Lukáš Bugaj

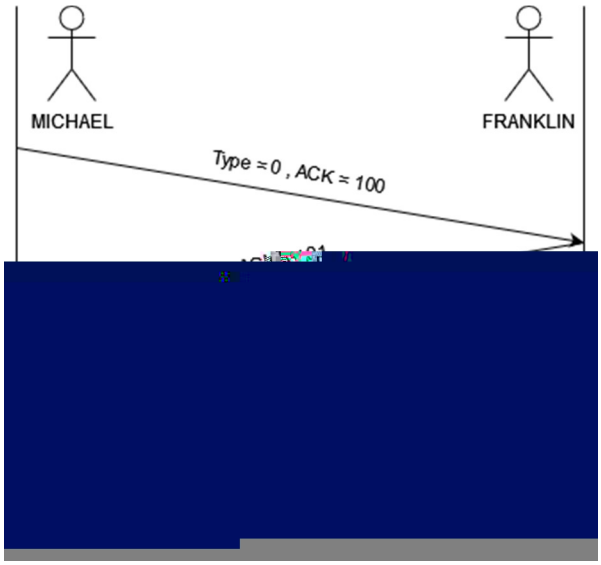
Teoretická časť: Návrh protokolu

V tejto časti dokumentácie Vám predstavím môj protokol, jeho hlavičku a vylepšenia UDP protokolu pre dosiahnutie vyššej spoľahlivosti.

Pole	Popis	Veľkosť (byte)	Poznámky
Type	Určuje typ správy.	1	0 = ackreq, 1 = ackresp, 2 = ackrdy ...
ACK	Číslo pre overenie odpovede na ack.	1	Opačný uzol odpovedá vždy + 1
Fragment flag	Určuje či je správa fragmentovaná.	1	0 = nie je, 1 = je a ešte sú fragmenty, 2 = je a je to posledný fragment
Fragment offset	Určuje poradie fragmentu	4	
Payload	Určuje veľkosť datagramu	2	
Checksum	Kontrolný súčet	2	

Na nadviazanie spojenia používam tzv. 3 – way handshake, ktorý má nasledovné fázy:

1. Fáza „ackreq“. Prvý dostupný uzol pošle v poli ACK číslo v rozsahu 0 – 253.
2. Fáza „ackresp“. Uzol na opačnej strane odpovie zmenou typu správy z 0 na 1, a k hodnote v poli ACK pripočíta 1.
3. Fáza „ackrdy“. Uzol ktorý posielal správu v prvej fáze, zmení typ správy z 1 na 2 a znovu pripočíta k hodnote v poli ACK hodnotu 1.



Ak dáta v správe presiahnu limit protokolu, budú musieť byť fragmentované. To znamená že budú musieť byť rozdelené na menšie časti, ktoré budú označené: 0, 1, 2.... Taktiež bude mať každý fragment nastavený flag v poli fragment flag podľa toho či, ako to je napísané v zadaní, správa nieje fragmentovaná (t.j. flag = 0), je fragmentovaná a má uzol na opačnej strane očakávať ďalšie fragmenty (t.j. flag = 1), alebo správa je fragmentovaná avšak už niesú žiadne ďalšie fragmenty (t.j. flag = 2). Podľa môjho názoru sa týmto spôsobom vyhnem prenosu nejakých zbytočných dát v hlavičke ako fragment size, čo zvýši kapacitu pre dáta.

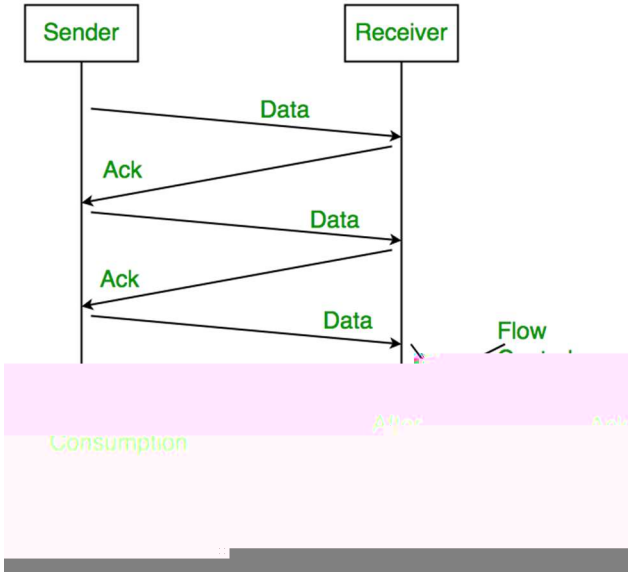
Na overenie integrity prenesených správ budem používať CRC 16. CRC-16 je 16-bitový (2 bajty) kontrolný súčet, ktorý sa vypočíta na základe dát, ktoré posielame. Tento kontrolný súčet sa pridá k správe, a keď prijímame správu, opäť si ho vypočítame a porovnáme s tým, čo bolo prijaté. Ak sa kontrolné súčty zhodujú, správa je nepoškodená.

