

Slovenská technická univerzita

Fakulta informatiky a informačných technológií
Ilkovičova 3, 812 19 Bratislava

Počítačové a komunikačné siete

Komunikácia s využitím UDP protokolu

Tomáš Vlha

Cvičiaci: Ing. Rastislav Bencel
Študijný odbor: Informatika Ročník: 2. Bc
Akademický rok: 2016/2017

Obsah

1. Znenie zadania.....	3
2. Analýza	4
2.1 Enkapsulácia hlavičky	4
2.2 Hlavné vlastnosti UDP protokolu	5
2.3 Chyby v prenose a znovu vyžiadané rámce (Checksum)	5
2.4 Pakety spojené s kontrolou spojenia	5
3. Špecifikácia požiadaviek	6
3.1 Špecifikácia programu	6
3.2 Požiadavky programu	6
4. Návrh riešenia	7
4.1 Návrh vlastného protokolu.....	7
4.2 Návrh hlavičky protokolu	7
4.3 Vývojový diagram	8
4.4 Návrh vlastného GUI	9
5. Zhodnotenie	10
6. Bibliografia.....	10

1. Znenie zadania

Nad protokolom UDP (User Datagram Protocol) transportnej vrstvy sieťového modelu TCP/IP navrhnete a implementujete program, ktorý umožní komunikáciu dvoch účastníkov v sieti Ethernet, teda prenos správ ľubovoľnej dĺžky medzi počítačmi (uzlami).

Program bude pozostávať z dvoch častí – vysielacej a prijímacej. Vysielací uzol pošle správu inému uzlu v sieti. Predpokladá sa, že v sieti dochádza k stratám dát. Vysielajúca strana rozloží správu na menšie časti - fragmenty, ktoré samostatne pošle. Správa sa fragmentuje iba v prípade, ak je dlhšia ako max. veľkosť fragmentu. Veľkosť fragmentu musí mať používateľ možnosť nastaviť aj menší ako je max. prístupný pre trasportnú vrstvu. Po prijatí správy na cieľovom uzle tento správu zobrazí. Ak je správa poslaná ako postupnosť fragmentov, najprv tieto fragmenty spojí a zobrazí pôvodnú správu. Komunikátor musí vedieť usporiadať správy do správneho poradia, musí obsahovať kontrolu proti chybám pri komunikácii a znovu vyžiadanie rámca, vrátane pozitívneho/negatívneho potvrdenia. Pri nečinnosti komunikátor automaticky odošle paket pre udržanie spojenia. Odporúčame riešiť cez vlastne definované signalizačné správy.

Program musí mať nasledovné vlastnosti:

1. Pri posielaní správy musí používateľovi umožniť určiť cieľovú stanicu.
2. Používateľ musí mať možnosť zvoliť si max. veľkosť fragmentu.
3. Obe komunikujúce strany musia byť schopné zobrazovať:
 - poslanú resp. prijatú správu,
 - počet fragmentov správy.
4. Možnosť odoslať chybný rámec
5. Možnosť odoslať dáta zo súboru a v tom prípade ich uložiť na prijímacej strane do súboru

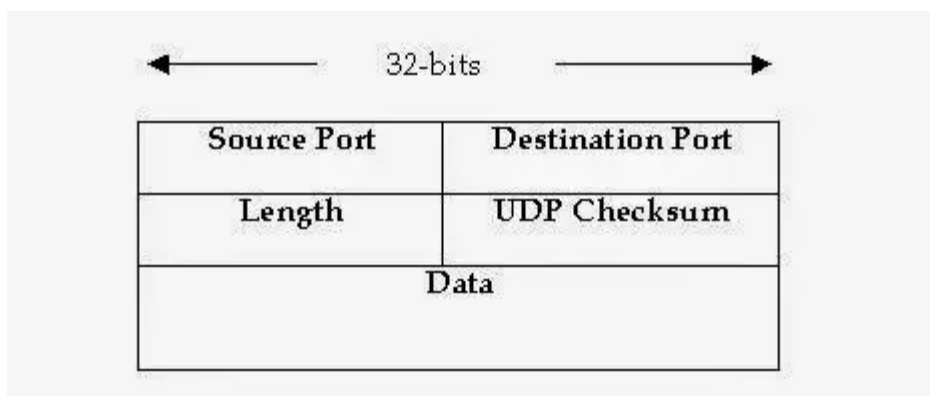
Zadanie sa odovzdáva:

