Výber komunikačného protokolu

Z dlhodobého hľadiska považujeme za kľúčové vybrať správny aplikačný protokol na prenos dát zo senzorov. Existuje niekoľko zabehnutých populárnych IoT riešení, pri ktorých sa pokúsime špecifikovať ich prípady použitia, vplyv na spotrebu a jednoduchosť implementácie na klientskej a serverovej časti. Výstup z tejto kapitoly použijeme na výber komunikačného protokolu v našom riešení.

Zamieramie sa hlavne na protokoly, ktoré fungujú nad TCP/IP architektúre a bežne fungujú nad IEEE 802.11 (WiFI) fyzickou vrstvou.

Analýza

HTTP

Populárny protokol textovo založený, ktorý bol navrhnutý pre účely WWW v roku 1991. V klient-server architektúre ho označujeme ako request-response protokol. Klient vykonáva požiadavku na ktorú server odpovedá. Na identifikáciu prostriedku používa tzv. Universal Resource Identificator (URI). Na transportnej vrstve sa používa TCP protokol (od verzie HTTP3 UDP).

Na šifrovanie komunikácie používa SSL/TLS na prezentačnej vrstve (od verzie HTTP2 je nevyhnutné požiť šifrovanie). QoS nie je riešený na úrovni protokolu a treba ho implementovať aplikačne alebo na úrovni TCP (limitované). Veľkosť hlavičky je flexibilná a veľkosť tela je závisí od serverovej implementácie.

Z aplikačného uhla pohľadu ho považujeme za relatívne komplikovaný na implementáciu na vnorených zariadeniach. Protokol neudržiava trvalo otvorené TCP okno, preto nie je úplne vhodný na kontinuálne posielanie dát (od verzie HTTP2 vieme vyžívať jedno TCP spojenie na viac HTTP požiadaviek). Z testovania vychádza, že HTTP by malo zmysel používať iba za predpokladu, že zariadenie po zobudení odošle práve jednu požiadavku na jeden centrálny uzol a po jeho úspešnom odoslaní ide opäť na nejaký čas do režimu spánku.

Na kontinuálne posielanie dát považujeme za nevhodný kvôli neustálemu otváraniu TCP okna. Rovnako považuje nevýhodu podpory iba jedného prijímacieho uzla.

Teoreticky by sa dalo polemizovať o využití v IoT za predpokladu, že použijeme HTTP3 pretože operuje nad UDP. Nakoľko sa jedná stále o experimentálnu technológiu, ktorá nie je globálne podporovaná v štandardných HTTP serveroch túto možnosť zatiaľ vypúšťame.

MQTT

Jeden z najstarších protokolov z kategórie machine-to-machine (M2M), ktorý bol vyvinulo IBM v roku 1999. Operuje nad architektonickým vzorom client – broker, kde zariadenia publikujú alebo sa prihlasujú dátové topics. V praxi to vyzerá tak, že klient publikuje správu na MQTT message broker, ktorý následne túto správu pošle všetkým zariadeniam, ktoré sú prihlásené na daný komunikačný kanál (topic). Klient môže publikovať alebo sa prihlasovať na viac komunikačných kanálov naraz. Dáta sa prenášajú v binárnej forme a zabezpečujú sa pomocou SSL/TLS. Na transportnej vrstve sa štandardne používa TCP.

Protokol je veľmi populárny v IoT riešeniach a to hneď z niekoľkých praktických dôvodov:

- Je veľmi light-weight a jednoduchý na implementáciu na strane klienta
- Prenášajú sa len nevyhnutné dáta
- Veľkosť hlavičky sú iba 2 bajty

- TCP spojenie sa dlhodobo udržiava, nie je potrebné ho znova otvárať pri kontinuálnom posielaní dát
- Súčasťou špecifikácie je QoS

Pri kontinuálnom posielaní dát a udržiavanom otvorenom TCP okne má MQTT ďaleko lepšie výsledky ako HTTP.

V našom prípade, budeme skôr vyžadovať princíp fungovania v režime, že sa meracie zariadenie najprv zobudí zo spánku, pošle merania na broker a vráti sa opäť do režimu spánku. Pre tento prípad je energetická spotreba porovnateľná s HTTP. Na naše účely by nám ideálne vyhovovala komunikácia na báze UDP. Kvôli tomuto vznikla modifikácia s názvom MQTT-SN, ktorá funguje na báze UDP. Je optimalizovaná pre prípady, kedy potrebujeme odosielať správy zo zariadenia pripojené pomocou bezdrôtovej siete na message broker s dôrazom na minimalizáciu prenášaných dát a nízku energetickú náročnosť. Táto implementácia má zatiaľ ale momentálne minimálnu podporu u populárnych open-source message brokerov (najčastejšie formou modulu). Za posledné roky ale začína naberať popularitu.

CoAP

Najmladší protokol zo spomínaných, ktorý navrhnutý presne pre účely IoT. Radíme ho do kategórie light-weight M2M protokolov a dokáže operovať na obidvoch zatiaľ spomenutých architektonických vzoroch (client/server a client/broker). Bol navrhovaný tak aby vedel jednoducho fungovať v spolupráci s HTTP RESTful modelom formou jednoduchej proxy.Na identifikáciu prostriedku sa používa URI podobne ako v HTTP. Keď prídu dáta na konkrétnu URI, sú notifikovaný všetky zariadenia, ktoré sú prihlásené na odber podobne ako v MQTT.

Hlavička má veľkosť 4 bajty. Na transportnej vrstve používa UDP a dáta zabezpečuje pomocou DTLS alebo IPSec.

Medzi jeho hlavné výhody patrí podobnosť s RESTful návrhom (prvá veta na stránke projektu), jedná sa stále o mladý protokol, čo je cítiť na podpore existujúcich open-source riešení prípadne na jeho integrácií do zabehnutých nástrojov. Toto považujeme za problém ak chceme implementovať stabilnú a spoľahlivú službu. Jedná sa o veľmi zaujímavé riešenie, ktoré ale ešte potrebuje overenie časom.