

# POČÍTAČOVÉ A KOMUNIKAČNÉ SIETE

## cvičenia

ak. rok 2022/23, zimný semester

### Zadanie 2: Komunikácia s využitím UDP protokolu

#### Zadanie úlohy

Navrhните 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 znovuvyžiadanie chybných fragmentov, vrátane pozitívneho aj negatívneho potvrdenia. Po zapnutí programu, komunikátor automaticky odosiela paket pre udržanie spojenia každých 5s pokiaľ používateľ neukončí spojenie ručne. Odporúčame riešiť cez vlastne definované signalizačné správy a samostatný thread.

#### Program musí mať nasledovné vlastnosti (minimálne):

1. Program musí byť implementovaný v jazykoch C/C++ alebo Python s využitím knižníc na prácu s UDP socket, skompilovateľný a spustiteľný v učebniciach. Odporúčame použiť python modul *socket*, C/C++ knižnice *sys/socket.h* pre linux/BSD a *winsock2.h* pre Windows. Iné knižnice a funkcie na prácu so socketmi musia byť schválené cvičiacim. V programe môžu byť použité aj knižnice na prácu s IP adresami a portami: *arpa/inet.h*  
*netinet/in.h*
2. Program musí pracovať s dátami optimálne (napr. neukladať IP adresy do 4x int).
3. Pri posielaní súboru musí používateľovi umožniť určiť cieľovú IP a port.
4. Používateľ (stačí na strane vysieláča) musí mať možnosť zvoliť si max. veľkosť fragmentu a meniť ju dynamicky počas behu programu pred posielaním správ (neplatí pre pakety na udržanie spojenia).
5. Obe komunikujúce strany musia byť schopné zobrazovať:
  - a. názov a absolútnu cestu k súboru na danom uzle,
  - b. veľkosť a počet fragmentov vrátane celkovej veľkosti správy/súboru.

6. Možnosť simulovať chybu prenosu odoslaním minimálne 1 chybného fragmentu pri prenose správy a súboru (do dátovej časti fragmentu alebo do checksum je cielene vnesená chyba, to znamená, že prijímajúca strana deteguje chybu pri prenose).
7. Prijímajúca strana musí byť schopná oznámiť odosielateľovi správne aj nesprávne doručenie fragmentov. Pri nesprávnom doručení fragmentu vyžiada znovu poslať poškodené dáta.
8. Možnosť odoslať 2MB súbor a v tom prípade ich uložiť na prijímacej strane ako rovnaký súbor, pričom používateľ zadáva iba cestu k adresáru kde má byť uložený.
9. Program musí byť organizovaný tak, aby oba komunikujúce uzly mohli prepínať medzi funkciou vysielача a prijímača bez reštartu programu automatizovane (jedna strana pošle správu na prepnutie, dostane ACK z druhej strany a uzly sa automaticky prepnú)  
- program nemusí (ale môže) byť vysielач a prijímač súčasne.

#### **Odvzdáva sa:**

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

Pri predvedení riešenia je podmienkou hodnotenia schopnosť doimplementovať jednoduchú funkcionality na cvičení.

#### **Hodnotenie**

Celé riešenie - max. 20 bodov (min. 8), z toho:

- max. 5 bodov za návrh riešenia;
- max. 1 bod za doplnenú funkčnosť (doimplementáciu) priamo na cvičení v požadovanom termíne podľa harmonogramu cvičení; V prípade, ak študent nesplní úlohu zadanú priamo na cvičeniach, nehodnotí sa výsledné riešenie;
- max. 14 bodov za výsledné riešenie.

Návrh a zdrojový kód implementácie študent odovzdáva v elektronickom tvare do AISu v určených termínoch.

