Semestrálne zadanie: Komunikácia s využitím UDP protokolu. Kontrolný bod.

Ruslan Gainetdinov, ID: 127150

October 21, 2024

Abstract

Cieľom tohto projektu je vytvoriť P2P aplikáciu, ktorá používa vlastný protokol postavený na transportnom protokole UDP pre výmenu textových správ a súborov. Protokol bude podporovať spoľahlivý prenos dát a tiež mechanizmy kontroly aktivity a integrity správ.

UDP sám o sebe nezabezpečuje spoľahlivosť prenosu dát, a preto vytvoríme dodatočné mechanizmy pre riešenie strát dát a kontrolu stavu spojenia. Nižšie je popísaný navrhovaný dizajn protokolu.

1 Štruktúra hlavičiek protokolu

Na prenos dát vytvoríme štruktúru hlavičiek protokolu, ktorá bude obsahovať dôležité informácie pre spracovanie paketov. Štruktúra hlavičky obsahuje nasledujúce polia:

Pole	**Popis**	**Dĺžka (bajtov)**
Typ správy	Určuje typ správy (napr. požiadavka, odpoveď, chyba)	1
Identifikátor	Unikátny identifikátor správy na jej sledovanie	2
Fragment	Číslo aktuálneho fragmentu správy (ak je správa fragmentovaná)	2
Príznaky	Príznaky stavu, ako napr. ACK (potvrdenie) alebo SYN (synchronizácia)	1
Kontrolný súčet	CRC16 na overenie integrity dát	2
Dáta	Užitočné zaťaženie prenášané v danom pakete	premenná

Table 1: Štruktúra hlavičky protokolu

Popis polí:

- **Typ správy:** Toto pole určuje typ paketu: dáta, potvrdenie (ACK), požiadavka na opätovné odoslanie (NACK), kontrola spojenia (Keep-Alive) a pod.
- **Identifikátor**: Každá správa bude mať unikátny identifikátor, aby bolo možné sledovať a rozlišovať pakety.
- **Fragment:** Toto pole udáva číslo fragmentu, ak je správa rozdelená na viac paketov.
- **Príznaky:** Príznaky slúžia na určenie stavu paketu. Napríklad, príznak ACK označuje potvrdenie prijatia paketu, SYN začiatok spojenia.
- **Kontrolný súčet:** CRC16 sa bude používať na overenie integrity paketu na strane prijímateľa.
- **Dáta:** Užitočné zaťaženie paketu, ktoré môže zahŕňať textové alebo binárne dáta.

2 Kontrola integrity dát

Na overenie integrity správ bude použitý CRC16. Táto metóda umožňuje detekovať chyby pri prenose dát. Ak sa kontrolný súčet vypočítaný na strane prijímateľa nezhoduje s kontrolným súčtom v pakete, predpokladá sa, že dáta boli poškodené, a paket bude požadovaný na opätovné odoslanie.

Proces výpočtu CRC16:

1. Na strane odosielateľa sa pre každú správu pred odoslaním vypočíta CRC16.
2. Kontrolný súčet sa pridá do hlavičky paketu.
3. Na strane prijímateľa sa po prijatí dát opäť vypočíta CRC16 a porovná s prijatým kontrolným súčtom.
4. k sa kontrolné súčty nezhodujú, paket sa považuje za poškodený a odošle sa požiadavka na jeho opätovné odoslanie (NACK).

3 Spoľahlivý prenos dát (ARQ)

Na zabezpečenie spoľahlivosti prenosu dát a opätovného odoslania poškodených alebo stratených paketov bude použitá metóda automatickej požiadavky na opätovné odoslanie (Automatic Repeat reQuest, ARQ). Základná myšlienka je nasledovná:

1. **Odoslanie paketu:** Odosielateľ odošle správu a uchováva jej kópiu, kým nedostane potvrdenie (ACK).
2. **Čakanie na potvrdenie (ACK):** Po odoslaní každého paketu odosielateľ očakáva potvrdenie od prijímateľa.
3. **Opätovné odoslanie (NACK):** Ak prijímateľ zistí chybu v pakete (na základe kontroly CRC), odošle požiadavku na opätovné odoslanie (NACK).
4. **Timeout a opätovné odoslanie:** Ak potvrdenie nepríde v stanovenej dobe (timeout), odosielateľ odošle paket znova.
5. **Fragmentácia správ:** Ak je správa príliš veľká, rozdelí sa na fragmenty. Každý fragment sa odošle a potvrdí samostatne.

