SLOVENSKÁ TECHNICKÁ UNIVERZITA
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislava 4

Počítačové a komunikačné siete

Zadanie 2 – Komunikácia s využitím UDP protokolu

Lucia Murzová

AIS ID: 103066

ZS 2021/22

Ing. Kristián Košťál, PhD.

Štvrtok 8:00

6.12.2021

Lucia Murzová
AIS ID: 103066
Štvrtok 8:00

OBSAH

Lucia Murzová
AIS ID: 103066
Štvrtok 8:00

1. Zadanie

Navrhnite a implementujte program s použitím vlastného protokolu nad protokolom UDP (User Datagram Protocol) transportnej vrstvy sieťového modelu TCP/IP. Program umožní komunikáciu dvoch účastníkov v lokálnej sieti Ethernet, teda prenos textových správ a ľubovoľného súboru medzi počítačmi (uzlami).

Program bude pozostávať z dvoch častí – vysielacej a prijímacej. Vysielací uzol pošle súbor inému uzlu v sieti. Predpokladá sa, že v sieti dochádza k stratám dát. Ak je posielaný súbor väčší, ako používateľom definovaná max. veľkosť fragmentu, vysielajúca strana rozloží súbor na menšie časti - fragmenty, ktoré pošle samostatne. Maximálnu veľkosť fragmentu musí mať používateľ možnosť nastaviť takú, aby neboli znova fragmentované na linkovej vrstve.

Ak je súbor poslaný ako postupnosť fragmentov, cieľový uzol vypíše správu o prijatí fragmentu s jeho poradím a či bol prenesený bez chýb. Po prijatí celého súboru na cieľovom uzle tento zobrazí správu o jeho prijatí a absolútnu cestu, kam bol prijatý súbor uložený.

Program musí obsahovať kontrolu chýb pri komunikácii a znovu vyžiadanie chybných fragmentov, vrátane pozitívneho aj negatívneho potvrdenia. Po prenesení prvého súboru pri nečinnosti komunikátor automaticky odošle paket pre udržanie spojenia každých 5-20s pokiaľ používateľ neukončí spojenie. Odporúčame riešiť cez vlastne definované signalizačné správy.

Lucia Murzová
AIS ID: 103066
Štvrtok 8:00

2. Spôsob riešenia

Program je riešený spôsobom klient – server. Používateľ dostane na začiatku možnosť zvoliť či si prajete odosielať alebo prijímať s tým, že na strane servera – prijímateľa sa zadá port na ktorom bude počúvať a vypíše sa IP adresa zariadenia.

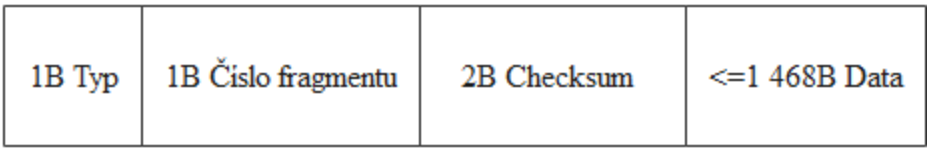
Na strane klienta je pre komunikáciu potrebné nastaviť práve túto IP adresu a port. Klient má potom možnosť zadať maximálnu veľkosť fragmentu a či si praje odosielať súbor alebo správu. Pokiaľ bude odosielaný súbor, má tiež možnosť zvoliť odosielanie so simulovaním chyby a obyčajné odoslanie.

Pre veľkosť fragmentu možno zadať číslo maximálne 1 465 (= 1 500 maximum B pre odosielané dáta - IP hlavička -UDP hlavička – moja hlavička). Po tejto úvodnej inicializácii prebehne spojenie pomocou 3 way handshake a prebieha odosielanie dát.

Pre ukončenie spojenia môže túto možnosť klient zvoliť v hlavnom menu. Pokiaľ si zvolil odosielanie správ, je potrebné poslať prázdnu správu – či už zo strany servera alebo klienta. Server má možnosť ukončiť spojenie po ukončení odosielania správ alebo súboru. Klient môže ukončiť spojenie vo svojom hlavnom menu, zvolením čísla 0.

