VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2019 Antonín Bohačík

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

SIMULACE KOMUNIKACE SCADA PROTOKOLŮ
SIMULATOR OF SCADA PROTOCOLS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **Antonín Bohačík**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. Petr Blažek**
SUPERVISOR

BRNO 2019

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní obor **Informační bezpečnost**
Ústav telekomunikací

Student: Antonín Bohačík ***ID:*** 195149
Ročník: 3 ***Akademický rok:*** 2018/19

NÁZEV TÉMATU:

Simulace komunikace SCADA protokolů

POKYNY PRO VYPRACOVÁNÍ:

Bakalářská práce je zaměřena návrh pracoviště, které bude simulovat komunikaci mezi zařízeními z odvětví SCADA. Cílem bakalářské práce je návrh pracoviště, které bude simulovat komunikaci dvou vybraných SCADA protokolů (DNP3, IEC 60870 nebo IEC 61850). V teoretické části bakalářské práce nastudujte zvolené protokoly. V praktické části proveďte návrh pracoviště a implementujte zvolené SCADA protokoly pro komunikaci mezi alespoň čtyřmi zařízeními, která budou reprezentovat různé SCADA prvky. Výstupem bakalářské práce bude navržené pracoviště, které bude schopno simulovat reálnou komunikaci alespoň dvou ze zmíněných protokolů.

DOPORUČENÁ LITERATURA:

[1] MAKHIJA, Jay; SUBRAMANYAN, L. R. Comparison of protocols used in remote monitoring: DNP 3.0, IEC 870-5-101 & Modbus. Electronics Systems Group, IIT Bombay, India, Tech. Rep, 2003.

[2] UZAIR, Muhammad. COMMUNICATION METHODS (PROTOCOLS, FORMAT & LANGUAGE) FOR THE SUBSTATION AUTOMATION & CONTROL (Project report of course 586 b) Dostupné z: http://www.eng.uwo.ca/people/tsidhu/Documents/project%20report%20Uzair.pdf

Termín zadání: 1.2.2019 ***Termín odevzdání:*** 27.5.2019

Vedoucí práce: Ing. Petr Blažek
Konzultant:

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

ABSTRAKT

Tato bakalářská práce je zaměřena na vytvoření plně funkčního generátoru komunikace protokolu IEC 60870-5-104. Teoretická část práce detailně vysvětluje základní principy, vlastnosti a možnosti komunikačních standardů DNP3, IEC 60870 a IEC 61850. Další část je zaměřena na rozbor komunikace a implementace této komunikace v elektroměru PQ monitor MEg44PAN. Poslední část se pak zabývá konfigurací zařízení Raspberry Pi 3 a emulací komunikace protokolu IEC 60870-5-104. Všechny programy byly napsány a testovány pomocí vývojového prostředí Clion.

ABSTRACT

This work is focused on creation of fully functional communication generator of IEC 60870-5-104 protocol. The theoretical part explains in detail the basic principles, properties and possibilities of communication standards DNP3, IEC 60870 and IEC 61850. The next part is focused on the analysis of communication and implementation of this communication in the PQ MEg44PAN device. The last part deals with the configuration of Raspberry Pi 3 devices and the communication emulation of the IEC 60870-5-104 protocol. All programs were written and tested using the Clion development environment.

KLÍČOVÁ SLOVA

DNP3, emulace, IEC-60870, IEC-60870-5-104, IEC-61850, klient–server, komunikace, MEg44PAN, RaspberryPi, SCADA, simulace

KEY WORDS

DNP3, emulation, IEC-60870, IEC-60870-5-104, IEC-61850, client-server, communication, MEg44PAN, RaspberryPi, SCADA, simulation

Citace práce:

BOHAČÍK, Antonín. *Simulace komunikace SCADA protokolů*. Brno, 2019. Dostupné také z: <`https://www.vutbr.cz/studenti/zav-prace/detail/118098`>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce Petr Blažek.

PROHLÁŠENÍ

Prohlašuji, že jsem bakalářskou práci na téma „Simulace komunikace SCADA protokolů“ vypracoval samostatně s použitím odborné literatury a pramenů, uvedených na seznamu, který tvoří přílohu této práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. Díl 4 Trestního zákoníku č. 40/2009 Sb.

Datum: ..... Podpis autora: .....

PODĚKOVÁNÍ

Děkuji vedoucímu bakalářské práce Ing. Petru Blažkovi za velmi užitečné konzultace, cenné rady, skvělý přístup ke studentům a trpělivost při zpracování této práce.

Datum: ..... Podpis autora: .....

Obsah

Seznam obrázků

Seznam tabulek

Seznam výpisů

Úvod

V průmyslovém odvětví se začal s příchodem moderní technologie objevovat trend dálkového řízení. K tomu účelu byl navržen software zvaný SCADA, který informace nejen shromažďuje, ale také podporuje vzdálenou regulaci a řízení.

V současné době vytvoření plně funkčního systému na bázi dálkového řízení není levná záležitost, a tak si mnozí výrobci chrání detailnější informace o svých systémech. Zde nastává problém vývoje nových technologií, neboť výrobci mnohdy nemají s kým spolupracovat na vývoji nových zařízení či nemají kde si tato nová zařízení otestovat. Proto se musí vytvářet generátory takové komunikace, aby si mohli ověřit funkčnost a propojitelnost těchto zařízení nebo odhalit nedostatky v protokolech.

Cílem této bakalářské práce je seznámit se nejen se standardem IEC 60870, ale i s dalšími z řad protokolů standardů DNP3 nebo IEC 61850, neboť vychází ze stejného konceptu. Dalším záměrem je porovnání jejich vlastností a ukázat principy, na kterých jsou postaveny.

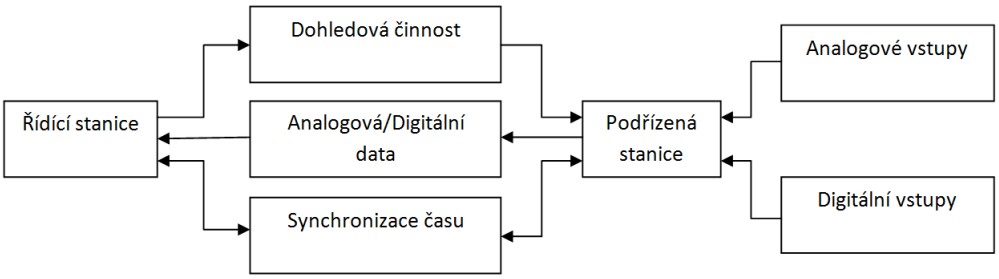
Tato práce se zabývá realizací pouze protokolu IEC 60870-5-104, neboť protokoly ze standardu DNP3 jsou převážně využívány v zahraničí (Amerika, Asie). Oproti standardu IEC 60870 tento standard nemá v Evropě velkého zastoupení a při přechodu ze starších protokolů se přechází převážně na standardy IEC 60870 a IEC 61850.

Druhá část této práce se zaměřuje na realizaci komunikace podle již zmíněného standardu IEC 60870, a to konkrétněji protokolu IEC 60870-5-104. Cílem je vytvoření generátorů komunikace mezi dvěma a více stanicemi. Tyto generátory budou emulovat komunikaci typickou pro zařízení komunikující protokolem IEC 60870-5-104, mezi které patří i elektroměr PQ monitor MEg44PAN od společnosti MEgA. U takto vytvořených generátorů komunikace bude možné měnit datovou část jednotlivých zpráv a celkový objem vygenerovaných dat. V neposlední řadě zde bude tato komunikace rozebrána a porovnána se skutečnými záznamy komunikace zařízením elektroměru PQ monitor MEg44PAN. Takto vytvořené generátory komunikace lze použít na testování zařízení, sítí či generování šumu v komunikačních kanálech.

1 SCADA

Supervisory Control And Data Acquisition neboli SCADA je souhrné označení pro sadu softwaru a zařízení, které slouží k dispečernímu řízení z centrálního pracoviště. Slouží k monitorování a řízení průmyslových a jiných technických zařízení a sběru dat. Tento systém tvoří řídící stanice (master) a podřízené stanice (slaves). Podřízená stanice vykonává povely od nadřazené stanice, jako povely k řízení, či nastavování výstupů, a poskytuje informace o vstupech. Podřízené stanice pouze sbírají data, ale samotný význam těchto dat zpracovává již řídící stanice. Základní schéma je zaznačeno na obr.1.1.

Standardizace těchto komunikačních protokolů přináší kompatibilitu zařízení od různých výrobců. Tato zařízení dále můžeme spojovat do větších systémových celků s jedním centrálním řízením. V praxi se využívá spoustu komunikačních protokolů, mezi které patří např.: Modbus, Modbus X, Profibus, Spabus, DNP3, IEC 60870 nebo IEC 61850. Poslední 3 zmíněné protokoly spadají do skupiny standardizovaných protokolů.[1, 2]



Obr. 1.1: Obecné schéma SCADA.

1.1 Protokol DNP3

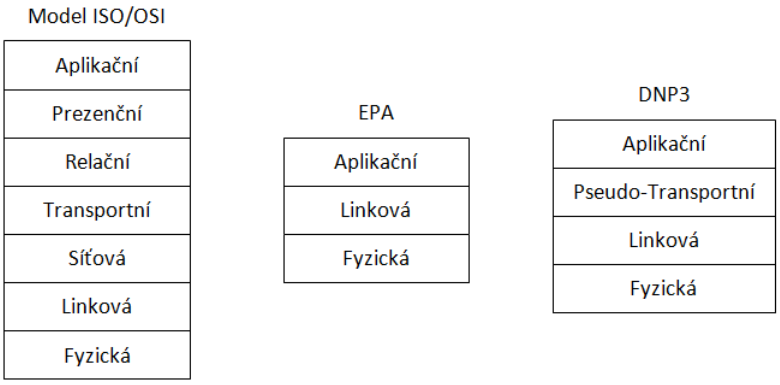
Protokol DNP3 (Distributed Network Protokol) spadá do standardu IEEE[1] Standard for Electric Power Systems Communications 1815-2012. Protokol zajišťuje komunikaci a výměnu dat mezi centrální stanicí a vzdálenými stanicemi. Nejedná se o univerzální komunikační protokol a slouží tedy především pro přenos sběrných

[1]Institute of Electrical and Electronics Engineers

dat a příkazů. Využívá sériových rozhraní RS-232, RS-422, RS-485, či ethernetového rozhraní. Využívá se v elektrických sítích, vodovodních a odpadních sítích, a to především v Severní Americe, Austrálii a v Číně.[1, 3]

1.1.1 Základní stavba

Protokol DNP3 byl založen na standardech IEC (International Electrotechnical Commission), proto přebral architekturu EPA (Enhanced Performance Architecture). DNP3 a IEC 60870-5 jsou proto velice podobné viz obr.1.2. Model EPA by se dal reprezentovat jako zjednodušený model ISO/OSI. Obsahuje fyzickou, linkovou, transportní a aplikační vrstvu.



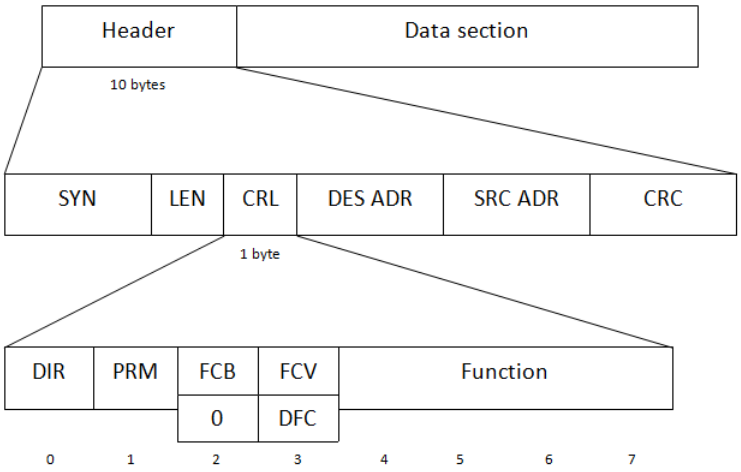
Obr. 1.2: Model ISO/OSI, EPA a DNP3.

Datový model protokolu DNP3 je rozdělen na skupiny definující typ dat ve zprávě např.: analogový či digitální, vstupní či výstupní atd. Skupiny jsou dále indexovány a každému indexu je přidělena kombinace (16 bitové číslo, 32 bitové číslo s příznakem atd.). Kombinace blíže specifikují typ dat skupiny daného indexu.[1, 3]

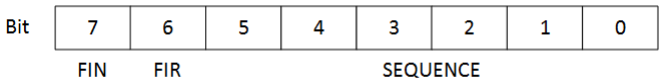
1.1.2 Hierarchie DNP3

- Fyzická vrstva je tvořena sériovou linkou RS-232, RS-422, RS-485 nebo ethernetem s podporou TCP/IP[2].
- Linková vrstva slouží k adresaci zdrojových a cílových zařízení a detekci chyb přijatých rámců. Tato struktura je zobrazena na obrázku 1.3.
- Transportní vrstva zajišťuje rozdělování a číslování zpráv. Hlavička transportní vrstvy obsahuje FIN (poslední segment), FIR (první segment) a SEQUENCE (pořadí segmentu) viz obr.1.4.

[2]Transmission Control Protocol/Internet Protocol



Obr. 1.3: Hlavička paketu na linkové vrstvě.



Obr. 1.4: Struktura segmentu na transportní vrstvě.

- Aplikační vrstva popisuje formát zpráv, služeb a procedur. Tato vrstva odpovídá za poskládání příchozích zpráv z transportní vrstvy a za vytvoření zpráv pro zpřístupnění uživateli. Fragmenty z řídící stanice jsou požadavky na provedení operací na podřízených stanicích a fragmenty z podřízené stanice jsou většinou odpovědi na tyto žádosti. Tento typ komunikace se nazývá Polling.[4]

1.1.3 Unsolicited response

Jedná se o typ komunikace, kdy podřízená stanice může přenést zprávu bez požadavku nadřízené stanici (nevyžádaná odpověď). Tato zpráva většinou informuje před změnou nějaké z hladin, která se nachází mimo standardní rozsah.

1.2 Standard IEC 60870

Protokol IEC 60870 byl vytvořen technickou komisí 57 v roce 1995. Jedná se o standard pro dálkové řízení, komunikaci elektronických napájecích systémů a ochranu. Využívá se především v energetice. Je taktéž postaven na architektuře EPA jako protokol DNP3.[5, 6]

1.2.1 Normy IEC 60870

IEC 60870 je u nás označena jako ČSN EN 60870. Jedná se o označení celé skupiny norem nazvanou Systémy a zařízení pro dálkové ovládání. Tento soubor je rozdělen do 5 částí. IEC 60870-1 (Všeobecné ustanovení) a IEC 60870-2 (Provozní podmínky) se zabývá všeobecnými zásadami této normy, IEC 60870-3 (Elektrické charakteristiky rozhraní) pojednává o elektrických charakteristikách rozhraní a IEC 60870-4 (Požadavky na vlastnosti) se zabývá požadavky na vlastnosti dálkového ovládání. Poslední částí je IEC 60870-5 (Komunikační protokoly), která se zabývá samotnými komunikačními protokoly.[5]

1.2.2 IEC 60870-5

Normy IEC 60870-5 specifikují funkce užitečné pro systémy dálkového ovládání, mezi které patří dvě nejdůležitější z nich Report By Exception a mechanismus přiřazování časových značek.[4]

Report by Exception

Tato funkce umožňuje vzdáleným podřízeným stanicím (slaves) požádat o komunikaci s řídící jednotkou (master). Podřízená jednotka je schopna inicializovat komunikaci s řídící jednotkou i bez dotázání, jinak by se o změně kritické proměnné dozvěděla až v okamžiku, kdy by na danou podřízenou jednotku došlo podle pravidelného dotazování.

