

Návrh a implementácia efektívneho komunikačného protokolu pre IoT meteorologické zariadenia

Zadanie z predmetu Počítačové a Komunikačné siete 2025/26

Kontext zadania:

Spoločnosť FIITMeteo vyvíja a nasadzuje meteorologické IoT senzory určené na dlhodobý zber meteorologických údajov. Zariadenia sú napájané z batérie a z tohto dôvodu je efektivita prenášaných dát rozhodujúcim faktorom. Pre tieto zariadenia je nutné pripraviť proprietárny komunikačný protokol, ktorým budú komunikovať so serverom. Nutné je aj naprogramovať aj samotný server spolu s testovacím programom, ktorý bude emulovať jednotlivé zariadenia.

Príbeh:

Predstavte si, že ste vo firme FIITMeteo na pozícii junior developera. Keďže ste vo firme na tejto pozícii už niečo vyše roka, rozhodol sa vás váš senior developer (*váš cvičiaci*) preveriť stredne náročnou úlohou – prvý samostatným projektom, pri ktorom budete musieť navrhnúť protokol a implementovať dva programy podľa požiadaviek klienta. Keďže ste takúto úlohu ešte nerobili, dohodol si s vami váš senior developer stretnutie v polovici trvania projektu, kde mu máte predviesť PoC (Proof-of-Concept) vášho riešenia – stačí, ak bude protokol navrhnutý a implementovaný pomocou štruktúr JSON (*Prvé odovzdanie zadania*). Po tomto stretnutí začnete protokol optimalizovať na binárny; keďže ide o IoT zariadenia napájané batériou, každý ušetrený bajt prenosu predlžuje životnosť zariadenia. Ako to už ale býva, klient niekedy nevie čo presne chce – sedíte v zasadačke spolu s vašim senior developerom a pred tým ako idete projekt odovzdať vám ešte rýchlo zavolá projektový manažér (*cvičiaci zodpovedný za zadanie*), že zabudol na jednu zmenu, na ktorej sa s klientom dohodli. Táto zmena nie je náročná ani veľká, ale pre klienta dôležitá a keďže máte ešte pol hodinu do stretnutia, rozhodli ste sa túto zmenu zapracovať (*doimplementácia*). Keď túto zmenu zapracovať zvládnete, prejdete si s klientom najprv *akceptačné testy* t. j. testy, ktoré musí vaše riešenie splniť, aby mohlo byť prebraté klientom. Potom odprezentujete projekt klientovi (*finálne preberanie zadania cvičiacim*).

Pred tým ako vám bol tento projekt zadaný, projektový manažér pripravil dokument na základe stretnutí s klientom, podľa ktorého budete musieť projekt vypracovať. V dokumente sú uvedené aj *Akceptačné testy (minimálne požiadavky zadania)*. Zároveň je tam aj vypracovaný systém ohodnotenia projektu, ktorý bol s klientom dohodnutý (*bodovanie*). Váš senior

developer je voči vám transparentný – ak nezískate pre firmu aspoň polovicu ohodnotenia (získanie polovice bodov pri prvom odovzdaní a získanie polovice bodov pri finálnom odovzdaní), nebude vás môcť odporučiť na vyššiu pracovnú pozíciu (pripustenie ku skúške).

Zápis zo stretnutia s klientom:

(požiadavky na zadanie)

Klient si vybral nasledujúce zariadenia pre svoj projekt:

1. ThermoNode – 4 parametre

Zameranie: Teplota, vlhkosť a tlak

- temperature: teplota v °C (-50.0 až +60.0, desatinné číslo – 1 desatinné miesto)
- humidity: relatívna vlhkosť v % (0.0 – 100.0, desatinné číslo – 1 desatinné miesto)
- dew_point: rosný bod v °C (-50.0 až +60.0, desatinné číslo – 1 desatinné miesto)
- pressure: atmosférický tlak v hPa (800.00 – 1100.00 - , desatinné číslo – 2 desatinné miesta)

2. WindSense – 4 parametre

Zameranie: Rýchlosť a charakter vetra

- wind_speed: rýchlosť vetra v m/s (0.0 – 50.0, desatinné číslo – 1 desatinné miesto)
- wind_gust: náraz vetra v m/s (0.0 – 70.0, desatinné číslo – 1 desatinné miesto)
- wind_direction: smer vetra v stupňoch (0 – 359, celočíselné)
- turbulence: index turbulencie (0.0 – 1.0, desatinné číslo – 1 desatinné miesto)