Lucia Murzová
AIS ID: 103066
Štvrtok 8:00

3. Štruktúra hlavičky vlastného protokolu



3-1 Znázornenie fragmentu s vlastnou hlavičkou

Hlavička môjho protokolu obsahuje typ a číslo odosielaného fragmentu a tiež kontrolnú sumu pre daný fragment. Typ fragmentu serveru určuje, ako je potrebné daný fragment spracovať. Tiež bude slúžiť na poslanie informácie klientovi – či boli fragmenty správne prijaté alebo je niektoré potrebné poslať znovu. Veľkosť 1B je z dôvodu, že sú do typu ukladané ACSII hodnoty jednotlivých písmen, ktoré majú veľkosť 0-255.

Typy fragmentov:

- ACK = **A = 65** – potvrdzovanie správne prijatých správ
- SYN = **S = 83** – nadviazanie spojenia
- FIN = **F = 70** – ukončenie spojenia
- NACK = **N**, v sekcii Dát sú uložené čísla fragmentov, ktoré neboli správne prijaté = 78
- Dáta = **D = 68**
- Správy = **P** = 80, v sekcii Dát je klientom zadaná veľkosť fragmentu a počet potrebných fragmentov
- Súbor = **U** = 85, v sekcii Dát bude uložený názov súboru a počet potrebných fragmentov.

Číslo fragmentu je potrebné aby bolo možné správne zapísať prijatý súbor aj v prípade, že by fragmenty prišli v inom poradí ako boli odoslané. Keďže program bude odosielať skupiny po N fragmentov, po ktorých dostane ACK/NACK, toto číslo bude 1-N (zadané v programe). Keďže veľkosť tejto skupiny nebude väčšia ako 255, pre uloženie čísla fragmentu je potrebný 1B.

V časti checksum je uložený zvyšok metódy kontrolnej sumy, ktorá je počítaná z odosielanej správy. Tento údaj je potrebný aby server vedel skontrolovať správnosť prijatých dát. Použila som metódu knižnice binascii - crc16, ktorá vráti 2B zvyšok.

Lucia Murzová
AIS ID: 103066
Štvrtok 8:00

3.1 Zmena hlavičky oproti návrhu

1B Typ	4B Číslo fragmentu	4B Checksum	<=1 463B Data

3-2 Znázornenie pôvodnej hlavičky

Oproti návrhu som zmenila v hlavičke veľkosť pre číslo fragmentu zo 4B na 1B a to z toho dôvodu, že je odosielané poradové číslo fragmentu v danej skupine, v ktorej je posielaný. Týmto spôsobom bude vždy číslo fragmentu menšie ako 255 (počet odosielaných fragmentov v jednej skupine závisí od nastavenia globálnej premennej N v programe) a preto pre jeho odoslanie stačí 1B.

Keďže programovací jazyk Python zaokrúhľuje na 4 miesta, hlavička s dátami 1B-1B-4B by bola zaokrúhlená na 8, zmenila som aj využitú crc metódu na crc16. Vďaka tomu pre Chceckum v hlavičke postačujú 2B a celá nová hlavička má tak veľkosť 4B.

Upravené boli aj potrebné typy správ – rozhodla som sa nevyužiť typ RST na okamžité ukončenie spojenia, ako ani Keep Alive z dôvodu neiplmenetovania metódy pre udržanie spojenia.

Lucia Murzová
AIS ID: 103066
Štvrtok 8:00

4. Metóda kontrolnej sumy

Pre vypočítanie kontrolnej sumy odosielaného fragmentu bude program využívať metódu crc_hqx knižnice binascii, ku ktorej som čerpala informácie aj na tejto stránke. Program má stanovený generujúci polynóm, ktorý je známy pre odosielajúci aj prijímajúci uzol.

Metóda robí nad generujúcim polynómom a danými dátami v bitoch postupne operáciu XOR (pravdivostná tabuľka tejto operácie je uvedená na obrázku 4-1), až pokiaľ výsledok tohto delenia nie je kratší ako náš generujúci polynóm. Keď je výsledok kratší, ide o zvyšok po tomto delení. Tento zvyšok sa ukladá do hlavičky – časti checksum a fragment sa odosiela. Ukážku tohto delenia je možné vidieť na obrázku 4-2 nižšie.