Časové značky

Časové značky umožňují zjistit čas určité události, ke které je automaticky připojena. Poskytuje informace o tom, kdy nastala první událost a co bylo její příčinou ve formátu rok-týden-den a sekundy. Pro přesné fungování je nezbytné zachovávat přesnou časovou synchronizaci mezi řídící jednotkou a podřízenými jednotkami.

1.2.3 Jednotlivé části IEC 60870-5

- IEC 60870-5-1 Formáty přenosového rámce – asynchronní přenos dat s linkovými protokoly half-duplex a full-duplex, standardy pro kódování, formátování a synchronizace datových rámců.
- IEC 60870-5-2 Procedury spojového přenosu – procedury pro sériový přenos kódovaných digitálních dat.
- IEC 60870-5-3 Obecná struktura aplikačních dat – pravidla pro strukturování jednotlivých aplikačních dat v přenosových rámcích.

- IEC 60870-5-4 Definice a kódování aplikačních prvků – pravidla pro definování informačních prvků, digitálních či analogových procesních proměnných.
- IEC 60870-5-5 Základní aplikační funkce – standardy pro zajištění kompatibility různých zařízení elektrizační soustavy.
- IEC 60870-5-6 Pokyny pro testování shody podle doprovodných norem.
- IEC 60870-5-101 Společná norma pro dálkové ovládání – cílem je umožnit funkční kompatibilitu mezi zařízeními dálkového řízení.
- IEC 60870-5-102 Přenosové protokoly – doprovodná norma pro přenos integrovaných součtů v elektrických systémech.
- IEC 60870-5-103 Přenosové protokoly – společný standard pro informační rozhraní ochranných zařízení.
- IEC 60870-5-104 Síťový přístup – stanovuje používání těchto norem v běžných komunikačních sítích, jako například Ethernet s podporou TCP/IP.[4, 5]

1.2.4 Struktura IEC 60870-5

Protokol IEC 60870-5 je založen na redukovaném referenčním modelu EPA, který obsahuje tři vrstvy modelu ISO/OSI: aplikační vrstvu (L7), vrstvu propojení (L2) a fyzickou vrstvu (L1).[3]

Tab. 1.1: Normy IEC 60870-5 v modelu EPA

Vybrané aplikační funkce podle IEC 60870-5-5	Uživatelský proces
Vybrané informační aplikační prvky podle IEC 60870-5-4	Aplikační vrstva
Vybrané datové jednotky aplikačních služeb podle IEC 60870-5-3	
Vybrané postupy přenosu podle IEC 60870-5-2	Linková vrstva
Vybrané formáty přenosových rámců podle IEC 60870-5-1	
Vybrané doporučení ITU-T	Fyzická vrstva

- Fyzická vrstva – definuje hardwarově závislé specifikace komunikačních rozhraní IEC 60870-5-101 a IEC 60870-5-104. Zahrnuje definici komunikačních rozhraní (např.: V.24/V.28 FSK), konfigurace sítě (např.: point-to-point).
- Linková vrstva – určuje formáty rámců (FT1.2 s pevnou nebo proměnnou délkou), bitové pořadí informací (počínaje LSB a končícím MSB) a postupy přenosu (vyvážený nebo nevyvážený režim, primární nebo sekundární stanice, SEND/NO REPLY, SEND/CONFIRM atd.).
- Aplikační vrstva – definuje informační prvky pro strukturování aplikačních dat a funkce komunikačních služeb. Definuje celkovou strukturu zpráv, strukturu ASDU[3], informační prvky, adresování a směrování zpráv, atd.

[3]Application Servise Data Unit

1.2.5 Přenos dat

IEC 60870-5-101 poskytuje komunikační profil pro odesílání základních zpráv o dálkovém řízení mezi centrální řídící stanicí (master) a dálkově ovládanými stanicemi (slaves), které využívají stálé přímo propojené datové okruhy mezi centrální stanicí a jednotlivými výstupy. IEC 60870-5-104 kombinuje aplikační vrstvu IEC 60870-5-101 a přenosové funkce poskytované protokolem TCP/IP.[3] IEC 60870-5-101 umožňuje dva způsoby přenosu komunikace:

1. Nesymetrický přenos (unbalanced) – řídící stanice řídí datovou komunikaci postupným dotazováním řízených výstupů. Zahájí veškeré přenosy zpráv, zatímco řízené stanice pouze reagují na tyto zprávy. Podporovány jsou následující služby:
   - SEND/NO REPLY – pro globální zprávy a cyklické příkazy
   - SEND/CONFIRM – pro ovládací příkazy a příkazy pro nastavení
   - REQUEST/RESPOND – pro volání dat z kontrolovaných stanic
2. Vyvážený přenos (balanced) – každá stanice v tomto režimu může iniciovat přenos zpráv. Stanice mohou pracovat současně jako řídící stanice a řízené stanice (kombinované stanice). Vyvážený přenos je omezen na konfiguraci bodového přenosu (point-to-point) a vícebodového přenosu (multiple point-to-point). Podporované služby jsou:
   - SEND/CONFIRM
   - SEND/NO REPLY – toto může být iniciováno pouze řídící stanicí s vysílací adresou v konfiguraci více bodů do bodu

1.2.6 Komunikace

Důležitým pojmem v pochopení adresování podle IEC 60870-5 je rozdíl mezi řídícími a monitorovacími směry. Předpokládá se, že celkový systém má hierarchickou strukturu zahrnující centralizovanou kontrolu. Podle protokolu je každá stanice buď řídící stanicí (master), nebo kontrolovanou stanicí (slave).

V tomto způsobu komunikace je jedna jednotka řídící, která odesílá požadavky postupně všem svým podřízeným jednotkám. Každá podřízená jednotka reaguje individuálně na tyto požadavky, které jí jsou určeny. Klasické schéma žádost-odpověď (request-response) má pevná pravidla, zvaná Polling. Postup dotazování může být přizpůsoben individuálním požadavkům.[4, 6]

1.2.7 Datové objekty

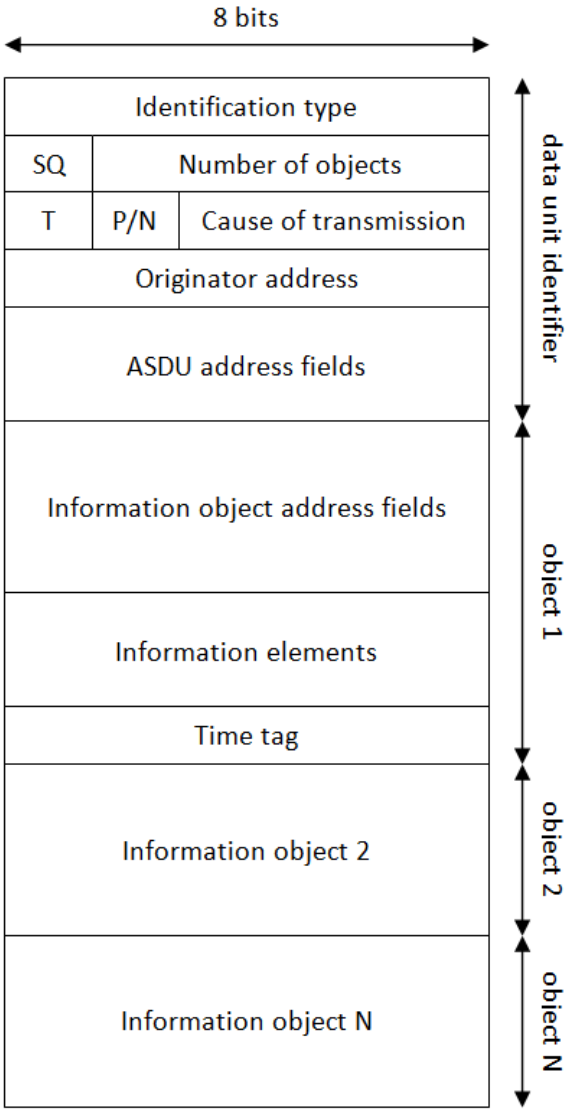
IEC 60870-5 obsahuje informace o sadě informačních objektů, které vyhovují jak obecným aplikacím SCADA, tak zejména aplikacím energetických systémů. Každý

typ dat má jedinečné typově identifikační číslo. V tabulce 1.2 jsou vypsány nejpoužívanější typově identifikační čísla tohoto protokolu. V jedné jednotce ASDU je zahrnut pouze jeden typ dat. Typ je v prvním poli ASDU jednotky. Typy informačních objektů jsou seskupeny podle směru (monitorování nebo řízení) a podle typu informací (informace o procesu, informace o systému, parametry přenosu souborů).

Tab. 1.2: Vybraná typově identifikační čísla IEC 60870-5

Typ ID	Název	Zkratka
1	Single point information	M_SP_NA_1
2	Single point information with time tag	M_SP_TA_1
3	Double point information	M_DP_NA_1
4	Double point information with time tag	M_DP_TA_1
9	Measured value, normalized value	M_ME_NA_1
10	Measured value, normalized value with time tag	M_ME_TA_1
11	Measured value, scaled value	M_ME_NB_1
12	Measured value, scaled value with time tag	M_ME_TB_1
13	Measured value, short floating point value	M_ME_NC_1
14	Measured value, short floating point value with time tag	M_ME_TC_1
30	Single point information with time tag CP56Time2a	M_SP_TB_1
31	Double point information with time tag CP56Time2a	M_DP_TB_1
35	Measured value, scaled value with time tag CP56Time2a	M_ME_TE_1
36	Measured value, short floating point value with time tag CP56Time2a	M_ME_TF_1
45	Single command	C_SC_NA_1
46	Double command	C_DC_NA_1
64	Bit string 32 bit with time tag CP56Time2a	C_BO_TA_1
100	General Interrogation command	C_IC_NA_1
101	Counter interrogation command	C_CI_NA_1
102	Read command	C_RD_NA_1
103	Clock synchronization command	C_CS_NA_1
110	Parameter of measured value, normalized value	P_ME_NA_1
111	Parameter of measured value, scaled value	P_ME_NB_1
112	Parameter of measured value, short floating point value	P_ME_NC_1
120	File ready	F_FR_NA_1
121	Section ready	F_SR_NA_1
122	Call directory, select file, call file, call section	F_SC_NA_1
123	Last section, last segment	F_LS_NA_1
124	Ack file, Ack section	F_AF_NA_1
125	Segment	F_SG_NA_1

Aplikační data jsou přenášena uvnitř ASDU v rámci jednoho nebo více informačních objektů viz obr.1.5. V závislosti na příznaku proměnné struktury zde může být více informačních objektů, z nichž každý obsahuje definovanou sadu jednoho nebo více informačních prvků, nebo může být pouze jeden informační objekt obsahující množství identických informačních prvků. V každém případě je informační prvek základním prvkem, který slouží k přenosu informací podle protokolu.[4]



Obr. 1.5: Struktura ASDU.

Formát CP56Time2a

Jedná se strukturovaný formát času, který je využíván k vytváření časového razítka. Převážně je využíván sedmi bajtový formát, neboť tří bajtový formát není v protokolu IEC 60870-5-104 povolen.

1.2.8 Adresace

IEC 60870-5-101 definuje adresování jak na linkové, tak na aplikační vrstvě. Pro identifikaci koncové stanice je uvedena adresa odkazu (nebo adresa zařízení) a adresa ASDU (nebo společná adresa). Adresa zařízení je identifikační číslo zařízení.

Linková adresa může být 1 nebo 2 bajty pro nevyváženou komunikaci (unbalanced) a 0, 1 nebo 2 oktety pro vyváženou komunikaci (balanced). Vzhledem k tomu, že vyvážená komunikace probíhá bodově (point-to-point), linková adresa je nadbytečná, ale může být zahrnuta pro zabezpečení. Rozsah hodnot závisí na délce linkové adresy, která může být jeden bajt, tj. rozsah 1 až 255, nebo dva bajty, tj. rozsah 1 až 65 535. Typická hodnota je 1 pro IEC 60870-5-101 a 2 pro IEC 60870-5-104. Linková adresa FF nebo FFFF je definována jako vysílací adresa a může být použita pro adresování všech stanic na linkové vrstvě.

Každé zařízení v komunikační síti má společnou adresu ASDU. Společná adresa ASDU kombinovaná s informační adresou objektu a samotnými daty vytváří jedinečnou adresu pro každý datový prvek. ADSU je obvykle aplikační adresa klienta, která musí odpovídat adrese definované v konfiguraci klienta. Toto je definováno jako adresa řídící stanice ve směru řízení. Ve směru sledování však pole společné adresy obsahuje adresu stanice, která vrací data (řízená stanice). To je nutné, aby byla data jednoznačně identifikována a mapována. Maximální hodnota závisí na délce adresy ASDU, která je jeden nebo dva bajty podobně jako adresa zařízení. Typické hodnoty jsou 1 pro IEC 60870-5-101 a 2 pro IEC 60870-5-104.[4]

1.3 Standard IEC 61850

Soubor norem IEC 61850, u nás označen jako ČSN EN 61850, byl zaveden pro sjednocení komunikačních standardů v energetice, neboť zde existuje mnoho typů protokolů, které jsou vzájemně nekompatibilní, a způsob, jak tyto systémy propojit, je někdy velice obtížný. Při vývoji tohoto protokolu byl kladen důraz především na spolehlivost a dlouholetou stabilitu. Je tvořen z několika dokumentů, které se zabývají požadavky na řízení, terminologií, komunikací či samotnými zkouškami a shodami zařízení používaných v automatických rozvodnách. Jedná se tedy o standardizovaný datový model, který využívá všechny vrstvy referenčního modelu ISO/OSI.

IEC 61580 je jediný standard, který vyhovuje všem požadavkům energetických a rozvodných společností na celém světě na kompatibilitu instalovaného souboru regulačních a řídících zařízení od různých výrobců. Každý uzel sítě podle IEC 61580 připojený jako klient může řídit provoz na síti a komunikovat se všemi servery i podřadnými zařízeními.

