

Objektovo  
orientované  
programovanie  
- *pokročilí*

Učiteľ:  
**Ing. Jozef Wagner, PhD.**

Učebnica:  
<https://oop.wagjo.com/>

# OPGP

## Pokročilí 19

1. Metódy smerovania správ
2. Unicast
3. Broadcast, Multicast
4. Anycast

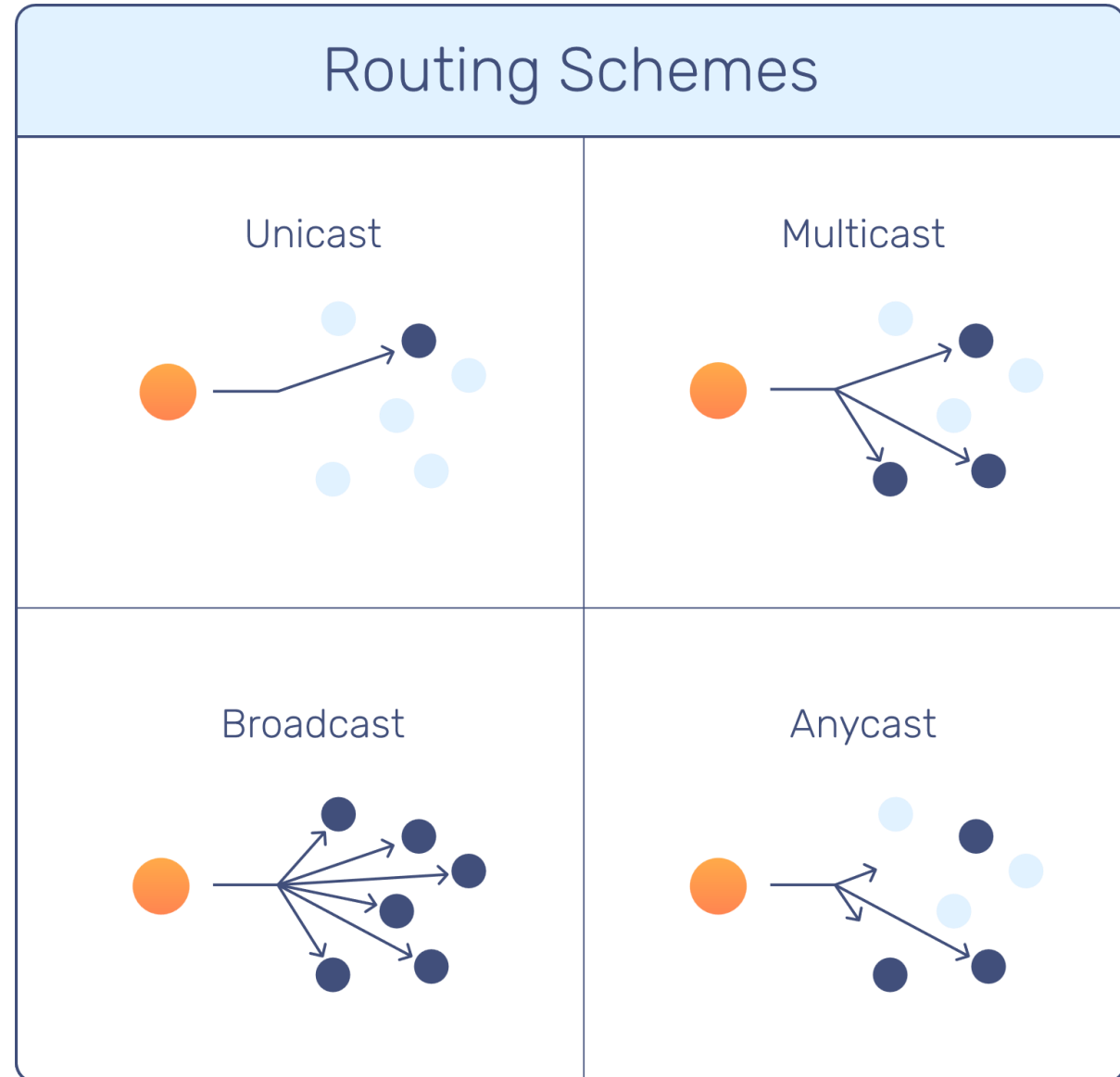
# Typy smerovania správ

**Unicast** - one-to-one - client/server a peer-to-peer komunikácia.

**Broadcast** - one-to-all – všetkým zariadeniam v sieti

**Multicast** - one-to-many – tým zariadeniam, ktoré sa sami prihlásia na odber

**Anycast** - one-to-one-of-many - jednému z množiny zariadení



# Unicast

## **one-to-one**

Správa je posielaná jednému zariadeniu,  
ktoré je identifikované IP adresou a portom