Po prijatí fragmentu server tento postup zopakuje a pokiaľ sa bude výsledok zhodovať vieme, že dáta neboli nijak poškodené. Pokiaľ bude výsledok na strane servera rozdielny, vyžiada od klienta odoslanie tohto fragmentu ešte raz.

XOR	0	1
0	0	1
1	1	0

4-1 XOR pravdivostná tabuľka

```
Input data is the byte 0xC2 = b11000010.
As generator polynomial (=divisor), let's use b100011101.
The divisor has 9 bits (therefore this is a CRC-8 polynomial), so append 8 zero bits to the input pattern .
Align the leading '1' of the divisor with the first '1' of the divident and perform a step-by-step school-like division, using XOR operation for each bit:
ABCDEFGHIJKLMNOP
1100001000000000
100011101
---------
010011001
 100011101
 ----------
 000101111
    100011101          (*)
    ---------
    001100101
      100011101
      ---------
      010001001
       100011101
       ---------
       000001111 = 0x0F
ABCDEFGHIJKLMNOP
The actual CRC value is 0x0F.
```

4-2 Ukážka delenie metódou crc16

Lucia Murzová
AIS ID: 103066
Štvrtok 8:00

4.1 Zmena kontrolnej sumy oproti návrhu

V návrhu som uviedla použitie metódy crc32, ktorej výsledok bol zvyšok po delení o veľkosti 4B. Po úprave hlavičky – veľkosti miesta pre odosielané číslo fragmentu by hlavička s funkciou crc32 reálne zaberala 6B, no programom by bola zaokrúhlená na 8.

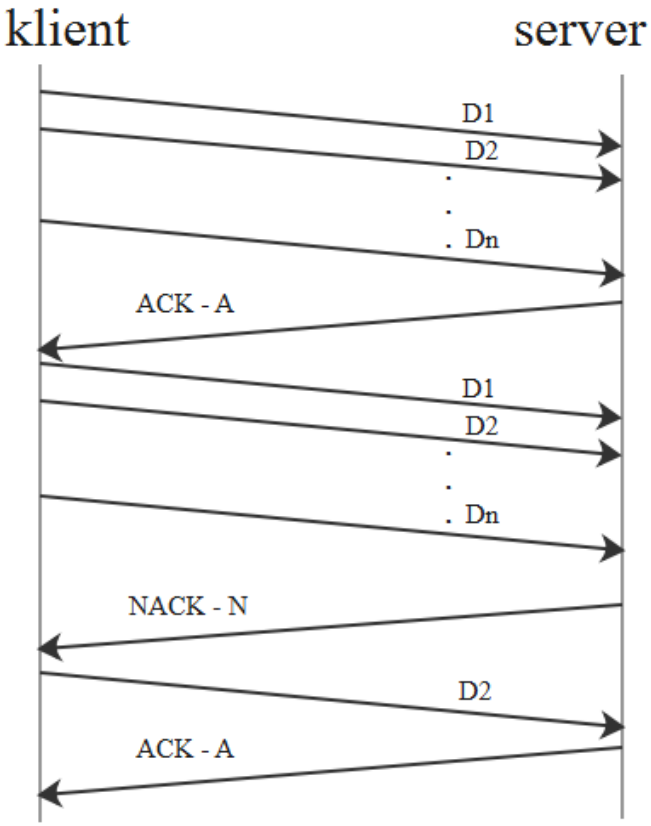
Z toho dôvodu som sa rozhodla pre použitie výpočtu kontrolnej sumy pomocou funkcie crc16 – binascii. crc_hqx, ktorej výsledok má veľkosť 2B a celá hlavička tak zaberá dokopy iba 4B.

