dotazů. Toto nastavení se bere v potaz, když není dostupný směrovač, který za normálních okolností zajišťuje odesílání IGMP dotazů. V našem případě směrovač není dostupný. Proto používám přepínač ke generování těchto zpráv s danou periodou a zvolenou dobou platnosti dotazu.

IGMP immediate-leave

Způsobí okamžité odstranění dané položky ze směrovací tabulky při příchodu odhlašovací zprávy na port. Přepínač neodesílá dotaz, zda existuje ještě nějaké další zařízení, které má zájem odebírat tuto multicastovou skupinu. Ponechá stávající položky v tabulce a odstraní odpovídající záznam. Pokud by byl zapojen směrovač, je vhodné tuto vlastnost vypnout na trunk portu jinak dojde k odhlášení celé skupiny odběratelů při příchodu odhlašovací zprávy od některého z odběratelů.

TCN (Topology Change Notification)

Vypneme TCN neboli nastavení parametrů vysílání multicastových skupin při změně topologie sítě. Pro náš případ zapojení automat-přepínač-automat tato vlastnost nemá smysl. Pokud bychom nechali TCN zapnuté, přepínač by posílal data na port, i v případě odpojení či přepojení automatu, což by zbytečné. Tuto vlastnost má smysl využívat v případě stromové struktury zapojení více přepínačů nad sebou. V našem případě bychom jen zbytečně nutili přepínač posílat data na port, který již není využíván. Navíc se při komunikaci mezi automaty (producent/konzument) využívá potvrzování přijatých zpráv a v případě nedoručení potvrzení se spouští čítač timeout. Po překročení doby timeout je spojení uzavřeno. Proto nemá smysl povolovat TCN.

Report-Suppression

Zabraňuje násobnému odeslání IGMP žádosti o začlenění do skupiny. Tato vlastnost je užitečná, pokud máme v síti multicastový směrovač, který se stará o odesílání IGMP dotazů. Přepínač, pokud je vlastnost zapnuta, při přijetí přihlašovacích zpráv do stejné multicastové skupiny z více portů odešle přepínač směrovači přihlašovací zprávu pouze jednou. Pokud je tato vlastnost vypnuta, přepínač odešle zprávu směrovači tolikrát, kolikrát ji přijme.

5.2 Implicitní komunikace

Implicitní komunikace se používá pro zasílání periodických zpráv o stavu modulů zapojených v automatu nebo o stavu proměnných v procesoru. Po nastavení a vytvoření komunikace manažerem nespojovaných zpráv předává řízení komunikace manažeru spojovaných zpráv. Mezi těmito manažery je zásadní rozdíl ve způsobu komunikace a ve využívaní systémových prostředků.

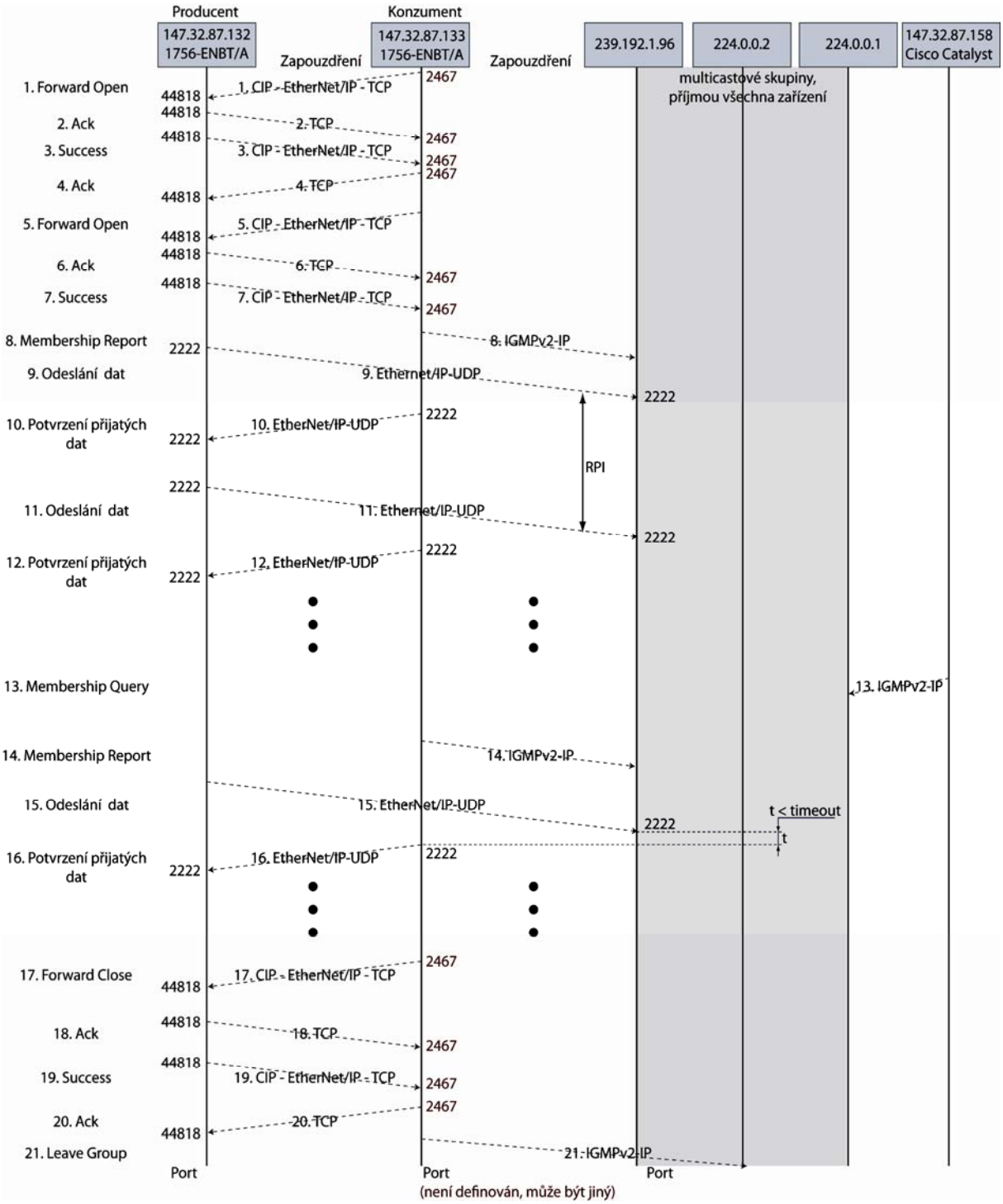
Manažer nespojovaných zpráv potřebuje, aby CIP zpráva obsahovala informaci o zdroji a cíli. Tento manažer pro zpracování zpráv rezervuje opětovně systémové prostředky nutné pro vykonání příkazu obsaženého ve zprávě. Po vykonání příkazu jsou rezervované prostředky uvolněny.

Manažer spojovaných zpráv ve zprávě potřebuje pouze identifikátor spojení. Tento manažer má rezervované systémové prostředky po celou dobu komunikace. To mu umožňuje zajistit včasné zpracování požadavku a vygenerování odezvy.

5.2.1 Navázání a ukončení komunikace typu producent/konzument

Konzument začíná komunikaci odesláním nespojované zprávy, pomocí níž přiměje producenta otevřít rozhraní komunikující po sběrnici ControlBus. Po této sběrnici se přenášejí tagy z procesoru do komunikačního zařízení (1756-ENBT/A). O uspěšném vykonání, producent informuje konzumenta, který mu v zápětí zašle další požadavek na otevření komunikace. Tentokrát však konzument vyžaduje po producentovi otevřít rozhraní

komunikující prostřednictvím Ethernetu. O úspěšném otevření rozhraní producent opět informuje konzumenta.



Obr. 5.2 Komunikace mezi moduly při implicitním spojení typu producent/konzument

Obě zařízení, které doposud komunikovaly prostřednictvím manažera nespojovaných zpráv, předávají řízení komunikace manažeru spojovaných zpráv. Tento manažer obdrží identifikátor spojení, který popisuje vytvořené spojení mezi producentem a konzumentem s požadovanými parametry. Nyní již producent začíná vysílat na rozhraní Ethernet získané data z procesoru přes rozhraní ControlBus.

Každý tag je vysílán prostřednictvím jedné multicastové skupiny. Konzument po úspěšném navázaní komunikace odesílá IGMP zprávu, kterou žádá směrovač resp. přepínač o přeposílání určité konkrétní multicastové skupiny. Způsob jak vytvořit komunikaci typu producent/konzument je detailně popsáno v Příloha A.2.

Slovní popis předchozích odstavců je graficky znázorněn v Obr. 5.2, který byl vytvořen podle dat odposlechnutých pomocí programu Wireshark a význam dílčích zpráv určen pomocí specifikace CIP protokolu [6].

Vysvětlení důležitých položek komunikace z Obr. 5.2:

1. Forvard Open

   CIP zpráva zapouzdřená v EtherNet/IP paketu odeslaná prostřednictvím TCP protokolu a využitím nespojovaného typu komunikace. Paket obsahuje zapouzdřené záhlaví a zapouzdřená data (CPF). Zapouzdřené záhlaví obsahuje příkaz SendRRdata a další nezbytné parametry definované podle Obr. 2.15. Zapouzdřená data, mimo jiné, obsahují položku Interface Handle, která určuje rozhraní komunikující pomocí sběrnice ControlBus. Vlastní zapouzdřená data obsahují požadovanou službu Forvard Open a další nezbytné parametry pro vytvoření spojení. Například se jedná o Timeout, RPI, Trigger Type, Electronic Keying a prioritu zprávy. Timeout určuje dobu, kterou se čeká na odpověď. RPI určuje interval, ve kterém získáme data (pro ControlBus je RPI vždy 0ms). Trigger Type určuje jakým způsobem budou data přenášena (pravidelně, při změně atd.).
2. ACK (Acknowledge)

   Potvrzovací zpráva zaslaná prostřednictvím TCP. Informuje o přijmutí zprávy 1.
3. Success

   CIP zpráva zapouzdřená v EtherNet/IP paketu odeslaná prostřednictvím TCP. Zapouzdřené záhlaví obsahuje příkaz SendRRdata. Tímto příkazem se provádí odeslání odpovědi o vykonání služby Forvard Open, podle zaslaných parametrů z bodu 1.
5. Forvard Open

   Tato zpráva obsahuje stejné údaje jako zpráva v bodě 1. Rozdíl je u Inerface Handle. Tentokrát nastavujeme komunikaci pro rozhraní komunikující přes Ethernet. Rovněž je zde jiné RPI. V tomto případě se použije RPI, jaké jsme zadali při konfigurování komunikačního modulu resp. při nastavení RPI u konzumovaného tagu.
6. ACK (Acknowledge)

   Potvrzovací zpráva zaslaná prostřednictvím TCP. Informuje o přijetí zprávy z bodu 5.
7. Success

   Stejná jako v bodě 3, jen s tím rozdílem, že potvrzuje úspěšné nastavení a otevření komunikace na rozhranní komunikující prostřednictvím Ethernet.

8. Membership Report

   Konzument informuje směrovací zařízení o tom, že chce odebírat danou multicastovou skupinu. Směrovač v našem případě přepínač, si zapamatuje port, ze kterého přišla tato zpráva a skupinu, o kterou žádá. Danou skupinu následně přeposílá na asociovaný port.

9. Odeslání dat

   Odeslání dat probíhá prostřednictvím UDP protokolu. Formát paketu je popsán v Obr. 2.16. Komunikace je řízena manažerem spojovaných zpráv. Identifikace spojení probíhá pomocí identifikátoru spojení (Connection ID). Paket obsahuje I/O data, které poskytuje producent. V datové části paketu je obsažen binárně reprezentovaný produkovaný tag.

10. Potvrzení přijatých dat

    Potvrzování probíhá prostřednictvím UDP paketů. Paket obsahuje identifikátor spojení a potvrzení dat.

13. Membership Query

    IGMPv2 dotaz zaslaný na multicastovou adresu 224.0.0.1. Tato adresa podle organizace IANA označuje všechna zařízení na podsíti. Zprávu tedy přijmou všechna zařízení podsítě. Zprávu generuje přepínač, aby si aktualizoval IGMP snooping tabulku odběratelů multicastových skupin.

14. Membership Report

    IGMPv2 zpráva zaslaná na multicastovou adresu 239.192.1.96. Podle IANA se jedná o adresu přiřazenou pro použití jednotlivými organizacemi (Organization-Local Scope). Přepínač zprávu přečte, zapamatuje si multicastovou skupinu a port, ze kterého zpráva přišla. Podle těchto informací následně distribuuje multicastovou skupinu na patřičný port.

17. Forvard Close

    CIP zpráva zapouzdřená v EtherNet/IP paketu zaslaná prostřednictvím TCP. Obsahuje Interface Handle, který určuje rozhraní komunikující přes ControlBus. Hlavička obsahuje příkaz SendRRdata. Zapouzdřená data obsahují službu Forvard Close, kterou se žádá o ukončení spojení.

20. Leave Group

    IGMPv2 zpráva, oznamující, že dané zařízení již nemá zájem odebírat konkrétní multicastovou skupinu obsaženou v této zprávě. Zpráva se zasílá na multicastovou adresu 224.0.0.2. Tato adresa podle IANA označuje všechny směrovače na dané podsíti. V našem případě tutu zprávu zpracuje přepínač a provede úpravu snooping tabulky.

5.2.2 Způsob navázaní dalšího konzumenta při komunikaci typu producent/konzument

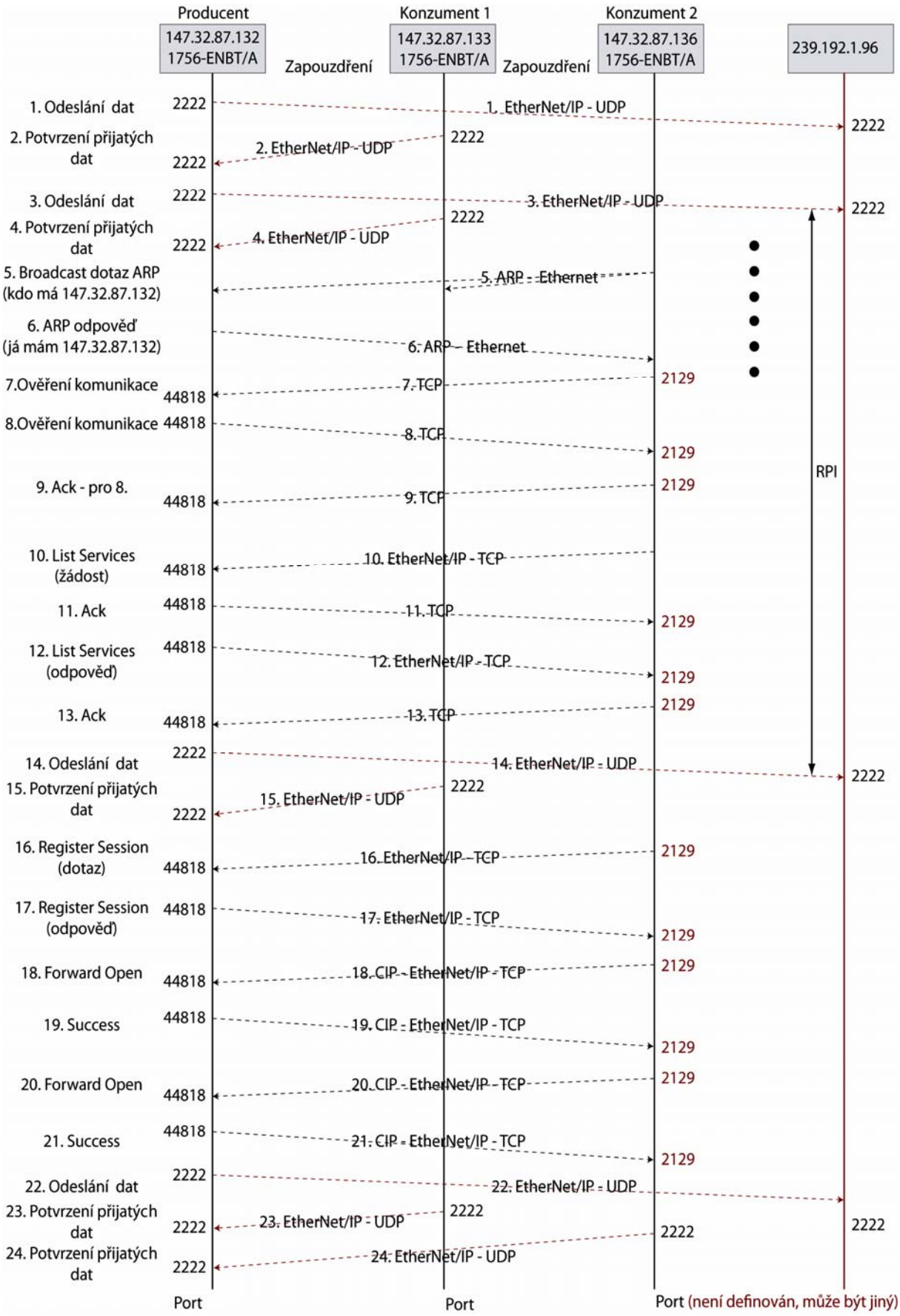
Na síti již probíhá komunikace mezi jedním producentem a jedním komzumentem vytvořená podle Obr. 5.2. Producent (147.32.87.132) odesílá data jako multicastovou skupinu 239.192.1.96. Konzument o tuto skupinu požádá přepínač pomocí IGMP zprávy a přepínač mu ji následně poskytuje. Připojení dalšího konzumenta k již vytvořené komunikaci producent/konzument probíhá následovně.

Nový konzument zasílá ARP (Address Resolution Protocol) broadcast dotaz. Tento dotaz přijmou všechna zařízení podsítě. Na dotaz odpovídá zařízení, které má odpovídající IP adresu definovou v ARP dotazu. Takto proběhne přiřazení MAC adresy k IP adrese a zařízení nyní již ví, s kým bude komunikovat. Pro jistotu si však ověří získané informace prostřednictvím TCP.

Po ověření komunikace konzument zasílá dotaz na seznam poskytovaných služeb producentem. Producent na tento dotaz odpovídá seznamem poskytovaných služeb. Nový konzument se dozví, že producent poskytuje jeden tag jednomu konzumentovi. Nový konzument, protože chce odebírat stejný tag jako původní konzument, musí provést zaregistrování do relace, která je vytvořená u producenta pro původního konzumenta. Nový konzument zasílá požadavek na zaregistrování do relace. Producent vyhodnotí možnost registrace podle počtu povolených konzumentů při vytváření produkovaného tagu. Pokud není vyčerpán maximální povolený počet konzumentů na daný tag, producent kontumentovi zašle zprávu o úspěšné registraci k relaci.

Následně nový konzument žádá producenta o otevření spojení přes ControlBus k procesoru a o otevření spojení na rozhraní Ethernet. Mohlo by se zdát, že tento požadavek na otevření spojení je zbytečný, když to již zajistil původní konzument. Opak je však pravdou, protože konzumenti mohou mít nastaveno rozdílné RPI (Requested Packet Interval - požadovaný interval odesílání paketů) a pak by otevřené spojení neplnilo správně funkci pro oba konzumenty. Proto se otvírá spojení pro oba konzumenty, přičemž data oběma konzumentům poskytuje pouze spojení od konzumenta s menším intervalem RPI. Volba využívaného spojení, je závislá na způsobu komunikace. Není podmínkou, že se vždy volí spojení s nejmenším RPI viz kapitola 5.3. Každý konzument potvrzuje přijaté data ve svém RPI, ať už jsou data vysílány s jakýmkoliv RPI.

Samozřejmě nově připojený konzument musí opět požádat přepínač o zasílání multicastové skupiny prostřednictvím IGMP zprávy. Kvůli přehlednosti IGMP není v Obr. 5.3 zobrazeno, ale způsob odesílání je naprosto totožný s Obr. 5.2 (slovní popis předchozích odstavců je graficky znázorněn v Obr. 5.3).



Obr. 5.3 Připojení dalšího konzumenta

Vysvětlení důležitých položek komunikace z Obr. 5.3:

5. Broadcast dotaz ARP

   ARP (Address Resolution Protocol) se používá pro zjištění a přiřazení IP adresy k MAC adrese. Tuto zprávu vysílá nový konzument, který chce začít komunikaci s producentem. Vyšle broadcast ARP dotaz, na který mu odpovídá zařízení s IP adresou obsaženou v dotazu.

6. ARP odpověď

   Zařízení, které má IP adresu odpovídající IP adrese v ARP dotazu, generuje odpověď, kterou odešle zařízení, které dotaz zaslalo.

7. Ověření komunikace

   Po příchodu ARP odpovědi, zařízení zkouší komunikaci prostřednictvím TCP protokolu.

10. List services (žádost)

    Používá se ke zjištění, které služby poskytuje cílové zařízení (producent).

11. ACK (acknowledge)

    Potvrzení přijetí paketu obsahující žádost List services.

12. List services (odpověď)

    Zařízení produkující data, pošle seznam služeb, které nyní poskytuje.

13. ACK (acknowledge)

    Potvrzení přijetí odpovědi List services (odpověď).

16. Register Session (dotaz)

    Dotaz na zaregistrování relace a povolení přístupu k produkovaným datům.

17. Register Session (odpověď)

    Zařízení, které bylo dotazováno, vrací identifikátor relace v případě úspěšného zaregistrování relace.

18. Forward Open

    Otevření spojení přes rozhraní ControlBus.

19. Success

    Potvrzení navázaní spojení přes rozhraní ControlBus.

20. Forward Open

    Otevření spojení se zvoleným RPI na rozhraní Ethernet.

21. Success

    Potvrzení navázaní spojení přes rozhraní Ethernet.

22. Odeslání dat

23. Potvrzení přijatých dat

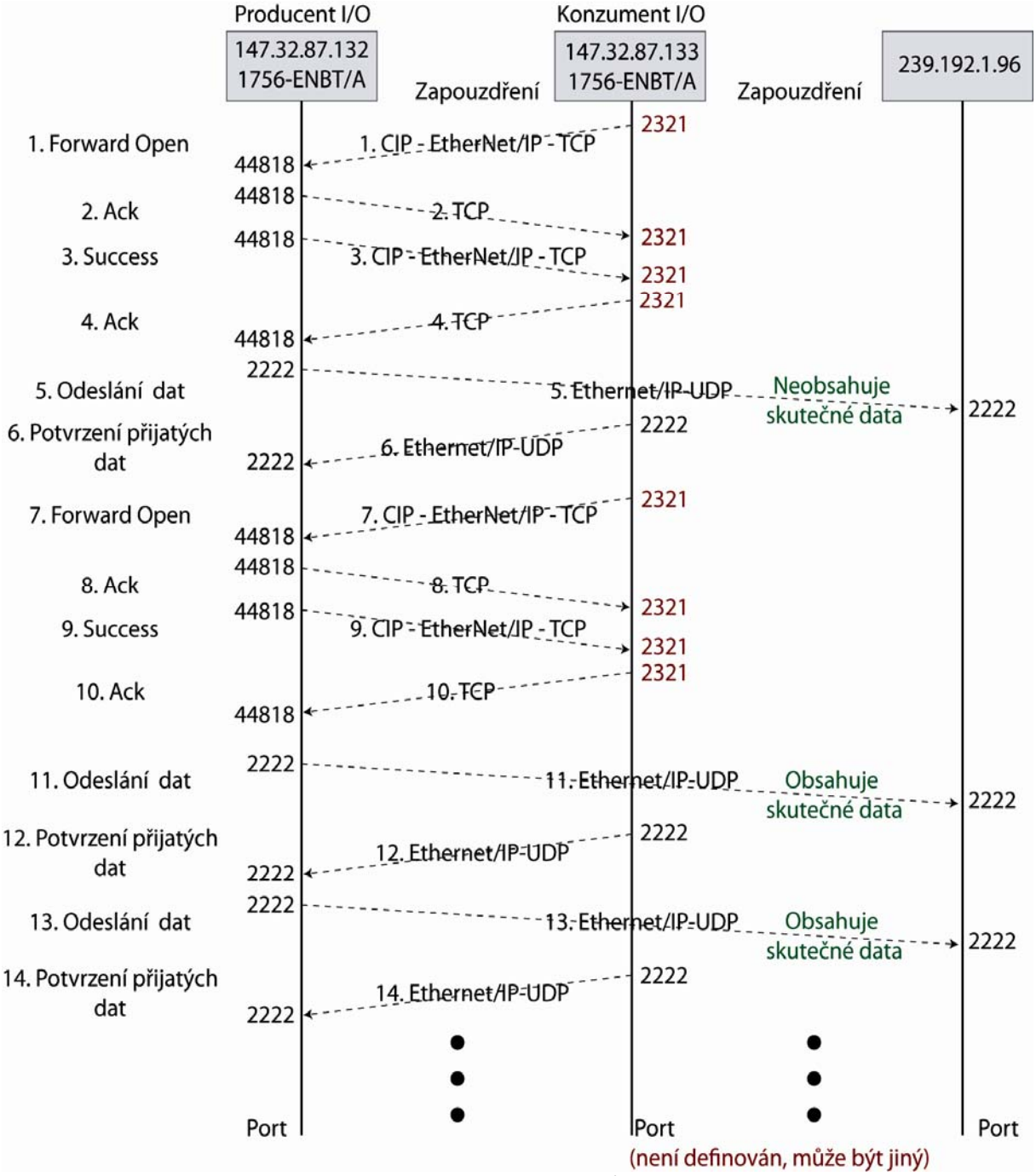
    Potvrzení přijatí dat od konzumenta jedna.

24. Potvrzení přijatých dat

    Potvrzení přijatí dat od konzumenta dva.

5.2.3 Přenášení dvou digitálních vstupů rámově optimalizovaným spojením (Rack Optimization)

Princip vytvoření komunikace je stejný jako v kapitole 5.2.1. Je zde však několik rozdílů (viz Obr. 5.4). Nejdříve se pomocí Forward Open otevírá spojení na rozhraní Ethernet. Nastavuje se interval vysílaných dat RPI podle hodnoty zadané uživatelem. Jakmile se otevře spojení, začnou se produkovat data. Tyto data však neobsahují reálné hodnoty - přenáší se data nulové hodnoty. Dalším Forward Open se otevře spojení na rozhraní ControlBus. Jakmile proběhne úspěšné otevření komunikace přes ControlBus, začnou se produkovat skutečná, reálná data.



Obr. 5.4 Rámově optimalizovaný způsob komunikace

Rámově optimalizované spojení funguje na principu vytvoření jedné multicastové skupiny, v jejímž UDP paketu se přenáší všechny digitální vstupy či výstupy modulů nakonfigurovaných pro rámově optimalizovaný přenos dat. V UDP paketu je vyhrazeno místo pro začlenění všech modulů, které mohou být zapojeny v rámu. V rámci je vyhrazeno tolik míst, jak velký je rám automatu. Data z nakonfigurovaných modulů pro rámově optimalizovanou komunikaci jsou umístěna na definovanou pozici v UDP rámci.

Pokud bychom použili neoptimalizovanou komunikaci (přímé spojení) pro přenášení digitálních I/O, pro každý modul by se použila jedna multicastová skupina. To zbytečně zvyšuje zatížení sítě.

5.2.4 Přímé spojení (Direct Connection)

Princip vytvoření komunikace je téměř totožný se způsobem uvedeným kapitole 5.2.1. Je zde prohozeno pořadí odesílání služby Forvard Open jako je tomu v situaci popsané v bodě 5.2.3. Přímé spojení se využívá pro přenášení dat z I/O modulů. Pro každý modul I/O se otevře jedna multicastová skupina, která obsahuje informace I/O.

5.3 Výpočet zatížení komunikačních modulů

CIP komunikace je dvousměrná. Dvousměrná komunikace je tvořena odeslanými daty a potvrzením o odeslaných dat. Určení minimálního počtu paketů/s vyslaných pomocí CIP, odpovídá následujícímu vzorci.

$$Pakety / s = \frac{2\,pakety}{RPI} \cdot počet_spojení \tag{5.1}$$

Podle [31] se určuje zatížení komunikačních modulů pro různé typy komunikace:

<u>Přímé spojení</u>

$$Pakety / s = \frac{2 \cdot počet_spojení}{RPI} \tag{5.2}$$

<u>Rámově optimalizované spojení</u>

$$Pakety / s = \frac{2 \cdot počet_spojení}{RPI} \tag{5.3}$$

<u>Producent/konzument</u>

Producent

$$Pakety / s = \frac{1 + počet_spojení}{RPI} \tag{5.4}$$

, pro každý produkovaný tag

Konzument

$$Pakety / s = \frac{2}{RPI} \tag{5.5}$$

, pro každý konzumovaný tag

Experimentálně jsem zjistil, že tyto rovnice mohou být použity pouze jako hrubý odhad. Nelze je použít pro určení zátěže, pro větší počet konzumentů přijímajících stejná

produkovaná data. U přímého spojení navíc není zohledněn typ digitálního modulu I/O, který má velký vliv na formát vysílání dat a značně změní zatížení komunikačních modulů. Pokud bychom navrhli složitější síť pomocí těchto rovnic, snadno dojde k překročení mezních hodnot propustnosti komunikačních modulů. Překročení mezních hodnot propustnosti způsobí poruchy při komunikace popsané v kapitole 5.5.

5.3.1 Rámově optimalizované spojení

5.3.1.1 Přenášení pouze digitálních vstupů, režim multiplekeeper

Režim multikeeper označuje násobného vlastníka modulu vstupů. Tento režim lze nastavit pouze na modulech vstupů, nikoliv na modulech výstupů. Vstupy může vlastnit více konzumentů, protože data pouze čtou.

Komunikační modul automatu, ze kterého se odesílají digitální vstupy pomocí optimalizovaného spojení (viz Obr. 5.6), je zatížen komunikací určenou jako

$$Pakety / s = \frac{1}{RPI_{vyhradní_vlastník}} + \sum_{i=1}^{počet_konzumetů} \frac{1}{RPI_i} \tag{5.6}$$

$RPI_{výhradní_vlastník}$ je RPI od konzumenta, který naváže komunikaci s producentem jako první. Konzument, který naváže komunikaci jako první, potvrzuje přijatá data v intervalu odpovídající $RPI_{výhradní_vlastník}$. Další konzument, přijímá data z producenta v intervalech $RPI_{výhradní_vlastník}$. Tyto data však nepotvrzuje v těchto intervalech, ale v intervalu $RPI_{cílové_zařízení}$, který má nastaven on. $RPI_{výhradní_vlastník}$ popisuje RPI konzumenta, který vytvoří komunikaci s producentem. $RPI_{cílové_zařízení}$ je RPI dalších konzumentů (RPI_2, RPI_3 , …).

Komunikační moduly automatů, které plní roli konzumentů jsou zatíženy komunikací určenou jako

$$Pakety / s = \frac{1}{RPI_{výhradní_vlastník}} + \frac{1}{RPI_{cílové_zařízení(konzument)}} \tag{5.7}$$

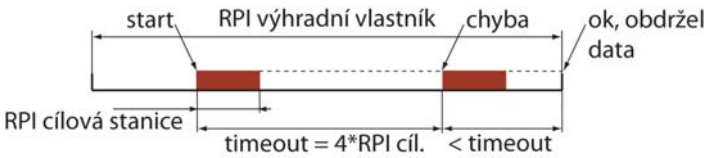
Pokud je $RPI_{výhradní_vlastník}$ menší jak $RPI_{cílové_zařízení}$ nenastane žádna chyba, i když interval $RPI_{cílové_zařízení}$ nebude násobkem dvou $RPI_{výhradní_vlastník}$. Přijaté pakety jsou u konzumentů uchovány po určitou dobu v paměti a data budou přečteny z této paměti.

