Návrh

Počítačové a Komunikačné siete

Communication over UDP

Jakub Martinák

Faculty of Informatics and Information Technology, Slovak Technical University

Obsah

1 Návrh 2

2 Checksum metóda 3

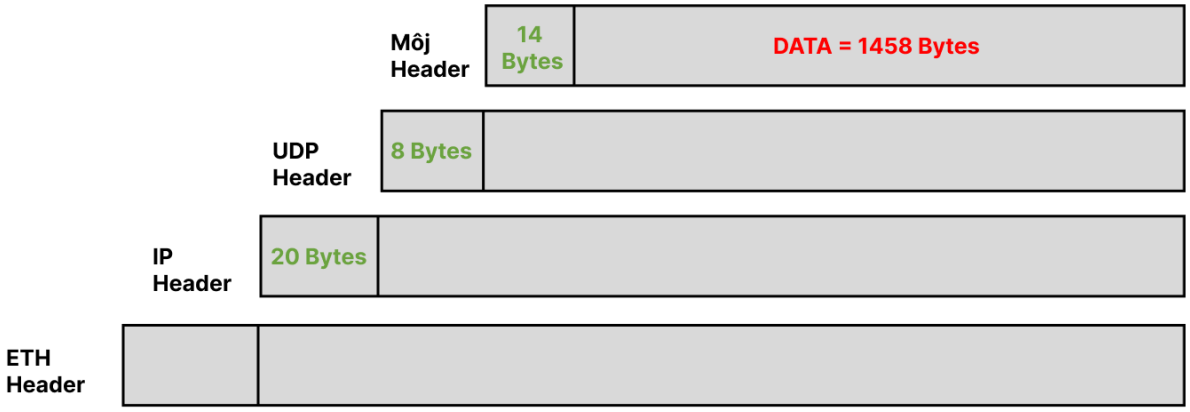
3 ARQ metóda 3

4 Metóda pre udržanie spojenia 3

5 Diagram 4

1 Návrh

Navrhnutý protokol bude vnorený do protokolu UDP ako je znázornené na obrázku 1. Maximálna veľkosť packetu je 1500B(ETH) - 20B(IP Header) - 8B(UDP Header) - 14B(Môj Header) = **1458 B**



Obr. 1: Návrh packetu

Typ	Sekvencia	Identifikátor súboru	Checksum (CRC16)	Payload	Flags	Data
1B	4B	4B	2B	2B	1B	-

Tabuľka 1: Môj header

Časť	Význam	Možnosti
Typ	Rola packetu	0x01(data), 0x02(ACK), 0x03(Controla), atď
Sekvencia	Na kontorlu správneho poradia Packetov	-
Identifikátor súboru	Unikátny identifikátor súbor	Pomoc pri fragmentovaní
Checksum(CRC16)	Pre kontrolu errorov	-
Velkosť Payloadu	Velkosť paylodu v bytoch	Dĺžka packetu indikujúca koľko dát packet obsahuje
Vlajky	Kontrolne Informacie	Vlajka pre znovuodoslanie, atď

Tabuľka 2: Typy

Komunikátor bude implementovaný v jazyku Python. Pri spustení programu bude mať užívateľ možnosť si vybrať či sa spustí klient alebo server. Pri výbere klienta sa bude musieť zadávať IP adresa servera a port, na ktorom počúva, veľkosť fragmentu a správu / súbor. Pri spustení programu ako server, užívateľ bude musieť zadať port, na ktorom má server počúvať a bude čakať na inicializačný packet od klienta. Keď klient zadá všetky potrebné informácie, vyšle sa inicializačný packet a počká na odpoveď ACK od serveru, a následne sa začnú posielať dáta. Ak je veľkosť packetu väčšia ako nastavená veľkosť na posielanie, súbor/správa sa rozdelí na fragmenty a bude sa posielať jednotlivo. Po každom odoslanom fragmente server overí pomocou checksumu packet, ak sedí pošle sa ACK a čaká na ďalší odoslaný packet.

Server bude fungovať na zvolenej metóde ARQ Stop and Wait, kde bude klient čakať s posielaním ďalšieho packetu až kým nedostane od serveru ACK o kompletnom prijatí packetu, ak by klient nedostal odpoveď do 6 sekúnd, packet sa pošle znovu. Po dokončení posielania, server čaká na ďalší signál od klienta.

2 Checksum metóda

Checksum je využívaný na overenie správy, či bola počas prenášania poškodená. V projekte bude použitý checksum CRC16(Cyclic Redundancy Check) z knižnice binascii (Python). Polynomická reprezentácia CRC16:

$$\mathbf{x^{16} + x^{15} + x^2 + 1}$$

Binárna reprezentácia:

1000000000000101

Hlavná operácia v CRC je binárne delenie, je to ale čisto založené na XOR operáciach, bez odčítania. Dáta, ktoré chceme skontrolovať sú považnované za veľké binárne číslo a to je vydelené polynómom CRC.

Pri inicializácií 16 núl je pridelených na koniec dát pred začiatkom delenia. To sa deje kvôli tomu že dĺžka CRC checksumu je 16 bitov. A zvyšok po delení CRC16 je 16-bitové číslo.