Lucia Murzová
AIS ID: 103066
Štvrtok 8:00

5. Fungovanie ARQ

Program odosiela N fragmentov za sebou (napríklad 12). Po prijatí týchto N fragmentov pošle server správu ACK – A ak prebehlo prijatie všetkých fragmentov v poriadku, inak pošle NACK – označenie N, kde v sekcii dáta sú zapísané čísla fragmentov, ktoré neboli prijaté správne. Následne prebehne odoslanie týchto fragmentov nanovo a po ich správnom prijatí sa posiela ďalšia N-tica.

Po 3 neúspešných pokusoch o doručenie fragmentu server ukončuje odosielanie správ, následne má tiež možnosť ukončiť aj celkové spojenie s klientom. Spôsob fungovania ARQ je zobrazený na obrázku 5-1 nižšie.

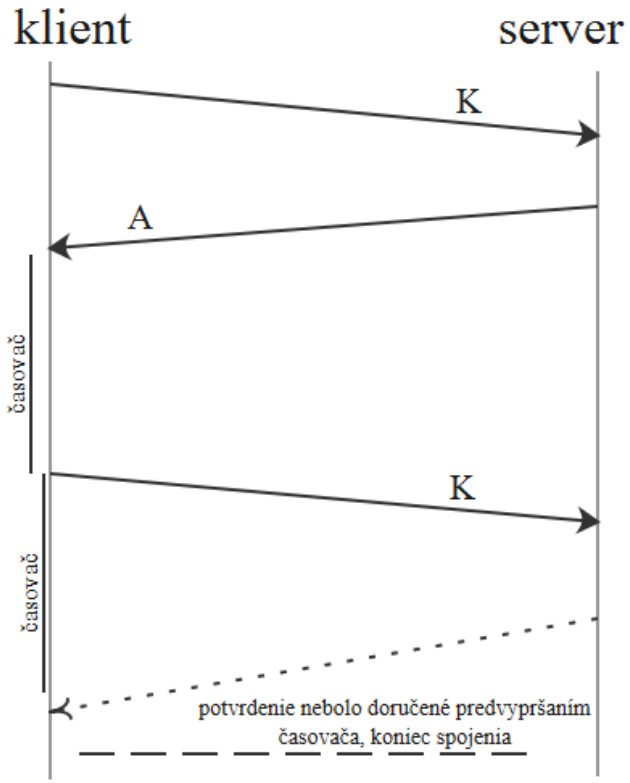


5-1 ARQ schéma

Lucia Murzová
AIS ID: 103066
Štvrtok 8:00

6. Metóda pre udržanie spojenia - návrh

Po odoslaní správy alebo súboru sa na strane klienta zapne časovač, ktorý každých 15 sekúnd pošle správu typu K – keep alive. Pokiaľ do 15 sekúnd od odoslania tejto správy klient nedostane od servera potvrdenie – správu typu A, spojenie sa uzatvára a program prechádza na začiatok, kde môže používateľ nastaviť nový server / klient. Schéma metódy je zobrazená na obrázku 6-1 nižšie.