3. RainDetect – 4 parametre

Zameranie: Zrážky a pôda

- rainfall: úhrn zrážok za poslednú hodinu v mm (0.0 – 500.0, desatinné číslo – 1 desatinné miesto)
- soil_moisture: vlhkosť pôdy v % (0.0 – 100.0, desatinné číslo – 1 desatinné miesto)
- flood_risk: úroveň rizika povodne (0 – none, 1 – low, 2 – medium, 3 – high, 4 hodnoty)
- rain_duration: trvanie zrážok v minútach za poslednú hodinu (0 – 60, celočíselné)

4. AirQualityBox – 3 parametre

Zameranie: Kvalita ovzdušia

- co2: koncentrácia CO₂ v ppm (300 – 5000, celočíselné)
- ozone: koncentrácia ozónu v µg/m³ (0.0 – 500.0, desatinné číslo – 1 desatinné miesto)
- air_quality_index: AQI (0 – 500, celočíselné)

Požiadavky:

- P 1.** Návrh a implementácia proprietárneho protokolu nad protokolom UDP.
- P 2.** 2 programy: server, ktorý spracováva a zobrazuje údaje, testovací program, ktorý emuluje všetky zariadenia (jeden program – všetky zariadenia).
- P 3.** Každý senzor si musí registračnou správou vypýtať token od servera, ktorý musí potom uvádzať.
- P 4.** Testovací program musí generovať správy a údaje, ktoré posielajú jednotlivé senzory periodicky, každých 10 sekúnd. – táto funkcionality sa dá vypnúť a znovu zapnúť. Odosielanie sa deje na pozadí a CLI je k dispozícii s ďalšími možnosťami.
- P 5.** Oba programy musia byť implementované v jazyku Python, dovolené knižnice sú JSON, SOCKET, CRC, ASYNCIO a THREADING (po dohode s cvičiacim aj iný jazyk).
- P 6.** Zabezpečenie integrity doručených dát na servery, ak sú dáta poškodené, server si ich znovu vypýta.

- P 7.** Možnosť vnieť chybu do prenášaných dát na overovanie a požiadavky č. 5 v testovacom programe.
- P 8.** Možnosť odoslať dáta zadané používateľom v testovacom programe. Nutné urobiť pre každý senzor.
- P 9.** Ak je nejaký senzor neaktívny, server upozorní klienta výpisom, periodicky každých 10 sekúnd, formát informácie DISCONNECTED špecifikovaný nižšie. Na základe periodickej správy, ktorou server overuje či je senzor aktívny, v prípade ak do časového limitu 15 sekúnd nepríde žiadna správa od senzora (to znamená, že server čaká na dáta od registrovaného senzora 15 sekúnd následne server pošle overovaciu správu na senzor ak do limitu nedostane správu tak sa aktivuje výpis DISCONNECTED ak na túto správu nedostane odpoveď. Ak dostane odpoveď, jeden krát sa vypíše informácia RECONNECTED).
- P 10.** V jednom čase môže byť pripojený iba jeden senzor z každého druhu.
- P 11.** Formát výpisov musí byť dodržaný. Špecifikovaný je v časti formát výpisov.
- P 12.** IP adresy a komunikačné porty možno nastaviť z rozhrania oboch programov.

Na základe požiadaviek bol pripravený zoznam odporúčaných správ na splnenie týchto požiadaviek, tento zoznam nemusí obsahovať všetky potrebné správy:

- Správa ktorou sa zariadenie registruje na server, odpoveďou bude správa obsahujúca token zariadenia. Tento token musí dané zariadenie uvádzať vždy, keď odosiela správu s dátami.
- Správa s dátami pre každé zariadenie.
- Správa, ktorou server odpovie, ak je token zariadenia neplatný.
- Správa, ktorou server potvrdí doručenie dát, ak táto správa došla na server.
- Správa, ktorou server overí, že daný senzor je aktívny a odpoveď senzora na takúto správu.
- Správa, ktorou si server vyžiada poslednú správu od senzora, ak mu nebola doručená.