Pokud však bude $RPI_{cílové_zařízení}$ menší jak $RPI_{výhradní_vlastník}$ může nastat chyba. Od okamžiku, kdy konzument začne čekat na příchod dalších dat, nesmí uběhnout delší doba jak timeout.

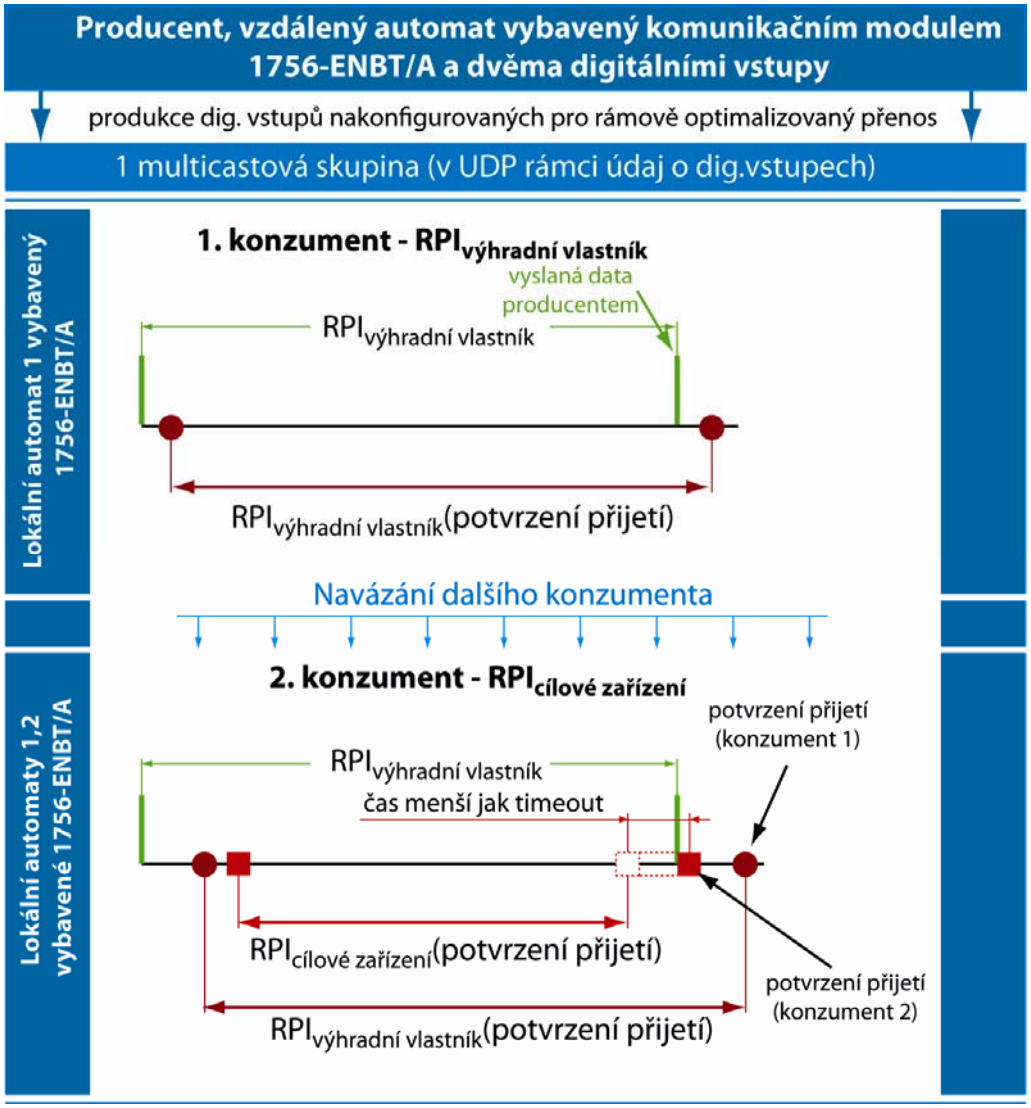
$$timeout = násobitel \cdot RPI_{cílová_zařízení} = 4 \cdot RPI_{cílová_zařízení} \tag{5.8}$$

(Násobitel je dán EDS souborem (Electronic Data Sheet) pro dané komunikační zařízení)

Pokud by paket v době timeout nepřišel, modul zahlásí chybu spojení



Obr. 5.5 Chyba spojení při nevhodně zvolených RPI

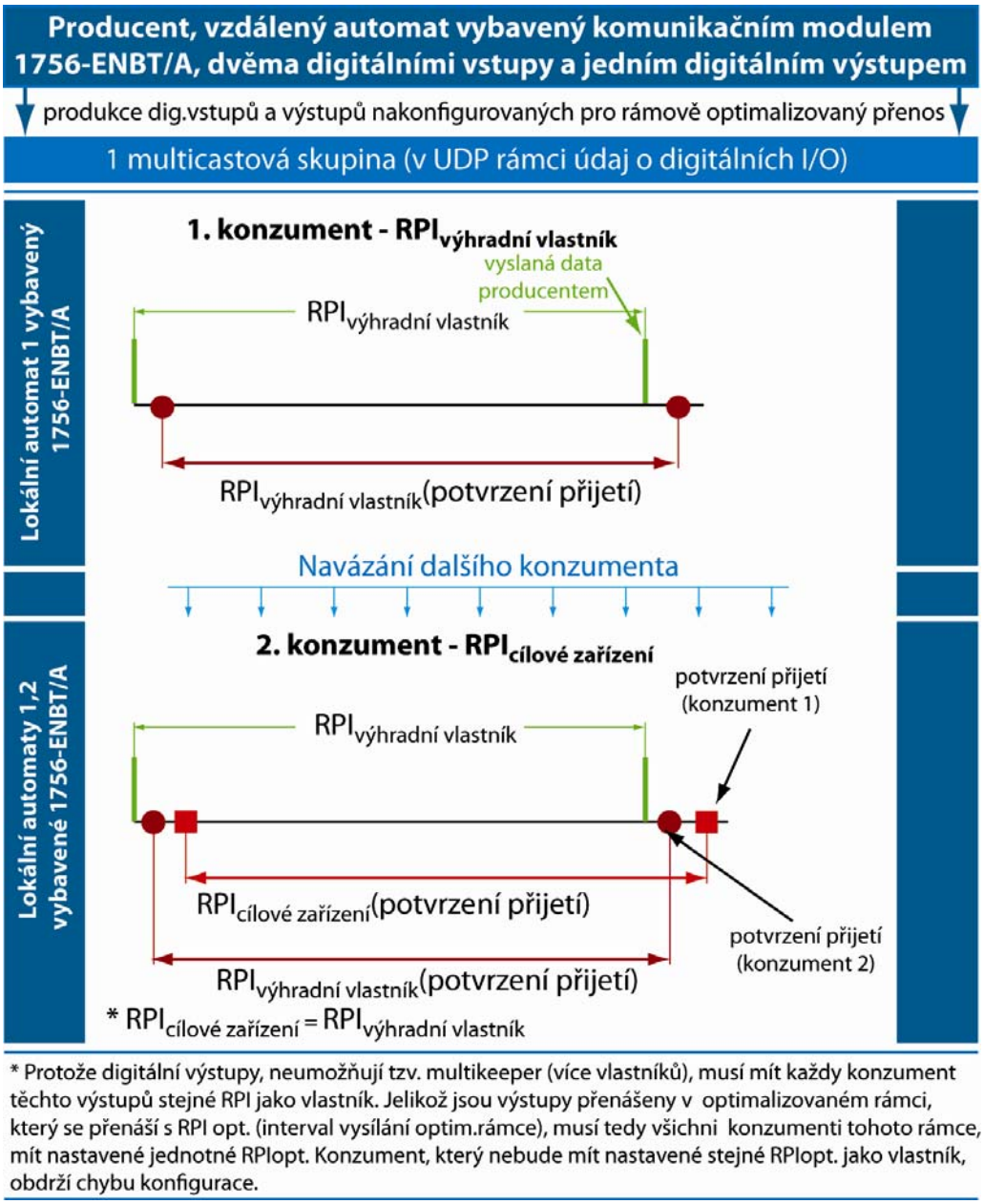


Obr. 5.6 Princip komunikace rámově optimalizovaného spojení pro přenos digitálních vstupů

5.3.1.2 Přenášení digitálních vstupů společně s digitálními výstupy

Digitální výstupy nelze vlastnit více moduly. Nelze nastavit komunikaci Rack optimization, Output data součastně u více zařízení. Tyto typy komunikace určují, kdo bude mít právo nastavovat výstupy digitálního modulu. Další konzumenti tedy mohou přistupovat k modulům pouze typem komunikace Listen Only – Rack Optimization nebo Listen Only – Output Data.

Z toho vyplývá, že je jeden vlastník (Rack Optimitaziton či Output Data), který provádní nastavení dig. výstupů a posluchači (Listen Only – Rack Optim nebo Listen Only – Output Data), kteří mohou poslouchat, co vlastník nastavuje. Posluchači musí mít nastaveno stejné RPI jako vlastník, jinak obdržíme chybu konfigurace (viz Obr. 5.7).



Obr. 5.7 Princip komunikace rámově optimalizovaného spojení při začlenění digitálních výstupů

Zatížení komunikačních modulů odpovídá podle Obr. 5.7:

Producent digitálních výstupů

$$Pakety / s = \frac{1}{RPI_{výhradní_vlastník}} + \frac{počet_spojení}{RPI_{výhradní_vlastník}} \tag{5.9}$$

Konzument digitálních výstupů

$$Pakety / s = \frac{2}{RPI_{výhradní_vlastník}} \tag{5.10}$$

Pokud provádíme změny RPI v RUN MODu automatu, musíme provést zastavení komunikace resp. modulu. Zůstává totiž uchované nastavené RPI vysílaných dat. Pouze potvrzování probíhá v nově nastaveném RPI. V případě, že tedy snížíme RPI, může se stát, že komunikace přestane fungovat. Nastane případ, že cílové zařízení neobdrží data v timeout

nového RPI, protože data jsou vysílány v původním nastaveném RPI. Jakmile však provedeme zastavení modulu, což ukončí komunikaci, tak se při spuštění modulu vytvoří nové připojení s požadovaným RPI.

Pokud provedeme zvětšení hodnoty RPI, data budou vysílána v původním nastaveném RPI. Při této změně nedojde k chybě komunikace, ale zbytečně je komunikační modul zatěžován daty v původním RPI, místo požadovaného většího RPI.

5.3.2 Přímé spojení

5.3.2.1 Určení zatížení pří přenášení digitálních vstupů 1756-IB16

Produkování digitálních vstupů modulu probíhá zcela specifickým způsobem. Způsob komunikace mezi producentem dat a konzumenty je závislá na počtu konzumentů a zvoleném RPI konzumentů (viz Obr. 5.8).

Způsob komunikace, jak je vidět z Obr. 5.8, lze rozdělit na čtyři základní typy:

1) <u>Producent a jeden konzument</u>

   Producent

$$Pakety/s = \frac{2}{RPI_{výhradní_vlastník}} + \frac{1}{RPI_{výhradní_vlastník}} \tag{5.11}$$

   Konzument

$$Pakety/s = \frac{2}{RPI_{výhradní_vlastník}} + \frac{1}{RPI_{výhradní_vlastník}} \tag{5.12}$$

2) <u>Producent a více konzumentů se stejným RPI</u>

   Producent

$$Pakety/s = \frac{3}{RPI_{výhradní_vlastník}} + \frac{počet_konzumentů}{RPI_{výhradní_vlastník}} \tag{5.13}$$

   Konzument

$$Pakety/s = \frac{3}{RPI_{výhradní_vlastník}} + \frac{1}{RPI_{konzument}} \tag{5.14}$$

3) <u>Producent a více konzumentů, konzumenti 2, 3, 4, … mají větší RPI jak konzument 1</u>

   Určení délky periodicky vysílaného intervalu

$$\begin{aligned} RPI_{cílové_zařízení} &= max\left(RPI_2, RPI_3, ..., RPI_i\right) = \\ &= max\left(n_2 \cdot RPI_{výhradní_vlastník}, ..., n_i \cdot RPI_{výhradní_vlastník}\right) \end{aligned} \tag{5.15}$$

   , kde i je index určující konzumenta a n odpovídá číslům 2,3,4,…

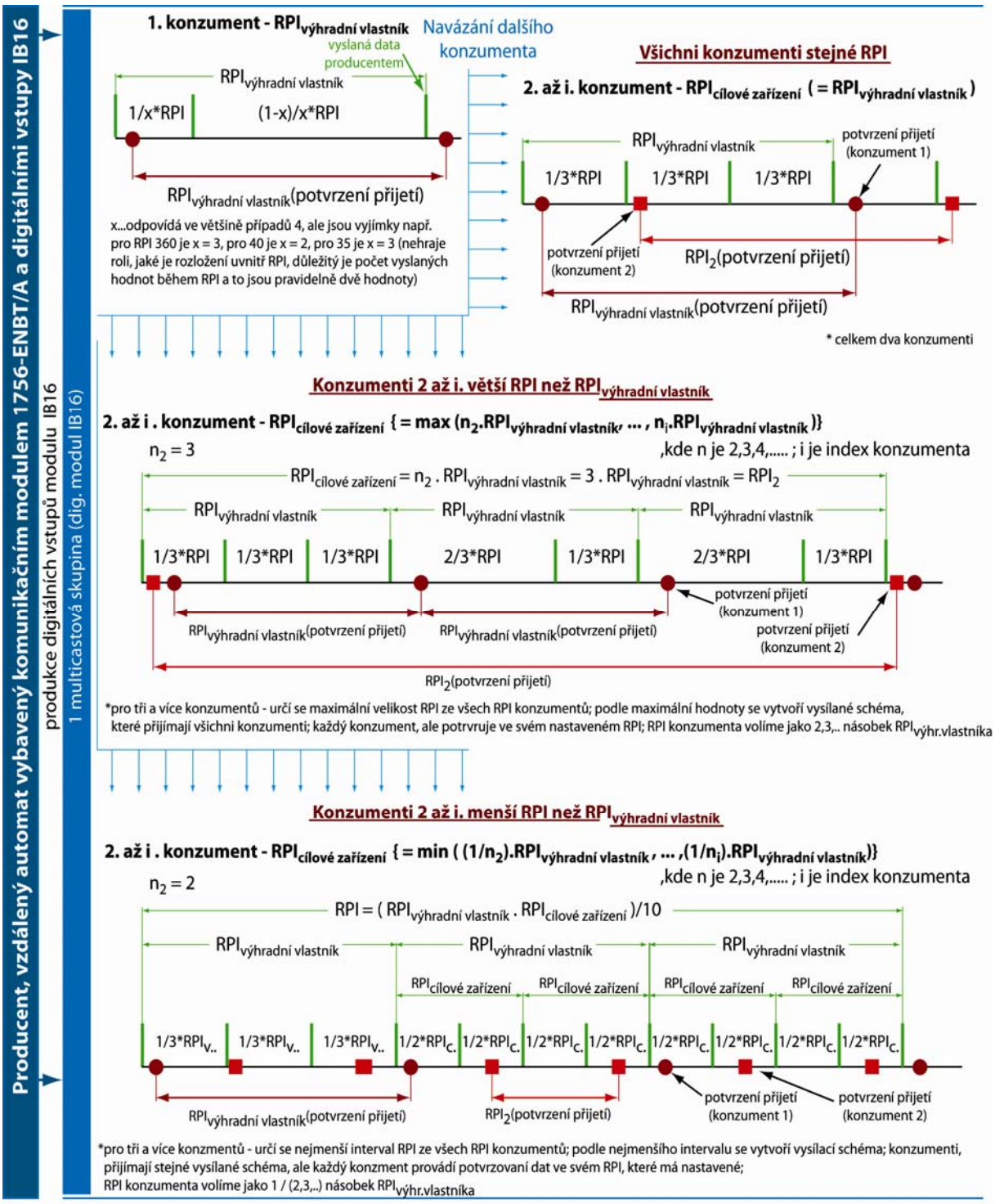
   Producent

$$Pakety/s = \frac{3 + \frac{RPI_{cílové_zařízení} - RPI_{výhradní_vlastník}}{RPI_{výhradní_vlastník}} \cdot 2}{RPI_{cílové_zařízení}} + \sum_{i=1}^{počet_konzumentů} \frac{1}{RPI_i} \tag{5.16}$$

Konzument

$$Pakety/s = \frac{3 + \frac{RPI_{cílové_zařízení} - RPI_{výhradní_vlastník}}{RPI_{výhradní_vlastník}} \cdot 2}{RPI_{cílové_zařízení}} + \frac{1}{RPI_{konzumenta}} \qquad (5.17)$$

Je vhodné dodržovat volbu RPI konzumentů 2, 3, 4, … jako násobek n RPI prvního konzumenta



Obr. 5.8 Přímé spojení (Direct Connection) a výpočet zatížení sítě při přenášení digitálních vstupů, modul 1756-IB16

4) <u>Producent a více konzumentů, konzumenti 2, 3, 4, … mají menší RPI jak kontument 1</u>

Určení délky periodicky vysílaného intervalu

$$RPI_{cilové_zařízení} = min\left(RPI_2, RPI_3, ..., RPI_i\right) = \\ = min\left(\frac{1}{n_2} \cdot RPI_{výhradní_vlastník}, ..., \frac{1}{n_i} \cdot RPI_{výhradní_vlastník}\right) \tag{5.18}$$

Producent

$$Pakety/s = \frac{3 + \left(\frac{RPI_{cilové_zařízení} \cdot RPI_{výhradní_vlastník}}{10 \cdot RPI_{výhradní_vlastník}} - 1\right) \cdot \left(\frac{RPI_{výhradní_vlastník}}{RPI_{cilové_zařízení}} \cdot 2\right)}{\frac{RPI_{cilové_zařízení} \cdot RPI_{výhradní_vlastník}}{10}} + \sum_{i=1}^{počet_konzumentů} \frac{1}{RPI_i} \tag{5.19}$$

$$Pakety/s = \frac{3 + \left(\frac{RPI_{cilové_zařízení}}{10} - 1\right) \cdot \left(\frac{RPI_{výhradní_vlastník}}{RPI_{cilové_zařízení}} \cdot 2\right)}{\frac{RPI_{cilové_zařízení} \cdot RPI_{výhradní_vlastník}}{10}} + \sum_{i=1}^{počet_konzumentů} \frac{1}{RPI_i} \tag{5.20}$$

Konzument

$$Pakety/s = \frac{3 + \left(\frac{RPI_{cilové_zařízení}}{10} - 1\right) \cdot \left(\frac{RPI_{výhradní_vlastník}}{RPI_{cilové_zařízení}} \cdot 2\right)}{\frac{RPI_{cilové_zařízení} \cdot RPI_{výhradní_vlastník}}{10}} + \frac{1}{RPI_{konzument}} \tag{5.21}$$

Je vhodné dodržovat volbu RPI konzumentů 2, 3, 4, … jako násobek *1/n* RPI prvního konzumenta.

5.3.2.2 Určení zatížení při přenášení izolovaných digitálních vstupů 1756-IB16I a digitálních výstupů 1756-OB16

Pro přenášení izolovaných digitálních vstupů a digitálních výstupů je způsob komunikace podobný typu produkování a konzumování tagů (viz 5.3.3). Je zde však odlišnost při ukončení komunikace konzumenta s nejnižším RPI. Jakmile se odpojí tento konzumet, data jsou vysílána neustále v jeho intervalu RPI a neproběhne změna intervalu odesílání dat.

Způsob komunikace, jak je vidět z Obr. 5.9, lze rozdělit na dva základní typy:

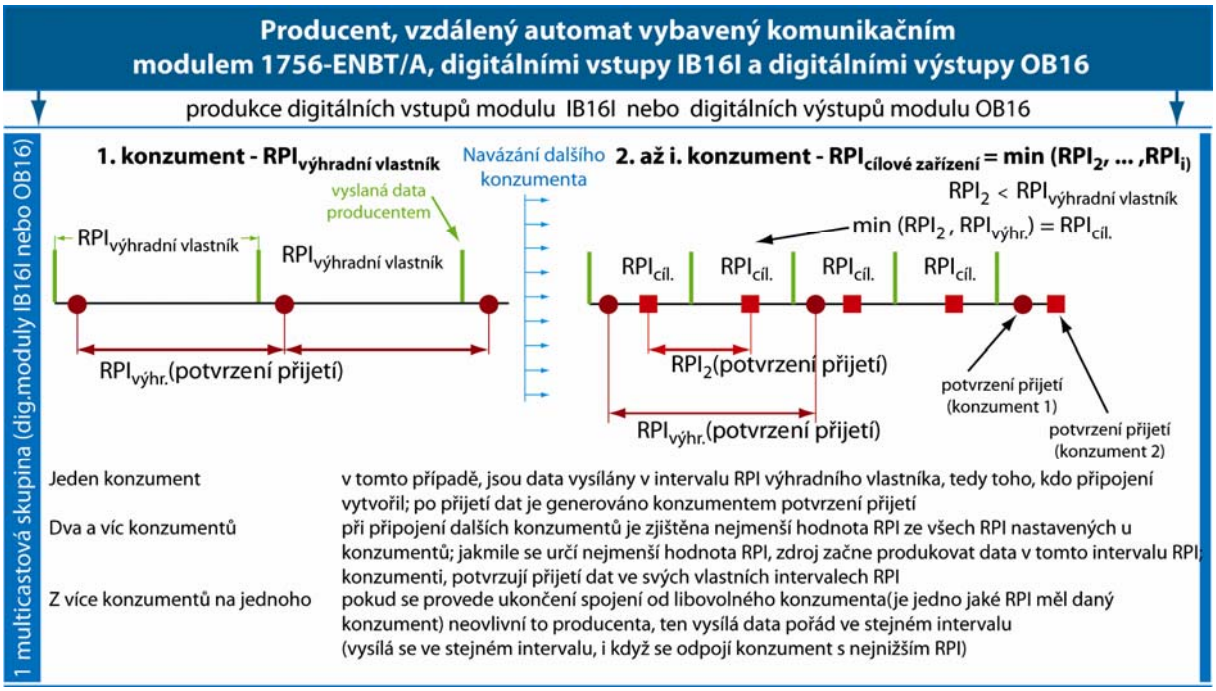
1) <u>Producent a jeden konzument</u>

Producent

$$Pakety/s = \frac{1}{RPI_{výhradní_vlastník}} + \frac{1}{RPI_{výhradní_vlastník}} \tag{5.22}$$

Konzument

$$Pakety/s = \frac{1}{RPI_{výhradní_vlastník}} + \frac{1}{RPI_{výhradní_vlastník}} \tag{5.23}$$



Obr. 5.9 Přímé spojení (Direct Connection) a výpočet zatížení sítě při přenášení izolovaných digitálních vstupů (modul 1756-IB16) a digitálních výstupů 1756-OB16

2) <u>Producent a více konzumentů</u>

Producent

$$Pakety/s = \frac{1}{min(RPI_1,..,RPI_i)} + \sum_{i=1}^{počet_konzumentů} \frac{1}{RPI_i} \tag{5.24}$$

Konzument

$$Pakety/s = \frac{1}{min(RPI_1,..,RPI_i)} + \frac{1}{RPI_{konzument}} \tag{5.25}$$

5.3.3 Produkovaní a konzumovaní tagů

Při produkování tagu se pro každý tag otevře vždy jedna multicastová skupina. Tato skupina je vysílána v intervalu nejmenšího RPI. Jakmile se přihlásí další konzument, vyhodnotí se, který z přítomných konzumentů má nejnižší interval RPI a v tom se budou data vysílat na síť. Jakmile se některý z konzumentů odhlásí, spustí se komunikace opět v intervalu nejmenší hodnoty RPI vybrané z přítomných konzumentů (Obr. 5.10).

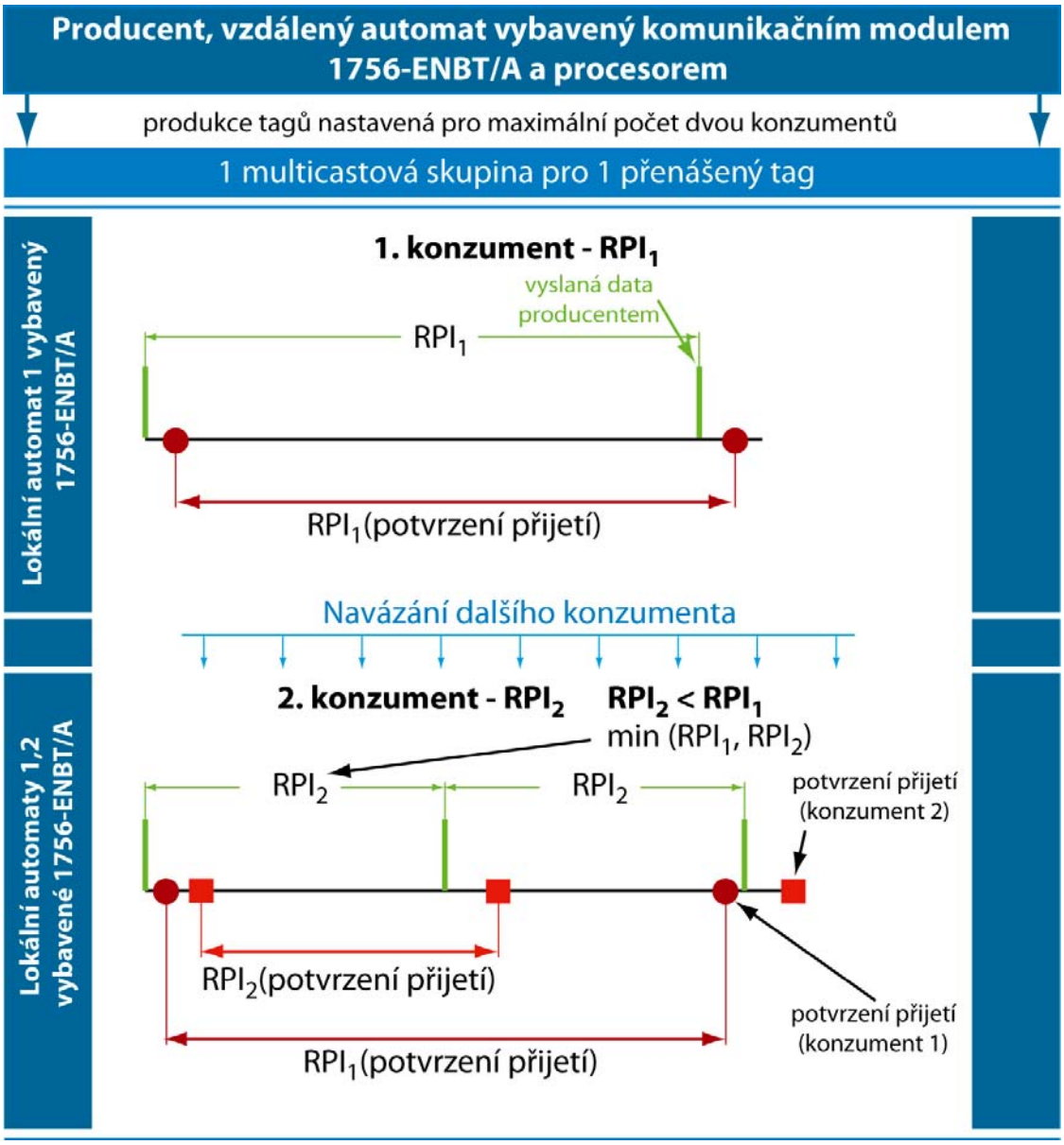
Producent produkuje jednu proměnnou, kterou odebírá více konzumentů

Producent

$$Pakety/s = \frac{1}{min\left(RPI_1,..,RPI_i\right)} + \sum_{i=1}^{počet_konzumentů} \frac{1}{RPI_i} \tag{5.26}$$

Konzument

$$Pakety/s = \frac{1}{min\left(RPI_1,..,RPI_i\right)} + \frac{1}{RPI_{konzumenta}} \tag{5.27}$$



Obr. 5.10 Princip komunikace při spojení typu Producent/Konzument

Pokud producent produkuje více proměnných a konzument přijímá více proměnných (tagů).

Producent

$$Pakety / s = \sum_{p=1}^{počet_proměnných} \left(\frac{1}{min\ \left(RPI_1,..,RPI_i\right)} + \sum_{i=1}^{počet_konzumentů} \frac{1}{RPI_i} \right) \tag{5.28}$$

Konzument

$$Pakety / s = \sum_{p=1}^{počet_proměnných} \left(\frac{1}{min\left(RPI_1,..,RPI_i\right)} + \frac{1}{RPI_{konzumenta}} \right) \tag{5.29}$$

5.4 Příklad výpočtu zatížení komunikačních modulů 1756-ENBT/A

5.4.1 Určení zatížení 1756-ENBT/A komunikačního modulu při komunikaci producent/konzument

Pro výpočet zatížení komunikačního modulu při komunikaci producent/konzument se používají následující rovnice.

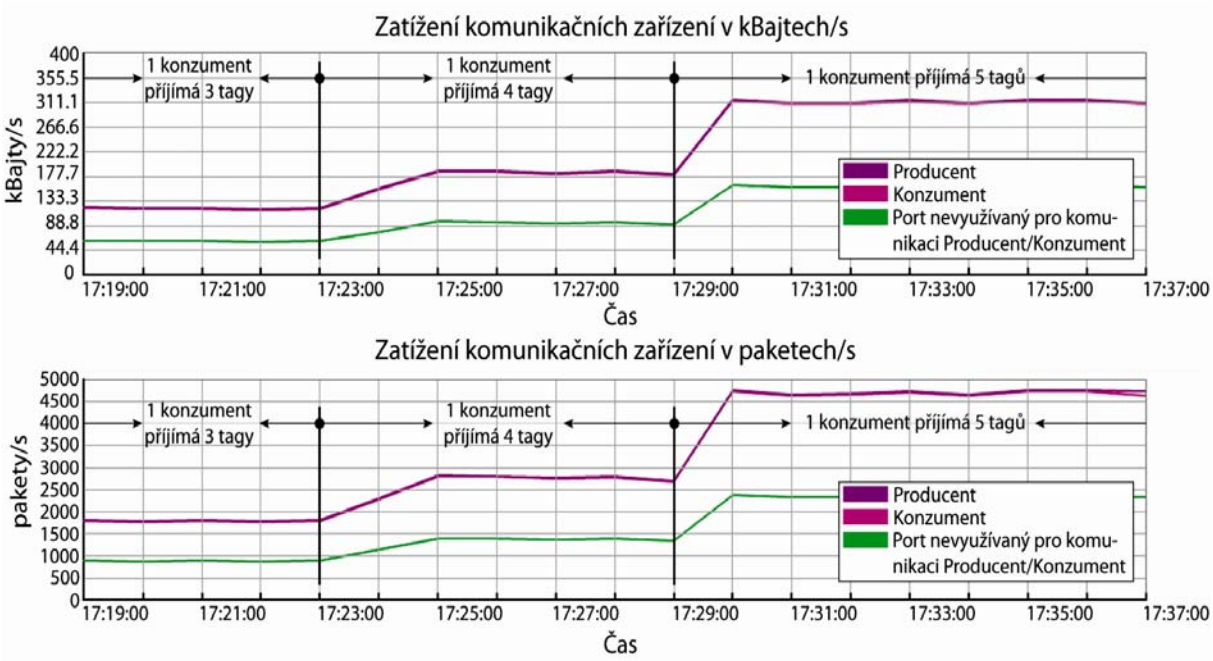
Producent

$$Pakety/s = \sum_{p=1}^{počet_proměnných} \left(\frac{1}{min\ \left(RPI_1,..,RPI_i\right)} + \sum_{i=1}^{počet_konzumentů} \frac{1}{RPI_i} \right) \tag{5.30}$$

Konzument

$$Pakety/s = \sum_{p=1}^{počet_proměnných} \left(\frac{1}{min\left(RPI_1,..,RPI_i\right)} + \frac{1}{RPI_{konzumenta}} \right) \tag{5.31}$$

Následující obrázek zobrazuje zatížení portů přenínače. Na jeho portech jsou zapojeny komunikační moduly, které plní roli producenta a konzumenta. Je zde vidět i port, na který je připojen komunikační modul, který se neučastní komunikace producent/konzument. Protože však není zapnut IGMP snooping, tento port je zatěžován vysílanými multicastovými skupinami, které přenáší hodnoty tagů. Vliv IGMP snooping je popsán v kapitole 5.6.