6-1 Metóda pre udržanie spojenia

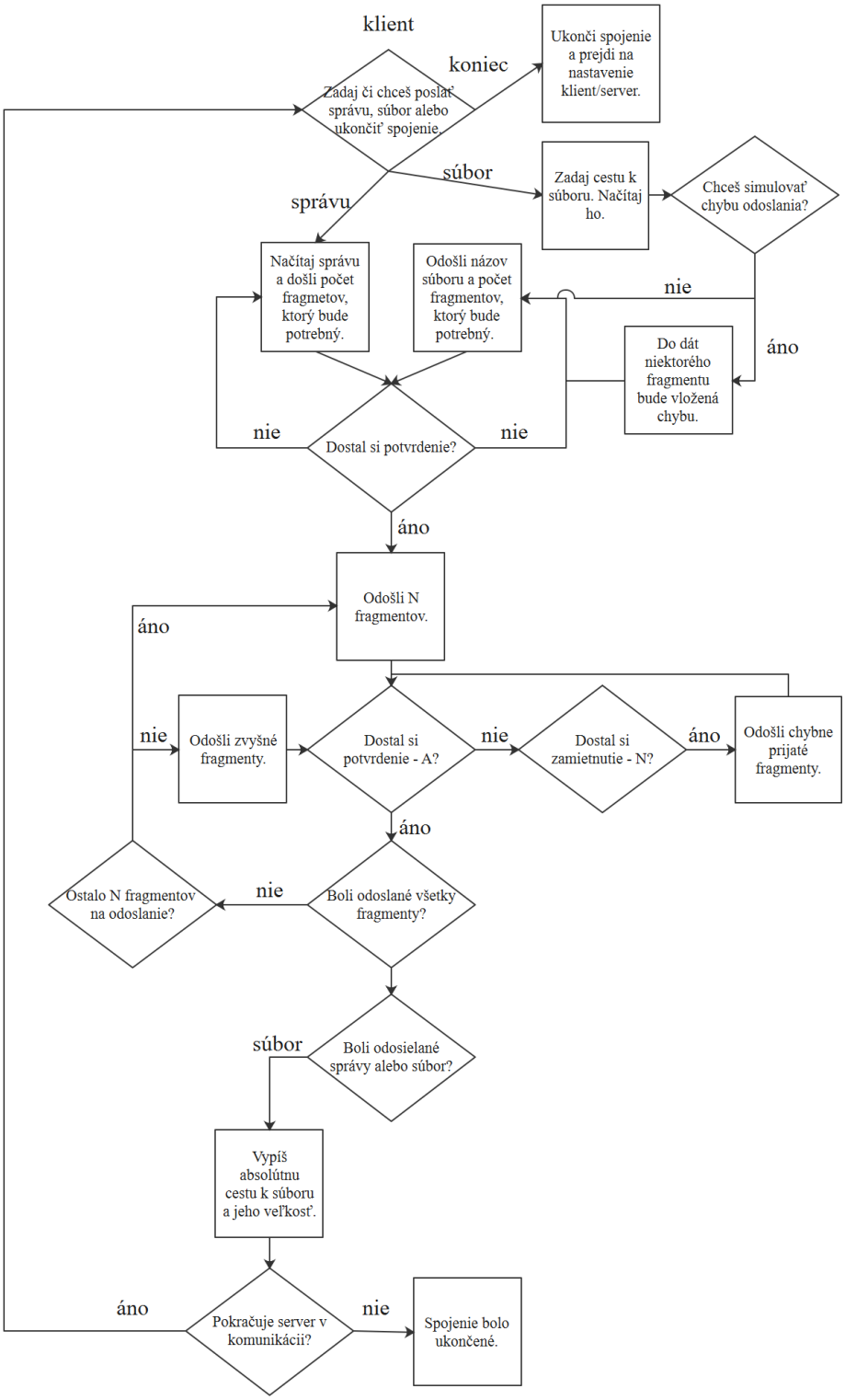
6.1 Zmena metódy pre udržanie spojenia oproti návrhu

Z dôvodu komplexnosti riešenia a časovej tiesne som sa rozhodla metódu pre udržanie spojenia neimplementovať.

Lucia Murzová
AIS ID: 103066
Štvrtok 8:00

7. Diagram spracovávania komunikácie – klient

Nižšie uvedený diagram spracovania komunikácie na strane klienta predpokladá, že nadviazanie spojenia so serverom prebehlo v poriadku a môže pristúpiť k odosielaniu správ / súborov.

