

PKS – Zadanie 2: Komunikácia s využitím UDP protokolu– návrh riešenia

Štruktúra hlavičky navrhnutého protokolu

Hlavička navrhnutého protokolu je dĺžky 5 bytov. V prvom byte sa nachádza typ správy a informácie o fragmentácii správy. V ďalších 2 bytoch je ID správy, čo je v podstate poradie správy v komunikácii. To sa využíva pri odosielaní ACK packetov a pri kontrole správnosti prenosu správ. Ďalšie 2 byty obsahujú kontrolnú sumu, ktorá slúži na detekciu chyby pri prenose správy.

0,5	1	2	3	4
Type	Fragment	Identifier	Checksum ...	
... checksum		Data ...		

Type:

- 0- ACK
- 1- Message
- 2- File transfer – file name
- 3- File transfer – file content
- 4- Remain connection
- 5- Error

Fragment:

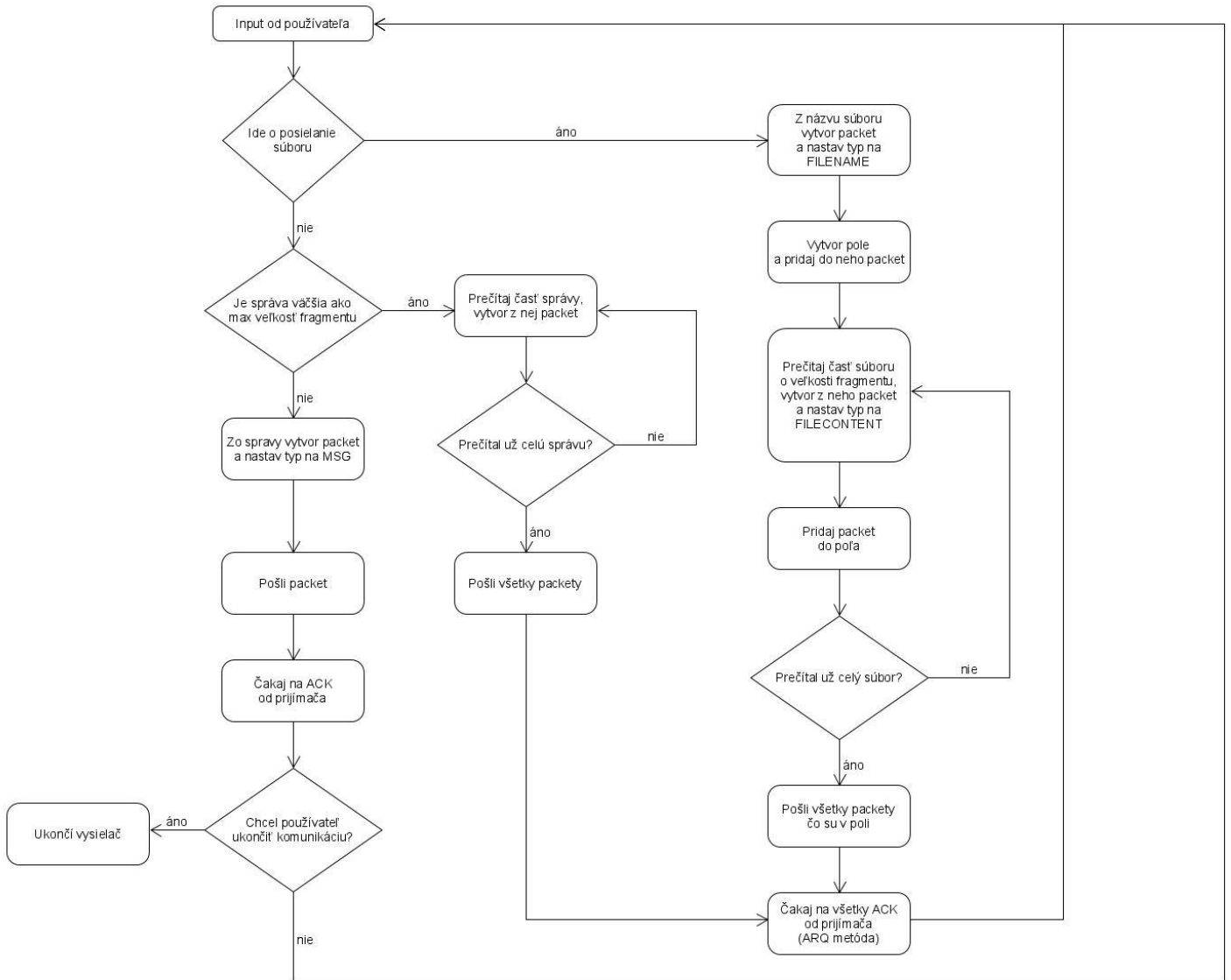
- 1- no fragment
- 2- first fragment
- 3- more fragments
- 4- last fragment