Každá z týchto správ musí obsahovať:

- Typ zariadenia

- Časovú značku (timestamp) v sekundách (UNIX time)
- Príznak, ak má zariadenie slabú batériu
- Token zariadenia, ktorý získa inicializačnou správou

Formát informačných výpisov, ktoré treba dodržať:

Legenda: VEĽKÝ_NÁPIS – zachovať tak ako je, @ - doplniť príslušnú hodnotu

- Registrácia senzora:
„INFO: @sensor_id REGISTERED at @timestamp \n.”
- Prijatá dátová správa od senzoru (výpis na servery):
„@timestamp - @sensor_id \n
name_of_param_1: @DATA_1; name_of_param_2: @DATA_2;... ..
name_of_param_n: @DATA_n; \n “
- Prijatá dátová správa od senzoru ak je nízka hodnota napätia na batérii (výpis na servery):
„@timestamp - WARNING: LOW BATTERY @sensor_id \n
name_of_param_1: @DATA_1; name_of_param_2: @DATA_2;... ..
name_of_param_n: @DATA_n; \n “
- DISCONNECTED informácia (výpis na servery):
„WARNING: @sensor_id DISCONNECTED! \n “
- RECONNECTED informácia (výpis na servery):
„INFO: @sensor_id RECONNECTED! \n “
- Poškodené dáta (výpis na servery):
„INFO: @sensor_id CORRUPTED DATA at @timestamp. REQUESTING DATA “

Akceptačné testy (UAT):

Podmienky testovania: komunikácia musí prebiehať medzi dvomi fyzickými zariadeniami, musí byť sledovaná pomocou programu Wireshark.

UAT 1: Pripojenie zariadení na server a automatické generovanie správ.	
Vstupné podmienky	<i>Server:</i> nachádza sa v menu. <i>Testovací program:</i> nachádza sa v menu, nič sa nikam zatiaľ nepripojilo.

Scenár testu	<p>1) <i>Server</i>: v menu sa vyberie možnosť konfigurovať. Nastaví sa IP a PORT. Program sa vráti do menu a zvolí sa možnosť počúvať.</p> <p>2) <i>Testovací program</i>: v menu sa vyberie možnosť konfigurovať. Nastaví sa IP a PORT/PORTY. Program sa vráti do menu.</p> <p>3) <i>Testovací program</i>: zvolí sa možnosť automatického generovania dátových správ. Program sa vráti do menu.</p>
Výstupné podmienky	<i>Server</i> : Server vypíše najprv informáciu o zaregistrovaní jednotlivých senzorov. Potom v pravidelných intervaloch (cca. raz za 10 sekúnd) sa zobrazí informácia o priatej správe. (pozri formát informačných výpisov)
Výsledok testu	SPLNENÉ/AKCEPTOVATEĽNÉ/NESPLNENÉ

UAT 2: Odoslanie vlastnej správy používateľom.

Vstupné podmienky	<p><i>Server</i>: nachádza sa v menu. Zariadenia sú registrované.</p> <p><i>Testovací program</i>: nachádza sa v menu, pravidelne odosiela správy na server.</p>
Scenár testu	<p>1) <i>Testovací program</i>: v menu sa vyberie možnosť odoslať vlastnú správu. Program zobrazí názvy senzorov spolu s možnosťou vybrať tento senzor.</p> <p>2) <i>Testovací program</i>: používateľ vyberie jedno zo zariadení. Program ponúkne používateľovi možnosť oznámiť serveru slabú batériu, používateľ vyberie túto možnosť.</p> <p>3) <i>Testovací program</i>: Ak existuje parameter, ktorý používateľ ešte nezadal, zobrazí názov parametra a jeho možné hodnoty. Používateľ zadá hodnotu z rozsahu parametra. Ak sú zadane všetky hodnoty, prejde sa na krok 4), inak sa opakuje krok 3).</p> <p>4) <i>Testovací program</i>: Odošle správu z používateľom zadanými údajmi na server.</p>
Výstupné podmienky	<i>Server</i> : Server vypíše dátovú správu pre prípad nízkej hodnoty napätia v batérii. (pozri formát informačných výpisov)
Výsledok testu	SPLNENÉ/AKCEPTOVATEĽNÉ/NESPLNENÉ

UAT 3: Zavedenie chyby do odosielaných dát.

Vstupné podmienky	<p><i>Server</i>: nachádza sa v menu. Zariadenia sú registrované.</p> <p><i>Testovací program</i>: nachádza sa v menu, pravidelne odosiela správy na server.</p>
Scenár testu	<p>1) <i>Testovací program</i>: v menu sa vyberie možnosť zaviesť chybu pri odosielaní. Program zobrazí názvy senzorov spolu s možnosťou vybrať jeden senzor.</p> <p>2) <i>Testovací program</i>: používateľ vyberie jedno zo zariadení. Správa tohto zariadenia sa vygeneruje. Pri výpočte sa však kontrolná suma zámerne pokazí.</p> <p>3) <i>Server</i>: Vypíše správu o prijatí poškodených dát. Následne si vyžiada tieto dáta od zariadenia ešte raz.</p>