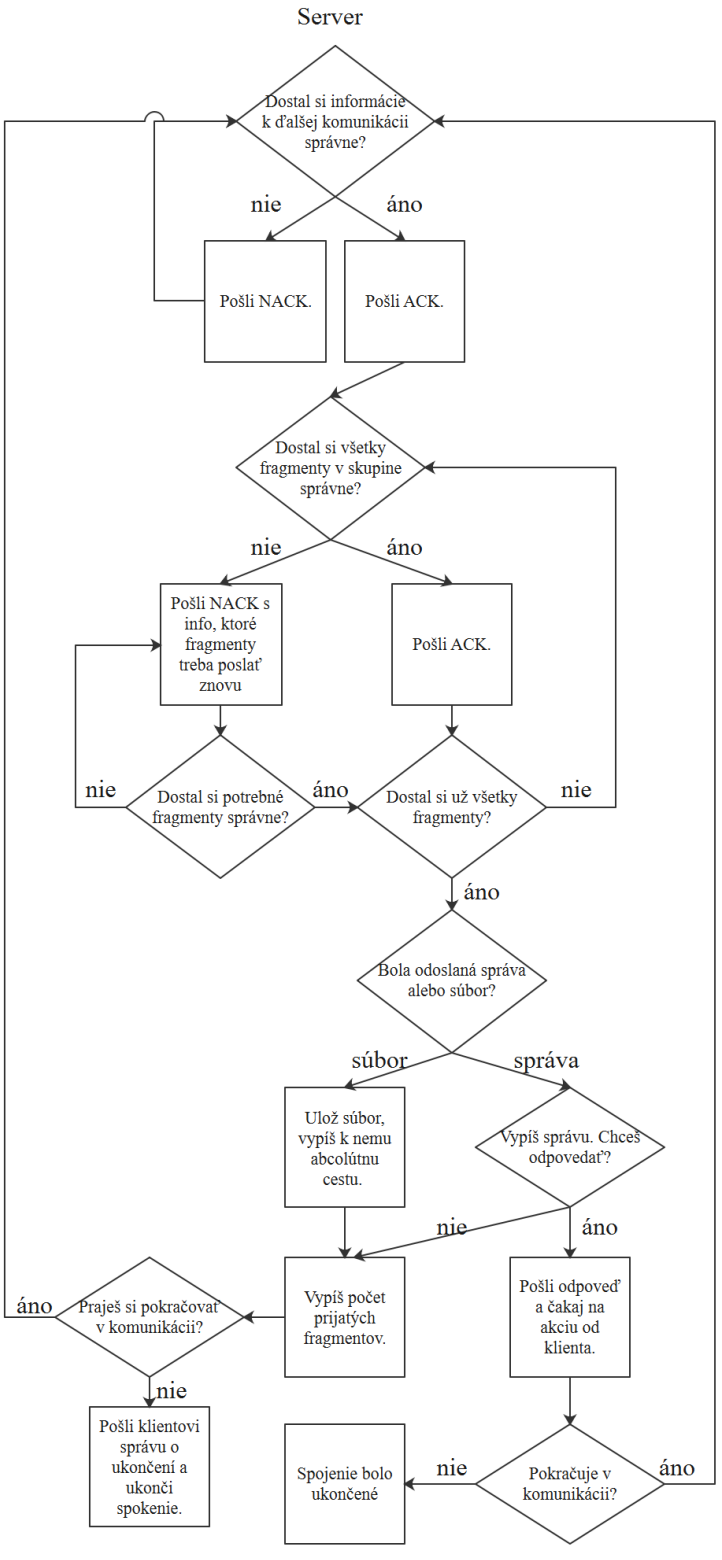


Zjednodušený diagram spracovávania komunikácie na strane klienta

Lucia Murzová
AIS ID: 103066
Štvrtok 8:00

8. Diagram spracovávania komunikácie – server

Nižšie uvedený diagram spracovania komunikácie na strane servera taktiež predpokladá, že pripojenie klienta prebehlo v poriadku a dochádza k čakaniu na doručenie dát. Diagram končí čakaním na akciu od klienta, kedy ten môže zvoliť ukončenie spojenia alebo nové odosielanie správy / súboru.



Zjednodušený diagram spracovávania komunikácie na strane servera

Lucia Murzová
AIS ID: 103066
Štvrtok 8:00

8.1 Zmena v spracovaní komunikácie

Pri spracovaní komunikácie som pridala možnosť ukončiť spojenie aj zo strany servera. Túto možnosť dostane vždy po ukončení odosielania správ alebo súboru. Následne sa klientovi odošle ACK správa ako potvrdenie, že pokračuje v komunikácii, alebo FIN správa ako ukončenie spojenia. Klient čaká na túto akciu, aby nedošlo k pokusu o doručenie dát vo fáze, kedy server nepočúva.