

Objektovo  
orientované  
programovanie  
- *pokročilí*

Učiteľ:  
**Ing. Jozef Wagner, PhD.**

Učebnica:  
<https://oop.wagjo.com/>

# OPGP

## Pokročilí 21

1. Prúd dát, Segmentácia
2. Posuvné okno, Retransmisia
3. Flow Control, Congestion Control

# Prúd údajov

TCP neposiela správy,  
ale prúd údajov - bajtov.

Časti dát nemajú začiatok ani  
koniec

`sock.recvfrom()` často vráti  
iba časť dát

Vo nasledujúcich príkladoch  
druhej strane postupne prídu  
dáta `HELLO WORLD`

```
sock.sendto(b"HELLO WORLD", addr)
```

```
sock.sendto(b"HELLO ", addr)
```

```
sock.sendto(b"WORLD", addr)
```

```
sock.sendto(b"HEL", addr)
```

```
sock.sendto(b" LLO W", addr)
```

```
sock.sendto(b"O", addr)
```

```
sock.sendto(b"RLD", addr)
```

# Segmentácia

Prúd údajov sa rozkúskuje na menšie časti. TCP správa sa volá TCP segment.

`Lorem ipsum dolor sit amet, co`

`nsectetur adipiscing elit, ...`

# Segmentácia

Všetky bajty v prúde sa očísľujú, aby sa zachovalo poradie.

```
 Lorem ipsum dolor sit amet, co  
 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
```

```
 nsectetur adipiscing elit, ...  
 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 .. .. ..
```

# Segmentácia

Dáta sa rozkúskujú do častí, zvyčajne okolo 1kB.

Lorem ipsum dolor sit amet, co

00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29

nsectetur adipiscing elit, ...

30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 .. .. ..

# Segmentácia

Z rozkúskovaných dát sa vytvoria segmenty. Každý segment má číslo sekvencie, ktoré určuje poradie prvého bajtu v segmente.