Podobně jako u DNP3 a IEC 60870 komunikuje pomocí modelu žádost–odpověď a nevyžádané zprávy. Ovšem protokol IEC 61850 definuje i další typy zpráv, neboť na rozdíl od klasické architektury klient–server umožňuje i klientským stanicím, aby řídily přenos dat.[2, 7]

Tab. 1.3: Typy zpráv v IEC 61850

Typ	Název	Příklad
1A	Rychlé zprávy - vypínací signál	Vypínací signál
1B	Rychlé zprávy - ostatní	Povely, jednoduché hodnoty
2	Středně rychlé zprávy	Měření hodnoty
3	Pomalé zprávy	Parametry
4	Zprávy s prvotními daty	Vstupní data z převodníků
5	Funkce přenosu souborů	Velké soubory
6A	Zprávy pro časovou synchronizaci A	Sběrnice dat
6B	Zprávy pro časovou synchronizaci B	Provozní sběrnice
7	Zprávy s povely pro řízení přístupu	Povely z HMI stanice

1.3.1 Obsah normy IEC 61850

1. IEC 61850-1 Úvod a přehled
2. IEC 61850-2 Výklad zvláštních výrazů
3. IEC 61850-3 Všeobecné požadavky
4. IEC 61850-4 Systémové a projektové řízení
5. IEC 61850-5 Požadavky na komunikaci pro funkce a modely zařízení
6. IEC 61850-6 Konfigurační popisový jazyk pro komunikaci v elektrických stanicích týkající se inteligentních elektronických zařízení
7. IEC 61850-7 Základní komunikační struktura pro rozvodná zařízení a napájecí zařízení
   - Část 1: Zásady a modely
   - Část 2: Abstraktní rozhraní pro komunikační služby (ACSI[4])
   - Část 3: Obecné třídy dat
   - Část 4: Třídy kompatibilních logických uzlů a třídy dat
8. IEC 61850-8 Mapování specifických komunikačních služeb (SCSM[5])
   - Část 1: Mapování na MMS[6] a na ISO/IEC 8802-3
9. IEC 61850-9 Mapování specifických komunikačních služeb (SCSM)

[4]Abstract Communications Service Interface
[5]System Center Service Manager
[6]Multimedia Messaging Service

    - Část 1: Přenos vzorkovaných hodnot po sériovém jednosměrném vícebodovém spoji bod-bod
    - Část 2: Vzorkované hodnoty
10. IEC 61850-10 Zkoušky shody

1.3.2 Datový model

Data jsou v tomto modelu uložena v ochranných terminálech s objektově orientovaným přístupem pro zachování lepší ochrany a stálosti systému. Pro vzájemnou kompatibilitu zařízení musí mít objekty v modelu předem definovanou syntaxi.[2, 7]

Dělení v datovém modelu

- Fyzické zařízení představuje ochranný terminál, který je definovaný IP adresou. Vnější zařízení může komunikovat s tímto zařízením pouze přes server. Fyzické zařízení je identifikováno unikátním názvem.
- Logické zařízení je podskupinou fyzických zařízení. Jedno fyzické zařízení může obsahovat několik logických zařízení. Logické zařízení definují logické uzly.
- Logický uzel jsou skupinou dat a služeb, které logicky souvisí se specifikovanou funkcí v dané soustavě. Jedná se o vizualizaci konkrétních prvků.
- Datový objekt je podmnožinou logického uzlu. Datový objekt je základní stavební kámen objektově orientovaného modelu IEC 61850.
- Datový atribut je nejmenší částí datového modelu a může reprezentovat logické stavy, např.: vypnuto či zapnuto.

2 Simulátor komunikace IEC 60870-5-104

V této části se budeme zabývat technickými požadavky pro vytvoření generátoru komunikace protokolu IEC 60870-5-104. Základním konceptem je komunikace stylem výzva–odpověď, která je pro tento protokol typická. Pro výzvu používáme ASDU zprávu spadající do skupiny příkazů s typově identifikačním číslem v rozsahu od 45 do 64, pro příkaz s časovým razítkem, a dále pak od 100 do 107 viz tab 1.2.

Odpověď na takovýto příkaz spadá do skupiny zpracované informace (1 až 21), zpracované informace s časovou značkou (30 až 40), parametr (110 až 113) nebo přenos souborů (120 až 127). Tyto odpovědi jsou závislé na typu dotazu a také na způsobu, který si výrobce daného systému zvolí. Tedy na zprávu „General Interrogation command“ můžeme odpovídat jak zprávou „Single point information“, tak i „Single point information with time tag CP56Time2a“. Záleží pouze na formátu, který je pro danou problematiku výhodnější.

Samotná zpráva obsahuje buď APCI[1] nebo APCI s ASDU zprávou, která již nese konkrétní informační objekty. Tyto informační objekty se vytváří dle potřeby dané informace. Obecně platí, že délka APCI zprávy je 6 bajtů. U většiny zpráv můžeme posílat více než jeden informační objekt, a tak zprávy můžeme rozdělit na ty s pevnou délkou, jako je například ASDU pro navázání komunikace, a na zprávy s proměnlivou velikostí obsahující většinou data aplikačních služeb.

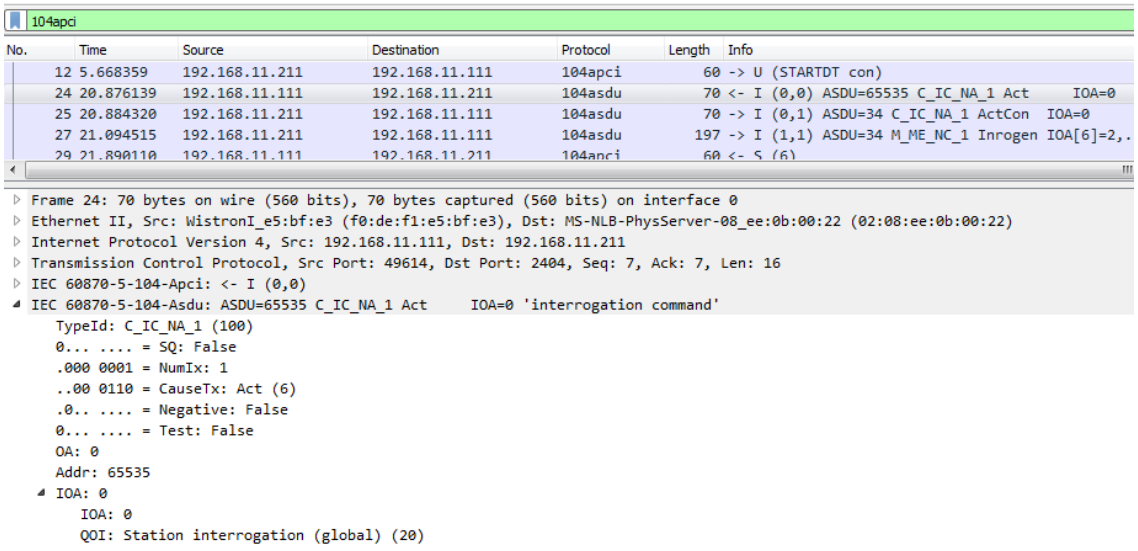
2.1 Způsob komunikace elektroměru PQ monitor MEg44PAN protokolem IEC 60870-5-104

Nyní si rozebereme komunikaci elektroměru PQ monitor MEg44PAN. Jak již bylo nastíněno, tento elektroměr funguje na bázi výzva–odpověď. Pro výčet většího počtu dat se nejprve vyšle „Interrogation command“ pro serverovou stanici, v našem případě elektroměr PQ monitor MEg44PAN, ve kterém se jej dotážeme na konkrétní skupinu. Ukázka ASDU dotazu pro elektroměr je na obrázku 2.1.

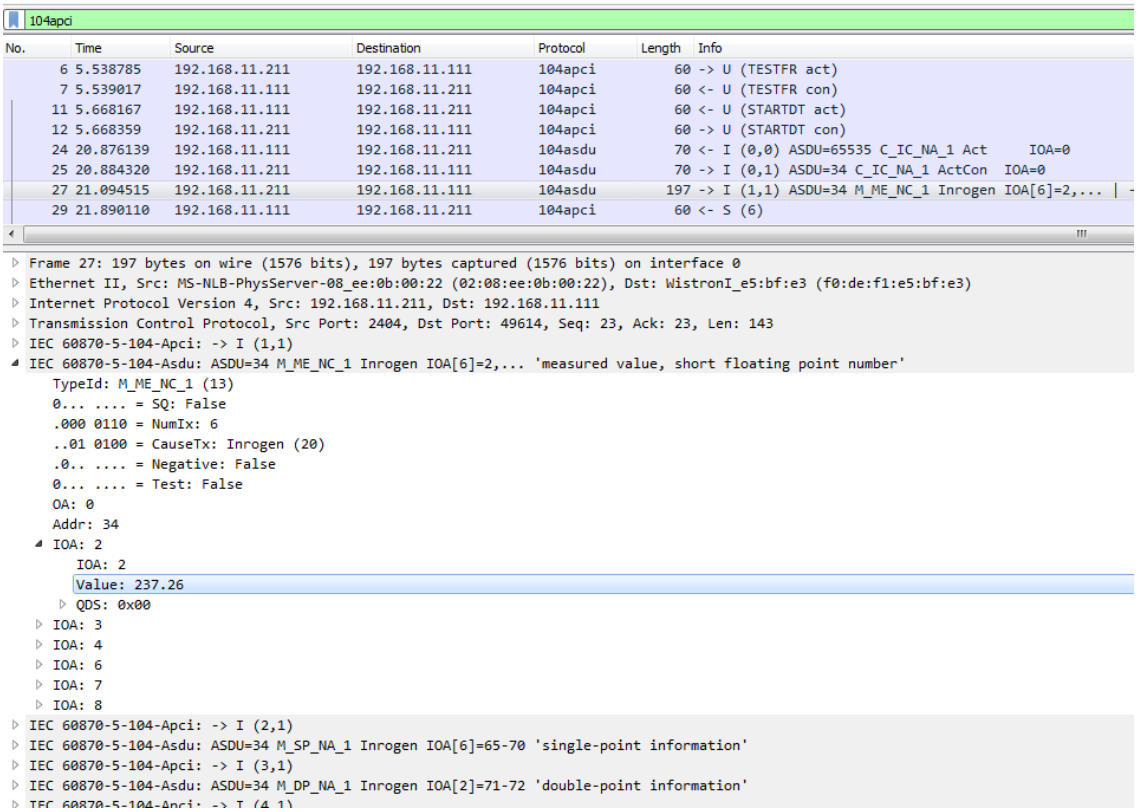
Následně se na takovouto výzvu odpovídá potřebnými daty spadajícími do dotazované skupiny viz obrázek 2.2. V tomto případě zasílá zprávu s informačními objekty typu „Measured value, short floating point value“, „Single point information“ a „Double point information“. Také si můžeme povšimnout u informačního objektu s IOA=2 již konkrétní hodnoty napětí o velikosti 237,26 V.

Dalším způsobem přenášení dat elektroměru PQ monitor MEg44PAN je přenos souboru. Tento soubor je přenášen periodicky nebo na vyžádání a je možné díky němu přenášet hodnoty proměnných v čase, registry kvality, registry událostí a rychlé

[1] Application Protocol Control Information

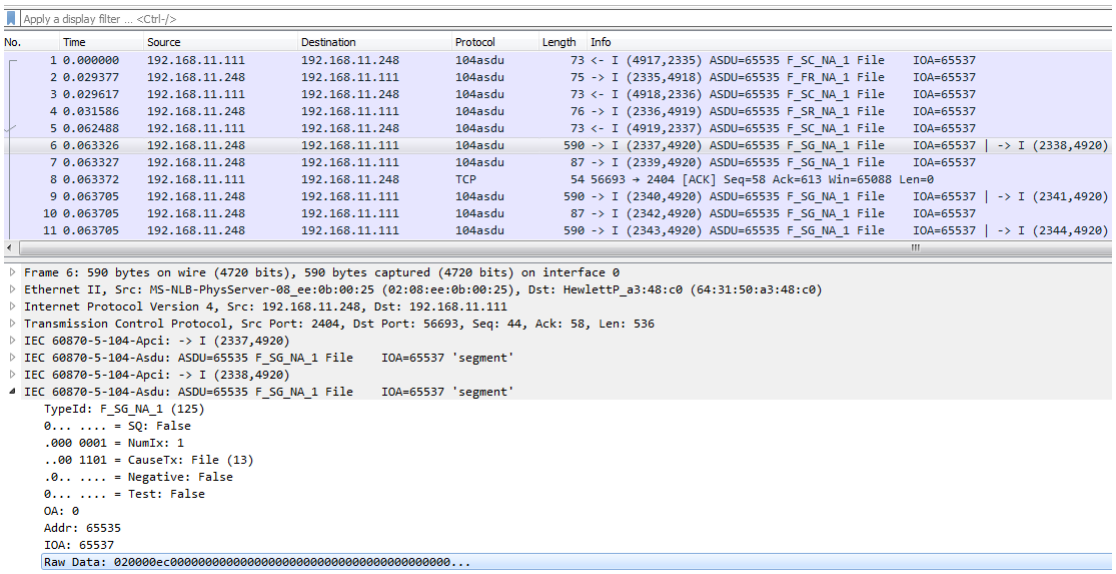


Obr. 2.1: Dotaz pro elektroměr PQ monitor MEg44PAN.



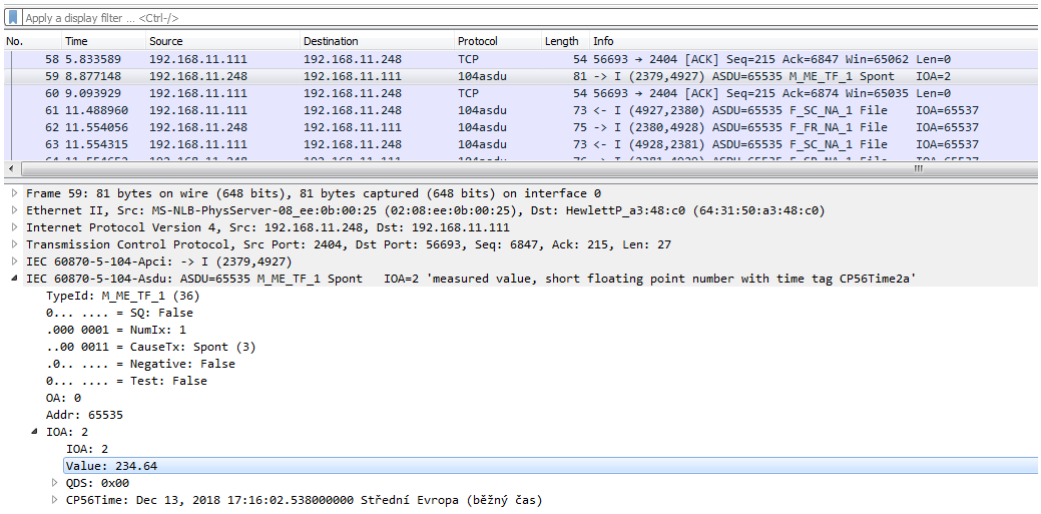
Obr. 2.2: Odpověď na dotaz elektroměrem PQ monitor MEg44PAN.