Povedzme, že chceme vypočítať **CRC-16** pre jednoduchú správu, ktorá obsahuje jedno číslo, napríklad 0x12 (hexadecimálne, čo je v desiatkovej sústave 18).

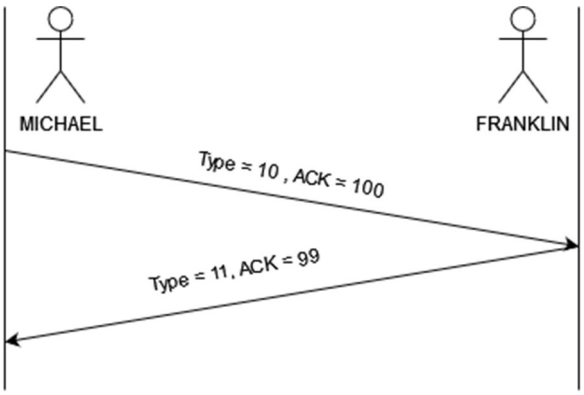
1. **Inicializujeme CRC**: Začiatočná hodnota bude 0xFFFF (1111 1111 1111 1111).
2. **XOR prvého bajtu**: Vykonáme XOR medzi hodnotou CRC a dátami.
    - Náš dátový bajt je 0x12 (v binárnej forme 0001 0010).
    - XOR medzi 0xFFFF a 0x12 bude:
        - 1111 1111 1111 1111 XOR 0000 0000 0001 0010 = 1111 1111 1110 1101 (hexadecimálne: 0xFFED).
3. **Posúvame bity a aplikujeme polynóm**: Keď máme výsledok po XOR, posunieme hodnotu doprava a aplikujeme polynóm 0x8005 na základe najnižšieho bitu. Tento krok opakujeme pre všetky bity. Ak je najnižší bit 1, vykonáme XOR s polynómom, ak je 0, posúvame ďalej doprava bez XOR.
4. **Konečný výsledok**: Na konci týchto operácií získame výsledný 16-bitový kontrolný súčet, ktorý by mohol vyzerať napríklad ako 0xA2B3.

Túto metódu som si zvolil ako metódu na zabezpečenie doručovania správ.

V **Stop & Wait** odosielateľ vyšle jeden paket a čaká, kým dostane ACK (potvrdenie) od prijímateľa predtým, než pošle ďalší paket. Tento proces je jednoduchý na implementáciu, pretože spracováva len jeden paket naraz a nemusí sledovať žiadne okno alebo spravovať viacero paketov naraz, ako je to pri zložitejších metódach ako **Go Back-N** alebo **Selective Repeat**.



Kontrolu aktívnosti spojenia som sa rozhodol robiť podobne ako pri otvorení spojenia. Obidva uzly budú mať časovač nastavený na nejaký timeout, ktorý bude plynúť od prijatia poslednej správy. Následne keď tento timeout uplynie, uzol vyšle správu „PING“, tak, že typ správy nastaví na 10 a pošle v poli ACK číslo. Opačný uzol odpovedá odoslaním správy „PONG“, t.j nastavením typu správy na 11 a znížením hodnoty ACK o 1.



V tejto dokumentácii som priblížil návrh môjho protokolu a jednotlivé kroky k jeho vytvoreniu, tak, aby zabezpečoval spoľahlivý prenos dát nad protokolom UDP. Protokol rieši základné problémy s prenosom dát ako je nadviazanie spojenia, fragmentácia, kontrola integrity správ alebo udržiavanie spojenia.