Je to najčastejší typ posielania a je využívaný v client/server a  
peer-to-peer komunikácii

Celú minulú hodinu sme využívali unicast

# Broadcast

**one-to-all** - *správa je poslaná všetkým zariadeniam v sieti*

Správy nie sú routrami prenášané na iné podsiete.

Nie je možné poslať broadcast správu celému "internetu"

Používa sa ak zariadenia ešte nepoznajú IP adresy iných zariadení v podsieti

Je potrebné explicitne povoliť `socket.SO_BROADCAST`

# Broadcast

Cieľová IP adresa je `255.255.255.255`, alebo `<broadcast>`.

Pri directed broadcast sa uvedie presná adresa podsiete, napr. `192.168.1.255`.

Router môže **directed broadcast** správu preposlať ďalej, ale je potrebné to explicitne na routoch povoliť.

Nastavenie `socket.SO_REUSEADDR` umožní, aby na porte mohli naraz počúvať viaceré aplikácie.

# Broadcast - vysielanie

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

```
# povolíme broadcast
```

```
sock.setsockopt(socket.SOL_SOCKET, socket.SO_BROADCAST, 1)
```

```
# Možnosti cieľovej adresy:
```

```
sock.sendto(b"HELLO", ("255.255.255.255", 9999)) # limited
```

```
sock.sendto(b"HELLO", ("192.168.1.255", 9999)) # directed
```

```
sock.sendto(b"HELLO", ("<broadcast>", 9999)) # špeciálny string
```

# Multicast

**one-to-many** - správa je poslaná viacerým zariadeniam a to tým, ktoré sa sami prihlásia na odber

Multicast správy von je možné na routoch povoliť ručne, napr. vo firemných sieťach

Využitie je v IPTV / live TV, IoT zariadeniach (Smart Home) a pri objavovaní (napr. tlačiarne)

# Multicast

Musíme zvoliť IP adresu skupiny, v ktorej budeme komunikovať.

Rozsah IP adries **224.0.0.0 - 239.255.255.255**

Pre aplikácie je časť **239.0.0.0 - 239.255.255.255**

Prijímateľ sa prihlási do skupiny a až potom môže začať prijímať správy

Po ukončení prijímania sa zo skupiny odhlási

# Multicast - prijímanie

```
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
sock.bind(("", 5007)) # dôležité: bind na prázdnu adresu!
```

```
# Prihlásenie sa do multicast skupiny
```

```
mreq = struct.pack("4s4s", socket.inet_aton("239.1.2.3"),
socket.inet_aton("0.0.0.0"))
```

```
sock.setsockopt(socket.IPPROTO_IP, socket.IP_ADD_MEMBERSHIP, mreq)
```

```
data, addr = sock.recvfrom(1024)
```

```
message = data.decode('utf-8', errors='replace')
```

```
print(f"Prijaté od {addr[0]}:{addr[1]} {message}")
```

```
# Odhlásenie z multicast skupiny (leave)
```

```
sock.setsockopt(socket.IPPROTO_IP, socket.IP_DROP_MEMBERSHIP, mreq)
```

# Anycast

**one-to-one-of-many** - správa je poslaná jednému zariadeniu (väčšinou najbližšiemu) z množiny zariadení

Všetky zariadenia v skupine majú **úplne rovnakú IP adresu!**

Smeruje správu fyzicky najbližšiemu zariadeniu.

Funguje hlavne na globálnej sieti a využíva sa na to, aby správy nešli zbytočne dlhú cestu.

Nastavuje sa iba na strane serverov a siete, **klient o ničom nevie a nič extra nerobí.**