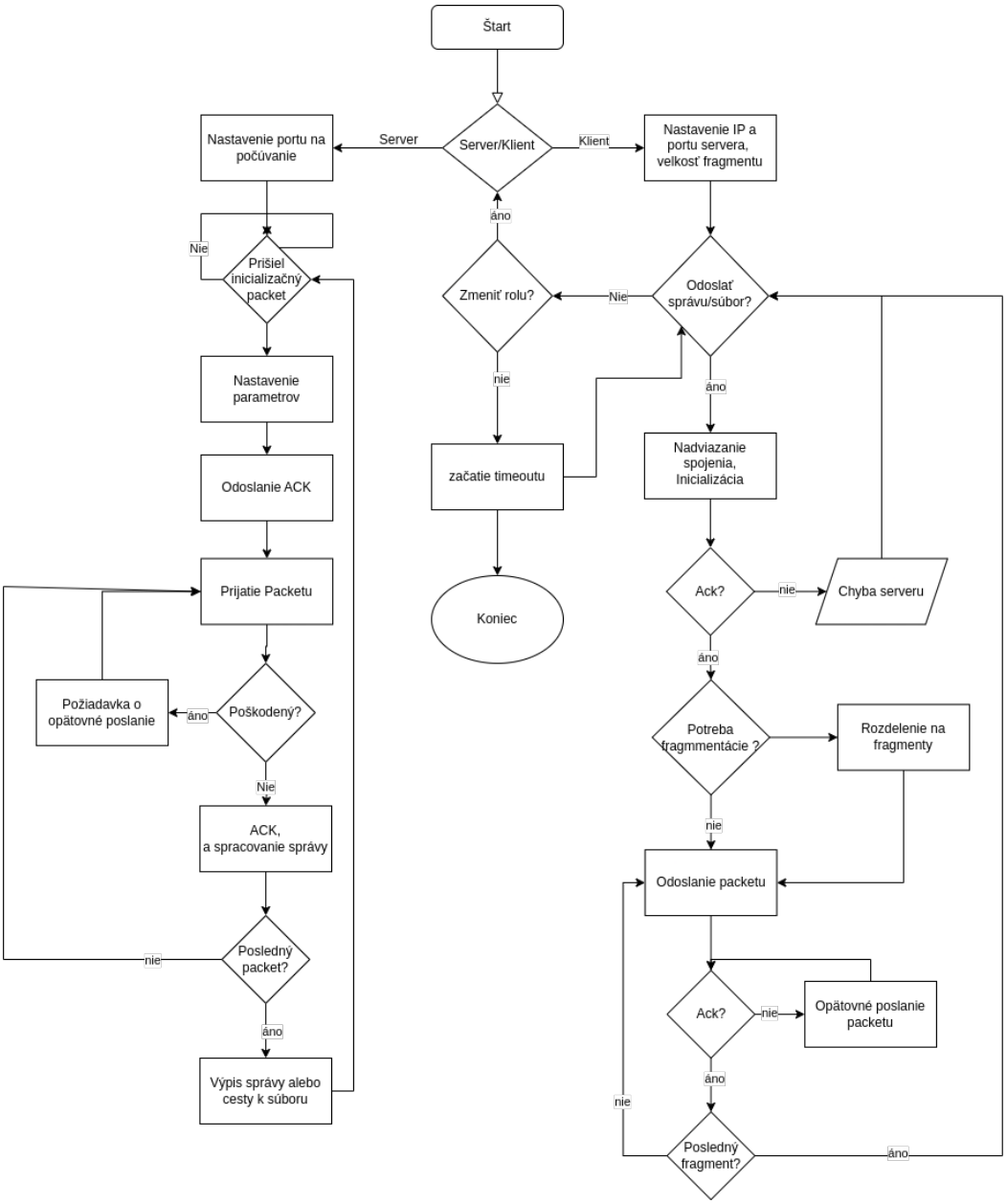
3 ARQ metóda

V projekte bude použitá ARQ metóda ***Stop and Wait***(Automatic Repeat reQuest) .
Táto metóda je jednoduchá ale účinná, zabezpečuje, že dáta prídu v poradí akom boli poslané. Funguje na princípe, že každý packet, ktorý bol odoslaný od klienta na server, klient čaka na spätné ACK aby mohol poslať ďalší. Ak klient do určitého časového limitu nedostane ACK pokúsi sa Packet poslať znovu, čo môže v nejakých prípadoch spôsobiť duplicitu. Táto metóda nie je efektívna na posielanie packetov o veľkej veľkosti alebo na nestabilných sieťach, kde kvôli čakaniu na ACK sa veľmi znižuje jeho efektivta.

4 Metóda pre udržanie spojenia

Spojenie bude udržiavané pomocou užívateľského rozhrania, kde si užívateľ bude môcť vybrať či chce zmeniť svoju rolu z klienta na server a vice versa. Po doknčení odosielania, bude toto menu opätovne zobrazené na na klientovej strane. Ak po skončení odosielania nedostane server po dobo 15 sekúnd packet od klienta na vymenenie ról alebo opätovné posielanie tak sa spojenie terminuje.

5 Diagram



Obr. 2: Flow chart Server/Klient