Obr. 5.11 Zatížení komunikačních modulů, komunikace producent/konzument. Není zapnut IGMP snooping. (výstup programu Statistika portů Cisco Catalyst 2960)

Obrázek zachycuje komunikaci mezi jedním producentem a jedním konzumentem. V průběhu měření byl měněn počet produkovaných resp. konzumovaných tagů a jejich RPI.

<u>Konzumování 3 tagů jedním konzumentem</u>

První a druhý tag byl produkován s RPI rovným 3 milisekundám. Třetí tag byl produkován s RPI rovným 4 milisekundám.

Zatížení producenta

$$Pakety/s = \left(\frac{1}{0.003} + \frac{1}{0.003}\right)_{1.tag} + \left(\frac{1}{0.003} + \frac{1}{0.003}\right)_{2.tag} + \left(\frac{1}{0.004} + \frac{1}{0.004}\right)_{3.tag} \tag{5.32}$$

$$Pakety/s = 666.66 + 666.66 + 500 = \underline{\underline{1833.32}}$$

Zatížení konzumenta

$$Pakety/s = \left(\frac{1}{0.003}+\frac{1}{0.003}\right)_{1.tag} + \left(\frac{1}{0.003}+\frac{1}{0.003}\right)_{2.tag} + \left(\frac{1}{0.004}+\frac{1}{0.004}\right)_{3.tag} \tag{5.33}$$

$$Pakety/s = 666.66 + 666.66 + 500 = \underline{\underline{1833.32}}$$

<u>Konzumování 4 tagů jedním konzumentem</u>

První a druhý tag byl produkován s RPI rovným 2 milisekundám. Třetí tag byl produkován s RPI rovným 3 milisekundám. Čtvrtý tag byl produkován s RPI rovným 11 milisekundám.

Zatížení producenta

$$Pakety/s = \left(\frac{1}{0.002}+\frac{1}{0.002}\right)_{1.tag} + \left(\frac{1}{0.002}+\frac{1}{0.002}\right)_{2.tag} + \left(\frac{1}{0.003}+\frac{1}{0.003}\right)_{3.tag} + \left(\frac{1}{0.011}+\frac{1}{0.011}\right)_{4.tag} \tag{5.34}$$

$$Pakety/s = 1000 + 1000 + 666.66 + 181.81 = \underline{\underline{2848.47}}$$

Zatížení konzumenta

$$Pakety/s = \left(\frac{1}{0.002}+\frac{1}{0.002}\right)_{1.tag} + \left(\frac{1}{0.002}+\frac{1}{0.002}\right)_{2.tag} + \left(\frac{1}{0.003}+\frac{1}{0.003}\right)_{3.tag} + \left(\frac{1}{0.011}+\frac{1}{0.011}\right)_{4.tag} \tag{5.35}$$

$$Pakety/s = 1000 + 1000 + 666.66 + 181.81 = \underline{\underline{2848.47}}$$

5.4.2 Určení zatížení 1756-ENBT/A komunikačních modulů při přímém spojení

Automat, který plní roli producenta, je vybaven digitálními vstupy 1756-IB16 a 1756-IB16I. Izolované digitální vstupy 1756-IB16I odebírá jeden konzument (konzument 1) v intervalu 2 milisekund. Neizolované digitální vstupy 1756-IB16 odebírají dva konzumenti (konzument 2 a konzument 3). Výhradní vlastník, první kdo naváže komunikaci s 1756-IB16, je konzument 2 s intervalem 10 milisekund. Druhý konzument (konzument 3), který náváže komunikaci s 1756-IB16, odebírá produkované data v dvojnásobném intervalu výhradního konzumenta (konzument 2) v intervalu 20 milisekund.

<u>Pro určení zatížení konzumenta 1 se použijí rovnice</u>

Zatížení producenta

$$Pakety/s = \frac{1}{RPI_{výhradní_vlastník}} + \frac{1}{RPI_{výhradní_vlastník}} \tag{5.36}$$

Zatížení konzumenta

$$Pakety/s = \frac{1}{RPI_{výhradní_vlastník}} + \frac{1}{RPI_{výhradní_vlastník}} \tag{5.37}$$

<u>Pro určení zatížení konzumenta 2 a 3 se použijí rovnice</u>

Určení délky periodicky vysílaného intervalu

$$\begin{aligned} RPI_{cílové_zařízení} &= max\left(RPI_2, RPI_3, ..., RPI_i\right) = \\ &= max\left(n_2 \cdot RPI_{výhradní_vlastník}, ..., n_i \cdot RPI_{výhradní_vlastník}\right) \end{aligned} \tag{5.38}$$

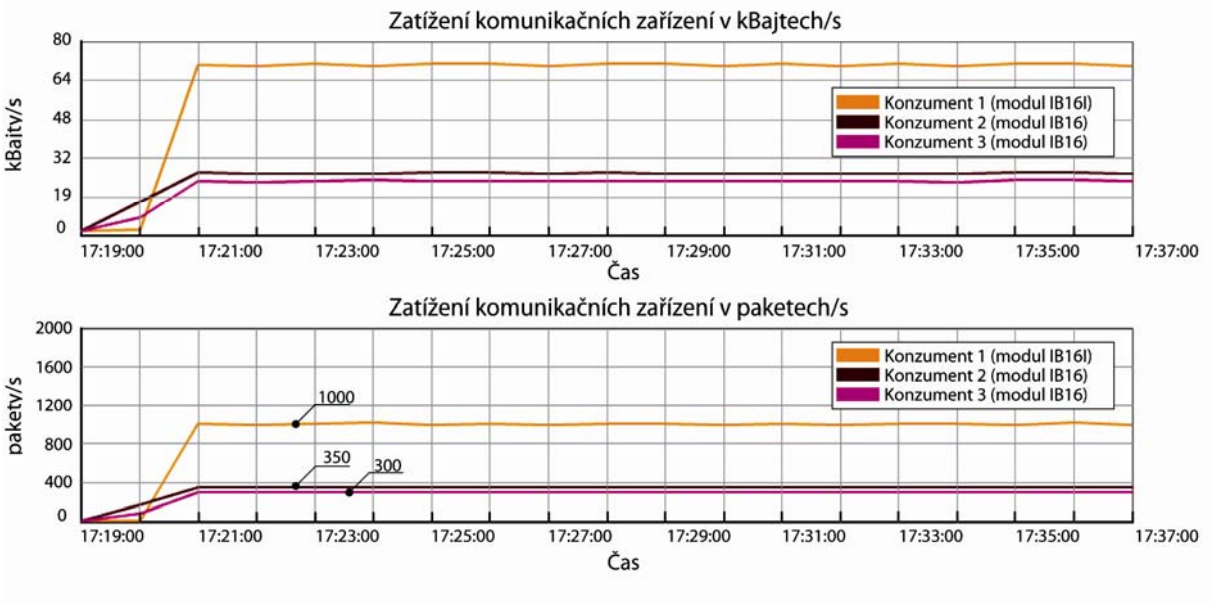
Zatížení producenta

$$Pakety/s = \frac{3 + \frac{RPI_{cílové_zařízení} - RPI_{výhradní_vlastník}}{RPI_{výhradní_vlastník}} \cdot 2}{RPI_{cílové_zařízení}} + \sum_{i=1}^{počet_konzumentů} \frac{1}{RPI_i} \tag{5.39}$$

Zatížení konzumenta

$$Pakety/s = \frac{3 + \frac{RPI_{cílové_zařízení} - RPI_{výhradní_vlastník}}{RPI_{výhradní_vlastník}} \cdot 2}{RPI_{cílové_zařízení}} + \frac{1}{RPI_{konzumenta}} \tag{5.40}$$

Následující obrázek je výstup programu na zobrazení statistik zatížení portů přepínače Cisco Catalyst 2960 doplněný o legendu. Zachycuje zatížení portů přepínače, na kterých jsou připojeny komunikační moduly plnící roli konzumentů.



Obr. 5.12 Zatížení komunikačních modulů, přímá komunikace (výstup programu Statistika portů Cisco Catalyst 2960)

<u>Výpočet zatížení pří přenášení digitálních vstupů IB16I</u>

Zatížení producenta

$$Pakety/s = \frac{1}{0.002} + \frac{1}{0.002} = 500 + 500 = \underline{\underline{1000}} \tag{5.41}$$

Zatížení konzumenta 1

$$Pakety/s = \frac{1}{0.002} + \frac{1}{0.002} = 500 + 500 = \underline{\underline{1000}} \tag{5.42}$$

<u>Výpočet zatížení při přenášení digitálních vstupů IB16</u>

Určení délky periodického vysílacího intervalu

$$RPI_{cílové_zařízení} = max(20) = \underline{\underline{20\ ms}} \tag{5.43}$$

Zatížení producenta

$$Pakety/s = \frac{3 + \frac{0.02 - 0.01}{0.01} \cdot 2}{0.02} + \left(\frac{1}{0.01} + \frac{1}{0.02} \right) = 250 + (100 + 50) = \underline{\underline{400}} \tag{5.44}$$

Zatížení konzumenta 2 (výhradní vlastník IB16 modulu)

$$Pakety/s = \frac{3 + \frac{0.02 - 0.01}{0.01} \cdot 2}{0.02} + \frac{1}{0.01} = \underline{\underline{350}} \tag{5.45}$$

Zatížení konzumenta 3

$$Pakety/s = \frac{3 + \frac{0.02 - 0.01}{0.01} \cdot 2}{0.02} + \frac{1}{0.02} = \underline{\underline{300}} \tag{5.46}$$

<u>Celkové zatížení</u>

Producent $Pakety/s = Producent_{IB161} + Producent_{IB16} = 1000 + 400 = \underline{\underline{1400}}$ (5.47)

Konzument 1 $Pakety/s = \underline{\underline{1000}}$ (5.48)

Konzument 2 $Pakety/s = \underline{\underline{350}}$ (5.49)

Konzument 3 $Pakety/s = \underline{\underline{300}}$ (5.50)

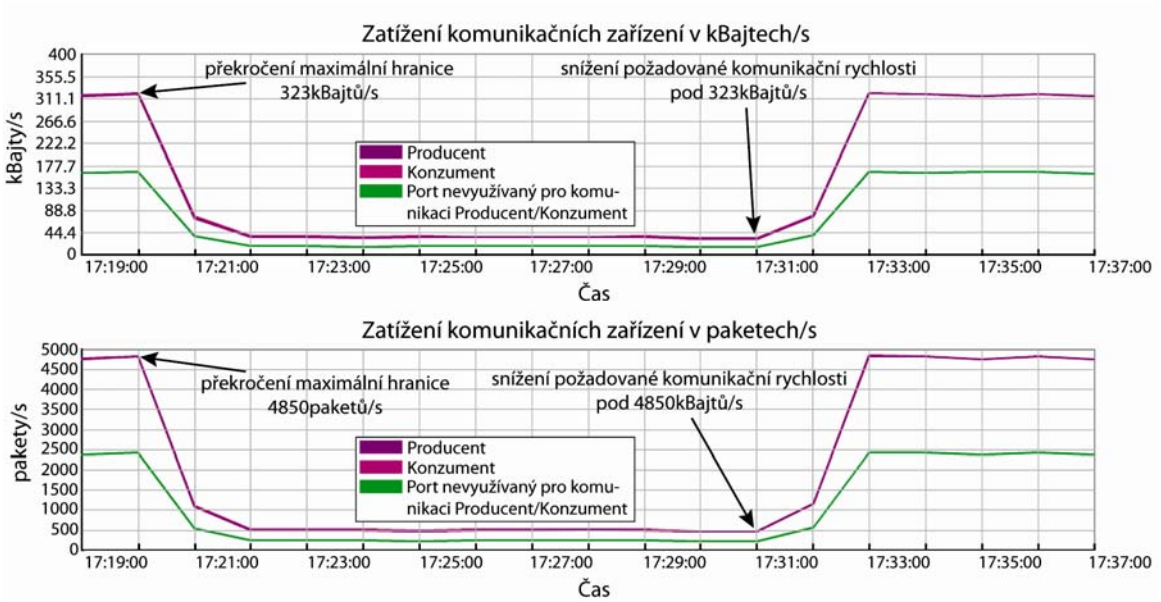
5.5 Maximální komunikační rychlost modulů 1756-ENBT/A revize 3.6

Podle manuálu [32] je schopen komunikační modul 1756-ENBT/A revize 3.6 poskytnout 5000 paketů/s, přičemž doporučované maximální zatížení je 90% z 5000 paketů/s. 10% pásma se ponechává jako rezerva pro zajištění komunikace s programy RSLinx a RSLogix 5000. V manuálu se uvádí, že pokud nezajistíme 10% komunikačního pásma pro tyto programy, nemusí se nám podařit uvést RSLogix 5000 do online řežimu (sledování průběhu vykonávání programu) a RSLinx se nemusí podařit zjistit a identifikovat zařízení zapojené přes Ethernet. I při nedodržení rezervovaného pásma pro tyto programy, se mi nepodařilo za žádných okolností nasimulovat popsanou nefunkčnost.

Experimentálně se mi podařilo maximální propustnost modulů 1756-ENBT/A s revizí 3.6 určit jako 4850 paketů/s. Pokud jsem nastavil komunikaci tak, že požadovaná komunikace překročí hranici 4850 paketů/s, dojde k výraznému poklesu rychlosti komunikace. Jakmile se překročí maximální hranice, dochází k chybám spojení vlivem překročení timeout mezi jednotlivými vyslanými daty. Při těchto chybách je spojení s daným konzumentem ukončeno. Jakmile se však spojení ukončí, konzument se pokouší opětovně navázat komunikaci. Odesílá UCCM zprávy, kterými navazuje komunikaci. Producent při příchodu těchto zpráv provede alokovaní prostředků pro implicitní komunikaci. Po alokování prostředků pro implicitní komunikaci, začne vysílat data a konzument odesílat potvrzení o doručení.

Navazování komunikace je však poměrně náročná operace, která značně zatěžuje komunikační moduly. Proto se systémové prostředky komunikačního modulu vyčerpají při

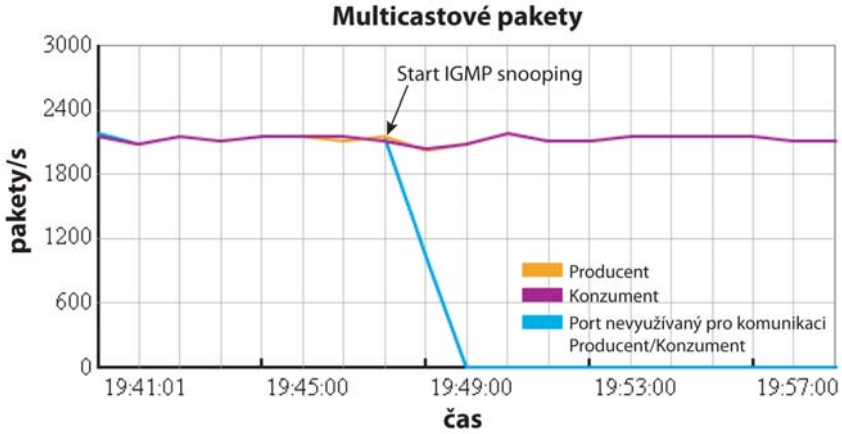
opakovaném navazování komunikace a již nezbývá dostatečné množství prostředků pro vlastní komunikaci. To vede k tomu, že při navazování komunikace pro jedno zařízení, dojde k chybě vlivem timeout u druhého zařízení atd. Toto samozřejmě rapidně sníží počet paketů vyslaných na komunikační linku, jak je vidět v Obr. 5.13.



Obr. 5.13 Překročení maximální komunikační rychlosti (výstup programu Statistika portů Cisco Catalyst 2960)

5.6 Vliv IGMP snoopingu na zatížení nevyužívaných portů přepínače při implicitní komunikaci

Zapnutí funkce IGMP snoopingu na přepínači způsobí, že multicastové skupiny budou doručovány pouze zařízením, které se o tyto skupiny přihlásí. Po zapnutí této funkce, již nebude multicastovou komunikací zatěžováno zařízení, který o data nepožádá. Vliv IGMP snooping je názorně zobrazen v Obr. 5.14. V obrázku si můžeme všimnout, že po zapnutí IGMP snoopingu, zařízení, které se neúčastní komunikace producent/konzument, přestane být zatěžováno vysílanou multicastovou skupinou sloužící pro odeslání dat. Nakonfigurování IGMP snooping viz. 5.1.2.3.3.



Obr. 5.14 Vliv IGMP snooping na zatížení portů přepínače (výstup programu Statistika portů Cisco Catalyst 2960)

6 Srovnání sítě EtherNet/IP a ControlNet

Srovnání sítí EtherNet/IP a ControlNet bylo prováděno pomocí implicitní komunikace typu producent/konzument. Přenášenými proměnnými mezi producentem a konzumentem byly proměnné typu DINT různých rozměrů. Program RSLogix5000 bez ohledu na použitou síť dovoluje maximální velikost přenášené proměnné 500 Bajtů, což odpovídá *DINT[125]*.

$$\begin{aligned} &DINT = 32\ bitů \\ &DINT\left[125\right] = 125 \cdot 32\ bitů = 4000\ bitů = \underline{\underline{500\ bajtů}} \end{aligned} \tag{6.1}$$

Toto omezení bylo zjistěno experimentálně změnami velikosti produkovanáho pole proměnných typu DINT. Omezení vychází z velikosti rámce pro síť ControlNet. Implicitní komunikace na ControlNetu probíhá prostřednictvím Generic Tag Lpakets (Obr. 2.11). Ty v sobě poskytují datům prostor velikosti 504 bajtů. Tato velikost odpovídá poli 126 DINT resp. *DINT[126]*.

$$\begin{aligned} &DINT\left[?\right] = 504\ bajtů = 504 \cdot 8 = 4032\ bitů \\ &? = \frac{4032\ bitů}{DINT} = \frac{4032\ bitů}{32} = \underline{126} \\ &DINT\left[126\right] = 126 \cdot 32\ bitů = 4032\ bitů = \underline{\underline{504\ bajtů}} \end{aligned} \tag{6.2}$$

Hodnota 504 bajtů je maximální velikost zapouzdřených dat, ale RSLogix 5000 využívá 500 z 504 možných bajtů. Toto omezení je vhodné pro síť ControlNet, kdy nám zabrání nastavit větší než maximální možnou velikost přenášených dat. Pro síť EtherNet/IP je však toto omezení velmi nepraktické, neboť velikost datového pole CPF (Obr. 2.14) v UDP rámci je mnohem větší.

6.1 ControlNet

Maximální rychlost ControlNetu je 5 Mbitů/s. Plánování komunikace na ControlNetu se provádí pomocí programu RSNetWorks for ControlNet. Tento program naplánuje komunikaci v uživatelem určeném intervalu NUT (Network Update Time) a informuje nás o plánovaném zatížení sítě. Rovněž nám informuje o chybách nastavení parametrů sítě. Ať už se jedná o chyby vlivem nesprávné volby RPI u konzumovaných dat, překročení maximální hodnoty 5 Mbitů/s na některém z modulů, či nesprávné volbě NUT. Pokud to lze, RSNetWorx for ControlNet za nás provede opravu nesprávně nastavených parametrů a uvede síť do funkčního stavu.

Chcete-li se vyhnout problémům při navrhování sítě ControlNet, dodržujte následující zásady:

a) Volba NUT

$$NUT = \frac{1}{2^n} \cdot min\left(RPI_1, RPI_2, ..., RPI_i\right) \tag{6.3}$$

, kde se NUT volí z intervalu 2 až 100 milisekund. Doporučovaná hodnota je 5 milisekund

b) Volba RPI konzumentů

$$RPI_i = 2^n \cdot NUT \tag{6.4}$$

, kde n je 1,2,…N a i je index konzumenta

6.2 EtherNet/IP

Maximální rychlost sítě EtherNet/IP je 100 Mbitů/s. Rychlost komunikace je omezena parametry komunikačních modulů. Pro moduly v laboratoři 1756-ENBT/A s revizí 3.6 jsem maximální rychlost komunikace určil jako 4850 paketů/s.

Chcete-li se vyhnout problémům při navrhování sítě EtherNet/IP, dodržujte následující zásady:

a) nepřekročit maximální počet 4850 paketů/s na komunikačním modulu 1756-ENBT/A
b) na jednom komunikačním modulu produkovat maximálně 32 proměnných
c) u optimalizovaného a přímého typu spojení nejdříve vytvořit komunikaci konzumenta s nejmenším RPI s producentem
d) RPI konzumentů volit jako

$$RPI_i = n \cdot RPI_{min} \tag{6.5}$$

, kde RPI_{min} je nejmenší RPI ze všech RPI, n je 1,2,...N nebo $1, \frac{1}{2}, ..., \frac{1}{N}$,

i je index konzumenta

Kvůli programu RSLogix 5000 a jeho omezení na velikost produkovaného tagu, nelze využít výhod sítě EtherNet/IP pro přenos velkých objemů dat.

7 Závěr

První část práce zahrnovala seznámení s komunikačními linkami architektury NetLinx firmy Rockwell Automation, se zaměřením zejména na linky ControlNet a EtherNet/IP. Jelikož EtherNet/IP v současné době představuje zapouzdřování CIP protokolu do rámce Ethernetu, je popsána i komunikační linka Ethernet.

Dalším krokem bylo srovnání linek Rockwell Automation s komunikačními linkami dalších předních světových výrobců. Při tomto srovnávání jsem se zaměřil na porovnání komunikace pro real-time řízení prostřednictvím Ethernetu. U řízení přes Ethernet jsou vidět značné rozdíly u zvolených řešení jednotlivých společností. Zásadní rozdíl lze shledat mezi produkty společnosti Siemens a Rockwell Automation, kdy tyto společnosti zvolily principielně zcela opačné řešení. Společnost Siemens využívá pro zajištění real-time komunikace na Ethernetu speciálních hardwarových zařízení, které mohou omezit kompatibilitu s ostatními komunikačními linkami. Oproti tomu společnost Rockwell Automation se zaměřila na optimalizaci vlastností CIP protokolu pro zajištění real-time řízení. Vzhledem k tomu, že CIP protokol tvoří horní vrstvy OSI modelu, není třeba žádných speciálních hardwarových produktů. Optimalizace probíhá na straně softwarových aplikací podporovanými rychlými komunikačními moduly, které se principiálně neliší od pomalejších modulů, pouze jsou doplněny o větší výpočetní výkon.

Další částí práce bylo vytvoření webové aplikace sloužící pro prezentaci Laboratoře Allen-Bradley K23 zajišťující jednotný design a snadnou údržbu. Obsah původních stránek laboratoře byl převzat a doplněn o aktuální informace, včetně doplnění nově vytvořených aplikací určených pro řízení a prezentaci zařízení Allen-Bradley. Do webových stránek laboratoře byla integrována Flash aplikace řídicí model dvouosého soustruhu. Rovněž byly doplněny Flash aplikace vytvořené pro měření a diagnostiku zatížení ethernetových komunikačním modulů a přepínače Cisco Catalyst. Tyto nástroje jsem vytvořil a používal pro analýzu zatížení sítě Ethernet při implicitní komunikaci mezi automaty. Nástroje jsou omezeny maximální délkou 343,6 sekund mezi zjišťováním stavu sítě, při zjišťovaní dat však využívají princip, který v budoucnu případně dovolí začlenit tyto aplikace do firmwaru komunikačních modulů Allen-Bradley.

Hlavní částí práce byla diagnostika způsobu komunikace mezi moduly při implicitní komunikaci a určení způsobu predikování zatížení sítě. Původní záměr byl pouze ověřit způsob predikování zatížení uvedený ve firemních manuálech Rockwell Automation. Experimentálně však bylo zjištěno, že tyto rovnice lze použít pouze jako hrubý odhad. Analýzou, resp. odposlechem na síti Ethernet při implicitní komunikaci, jsem proto určil nové rovnice, které umožňují přesně předpovídat zatížení při různých typech implicitní komunikace. Při analýze principu komunikace byla rovněž zjištěna nesprávná funkce čítačů UDP paketů a prohození indexování rozhranní u starších a nových komunikačních modulů. Rovněž údaje uvedené o zatížení v programu RSLinx poskytovaly odlišné informace o zatížení ve srovnání s daty odposlechnutými na síti.

Při provádění a srovnávání implicitní komunikace na linkách ControlNet a EtherNet/IP jsem zjistil nevhodnou vlastnost programu RSLogix 5000, který omezuje velikost produkovaného tagu na 500 bajtů, a to bez ohledu na použitou komunikační linku. Tato vlastnost zabránila provedení nejzajímavějšího srovnání při přenášení různě velkých dat na linkách ControlNet a EtherNet/IP. Při nastavitelné maximální velikosti proměnné DINT[125] jsem nezaznamenal žádné významné rozdíly mezi komunikací prostřednictvím těchto linek. Obě linky zvládaly přenos proměnné v této velikosti bezproblémově a s nastaveným RPI. Rozdíl však nastává při přenášení velkých dat, kdy EtherNet/IP při přenášení velkých dat dokáže poskytnout větší rychlost komunikace než ControlNet. Tento závěr je však dán pouze teoreticky, neboť kvůli omezení v RSLogix 5000 jej nebylo možné prakticky ověřit. K tomuto

závěru jsem dospěl zkoušením přenášení většího počtu maximálně velkých proměnných. EtherNet/IP zvládal komunikaci bez problému, i když vzrostla zátěž sítě vlivem otevření jedné multicastové skupiny pro každý tag. U ControlNetu jsem ale musel nastavit nastavit menší NUT a větší intervaly RPI (RPI u EtherNet/IP bylo menší u jednotlivých tagů) pro zajištění stejného počtu přenášených tagů, než tomu bylo u EtherNet/IP. Je však třeba uvést, že ačkoli ControlNet přenáší velká data pomaleji, je jeho výhodou, že umožňuje použití programu RSNetWorkx for ControlNet, který provede plánování přenosů dat na síti. EtherNet/IP takovouto možnost nenabízí a zátěž musí být předpovídána podle určených vzorců.

Pokračováním práce by mohla být analýza komunikace na ControlNetu použitím skeneru ControlNet sítě. Dále na základě vytvořeného návodu pro vytváření webového řízení provést realizaci řízení pro další modely v laboratoři s rezervačním systémem, který umožní na dálku provádět měření přechodových charakteristik a ověřovat navržené parametry regulátorů.

8 Literatura

[1] Rockwell Automation: *NetLinx – Rockwel Automation's Open Network Architecture* [online]. Publication NETS-BR001A-EN-P. August 2000. <http://www.ab.com>

[2] Kožnar, J.: *Diplomová práce EtherNet/IP a ControlNet* [online]. 2003. <http://dce.felk.cvut.cz/dolezilkova/diplomky/2003/dp_2003_koznar_jakub/dp_2003_koznar_jakub.pdf>

[3] Jágr, M.: *Svět sítí - OSI model* [online]. Poslední revize 2001-12-10 [cit.2007-05-24]. <http://www.pcsvet.cz/art/article.php?id=821>

[4] Rockwell Automation: *CIP: Not Just Another Pretty Akronym* [online]. Publication NETS-QPP004A-EN-P. July 2001. <http://www.ab.com>

[5] Rockwell Automation: *The CIP Family of Fieldbus Protocols and its Newest Member – EtherNet/IP* [online]. Publication NETS-WP003A-EN-P. October 2001. <http://www.ab.com>

[6] Open DeviceNet Vendor Association, ControlNet International: *The Common Industrial Protocol (CIP) and the Family of CIP Networks* [online]. Publication PUB00123R0. 2006. <www.odva.org>

[7] Open DeviceNet Vendor Association: *Media planning and Installation Manual – EtherNet/IP* [online]. September 2006. <www.odva.org>

[8] BOPLA Gehäuse Systeme GmbH: *General Technical Information – Protection Classification* [online]. Poslední revize 2005-10-5 [cit.2007-05-24]. <http://www.bopla-keypads.com/english/technische_informationen/ip66.html>

[9] Perkins, C.: *IP Encapsulation within IP* [online]. RFC 2003. October 1996. <http://www.ietf.org/rfc/rfc2003.txt>

[10] Perkins, C.: *Minimal IP Encapsulation within IP* [online]. RFC 2004. October 1996. <http://www.ietf.org/rfc/rfc2004.txt>

[11] Postel, J.: *User Datagram Protocol* [online]. RFC 768. August 1980. <http://www.ietf.org/rfc/rfc768.txt>

[12] Information Sience Institute: *Internet Protocol* [online]. RFC 791. September 1981. <http://www.ietf.org/rfc/rfc791.txt>

[13] Information Science Institute: *Transmission Control Protocol* [online]. RFC 793. September 1981. <http://www.ietf.org/rfc/rfc793.txt>

[14] Socolofsky T.: *TCP/IP Tuturial* [online]. RFC 1180. January 1991. <http://www.ietf.org/rfc/rfc2003.txt>

[15] Cisco Systems, Inc.: *Internetworking Technologies Handbook, chapter Ethernet Technologies* [online]. *<http://www.cisco.com/en/US/tech/tk1330/tsd_technology_support_technical_reference_chapter09186a008075974f.htm >*

[16] Wikipedie: *Kroucená dvoulinka* [online]. Poslední revize 2007-05-01 [cit.2007-05-24]. <http://cs.wikipedia.org/wiki/Kroucen%C3%A1_dvoulinka>

[17] Wikipedie: *Gigabit Ethernet* [online]. Poslední revize 2007-04-01 [cit.2007-05-24]. <http://en.wikipedia.org/wiki/Gigabit_Ethernet>

[18] IEEE Computer Society: *Part3: Carrier sense multiple Access with collision detection (CSMA/CD) access Metod and physical layer specifications* [online]. December 2005. <http://www.ieee.org>

[19] Flur, M.: *Real-time Ethernet Standrads* [online]. Poslední revize 2006-04-10. [cit.2007-05-24]. <http://docweb.khk.be/khk/embedded/ppt/NewEthernetStandards.pdf>

[20] Ogden, F.: *SPS Nuremberg sees wheels set in motion over Ethernet* [online]. Poslední revize 2007-02-20. [cit.2007-05-24]. <http://ethernet.industrial-networking.com/news/newsdisplay.asp?id=384>

[21] Sercos International: *SERCOS III interface* [online]. <http://www.sercos.com/>

[22] Novák, J.: *Diplomová práce Model lineárního servomechanismu a jeho řízení* [online]. 2004.<http://dce.felk.cvut.cz/dolezilkova/diplomky/2004/dp_2004_novak_jiri/DP_2004_Novak_Jiri.pdf>

[23] Prokůpek, Z.: *Semestrální práce Popis modulu 1756-EWEB automatu ControlLogix* [online]. 2005.<http://147.32.87.135/user/web/Soustruh/soustruh.html>

[24] Prokůpek, Z.: *Semestrální práce Popis webu na EWEBu*. 2005 <http://147.32.87.135/user/web/Soustruh/soustruh.html>

[25] Tomášek, R.: *Diplomová práce Monitorování a řízení s využitím Internetu* [online]. 2004.<http://dce.felk.cvut.cz/dolezilkova/diplomky/2004/dp_2004_tomasek_radek/dp_2004_tomasek_radek.pdf>

[26] Matouch, P.: *Diplomová práce Model víceosého servosystému* [online]. 2004. <http://dce.felk.cvut.cz/dolezilkova/diplomky/2004/dp_2004_matouch_petr/dp_2004_matouch_petr.pdf>

[27] Rockwell Automation: *ConrolLogix Ethernet Communication Interface Module 1756-ENET/B (User Manual)* [online]. Publication 1756-UM051B-EN-P. November 2000. <http://www.ab.com>

[28] Rockwell Automation: *EtherNet/IP Web Server Module 1756-EWEB, 1768-EWEB (User Manual)* [online]. Publication ENET-UM527E-EN-P. October 2006. <http://www.ab.com>

[29] W3 schools: *PHP tutorial* [online]. Poslední revize 2007-03-05 [cit.2007-01-13]. <www.w3schools.com/php>

[30] Rockwell Automation: *Rockwell Automation Encompass program* [online]. <http://www.rockwellautomation.com/encompass/>

[31] Rockwell Automation: *EtherNet/IP Performance (Application Solution)* [online]. Publication ENET-AP001D-EN-P. October 2004. <http://www.ab.com>

[32] Rockwell Automation: *EtherNet/IP Modules in Logix5000 Control System (User Manual)* [online]. Publication ENET-UM001F-EN-P. November 2006. <http://www.ab.com>

[33] Open DeviceNet Vendor Association: EtherNet/IP – CIP on Ethernet technology [online]. October 2006. Publication PUB00138R2. <http://www.odva.org>

[34] Rockwell Automation: *NetLinx Selection Guide* [online]. Publication NETS-SG001B-EN-P. December 2004. <http://www.ab.com>

[35] Beckhoff: Ethernet for Control and Automation [online]. 2003. <ftp://ftp.beckhoff.com/press/2003/usa/Ethernet_for_Control_and_Automation.pdf>

[36] Cisco Systems, Inc.: *All Transparent VTP Domain To Server-Client VTP Domain Migration Configuration Example* [online]. Document ID 81682. <*http://www.cisco.com*>

[37] Cisco Systems, Inc.: *Getting Started with Cisco Network Assistant (Version 5.0)* [online]. Publication OL-6194-06. <*http://www.cisco.com*>

[38] Cisco Systems, Inc.: *Catalyst 2960 Switch Getting Started Guide* [online]. Publication OL-9368-01. <*http://www.cisco.com*>

[39] Cisco Systems, Inc.: *Catalyst 2960 Switch Software Configuration Guide(Cisco IOS Release 12.2(25)FX)* [online]. November 2005. Publication 78-16881-02. Order Number DOC-7816881.<*http://www.cisco.com*>

[40] Cisco Systems, Inc.: *Cisco Catalyst 2960 Series Switch* [online]. November 2006. Publication C97-373926-00. <*http://www.cisco.com*>

9 Seznam software

Adobe Acrobat [počítačový program]. Ver. 7.0.9 CE. Adobe System Inc.

