

Objektovo  
orientované  
programovanie  
- *pokročilí*

Učiteľ:  
**Ing. Jozef Wagner PhD.**

Učebnica:  
<https://oop.wagjo.com/>

# OPGP

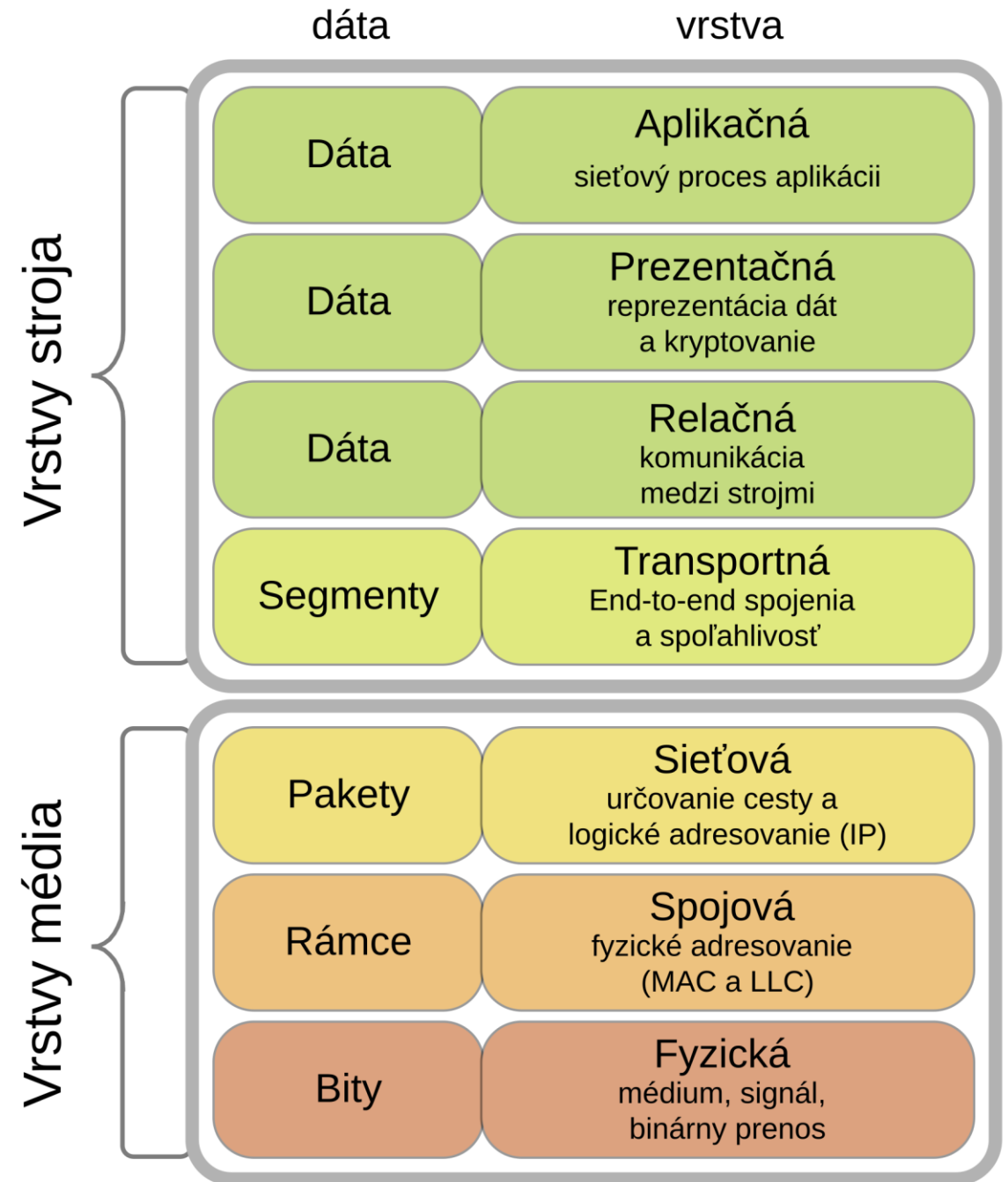
## Pokročilí 18

1. User Datagram Protocol
2. Prijímanie UDP správ
3. Posielanie UDP správ
4. Vlákna, klient/server

# OSI model

**UDP** (User Datagram Protocol)  
je protokol na úrovni  
transportnej vrstvy (layer 4)

UDP a TCP sú základné  
transportné protokoly



# UDP (User Datagram Protocol)

## Nevytvára spojenie

Je **nespoľahlivý** (bez záruky doručenia alebo poradia paketov)

Pracuje s **datagramami** (malými paketmi dát)

Vlastnosti:

- IP adresa a port (napr. **127.0.0.1:12345** pre localhost) pre adresovanie
- Veľkosť správy (datagramu) je max. **65kB**
- podporuje **broadcast** a **multicast**

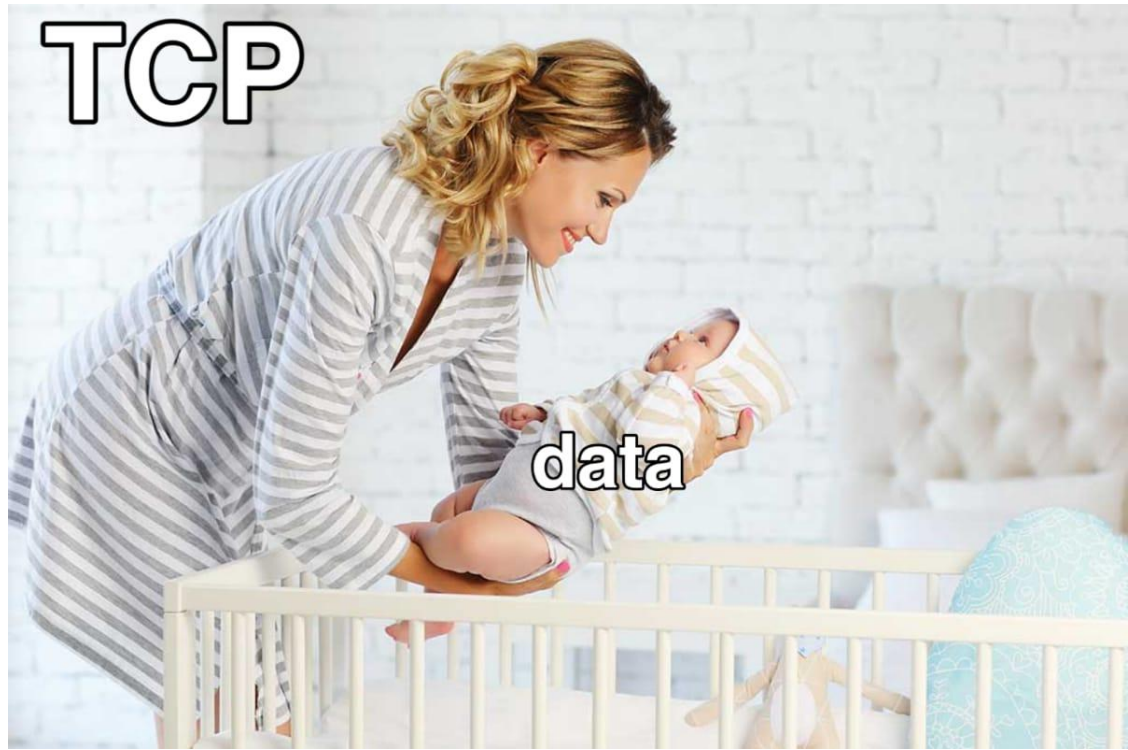
# Nevýhody UDP

Žiadna kontrola chýb (chyby sa ignorujú)

Žiadne potvrdenie doručenia ani garancia poradia

Pakety sa môžu stratiť alebo prísť v inom poradí

Je často obmedzovaný, pretože sťažuje firewallom a NAT zariadeniam sledovanie



# UDP v Pythone

Štandardný modul `socket`

```
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
```

UDP na adresovanie používa tuple IP + port: `("127.0.0.1", 5000)`

# Prijímanie dát

Je potrebné nastaviť socket aby sledoval určitý port

```
sock.bind(("0.0.0.0", 5000))
```

Prijatie správy – `recvfrom`

Prijaté dáta je nutné dekodovať z bajtov napr. na reťazec.

# Prijímanie dát

```
sock.bind(("0.0.0.0", 5000))
while True:
    try:
        # Prijatie správy (datagramu)
        data, addr = sock.recvfrom(BUFFER_SIZE)
        # Dekódovanie správy
        message = data.decode('utf-8')
        # Vypísanie správy do konzoly
        print(f"Nová správa od {addr[0]}:{addr[1]}: {message}")
    except Exception as e:
        print("Chyba v príjme:", e)
        break
```

# Odosielanie dát

Potrebuje IP adresu a port adresáta

Správu musíme zakódovať do poľa bajtov

```
addr = (IP, PORT)
```

```
message = input("Zadaj správu: ")
```

```
bytes = message.encode('utf-8')
```

```
sock.sendto(bytes, addr)
```

# Súčasné prijímanie aj odosielanie

Ak chceme naraz dáta prijímať aj odosielať,  
je potrebné prijímanie mať na samostatnom vlákne

Používame štandardný modul `threading`

```
thread = threading.Thread(target=receive_messages, args=  
(sock,), daemon=True)  
thread.start()
```