1. Návrh riešenia
2. Predvedenie riešenia v súlade s prezentovaným návrhom

Program musí byť organizovaný tak, aby oba komunikujúce uzly mohli byť (nie súčasne) vysielateľom a prijímačom sprá

2. Analýza

Hlavnou úlohou v tomto zadaní je vytvoriť jednoduchý program na komunikáciu, ktorý bude figurovať nad vlastným UDP protokolom s sieťou Ethernet. UDP protokol je považovaný za jeden z najjednoduchších protokolov, vzhľadom na jeho obsluhu z pohľadu používateľa. Pakety UDP posielame cez port prvého komunikujúceho programu na port druhého v jednosmernom komunikujúcom kanáli. Pri UDP protokole ide spravidla o prenos jednotlivých paketov, a nie o prenos dátového prúdu.



Obr. 1 – UDP protokol

V UDP protokole figurujú oproti sebe dve strany. 1. prijímač alebo server a 2. vysielač alebo klient. Jednotlivé prijímače môžu tak isto zastávať funkciu vysielačov a naopak, táto funkcia sa nazýva poloduplexnosť. Bohužiaľ táto funkcia má aj jednu nevýhodu, prijímač nemôže figurovať v tom istom čase ako vysielač a naopak. Na vyriešenie tohto problému môžeme použiť napríklad Multi-threading s novo otvoreným socketom (táto verzia nie je potrebná na splnenie úlohy). V grafickom rozhraní budeme mať teda možnosť či budeme figurovať ako server alebo ako klient.

2.1 Enkapsulácia hlavičky

V počítačových sieťach, sa využíva metóda zvaná enkapsulácia hlavičky. Počas enkapsulácie každá vrstva postaví tzv. PDU (Protocol Data Unit) pridaním hlavičky (a niekedy aj chvostu), ktoré obsahujú informácie z PDU vrstvy nad danou vrstvou. Z toho vyplýva že úlohou a výsledkom enkapsulácie je že každá vrstva posadená na nižšej úrovni poskytuje služby vrstvám alebo vrstve nachádzajúcej sa nad danou vrstvou (OSI model /TCP model).

2.2 Hlavné vlastnosti UDP protokolu

The User Datagram Protocol (UDP) je protokol založený na báze transportnej vrstvy definovaný na využitie spolu s IP protokolom. Je stanovený RFC 768 napísaný Johnom Postelom.

UDP poskytuje minimálny no nespoľahlivý a jednoduchý na obsluhu, transport informácií z nižších vrstiev na tie vyššie vrstvy. V porovnaní s ostatnými transportnými protokolmi je UDP jedinečné v tom že nezriaďuje end-to-end pripojenie medzi komunikujúcimi systémami.

Hlavička UDP protokolu sa skladá z 5 dôležitých pamäťových blokov

-Zdrojový port (16bit)

-Cieľový port (16bit)

-UDP dĺžku

-UDP Checksum

2.3 Chyby v prenose a znovu vyžiadané rámca (Checksum)

Jedným z druhov checksumu je CRC (z angl. cyclic redundancy check). Na základe jednotlivých bitov sa vypočítava zabezpečovací údaj. Ten sa na konci celého bloku porovná so zabezpečovacím údajom, ktorý podľa rovnakých pravidiel vypočítal odosielateľ a pripojil k prenášanému bloku dát. Ak sa tieto dva údaje zhodujú, dá sa prenesený blok s vysokou pravdepodobnosťou predpokladať za správny.

V sieťovej komunikácii sa najčastejšie používa polynóm $x^{16} + x^{12} + x^5 + 1$, doporučený organizáciou CCITT. Tento polynóm použijem aj pri svojej implementácii.