Apache2 [počítačový program]. Ver. 2.2.4. The Apache Software Foundation., 2003. [cit. 8.5.2007]. <http://www.apache.org/>.

Internetová informační služba IIS [počítačový program]. Ver 5.1 CE, Microsoft Corporation.

PHP 5.2.2 [počítačový program]. Ver. 5.2.2. The PHP Group, 2004. [cit. 8.5.2007]. <http://www.php.net/>.

Adobe Illustrator [počítačový program]. Ver. 11.0.0. Adobe System Inc.

Macromedia Dreamweaver [počítačový program]. Ver. 8.0. Macromedia Inc.

Macromedia Flash Professional [počítačový program]. Ver. 8.0. Macromedia Inc.

Wireshark [počítačový program]. Ver. 0.99.5. Gerald Combs 2007. [cit. 8.5.2007]. <http://www.wireshark.org/>.

Cisco Network Assistant [počítačový program]. Ver 5.1. Cisco Systems Inc.

Microsoft Office 2003 [počítačový program]. Ver. 11.5604.5606. Microsoft Corporation, 2003.

RSLinx [počítačový program]. Ver. 2.20.02. Rockwell Software Inc. 2005.

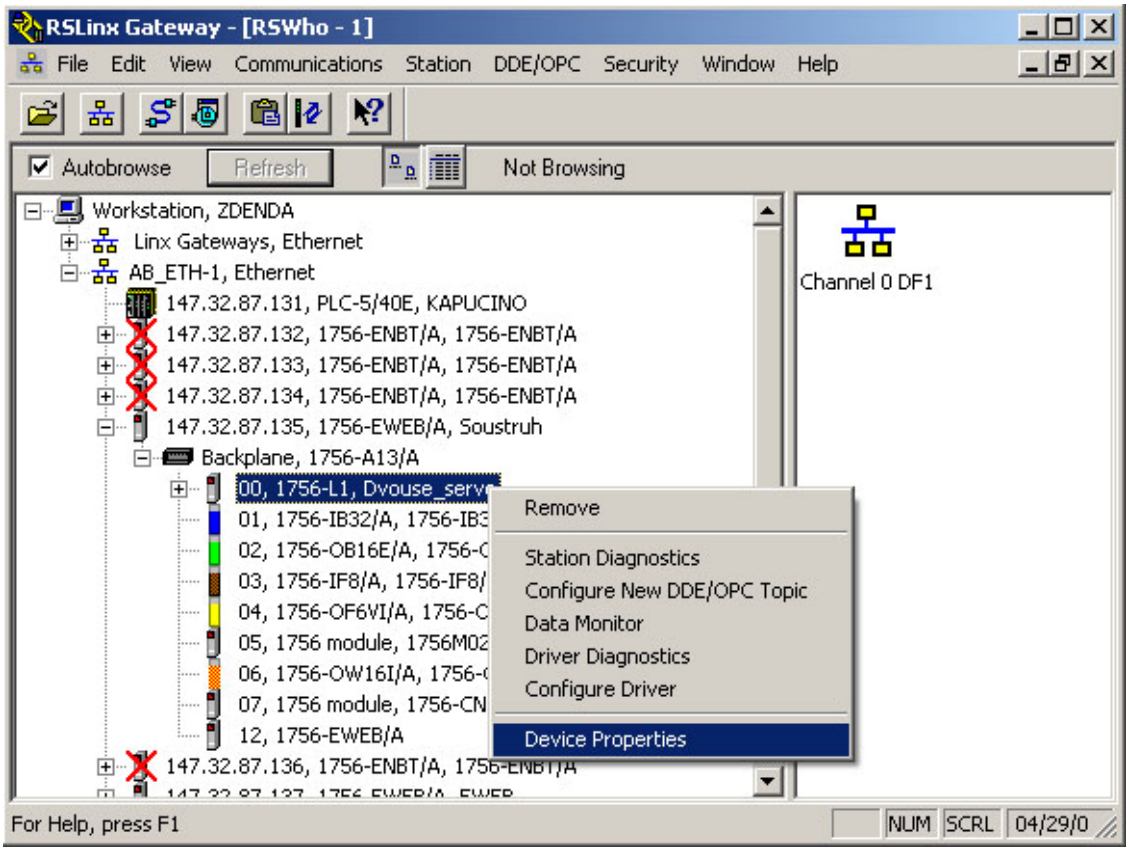
RSLogix5000 [počítačový program]. Ver. 15.02.00. Rockwell Software Inc. 2005

Příloha A – Komunikace pomocí modulů 1756-ENBT/A a 1756-EWEB/A

A.1 Webové řízení pomocí 1756-EWEB, podrobný návod

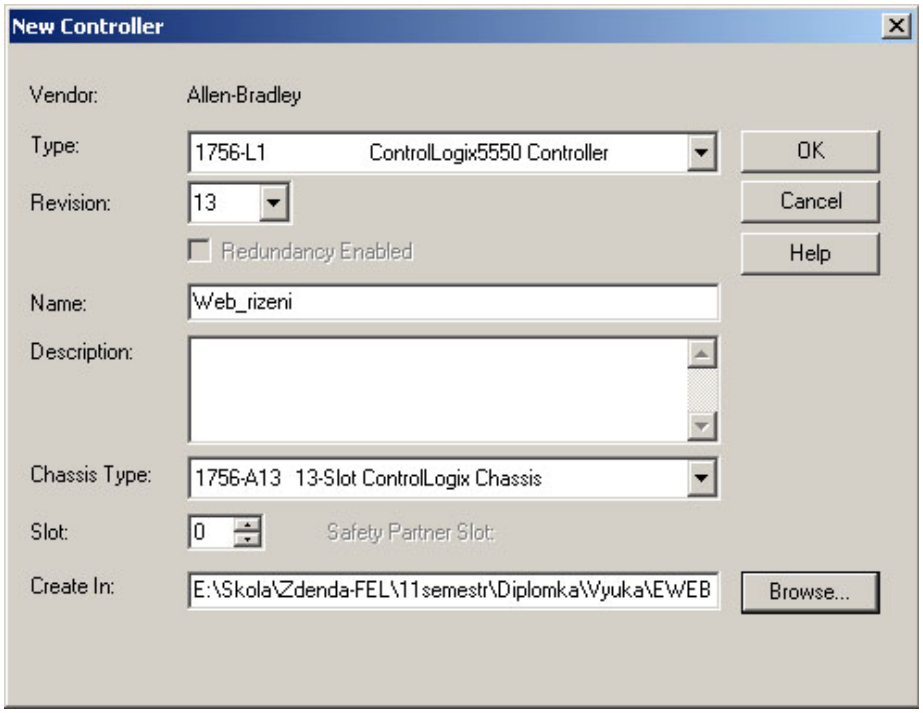
Vytvoření projektu v RSLogix 5000

1) Spustíme RSLinx.
2) V programu RSLinx klikneme na ikonu RSWho.
3) V nově otevřeném okně RSWho projdeme stromovou strukturou k větvi Ethernet, kterou pro získávání informací o modulech pro náš případ budeme preferovat.
4) Vybereme ethernetový modul (podle IP adresy a typu modulu), který bude zajišťovat HTTP komunikaci s internetovými prohlížeči. Modul, který poskytuje HTTP komunikaci je webový server 1756-EWEB/A.
   a) V jeho větvi je položka Backplane a za ním je uveden typ rámu (chassis) (např. 1756-A13/A).
   b) Dále se podíváme na procesor a slot, který určuje pozici procesoru v rámu. V našem případě je to 0. Pozice muže být maximálně velikost rámu zmenšená o jedna, protože sloty jsou číslovány od nuly.
   c) Vlastnosti procesoru si zobrazíme kliknutím pravého tlačítka na procesor a následnou volbou Device Properties. Najdeme revizi firmwaru (Revision).



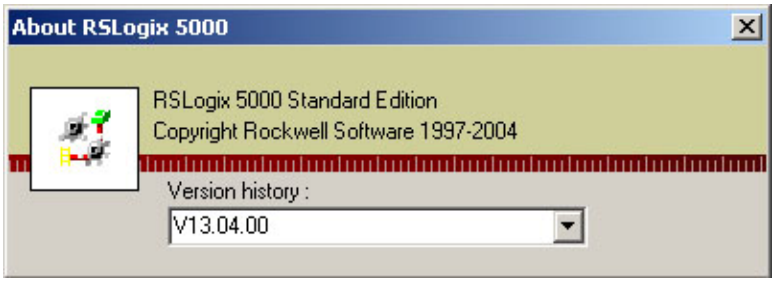
Obr. A.1 RSLinx zjištění hardwarové konfigurace

5) Spustíme program RSLogix 5000.
6) Vytvoříme nový projekt (File – New) a vyplníme zjištěné informace z RSLinx.



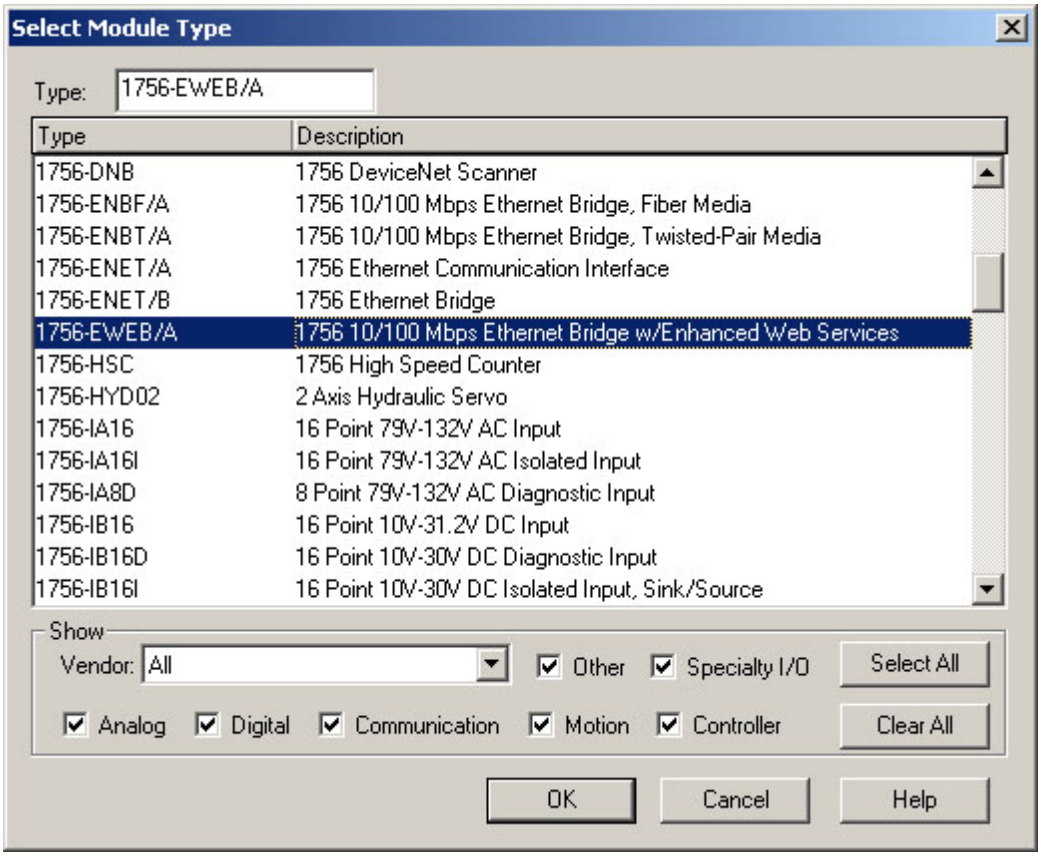
Obr. A.2 Založení nového projektu

7) Vyplníme typ procesoru (Type), revizi firmwaru (Revision), jméno projektu (Name), typ rámu (Chassis Type), pozici procesoru v rámu (Slot) a adresář pro uložení (Create In).
8) Po stisknutí tlačítka OK se spustí RSLogix ve verzi odpovídající zadané revizi firmwaru procesoru.



Obr. A.3 Informace o verzi spuštěného programu

9) Nyní zpřístupníme ethernetovský modul 1756-EWEB/A našemu projektu.
   a) Klikneme pravým tlačítkem na I/O konfiguraci (I/O configuration), vybereme nový modul (New Module) a zvolíme modul 1756-EWEB/A.



Obr. A.4 Přidání zvoleného modulu do I/O konfigurace

b) V nově otevřeném okně vyplníme parametry 1756-EWEB/A (pozici a IP adresu modulu), které zjistíme přes RSLinx stejným způsobem jako jsme to dělali u procesoru v bode [4.c]. Dále nastavíme Electronic Keying jako Disable keying a klikneme na dokončit (Finish).

10) Vytvoříme proměnnou (tag), který je přístupný přes Internet. Klikneme na Controller Tags pravým tlačítkem a vybereme New Tag.

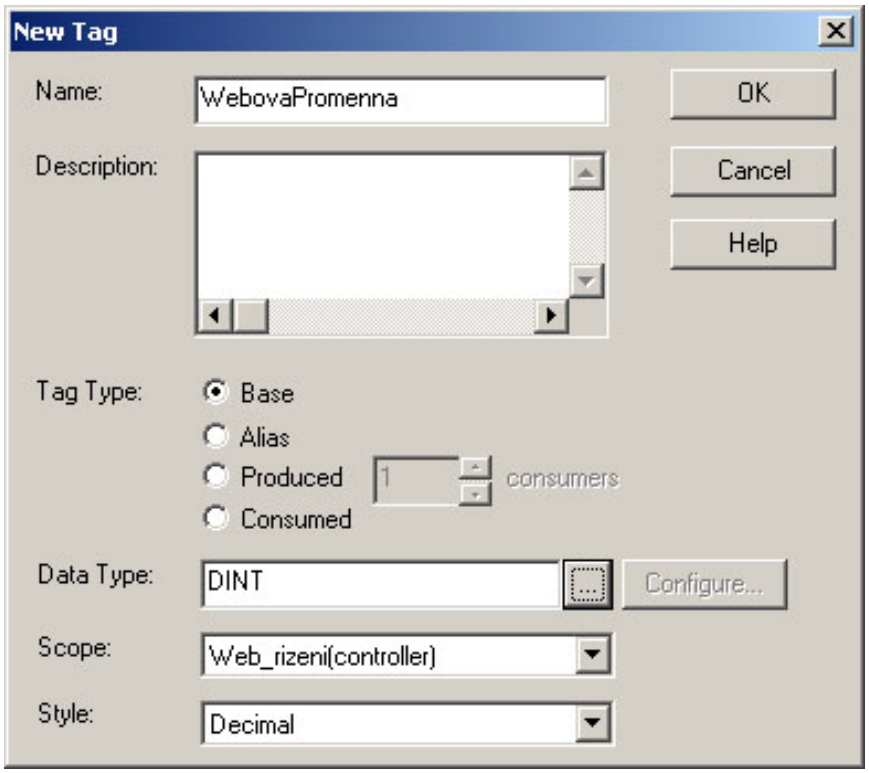
a) Vyplníme Name, Data Type, Scope, Tag Type.

b) Name je jméno proměnné.

c) Tag type určuje typ proměnné a v našem případě se jedná o základní typ (Base).

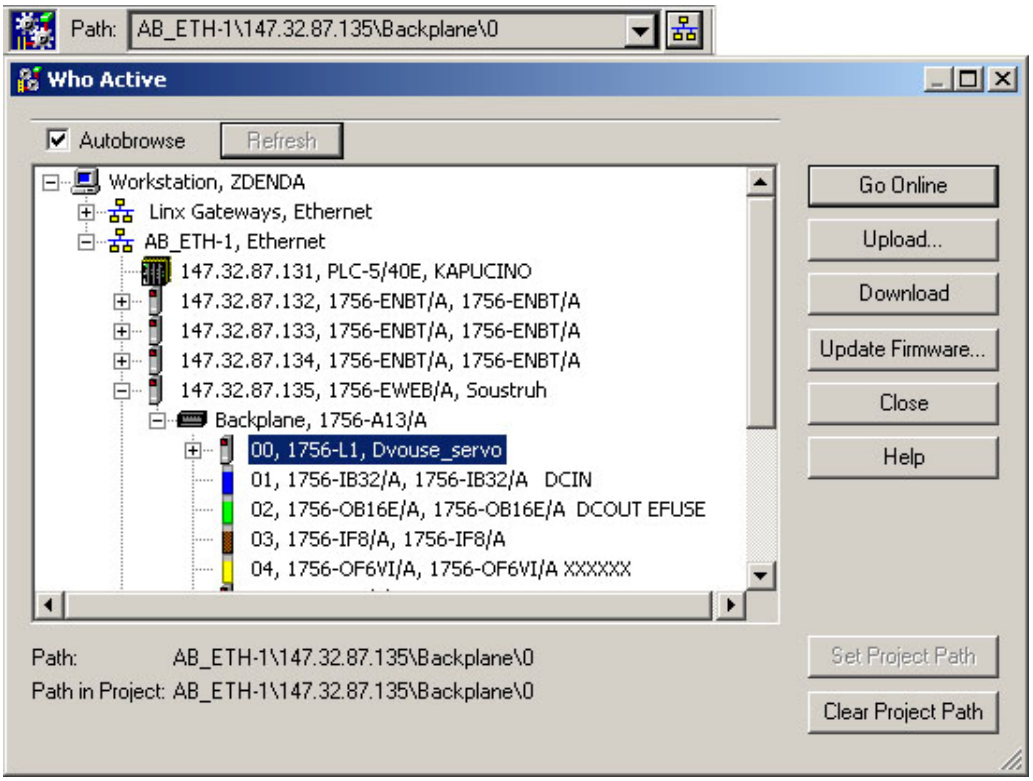
d) Data Type určuje datový typ proměnné, zvolíme DINT.

e) Scope určuje umístění (dostupnost) proměnné. Vytvoříme Controller tag (globální proměnnou).



Obr. A.5 Vytvoření nové proměnné

11) Zadáme cestu (Path), kam se námi vytvořený projekt nahraje (Download). Klikneme na ikonu Who active. V nově otevřeném okně se nám objeví dostupné cesty. Vybereme zařízení s odpovídající IP adresou, pro které jsme projekt vytvářeli a projdeme stromovou strukturou až k procesoru, do kterého chceme vytvořený projekt nahrát. Po označení procesoru se zpřístupní tlačítko Set Project Path, kterým cestu nastavíme.



Obr. A.6 Nastavení cesty projektu

12) Klikneme v menu na položku komunikace (Communications) a vybereme nahrát (Download).
13) Nyní je v automatu s odpovídající IP adresou nahrán program, který bude poskytovat proměnnou přístupnou přes http protocol.
14) Je třeba ručně (klíčkem na procesoru) nebo softwarově uvést automat do RUN MODu(tím se program začne vykonávat).

Vytvoření webových stránek

Práce s proměnnými v ControlLogix pomocí formulářů

a) nastavení hodnoty proměnné v procesoru ControlLogix

```
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Nastavení proměnné</title>
</head>

<body>
    <form action="http://147.32.87.135/rokform/WriteLogixTags" method="POST" >
        <input type='hidden'  name='redirect' value="http://dce.felk.cvut.cz/ab">  <!--kam
        přesměrovat  -->
        <input type='hidden'  name='numtags' value='1'>                    <!-- počet tagů -->
        <input type='hidden'  name='t_1_tagname' value='WebovaPromenna'> <!-- jméno
        tagu -->
        <input type='hidden'  name='t_1_slot' value='0'>            <!-- pozice procesoru  -->
        <input type='hidden'  name='t_1_type' value='DINT'>                <!-- typ tagu  -->
        <input type='hidden'  name='t_1_display' value='Decimal'> <!-- způsob zobrazeví
        -->
        <input type='hidden'  name='t_1_changed' value='1'> <!-- zapiš novou hodnotu do
        procesoru -->
        <input type='text'      name='t_1_value' value='0'>        <!-- hodnota k nahrání -->
        <input type="submit" value="Nastav data - tag">
    </form>
</body>
</html>
```

b) přečtení hodnoty proměnné z procesoru ControlLogix

```
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Přečtení proměnné</title>
</head>

<body>
    <form action="http://147.32.87.135/rokform/ReadLogixTag" method="POST">
        <input type="hidden" name="name" value="WebovaPromenna"><!-- hodnota k
nahrání -->
        <input type="hidden" name="path" value="1,0">            <!-- backplane, slot -->
        <input type="hidden" name="type" value="DINT">           <!-- typ proměnné -->
        <input type="hidden" name="connected" value="true">   <!-- typ komunikace -->
        <input type="submit" value="Přečti data - tag">
    </form>
</body>
</html>
```

Administrace webového serveru

Pokud nechceme při každém odeslání požadavků zadávat uživatelské jméno a heslo administrátora, musíme provést administraci webového serveru 1756-EWEB. Upravíme přístupová práva ke skriptům, které zpracovávají příchozí požadavky zaslané formuláři, ReadLogixTag a WriteLogixTag pro anonymní přístup.

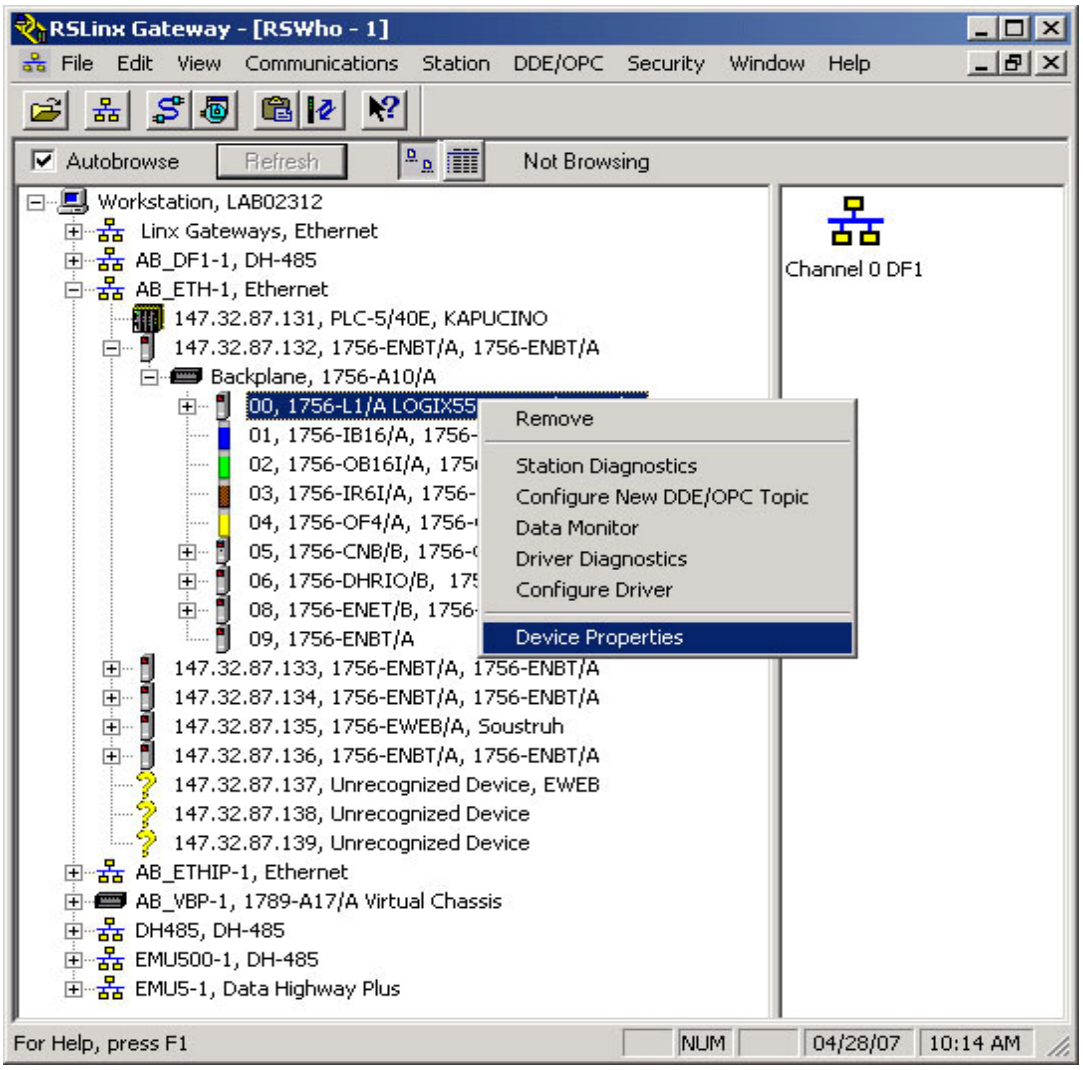
Do internetového prohlížeče zadáme http://ipadresa_eweb_serveru/index.html. V navigačním menu najdeme položku Administrative settings – User Management – Edit Access Limits a kliknutím na červený křížek provedeme zrušení administrátorského zabezpečení pro tyto skripty. (Nebudou-li tyto položky viditelné zobrazíme si je odškrtnutím položky Hide System Access Limits)

Obr. A.7 Weboé rozhraní server 1756-EWEB

A.2 Komunikace producent/konzument 1756-ENBT, přenášení proměnné

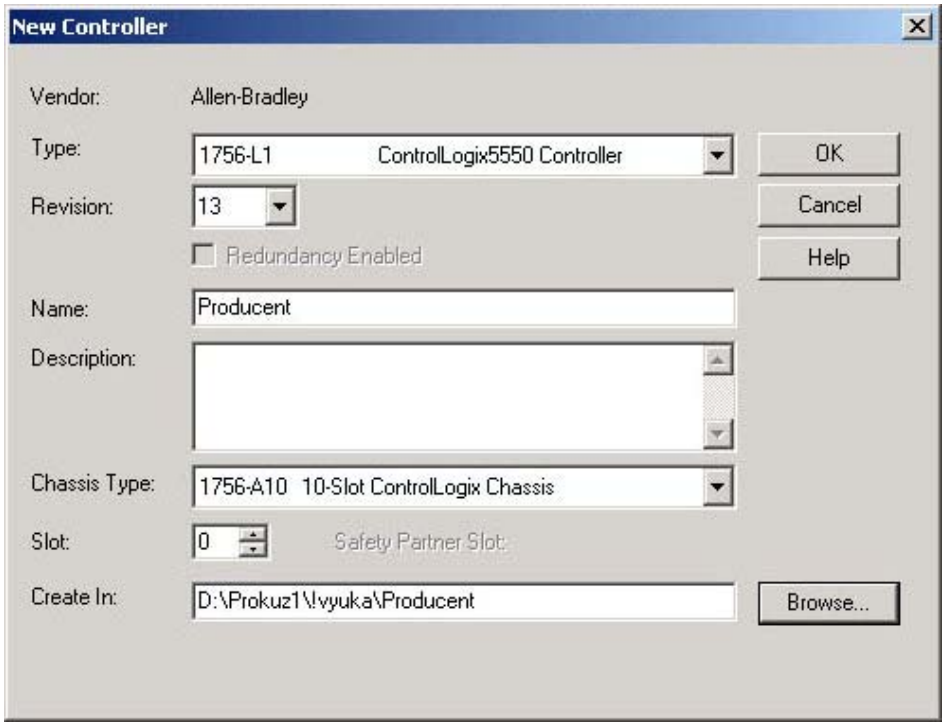
Vytvoření Producenta

1) Spustíme RSLinx.
2) V programu RSLinx klikneme na ikonu RSWho.
3) Projdeme stromovou strukturou k větvi Ethernet, kterou pro získávání informací v našem případě preferujeme.
4) Vybereme automat resp. ethernetový modul (podle IP adresy), který bude zajišťovat produkci proměnných (tags) z procesoru zapojeného ve stejném rámu (chassis).
   a) V jeho větvi je položka Backplane a za ním je uveden typ rámu (např. 1756-A10/A).
   b) Dále se podíváme na procesor a slot, který určuje pozici procesoru v rámu. V našem případě je to 0. Pozice muže být maximálně velikost rámu zmenšená o jedna, protože sloty jsou číslovány od nuly.
   c) Vlastnosti procesoru si zobrazíme kliknutím pravého tlačítka na náš procesor a následnou volbou Device Properties, najdeme revizi firmwaru (Revision).