(skokové) změny měřených hodnot. Data přenášená pomocí souboru závisí čistě na implementaci výrobce, proto se tímto způsobem komunikace v rámci této práce již nebudeme zabývat. Ukázka jednoho segmentu zprávy je na obrázku 2.3.



Obr. 2.3: Přenos souboru elektroměrem PQ monitor MEg44PAN.

Dále je zde taktéž uplatněna funkce „report by exception“, díky které je možné vyslat zprávu nadřazené stanici v případě, kdy došlo k neočekávané změně nějaké z proměnných. U elektroměru PQ monitor MEg44PAN je tato funkce implementována pro hodnotu efektivního napětí. Ukázka spontánní zprávy je na obrázku 2.4.



Obr. 2.4: Spontánní zpráva elektroměru PQ monitor MEg44PAN.

2.2 Vytvořené programy

Komunikace je realizována mezi programem *cs104_client* běžícím na klientském zařízení a programem *cs104_server*, který běží na jednom či více serverových zařízeních. Tyto programy byly vytvořeny pro emulování komunikace v různých režimech elektroměru PQ monitor MEg44PAN na zařízení Raspberry Pi. Prvním je režim NN[2] rozvaděče, pro práci s napětím do $1000\,V$, a druhým VN[3] rozvaděče, pro práci s napětím od $1000\,V$ do $52\,kV$. Klientská stanice je nastavena tak, že dokáže obsluhovat až 6 serverů a vytváří TCP spojení s přednastaveným cílovým portem 2404.

Dále tyto programy generují textové logy pro účel testování v reálném provozu, a to především zda-li při přenosu nenastala chyba či změna hodnoty vlivem komunikačního šumu nebo modifikace konkrétní hodnoty ze strany útočníka.

Tyto programy byly napsány v programovacím jazyce C ve vývojovém prostředí CLion verze 2018.3.4 a byla zde využita veřejná knihovna lib60870[4], která definuje základní funkce protokolu IEC 60870-5-104. Samotné testování probíhalo ve vývojovém prostředí CLion a také v laboratoři VUT na reálné síti za účelem testování komunikace v reálném provozu.

2.2.1 Program cs104_client

Tento program plní funkci nadřazené stanice (klient) a řídí běh celé komunikace. Je vytvořen tak, že při spuštění se nás dotáže na seznam přednastavených IP adres. V případě, že s tímto seznamem nesouhlasíme, můžeme tento seznam pozměnit a nastavit do něj námi potřebné IP adresy podřízených stanic (serverů). Dále se nás program dotáže v jakém režimu jej chceme spustit, jestli v NN nebo VN. Pro obě možnosti jsou definovány jiné informační objekty, které se mají po spuštění vytvářet, neboť v reálném provozu se v daných režimech pracuje s odlišnými informačními objekty. Poté si můžeme zvolit jaký počet a jaké typy zpráv chceme generovat. Poslední volbou je počet opakování daného cyklu. Tato volba slouží k simulování více NN či VN rozvaděčů v jednom bodě. Tento případ, kdy jedním bodem prochází více než jeden datový tok, je v praxi běžnou záležitostí. Na obrázku 2.5 je zobrazeno menu při spuštění programu *cs104_client*.

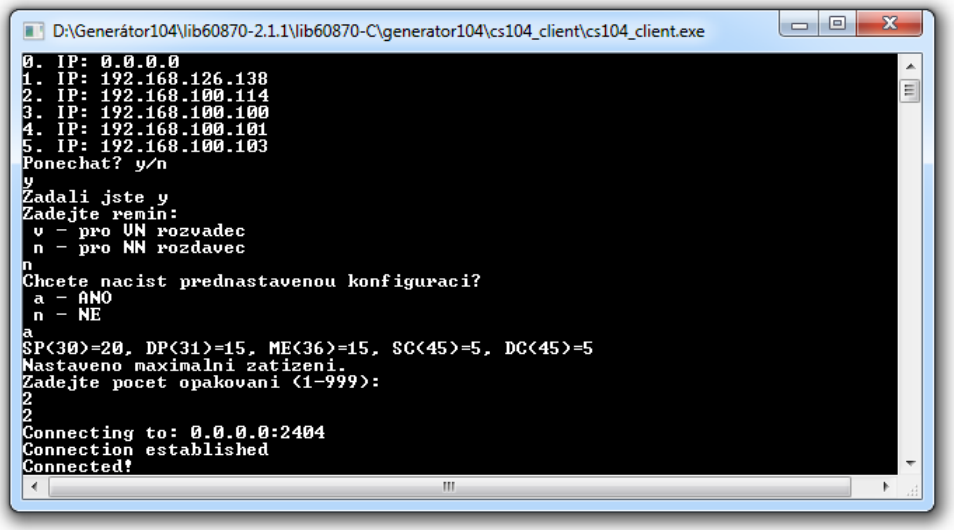
Následně co tento program spustíme, začne periodicky navazovat spojení se servery na IP adresách námi vytvořeného seznamu IP adres. V případě, že na takové adrese existuje běžící server, se na něj připojí a začne s ním komunikovat. Nejprve vyšle příkaz na synchronizaci času, počká 3 vteřiny a poté odešle dotazy a příkazy.

[2]Nízké napětí

[3]Vysoké napětí

[4]Dostupná na adrese: `https://libiec61850.com/libiec61850/downloads/`

Po zhruba dalších 7 vteřinách, kdy čekal na odpovědi od serveru, se od daného serveru odpojí a pokusí se připojit na další server v seznamu. Tento proces běží periodicky do ukončení programu.



Obr. 2.5: Ukázka menu programu cs104_client.

Vývoj tohoto programu byl zaměřen na komunikaci jednoho klienta až se 6 servery, ovšem architektura programu dovoluje toto číslo zvýšit. Při vývoji byl tento program schopný pracovat se seznamem čítající 16 IP adres. Ukázka kódu pro navázání spojení je zobrazena ve výpisu 2.1.

Výpis 2.1: Výpis hlavní funkce programu cs104_client.

```
1  public static void Main (string[] args){
2
3                  /*NAVAZANI SPOJENI*/
4    printf("Connecting to: %s:%i\n",ip,port);
5    CS104_Connection con = CS104\_Connection_create(ip, port);
6
7                  /*NASTAVENI HLAVICKY*/
8    CS104_Connection_setConnectionHandler (con,
9    connectionHandler, NULL);
10   CS104_Connection_setASDUReceivedHandler (con,
11   asduReceivedHandler, NULL);
12
13                 /*VYHODNOCENI STAVU SPOJENI*/
14   if (CS104_Connection_connect(con)) {
15       printf("Connected!\n");
16       ...
```

```
17          samotné tělo programu
18          ...
19          }
20      else
21          printf("Connect failed!\n");
22
23                      /*UKONCENI SPOJENI*/
24      CS104_Connection_destroy(con);
25      }
```

Součástí tohoto programu jsou také funkce, které nastavují CoT[5] nebo-li příčinu přenosu. Další součástí tohoto programu je funkce na synchronizaci času mezi klientem a serverem. Jedná se o klíčovou funkci pro fungování logování, neboť při rozdílných časech na jednotlivých zařízeních je téměř nemožné zpětně zjistit k čemu a v jakém čase došlo.

Dále se v rámci tohoto programu vyskytují i funkce pro odesílání různých dotazů či příkazů z tabulky 1.2, jako například „General Interrogation command“ či již zmíněný příkaz na synchronizaci času „Clock synchronization command“. Také jsou zde přítomny funkce na rozpoznávání typu příchozích zpráv, jako například „Single point information“, „Single point information with time tag CP56Time2a“ či „Measured value, short floating point value with time tag CP56Time2a“, která představuje měřenou hodnotu na desetinná místa s časovým razítkem.

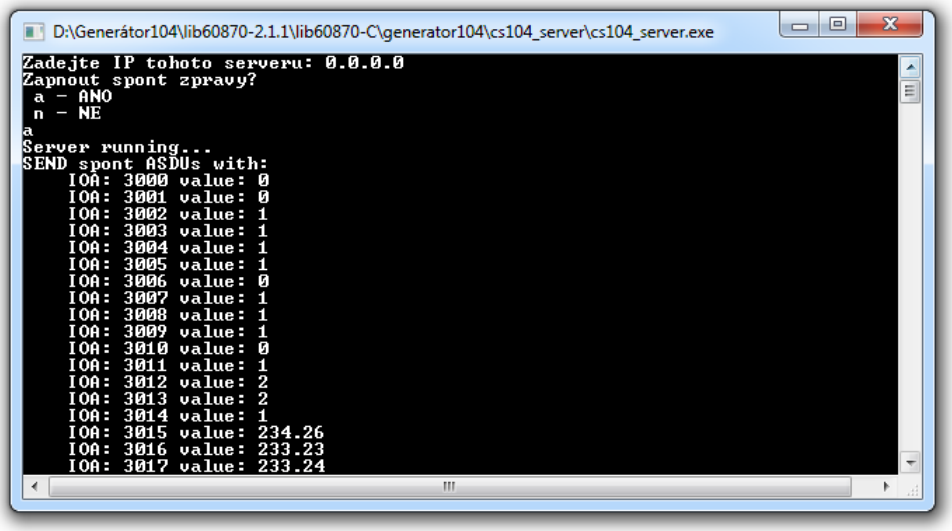
2.2.2 Program cs104_server

Tento program plní funkci podřízené stanice (server), která přijímá příkazy či dotazy a odesílá na ně odpovědi. Také zastává funkci sběrače dat, v našem případě generátoru dat, který tato data následně přeposílá na klientskou stanici k dalšímu zpracování.

Po samotném spuštění se nás program dotáže na IP adresu daného serveru, kterou potřebuje pro další nastavování. Dále se nás zeptá, zda-li chceme zapnout funkci na generování spontánních zpráv. V případě, že souhlasíme, bude v náhodném intervalu 5 až 30 sekund generovat spontánní zprávy pro klientskou stanici. Tato funkce napodobuje chování serveru, který oznamuje neočekávaný stav měřené proměnné klientské stanici. V opačném případě by se klientská stanice o takovémto stavu dozvěděla až v následujícím periodickém cyklu. Na obrázku 2.6 je zobrazeno menu při spuštění programu *cs104_server*.

Při spuštění si program nejdříve nastaví parametry IP adresy a portu. Následně se dotáže na další parametry aplikační vrstvy pro správné tvoření ASDU zpráv. Dále

[5]Cause of Transmission



Obr. 2.6: Ukázka menu programu cs104_server.

pak přednastaví hlavičky pro zpětné volání na dotaz od klientské stanice a nakonec se pokusí vytvořit server. Pokud vše při vytváření proběhlo korektně, začne se vykonávat nekonečná smyčka programu. V případě, že při vytváření serveru došlo k chybě, vypíše se chybová hláška „Starting server failed!“. Ukázka kódu vytváření serveru je zobrazena ve výpisu 2.2.

Výpis 2.2: Výpis hlavní funkce programu cs104_server.

```
1  int main(int argc, char** argv) {
2
3                  /*NASTAVENI PARAMETRU*/
4    CS104_Slave slave = CS104_Slave_create(100, 100);
5    CS104_Slave_setLocalAddress(slave, "192.168.100.111");
6    CS101_AppLayerParameters alParams =
7      CS104_Slave_getAppLayerParameters(slave);
8
9                  /*PREDNASTEVENI HLAVICEK*/
10   CS104_Slave_setClockSyncHandler(slave,
11     clockSyncHandler, NULL);
12   CS104_Slave_setInterrogationHandler(slave,
13     interrogationHandler, NULL);
14   ...
15   další nastavování hlaviček
16   ...
17
18                 /*VYTVORENI SERVERU*/
```

```
19  CS104_Slave_start(slave);
20
21                  /*VYHODNOCENI STAVU SERVERU*/
22  if (CS104_Slave_isRunning(slave) == false) {
23    printf("Starting server failed!\n");}
24  else{
25    printf("Server running...\n");}
26
27  while (running) {
28    ...
29    samotné tělo programu
30    ...
31    }
32                  /*UKONCENI SERVERU*/
33  CS104_Slave_stop(slave);
34  CS104_Slave_destroy(slave);
35  }
```

Tento program je vytvořen tak, že poté co jej spustíme začne generovat odpovědi na klientské dotazy a příkazy. Generuje například hodnoty měřených fyzikálních veličin, jednotlivé stavy vypínačů či polohy, ve kterých se nachází dané přepínače. Tato data jsou podle jednotlivých typů zpracována do zpráv a ukládána do odesílací fronty. Poté co se na něj připojí klientská stanice, jsou tyto zprávy odeslány.

Součástí tohoto programu je taktéž funkce ke zpracování příkazu na synchronizaci času mezi klientem a serverem. Další součástí tohoto programu jsou funkce pro odesílání konkrétních typů zpráv jako je třeba „Single point information with time tag CP56Time2a“, „Double point information with time tag CP56Time2a“, „Measured value, scaled value with time tag CP56Time2a“ nebo v našem případě nejpoužívanější „Measured value, short floating point value with time tag CP56Time2a“ viz tabulka 1.2.

2.3 Komunikační scénáře

V rámci této bakalářské práce byl vytvořen generátor komunikace protokolu IEC 60870-5-104, který je schopen simulovat na straně klienta komunikaci NN či VN rozvaděče představující elektroměr PQ monitor MEg44PAN se serverovými stanicemi, které generují již konkrétní data. Mezi základní komunikační scénáře patří scénář periodických dotazů či příkazů, scénář periodického výčtu dat a scénář spontánního zasílání zpráv. Tyto scénáře tvoří dílčí součásti finálního generátoru. Mezi jeho další funkci patří možnost volby obsahu datové části a celkového objemu vygenerovaných

dat. Tento generátor je vytvořen pro testování zařízení komunikující protokolem IEC 60870-5-104, testování sítí či vytváření šumu v komunikačních kanálech.

2.3.1 Scénář periodického dotazování a přikazování

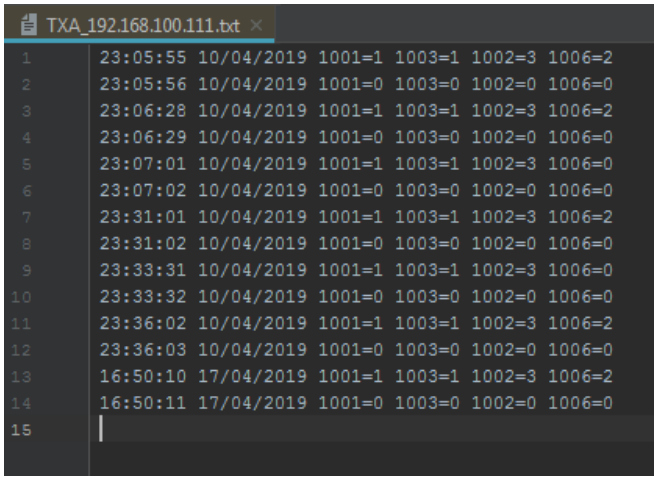
První testovací scénář představuje komunikaci mezi klientem a serverem za účelem zjištění hodnot konkrétních proměnných a změny určitých proměnných na serveru z klientské stanice. Celý proces je periodický, takže k němu dochází opakovaně v přednastaveném čase jedné minuty. Tento časový interval lze však lehce měnit.