2.4 Packety spojené s kontrolou spojenia

Počas komunikácie medzi serverom a klientom je potrebné sa ubezpečiť, či je spojenie stále funkčné. Na to slúžia packety, ktoré sa vysielajú v určitých časových intervaloch z klienta na server. Server musí následne správne zareagovať aby sa spojenie neprerušilo a zostalo naďalej aktívne, ak server nezareaguje alebo zareaguje nekorektne, spojenie medzi klientom a serverom zlyháva. Tieto packety sú tak isto známe aj pod menom „keep-alive packets“.

3. Špecifikácia požiadaviek

Pri implementácii je používam programovací jazyk Java Script v Node JS s použitím Dgram, Electron, Buffer-crc32 knižníc ktoré umožňujú prácu s UDP soketmi a paketmi. Potenciál tohto jazyka a vývojového prostredia sa prejaví aj pri následnej implementácie grafického rozhrania.,,

3.1 Špecifikácia programu

Používateľovi sa umožní určiť si cieľovú stanicu.

Program obsahuje user-friendly GUI

Program dokáže nadviazať spojenie so serverom a klientom, následne ho udržiavať „živé“

Program dokáže pomocou CRC zistiť či ide o chybný packet

Program dokáže opätovne si vyžiadať chybný packet

Program dokáže správne enkapsulovať prijaté packety

Používateľ si v programe bude môcť zvoliť veľkosť maximálneho a minimálneho fragmentu.

Klientovi ale aj serveru bude umožnené zobrazit' poslednú prijatú správu prípadne počet fragmentov danej správy.

Používateľovi bude umožnené odoslať chybný rámec.

Používateľ bude môcť odoslať dáta zo súboru tak aby sa uložili na prijímacej strane do súboru.

3.2 Požiadavky programu

Počítač, ktorý obsahuje funkčnú grafickú kartu.

4. Návrh riešenia

4.1 Návrh vlastného protokolu

Pre začiatok by sme si mali vyriešiť problém s hlavičkou, a jej návrhom a určiť si pravidlá podľa ktorých bude náš UDP protokol transportnej vrstvy sieťového modelu TCP/IP fungovať a umožní nám komunikáciu medzi dvoma zariadeniami. Náš protokol by mal nadviazať stabilné spojenie po nadviazaní pripojenia medzi serverom a klientom, tak dlho pokiaľ ho sami neukončíme.

4.2 Návrh hlavičky protokolu

Naša hlavička bude obsahovať dokopy 11B, ktoré sú následne rozdelené do týchto kategórií :

- CRC kontrola (2B)
- Type of packet (1B)
- Tag (2B)
- Number of Fragments (2B)
- Size (4B)

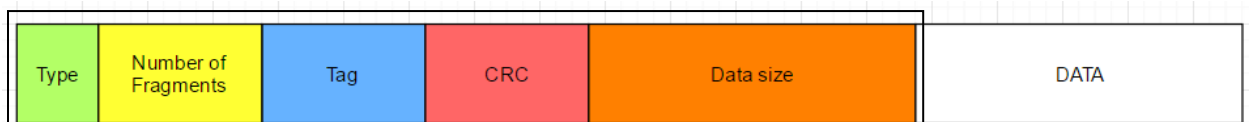
CRC kontrola: slúži na určenie správnosti packetov. Ak nájde chybu vyžiada si opätovné odoslanie.

Type of packet: Označuje typ packetu čo prichádza (Regular/Keep Alive).

Tag: Označenie fragmentov na opätovné zloženie dát.