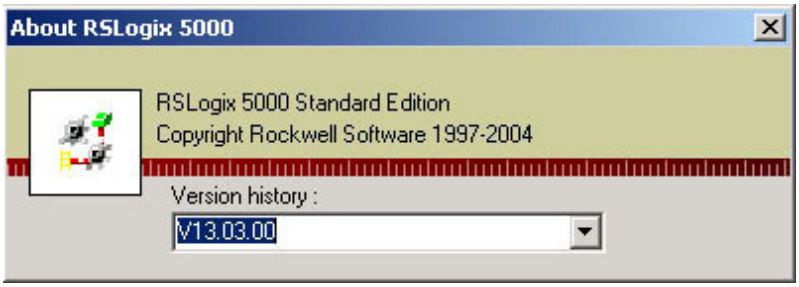
Obr. A.8 RSLinx, zjištění hardwarové konfigurace

5) Spustíme program RSLogix 5000.
6) Vytvoříme nový projekt (File – New) a vyplníme zjištěné informace z RSLinx.



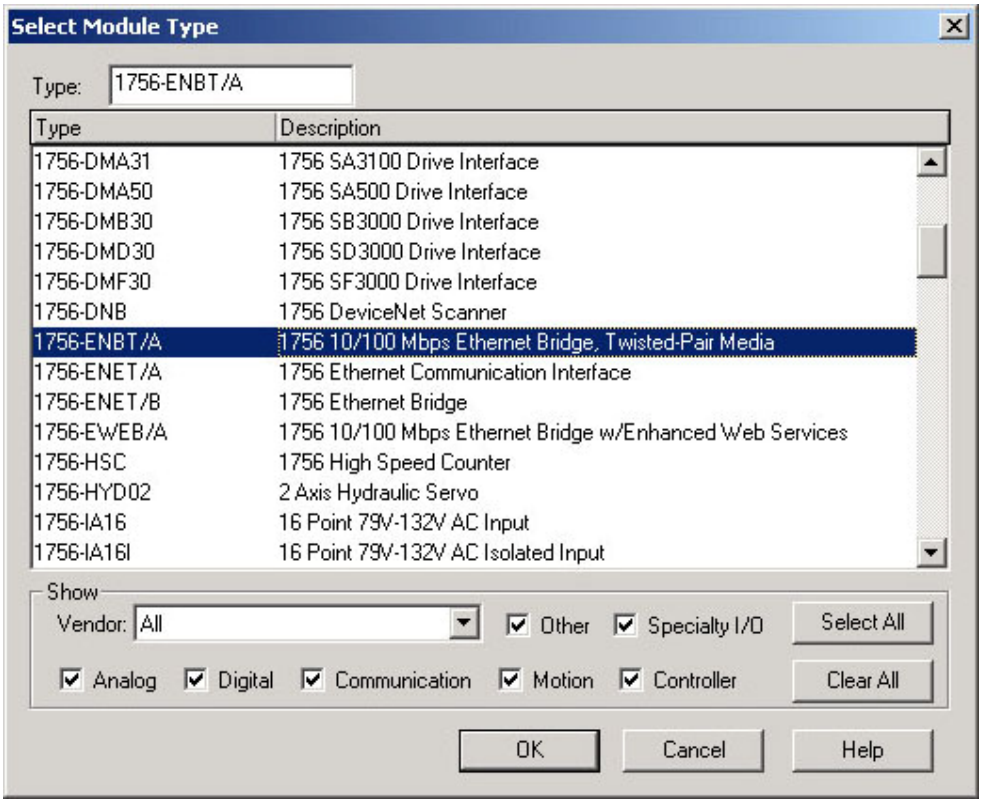
Obr. A.9 Založení nového projektu

7) Vyplníme typ procesoru (Type), revizi firmwaru (Revision), jméno projektu (Name), typ rámu (Chassis Type), pozici procesoru v rámu (Slot) a adresář pro uložení (Create In).
8) Po stisknutí tlačítka OK se spustí RSLogix ve verzi odpovídající zadané revizi firmwaru procesoru.



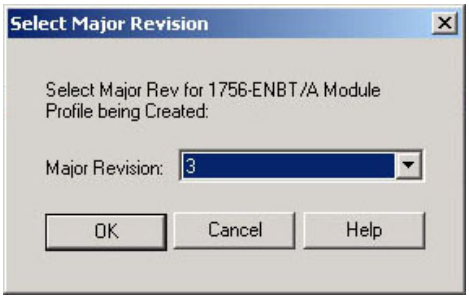
Obr. A.10 Ověření spuštění RSLogix v odpovídající verzi

9) Nyní zpřístupníme ethernetový modul našemu projektu (tento bod můžeme přeskočit pokud nechceme zasílat CIP zprávy pomocí instrukce MSG).
   a) Klikneme pravým tlačítkem na I/O konfiguraci (I/O configuration), vybereme nový modul (New Module) a vybereme modul 1756-ENBT/A. To je modul, který se nachází v našem rámu a komunikuje přes Ethernet resp. Ethernet/IP (Ethernet + CIP).



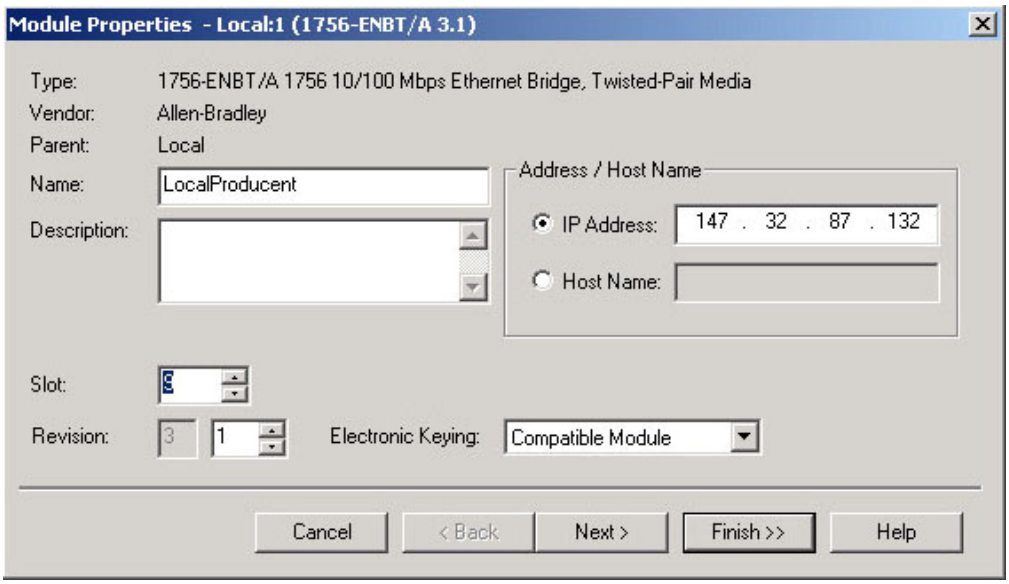
Obr. A.11 Začlenění komunikačního modulu do I/O konfigurace

b) Jakmile potvrdíme volbu modulu, musíme zadat hlavní revizi firmwaru (Major revision). V RSLinx najdeme náš zvolený 1756-ENBT/A modul a zobrazíme si jeho vlastnosti stejně jako jsme to dělali v bodě [4.c] . Řádek Revision obsahuje číslo např. 3.9. Číslo 3 je hlavní revize a 9 je minoritní (Minor) revize.



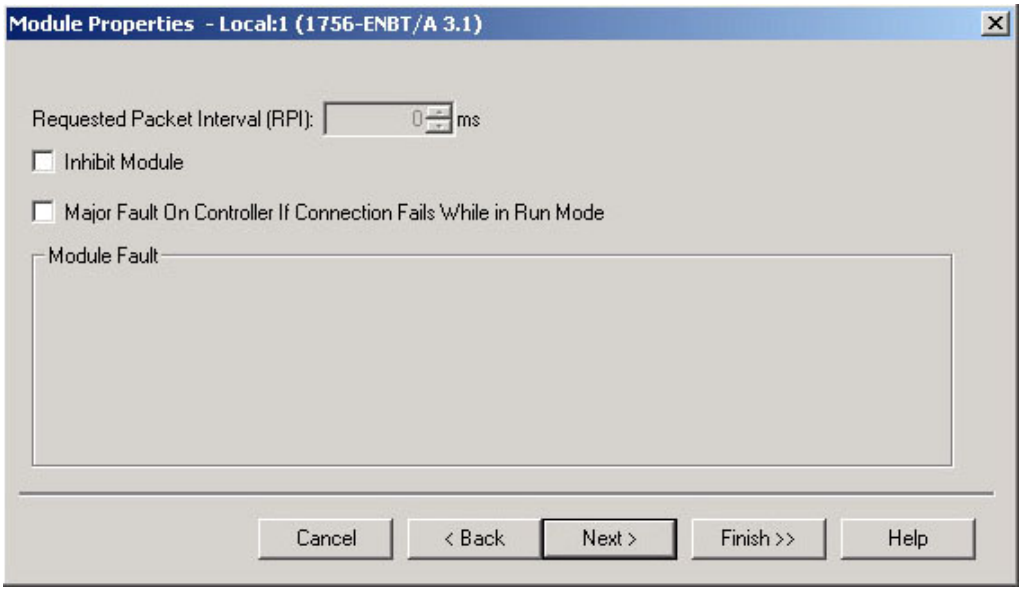
Obr. A.12 Volba hlavní revize pro ENBT modul

c) Vyplníme parametry 1756-ENBT/A (pozici a IP adresu modulu), které zjistíme přes RSLinx. Dále zvolíme Electronic Keying jako Compatible module. Minoritní revizi necháme 1.



Obr. A.13 Nastavení parametrů modulu 1756-ENBT/A

d) V okně parametrů modulu (Module properties) se nastavuje RPI (Requested Packet Interval). Pro producenta, však tato možnost není přístupná, protože producent vysílá pakety podle požadavků resp. nastavení konzumenta. Klikneme na Finish.



Obr. A.14 Nastavení RPI pro producenta

10) Vytvoříme produkovanou proměnnou (tag). Klikneme na Controller Tags pravým tlačítkem a vybereme vytvoření nového tagu (New Tag).

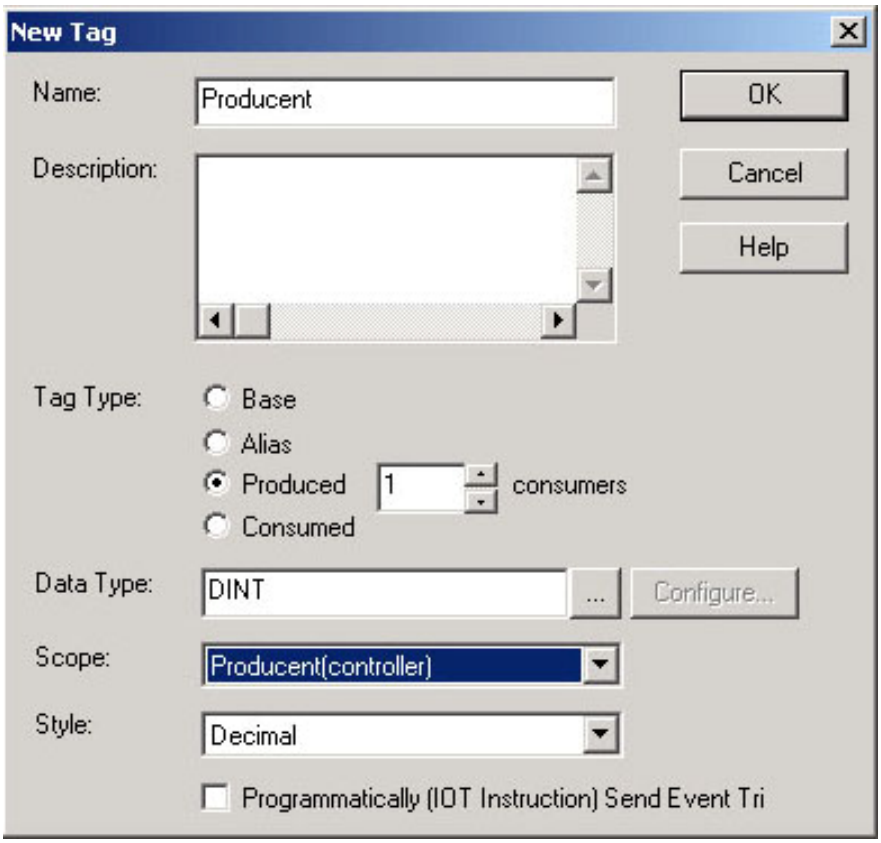
a) Vyplníme Name, Data Type, Scope, Tag Type.

b) Name je jméno proměnné.

c) Tag type určuje typ proměnné a v našem případě se jedná o typ produkované proměnné (Produced). Navíc určíme počet konzumentů, tedy počet příjemců této proměnné.

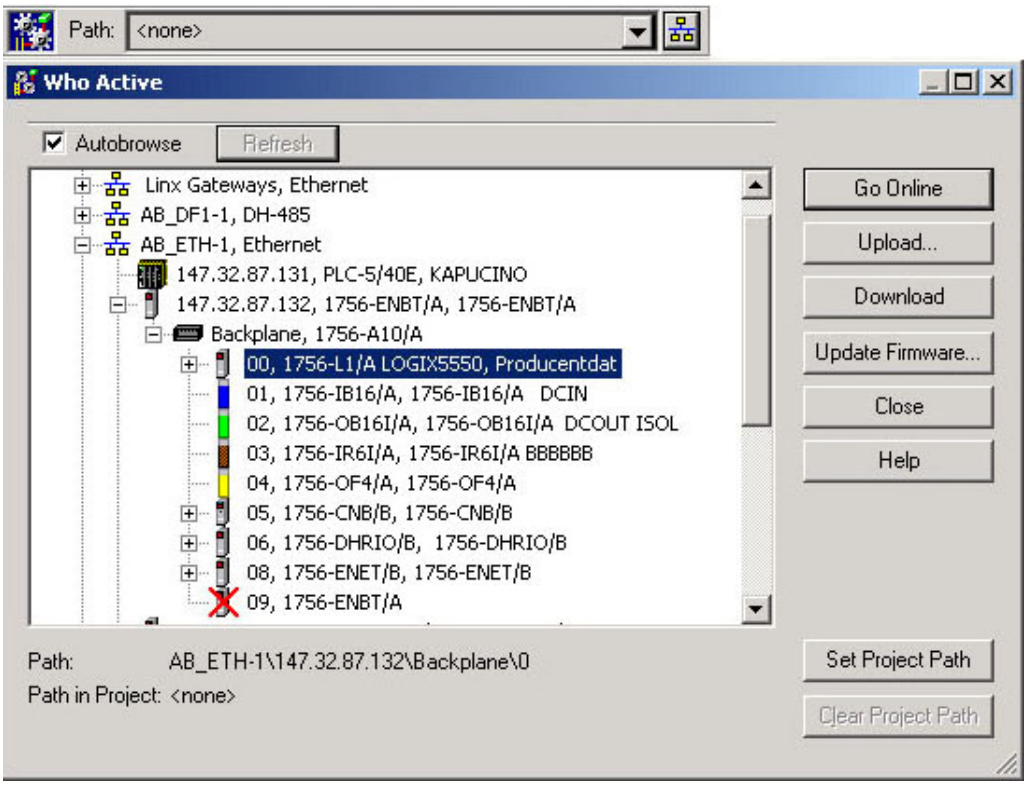
d) Data Type určuje datový typ proměnné.

e) Scope určuje umístění (dostupnost) proměnné. Vytvoříme Controller tag (globální proměnnou), která lze produkovat a navíc je přístupná ve všech podprogramech (routine).



Obr. A.15 Vytvoření produkované proměnné

11) Nastavíme cestu (Path). Cestu kam se námi vytvořený projekt nahraje (Download). Klikneme na ikonu Who active. V nově otevřeném okně se nám objeví dostupné cesty. Vybereme zařízení s odpovídající IP adresou, pro které jsme projekt vytvářeli a projdeme stromovou strukturou až k procesoru, do kterého chceme vytvořený projekt nahrát. Po označení procesoru se zpřístupní tlačítko nastav cestu (Set Project Path), kterým cestu nastavíme.

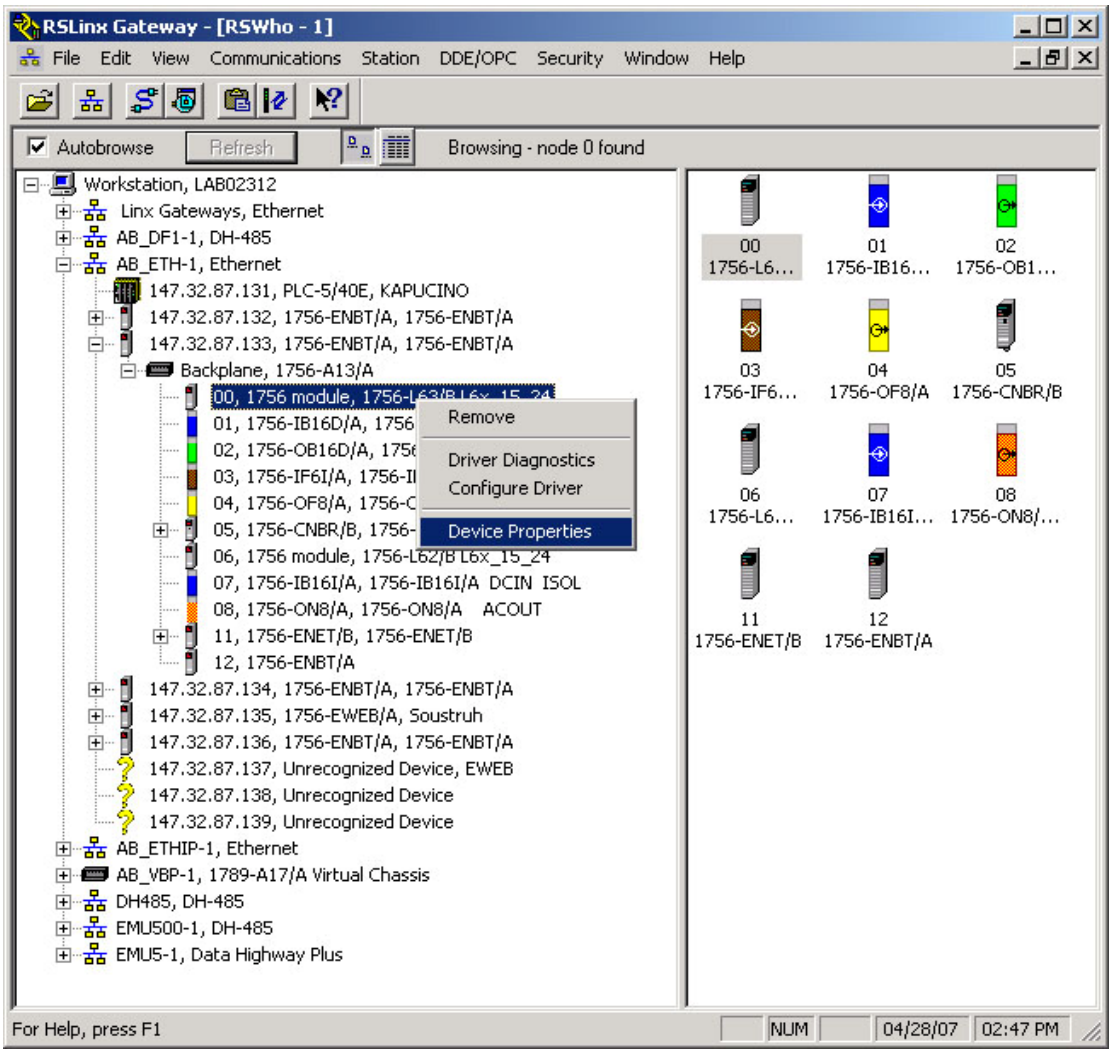


Obr. A.16 Nastavení cesty projektu

12) Klikneme v menu na položku komunikace (Communications) a vybereme nahrát (Download).
13) Nyní je v automatu s odpovídající adresou nahrán program, který bude produkovat data pro ostatní zařízení.
14) Je třeba ručně (klíčkem na procesoru) nebo softwarově uvést automat do RUN MODu(tím se program začne vykonávat).

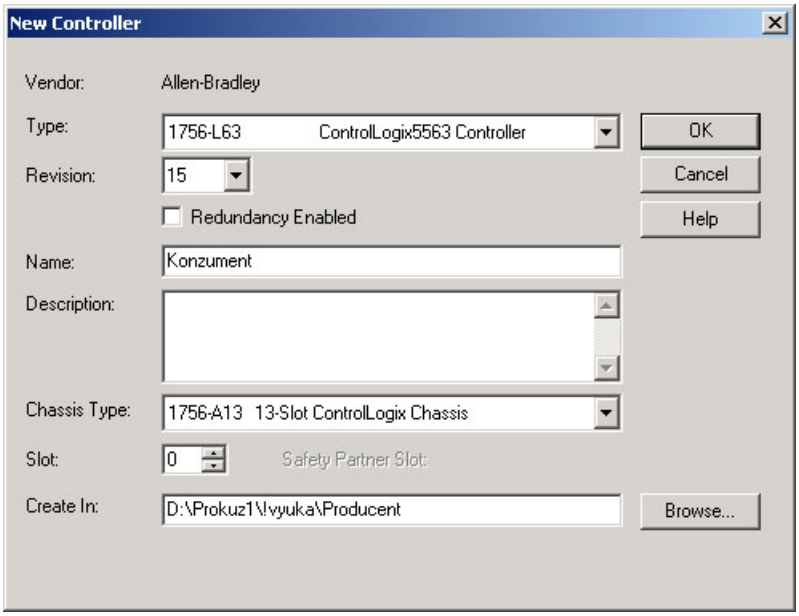
<u>Vytvoření Konzumenta</u>

1) Spustíme RSLinx.
2) V programu RSLinx klikneme na ikonu RSWho.
3) V okně RSWho projdeme stromovou strukturou k větvi Ethernet, kterou pro získávání informací pro náš případ budeme preferovat.
4) Vybereme ethernetový modul (podle IP adresy), který bude zajišťovat získávání proměnných (tags) ze vzdáleného procesoru.
   a) V jeho větvi je položka Backplane a za ním je uveden typ rámu (chassis) (např. 1756-A13/A).
   b) Dále se podíváme na procesor a slot, který určuje pozici procesoru v rámu. V našem případě je to 0. Pozice muže být maximálně velikost rámu zmenšená o jedna, protože sloty jsou číslovány od nuly.
   c) Vlastnosti procesoru si zobrazíme kliknutím pravého tlačítka na procesor a následnou volbou Device Properties. Najdeme revizi firmwaru (Revision).



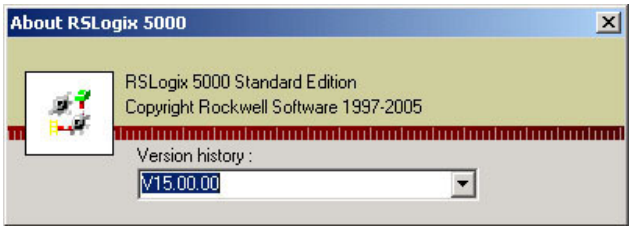
Obr. A.17 Zjištění hardwarové konfigurace

5) Spustíme program RSLogix 5000.
6) Vytvoříme nový projekt File – New a vyplníme zjištěné informace z RSLinx.



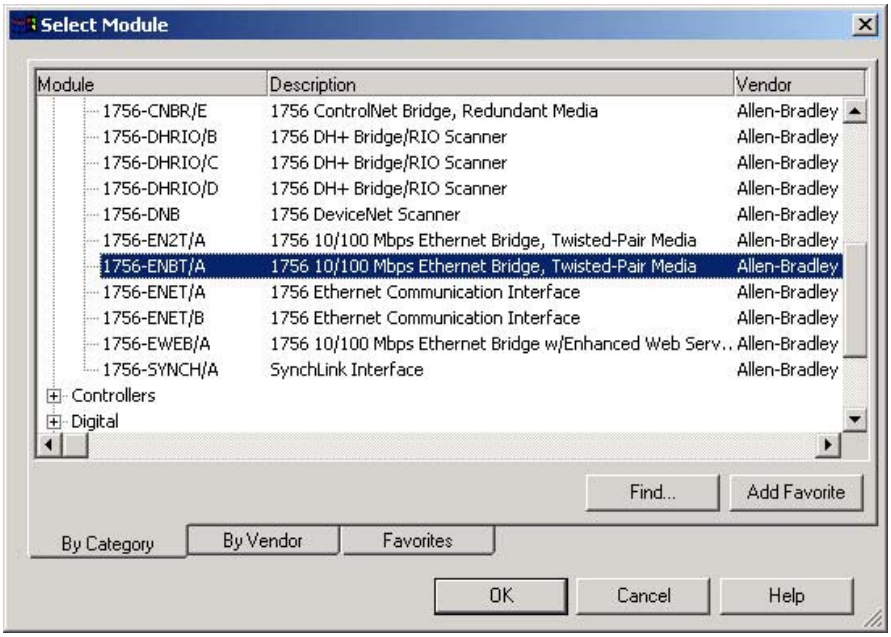
Obr. A.18 Založení nového projektu

7) Vyplníme typ procesoru (Type), revizi firmwaru (Revision), jméno projektu (Name), typ rámu (Chassis Type), pozici procesoru v rámu (Slot) a adresář pro uložení (Create In).
8) Po stisknutí tlačítka OK se spustí RSLogix ve verzi odpovídající zadané revizi firmwaru procesoru.



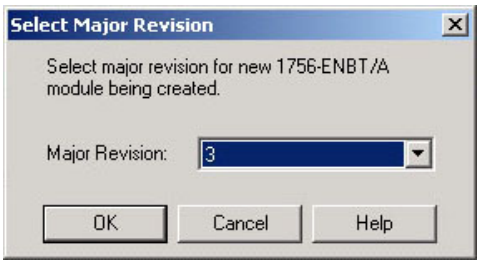
Obr. A.19 Ověření spuštění RSLogix v odpovídající verzi

9) Nyní zpřístupníme lokální ethernetovský modul našemu projektu. Tento modul bude přijímat produkovaná data přes EtherNet/IP z ethernetovského modulu vzdáleného automatu.
   a) Klikneme pravým tlačítkem na I/O konfiguraci (I/O configuration), vybereme nový modul (New Module) a zvolíme modul 1756-ENBT/A. To je modul, který se nachází v našem rámu a komunikuje přes Ethernet resp. Ethernet/IP (Ethernet + CIP).



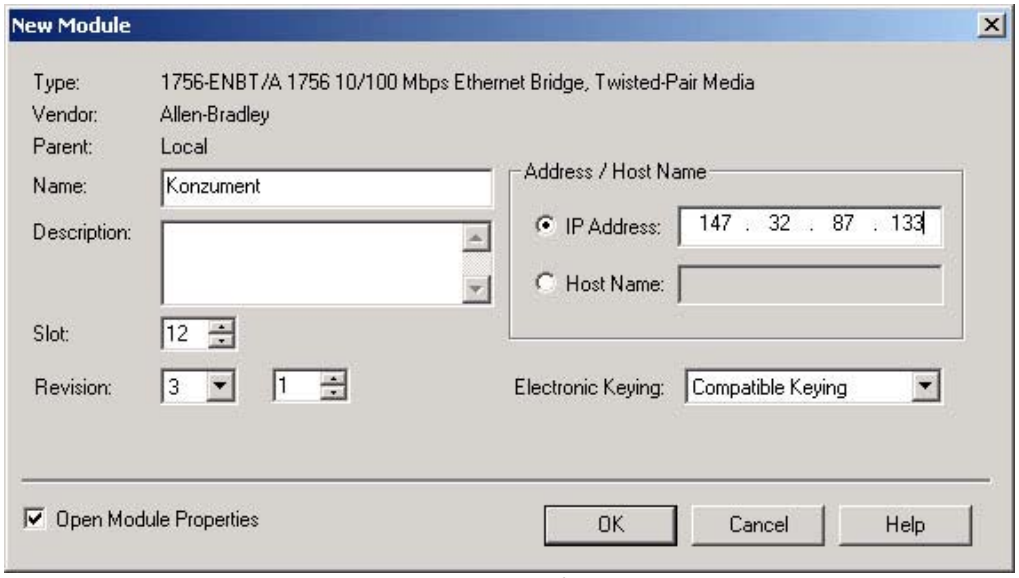
Obr. A.20 Přidání komunikačního modulu di I/O konfigurace

   b) Jakmile potvrdíme volbu modulu, musíme zadat hlavní revizi firmwaru (Major revision). V RSLinx najdeme náš zvolený 1756-ENBT/A modul a zobrazíme si jeho vlastnosti stejně jako jsme to dělali v bodě [4.c] pro procesor. Řádek Revision obsahuje číslo např. 3.9. Číslo 3 je hlavní revize a 9 je minoritní (Minor) revize.



Obr. A.21 Nastavení hlavní revize

c) Vyplníme parametry 1756-ENBT/A (pozici a IP adresu modulu), které zjistíme přes RSLinx. Dále zvolíme Electronic Keying jako Compatible module. Minoritní revizi necháme 1.

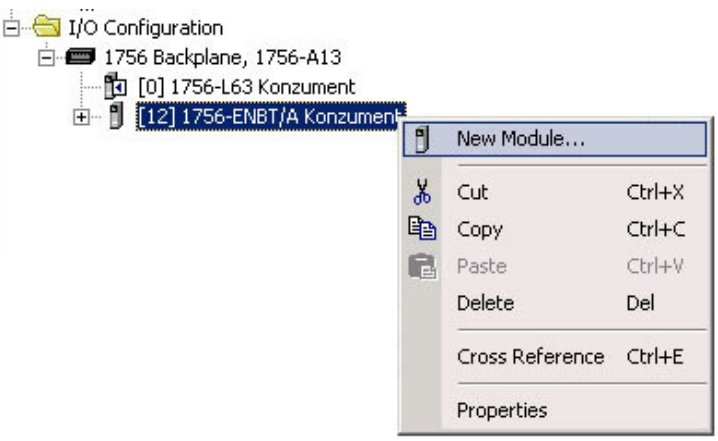


Obr. A.22 Nastavení parametrů přidávaného modulu

d) V okně parametrů modulu (Module properties) se nastavuje RPI (Requested Packet Interval). Pro lokální ethernetovský modul konzumenta, však tato možnost není přístupná, protože parametry vysílání paketů se nastavují až v části vzdáleného ethernetovského modulu.

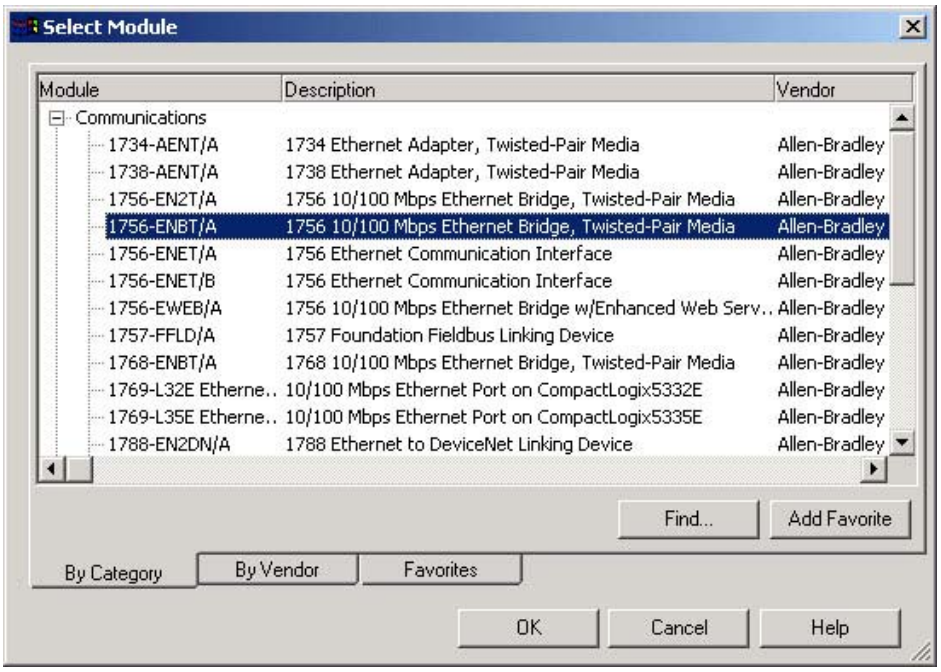
10) Nyní zpřístupníme vzdálený ethernetovský modul našemu projektu. Je to modul, který produkuje data a opět to je modul 1756-ENBT/A.

a) Klikneme pravým tlačítkem na lokální ethernetovský modul 1756-ENBT/A a zvolíme přidat nový modul (New module).