TCP Segment, sequence number **00**

Data: **Lorem ipsu**

TCP Segment, sequence number **10**

Data: **m dolor si**

TCP Segment, sequence number **20**

Data: **t amet, co**

TCP Segment, sequence number **30**

Data: **nsectetur**

TCP Segment, sequence number **40**

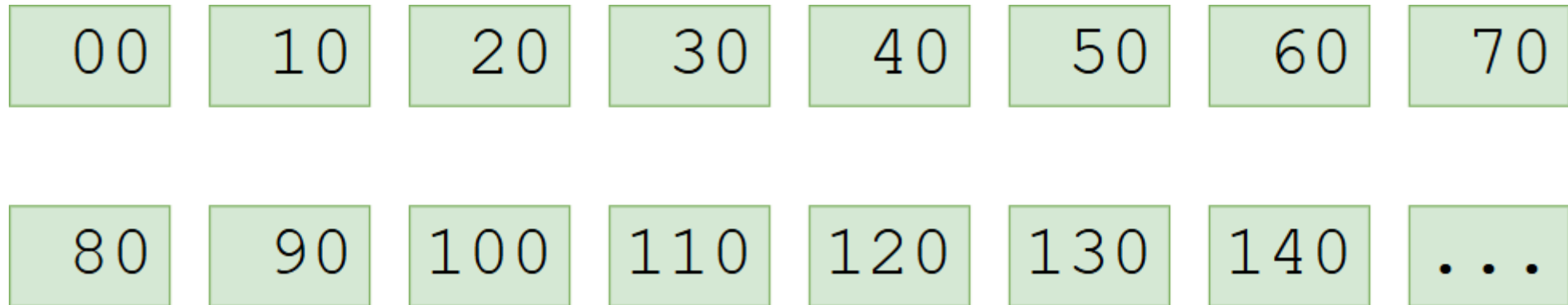
Data: **adipiscing**

TCP Segment, sequence number **50**

Data: **elit, ...**

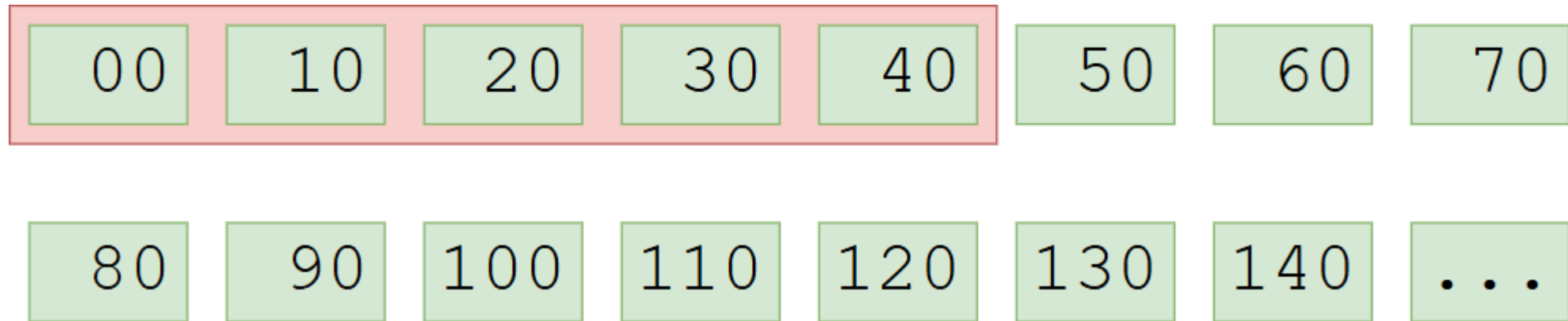
# Segmentácia

Prijímateľ musí každý segment potvrdiť ACK správou



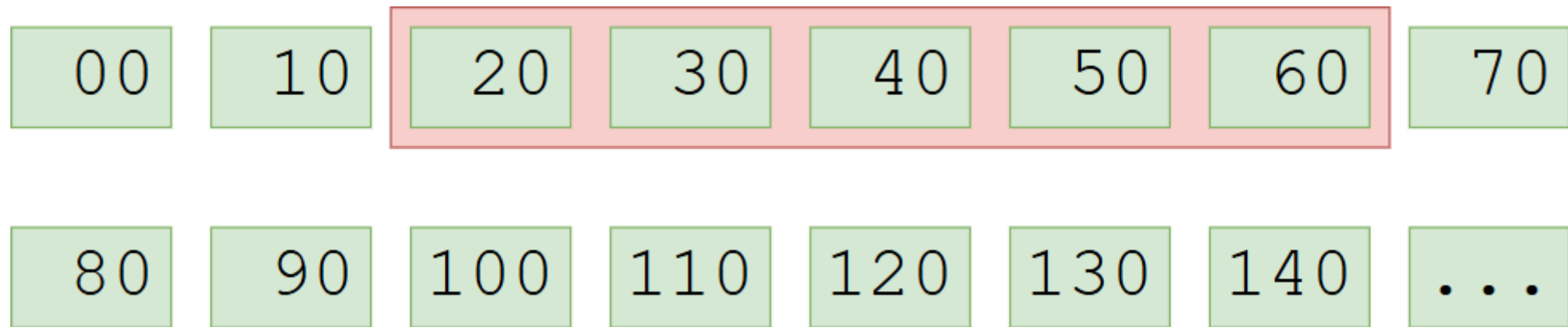
# Posuvné okno

Kvôli rýchlosti sa posiela viacero segmentov naraz - tzv. posuvné okno - **sliding window**