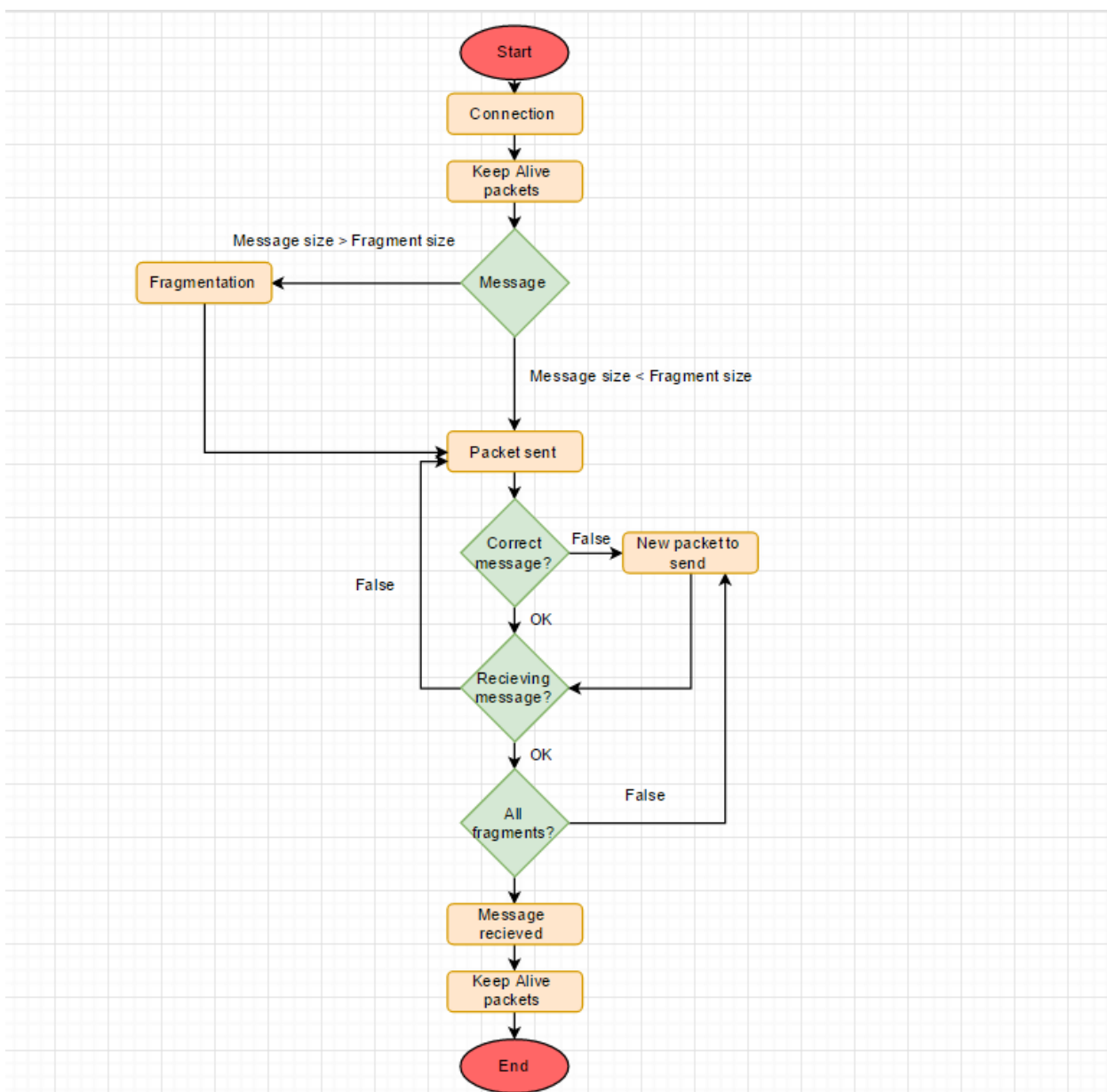
Number of Fragments: Počet odoslaných fragmentov

Size: Obsahuje veľkosť odosielaných dát, v prípade že fragment je menší vyplníme ich aby sa splnila dĺžka fragmentu