4 Udržanie spojenia (Keep-Alive)

Aby sa zabezpečilo, že oba uzly sú aktívne a spojenie je udržiavané, bude použitý mechanizmus **Keep-Alive.** V pravidelných intervaloch jeden uzol pošle správu typu Keep-Alive druhému uzlu. Ak sa v stanovenom čase nedostaví odpoveď, uzol sa považuje za neaktívny.

Proces:

1. Uzly si pravidelne vymieňajú správy Keep-Alive.
2. Ak jeden z uzlov nedostane odpoveď v stanovenom čase, spojenie sa preruší alebo sa odošle chyba.

5 Úmyselné vytváranie chýb pre testovanie

Pre testovanie protokolu bude zahrnutá možnosť úmyselného vytvárania chýb v dátových paktoch. To môže byť vykonané pridaním voľby v užívateľskom rozhraní pre úmyselné poškodenie paketov alebo odstránenie časti dát pred odoslaním.

Spôsoby vytvárania chýb:

- **Poškodenie kontrolného súčtu:** Úmyselné zmenenie kontrolného súčtu na simuláciu chyby pri prenose.

- **Vynechanie paketu:** Vynechanie odoslania jedného alebo viacerých fragmentov, aby sa otestoval mechanizmus opätovného odoslania.
- **Zmena dát:** Zmena časti dát v správe na simuláciu chyby v prenose bitov.

6 Diagramy protokolu

6.1 Sekvenčný diagram (Sequence Diagram)

Sekvenčný diagram opisuje proces prenosu dát medzi dvoma uzlami, vrátane začatia spojenia, odoslania fragmentov a potvrdení, a tiež spracovanie strát dát. Základný postup:

1. Uzol A iniciuje spojenie (SYN).
2. Uzol B odpovedá potvrdením (SYN-ACK).
3. Uzol A odosiela dáta, ktoré môžu byť fragmentované.
4. Uzol B posiela ACK na každý prijatý fragment.
5. V prípade chyby uzol B odošle NACK a fragment sa odošle znova.

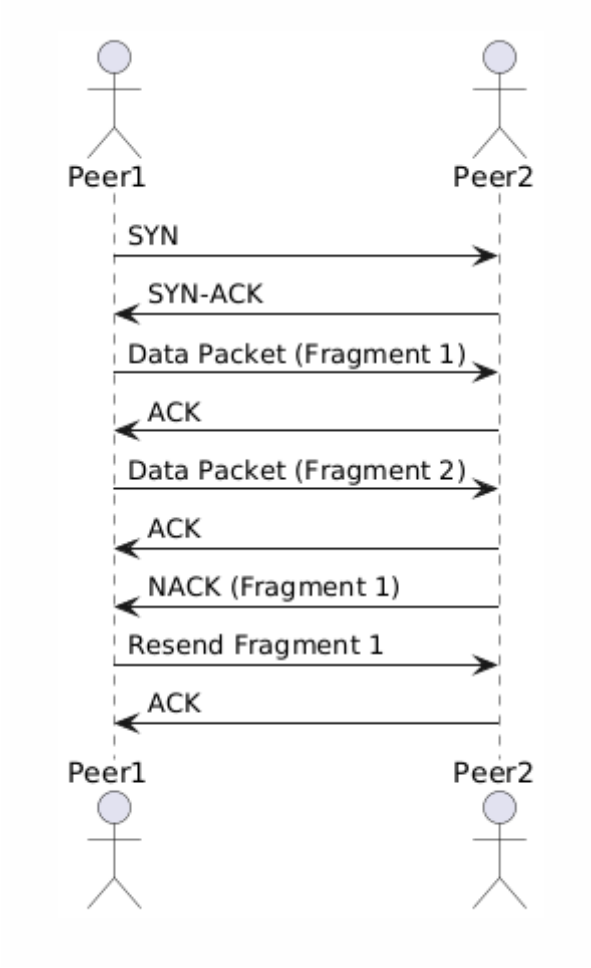

Figure 1: Sekvenčný diagram

6.2 Diagram aktivity (Activity Diagram)

Diagram aktivity opisuje správanie uzlov pri prenose dát, detekcii chýb, čakaní na potvrdenia a opätovnom odosielaní dát.

