

OSI 4 - Transportní vrstva - UDP, TCP, segmentace a znovusestavení, řízení přenosové rychlosti, kontrola chyb, ne python jenom teorie

OSI 4 - transportní vrstva

Hlavním úkolem je zajistit přenos dat mezi koncovými zařízeními, od odesílatele k příjemci. Tato vrstva poskytuje spojení mezi aplikacemi, adresování pomocí portů, segmentace dat a jejich znovusestavení, řízení toku dat a kontrolu chyb

### **TCP**

Spojově orientovaný protokol - před samotným přenosem dat vytváří spojení mezi odesílatelem a příjemcem

#### Vlastnosti TCP

Spolehlivý přenos dat - doručení paketů ve správném pořadí

Navazování spojení - three-way handshake (1. Klient pošle SYN packet serveru, 2. Server odpoví SYN-ACK, 3. Klient odpoví ACK)

Potvrzování příjmu - ACK

Řízení toku - zabraňuje přehlcení

Kontrola - součty detekující poškození dat

TCP se používá tam, kde je potřeba spolehlivost přenosu dat -  
Webové stránky, Email, Přenos souborů.....

### **UDP**

Nespojově orientovaný protokol - nevytváří spojení před přenosem dat

#### Vlastnosti UDP

Nespolehlivý přenos - negarance přenosu všech paketů

Bez navazování spojení - odesílána před domluvou

Bez potvrzování příjmu - žádné ACK

Bez řízení toku - odesílá data bez ohledu na kapacitu

Rychlý přenos

Používá se tam, kde je důležitější rychlost než spolehlivost -  
Streamování, Online hry, VoIP, DNS.....

## **Segmentace a znovusestavení**

Transportní vrstva rozděluje velké bloky dat na malé části, které se nazývají segmenty nebo datagramy. Zefektivňují přenos (kvůli MTU), snazší oprava chyb a sdílejí přenosové médium

Proces segmentace

1. data z aplikační vrstvy jsou přijata
2. rozdělení na segmenty
3. k segmentům přidá hlavičku
4. segmenty předány k dalšímu zpracování

Proces znovusestavení

1. příjem segmentů
2. podle sekvenčních čísel seřadí segmenty do pořadí (pouze TCP)
3. z hlaviček se získají potřebné informace pro kontrolu a zpracování
4. složí segmenty zpět do bloku
5. předání bloků

## **Řízení přenosové rychlosti**

Mechanismus, který zabraňuje zahlcení příjemce daty. Především používáno v TCP protokolech

### Metody

1. Sliding window
  - příjemce informuje odesílatele o své dostupnosti bufferu
  - odesílatel může poslat jenom kolik se vejde - okno
  - okno se dynamicky mění v závislosti na prostředcích příjemce
2. Congestion control
  - předcházení zahlcení sítě
  - TCP používá několik algoritmů - Slow Start, Congestion avoidance, fast retransmit, fast recovery

## **Kontrola chyb**

Zajišťuje detekci a opravu chyb, které mohou nastat během přenosu

### Metody

#### 1. Checksum

- TCP i UDP
- odesílatel vypočítá checksum a předá jej do hlavičky
- příjemce znovu checksum a porovná hlavičku
- pokud neshoda = poškozená data

#### 2. Potvrzování a časovače

- Pouze TCP
- každé přijetí = potvrzení
- odesílatel nastaví časovač = pokud nepřijde v intervalu, data jsou ztracena odeslána znovu

#### 3. Sekvenční čísla

- segment má svoje číslo
- příjemce může identifikovat chybějící segmenty