	3) <i>Testovací program</i> : Po prijatí požiadavky na odoslanie posledných dát, odošle tieto dáta bez zavedenia chyby.
Výstupné podmienky	<i>Server</i> : vypíše výpis pre poškodené dáta, a potom vypíše aj dátovú správu, hneď ako bola doručená správne. (pozri formát informačných výpisov). Nutné ukázať tento proces vo Wireshark-u.
Výsledok testu	SPLNENÉ/AKCEPTOVATEĽNÉ/NESPLNENÉ

UAT 4: Odpojenie zariadenia a mechanizmus kontroly aktívnosti zariadenia.

Vstupné podmienky	<i>Server</i> : nachádza sa v menu. Zariadenia sú registrované. <i>Testovací program</i> : nachádza sa v menu, pravidelne odosiela správy na server.
Scenár testu	1) <i>Testovací program</i> : v menu sa vyberie možnosť kontrola mechanizmu aktívnosti senzora. Program zobrazí názvy senzorov spolu s možnosťou vybrať tento senzor. 2) <i>Testovací program</i> : používateľ vyberie jedno zo zariadení. Správy tohto senzora prestanú byť odosielané. 3) <i>Server</i> : Po uplynutí limitu 15 sekúnd od poslednej prijatej správy daného zariadenia odošle overovaciu správu na testovací program. Ak do sekundy nedostane odpoveď, vypíše informáciu pre používateľa DISCONNECTED. Overovaciu správu potom posiela každých 5 sekúnd, najviac však 10 krát. 4) <i>Testovací program</i> : odpovie až po tom, ako sa ho server pokúsi kontaktovať tretíkrát a ďalej už štandardne pokračuje v automatickom generovaní a odosielaní správ. Program sa vráti do hlavného menu.
Výstupné podmienky	<i>Server</i> : vypíše informáciu DISCONNECTED dva krát, potom vypíše informáciu pre používateľa RECONNECTED. (pozri formát informačných výpisov). Nutné ukázať tento proces vo Wireshark-u.
Výsledok testu	SPLNENÉ/AKCEPTOVATEĽNÉ/NESPLNENÉ

UAT 5: Znovu odoslanie dát, ak server nepotvrdil prijatie správy.

Vstupné podmienky	<i>Server</i> : nachádza sa v menu. Zariadenia sú registrované. <i>Testovací program</i> : nachádza sa v menu, pravidelne odosiela správy na server.
Scenár testu	1) <i>Server</i> : v menu sa vyberie možnosť nepotvrdzuj dátovú správu. Program zobrazí názvy senzorov spolu s možnosťou vybrať tento senzor. 2) <i>Server</i> : používateľ vyberie jedno zo zariadení. Na dátové správy tohto zariadenia server trikrát neodpovie. 3) <i>Testovací program</i> : po uplynutí 1 sekundy odošle správu znova. Opakuje sa až kým neprijme odpoveď od serveru.
Výstupné podmienky	<i>Server</i> : vypíše dátovú správu. (pozri formát informačných výpisov), Nutné ukázať tento proces vo Wireshark-u.

Výsledok testu	SPLNENÉ/AKCEPTOVATEĽNÉ/NESPLNENÉ
----------------	----------------------------------

Proof of Concept (prvé odovzdanie):

Návrh a implementácia JSON protokolu – formát JSON v čitateľnej forme na zjednodušenie odladenia a overenie funkčnosti jednotlivých programov ako aj toho ako máte protokol navrhnutý na úrovni správ.

Hodnotenie:

Všetky položky označené červeným sú povinné aspoň v akceptovateľnej podobe. Zároveň však platí, že je nutné získať minimum 2,5 bodu v súčte. Je to pre to, aby sme mohli prebrať aj zadanie, ktoré má menšie nedostatky.