Trieda Protocol

V tomto programe je implementovaná vlastná trieda Protocol, ktorá reprezentuje packet, ktorý je odoslaný / prijatý v komunikácii. Je to vrstva abstrakcie nad surovými dátami, ktoré sa prenášajú v komunikácii, ktorá slúži na jednoduchšiu prácu s týmito dátami a zaručuje väčšiu prehľadnosť kódu. Po načítaní dát, ktoré chce vysielateľ poslať, sa vytvorí objekt tejto triedy, čo môžeme chápať že sa vytvorí korektná hlavička, ku ktorej sa pripoja načítané dáta a to sa spolu pošle. Prijímač, po prijatí týchto dát opäť vytvorí objekt tejto triedy a následne tak vie osobitne pracovať s hlavičkou a samotnými dátami, ktoré chcel vysielateľ poslať.

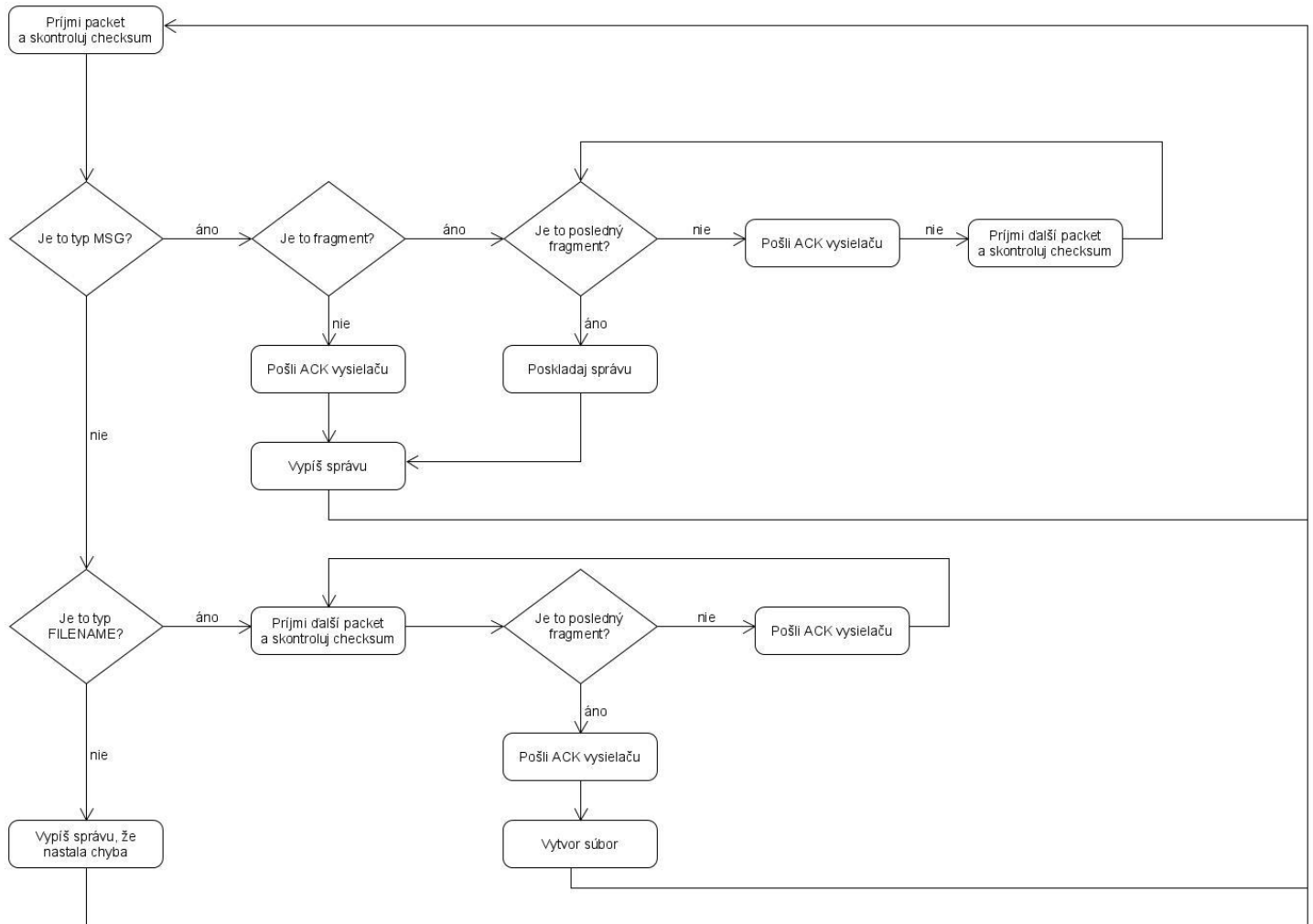
Diagramy spracovania komunikácie

Vysielač



Vytváranie packetov môžeme chápať ako vytvorenie objektu Protocol, nastavenie dát objektu (správa zadaná používateľom, alebo časť súboru), nastavenie fragment flagov, nastavenie identifier, vypočítanie a nastavenie checksum.

Prijímač



Ak pri kontrole checksum program zistí, že prenesené dáta boli poškodené, tak vysielacu pošle správu typu error a prejde do stavu, kedy čaká na prijatie správy

Kontrolná suma

V tomto programe sa na overenie integrity a správnosti posielaných správ používa CRC (cyclic redundancy check). Na strane vysieláča sa po vytvorení objektu Protocol vypočíta kontrolná suma, ktorá sa uloží do poľa Checksum. Na strane prijímača sa po prijatí správy opäť spraví rovnaký výpočet a ak sa výsledok rovná s kontrolnou sumou prijatou zo správy, tak môžeme správu považovať za korektnú.

Výpočet kontrolnej sumy prebieha tak, že vysieláč a prijímač majú určenú rovnakú masku (v tomto prípade **0x176a7** – zdroj: <https://users.ece.cmu.edu/~koopman/crc/>), ktorou sa dáta (reprezentované ako číslo v hexadecimálnom tvare) vydedia (použitie operácie XOR). Zvyšok po delení sa bude brať ako kontrolná suma.