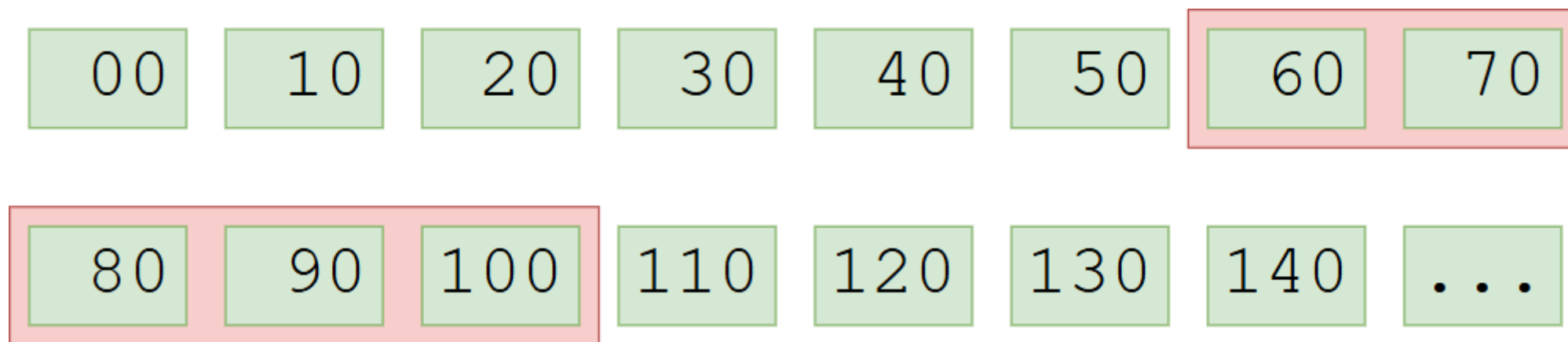
# Posuvné okno

Ak prijímateľ potvrdí prijatie segmentu, okno sa posunie



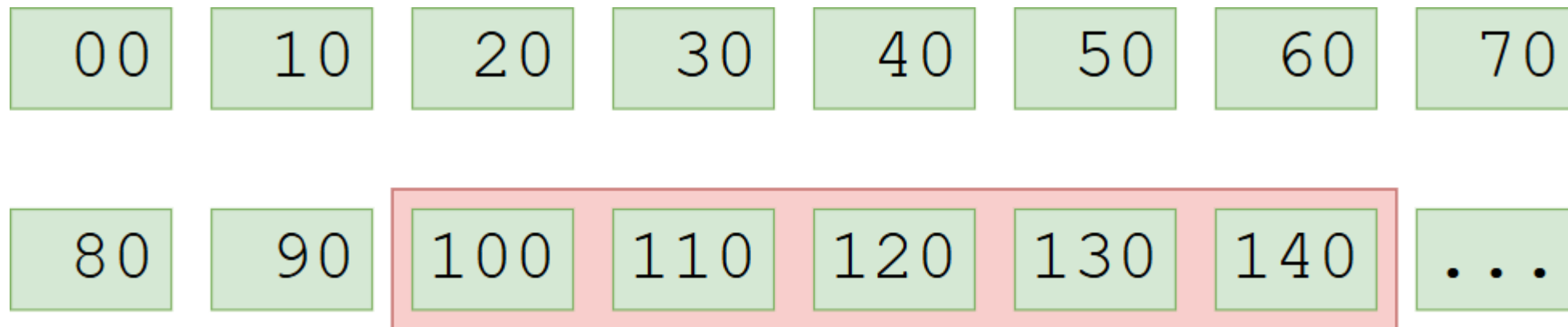
# Posuvné okno

Nové segmenty, ktoré sa dostanú do okna, sa pošlú



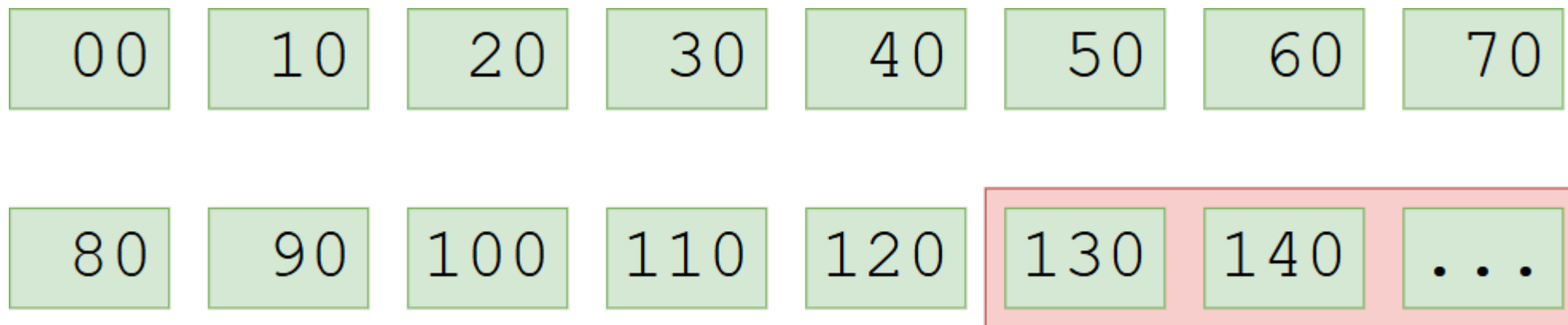
# Posuvné okno

Okno sa posúva naprieč segmentami



# Posuvné okno

Okno postupne prejde všetkými dátami, ktoré sa majú odoslať



# Retransmisia stratených segmentov

ACK má v sebe informáciu, ktoré bajty má prijímateľ úspešne prijaté.