Klient vygeneruje dotaz na daný informační objekt, který je spojený již s konkrétní proměnnou. Tyto proměnné představují stavy vypínačů či přepínačů. Dále pak vygeneruje příkaz na nastavení konkrétní proměnné a tento příkaz odešle. Ještě před odesláním se tyto hodnoty zaznamenají do logovacího souboru TXA_*(IP adresa serveru)*.txt. Server následně odešle odpověď klientovi k jaké změně a v jakém čase došlo. Klient tuto odpověď očekává a po obdržení ji vypíše a zaznamená do souboru RXA_*(IP adresa serveru)*.txt. Ukázka kódu řešení nastavovacího příkazu je zobrazena ve výpisu 2.3.

Výpis 2.3: Výpis dotazu na konkrétní hodnoty.

```
1                   /*NASTAVENI CASU PRO DOTAZ*/
2  int cas_dotazu=60000; //1min = 60,000ms
3  int cas_aktualni=0;
4    if(cas_aktualni>=cas_dotazu){
5
6                   /*VYTVORENI INFORMACNIHO OBJEKTU*/
7      InformationObject sc = (InformationObject)
8        SingleCommand_create(NULL, 1001, true, false, 0);
9      printf("SEND single command to switch IOA: 1001 to 1.\n");
10
11                  /*ODESLANI ZPRAVY*/
12     CS104_Connection_sendProcessCommandEx(con,
13       CS101_COT_ACTIVATION, 1, sc);
14
15                  /*ZAPIS DO LOGOVACIHO SOUBORU*/
16     zapisTX(1001,SingleCommand_getState((SingleCommand)sc));
17
18                  /*ZNICENI INFORMACNIHO OBJEKTU*/
19     InformationObject_destroy(sc);
20
21     cas_aktualni=0;
22     }
```

Pokud nastavení hodnoty proběhlo korektně, měly by být záznamy v souborech TXA a RXA totožné. V opačném případě došlo cestou k chybě či útoku na komunikaci. Tento soubor je složen z časového razítka, kdy byla daná zpráva odeslána ve formátu *hodina*:*minuta*:*vteřina den*/*měsíc*/*rok*. Dále jsou zde adresy informačních objektů a jejich velikosti. Objekty s adresou 1001 a 1003 představují jednobodovou informaci (0/1) a objekty s adresou 1002 a 1006 představují vícebodovou informaci, která nabývá více než dvou stavů (0/1/2/3). Ukázka struktury logovacího souboru TXA_*(IP adresa serveru)*.txt a RXA_*(IP adresa serveru)*.txt je zachycena na obrázku 2.7.



Obr. 2.7: Logovací soubor TXA.txt.

2.3.2 Scénář periodického výčtu dat

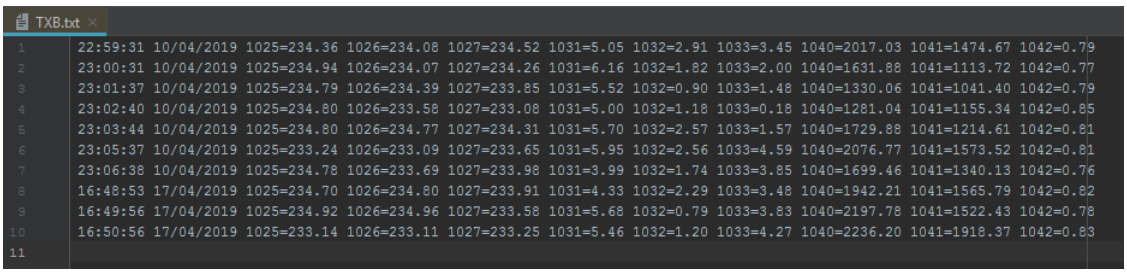
Druhý testovací scénář představuje komunikaci mezi serverem a klientem, kdy server periodicky generuje velký objem jednorázových dat. V našem případě server generuje hodnoty proměnných U_{f1}, U_{f2}, U_{f3}, I_1, I_2, I_3, P_1, P_2, P_3, P, Q a kosinus φ každou minutu. Ovšem z výpisu 2.4 na řádku 2 je patrné, že lze tento časový úsek měnit dle potřeby. Tyto proměnné jsou typické pro měření elektroměrem PQ monitor MEg44PAN, který tyto hodnoty odesílá v průměru každých 15 minut. Poté co jsou tato data odesílána na klientskou stanici, jsou také uložena do logovacího souboru. Ukázka kódu řešení je zobrazena ve výpisu 2.4.

Výpis 2.4: Výpis periodického zasílání zpráv.

```
1                    /*NASTAVENI CASU PRO ODESLANI*/
2 int cas_vypisu=60000; // 1 minuta = 60,000ms
3 int cas_periodicky=0;
4   if (cas_periodicky >= cas_vypisu) {
```

```
5
6                   /*VYTVORENI ASDU*/
7       CS101_ASDU newAsdu = CS101_ASDU_create(alParams,
8         false, CS101_COT_PERIODIC, 0, 1, false, false);
9
10                  /*VYTVORENI INFORMACNIHO OBJEKTU*/
11      InformationObject io = (InformationObject)
12        MeasuredValueShortWithCP56Time2a_create(NULL, 1025, uf1,
13        IEC60870_QUALITY_GOOD, CP56Time2a_createFromMsTimestamp
14        (NULL, Hal_getTimeInMs()));
15
16                  /*ZAPIS DO LOGOVACIHO SOUBORU*/
17      zapisTXB(InformationObject_getObjectAddress(
18        (InformationObject) io);
19
20                  /*ODESLANI ZPRAVY*/
21      CS104_Slave_enqueueASDU(slave, newAsdu);
22
23                  /*ZNICENI INFORMACNIHO OBJEKTU A ASDU*/
24      InformationObject_destroy(io);
25      CS101_ASDU_destroy(newAsdu);
26
27      cas_periodicky=0;
28      }
```

Vytvořený logovací soubor odeslaných zpráv slouží pro případné zjištění chybovosti během přenosu a k zaznamenání stavů jednotlivých proměnných. Tato data se zapisují do souboru TXB.txt a tento záznam obsahuje časové razítko, kdy byla daná zpráva odeslána, stejně jako u předchozího scénáře. Dále obsahuje adresy informačních objektů a jejich velikosti. Jedná se o fyzikální veličiny typické pro elektroměr PQ monitor MEg44PAN a význam těchto proměnných je zaznamenán v tabulce 2.1. Ukázka struktury souboru TXB.txt je zachycena na obrázku 2.8.



Obr. 2.8: Logovací soubor TXB.txt.

Tab. 2.1: Jednotlivé informační objekty a jejich význam pro scénář výčtu dat.

IOA	Název	Veličina	Rozsah
1025	Napětí první fáze	U_1	233 - 235 V
1026	Napětí druhé fáze	U_2	233 - 235 V
1027	Napětí třetí fáze	U_2	233 - 235 V
1031	Proud první fáze	I_1	0 - 10 A
1032	Proud druhé fáze	I_2	0 - 10 A
1033	Proud třetí fáze	I_3	0 - 10 A
1040	Celkový výkon	P	0 - 7050 W
1041	Celkový zdánlivý výkon	Q	0 - 7050 VA
1042	Efektivita	$cos\phi$	0 - 1

2.3.3 Scénář spontánního zasílání zpráv

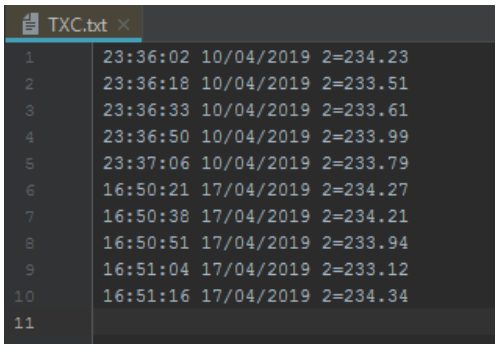
Poslední testovací scénář představuje komunikaci mezi serverem a klientem z dlouhodobého hlediska, za účelem minimálního objemu dat. V našem případě server vygeneruje každých 5 až 20 sekund zprávu nesoucí velikost efektivního napětí U_f, kterou zasílá na klientskou stanici. Hodnota napětí se pohybuje od 233 V do 235 V. Toto rozmezí vychází ze záznamů komunikace elektroměru PQ monitor MEg44PAN. Ukázka kódu řešení je zobrazena ve výpisu 2.5.

Výpis 2.5: Výpis spontánního zasílání zpráv.

```
1                   /*NASTAVENI CASU PRO DOTAZ*/
2  int cas_intervalu=getRandomTime(); //nahodna hodna 5-20vterin
3  int cas_prubezny=0;
4    if (cas_prubezny>=cas_intervalu){
5
6                   /*VYTVORENI ASDU*/
7      CS101_ASDU newAsdu = CS101_ASDU_create(alParams,
8        false, CS101_COT_SPONTANEOUS, 0, 1, false, false);
9
10                  /*VYTVORENI INFORMACNIHO OBJEKTU*/
11     InformationObject io = (InformationObject)
12       MeasuredValueShortWithCP56Time2a_create(NULL, 2,
13       getRandomValueUf(), IEC60870_QUALITY_GOOD,
14       CP56Time2a_createFromMsTimestamp
15       (NULL, Hal_getTimeInMs()));
16
17                  /*ZAPIS DO LOGOVACIHO SOUBORU*/
18     zapisTXC(InformationObject_getObjectAddress
19     ((InformationObject)io), MeasuredValueShort_getValue
```

```
20      ((MeasuredValueShort)io), CP56Time2a_createFromMsTimestamp
21      (NULL, Hal_getTimeInMs()));
22
23                  /*ODESLANI ZPRAVY*/
24      CS104_Slave_enqueueASDU(slave, newAsdu);
25
26                  /*ZNICENI INFORMACNIHO OBJEKTU A ASDU*/
27      InformationObject_destroy(io);
28      CS101_ASDU_destroy(newAsdu);
29
30      cas_intervalu=getRandomTime();
31      }
```

Dále bylo taktéž potřeba vytvořit záznam odeslaných zpráv pro případné testování či analýzu přenesených dat. Tyto soubory se zapisují do souboru TXC.txt. Tento záznam obsahuje časové razítko, kdy byla daná zpráva odeslána. Toto časové razítko má stejný formát jako u předešlých scénářů. Dále obsahuje adresu informačního objektu (v našem případě je IOA=2, neboť máme pouze jeden informační objekt, který představuje velikost efektivního napětí U_f) a jeho velikost. Ukázka struktury je zachycena na obrázku 2.9.



Obr. 2.9: Logovací soubor TXC.txt.

2.3.4 Generátor s volitelným obsahem datové části

Tento generátor představuje spojení všech tří předchozích scénářů za účelem vytvoření generátoru komunikace, u kterého můžeme měnit datovou část a celkový objem generovaných dat. U tohoto generátoru můžeme nastavit, zda-li bude pracovat v režimu NN či VN rozvaděče. U každého z nich si můžeme zvolit, jaký počet „Single point information with time tag CP56Time2a“, „Double point information with time tag CP56Time2a“, „Measured value, short floating point value with time tag CP56Time2a“, „Single command“ a „Double command“ bude generován. Další

volbou je, kolikrát se v jednom čase tyto zprávy vygenerují. To slouží k razantnímu zvýšení datového toku, neboť se zadaným číslem n se nám datový tok n-násobně zvětší. Na obrázku 2.5 je zobrazeno menu tohoto generátoru na straně klienta.

Na straně serveru je také přítomna funkce na zasílání zpráv v náhodném čase, obdobně jako u scénáře spontánního zasílání zpráv, ovšem s jinými daty. Tato funkce se však dá při spouštění serveru vypnout. Možnost vypnutí této funkce je zde z důvodů, kdy testujeme pouze datový tok žádaných zpráv a nechceme linku zatěžovat dalšími daty. Na obrázku 2.6 jsme již mohli vidět menu tohoto generátoru na straně serveru.

Při vyžádání určitého objemu dat v režimu VN nebo NN jsou na server poslány zprávy obsahující počty požadovaných zpráv a počet opakování v jednom čase. Za tímto účelem posílá zprávu „Bit string 32 bit with time tag CP56Time2a“, která v sobě oproti normálnímu příkazu nese také časové razítko. Tuto zprávu server zpracuje a následně odešle potřebná data. Jako identifikátor konkrétního požadavku se zde používá časové razítko, kdy byl požadavek vytvořen. Ukázka kódu řešení pro klientskou stanici je zobrazena ve výpisu 2.6.

Výpis 2.6: Výpis řešení generátoru s volitelným obsahem datové části.

```
1  main(int argc, char** argv){
2
3                  /*PROJIZDENI SEZNAMU IP ADRESS*/
4    while(stav){
5      if(counter==sizeOfListIP)
6        counter=0;
7      ip=ip_arr[counter];
8
9                /*PODMINKA AKTIVNIHO SPOJENI*/
10     if (CS104_Connection_connect(con)) {
11       CS104_Connection_sendStartDT(con);
12
13               /*ODESLANI POZADAVKU*/
14       InformationObject io=(InformationObject)
15         Bitstring32CommandWithCP56Time2a_create(NULL,30,
16         num_sp,CP56Time2a_createFromMsTimestamp(NULL,
17         Hal_getTimeInMs()));
18       CS104_Connection_sendProcessCommandEx(con,
19         CS101_COT_INTERROGATED_BY_GROUP_1, 1, io);
20       InformationObject_destroy(io);
21
22               /*ZAPIS DO LOGOVACIHO SOUBORU*/
23       log_request(sp,dp,me, num_opakovani,
```

```
24          CP56Time2a_createFromMsTimestamp(NULL,
25          Hal_getTimeInMs()));
26        log_RX_timeID(CP56Time2a_createFromMsTimestamp
27          (NULL, Hal_getTimeInMs()));
28      }else
29        printf("Connect failed!\n");
30      counter+=1;
31    }
32  }
```

Obdobně jako u testovacích scénářů jsou zde přítomny taktéž logovací soubory, které zde hrají významnější roli, neboť je zde generováno mnohonásobně více dat a při zpětné analýze komunikace se můžeme spoléhat pouze na tyto soubory. Každý logovací soubor obsahuje adresy informačních objektů a jejich hodnoty. Dále je zde zaznamenáno časové razítko, v případě že jej daná zpráva obsahuje, a časové razítko, kdy byla tato zpráva přijata pro zjištění zpoždění mezi stanicemi.