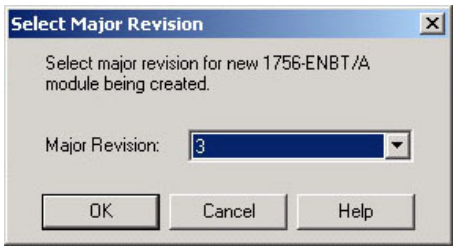
Obr. A.23 Přidání vzdáleného komunikačního modulu

b) V nově otevřeném okně volby modulu (Select Module) najdeme ethernetovský modul 1756-ENBT/A.



Obr. A.24 Volba vzdáleného komunikačního modulu

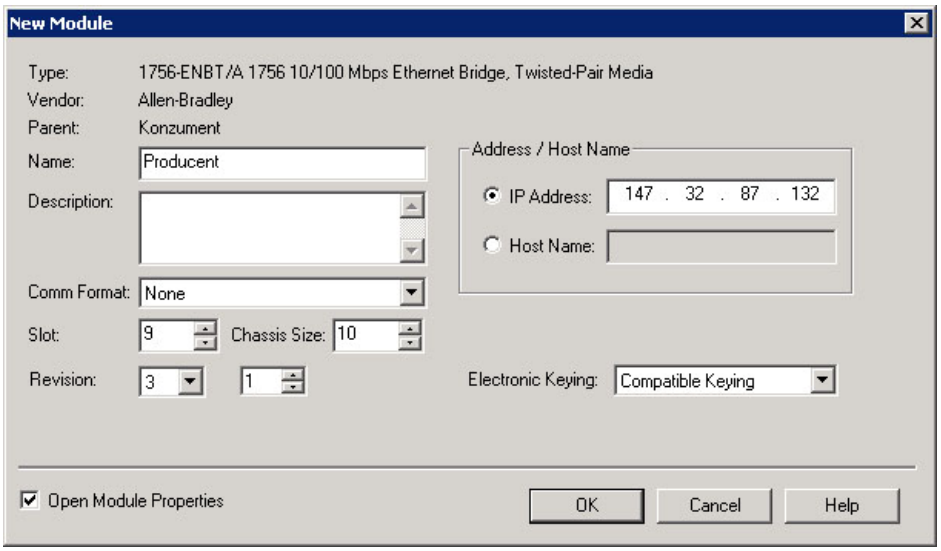
c) Jakmile potvrdíme volbu modulu, musíme opět zadat hlavní revizi firmwaru (Major revision). Postup je stejný jako v bodě [9.b] s tím rozdílem, že nyní se díváme v RSLinx na poskytovatele produkovaných dat, tedy na vzdálený ethernetový modul.



Obr. A.25 Volba hlavni revize

d) Vyplníme parametry 1756-ENBT/A (pozici, velikost rámu a IP adresu modulu), které zjistíme přes RSLinx. Dále zvolíme Electronic Keying jako Compatible module.

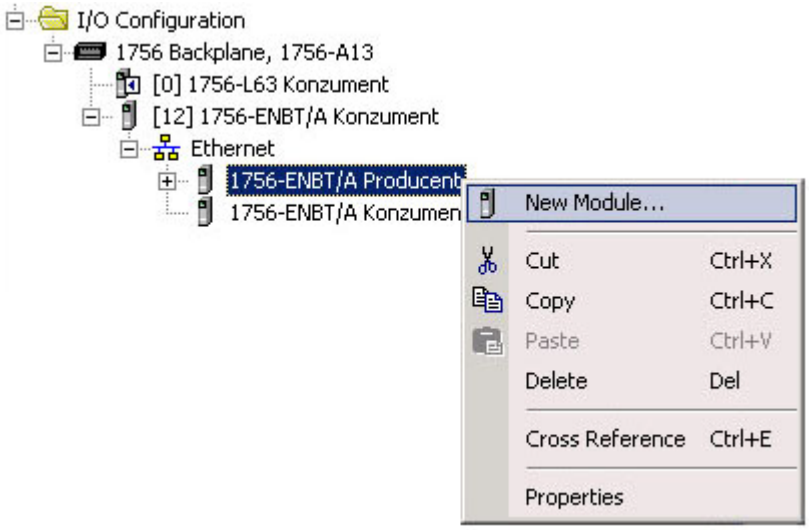
Minoritní revizi necháme 1. Dále je potřeba nastavit typ komunikace (Comm format). Nastavíme typ „None“. (Pzn. Typ Rack optimized je výhodné používat pouze při přenášení více digitálních I/O, aby se snížil počet připojení)



Obr. A.26 Nastavení parametrů komunikačního modulu

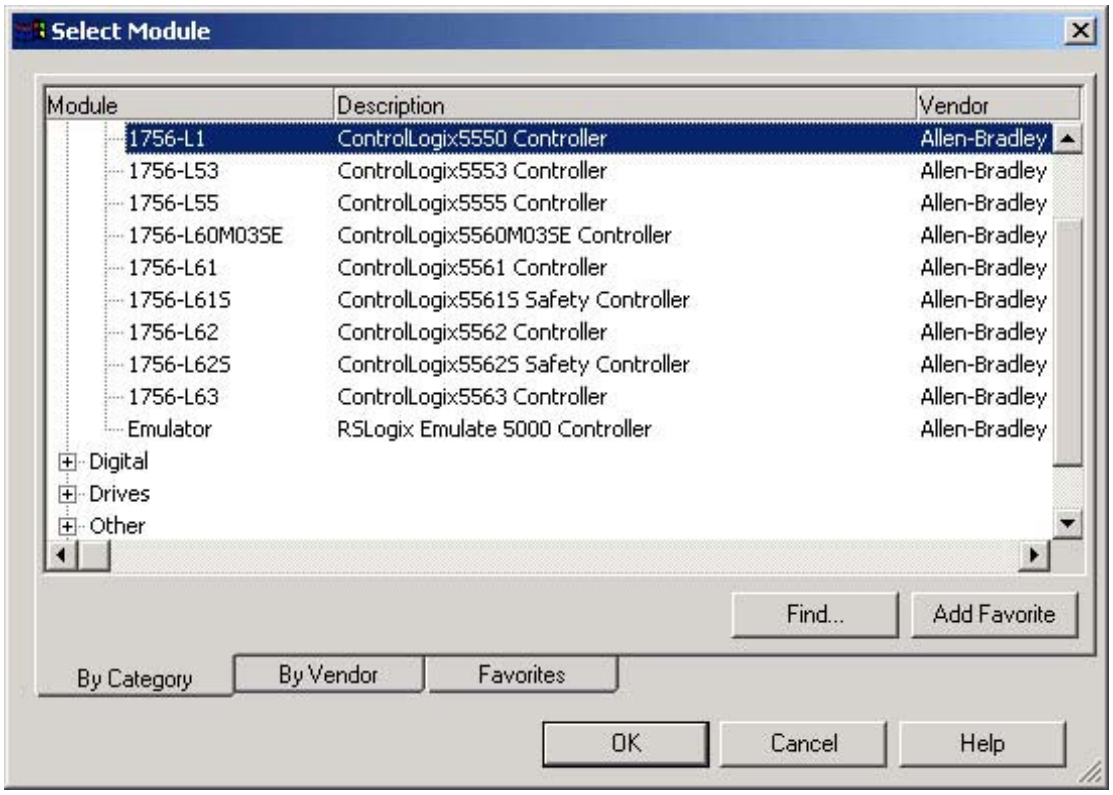
11) Nyní zpřístupníme vzdálený procesor, který poskytuje produkovaný tag, přes již zpřístupněný vzdálený ethernetový modul.

a) Klikneme pravým tlačítkem na vzdálený ethernetový modul 1756-ENBT/A a zvolíme přidat nový modul (New Module).



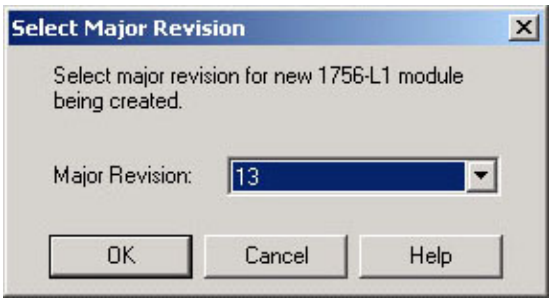
Obr. A.27 Zpřístupnění procesoru, ze kterého budeme číst produkovanou proměnnou

b) V nově otevřeném okně volby modulu (Select Module) najdeme vzdálený procesor 1756-L1, který přenáší produkované data.



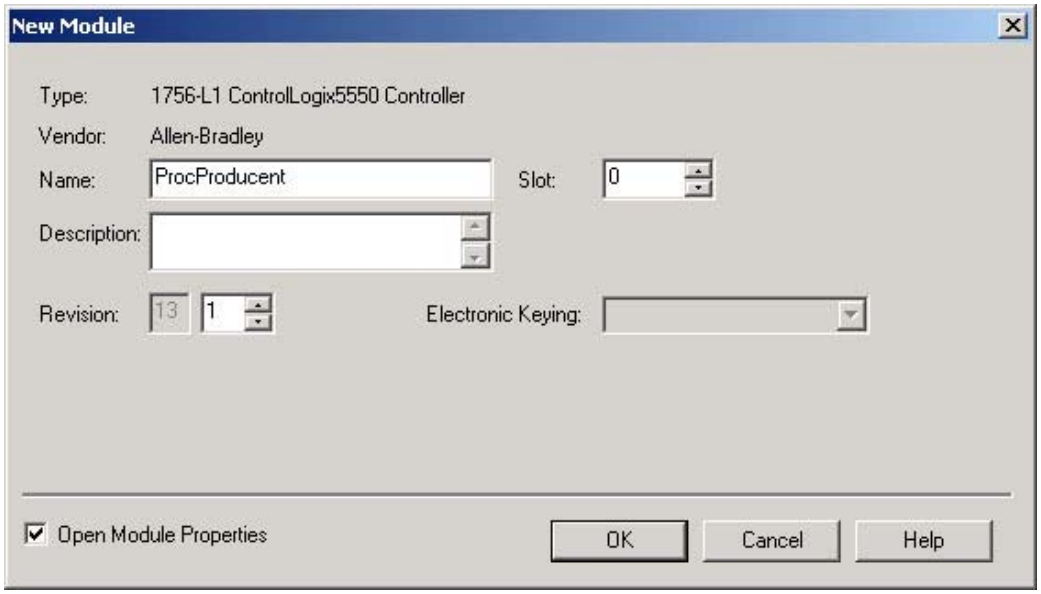
Obr. A.28 Zvolení odpovídajícího typu procesoru

c) Jakmile potvrdíme volbu modulu, musíme opět zadat hlavní revizi firmwaru (Major revision). Postup je stejný jako v bodě [9.b] s tím rozdílem, že nyní se díváme v RSLinx na producenta dat, tedy na vzdálený procesor.



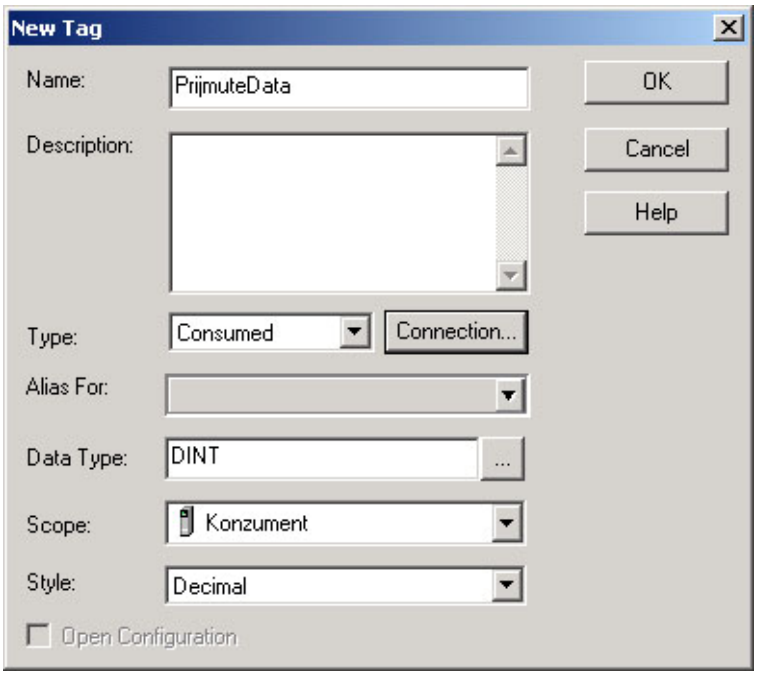
Obr. A.29 Volba hlavní revize přidávaného procesoru

d) V nově otevřeném okně zvolíme jméno procesoru a nastavíme pozici (slot), ve kterém se nachází. Pozici v rámu opět zjistíme přes RSLinx.



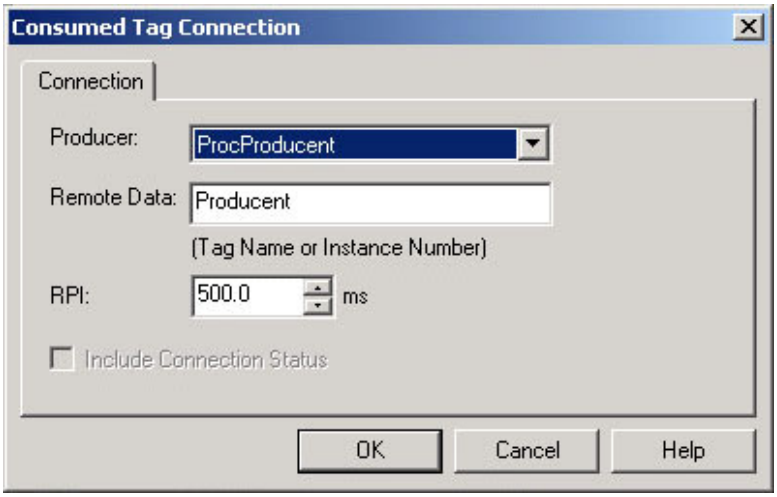
Obr. A.30 Nastavení parametrů přidávaného procesoru

12) Vytvoříme proměnnou (tag), do které se budou ukládat příchozí data. Klikneme na Controller Tags pravým tlačítkem a vybereme New Tag.
   a) Vyplníme Name, Data Type, Scope, Tag Type.
   b) Name je jméno proměnné.
   c) Tag type určuje typ proměnné a v našem případě se jedná o typ proměnné, do které se získané data ukládají (Consumed).
   d) Data Type určuje datový typ proměnné.
   e) Scope určuje umístění (dostupnost) proměnné. Vytvoříme Controller tag (globální proměnnou), do které lze data ukládat a navíc je přístupná ve všech podprogramech (routine).



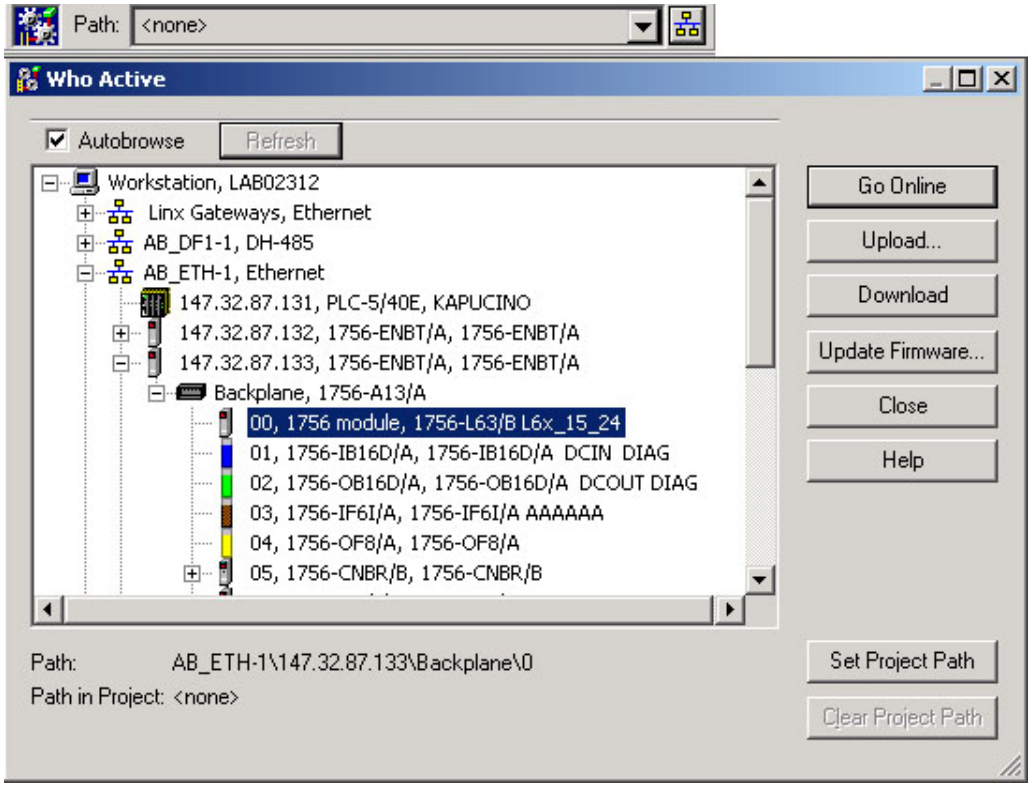
Obr. A.31 Vytvoření nové proměnné typu konzment

f) Dále klikneme na tlačítko Connection a vybereme jméno procesoru, který produkuje data. Rovněž vyplníme pole vzdálené data (Remote Data), ve kterém musíme uvést přesný název produkované proměnné. RPI nastavíme stejné jako v bodě [10.e] tedy 500ms.



Obr. A.32 Napojení vytvořeného tahu na vzdálený produkovaný tag

13) Zadáme cestu (Path), kam se námi vytvořený projekt nahraje (Download). Klikneme na ikonu Who active. V nově otevřeném okně se nám objeví dostupné cesty. Vybereme zařízení s odpovídající IP adresou, pro které jsme projekt vytvářeli a projdeme stromovou strukturou až k procesoru, do kterého chceme vytvořený projekt nahrát. Po označení procesoru se zpřístupní tlačítko Set Project Path, kterým cestu nastavíme.



Obr. A.33 Nastavení cesty projektu

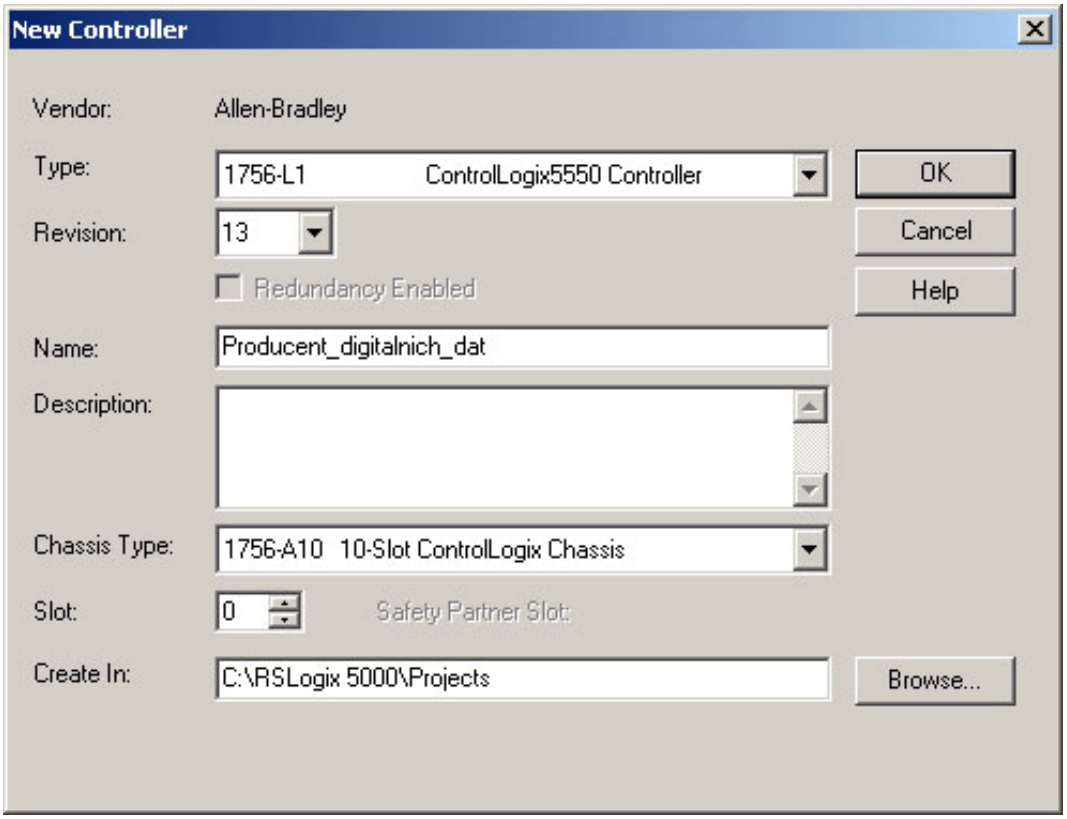
14) Klikneme v menu na položku komunikace (Communications) a vybereme nahrát (Download).
15) Nyní je v automatu s odpovídající adresou nahrán program, který bude produkovat data pro ostatní zařízení.
16) Je třeba ručně (klíčkem na procesoru) nebo softwarově uvést automat do RUN MODu (tím se program začne vykonávat).

A.3 Komunikace producent/konzument 1756-ENBT, přenášení digitálních I/O

Produkování digitálních vstupů a výstupů

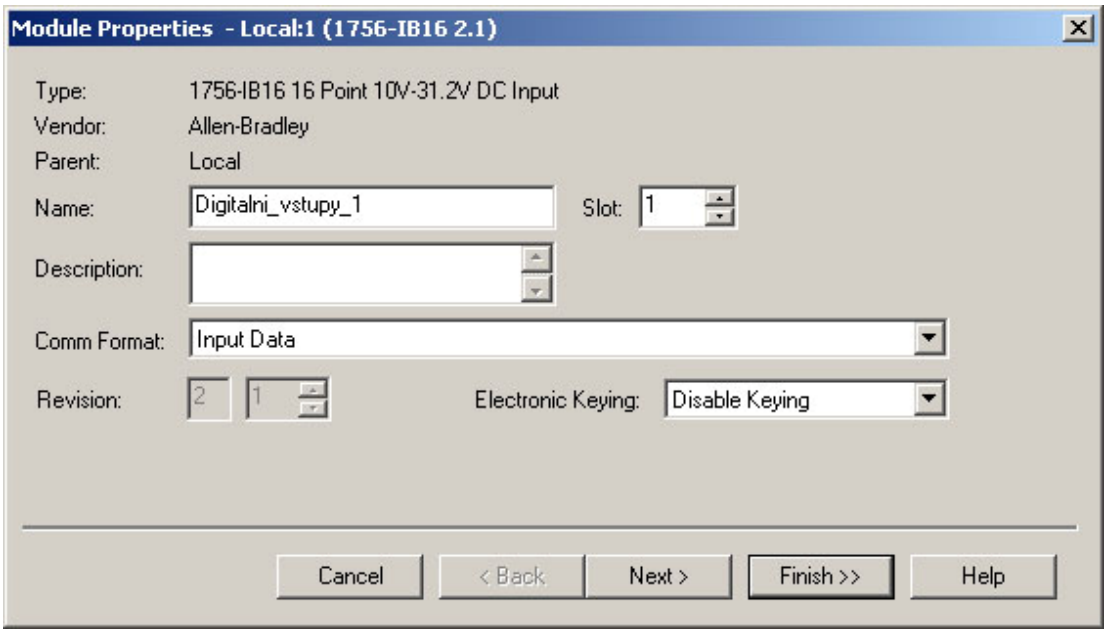
Předpokládá se znalost práce v programech RSLinx a RSLogix 5000 ukázané v předchozích dokumentech Vytvoření producenta a Vytvoření konzumenta.

1. Spustíme RSLinx.
2. Zjistíme potřebné parametry procesoru nezbytné pro založení nového projektu.
3. Spustíme RSLogix 5000 a založíme nový projekt.



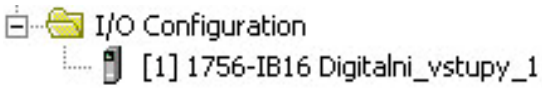
Obr. A.34 Založení nového projektu

4. Do I/O konfigurace přidáme modul digitálních vstupů 1756-IB16/A, jehož vstupy se budou produkovat. Formát komunikace (Comm Format) nastavíme na typ vstupních dat (Input Data).



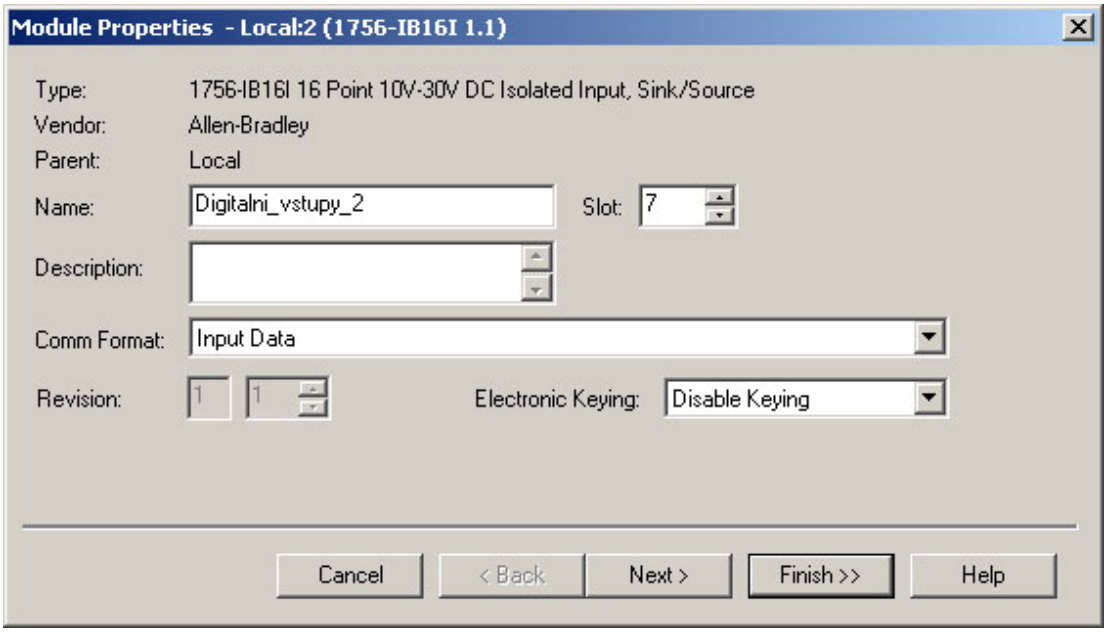
Obr. A.35 Přidaní digitálních vstupů do I/O konfigurace

5. V nově otevřeném okně vlastností modulu nastavíme požadovaný interval odesílání paketů o stavu digitálních vstupů (Requested Packet Interval).Kliknutím na dokončit (Finish) se nám přidá modul digitálních vstupů do I/O konfigurace a ta bude vypadat následovně.



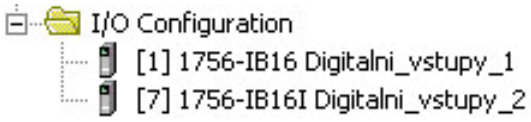
Obr. A.36 I/O konfigurace po přidání digitálních vstupů

6. Do I/O konfigurace přidáme další modul digitálních vstupů 1756-IB16I/A jehož vstupy se budou produkovat. Formát komunikace (Comm Format) nastavíme na typ vstupních dat (Input Data).



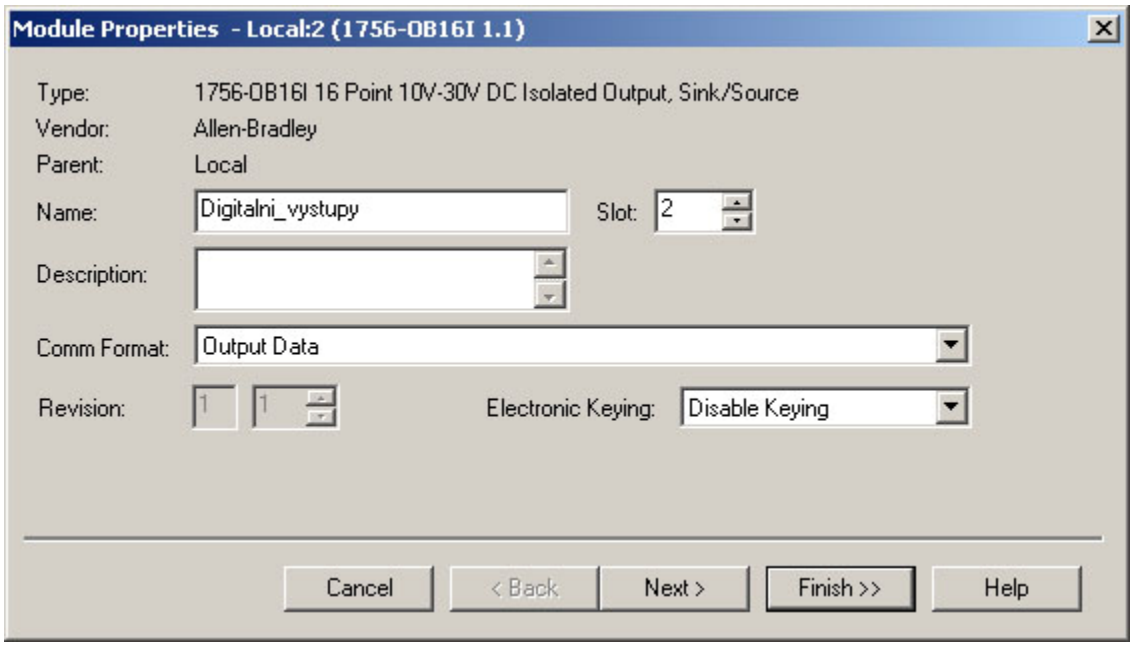
Obr. A.37 Přidání dalších digitálních vstupů

7. Nastavíme požadované RPI a klikneme na dokončit (Finish). I/O konfigurace nyní vypadá následovně.



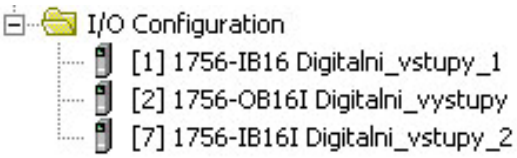
Obr. A.38 I/O konfigurace

8. Do I/O konfigurace přidáme modul digitálních výstupů 1756-IB16/A jehož výstupy se budou produkovat. Formát komunikace (Comm Format) nastavíme na typ výstupních dat (Output Data).