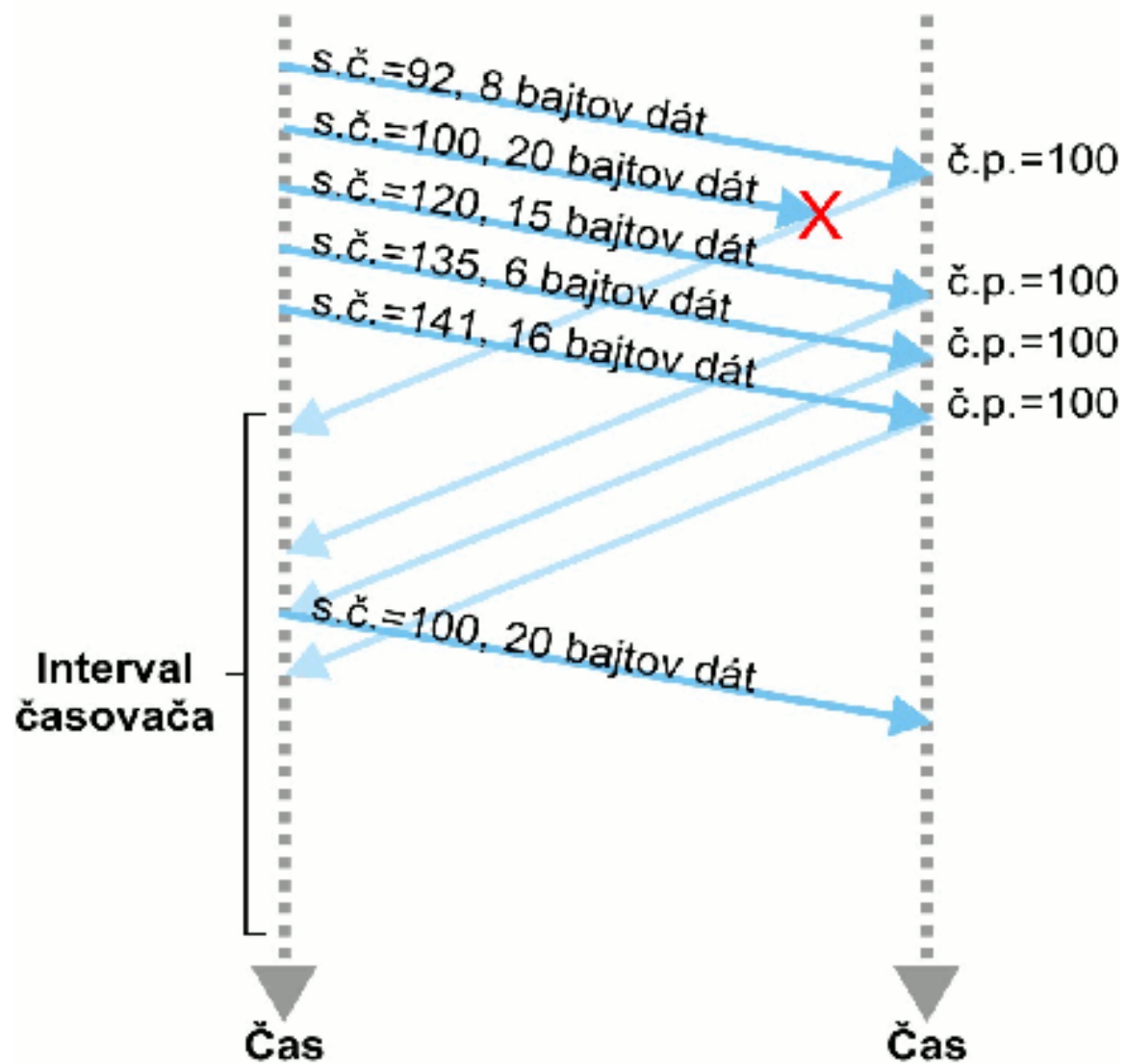
Pri strate TCP segmentu odosielateľ znova odošle pôvodný segment

Segment je stratený:

1. Ak neprišla ACK odpoveď do určitého času (**timeout**).
2. Ak prijímateľ poslal štyri rovnaké ACK odpovede (**tri duplikáty**).  
Chýba mu segment