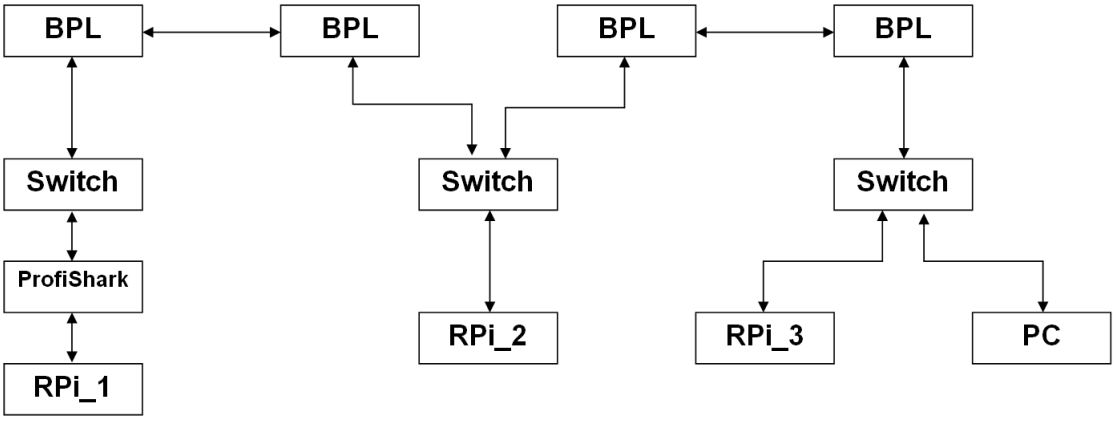
Tyto logy tvoří na straně klienta a serveru pomyslné dvojice, které za normální situace obsahují totožná data. Jsou zde logy pro odeslané a přijaté zprávy obsahující měřené proměnné, logovací soubory pro příkazy a logovací soubory pro spontánní zprávy.

3 Testování simulované komunikace

Tato část bakalářské práce se zabývá testováním vytvořených testovacích scénářů a generátoru komunikace s volitelnou datovou částí protokolu IEC 60870-5-104. Realizace bude probíhat v laboratorní síti mezi dvěma a více zařízeními Raspberry Pi 3 model B+ s operačními systémy Raspbian.

První část testování bude zaměřena na testovaní komunikačních scénářů tvořící dílčí části komunikace elektroměru PQ monitor MEg44PAN od společnosti MEgA. Druhá část pak bude zaměřena na testování komplexního generátoru komunikace. Tento generátor již spojuje všechny vlastnosti a funkce testovacích scénářů, za účelem vytvoření emulátoru komunikace NN a VN rozvaděče na zařízení Raspberry Pi. Následně bude tato komunikace zaznamenána pomocí zařízení ProfiShark a porovnána se záznamem reálného zařízení.

Na obrázku 3.1 je zobrazeno schéma zapojení komunikační sítě. Celé zapojení se nachází v síti 192.168.100.0/24. Klientská stanice má pevně nastavenou IP adresu 192.168.100.110 a plní funkci nadřazené stanice, která napodobuje komunikaci reálného zařízení PQ monitoru MEg44PAN. Oproti tomu servery plní funkci podřízených stanic, které generují zprávy. Tyto servery mají taktéž pevně dané IP adresy a to 192.168.100.111 a 192.168.100.114.



Obr. 3.1: Schéma zapojení sítě.

3.1 Konfigurace prostředí

Vzhledem k faktu, že se jedná o programy běžící na samostatných zařízeních Raspberry Pi, je budeme spravovat vzdáleně a to z centrálního počítače (ve schématu 3.1 označen jako PC s IP adresou 192.168.100.2), na kterém běží ve virtuálním prostředí operační systém OpenSCADA.

Nejdříve se vzdáleně připojíme na jednotlivá zařízení například pomocí příkazu *ssh*, ve formě *ssh@192.168.100.110* pro klientskou stanici RPi_1. Na tomto zařízení zkontrolujeme příkazem *dpkg –list*, zda-li obsahuje nejnovější balíčky a nainstalované balíčky gcc verze 6.3., make verze 4.9.1 a vyšší a nakonec i linkovací knihovnu lpthread.

Poté musíme na cílové zařízení překopírovat upravenou knihovnu lib60870–2.1.1, která v sobě obsahuje potřebné soubory. Tato knihovna je nahrána jako příloha této bakalářské práce. Nejjednodušší způsob je použít příkaz *scp* ve formě scp <source> <destination>, kde na místo source a destination dosadíme již konkrétní lokaci. Jednou z výhod je fakt, že se tato knihovna nemusí instalovat na koncové zařízení, neboť potřebné soubory ke spuštění jsou přímo v této knihovně.

3.1.1 Spuštění klientské stanice

V předpřipravené knihovně lib60870–2.1.1 se pomocí příkazu *cd* odnavigujeme do složky pi@raspberrypi:~\lib60870-2.1.1\lib60870-C\generator104\cs104_client. Zde se nachází soubor cs104_client.c, který zkompilujeme tak, že do konzole zadáme příkaz *make*. Po zadání tohoto příkazu se nám vytvoří spustitelný soubor cs104_client, který můžeme jednoduše spustit příkazem *./cs104_client*.

Dále se nás program dotáže, zda-li souhlasíme se seznamem přednastavených IP a v případě, že souhlasíme, zadáme *y*. V opačném případě zadáme *n* a můžeme tento seznam upravit dle naší potřeby. Následně se nás program zeptá, v jakém režimu jej chceme spustit. Poté se nás dotáže, zda-li chceme načíst přednastavenou konfiguraci, která odpovídá typickému nastavení pro daný rozvaděč u zařízení PQ monitoru MEg44PAN. V případě, že odmítneme, si můžeme postupným dotazováním sami navolit jaký počet a jaké datové toky se budou generovat. Posledním parametrem zadání je počet opakování vysílaných zpráv v jednom čase. Poté již program začne navazovat spojení s jednotlivými servery.

3.1.2 Spuštění serverové stanice

Při spouštění serveru postupujeme obdobně jako u spouštění klientské stanice. Pomocí příkazu *cd* se dostaneme do složky pi@raspberrypi:~\lib60870-2.1.1\lib60870-C\generator104\cs104_server. Poté vše zkompilujeme zadáním příkazu *make* do konzole. Po zadání tohoto příkazu se nám vytvoří spustitelný soubor cs104_server, který můžeme jednoduše spustit příkazem *./cs104_server*.

Po spuštění nás program požádá o zadání IP adresy daného serveru, kterou si můžeme zjistit příkazem *ifconfig*. Následně se nás dotáže, zda-li chceme zapnout funkci na generování spontánních zpráv. Poté by se nám měla do konzole vypsat hláška „Server running...“. V opačném případě, při vypsání hlášky „Starting server

failed!“, nedošlo ke korektnímu nastavení serveru. To může zapříčinit špatné zadání IP adresy nebo fakt, že na daném zařízení již takový server běží.

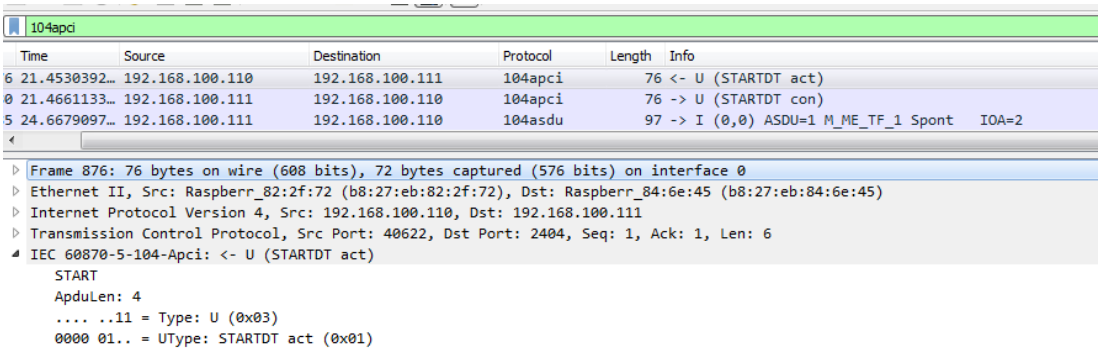
3.1.3 Konfigurace zařízení ProfiShark

Zařízení ProfiShark slouží jako přemosťovací zařízení, které jednoduše připojíme mezi dvě komunikující zařízení v síti. Ve schéma 3.1 je vloženo mezi zařízení RPi_1 a Switch. Tím dosáhneme efektu, kdy se nám bude zaznamenávat komunikace jak od RPi_2, tak i od RPi_3, neboť zařízení RPi_1 představuje klientskou stanici, do které směřují veškerá data ze serverů. Následně zařízení ProfiShark připojíme k centrálnímu počítači (ve schématu 3.1 označen jako PC).

3.2 Rozbor komunikace vytvořených programů

Komunikace probíhá stylem klient–multi server. Klient posílá v periodickém čase žádosti o zaslání dat od jednotlivých serverů. Také posílá příkaz na vzájemnou synchronizaci času. Serverové stanice generují data a zasílají odpovědi v definovaném formátu ASDU na klientskou stanici. V našem případě se jedná o stavy proměnných a hodnoty fyzikálních veličin běžné pro elektroměr PQ monitor MEg44PAN.

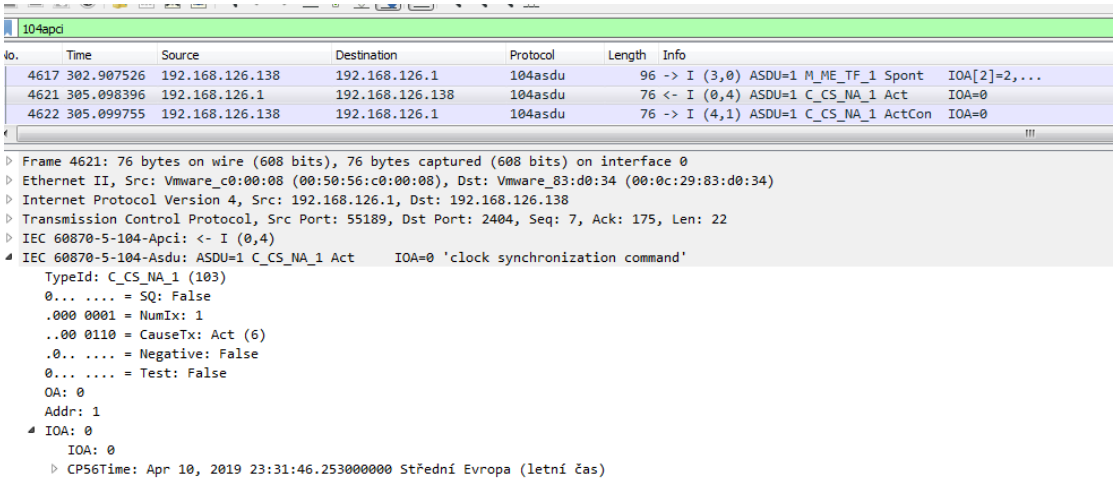
Při navazování spojení mezi klientem a serverem dochází z klientské stanice k vytvoření TCP spojení na předdefinovaném cílovém portu 2404. Poté odešle APCI zprávu STARTDT act, kterým zjišťuje, jestli je daný server připraven ke vzájemné komunikaci. V případě, že server tuto zprávu obdrží a je připraven, odpoví na ni zprávou STARTDT con. V opačném případě na tuto zprávu nereaguje. Poté je komunikace úspěšně navázána. Na obrázku 3.2 je zobrazena APCI zpráva STARTDT act.



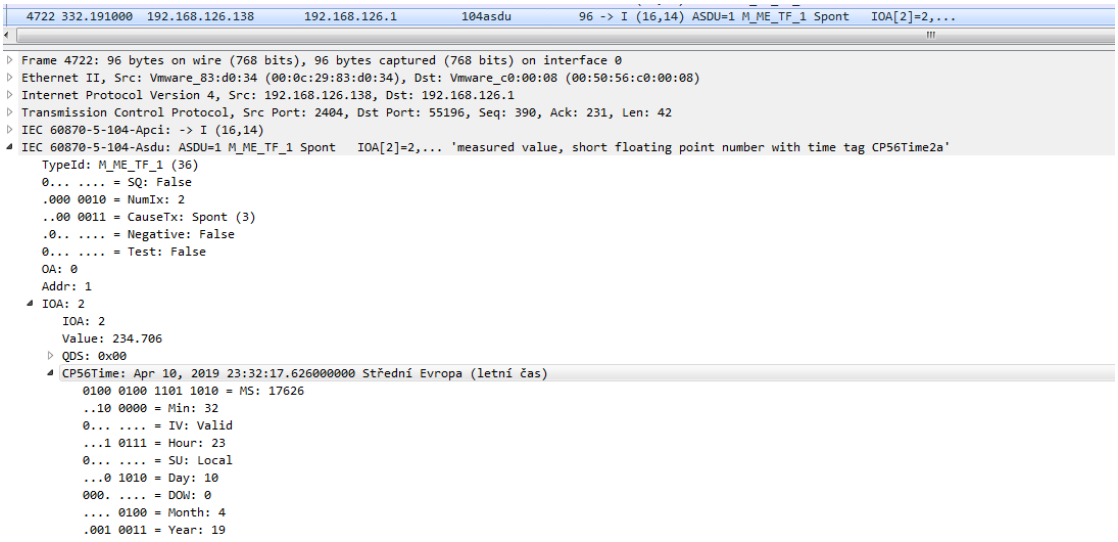
Obr. 3.2: Zpráva STARTDT act.

V rámci těchto programů se dá měnit obsah ASDU zpráv za předpokladu opětovného spuštění. Tento program se nedá v průběhu komunikace modifikovat, neboť je nutné jej restartovat. Ovšem v rámci vývojového prostředí se zde mohou nastavit

zcela odlišné typy zpráv, dotazy a příkazy. Dají se zde také měnit intervaly zasílání cyklických a spontánních zpráv či měnit dobu pro vyslání příkazu na synchronizaci času. Na obrázku 3.3 je zobrazena ASDU zpráva pro synchronizaci času. Tato synchronizace času je nedílnou součástí pro správné fungování časového razítka. Ukázka časového razítka je zobrazena na obrázku 3.4.



Obr. 3.3: Příkaz na synchronizaci času.



Obr. 3.4: Časové razítko.

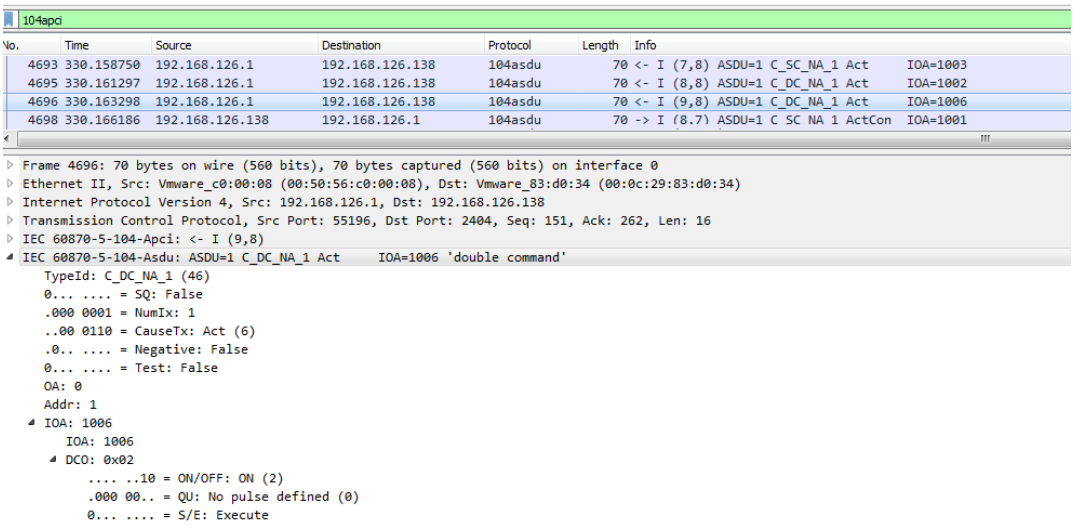
A nakonec při ukončení spojení mezi klientem a serverem dojde k tomu, že klient zruší spojení s daným serverem a tím dojde i k ukončení spojení přes TCP protokol. Server tedy nemá kam dané zprávy posílat a ukládá si je do odesílací fronty. Po opětovném připojení klienta, v dalším periodickém cyklu, jsou tyto zprávy odeslány, takže se žádné zprávy nezahodí.