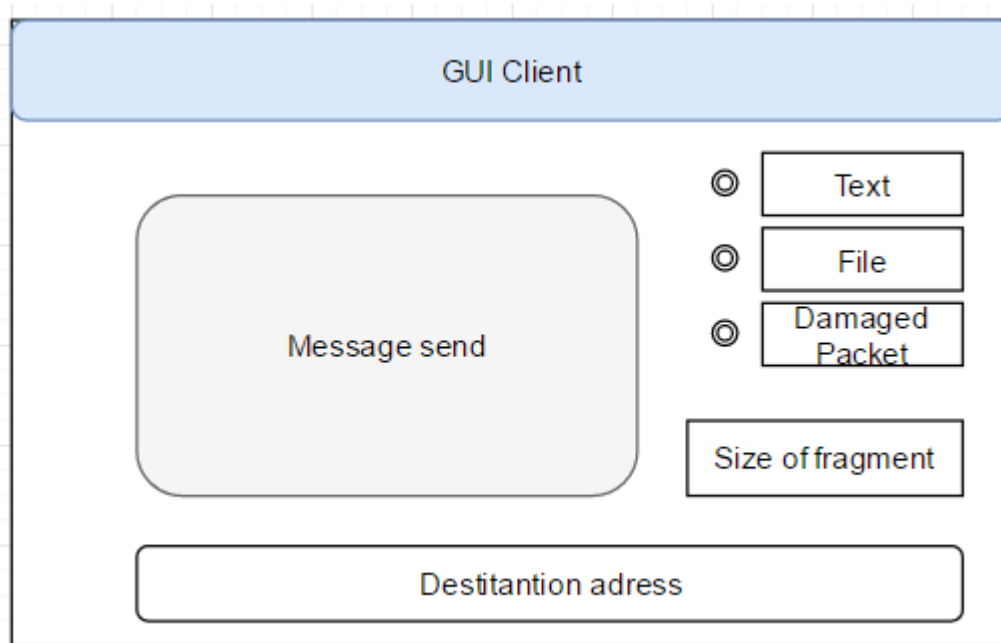
Obr. 2 – Custom header of UDP

4.3 Vývojový diagram

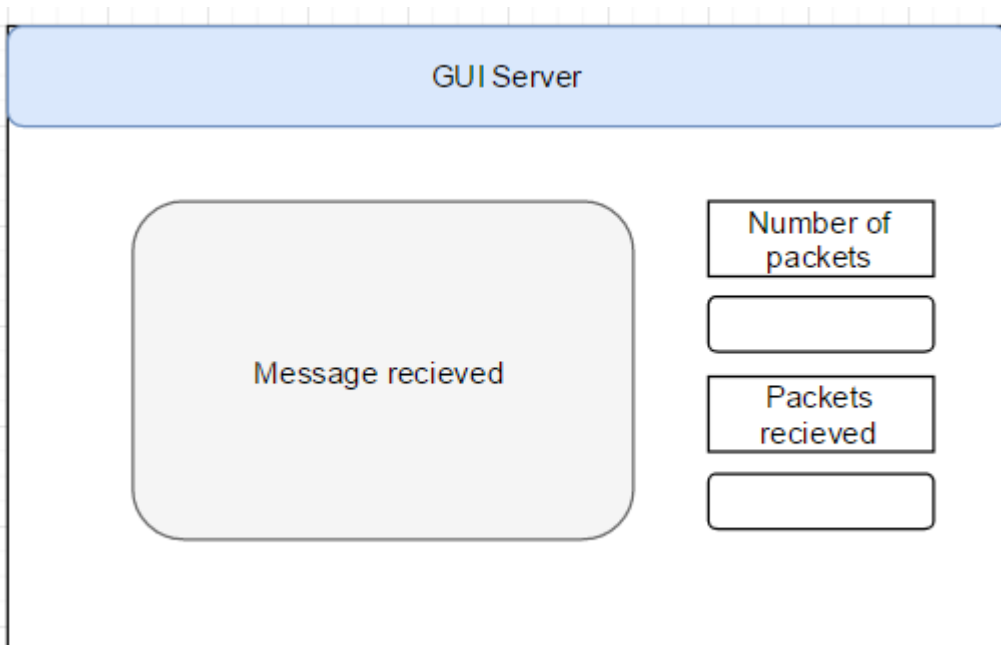


Obr. 3 – UDP process diagram

4.4 Návrh vlastného GUI



Obr. 4 – GUI Client



Obr. 2 – GUI Server

5. Zhodnotenie

6. Bibliografia

[KOTULIAK, 2015] KOTULIAK, I.: Prednášky k predmetu Počítačové a komunikačné siete. Transportná až aplikačná vrstva, STU Fakulta informatiky a informačných technológií, 2015. 70 s.

[The User Datagram Protocol (UDP)] : <http://www.erg.abdn.ac.uk> Date: 16.10. 2016