Čo sa hodnotí	Nesplnené (body)	Akceptovateľné	Splnené / Maximum
UAT 1	0 b	0,5 b	1 b
UAT 2	0 b	0,5 b	1 b
UAT 3-5	0 b	0,5 b	1 b
UAT 3-5	0 b	0,5 b	1 b
Dokumentácia	0 b	0,5 b	1 b
SPOLU		Min 2,5 b	5 b

Dokumentácia musí v tomto bode obsahovať aspoň tieto náležitosti. Zároveň musí byť odovzdaná ako PDF podľa vzoru, ktorý máte k dispozícii:

- Vývojové diagramy s opisom cyklu prijímania a cyklu odosielania v oboch programoch. Krátky slovný popis.
- Návrh jednotlivých správ v JSON aj s príkladom pre každú správu
- Sekvenčný diagram pre komunikáciu pre UAT1 a UAT2.
- Zoznam použitých knižníc.
- Návrh merania efektivity prenášaných dát (nezabudnúť počítať aj s nižšími vrstvami)

Výstupy:

Súbor zip pomenovaný xpriezvisko.zip (NIE RAR!) obsahujúci:

- Program servera pomenovaný server.xxx
- Testovací program pomenovaný tester.xxxx
- PDF súbor dokumentácie pomenovaný xpriezvisko.pdf

Optimalizácia na binárny protokol (Finálne preberanie):

Optimalizovaný protokol, ktorého funkčnosť je overená dvomi implementovanými programami. Cieľom je reprezentovať typ správy efektívne (napr. 2 bity pre 4 typy zariadení alebo napr. Timestamp zakódovať ako 32-bitové číslo, ak to stačí). Použite čo najmenší možný počet bitov pre každý parameter (napr. ak parameter má obmedzený rozsah a presnosť, zvolí sa vhodná reprezentácia), rovnako treba využívať aj príznaky (jednabitové parametre). Pripravte dissector pre Wireshark vášho protokolu v jazyku LUA.

Všetky položky označené červeným sú povinné aspoň v akceptovateľnej podobe. Zároveň však platí, že je nutné získať minimum 7 bodov v súčte. Je to pre to, aby sme mohli prebrať aj zadanie, ktoré má menšie nedostatky.

Čo sa hodnotí	Nesplnené	Akceptovateľné	Splnené / Maximum
UAT 1	0 b	1 b	1,5 b
UAT 2	0 b	1 b	1,5 b
UAT 3	0 b	1 b	1,5 b
UAT 4	0 b	1 b	1,5 b
UAT 5	0 b	1 b	1,5 b
LUA SCRIPT	0 b	0,5 b - 1 b – podľa rozsahu prípadov, ktoré skript pokrýva	
Meranie a porovnanie efektivity		0,5 b - 1 b – porovnanie vlastných protokolov, podporené malou tabuľkou alebo grafom.	1,5b - 2 b – navyše aj porovnanie efektivity vlastného protokolu oproti aspoň jednému existujúcemu, v praxi používaným protokolom podporené tabuľkou alebo grafom
Efektivita prenášania údajov	0 b - použité veľké dátové typy na malé čísla a pod.	0,5 b - 1,5 b – vhodné použitie dátových typov, ukázané vo Wiresharku	2 b – vysoko efektívny protokol, meniace sa dĺžky podľa potreby a pod.
Dokumentácia		0,5 b – obsahuje všetky náležitosti	1 b – dokumentácia, podľa ktorej by bolo možné dané programy a protokol naimplementovať bez dodatočných otázok.
Pochopenie problematiky v širšom kontexte	0 b – 0,5 b – zhodnotenie projektu, protokolov a samotných výsledkov meraní a porovnávaní.		
Doimplementácia		0,5 b	1 b
SPOLU		min 7 b	15 b

Dokumentácia **musí obsahovať všetky náležitosti**. Zároveň musí byť odovzdaná ako PDF podľa vzoru, ktorý máte k dispozícii:

- Vývojové diagramy s opisom cyklu prijímania a cyklu odosielania v oboch programoch. Krátky slovný popis.
- Návrh jednotlivých správ v JSON aj s príkladom pre každú správu.
- Štruktúry binárnych hlavičiek a polí.
- Sekvenčný diagram pre komunikáciu pre UAT1-5.
- Zoznam použitých knižníc.
- Príklad zapúzdrenia správy až na úroveň L2
- Meranie efektivity prenášaných dát (nezabudnúť počítať aj s nižšími vrstvami).
- Porovnanie efektivity prenášaných dát medzi vlastnými protokolmi
- Príloha A: (okolo 150 slov) – Zhodnotenie edukatívnej stránky projektu: Čo som sa vďaka tomuto zadaniu naučil. Ako my môžeme tieto znalosti pomôcť v ďalšom štúdiu a na trhu práce.

Výstupy:

Súbor .zip pomenovaný xpriezvisko.zip (NIE RAR!) obsahujúci:

- Program servera pomenovaný server.xxx
- Testovací program pomenovaný tester.xxxx
- PDF súbor dokumentácie pomenovaný xpriezvisko.pdf