3.2.1 Testování pomocných scénářů

V rámci vytváření generátoru komunikace protokolu IEC 60870-5-104 bylo potřeba nejprve otestovat pomocné scénáře, které reprezentovaly dílčí části finálního generátoru, za účelem odhalení chyb a nedostatků.

Testování scénáře dotazu a příkazu k nastavování hodnot

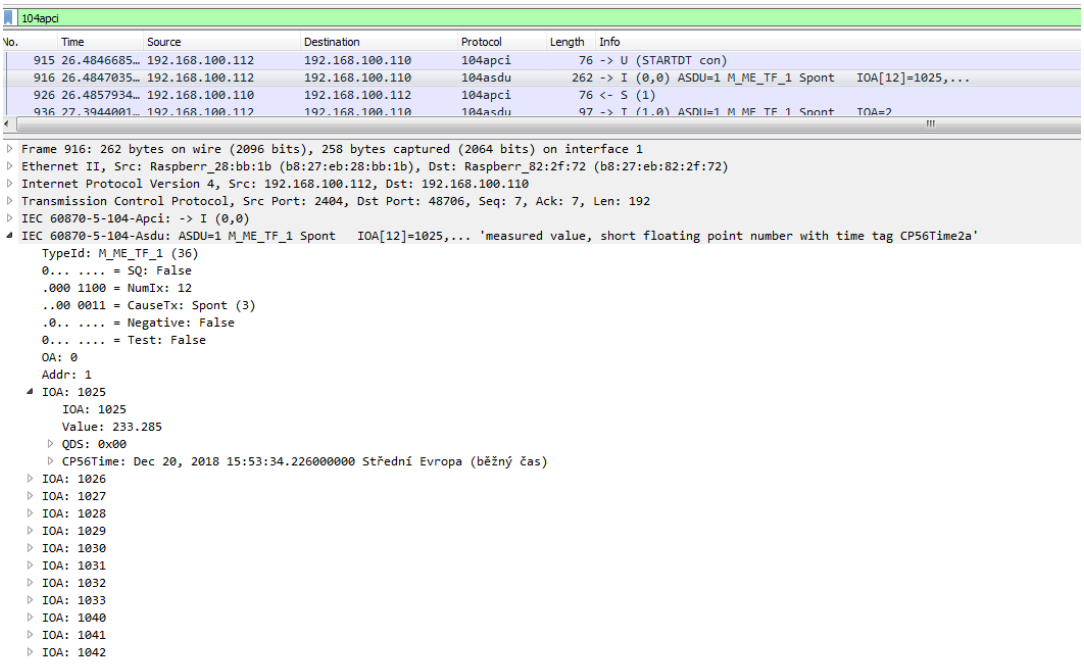
V tomto komunikačním scénáři dochází k periodickému dotazování se na stavy určitých proměnných a také k příkazům k nastavení určitých proměnných. Každý informační objekt představuje již konkrétní proměnou. V našem případě IOA=1001 představuje proměnnou vypínače, kdy má pouze 2 stavy (0/1). Proto se pro přenos tohoto typu proměnné používá typ „Single point information“. Dále se zde vyskytuje například IOA=1006, který představuje přepínač, který může nabývat více než dvou stavů. Pro přenos této proměnné používáme již informační objekt typu „Double point information“. Ukázka ASDU nesoucí příkaz ke změně hodnoty, konkrétně ke změně proměnné s IOA=1006, je zobrazena na obrázku 3.5.



Obr. 3.5: Příkaz pro nastavení hodnoty na serveru.

Testování scénáře periodického výčtu dat

V rámci tohoto scénáře dochází k periodickému přenosu více hodnot proměnných naráz směrem od serveru ke klientovi. Význam těchto proměnných byl již popsán v tabulce 2.1. Čas pro odesílání je nastaven na 1 minutu, ovšem v praxi se převážně používají časové intervaly 10 či 15 minut. Ukázka zachycené zprávy testovacího scénáře periodického výčtu dat je zobrazena na obrázku 3.6.



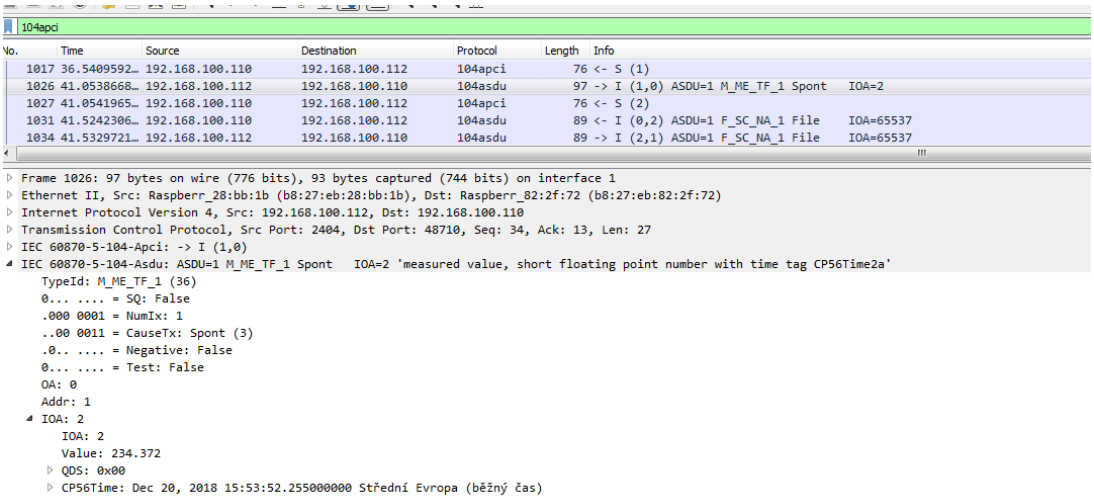
Obr. 3.6: Zpráva periodického výčtu dat.

Testování scénáře spontánního zasílání zpráv

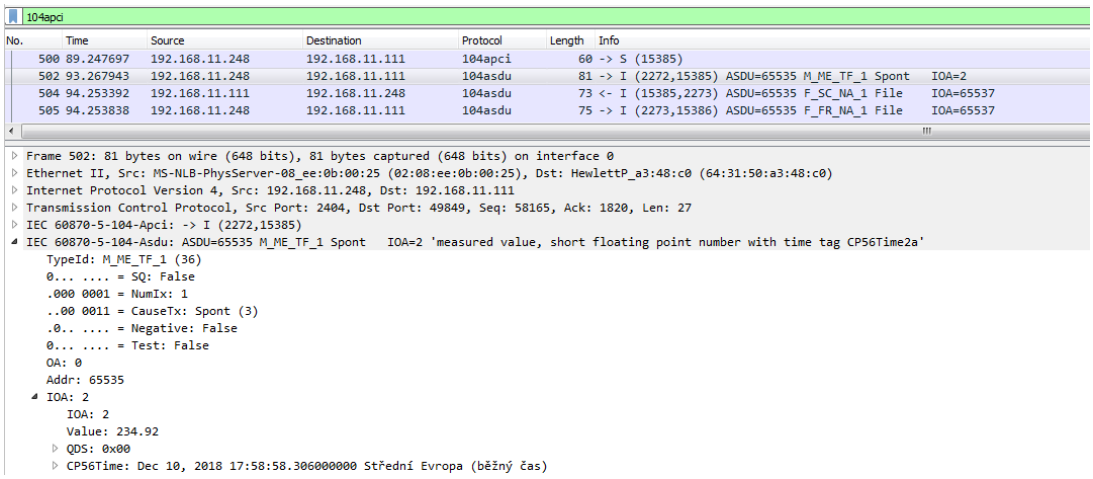
Poslední testovací scénář je zaměřen na odesílání zpráv ve spontánním čase. Tento čas se generuje náhodně v rozmezí od 5 do 20 vteřin. Když tento čas nastane, server vygeneruje hodnotu proměnné U_f, zpracuje ji do potřebného formátu a následně ji odešle na klientskou stanici. V praxi se tato zpráva posílá při změně měřené hodnoty efektivního napětí nebo při překročení určité bezpečné hranice, za účelem informování klientské stanice. Na obrázku 3.7 je zobrazena zachycená spontánní zpráva vytvořeného testovacího scénáře a na obrázku 3.8 je zachycena originální spontánní zpráva elektroměru PQ monitor MEg44PAN. Jsou zde patrné drobné odlišnosti, ovšem v porovnání s originální zprávou jsou téměř identické.

3.2.2 Testování komunikace vytvořeného generátoru

V rámci tohoto generátoru dochází k cyklickému navazování spojení mezi klientem a serverem, kdy při každém navázání spojení klient zašle serveru dotaz prostřednictvím „Bit string 32 bit with time tag CP56Time2a“ zprávu pro zaslání vyžádaných dat. Oproti elektroměru PQ monitor MEg44PAN, který zasílá zprávu „Interrogation command“, obsahuje mimo jiné i časové razítko. Jedná se tedy o vhodnější variantu při vytváření generátoru komunikace, u kterého chceme dále pracovat s logovacími soubory.



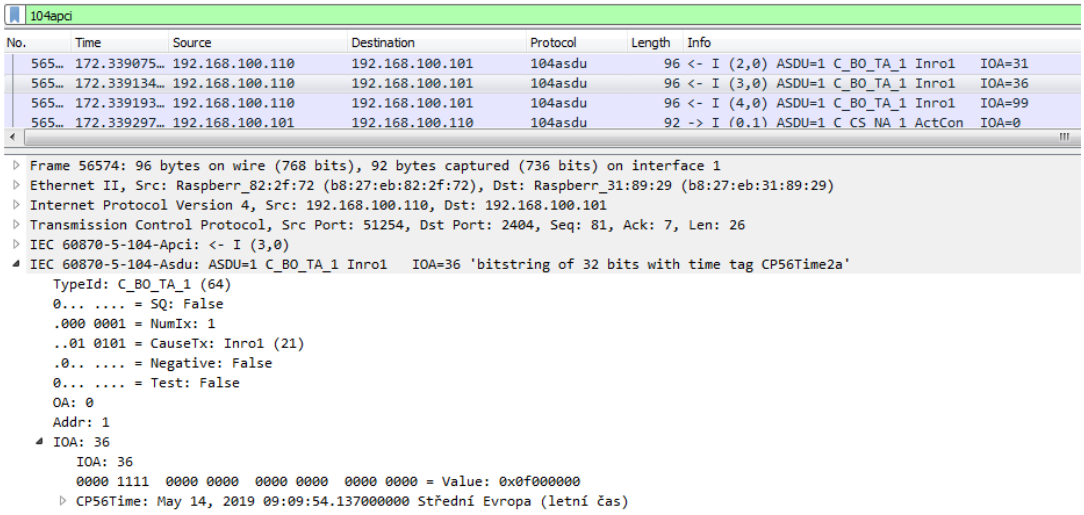
Obr. 3.7: Spontánní zpráva testovacího scénáře.



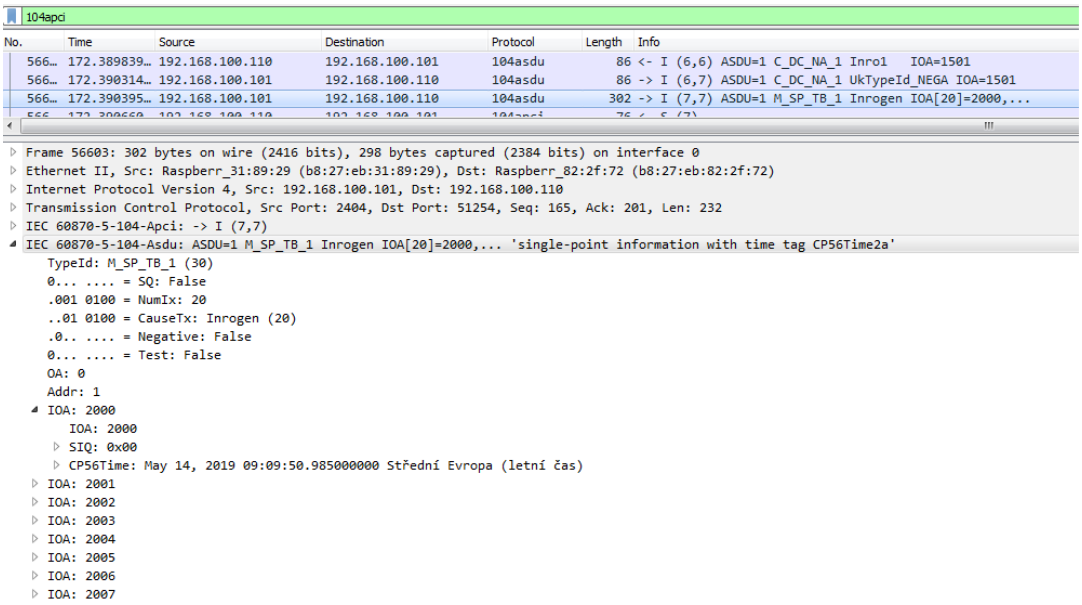
Obr. 3.8: Spontánní zpráva elektroměru PQ monitor MEg44PAN.

Na obrázku 3.9 je zachycena zpráva dotazu pro serverovou stanici vytvořeného generátoru. Oproti originálnímu dotazu elektroměru PQ monitor MEg44PAN, který je zobrazen na obrázku 2.1, se jedná sice o jiný typ zprávy, ovšem z pohledu serveru se jedná o totožný dotaz. Jediný znatelnější rozdíl je v již zmíněném použití časového razítka.

Poté co server obdrží tuto zprávu a zpracuje ji, odešle odpověď, do které vloží žádané množství informačních objektů. Obrázek 3.10 ukazuje takovouto odpověď. V tomto případě se jedná o odpověď na dotaz k zaslání 20-ti „Single point information with time tag CP56Time2a“. Originální odpověď u elektroměru PQ monitor MEg44PAN může být zprávou „Single point information“, která je bez časového razítka, nebo „Single point information with time tag CP56Time2a“ viz obrázek 2.2. Pro účel tohoto generátoru je tedy vhodnější použít zprávu s časovým razítkem.