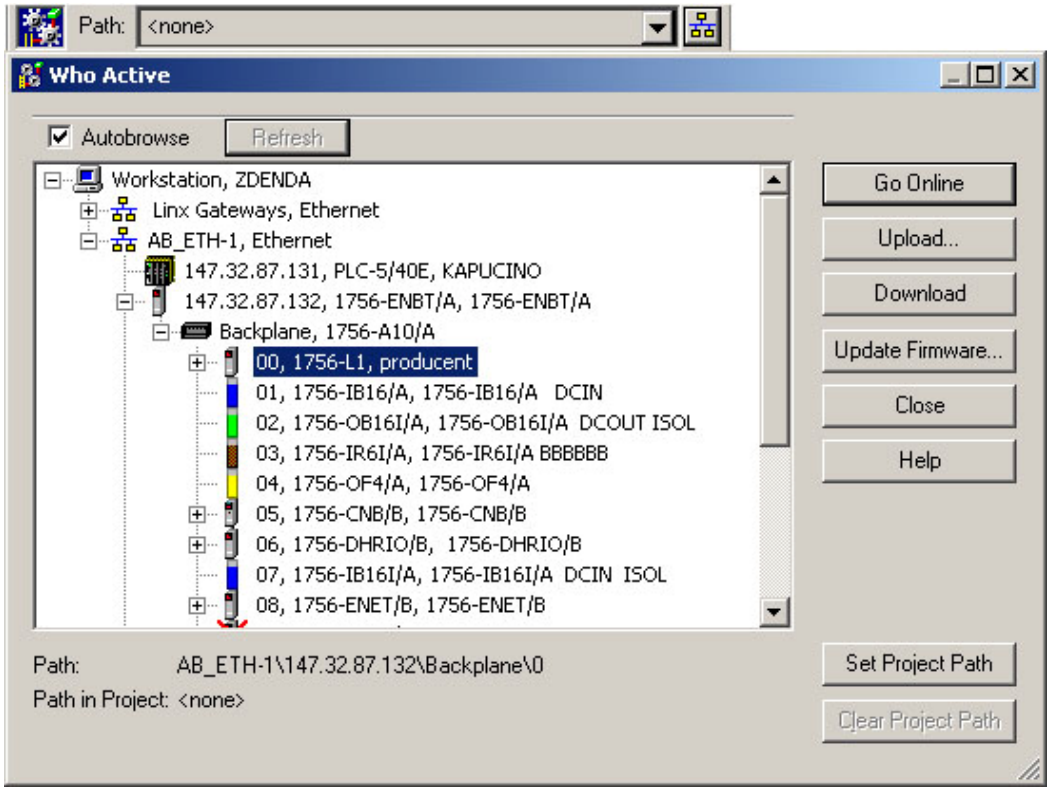
Obr. A.39 Přidání digitálních výstupů do I/O konfigurace

9. Nastavíme požadované RPI a klikneme na dokončit (Finish). I/O konfigurace nyní vypadá



Obr. A.40 I/O konfigurace

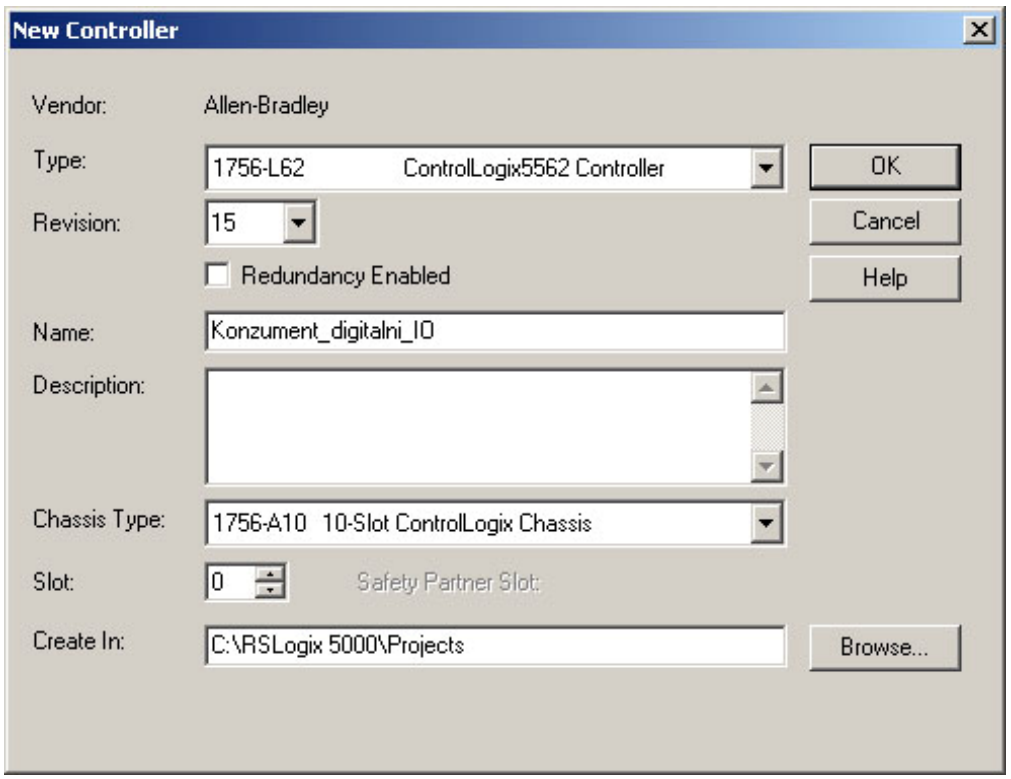
10. Nastavíme cestu projektu a nahrajeme vytvořený program do automatu.



Obr. A.41 Nastavení cesty

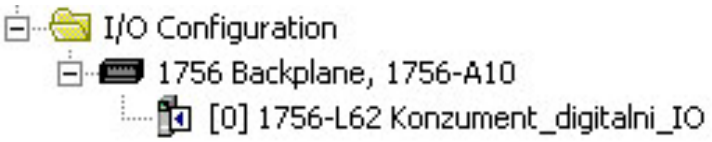
Konzumování digitálních vstupů s výstupů

1. Spustíme RSLinx
2. Zjistíme potřebné parametry procesoru nezbytné pro založení nového projektu.
3. Spustíme RSLogix 5000 a založíme nový projekt.



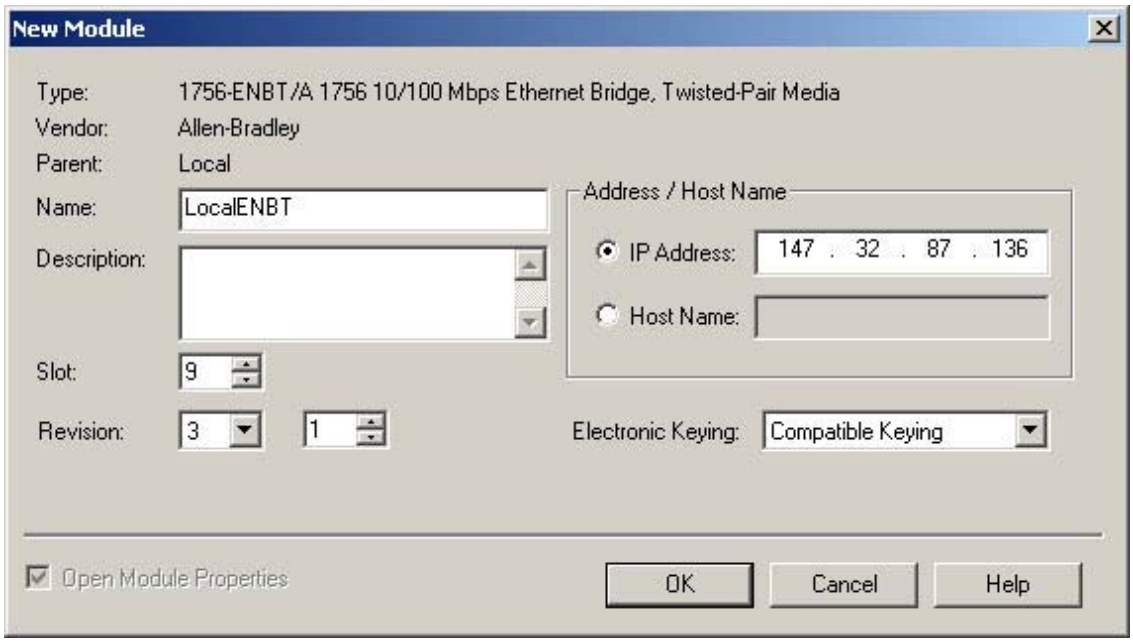
Obr. A.42 Vytvoření nového projektu

4. Na rozdíl od I/O konfigurace RSLogix 5000 ve verzi 13 je ve verzi 15 automaticky do této části začleněn procesor. Procesor se přidá automaticky po založení nového projektu.



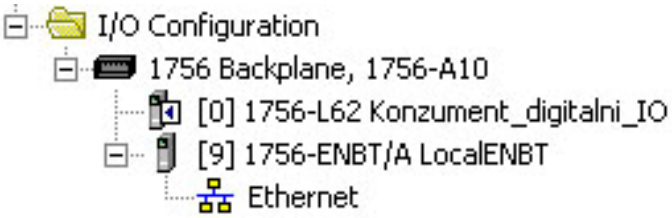
Obr. A.43 I/O konfigurace

5. Do této konfigurace přidáme komunikační modul 1756-ENBT/A, přes který budeme komunikovat se vzdáleným automatem.



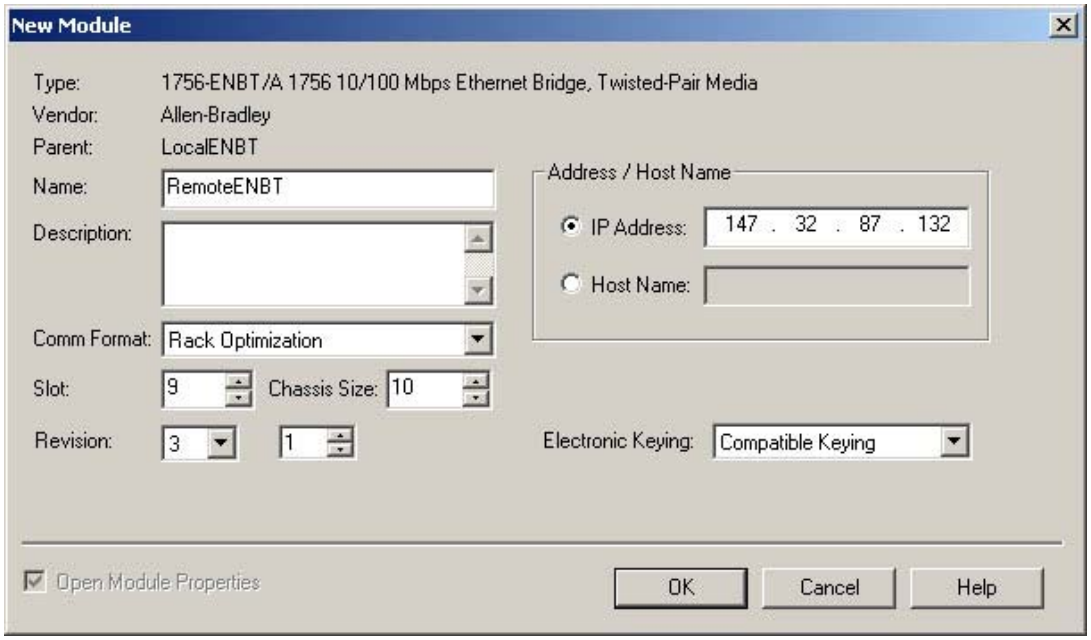
Obr. A.44 Přidání komunikačního modulu

6. I/O konfigurace po nastavení modulu vypadá následovně



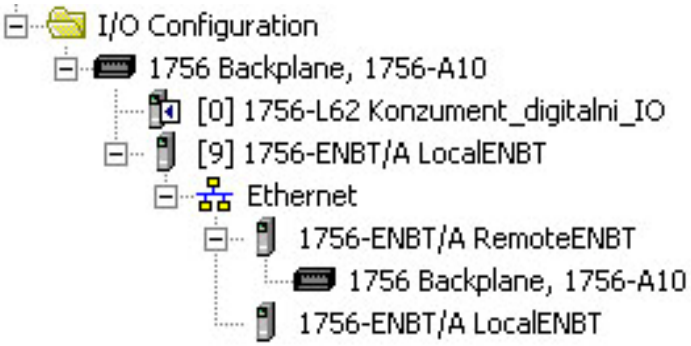
Obr. A.45 I/O konfigurace po přidání komunikačního modulu

7. Přidáme komunikační modul vzdáleného automatu 1756-ENBT/A, ve kterém se nacházejí digitální vstupy a výstupy, které chceme přijímat. Nastavíme komunikační formát jako síťově optimalizovaný (Rack Optimization).



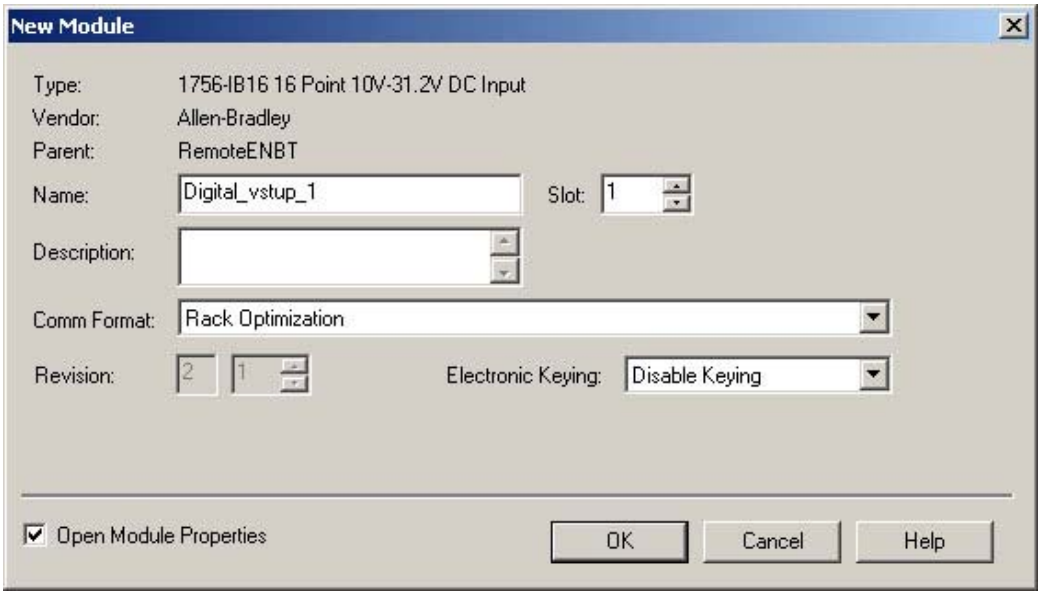
Obr. A.46 Přidání komunikačního modulu

8. Po potvrzení nastavených parametrů nově vytvářeného modulu (New Module) se otevře nové okno s možností upřesnění nastavení (Module Properties). V tomto okně zadáme RPI (Requested Packet Interval), kterým určíme interval přenášení dat mezi těmito moduly. I/O konfigurace po přidání tohoto modulu vypadá následovně.



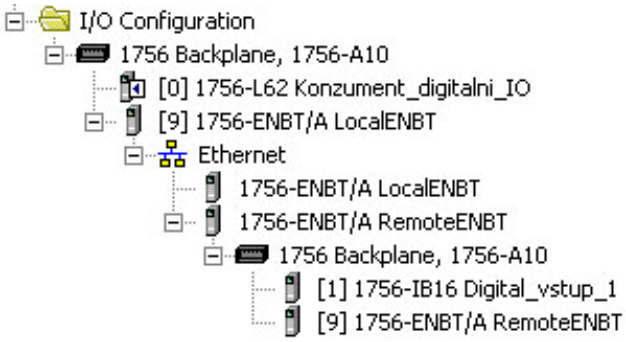
Obr. A.47 I/O konfigurace

9. Přidáme digitální vstupy 1756-IB16/A, které chceme číst vzdáleným automatem. Formát komunikace nastavíme jako síťově optimalizovaný (Rack Optimization).



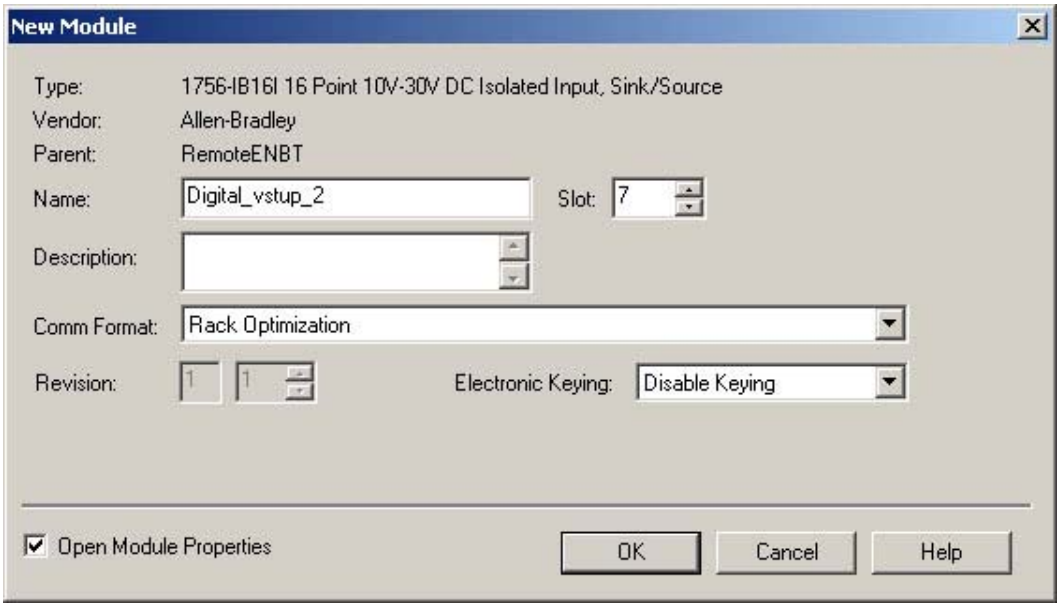
Obr. A.48 Přidání digitálních vstupů

10. Po přidání digitálních vstupů I/O konfigurace vypadá následovně.



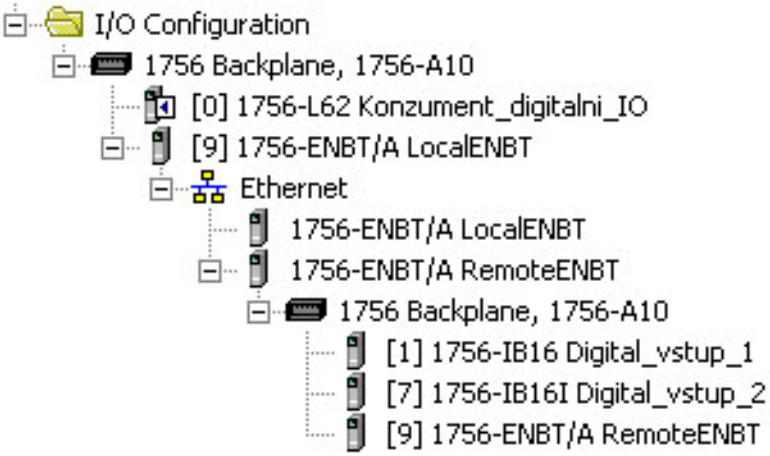
Obr. A.49 I/O konfigurace po přidání digitálních vstupů

11. Přidáme digitální vstupy 1756-IB16I/A, které chceme číst vzdáleným automatem. Formát komunikace nastavíme jako síťově optimalizovaný (Rack Optimization).



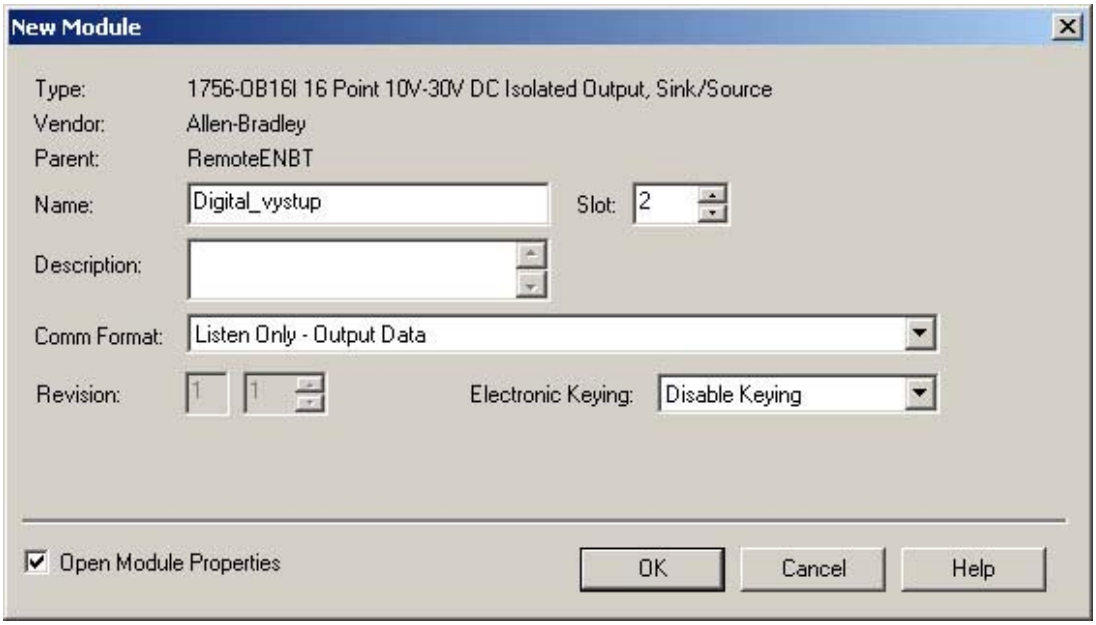
Obr. A.50 Přidání dalších digitálních vstupů

12. Po přidání digitálních vstupů I/O konfigurace vypadá následovně.



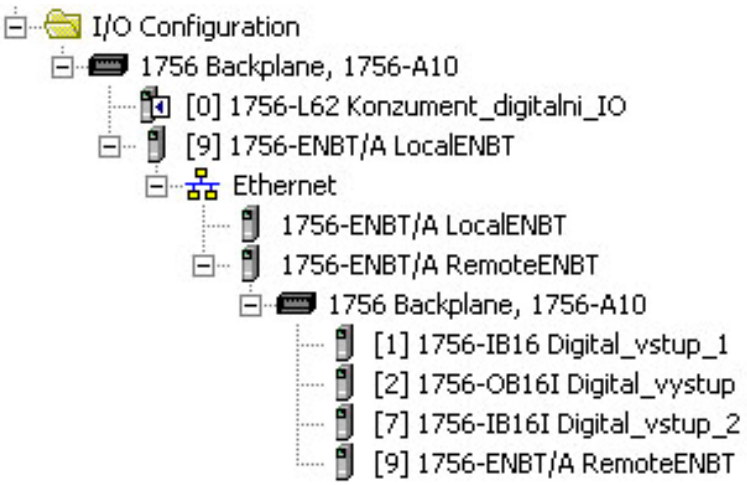
Obr. A.51 I/O konfigurace

13. Přidáme digitální výstupy 1756-OB16/A, které chceme číst vzdáleným automatem. Formát komunikace nastavíme jako typ výstupních dat v režimu naslouchání (Listen Only – Output Data).



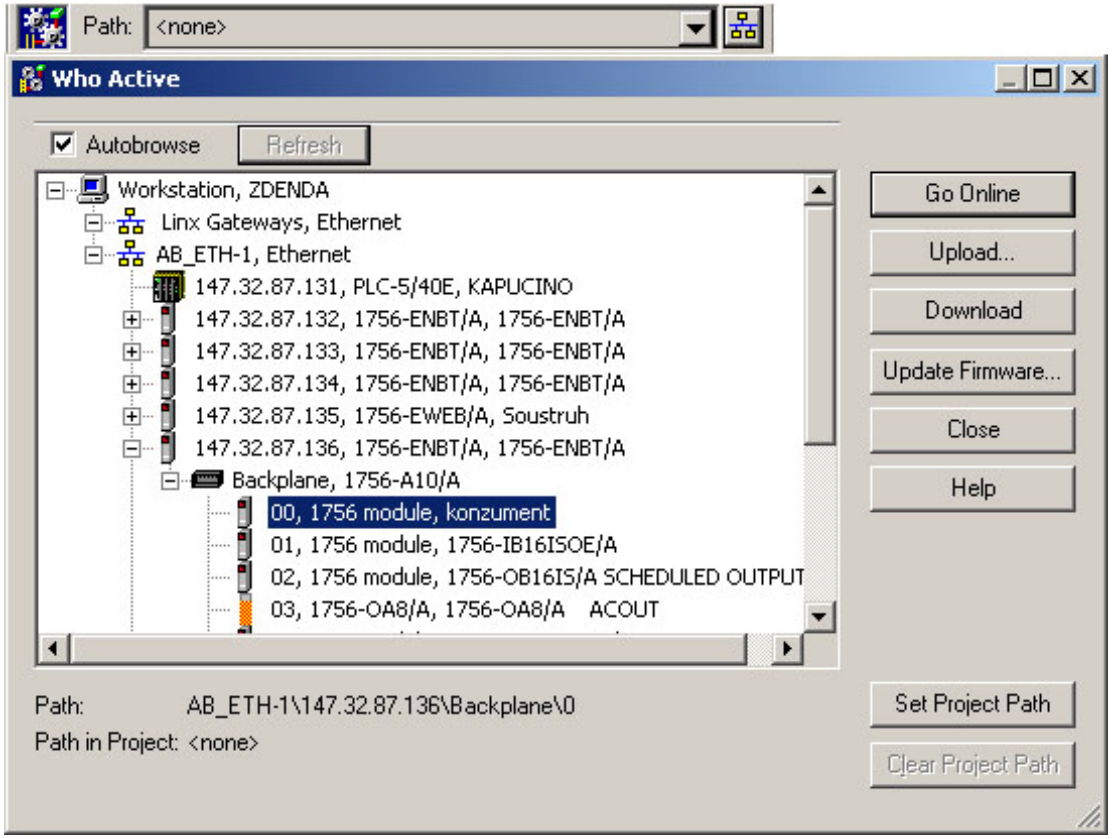
Obr. A.52 Přidání digitálních výstupů

14. Po potvrzení nastavených základních parametrů přidávaného modulu (New Module) se otevře nové okno s upřesňujícími volbami (Module Properties). V tomto okně zadáme RPI (Requested Packet Interval), kterým určíme interval přenášení digitálních výstupů. Po přidání digitálních výstupů I/O konfigurace vypadá následovně.



Obr. A.53 Kompletní I/O konfigurace

15. Nastavíme cestu projektu a nahrajeme vytvořený program do automatu.



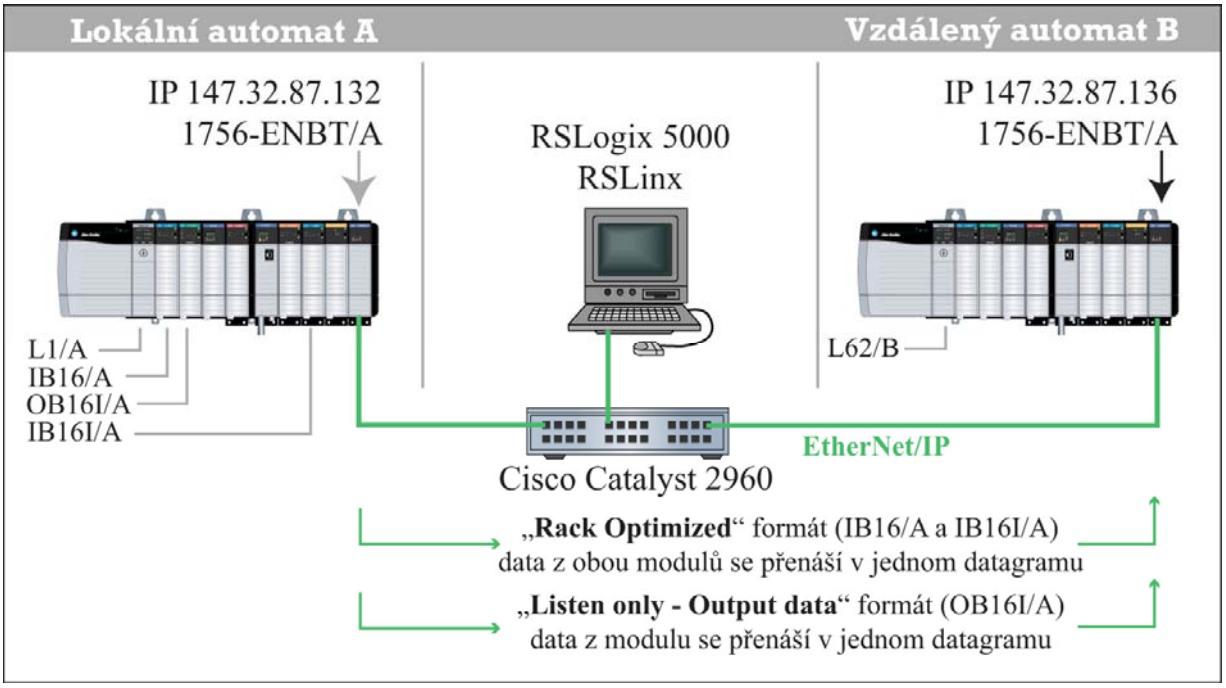
Obr. A.54 Zadání cesty pro nahrání projektu

Komunikace mezi automaty

Lokální automat A

Nastavení RPI pro moduly zapojené ve stejném rámu, určuje interval přenášení paketů mezi moduly (IB16/A, OB16/IA, IB16I/A) a procesorem (L1/A) po sběrnici ContolBus. Nastavení formátu komunikace, určuje způsob práce s moduly. Nastavíme-li formát typu „Input Data“ pro digitální vstupy, nebo „Output data“ pro digitální výstupy, určíme tím, že moduly budou vlastněny nakonfigurovaným procesorem (L1/A). Určíme-li typ komunikace

jako „Listen Only – Input Data“ pro digitální vstupy a „Listen Only – Output data“ pro digitální výstupy, říkáme, že procesor (L1/A) moduly nevlastní. Pouze naslouchá resp. čte data, které modul poskytuje. V tomto případě je nutné nastavit stejné RPI jako má vlastník daného modulu.