#### **Bodovací klíč:**

Úloha	Body	Min bodov
<b>Návrh programu a komunikačného protokolu</b> <i>Hodnotí sa prehľadnosť a zrozumiteľnosť odovzdanej dokumentácie ako aj kvalita navrhovaného riešenia. 5 bodov získa študent, ktorý má v dokumentácii uvedené všetky podstatné informácie o fungovaní jeho programu. <u>Korektne navrhnutú štruktúru hlavičky vlastného protokolu, opis použitej metódy kontrolnej sumy a fungovania ARQ, metódy pre udržanie spojenia, diagram spracovávania komunikácie na oboch uzloch (sekvenčný), popis jednotlivých častí zdrojového kódu (knížnice, triedy, metódy, ...).</u> Podčiarknuté sú minimálne požiadavky.</i>	0-5	2

<b>Príprava</b> <i>Nastaviť a overiť konektivitu medzi 2 uzlami, spustiť na oboch uzloch Wireshark. Spustiť zachytávanie vo Wireshark a nastaviť filter na zobrazenie iba komunikácie vlastného programu. Každý prenos sa kontroluje aj cez Wireshark.</i>	0,5	
<b>Nastavenie IP a port</b> <i>1 bod získa študent, ktorého program umožňuje nastaviť na prijímajúcom uzle port, na ktorom počúva a na vysielajúcom uzle IP adresu a port prijímača.</i>	0,5	
<b>Prenos súboru menšieho ako nastavená veľkosť fragmentu</b> <i>2 body získa študent, ktorého program umožňuje úspešne preniesť súbor menší ako je nastavená veľkosť fragmentu podľa pokynov cvičiaceho.</i>	0-2	
<b>Simulácia chyby pri prenose súboru a správy</b> <i>2 body získa študent, ktorého program umožňuje úspešne preniesť súbor/správu pri simulácii chyby prenosu a má korektne implementovanú detekciu chyby a ARQ metódu. Korektným použitím komplexnejšej ARQ metódy je možné získať ďalší 1b. Príklady ARQ metód su medzi zdrojmi nižšie. Vaším návrhom na zlepšenie sa nebránime.</i>	0-3	
<b>Prenos 2MB súboru</b> <i>2 body získa študent, ktorého program umožňuje úspešne preniesť súbor s veľkosťou 2MB pri nastavení veľkosti fragmentu podľa pokynov cvičiaceho a uložiť ho ako rovnaký súbor, zobrazuje absolútnu cestu k súboru a počet fragmentov spolu s ich veľkosťou.</i>	0-2	6
<b>Udržiavanie spojenia</b> <i>2 body získa študent, ktorého program po prenesení súboru udržuje spojenie pomocou vlastných signalizačných správ a zobrazí informáciu, ak bolo spojenie prerušené. Korektným použitím komplexnejšej metódy pre udržanie spojenia je možné získať ďalší 1b.</i>	0-3	
<b>Finálna dokumentácia a kvalita spracovania</b> <i>Hodnotí sa prehľadnosť a zrozumiteľnosť odovzdanej dokumentácie ako aj kvalita pracovania celového riešenia. 3 body získa študent, ktorý má v dokumentácii uvedené všetky podstatné informácie o fungovaní jeho programu vrátane zmien, ktoré nastali v implementácii oproti návrhu. Dokumentácia tiež musí obsahovať aspoň 1 ukážku testovacieho scenáru (ideálne ako screeny z Wireshark).</i>	0-3	
<b>Doplnená funkčnosť (doimplementácia) priamo na cvičení.</b> <i>1 bod získa študent, ktorý doimplementuje úlohu v jej plnom rozsahu a predvedie jej funkčnosť bez toho, aby program padal alebo vyhadzoval akékoľvek chybové hlášky súvisiace s touto úlohou.</i>	0-1	0,5
<b>Spolu:</b>	20	8,5

Literatúra:

- materiály k zadaniu - dokumentový server v AIS,
- prednášky PKS

ARQ literatúra:

- Stop & Wait - [Sliding Window Protocol](#)
- Selective repeat - <https://www.youtube.com/watch?v=WflhQ3o2xow>
- Go Back N - <https://www.youtube.com/watch?v=QD3oCelHJ20>