Obr. 3.9: Zpráva dotazu vytvořeného generátoru.

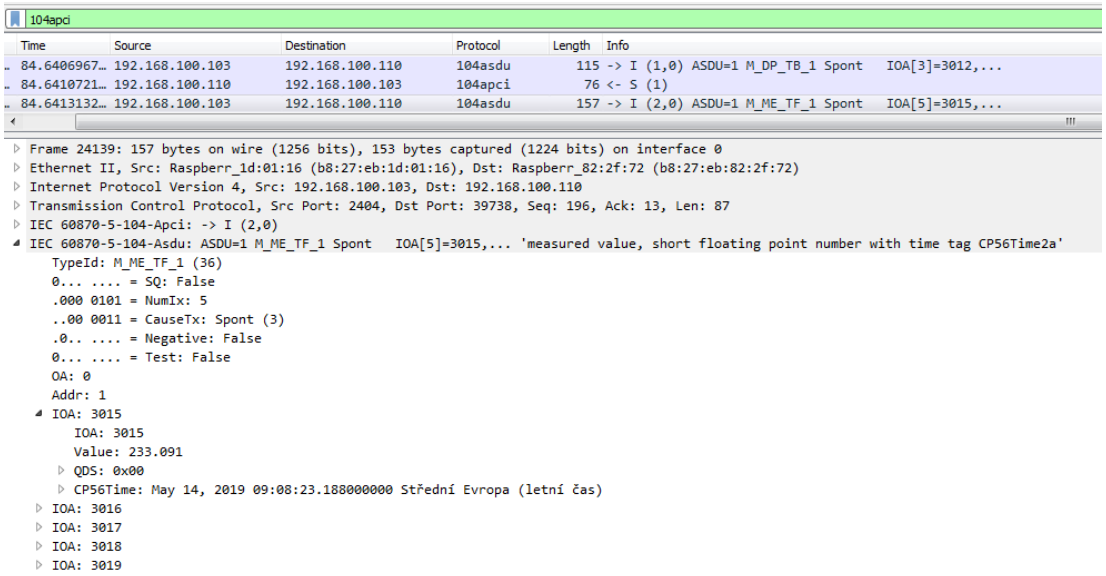


Obr. 3.10: Zpráva odpovědi vytvořeného generátoru.

U zprávy typu „Single command“ a „Double command“ je postupováno obdobně, jen s tím rozdílem, že na ni odpovídáme zprávou „Single point information with time tag CP56Time2a“, respektive „Double point information with time tag CP56Time2a“.

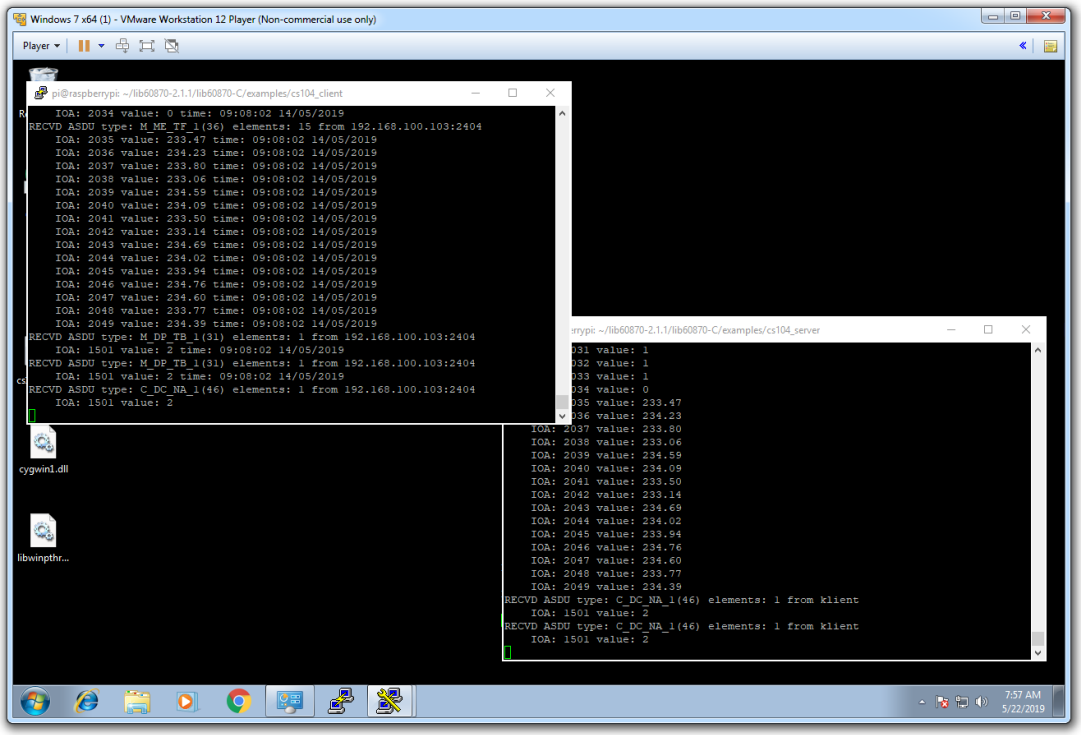
Poslední funkcionalitou tohoto generátoru je zasílání spontánních zpráv směrem od serveru ke klientovi. Jsou zde posílána téměř totožná data jako u vyžádaných dat. Jediný rozdíl je v množství posílaných dat a faktu, že přenos těchto dat je inicializován serverem. Tato data jsou pak posílána spontánně v intervalu od 5 do 30

sekund. Obrázek 3.11 zachycuje spontánní zprávu vytvořeného generátoru nesoucí 5 informačních objektů s hodnotou náhodné veličiny představující velikost efektivního napětí.

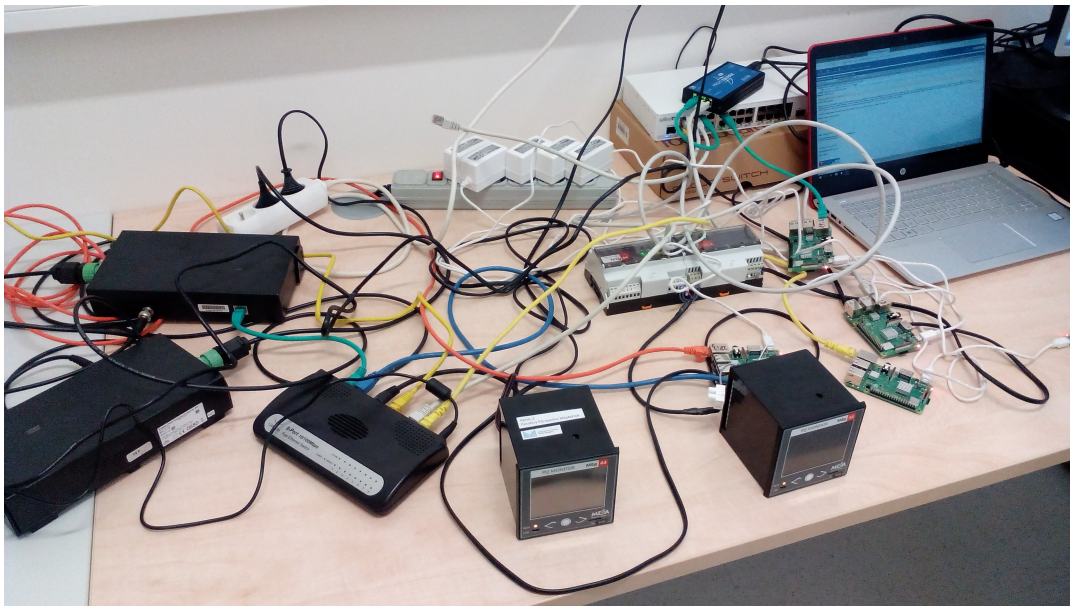


Obr. 3.11: Spontánní zpráva vytvořeného generátoru.

Na obrázku 3.12 je zachycen běh komunikace vytvořeného generátoru protokolem IEC 60870-5-104 mezi klientem s IP adresou 192.168.100.110 a serverem s IP adresou 192.168.100.103. A nakonec na obrázku 3.13 můžeme vidět reálné pracoviště.



Obr. 3.12: Záznam emulování komunikace protokolu IEC 60870-5-104.



Obr. 3.13: Ukázka pracoviště.

4 Závěr

Náplní této bakalářské práce bylo seznámení se se SCADA komunikačními standardy (DNP3, IEC 60870 a IEC 61850), se zaměřením na protokol IEC 60870-5-104. Této problematice je věnována první teoretická část, která popisuje již zmíněné protokoly, uvádí důvody jejich vzniku a krátce shrnuje jejich historii. Dále jsou zde popsány normy, ze kterých tyto protokoly vychází, a jejich architektura vzhledem k modelu ISO/OSI. Také jsou zde popsány vlastnosti a především způsoby, jakým se jednotlivé typy protokolů utváří a komunikují.

V rámci praktické části této práce byl realizován generátor komunikace protokolu IEC 60870-5-104, u kterého můžeme nastavit velikost datové části zprávy a celkový objem generovaných dat. Tento generátor emuluje chování zařízení PQ monitor MEg44PAN od společnosti MEgA v režimu NN a VN rozvaděče. V rámci jeho realizace byly vytvořeny 3 testovací komunikační scénáře, které simulují dílčí části komunikace tohoto elektroměru. Tato komunikace probíhá stylem klient–server, kde klientská stanice dokáže přijímat data z jednoho či více serverů. Dále tyto programy obsahují příkazy pro vzájemnou synchronizaci času, která je pro tento protokol velice důležitá. Také jsou zde přítomny funkce pro vytváření záznamových logů pro další testování a analýzu dat.

Poslední částí bylo samotné testování vytvořených komunikačních scénářů, které tvořily základní části výsledného generátoru komunikace protokolu IEC 60870-5-104. Výsledný generátor a samotné testovací scénáře byly testovány mezi třemi zařízeními Raspberry Pi. Toto testování probíhalo na reálné síti v laboratoři VUT, za účelem testování zařízení PQ monitor MEg44PAN a generování komunikačního šumu pro jiné protokoly.

Literatura

[1] MAKHIJA, J; SUBRAMANYAN, L.R. *Comparison of protocols used in remote monitoring: DNP 3.0, IEC 870-5-101 and Modbus.* Electronics Systems Group, IIT Bombay, India, Tech. Rep, 2003

[2] *IEC 61850 Communication Networks and Systems In Substations: An Overview for Users.* 2009, [online]. Copyright ©Ig [cit. 20.11.2018]. Dostupné z URL: <http://www.gegridsolutions.com/multilin/journals/issues/spring09/iec61850.pdf>

[3] UZAIR, M. *Communication methonds (Protocols, format and language) for the substation automation and control (Project report od course 586 b).* [cit. 20. 11. 2018]. Dostupné z URL: <http://www.eng.uwo.ca/people/tsidhu/Documents/project%20report%20Uzair.pdf>

[4] CLARKE, G; REYNDERS, D; WRIGHT, E. *Practicalmodern SCADA protocols: DNP3, 60870.5 and relatedsystems.* 2004, [online]. Dostupné z URL: <https://www.julesbartow.com/Pictures/RF/Practical_modern_SCADA_protocols_-_dnp3,_60870-5_and_Related_Systems.pdf>

[5] MATOUŠEK, P. *Description and analysis of IEC 104 Protocol.* 2017, [online]. Copyright © [cit. 21.11.2018]. Dostupné z URL: <http://www.fit.vutbr.cz/research/view_pub.php.cs?id=11570>

[6] *Komunikační protokoly pro dálkové ovládání IEC/ISO 60870-5.* 2010, [online]. Dostupné z URL: <http://automa.cz/cz/casopis-clanky/komunikacni-protokoly-pro-dalkove-ovladani-iec/iso-60870-5-2010_02_40552_5799/>

[7] VLADYKA, B; VLADYKA, P. IEC 61850. *Soubor norem pro komunikaci v energetice s velkým potenciálem výhod.* 2010, [online]. Copyright ©US [cit. 15.03.2019]. Dostupné z URL: <http://automa.cz/Aton/FileRepository/pdf_articles/40771.pdf>

[8] SCADA Guardian 848. *Remote Monitoring and Control Systems Company | DPS Telecom* [online]. Dostupné z URL: <https://www.dpstele.com/dnp3/index.php>

[9] MATOUŠEK, P. *Description of IEC 61850 Communication.* [online]. Copyright © [cit. 16.03.2019]. Dostupné z URL: <http://www.fit.vutbr.cz/research/pubs/index.php?file=%2Fpub%2F11832%2FTR-61850.pdf&id=11832>

[10] ČSN EN 60870-5-104 (334650). *Systémy a zařízení pro dálkové ovládání – Část 5-104: Přenosové protokoly – Síťový přístup pro IEC 60870-5-101 používající normalizované transportní profily*, 2007.

[11] Downloads | LibIEC61850 / LibIEC60870-5| *Open source librariesfor IEC 61850 and IEC 60870.* 2018, [online]. Dostupné z URL: <`https://libiec61850.com/libiec61850/downloads/`>

[12] Raspberry Pi Downloads - Software for the Raspberry Pi. *Teach, Learn, and Make with Raspberry Pi – Raspberry Pi* [online]. Dostupné z URL: <`https://www.raspberrypi.org/downloads/`>

[13] Monitor MEg44PAN | MEgA - Měřící Energetické Aparáty, a.s.. *Úvodní stránka MEgA - Měřící Energetické Aparáty, a.s.* [online]. Copyright © [cit. 16.03.2019]. Dostupné z URL: <`http://www.e-mega.cz/?pg=meg44pan`>

[14] lib60870-C: README lib60870-C. [online]. Dostupné z: <`https://support.mz-automation.de/doc/lib60870/latest/index.html`>

[15] IDA > Home. *Understanding IEC-60870-5-104 Traffic Patterns in SCADA Networks* [online]. Copyright © [cit. 16.03.2019]. Dostupné z: <`https://www.ida.liu.se/labs/rtslab/publications/2018/CPSS_2018.pdf`>

[16] Beckhoff Information System - English. *Beckhoff Information System - German* [online]. Dostupné z: <`https://infosys.beckhoff.com/english.php?content=../content/1033/tcplclibiec870_5_104/html/tcplclibiec870_5_104_objref_overview.htm&id`>

[17] Brodersen simplifying systems. *RTU32 IEC60870 Drivers* [online]. Copyright © [cit. 16.04.2019]. Dostupné z: <`http://brodersen.com/wordpress/wp-content/uploads/BS_RTU32_IEC60870Config.pdf`>

Seznam příloh

A Příloha

V příloze je s prací odevzdána také upravená knihovna *lib60870-2.1.1*, která obsahuje soubory cs104_client.c a cs104_server.c. Dále pak na webové adrese <`https://drive.google.com/drive/u/1/folders/1pAUkahH6E972TDUXVCb6yQ5xT19O6nNX`> jsou poskytnuty veškeré programy, včetně testovacích scénářů a pcap soubory se záznamy těchto komunikací.