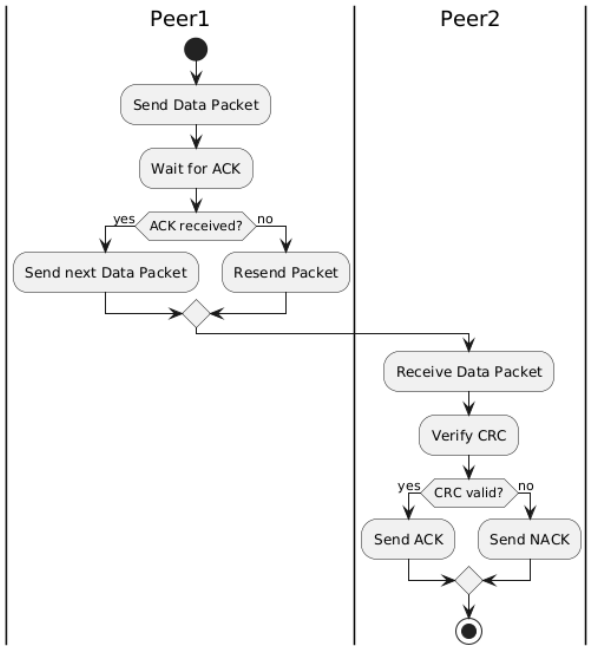

Figure 2: Sekvenčný diagram

6.3 Stavový diagram (State Diagram)

Stavový diagram opisuje rôzne stavy, v ktorých sa uzol môže nachádzať: čakanie na dáta, čakanie na potvrdenie, prenos dát, spracovanie chýb a pod.

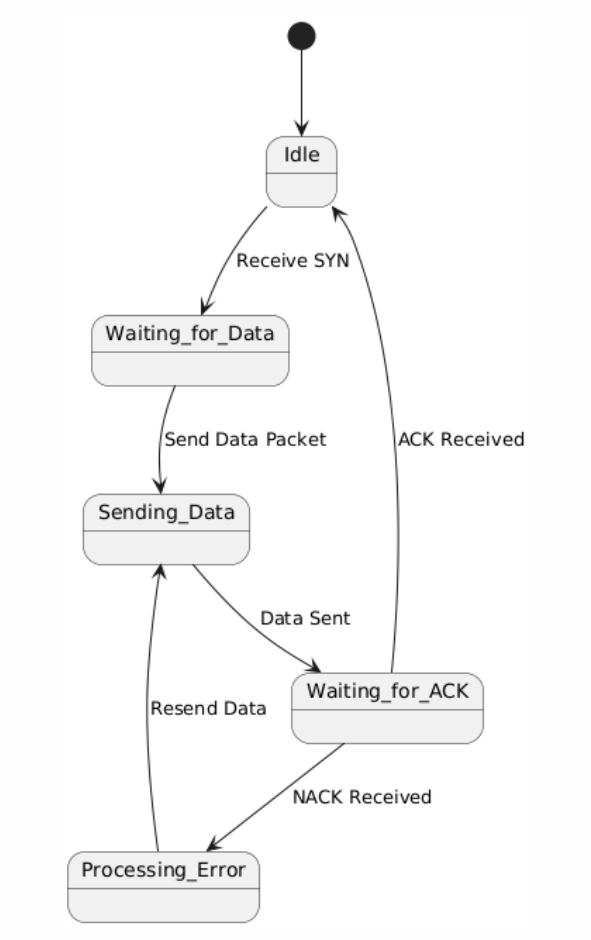

Figure 3: Sekvenčný diagram