ARQ metóda

Metóda na zabezpečenie spoľahlivého prenosu dát v tomto programe funguje na princípe podobnom Selective Repeat ARQ. Na strane vysielача sa nastaví sliding window o veľkosti N a toľko packetov sa pošle. Vysielач potom čaká na N ACK packetov od prijímača. Akonáhle vysielач prijme ACK, tak odošle ďalší packet a opäť čaká na N ACK packetov od prijímača.

Vysielачom odoslané packety sa postupne pridávajú do queue a postupne ako prichádzajú jednotlivé packety ACK, tak sa z queue odoberajú. To znamená, že v určitý moment je v danej queue maximálne N odoslaných packetov.

Ak nastane situácia, že na určitý odoslaný packet nepríde ACK a na packet, ktorý bol odoslaný až po ňom príde, tak sa tento packet odoberie z queue, pošle sa znova a opäť sa pridá do queue. To znamená, že odoslaných packetov a prijatých ACK bude $n+m$, kde n je počet packetov, ktoré sa majú odoslať a m je počet packetov, ktoré boli odoslané znova.

Udržanie spojenia

Ak sa zo strany vysielача 5 sekúnd nepošle žiadna správa tak, vysielач prejde do stavu na udržanie spojenia. To znamená, že na druhom vlákne vysielач začne posilať správy s typom Remain connection každých 5 sekúnd a čaká na odpoveď od prijímača. Ak nedostane odpoveď na 3 po sebe poslané správy Remain connection, tak môžeme usúdiť, že spojenie bolo prerušené a preto vysielач vypíše správu o prerušenej komunikácii a dá používateľovi možnosť ukončiť spojenie. Ak vysielач dostane odpoveď od prijímača, tak posila tieto správy naďalej, až pokým sa používateľ nerozhodne poslať správu, potom vysielач prejde naspäť do stavu na posielanie správ.