Stanica A

Stanica B



# Flow Control - riadenie toku dát

Odosielateľ nemôže posielateľ dáta rýchlejšie, než ich dokáže príjemca spracovať

Príjemca v každom ACK segmente informuje, koľko bajtov je ešte schopný prijať

Flow control je tzv. **back pressure** - prijímateľ spomalí odosielateľa

Flow control riadi prijímateľ

*„Nenos mi viac tanierov, mám plný dres.“*

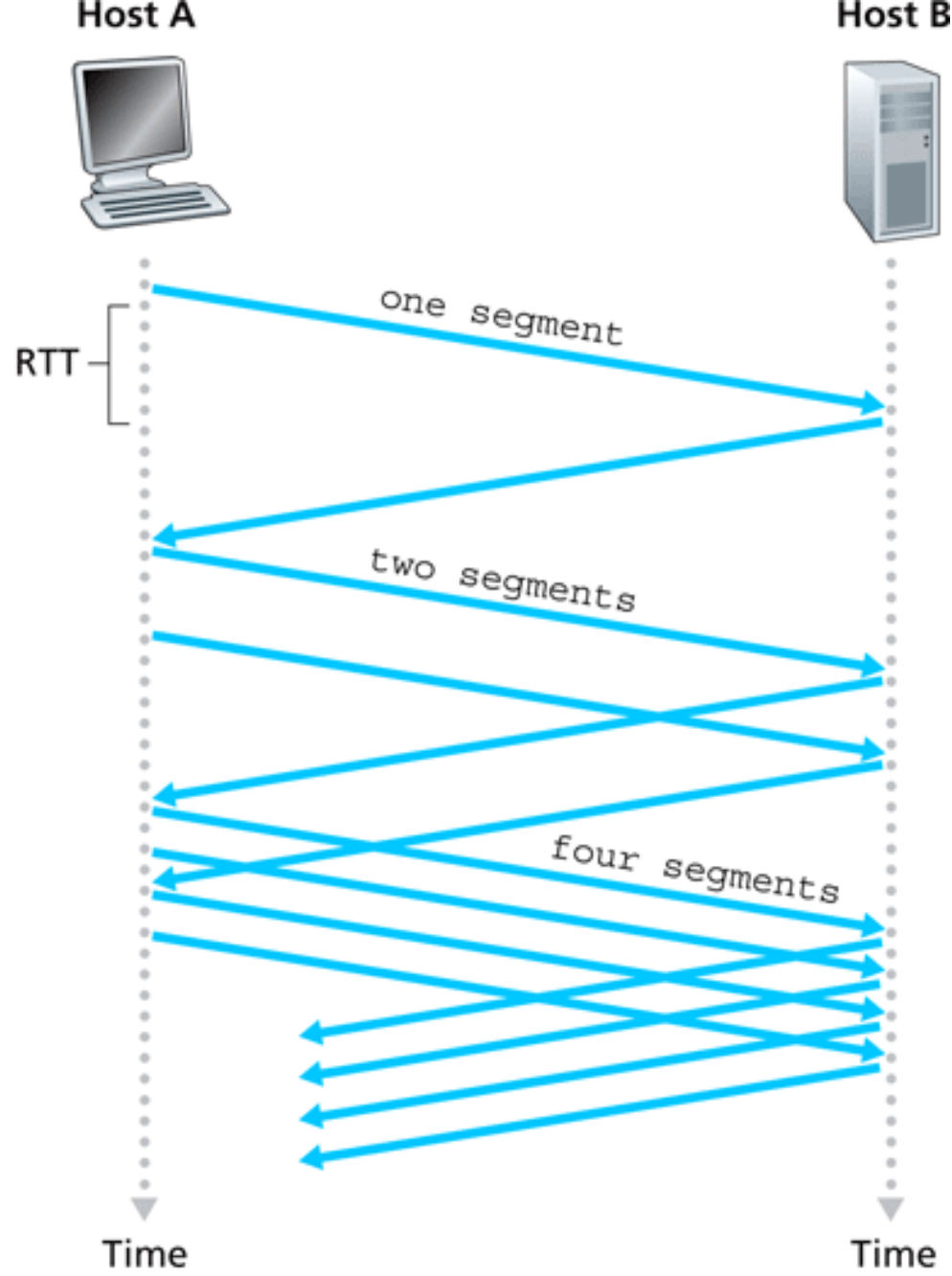
# Congestion Control - ochrana pred zahltením

O rýchlosti nerozhoduje iba prijímateľ, ale aj stav danej siete.

Dva hlavné režimy:

1. **Slow start** - okno je na začiatku malé, ale rastie exponenciálne
2. **Congestion Avoidance** - okno rastie lineárne a pri strate segmentu sa zníži na polovicu

Congestion control riadi odosielateľ



**Figure 3.52** ♦ TCP slow start

# TCP fast retransmit and fast recovery