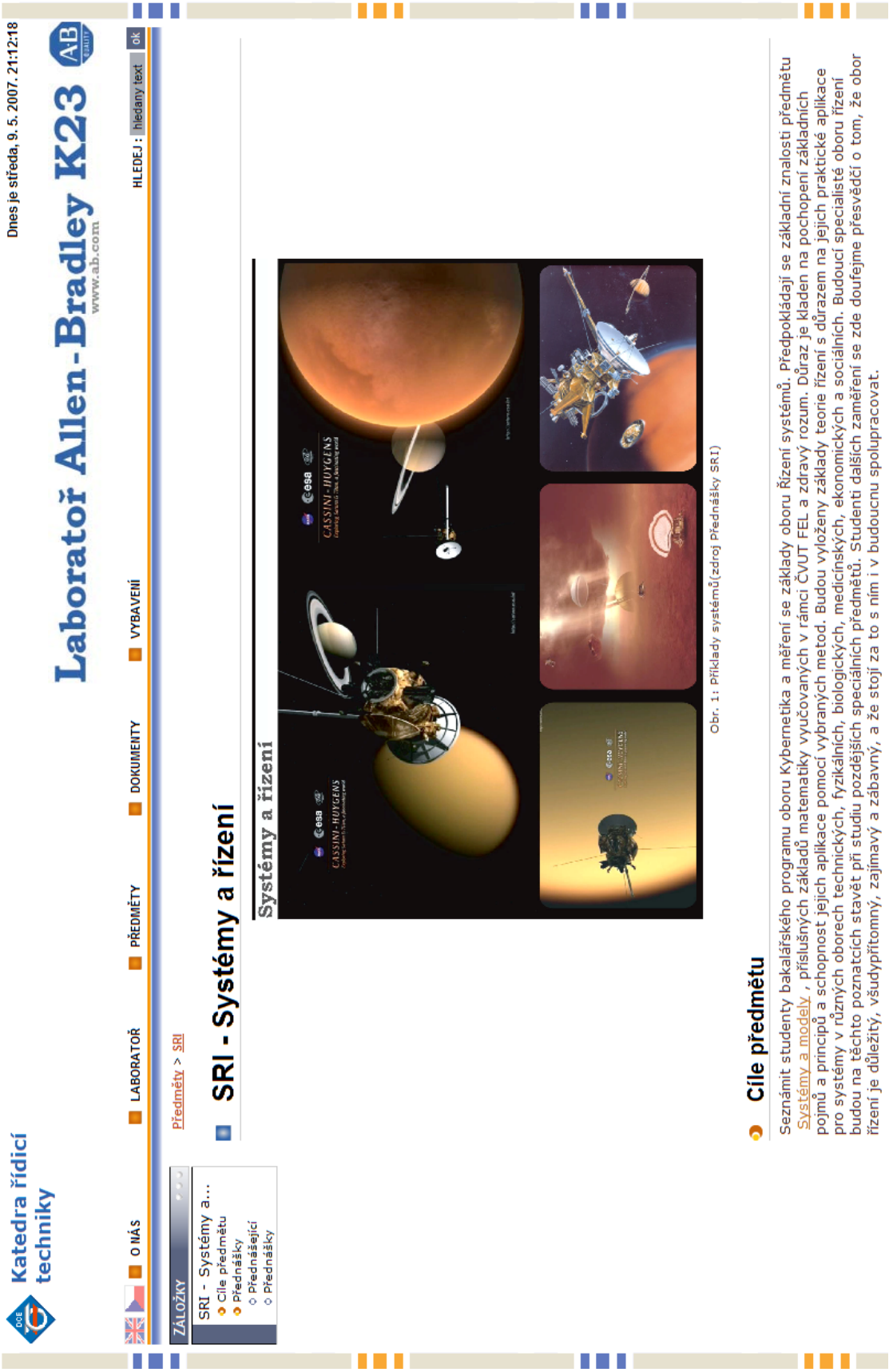


Obr. A.55 Zadání cesty pro nahrání projektu

Vzdálený automat B

Nastavené RPI určuje interval v jakém budeme číst vzdáleným automatem data z lokálního automatu přes Ethernet/IP. Čtení(zápis) dat může být buď optimalizovaný nebo neoptimalizovaný. To určujeme volbou síťově optimalizované spojení (Rack Optimization) na komunikačním modulu ENBT/A. Digitální vstup, či výstup, který má v I/O konfiguraci za otce tento ENBT/A modul, získá možnost komunikovat tímto optimalizovaným formátem. To znamená, že data z modulů, které jsou nakonfigurovány pro optimalizovaný přenos, jsou přenášeny najednou v jednom rámci (datagramu). Při normálním typu spojení je pro každý digitální modul otevřeno jedno samostatné spojení, což zvyšuje zatížení sítě.

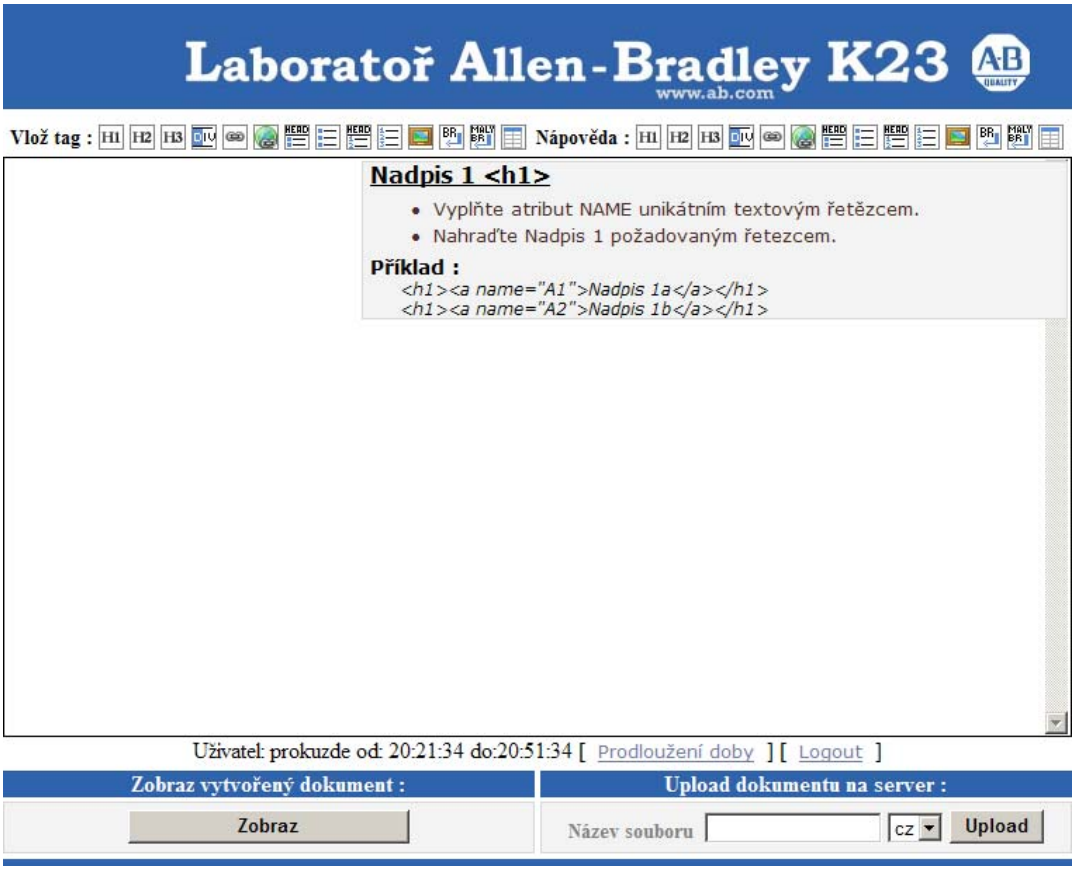
Příloha B – Webové stránky Laboratoře Allen-Bradley K23



Obr. B.1 Vzhled stránek laboratoře

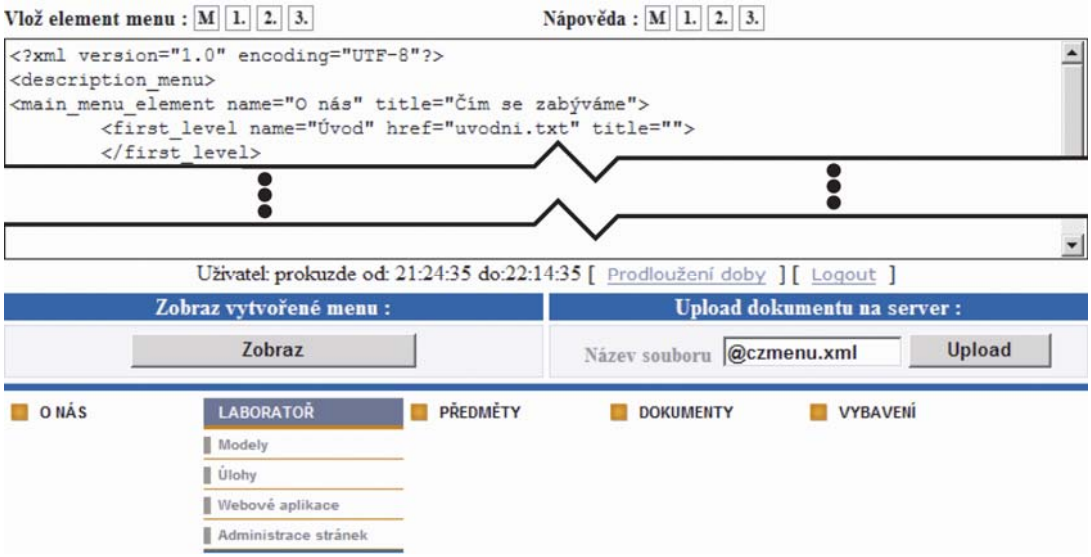
Příloha C – Administrativní rozhraní stránek

C.1 Administrace webových stránek



Obr. C.1 Nástroj na tvoření webových stránek

C.2 Administrace menu webových stránek



Obr. C.2 Nástroj na úpravu menu

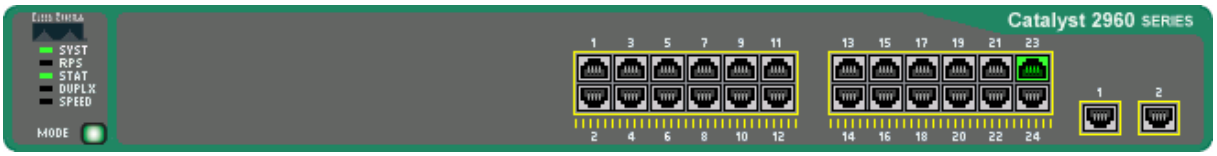
Příloha D – Zprovoznění přepínače Cisco Catalyst 2960

D.1 Bootovací proces

Zavaděč (BootLoader) poskytuje přístup k systému souborů (Flash File System), před tím, než se nahraje operační systém a předá kontrolu procesoru (CPU). Zavaděč provádí nahrání, dekompresi a spuštění operačního systému z obrazu systému (image). Zavaděč umožňuje v případě nefunkčnosti operačního systému provést tzv. trap-door (vstup zadními vrátky). Při této akci se provede formát souborového systému a následná reinstalace operačního systému z obrazu systému. Tento postup umožňuje zotavení ze zapomenutého hesla.

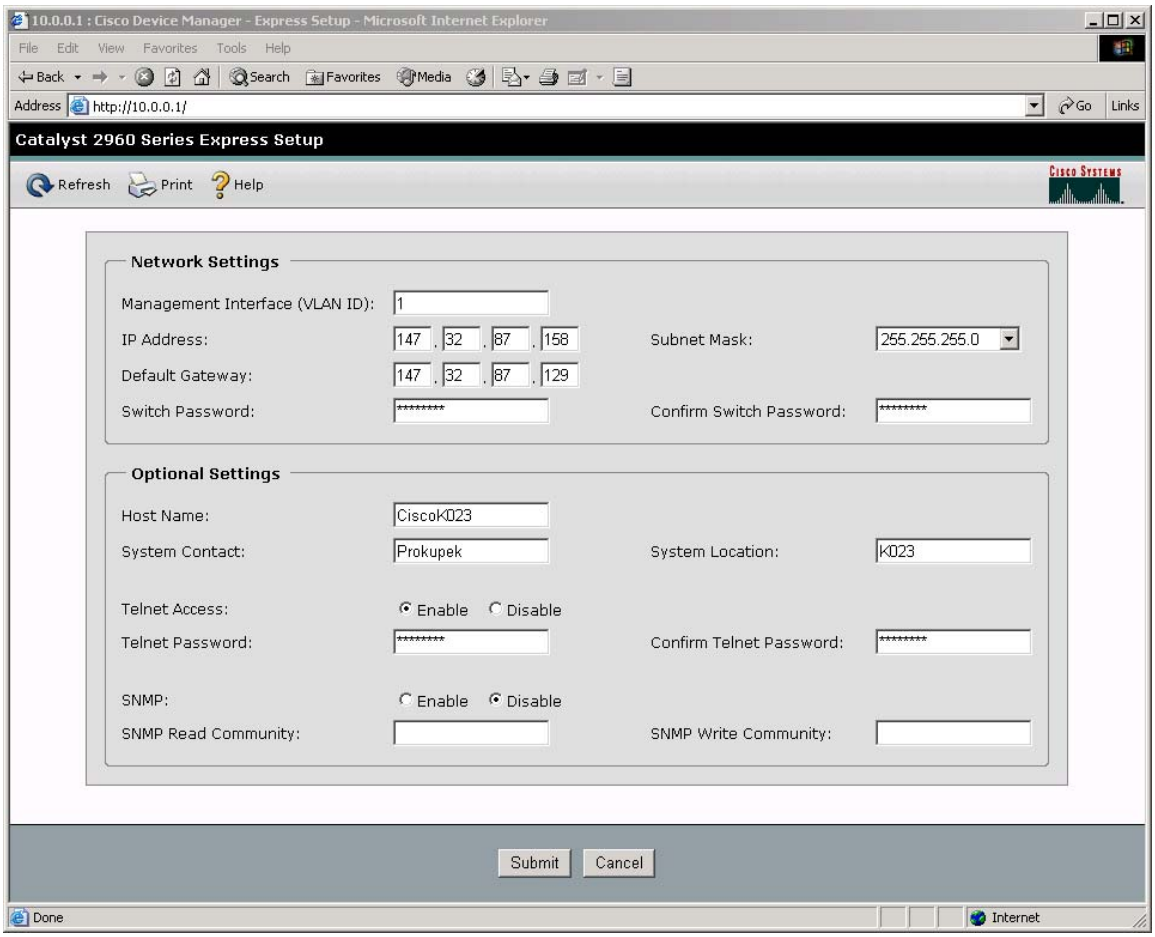
D.2 Zprovoznění přepínače Cisco Catalyst 2960

1) Provedeme restart přepínače, pro případ, že byl již konfigurován. Podržíme 7 sekund tlačítko Mode.
2) Počkáme do ukončení práce zavaděče. Dokud se trvale nerozsvítí LED dioda SYST na předním panelu přepínače.



Obr. D.1 Přední panel přepínače

3) Na tři sekundy podržíme tlačítko Mode. Rozsvítí se spodní tři LED diody. Nyní je přepínač připraven pro komunikaci s počítačem.
4) Zapojíme síťový kabel do libovolného portu přepínače a do sítového zařízení v počítači. Na počítači provedeme nastavení vlastností síťového připojení, nastavíme získání IP adresy z DHCP serveru.
5) Na počítači otevřeme internetový prohlížeč a zadáme do panelu adresy http://10.0.0.1. Spustí se Express Setup přepínače, který nám umožní provést základní nastavení přepínače.



Obr. D.2 Express Setup pro základní nastavení přepínače

6) Vyplníme požadované údaje a odešleme tlačítkem Submit přepínači. Přepínač zpracuje zaslané data a pokusí se nás přesměrovat na stránky Device Managera. Pokud byla nastavená IP adresa stejného rozsahu jako IP adresa počítače, přesměrování se zdaří a můžeme začít s rozšířenou konfigurací přepínače.

Příloha E – Základní popis funkcí přepínače Cisco

VTP (VLAN Trunking Protocol)

Umožňuje provádět centrální nastavení VLAN a jejich nasledné sdílení přes trunk protokol. Zabraňuje vzniku nekonzistentností v rozsáhlých sítích.

Trunking protokol

Je to P2P (point-to-point) spojení mezi dvěma ethernetovými přepínači nebo mezi jinými zařízeními jako je např. směrovač (router). Umožňuje rozšířit VLAN (Virtual LAN) přes celou síť.

Módy:

a) ***Access mód (port)***

Označený paket je na access portu (IEEE 802.1Q tagged) zahozen a MAC adresa není zapamatována. Dva přístupů

i) Statický přístup k portu

ii) Dynamický přístup k portu – port zařazen do VLAN podle VLAN Membership Policy Server VMPS (naše Cisco toto nepodporuje)

b) ***Trunk mód (port)***

Přenáší trafik z násobných VLAN přes jeden port a je členem všech VLAN ve VLAN Databázi. Podporuje průběžné značení a odznačování paketů.

c) ***Nonegotiate mód (port)***

Zabraňuje rozhraní (portu) generovat DTD (Dynamic Trunk Protocol) rámec, jehož prostřednictvím se domlouvají zažízení na VTP.

EtherChannel Port Group

Vytvoření logické skupiny, která se chová jako jeden port. Používá se pro zajištění komunikace mezi přepínači či mezi přepínačem a serverem. Komunikace se rozděluje na porty kanálu. Pokud nějaký port selže, pak je komunikace, která byla směrována, přes tento port rozdělena na zbývající funkční porty.

Flow control IEEE 802.3x

Umožňuje či zakazuje přijímání resp. odesílání brzdícího rámce (pause frame). Ten se odesílá v případě přetížení portu a způsobuje přerušení komunikace, dokud se neuvolní prostředky pro vyřízení další komunikace.

Auto-MDIX (Medium Dependent Interface Crossover)

Zjišťování typu kabelu (přímého či kříženého) a následné nastavení portu pro korektní komunikaci.

MTU (Maximum Transmission Unit)

Nastavení velikosti komunikačního rámce (pro všechna rozhraní). Pro náš typ přepínače můžeme nastavit velikost rámece pro 10/100Mbit a 1GBit porty zvlášť. Pro 1GBit porty se nastavuje tzv. jumbo frames od 1500 do 9000 bajtů. Obvyklá velikost rámce pro 10/100Mbit porty je 1500 bajtů. Může se pohybovat od 64bajtů do 1500bajtů.

Spanning Tree Protocol (STP)

Existují dva druhy STP protokolu. PVST+ protokol založený na IEEE 802.1D a rapid-PVST+ protokol založený na IEEE 802.1w. Pro správné fungování sítě je nutná síť bez smyček. Pokud by síť obsahovala smyčky, přepínač by mohl obdržet stejný paket dvakrát, a může se naučit dvakrát různou cestu. Takováto síť by pak byla nestabilní. V rozlehlých sítích se však existenci smyček obvykle nevyhneme, a proto se používá STP pro výběr nejlepší cesty. STP používá spanning-tree algoritmus k výběru jednoho přepínače v redundantní síti k výběru hlavního bodu (root) v STP. Algoritmus počítá nejlepší cestu bez smyček, tak že přidělí role každému portu v aktivní topologii.

Root – „forwarding“ port vybraný pro STP

Designated – „forwarding“ port vybraný pro každý přepínaný LAN segment

Alternace – blokovaný port poskytující alternativní cestu k root

Backup – blokovaný port ve zpětné smyčce (loopback)

Všechny porty označeny jako designated nebo backup, znamenají hlavní (root) přepínač. Alespoň jeden z portů označen designated, jedná se o vyhrazený přepínač (designated). Alternativní cesta se zvolí v případě nefunkčnosti předchozí cesty. STP vysílá a přijímá tzv. BPDU rámce (Bridge Protocol Data Units) v pravidelných intervalech. Tyto rámce se nepřesměrovávají, ale používají se pro vytvoření cesty bez smyček. Obsahují informace o přepínači, jeho portech, včetně MAC adres, prioritu přepínače a cenu cesty.

Port Fast

Provádí okamžitý převod rozhranní nakonfigurovaného jako Access či Trunk port do „forwarding“ stavu ze stavu blokování. Provede okamžité spojení do sítě. Při Port Fast se vynechávají počáteční kroky naslouchání a učení. Nastavení Port Fast je vhodné používat pouze tam, kde nehrozí výskyt smyček, jinak tato vlastnost může způsobit nestabilitu sítě.

BPDU Guard

Po příchodu BPDU Guard rámce na rozhraní, je rozhraní zakázáno a vyžaduje se pro jeho zprovoznění zásah administrátora. Pomocí BPDU rámců jsme schopni zajistit odstranění případných problémů na síti. Pokud je povolen Port Fast je vhodné BPDU Gard ponechat zapnuté. Umožňuje ostatním zařízením, pokud bude dané rozhraní působit potíže, odeslat BPDU rámec a toto rozhraní vypnout.

BPDU Filtering

Umožňuje zabránit na daném rozhraní přijímat a odesílat BPDU rámce. Pro všechny Port Fast porty, zapnutý BPDU filter, je to samé, jako vypnout STP na všech portech.

UplinkFast

Umožňuje rychlejší volbu root portu při selhání linky na přístupovém přepínači. Při nastavení UplinkFast neprobíhá celá procedura STP, ale náhradní port root se uvede do „forwarding“ stavu. Když STP hledá nový root port, zaplaví ostatní zařízení, která byla naučena na daném rozhraní, mutlicastovými pakety, aby získaly root port. Toto množství se dá omezit pomocí

max-update-rate. Je vhodné nakonfigurovat UplinkFast, sníží se tím doba zaplavovaní sítě multicastovými pakety.

BackBoneFast

Jedná se o doplňkovou technologii k UplinkFast, a umožňuje optimalizovat maximum-age timer, který kontroluje množství času přepínače, pro ukládání informací obdržených na rozhraní. Snaží se najít alternativní cestu v okamžiku, kdy přepínač na nižší úrovní odešle BPDU rámec root portu na hlavním přístupovém přepínači. Hlavní přepínač při obdržení BPDU rámce, zakáže tento port a přepínač přijde o spojení. Proto potřebuje najít nové.

EtherChanel Guard

Provádí kontrolu Etherchannel konfigurace mezi přepínačem a připojeným zařízením. Chyba může nastat, pokud parametry komunikačního kanálu nejsou na obou zařízeních správně nastaveny. Pro tento případ je port či rozhranní vypnuto pomocí EtherChannel Guard.

Root Guard

V síti může být zapojeno mnoho přepínačů, které nepatří SP (service provider). Pokud STP zvolí jako root přepínač zákaznický přepínač, což je nežádoucí, Root Guard uvede dané rozhraní do blokovaného stavu. Tímto způsobem se zabraňuje vzniku root zákaznického přepínače.

Loop Guard

Zabraňuje změně náhradního nebo hlavního portu stát se vlivem selhání, které vede na jednosměrnou linku, designated portem. STP zároveň neodesílá BPDU rámce na root či náhradní porty.

Flex Links

Jedná se pár rozhraní, portů, kde jeden port je zálohou druhého. Je to alternativa STP, takže pokud je STP povoleno, není Flex Links třeba. Pokud STP není zapnuto, jedna Flex linka je zálohou druhé, takže při selhání jedné z nich druhá přebírá kontrolu.

Multicast VLAN Registration (MVR)

Dovoluje jedné multicastové VLAN, aby byla sdílena v síti, zatímco účastníci zůstávají v oddělených VLAN. MVR poskytuje možnost průběžně posílat multicastový proud zařízením v různých VLAN, ale zároveň umožňuje izolovat multicastový proud od účastníků VLAN, kvůli snížení zatížení a bezpečnostním důvodům. MVR předpokládá úzkou spolupráci s IGMP zprávami (přihlašování do skupiny a odhlašování z ní). IGMP snooping a MVR fungují nezávisle na sobě. Jakmile je však MVR nakonfigurováno, reaguje pouze na přihlašovací a odhlašovací zprávy nakonfigurovaných pod MVR. Všechny ostatní IGMP zprávy jsou řízeny IGMP snooping. CPU přepínače indikuje MVR multicast proud a jejich asociované skupiny v „forwarding“ tabulce. Tato tabulka je aktualizována, při odstranění či začlenění účastníků multicastového proudu.

Módy MVR

a) compatible – multicastové data přijaté MVR hostem, jsou směrována na všechny MVR porty bez ohledu na MVR příslušenství těchto portů.

b) dynamic – multicastové data přijaté MVR hostem, jsou směrovány pouze z MVR zdrojových portů, které má MVR příhlášené buď pomocí IGMP zpráv nebo statickou konfigurací MVR.

Storm Control

Zabraňuje, aby komunikace byla narušena broadcast, multicast, unicast vysíláním. Provádí kontrolu počtu přijatých paketů těchto typů za interval 1 sekundy a porovnává jej s definovaným prahovým stupněm.

Použité metody měření komunikace

- procentuální vytížení aktuálního pásma proti celkovém pásmu definovaném na daném portu
- vytížení pakety/s z broadcastu, multicastu, unicastu
- vytížení bity/s

Protected Port

Některé aplikace vyžadují, aby mezi porty ne stejném přepínači nedocházelo k „forrwarding" komunikace z jednoho portu na druhý. Chráněný port tedy neprovádí směrovaní komunikace (broadcast, multicast, unicast) na jinší chráněný port. Jakákoliv komunikace, kromě paketů PIM (Protocol Independent Multicast) od CPU, musí být směřována pomocí směrovače (zařízení Layer 3). Směřování mezi chráněným a nechráněným portem probíhá normálně, není chráněna.

Port Blocking

Umožňuje nastavit blokování neznámé komunikace ze zařízení s neznámou MAC adresou na jinší porty přepínače. Blokuje se unicastová a multicastová komunikace.

Port Security

Umožňuje omezení přístupu k přepínači pomocí MAC adresy zařízení. Jestliže přidělíme MAC adresu (zadáme ji jako bezpečnou) k bezpečnému portu, port nebude směrovat pakety se zdrojovou bezpečnou MAC adresou ven ze skupiny definovaných adres. Jestliže přidělíme jednu bezpečnou adresu k portu, pak připojené zařízení má jistotu plné šíře komunikačního pásma. K bezpečnému portu se může připojit pouze zařízení s bezpečnou MAC adresou.

Secure MAC address

Bezpečná adresa lze nastavit třemi způsoby. Staticky, kdy je vyplněna tabulka bezpečných adres a nahrána do běžící konfigurace. Dynamicky, kdy jsou bezpečné adresy zjištěny po restartu a uloženy do dalšího restartu v běžící konfiguraci. Poslední způsob kombinuje statický a dynamický přístup. Při restartu se zjistí bezpečné MAC adresy a jsou následně uloženy do běžící konfigurace. Při pokusu o připojení zařízení s neznámou MAC adresou, můžeme nastavit několik možných reakcí (protected, restrict, shutdown).

Switched Port Analyzer (SPAN)

Používá se k analýze komunikace, které prochází přes porty nebo VLAN odesláním kopie komunikace na jiný port, na který se připojí síťový analyzátor nebo monitorovací zařízení. Při vytváření SPAN se musí zadat cílový port, na který se bude komunikace směřovat. Cílový

port, nesměřuje ani nepřijímá komunikaci. Komunikace, která vstupuje a odchází přes zdrojový port, či VLAN může být monitorována použitím SPAN. Komunikace, která se přenáší z jedné VLAN na zdrojovou VLAN, monitorována být nemůže. SPAN obvykle přenáší neoznačené pakety. Označené pakety BPDU (Bridge Protocol Data Unit), CDP (Cisco Discover Protocol), STP (Spanning Tree Protocol), PagP (Port Aggregation Protocol) nepřenáší, to lze však povolit změnou klíčového slova native na replicate při vytváření SPAN.

Příloha F – Rozbor přenášených dat při Rack Opitimization komunikaci

V jednom optimalizovaném rámci jsou přenášeny dva digitální moduly vstupů IB16.

V EtherNet/IP paketu jsou zapouzdřená následující data (část paketu Connected Data Item):

0020	01	60	08	ae	08	ae	00	70	1d	ce	02	00	02	80	08	00
0030	81	18	35	00	ea	00	00	00	b1	00	56	00	02	00	7d	ff
0040	ff	ff	ff	ff	ff	ff	00	00	00	00	**00**	**00**	**00**	**00**	**c0**	**84**
0050	**00**	**00**	ff	ff	ff	ff	00	00	00	00	ff	ff	ff	ff	00	00
0060	00	00	ff	ff	ff	ff	00	00	00	00	ff	ff	ff	ff	00	00
0070	00	00	ff	ff	ff	ff	00	00	00	00	**00**	**00**	**00**	**00**	**00**	**00**
0080	**00**	**00**	ff	ff	ff	ff	00	00	00	00	ff	ff	ff	ff	00	00
0090	00	00														

Rozbor:

02 00	-- počet přenášených zařízení
7d ff ff ff	-- status (popisuje které moduly jsou zapojeny)

Data	**Systémem vytvořená proměnná**	
ff ff ff ff	-- producent.I.slot[0].Fault	
00 00 00 00	-- producent.I.slot[0].Data	
00 00 00 00	-- producent.I.slot[1].Fault	
c0 84 00 00	-- producent.I.slot[1].Data	-- přenášený dig. vstup 1, slot modulu 1
ff ff ff ff	-- producent.I.slot[2].Fault	
00 00 00 00	-- producent.I.slot[2].Data	
ff ff ff ff	-- producent.I.slot[3].Fault	
00 00 00 00	-- producent.I.slot[3].Data	
ff ff ff ff	-- producent.I.slot[4].Fault	
00 00 00 00	-- producent.I.slot[4].Data	
ff ff ff ff	-- producent.I.slot[5].Fault	
00 00 00 00	-- producent.I.slot[5].Data	
ff ff ff ff	-- producent.I.slot[6].Fault	
00 00 00 00	-- producent.I.slot[6].Data	
00 00 00 00	-- producent.I.slot[7].Fault	
00 00 00 00	-- producent.I.slot[7].Data	-- přenášený dig. vstup 2, slot modulu 7
ff ff ff ff	-- producent.I.slot[8].Fault	
00 00 00 00	-- producent.I.slot[8].Data	
ff ff ff ff	-- producent.I.slot[9].Fault	
00 00 00 00	-- producent.I.slot[9].Data	

Příloha G – Multicastové skupiny využité pro komunikaci

Rozsahy adres, jak je přiděluje IANA (Internet Assigned Numbers Authority)

224.0.0.1 All Systems on this Subnet

224.0.0.2 All Routers on this Subnet

224.0.1.1 NTP Network Time Protocol

239.192.000.000-239.251.255.255 Organization-Local Scope

Přepínač odesílá IGMP žádost (Membership Query) na adresu 224.0.0.1. Pro IGMP odhlášení (Leave Group) se používá adresa 224.0.0.2.

Ip adresa třídy B

Rozsah decimálně : 128.0.0.0 – 191.255.255.255

Binárně : **10**xxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx

Ip adresy laboratoře

Decimálně : 147.32.87.x

Binárně : **10**010011.00100000.01010111.xxxxxxxx

Maska podsítě

Decimálně : 255.255.255.128

Binárně : 11111111.11111111.11111111**.1**0000000

Vyplívající rozsah adresa

Decimálně : 147.32.87.129 – 147.32.87.254

Binárně : 10010011.00100000.01010111**.1**0000001 až
10010011**.**00100000.01010111**.1**1111110

Broadcast: 10010011**.**00100000.01010111**.**11111111

Ip adresa třídy D

Rozsah : 224.0.0.0 – 239.255.255.255

Binárně : **1110**0000.00000000.00000000.00000000 až
11101111.11111111.11111111.11111111

Multicastové skupiny, které budou použity pro dané IP adresy v laboratoři:

IP adresa 1756-ENBT/A modulu	Multicastová skupina	Produkovaný Tag (proměnná)
147.32.87.130	239.192.1.0	1
	239.192.1.1	2
	:	:
	239.192.1.31	32
147.32.87.131	239.192.1.32	1
	:	:
	239.192.1.63	32
147.32.87.132	239.192.1.64	1
	:	:
	239.192.1.95	32
147.32.87.133	239.192.1.96	1
	:	:
	239.192.1.127	32

:

147.32.87.136	239.192.1.224	1
	:	:
	239.192.1.254	32
147.32.87.137	239.192.2.0	1
	:	:
	239.192.2.31	32

: