

**Računarske mreže**  
(2OER5O03)

# Kontrola toka

**Auditivne vežbe**



# 1

## Nivo veze podataka

funkcije, protokoli, iskorišćenost,  
zadaci

## Funkcije nivoa veze podataka (DDL – Data Link Level)

- Uokviravanje (*framing*)
- Adresiranje (*addressing*)
- Kontrola grešaka (*error control*)
- **Kontrola toka (*flow control*)**
- Tipovi servisa
  - ▷ Unacknowledged connectionless servis
  - ▷ Acknowledged connectionless servis
  - ▷ Acknowledged connection-oriented servis

# Kontrola toka

- Problem nastaje kada pošiljalac šalje brže frejmove nego što je primalac u stanju da ih primi. Da bi se sprečilo “pretrpavanje” primaoca i gubitak poruka uvodi se kontrola toka koja je organizovana kao:
  - povratna sprega (*feedback-based flow control*)
- Prijemnik vraća pošiljaocu informaciju šta je primio omogućujući da nastavi sa slanjem novi frejmova.

# Protokoli

## ■ Simplex

- ▷ Simplex protokol bez ograničenja
- ▷ Simplex Stop-and-Wait protokol
- ▷ Simplex protokol za kanal sa šumovima

## ■ Sliding-Window

- ▷ Povratak na N (Go-Back-N)
- ▷ Selektivna retransmisija (Selective reject, Selective repeat)

# Protokoli - zaglavlje

```
#define MAX_PKT 1024

typedef enum {false,true} boolean;

typedef unsigned int seq_nr;                                // redni ili ack-brojevi

typedef struct {unsigned char data[MAX_PKT];} packet;        // def. paketa

typedef enum {data,ack,nak} frame_kind;                     // def. tipa frejma

typedef struct {
    frame_kind    kind;           // tip frejma
    seq_nr        seq;           // redni broj
    seq_nr        ack;           // ack-broj
    packet         info;         // paket sa nivoa mreže
} frame;

void wait_for_event(event_type *event); // f-ja čeka sledeći događaj i vraća njegov tip
void from_network_layer(packet *p);    // f-ja uzima paket sa mrežnog sloja
```

# Protokoli - zaglavlje

```
//f-ja šalje podatke iz okvira mrežnom sloju
void to_network_layer(packet *p);
//f-ja uzima frejm sa fizičkog nivoa i kopira ga u r
void from_physical_layer(frame *r);
//f-ja prosleđuje frejm ka fizičkom nivou radi prenosa
void to_physical_layer(frame *s);
//f-ja startuje odbrojavanje kloka i omogućava timeout događaj.
void start_timer(seq_nr k);
//f-ja startuje pomoćni brojač i omogućava ack_timeout događaj.
void start_ack_timer(void);
//f-ja zaustavlja pomoćni brojač i sprečava ack_timeout događaj.
void stop_ack_timer(void);
//f-ja dozvoljava mrežnom sloju da izazove network_layer_ready događaj.
void enable_network_layer(void);
// makro-naredba za inkrementaciju k po modulu MAX_SEQ
#define inc(k) if (k<MAX_SEQ) k=k+1; else k=0
// Redni brojevi frejmova se kreću od 0 do MAX_SEQ, MAX_SEQ je različito za različite
protokole!.
```

# Simplex protokol bez ograničenja ( Protokol 1 )

■ Najprostiji (i najidealniji) tip protokola. Podrazumeva:

- prenos se obavlja samo u jednom smeru,
- mrežni slojevi i na prijemnoj i na predajnoj strani su uvek spremni,
- vreme obrade se zanemaruje,
- raspoloživi su baferi neograničenog kapaciteta,
- komunikacioni kanal nikada ne oštećuje i ne gubi frejmove.



# Protokol 1 - kod

```
typedef enum {frame_arrival} event_type;  
#include "protocol.h"
```

```
void sender1(void)  
{  
    frame s;  
    packet buffer;  
    while (true) {  
        from_network_layer(&buffer);  
        s.info=buffer;  
        to_physical_layer(&s);  
    }  
}
```

```
void receiver1(void)  
{  
    frame r;  
    event_type event;  
    while(true) {  
        wait_for_event(&event);  
        from_physical_layer(&r);  
        to_network_layer(&r.info);  
    }  
}
```

## Simplex Stop-and-Wait protokol ( Protokol 2 )

- Za razliku od prethodnog slučaja ovde se ne podrazumevaju sledeće stvari:
  - neograničeno brza obrade podataka i
  - neograničeni baferski prostor.

# Protokol 2 - kod

```
typedef enum {frame_arrival} event_type;  
#include "protocol.h"
```

```
void sender2(void)  
{  
    frame s;  
    packet buffer;  
    event_type event;  
  
    while (true) {  
        from_network_layer(&buffer);  
        s.info=buffer;  
        to_physical_layer(&s);  
        wait_for_event(&event);  
    }  
}
```

```
void receiver2(void)  
{  
    frame r, s;  
    event_type event;  
  
    while(true) {  
        wait_for_event(&event);  
        from_physical_layer(&r);  
        to_network_layer(&r.info);  
        //slanje posebnog frejma da bi se  
        //probudio pošiljalac  
        to_physical_layer(&s);  
    }  
}
```

## Simplex za kanal sa šumovima ( Protokol 3 )

Ovaj protokol podrazumeva da kanal može da izazove grešku. Frejmovi mogu biti oštećeni ili potpuno izgubljeni.

```
#define MAX_SEQ 1 // ovaj prtokol broji frejmove po modulu 2
typedef enum {frame_arrival, cksum_err, timeout} event_type;
#include "protocol.h"
```

```
void sender3(void) {
    seq_nr next_frame_to_send;
    frame s;
    packet buffer;
    event_type event;
    next_frame_to_send = 0;
    from_network_layer(&buffer);
    while (true) {
        s.info = buffer;
        s.seq = next_frame_to_send;
        to_physical_layer(&s);
        start_timer(s.seq);
        wait_for_event(&event);
        if(event == frame_arrival) {
            from_physical_layer(&s);
            if (s.ack == next_frame_to_send) {
                stop_timer(s.ack);
                from_network_layer(&buffer);
                inc(next_frame_to_send);
            }
        }
    }
}
```

```
void receiver3(void)
{
    seq_nr frame_expected;
    frame r, s;
    event_type event;
    frame_expected = 0;

    while(true) {
        wait_for_event(&event);
        if (event == frame_arrival) {
            from_physical_layer(&r);
            if (r.seq == frame_expected) {
                to_network_layer(&r.info);
                inc(frame_expected);
            }
            s.ack=1-frame_expected;
            to_physical_layer(&s);
        }
    }
}
```

## Numerisanje poruka ( Protokol 4 )

Ovaj protokol je bidirekcion ( omogućava slanje podataka u oba smera ).

```
#define MAX_SEQ 1
typedef enum {frame_arrival, cksum_err, timeout}
event_type;
#include "protocol.h"

void protocol4(void)
{
    seq_nr next_frame_to_send;
    seq_nr frame_expected;
    frame r, s;
    packet buffer;
    event_type event;
    next_frame_to_send = 0;
    frame_expected = 0;
    from_network_layer(&buffer);
    s.info=buffer;
    s.seq=next_frame_to_send;
    s.ack=1-frame_expected;
    to_physical_layer(&s);
    start_timer(s.seq);
```

```
while (true) {
    wait_for_event(&event);
    if(event==frame_arrival) {
        from_physical_layer(&s);
        if (r.seq==frame_expected) {
            to_network_layer(&r.info);
            inc(frame_expected);
        }
        if (r.ack==next_frame_to_send) {
            from_network_layer(&buffer);
            inc(next_frame_to_send);
        }
    }
    s.info=buffer;
    s.seq=next_frame_to_send;
    s.ack=1-frame_expected;
    to_physical_layer(&s);
    start_timer(s.seq);
}
```

## Sliding-Window (Protokol 5)

- Dozvoljava da se pošalje više frejmova pre nego što se neki od njih potvrdi
- Uvode se novi pojmovi:
  - ▷ prozor na predajnoj strani
  - ▷ bafer na predajnoj strani
  - ▷ prozor na prijemnoj strani
  - ▷ bafer na prijemnoj strani
  - ▷ numerisanje poruka

## Sliding-Window parametri

- **Prozor na predajnoj strani** – opseg frejmova koji mogu biti poslani a nepotvrđeni
- **Bafer na predajnoj strani** – čuva poruke koje su poslate a nepotvrđene
- **Prozor na prijemnoj strani** – opseg poruka koje se očekuje
- **Bafer na prijemnoj strani** – čuva poruke koje su stigle preko reda

## Protokol 5 - kod

Ovaj protokol dozvoljava pošiljaocu da isporuči više frejmova (do MAX\_SEQ), bez čekanja na ack.

Za razliku od prethodnih protokola, ovde se ne podrazumeva da mrežni sloj uvek ima spreman paket. Zato se uvodi događaj `network_layer_ready`, koji označava kada je paket spreman za slanje.

```
#define MAX_SEQ 7 // Treba da bude 2n-1
typedef enum {frame_arrival, cksum_err, timeout,
network_layer_ready} ev_type;
#include "protocol.h"
static boolean between(seq_nr a, seq_nr b, seq_nr c)
{
    if (((a<=b) && (b<c)) || ((c<a) && (a<=b))
|| ((b<c) && (c<a))) return (true);
    else return (false);
}
static void send_data(seq_nr frame_nr, seq_nr
frame_expected, packet buffer[])
{
    frame s;
    s.info = buffer[frame_nr];
    s.seq = frame_nr;
    s.ack = (frame_expected + MAX_SEQ) % (MAX_SEQ + 1);
    to_physical_layer (&s);
    start_timer (frame_nr);
}
```

```
void protokol5(void)
{
    seq_nr nex_frame_to_send;
    seq_nr ack_expected; //najstariji a još ne potvrđeni frejm
    frame f;
    packetbuffer[MAX_SEQ+1]; //baferi za izlazne frejmove
    seq_nr nbuffered; //br. bafera trenutno zauzetih
    seq_nr i; //za indeksiranje u okviru niza bafera
    ev_type event;
    enable_network_layer(); //dozvoljava događaj network_layer_ready
    ack_expected=0;
    next_frame_to_send = 0;
    frame_expected = 0;
    nbuffered=0;
    nbuffered=0;
    while (true) {
        wait_for_event(&event);
```



## Protokol 5 - kod

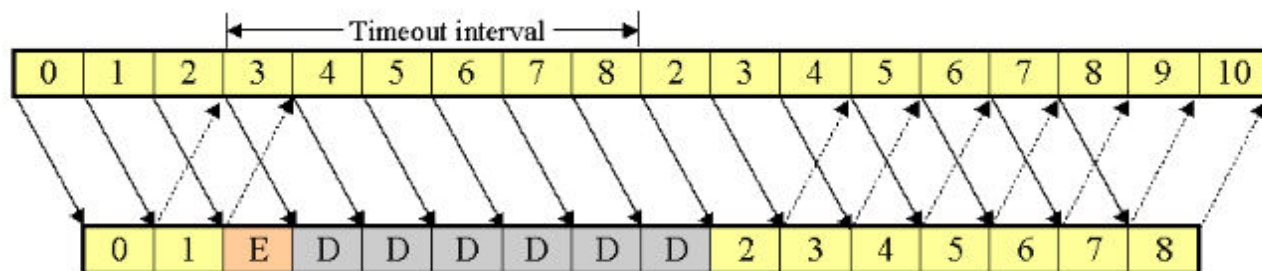
Za razliku od prethodnih protokola, ovde se ne podrazumeva da mrežni sloj uvek ima spreman paket. Zato se uvodi događaj `network_layer_ready`, koji označava kada je paket spreman za slanje.

```
switch (event) {
    case network_layer_ready: //mrežni sloj ima paket za slanje
        /* Prihvati, zapamti i pošalji novi frejm */
        from_network_layer(&buffer[next_frame_to_send]);
        nbuffered=nbuffered + 1;
        send_data (next_frame_to_send, frame_expected, buffer);
        inc (next_frame_to_send);
    break;
    case frame_arrival: //data ili control frejm je stigao
        from_physical_layer (&r);
        // frejmovi se prihvataju samo ako stignu po redu
        if (r.seq==frame_expected) {
            to_network_layer (&r.info);
            inc (frame_expected);
        }

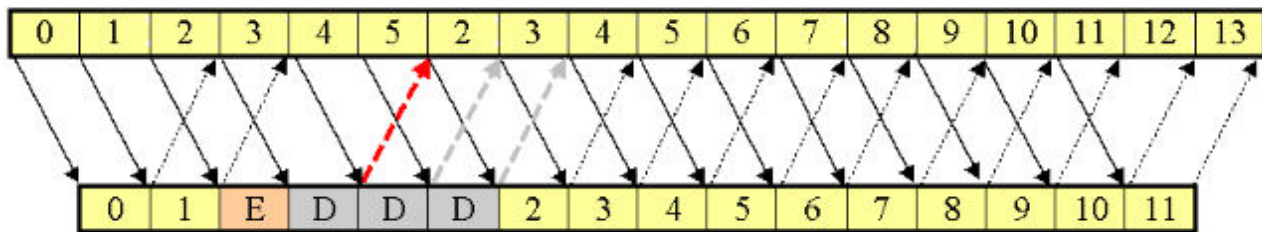
        // Ack za N podrazumeva da su primljeni i N-1, N-2 itd.
        while (between(ack_expected, r.ack, next_frame_to_send)) {
            nbuffered= nbuffered-1;
            stop_timer (ack_expected);
            inc (ack_expected);
        }
    break;
    case cksum_err: break; //loši frejmovi se ignorišu
```

```
        case timeout: // !?!. Ponovo poslati sve frejmove iz bafera
            next_frame_to_send=ack_expected;
            for (i=1; i<=nbuffered; i++) {
                send_data (next_frame_to_send, frame_expected, buffer);
                inc (next_frame_to_send);
            }
            if (nbuffered < MAX_SEQ) enable_network_layer ();
            else disable_network_layer ();
            break;
    } //end switch
} //end protokol5
```

## Primeri



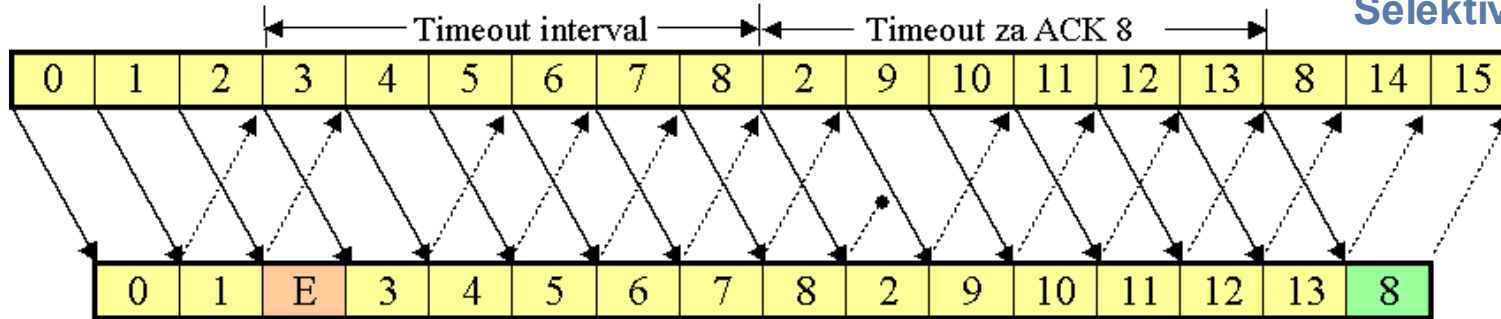
Povratak na N bez NAK poruke



Povratak na N sa NAK porukom

# Primeri

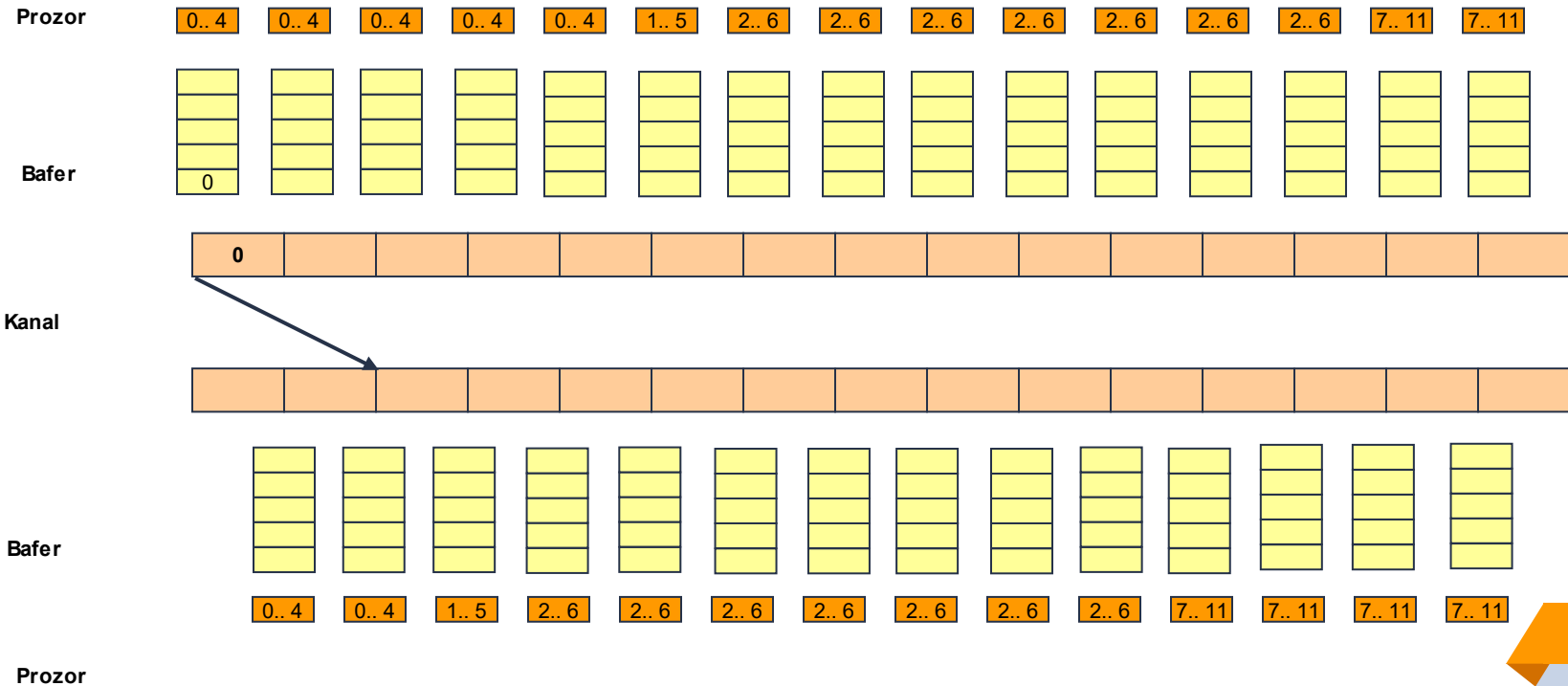
## Selektivna retransmisija



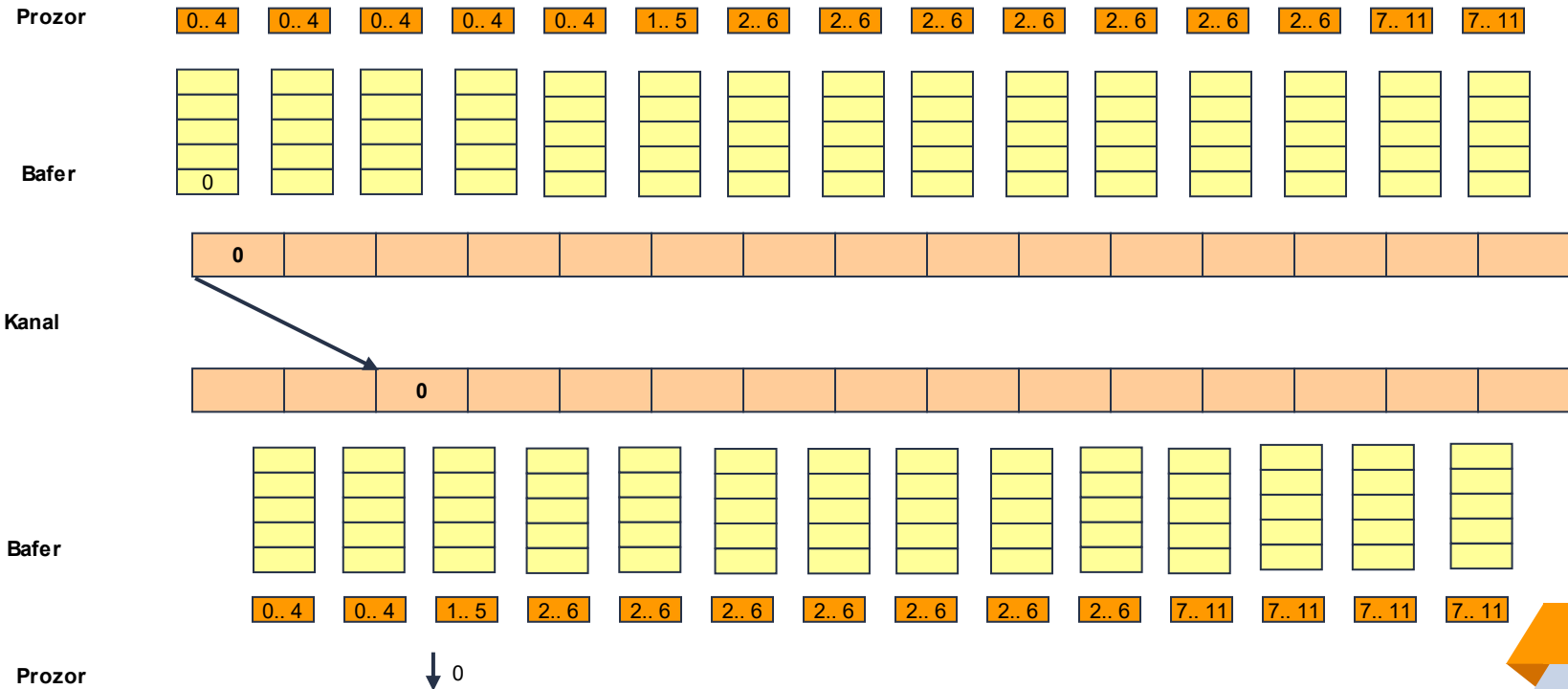
Selektivna retransmisija može da funkcioniše na:

- **data-link** nivou – obzirom da frejmovi putuju istom trasom (medijumom) ne postoji “preticanje”, pa se nakon pristizanja ACK sa višim rednim br. odmah može početi retransmisija prethodnih frejmova
- **transportnom** nivou – ukoliko je ispod mrežni nivo “sa skretnim paketima”, svaki paket može putovati zasebnom trasom, pa se čeka istek time-out-a pre početka retransmisije

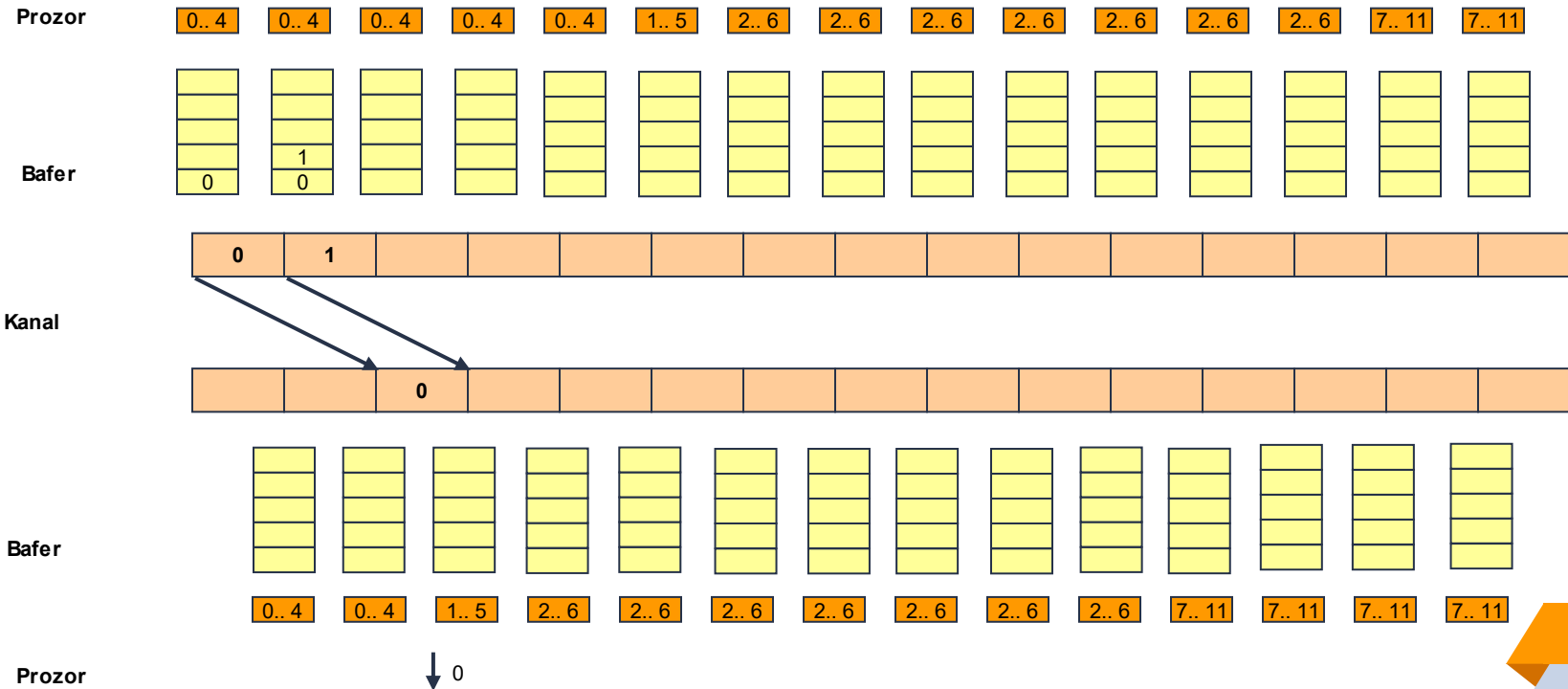
# Primer - Selektivna retransmisija sa gubitkom Info(2) frejma



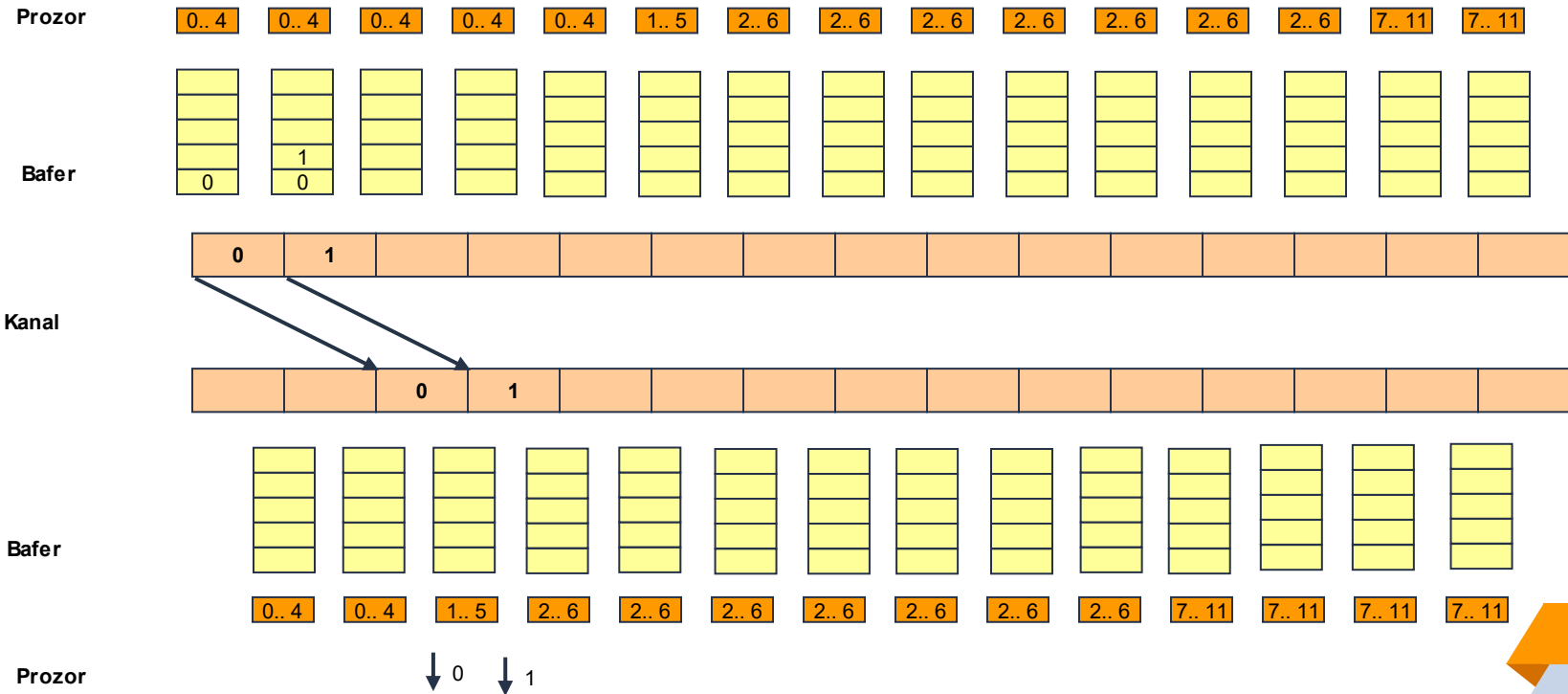
# Primer - Selektivna retransmisija sa gubitkom Info(2) frejma



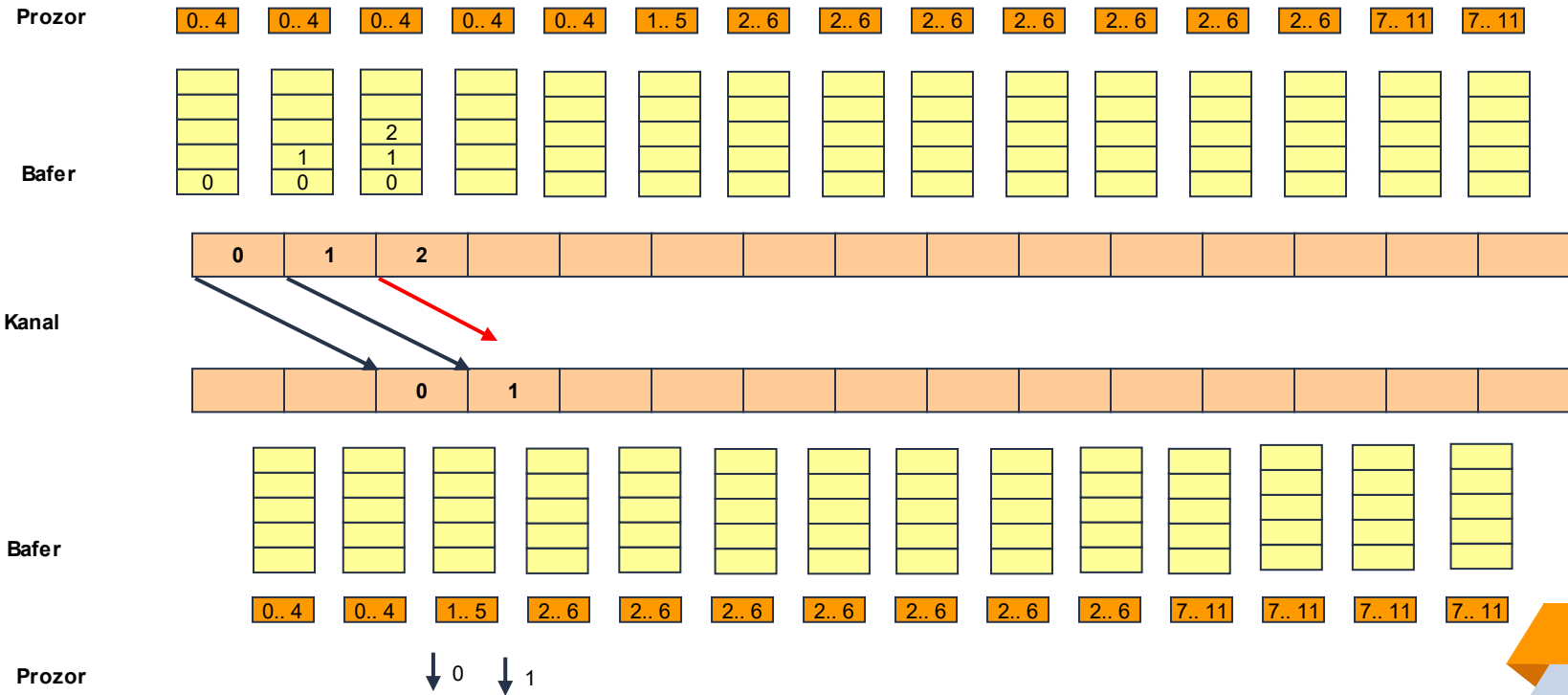
# Primer - Selektivna retransmisija sa gubitkom Info(2) frejma



# Primer - Selektivna retransmisija sa gubitkom Info(2) frejma

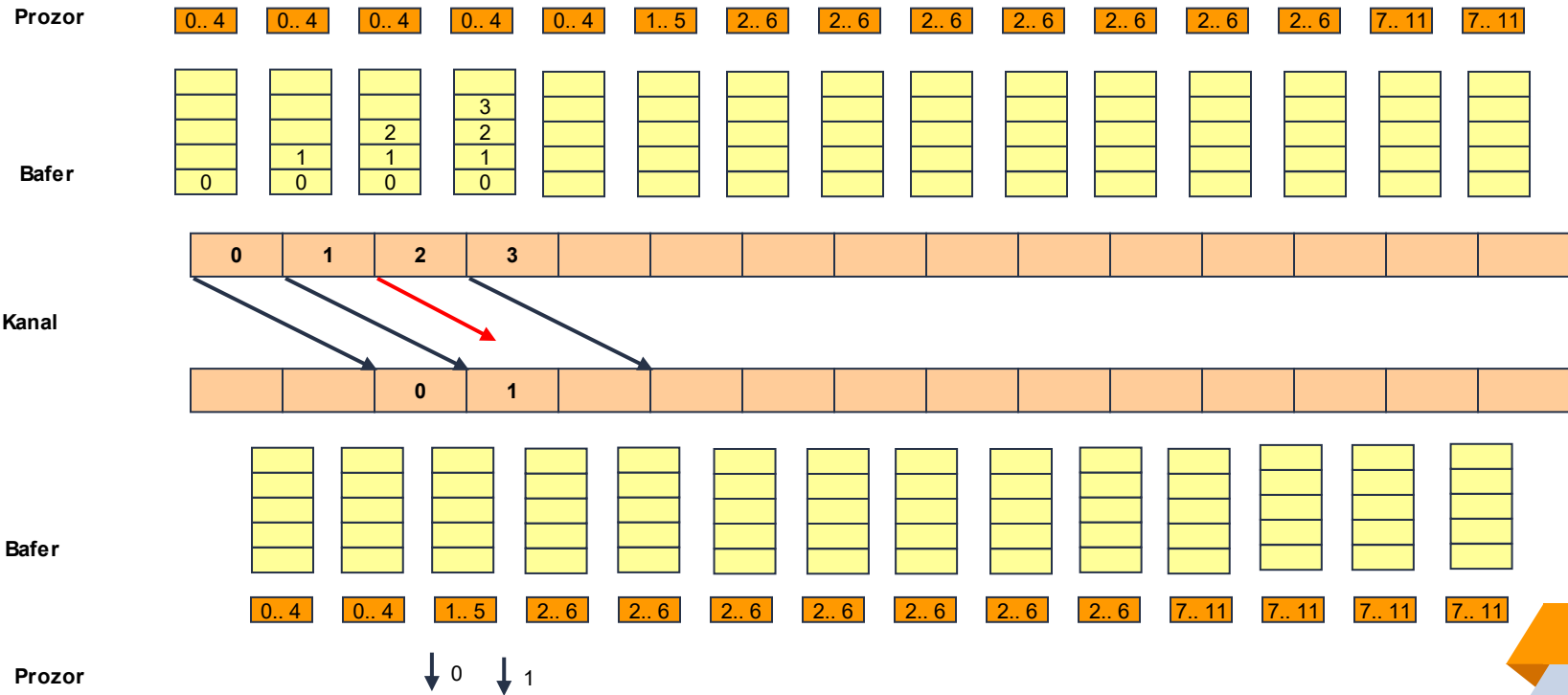


# Primer - Selektivna retransmisija sa gubitkom Info(2) frejma

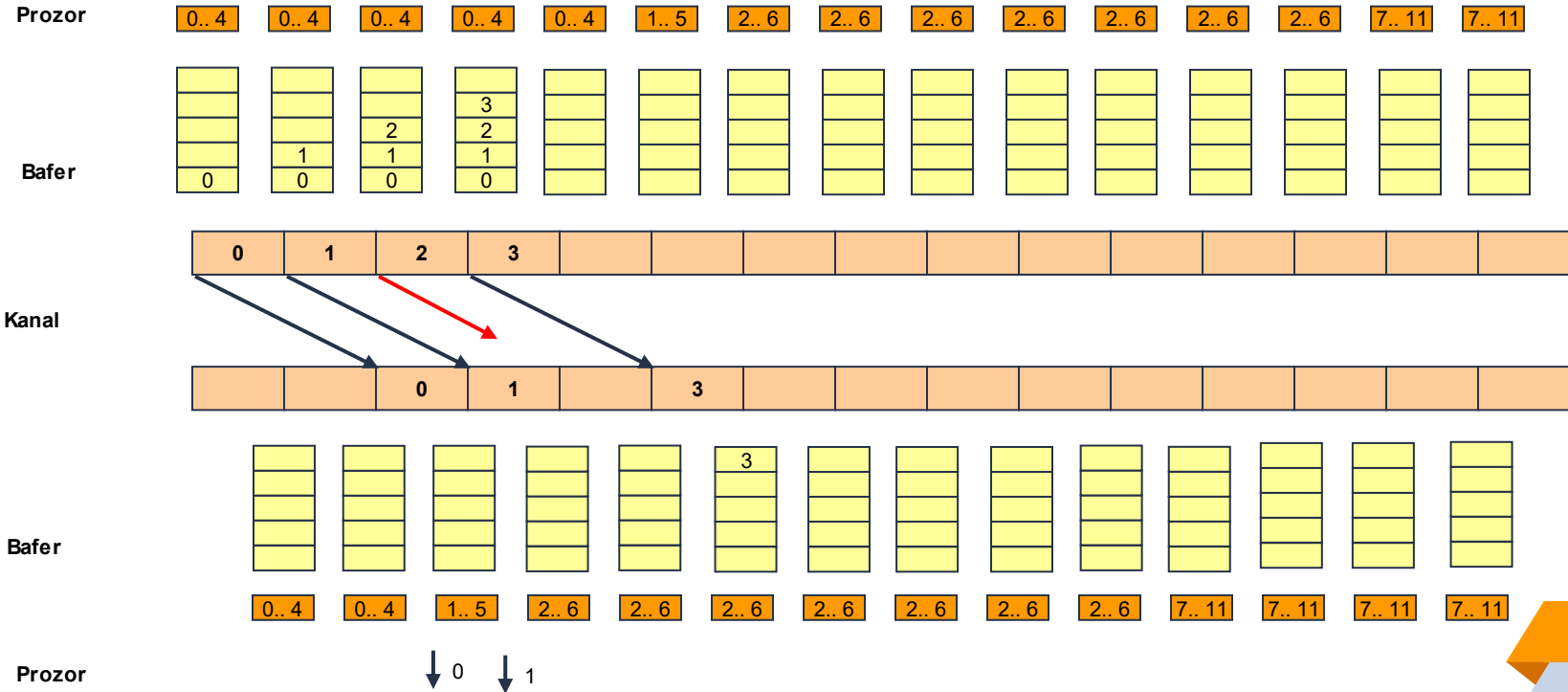




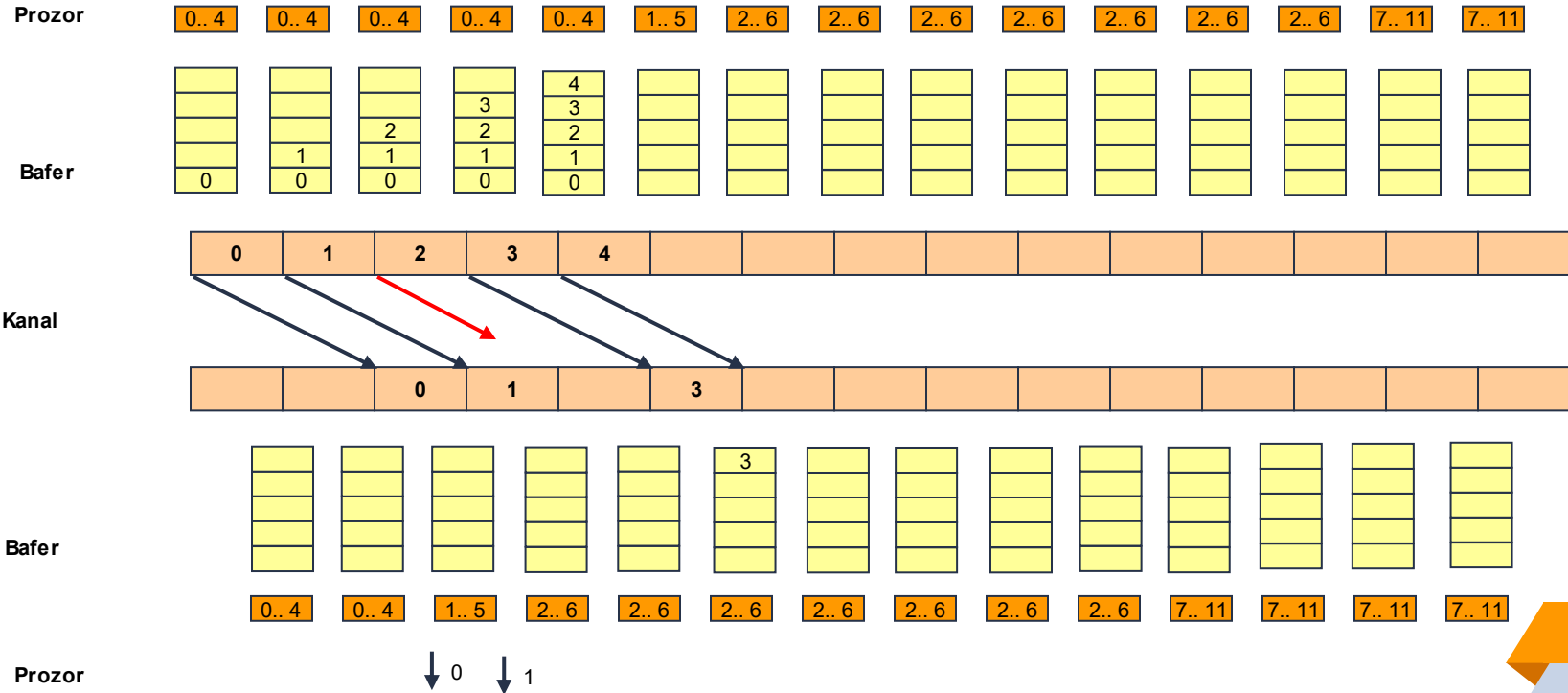
# Primer - Selektivna retransmisija sa gubitkom Info(2) frejma



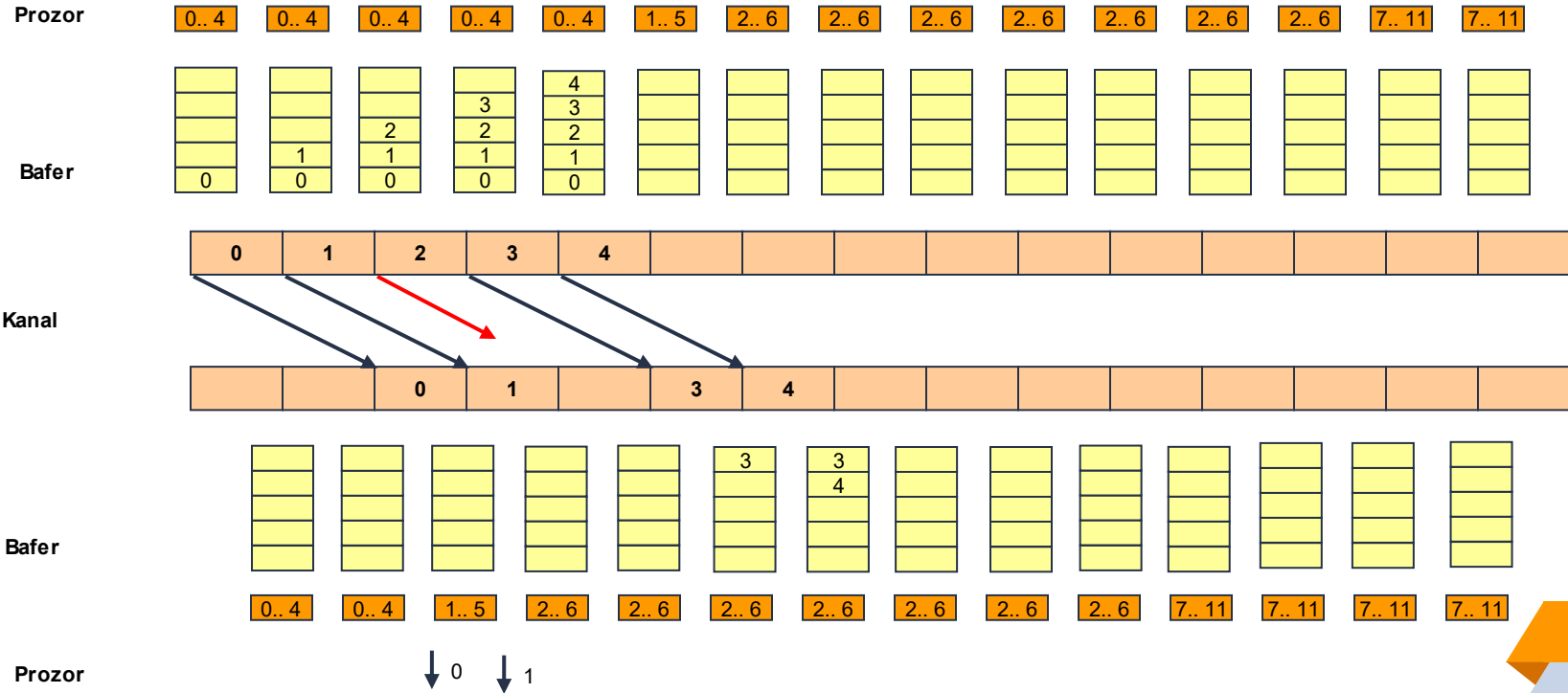
# Primer - Selektivna retransmisija sa gubitkom Info(2) frejma



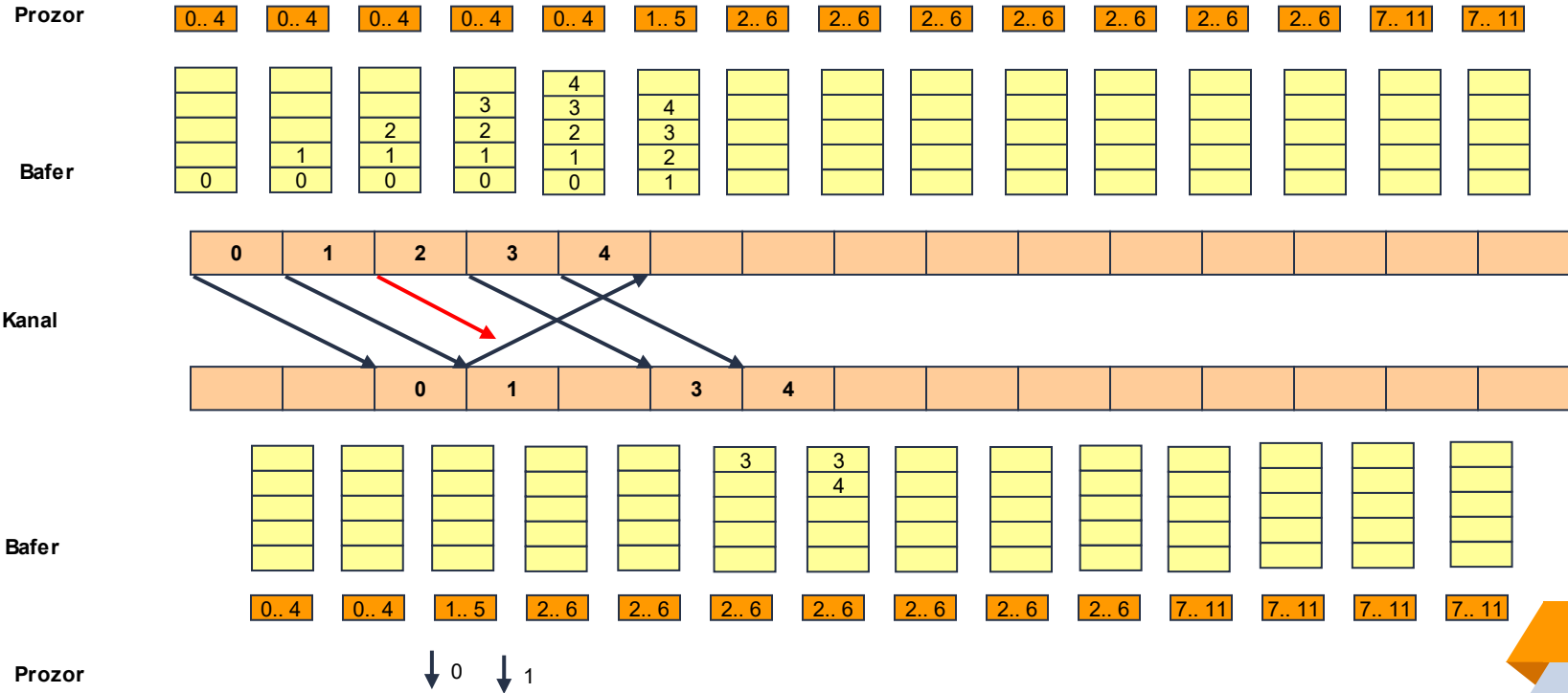
# Primer - Selektivna retransmisija sa gubitkom Info(2) frejma



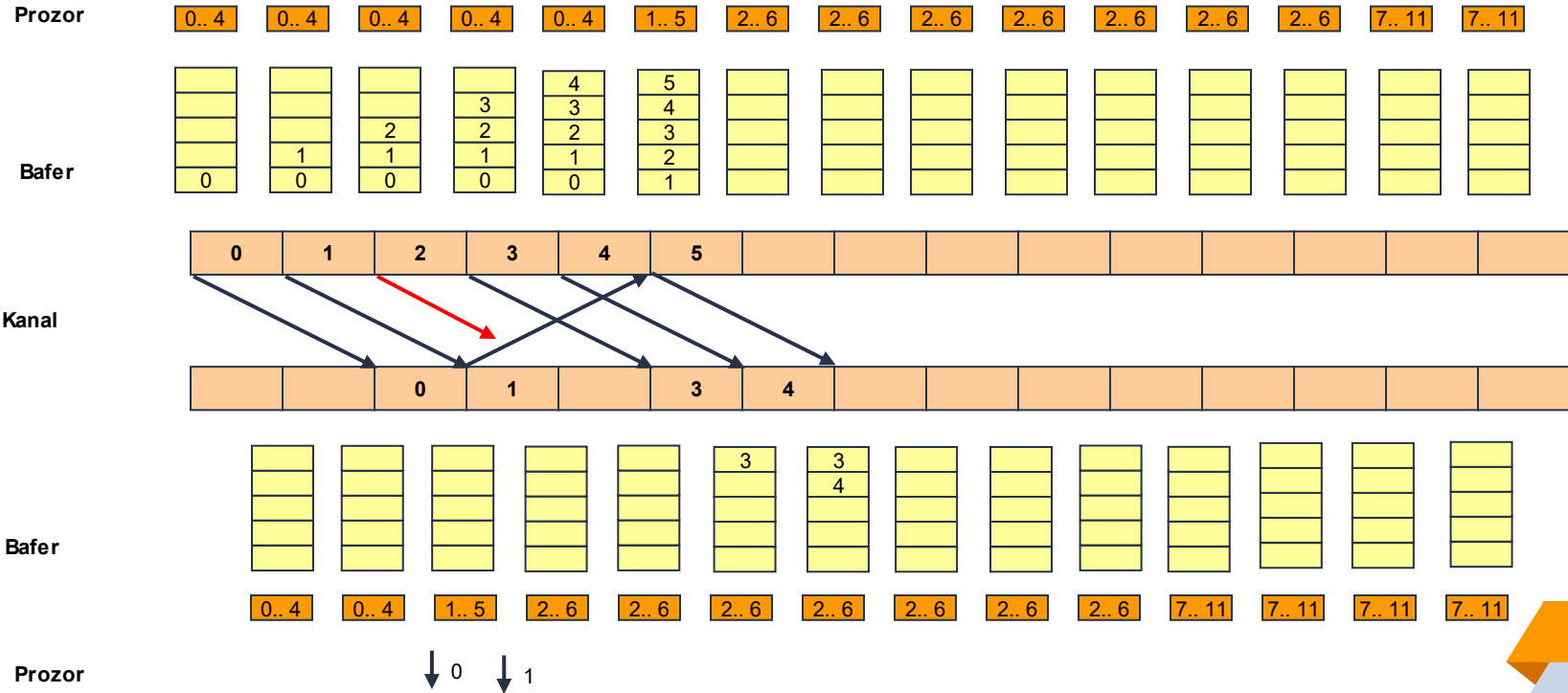
# Primer - Selektivna retransmisija sa gubitkom Info(2) frejma



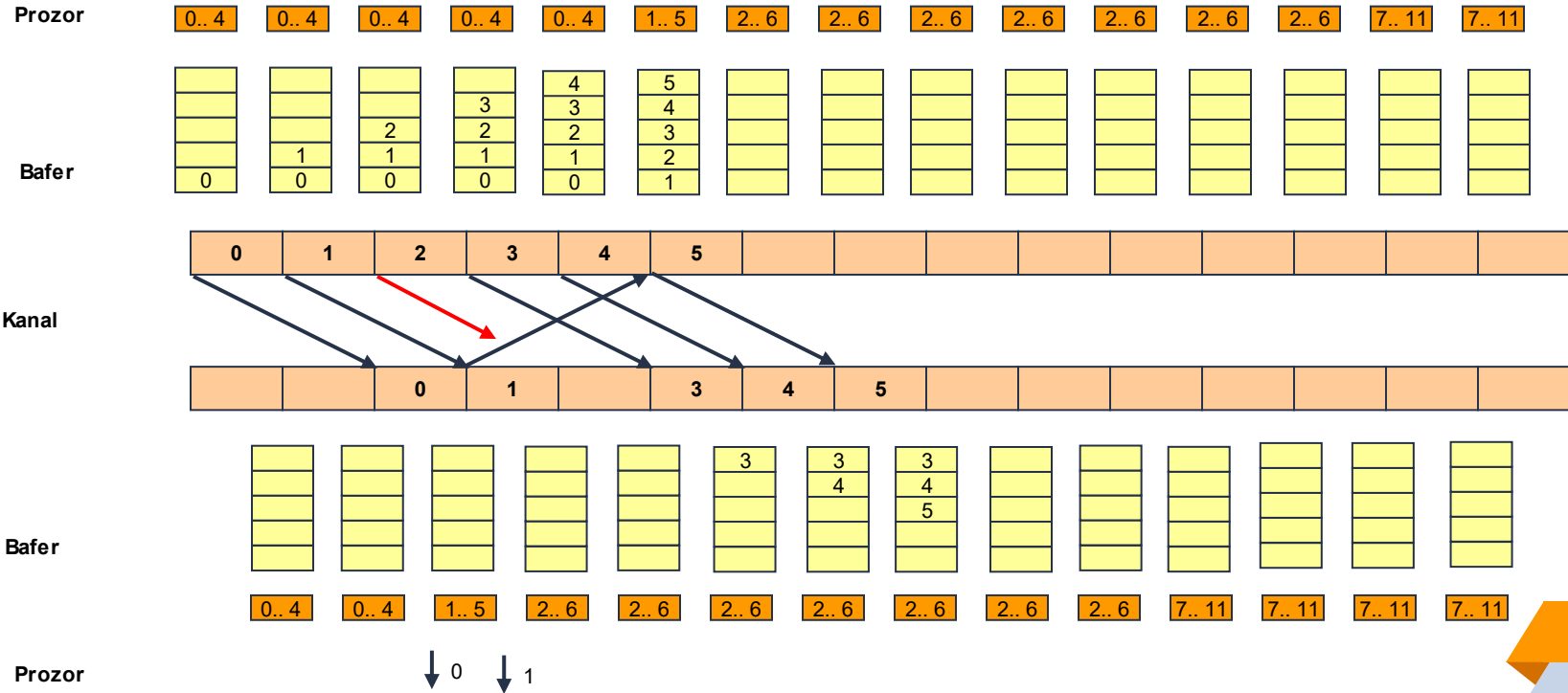
# Primer - Selektivna retransmisija sa gubitkom Info(2) frejma



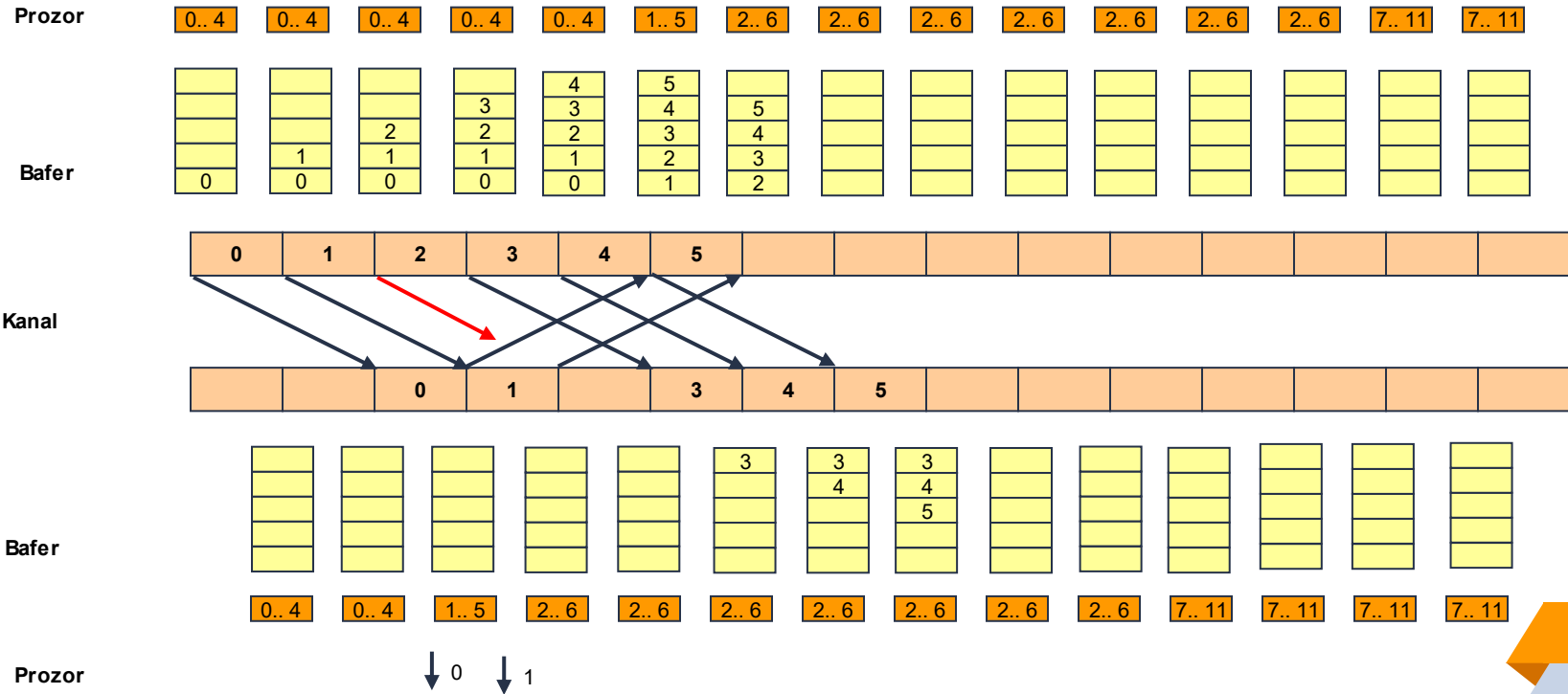
# Primer - Selektivna retransmisija sa gubitkom Info(2) frejma



# Primer - Selektivna retransmisija sa gubitkom Info(2) frejma

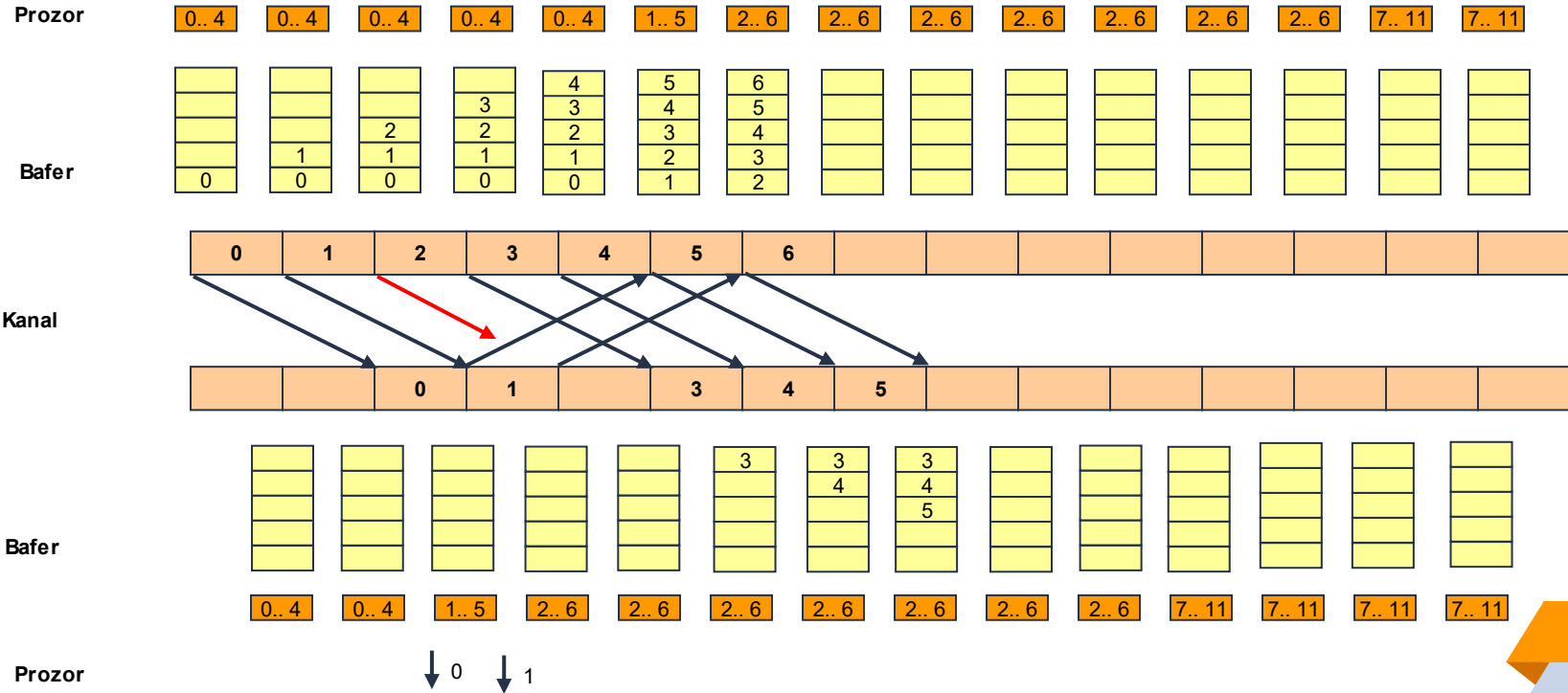


# Primer - Selektivna retransmisija sa gubitkom Info(2) frejma

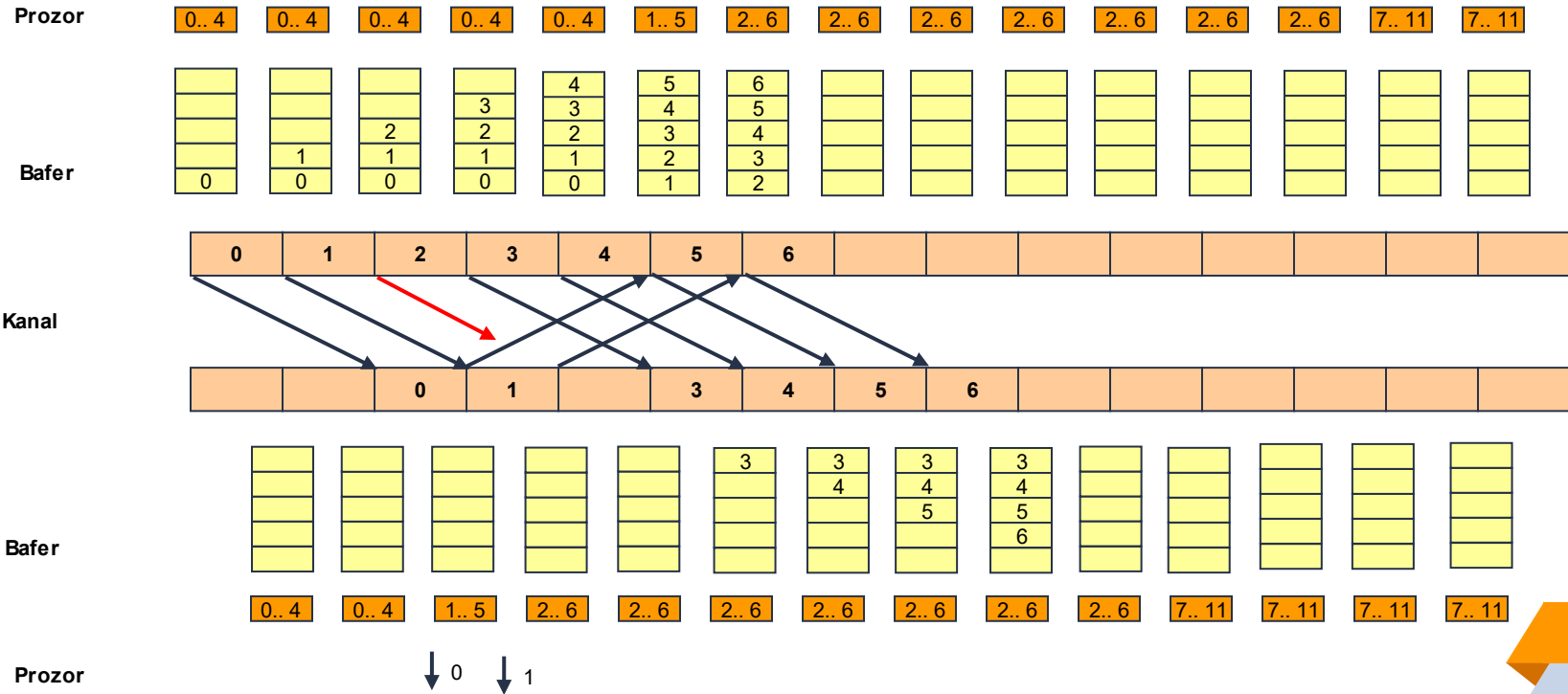




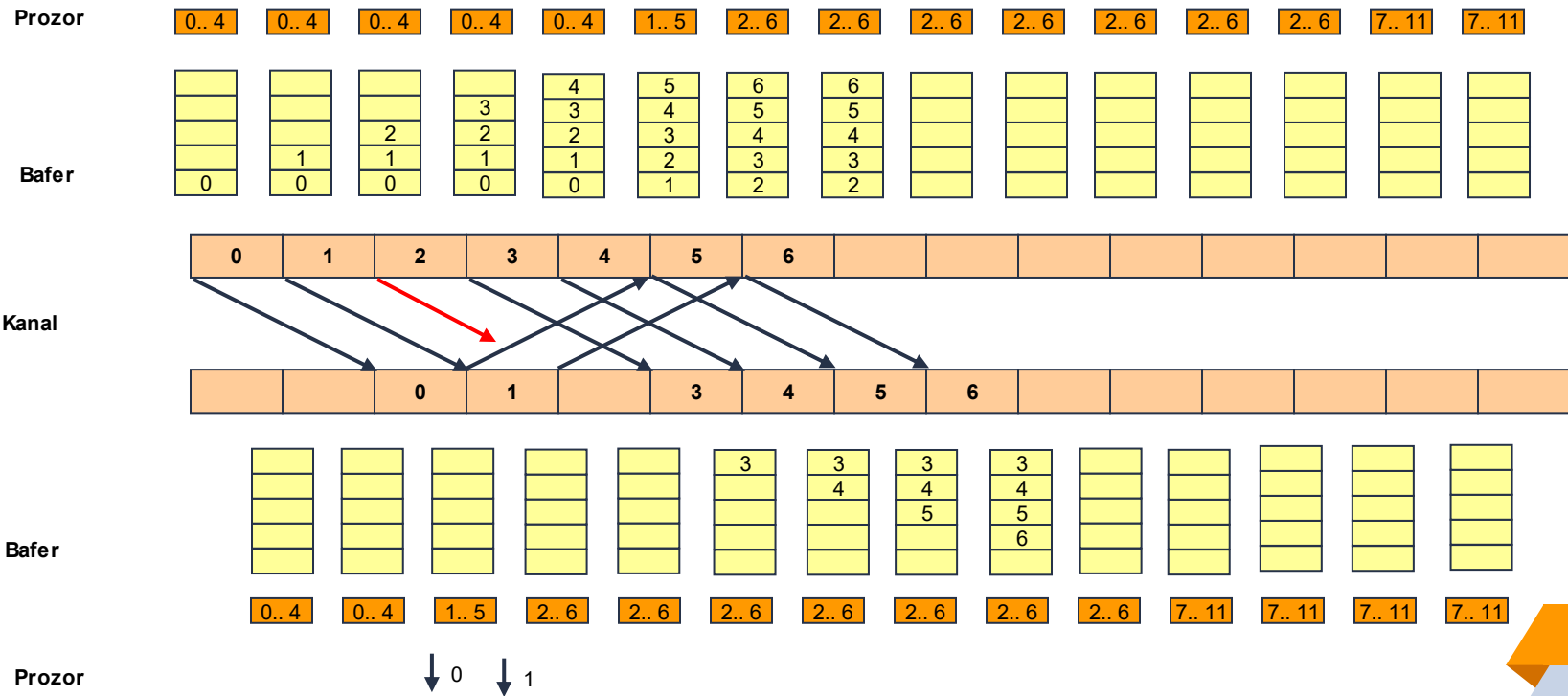
# Primer - Selektivna retransmisija sa gubitkom Info(2) frejma



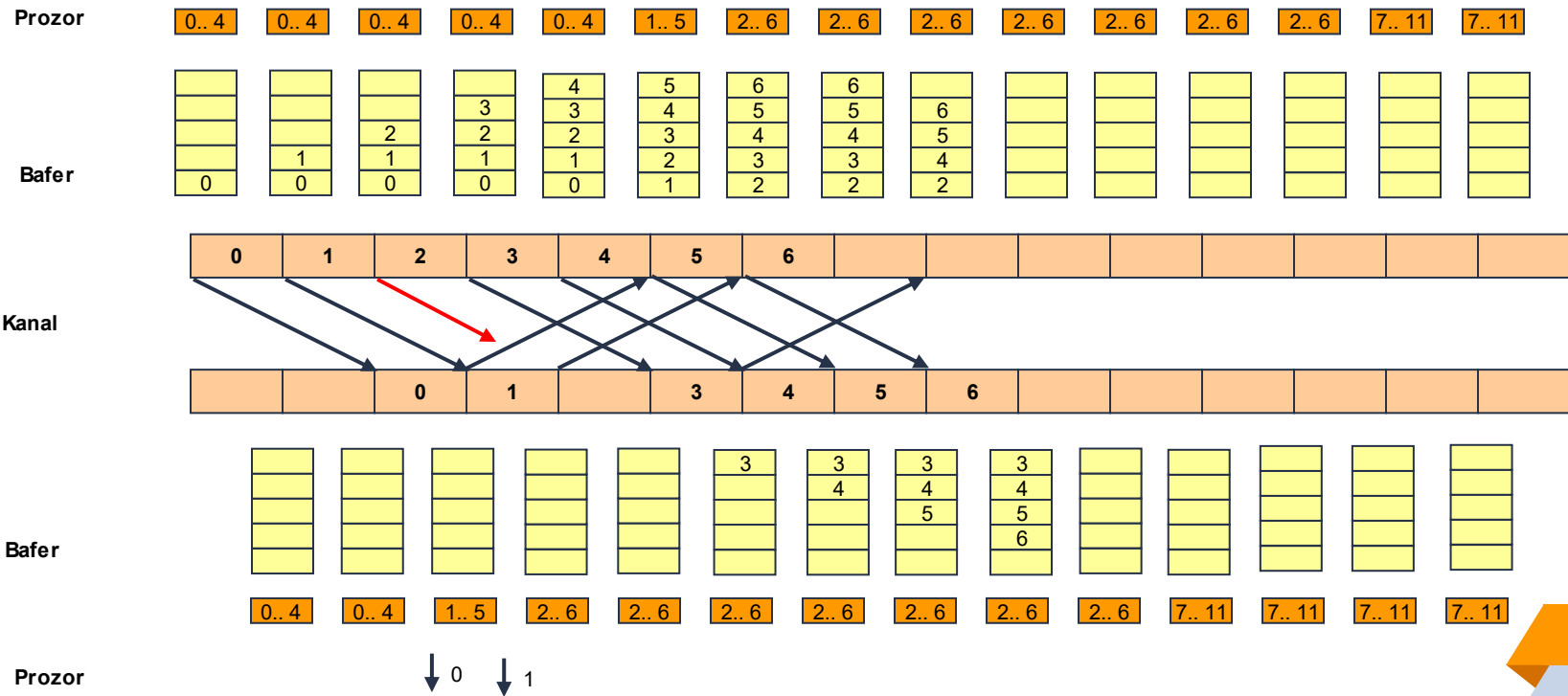
# Primer - Selektivna retransmisija sa gubitkom Info(2) frejma



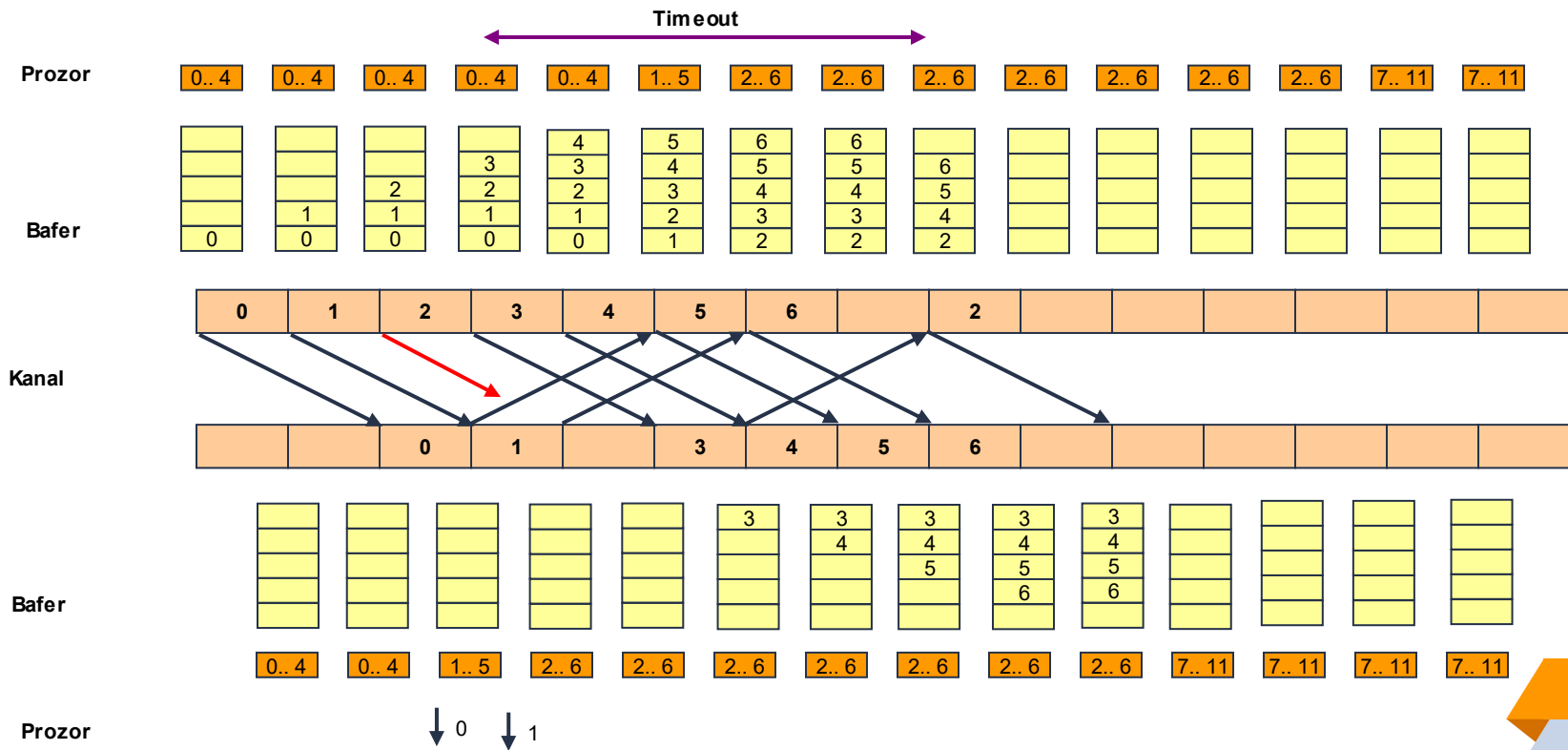
# Primer - Selektivna retransmisija sa gubitkom Info(2) frejma



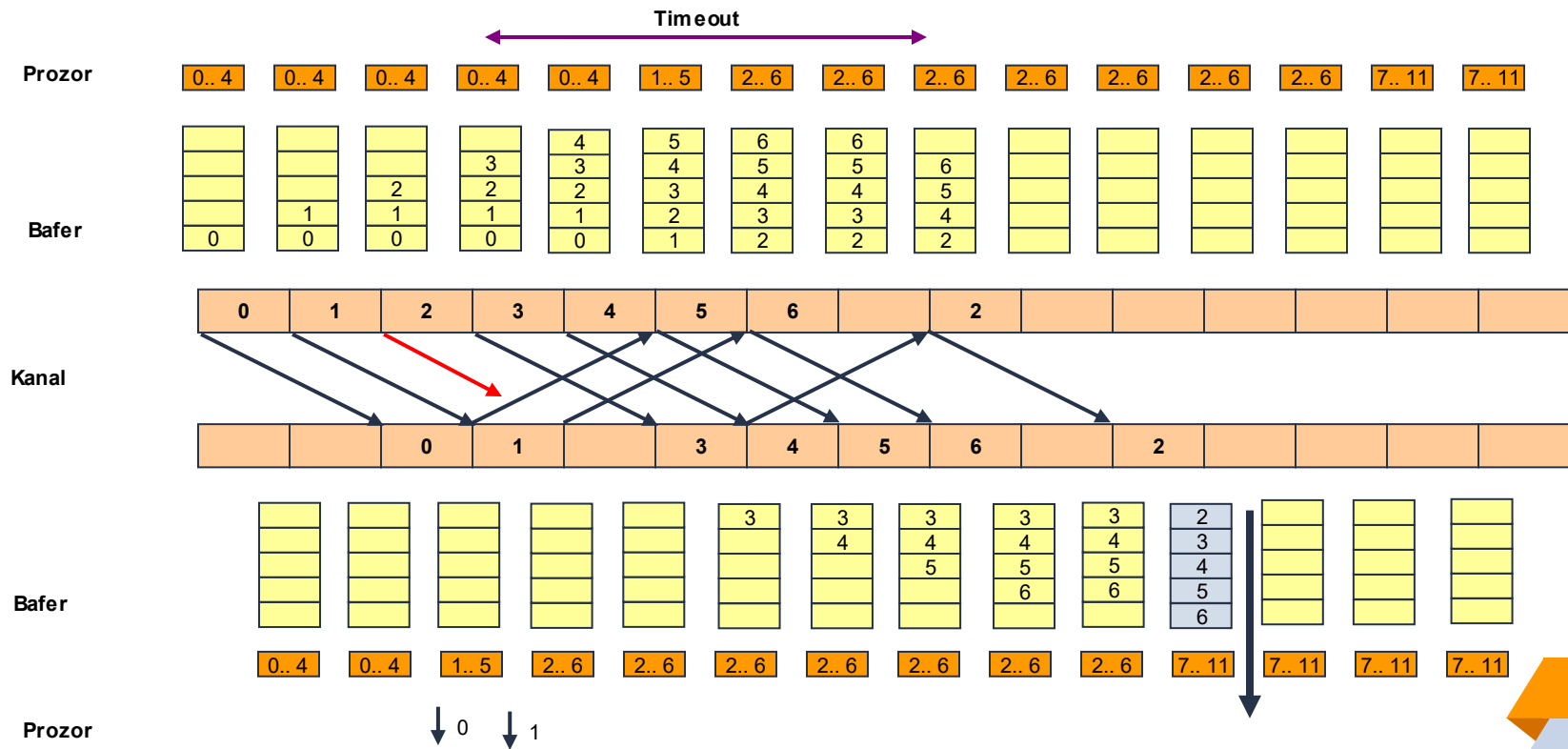
# Primer - Selektivna retransmisija sa gubitkom Info(2) frejma



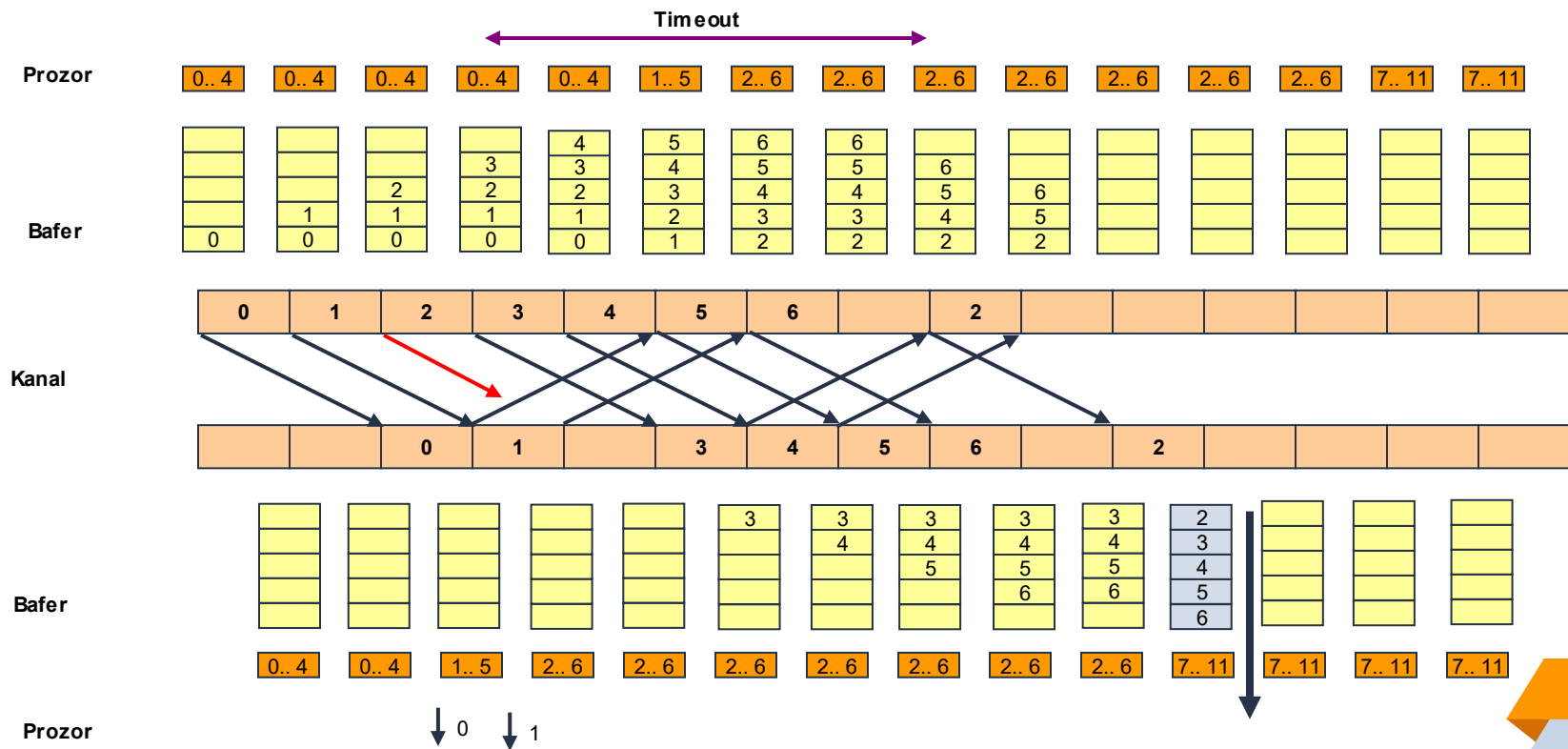
# Primer - Selektivna retransmisija sa gubitkom Info(2) frejma



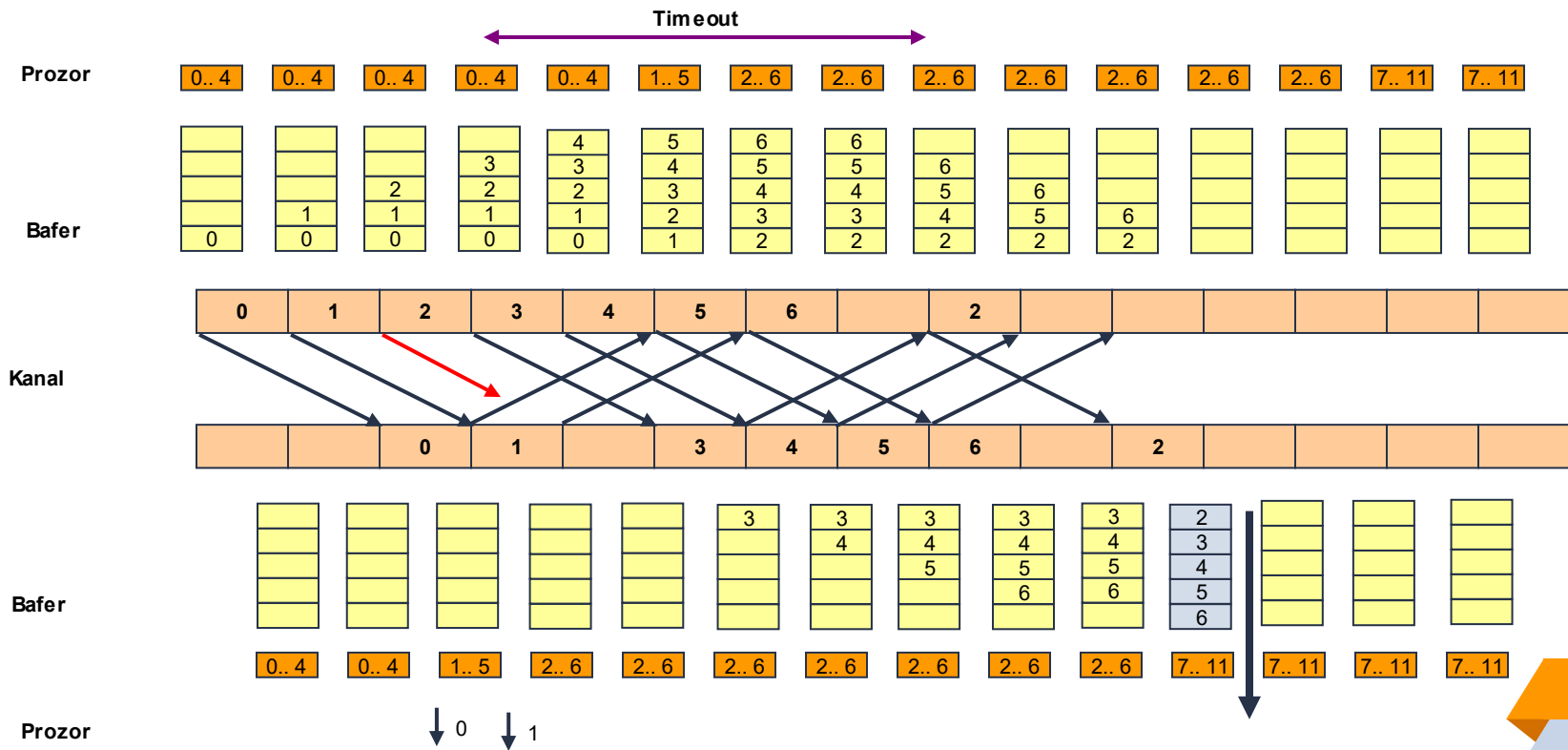
# Primer - Selektivna retransmisija sa gubitkom Info(2) frejma



## Primer - Selektivna retransmisija sa gubitkom Info(2) frejma

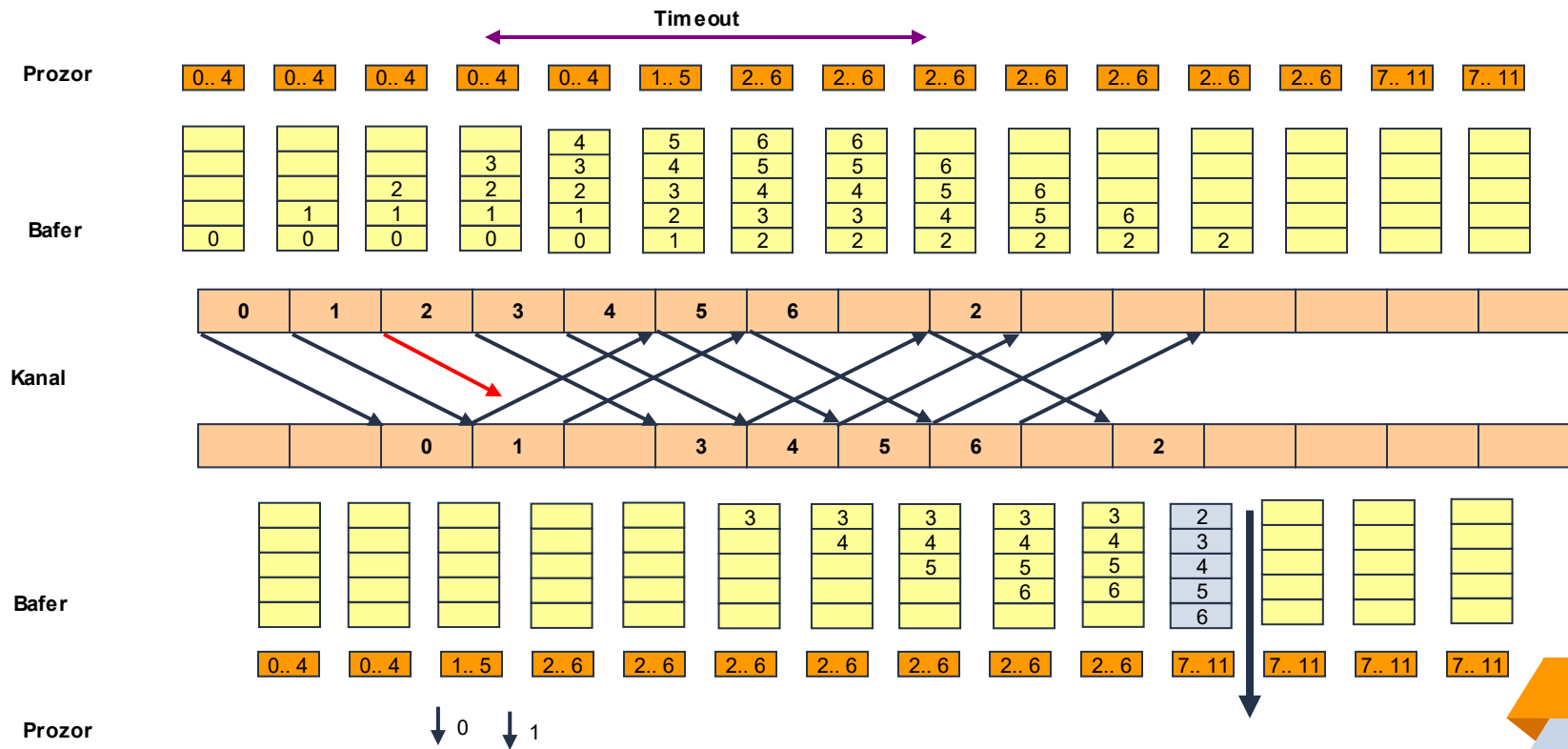


# Primer - Selektivna retransmisija sa gubitkom Info(2) frejma

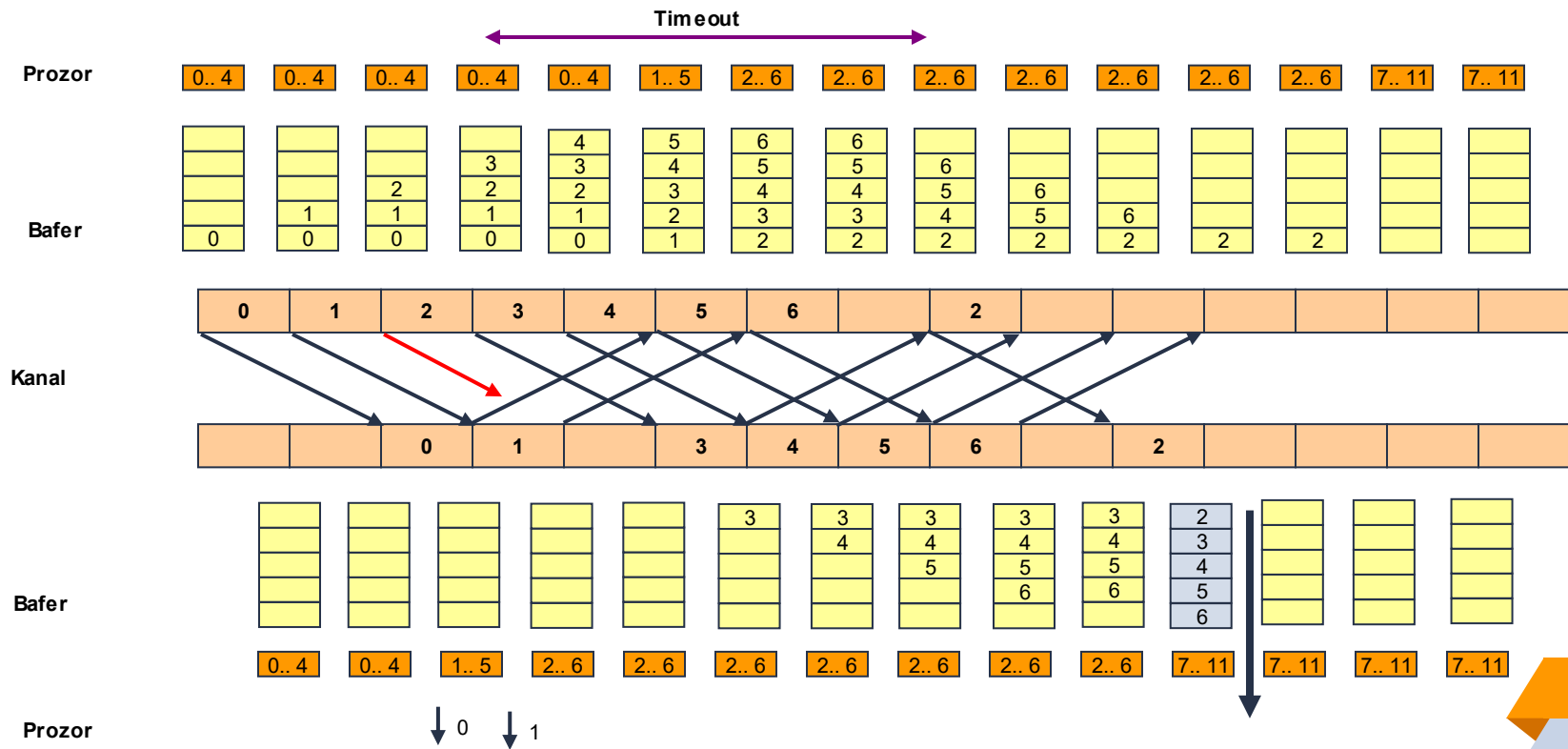




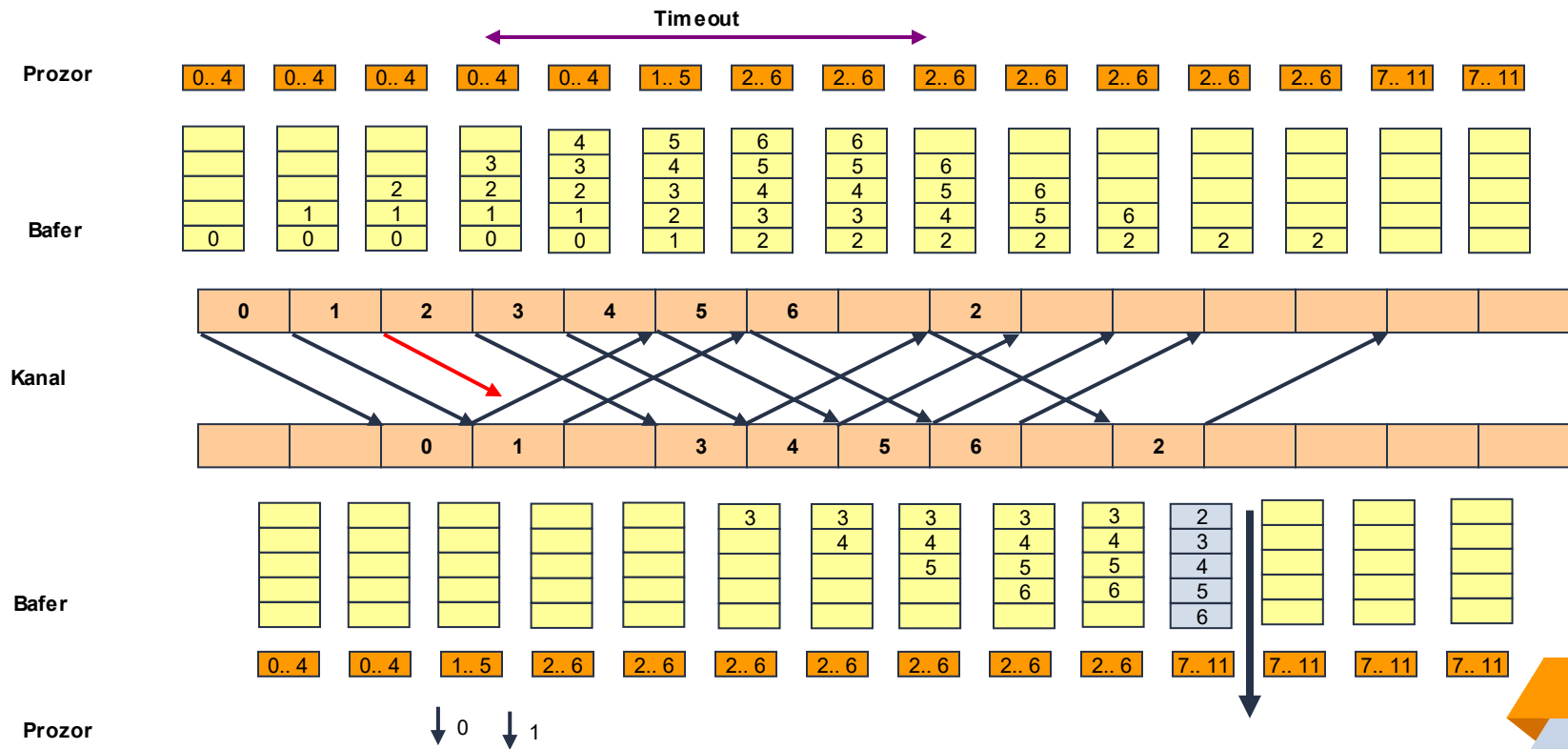
# Primer - Selektivna retransmisija sa gubitkom Info(2) frejma



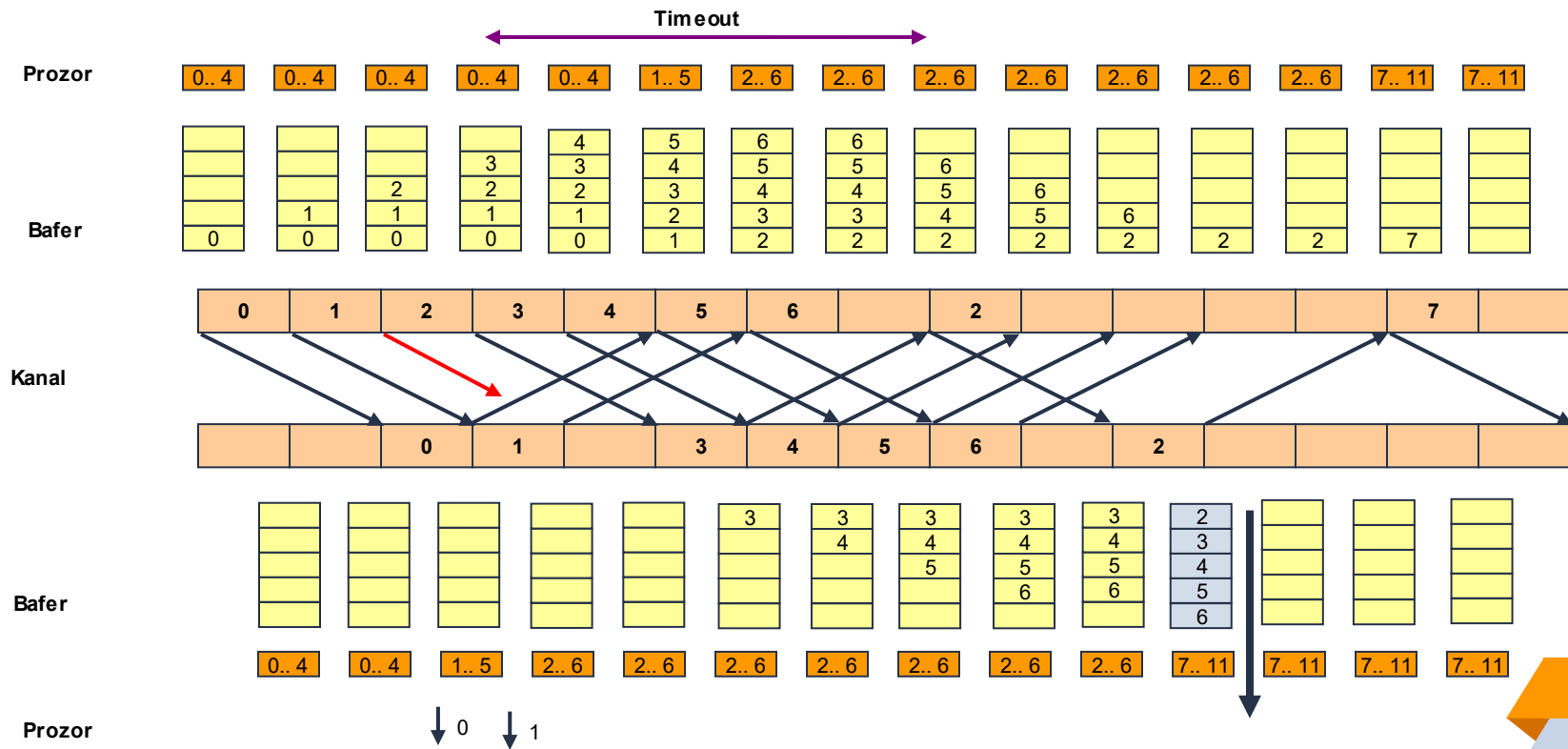
# Primer - Selektivna retransmisija sa gubitkom Info(2) frejma



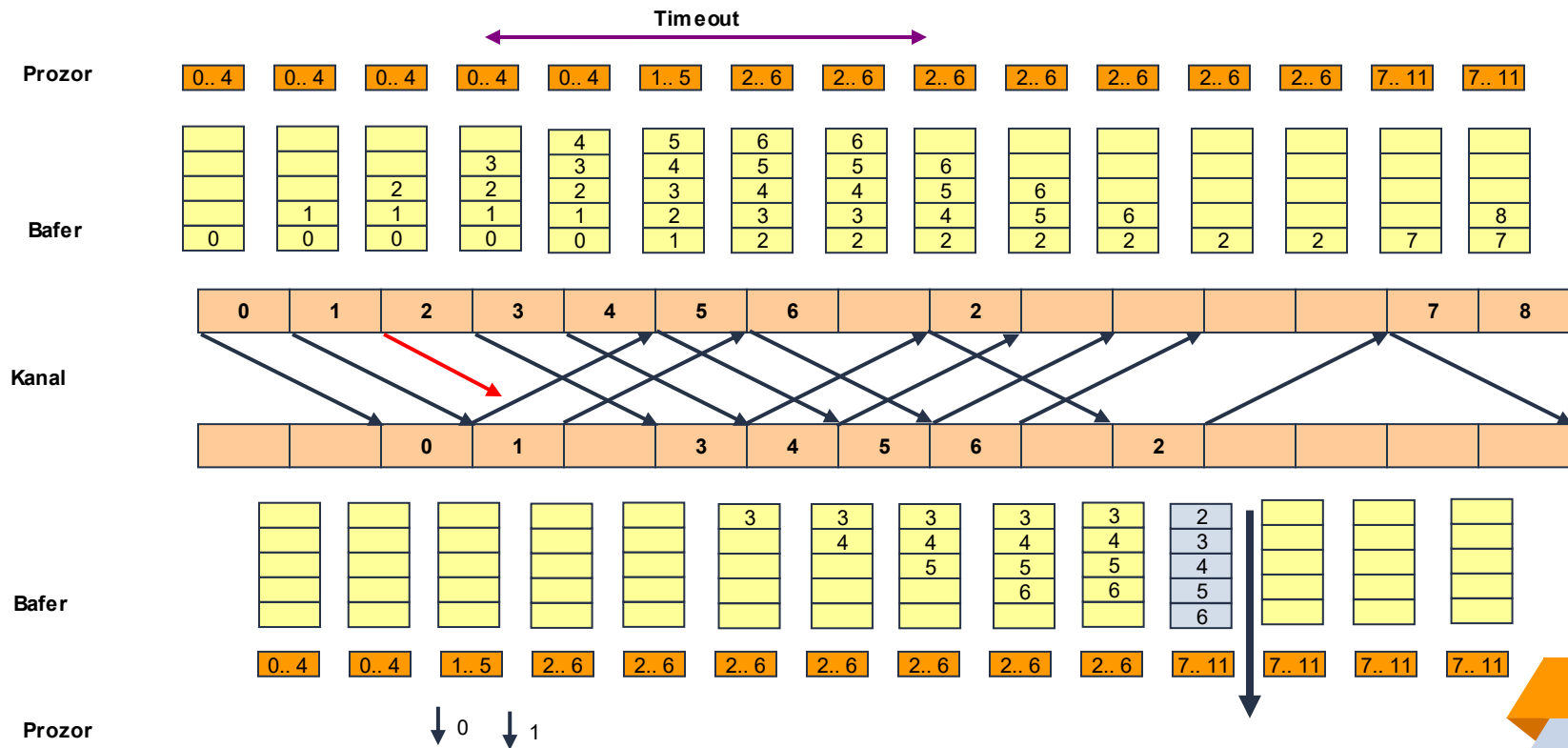
# Primer - Selektivna retransmisija sa gubitkom Info(2) frejma



# Primer - Selektivna retransmisija sa gubitkom Info(2) frejma



# Primer - Selektivna retransmisija sa gubitkom Info(2) frejma



# Numeracija frejmova

- Stani i čakaj (Stop-and-Wait)

$$No \geq 2$$

- Povratak na N (Go-Back-N)

$$No \geq w + 1$$

- Selektivna retransmisija (Selective Reject)

$$No \geq 2w$$

## Zašto $No > w + 1$ ?

Scenario katastrofe:

- Neka se numeracija vrši po modulu  $W$  ili manjem ( $W$  je veličina prozora).
- Neka je pošiljalac poslao  $W$  frejmova.
- Prijemnik je primio sve korektno i poslao ACK.
- **Svi ACK su izgubljeni** (prekid veze, šum ili nešto drugo)
- Ističe time-out na predajnoj strani i pošiljalac ponovo šalje pakete (numerisane  $0..W-1$ )
- Pošto je primio sve frejmove regularno, primalac pomera svoj prozor. Posle  $W-1$  ponovo očekuje 0, ali iz drugog kontingenta frejmova. Kako ponovo dobija 0, on to shvata kao sasvim novi paket, koji je regularno stigao, a zapravo se radi o retransmisiji. ČITAVA PORUKA JE UNIŠTENNA !!!

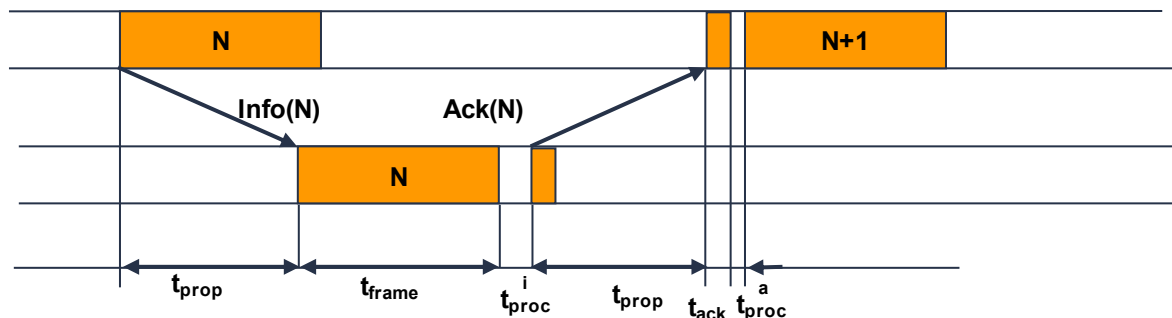
## Zašto $N_o \geq 2w$ ?

Scenario katastrofe:

- Neka se numeracija vrši po modulu  $W+K$  ( $K < W$ ).
- Neka je pošiljalac poslao  $W$  frejmova.
- Prijemnik je primio sve korektno i poslao ACK.
- **Neki od ACK su izgubljeni** (bar jedan)
- Ističe time-out na predajnoj strani i pošiljalac ponovo šalje pakete za koje nije dobio ACK
- Pošto je primio sve frejmove regularno, primalac pomera svoj prozor. Obzirom da novi prozor nije disjunktan po brojevima frejmova u odnosu na stari, sigurno se i retransmitovani frejmovi nalaze u njemu. Primalac ih prima kao nove i ČITAVA PORUKA JE UNIŠTENA!!!



# Iskorišćenost kanala



Ukupno vreme potrebno za slanje poruke koja se sastoji od  $n$  frejmova je  $T = n T_f$ , gde je  $T_f$  vreme koje zahteva svaki pojedinačni frejm i ono iznosi:

$$T_f = t_{prop} + t_{frame} + t_{proc}^i + t_{prop} + t_{ack} + t_{proc}^a$$

- $t_{prop}$  - propagaciono kašnjenje kroz kanal
- $t_{frame}$  - vreme potrebno za slanje jednog frejma
- $t_{proc}^i, t_{proc}^a$  - vreme potrebno da bi se obradila pristigli **info** ili **ack** frejm, respektivno
- $t_{ack}$  - vreme potrebno za slanje ACK frejma

# Iskorišćenost kanala

## Stop-and-Wait bez grešaka

Zanemarujući neka vremena, možemo uprostiti izraz za T:

$$T = n ( 2 t_{prop} + t_{frame} )$$

Sada se može definisati iskorišćenost kanala kao:

$$U = \frac{n t_{frame}}{n (2 t_{prop} + t_{frame})} = \frac{t_{frame}}{2 t_{prop} + t_{frame}}$$

Ako uvedemo da je  $a = \frac{t_{prop}}{t_{frame}}$ , tada dobijamo:  $U = \frac{1}{1+2a}$

Parametar  $a$  možemo izraziti i kao:

$$a = \frac{\text{propagaciono\_vreme}}{\text{transmisiono\_vreme}} = \frac{d/V}{L/R} = \frac{Rd}{VL}$$

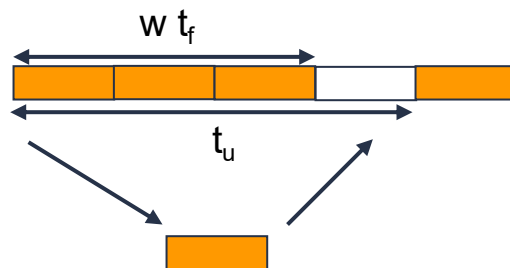
$d$  - rastojanje (*distance*) prijemnika i predajnika,  $V$  - brzina (*velocity*) propagacije (za optička vlakna približno brzini svetlosti  $V_1 = 3 * 10^8$  m/sec, a za bakarne provodnike oko 0.67 brzine svetlosti),  $L$  - dužina frejma u bitovima, a  $R$  - brzina slanja (*data rate*).

# Iskorišćenost kanala

## Sliding Window bez grešaka

Efikasnost iskorišćenja kanala se značajno povećava ako se omogući slanje većeg broja frejmove pre potvrde bilo koga od njih. Ako sa  $w$  označimo veličinu prozora na predajnoj strani, a sa  $a$  odnos propagacionog kašnjenja i vremena potrebnog za slanje jednog frejma ( kao u prethodnom primeru ), iskorišćenost kanala se može aproksimirati sledećim izrazom:

$$U = \begin{cases} 1 & w \geq 2a + 1 \\ \frac{w}{2a + 1} & w < 2a + 1 \end{cases}$$



Ovi izrazi dobijeni su pod pretpostavkom da je vreme prenosa 1 (normalizacija), te da je  $a$  zapravo propagaciono kašnjenje. Ukoliko je broj bafera (tj. veličina prozora) na predajnoj strani dovoljno veliki (  $w t_{trans} > 2 t_{prop} + t_{frame}$  ), iskorišćenost kanala je 100% (u idealnom slučaju bez grešaka u kanalu i uz zanemarivanje vremena obrade i slanja ACK frejmove). Ako broj bafera na predajnoj strani nije dovoljan, nastaje zastoј nakon poslatih  $w$  frejmove sve dok se ne primi prva potvrda.

# Iskorišćenost kanala

## Realni protokoli

Umesto da se koristi obrazac  $U = T_f / T_t$ , gde je  $T_f$  vreme slanja jednog frejma, a  $T_t$  ukupno vreme za koje je kanal zauzet tokom slanja jednog frejma, ukoliko postoji mogućnost pojave grešaka u kanalu, neki frejmovi se moraju ponovo poslati.

To degradira iskorišćenost kanala i ona se sada računa po obrascu:

$$U = \frac{T_f}{N_r T_t}$$

gde je  $N_r$  očekivani broj slanja frejma.

Sada izrazi za iskorišćenost kanala imaju sledeći oblik:

# Iskorišćenost kanala

## Realni protokoli

### Stop-and-Wait

$$U = \frac{1}{N_r(1+2a)} = \frac{1-P}{1+2a}$$

### Selectiv reject (selektivna retransmisija)

$$U = \begin{cases} 1-P & w \geq 2a+1 \\ \frac{w(1-P)}{2a+1} & w < 2a+1 \end{cases}$$

### Go-back-N (povratak na N)

$$U = \begin{cases} \frac{1-P}{1+2aP} & w \geq 2a+1 \\ \frac{w(1-P)}{(2a+1)(1-P+wP)} & w < 2a+1 \end{cases}$$

P je verovatnoća da je jedan frejm narušen ( smatra se da ACK i NAK frejmovi nisu narušeni ). Za stani i čekaj i selektivnu retransmisiju  $N_r=1/(1-P)$ , dok je za povratak na N veza daleko komplikovanja.

## Verovatnoća narušavanja frejmova

- Ukoliko je verovatnoća greške jednog bita ( *bit error rate* ) – **E**, tada je verovatnoća narušenosti frejma veličine **L** bitova:

$$P = 1 - ( 1 - E )^L$$

Obrazloženje:

- Ako je E verovatnoća da je bit narušen, verovatnoća da nije narušen je (1-E)
- Frejm nije narušen ako i samo ako ni jedan bit u njemu nije narušen, a verovatnoća za to je  $= (1-E) (1-E) (1-E) \dots (1-E) = (1-E)^L$ , gde je L broj bitova u frejmu.
- Verovatnoća da je frejm narušen je komplementaran verovatnoći da nije narušen, tj.

$$P = 1 - \bar{P}$$

Primer: Za  $E = 10^{-4}$  i  $L = 2048$ ,  $P = 0.185$ . Ako je narušen savaki 10-to hiljaditi bit, narušeno je 18,5% frejmova, tj. gotovo svaki peti!

# Primer izračunavanja iskorišćenosti kanala u slučajevima nastajanja grešaka

Niz poruka veličine  $L=1000$  bitova prenosi se preko 100km dugog komunikacionog kanala, pri brzini slanja od 20Mb/s. Brzina propagacije signala u kanalu je  $2 \cdot 10^8$  m/s. Ukoliko je  $E = 4 \cdot 10^{-5}$ , odrediti iskorišćenost komunikacionog kanala za protokole: stop-and-wait, go-back-N, selective-reject.

$$t_f = \frac{L}{R} = \frac{1000b}{20 \cdot 10^6 b/s} = 0.05ms \quad t_p = \frac{d}{V} = \frac{100 \cdot 10^3 m}{2 \cdot 10^8 m/s} = 0.5ms \quad a = \frac{t_p}{t_f} = 10$$

- **Stop-and-wait**

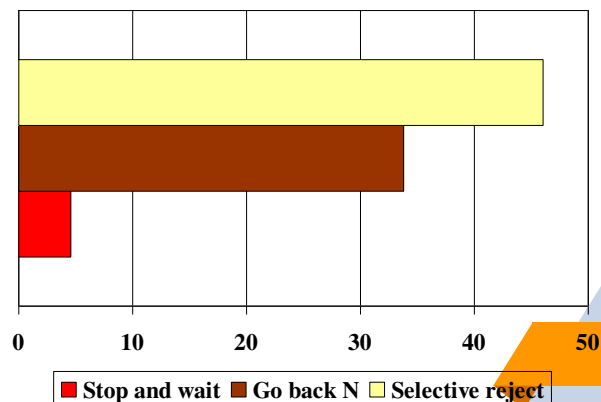
$$U = \frac{1-P}{2a+1} = 0.046$$

- **Go-back-N**

$$U = \frac{w(1-P)}{(2a+1)(1-P+wP)} = 0.338$$

- **Selective-reject**

$$U = \frac{w(1-P)}{2a+1} = 0.46$$



# Zadaci

- Prikazati saobraćaj između 2 čvora koji za kontrolu toka poruka koristi selektivnu retransmisiju. Paketi su veličina 1000b, brzina slanja 100 Kb/s, kašnjenje full-duplex linije je  $10 \mu\text{s/KM}$ , a razdaljina između čvorova je 2000 KM. Postaviti da su Info-ramovi 2 i 5 narušeni, kao i ACK-ram 3. (Numeracija počinje od nule.) Veličine predajnog i prijemnog prozora su po 5, a bafera 4 poruke.
- Predajnik i prijemnik se nalaze na udaljenosti od 1000 KM, a prenos poruka se obavlja pod kontrolom protokola povratak\_na\_N sa veličinom prozora 7 i u predajniku. Paketi su veličine 1000b. Signal se kroz kanal prostire brzinom od  $2 \cdot 10^8 \text{ m/s}$ .
  - Odrediti brzinu slanja podataka tako da iskorišćenost kanala u slučaju da u njemu ne nastaju greške, bude bar 50%.
  - Za brzinu slanja podataka dobijenu pod a) skicirati postupak slanja paketa između predajnika i prijemnika u slučaju da su narušeni INFO2 i ACK3 ramovi.
  - Za brzinu slanja podataka dobijenu pod a) izračunati iskorišćenost kanala u slučaju da je verovatnoća narušenosti nekog bita u poruci  $E = 2 \cdot 10^{-5}$ .



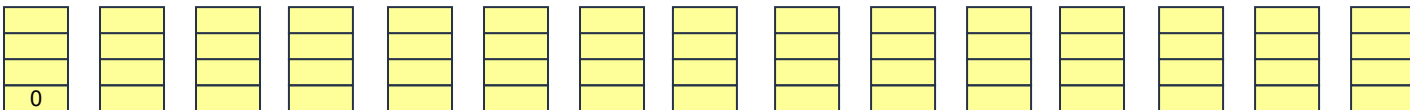
# Zadatak 1 – Rešenje

$$t_f = 10 \text{ ms } t_p = 20 \text{ ms} \Rightarrow a = 2$$

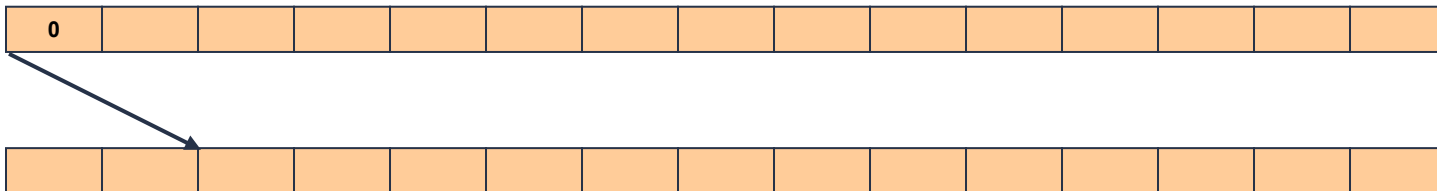
Prozor

0..4 0..4 0..4 0..4 0..4 1..5 2..6 2..6 2..6 2..6 2..6 2..6 2..6 3..7 5..9

Bafer



Kanal



Bafer

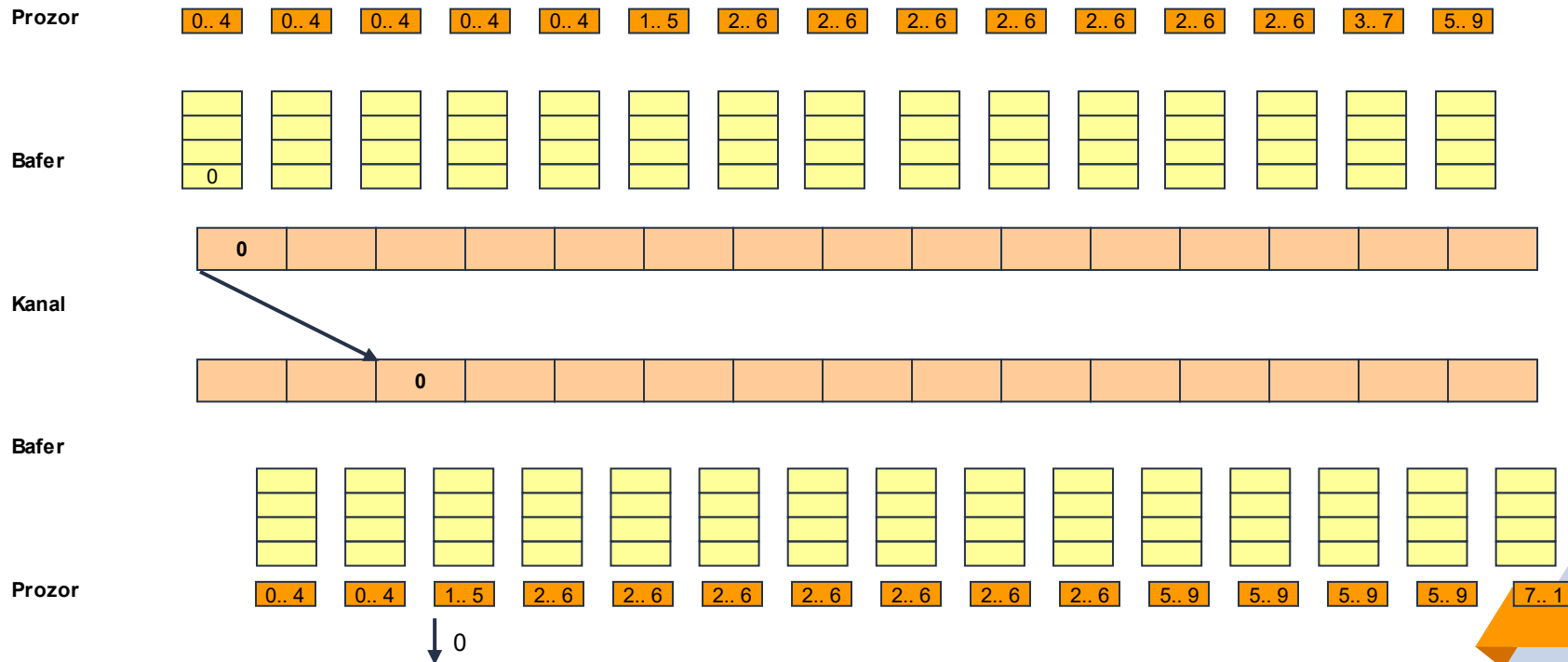


Prozor

0..4 0..4 1..5 2..6 2..6 2..6 2..6 2..6 2..6 2..6 5..9 5..9 5..9 5..9 7..1

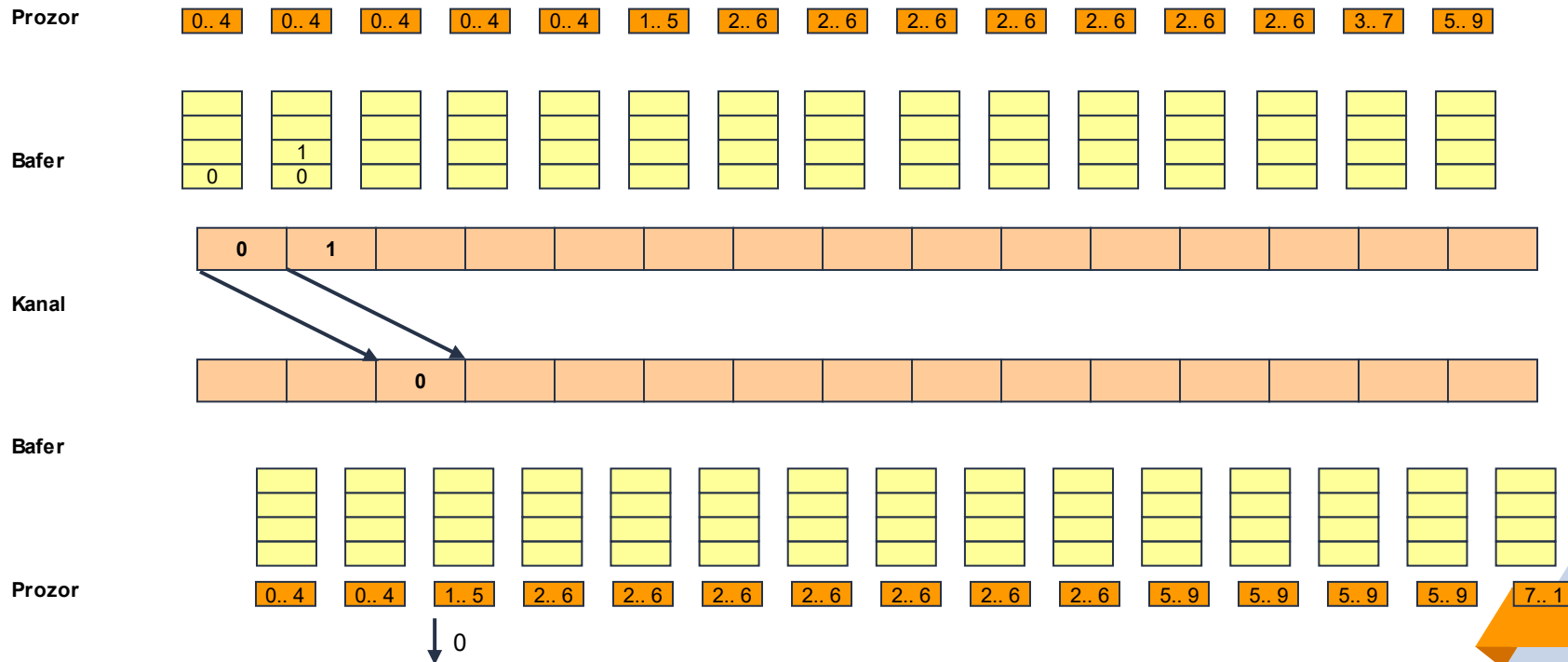
# Zadatak 1 – Rešenje

$$t_f = 10 \text{ ms } t_p = 20 \text{ ms} \Rightarrow a = 2$$



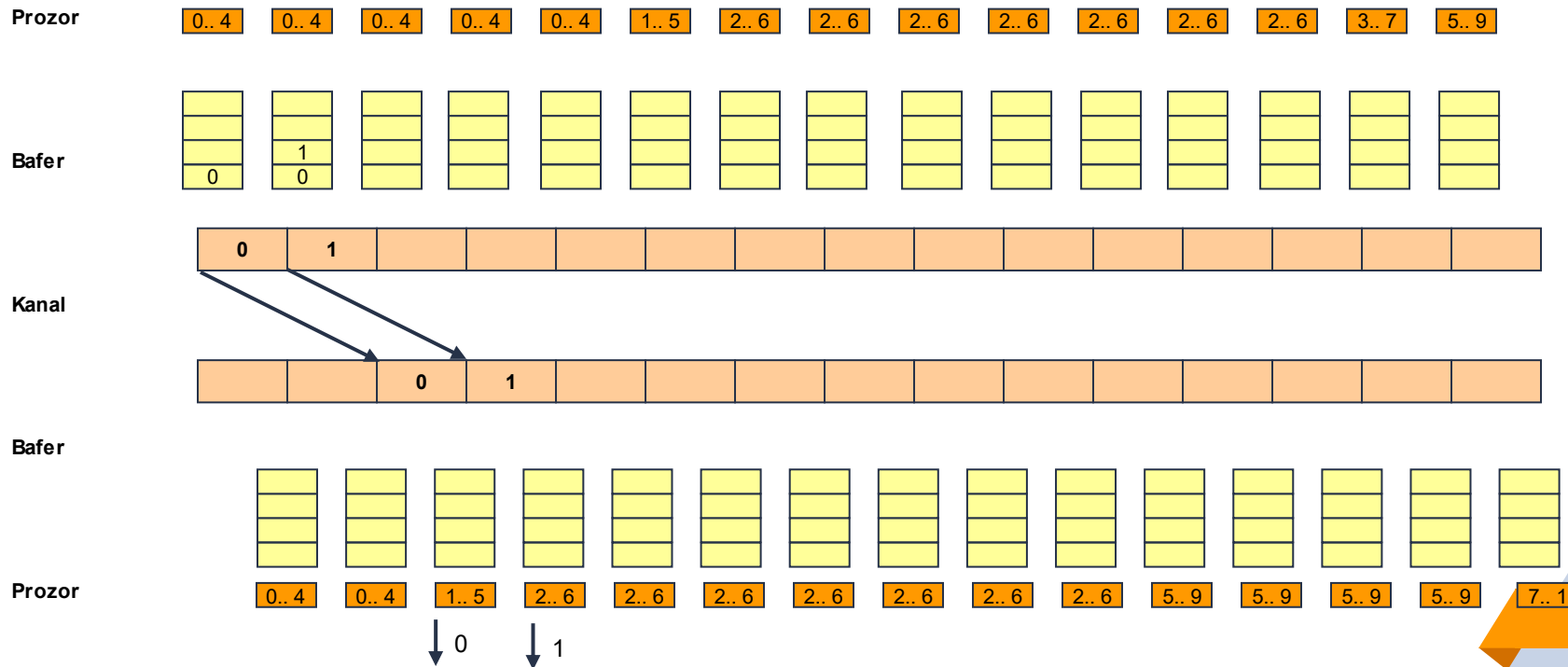
# Zadatak 1 – Rešenje

$$t_f = 10 \text{ ms } t_p = 20 \text{ ms} \Rightarrow a = 2$$



# Zadatak 1 – Rešenje

$$t_f = 10 \text{ ms } t_p = 20 \text{ ms} \Rightarrow a = 2$$



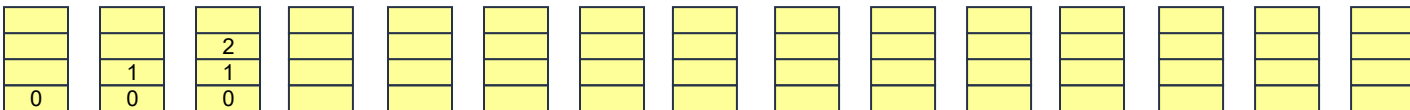
# Zadatak 1 – Rešenje

$$t_f = 10 \text{ ms } t_p = 20 \text{ ms} \Rightarrow a = 2$$

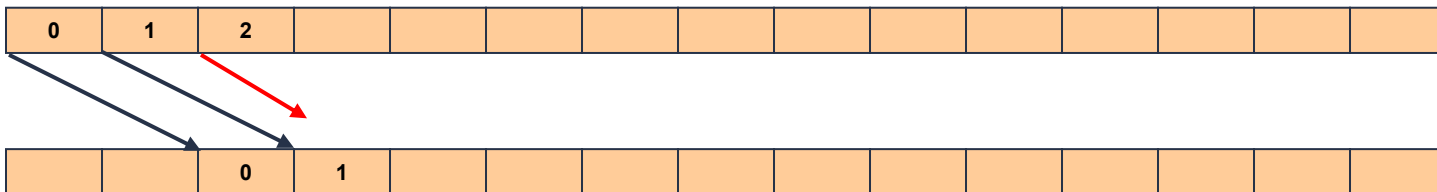
Prozor

0..4 0..4 0..4 0..4 0..4 1..5 2..6 2..6 2..6 2..6 2..6 2..6 2..6 3..7 5..9

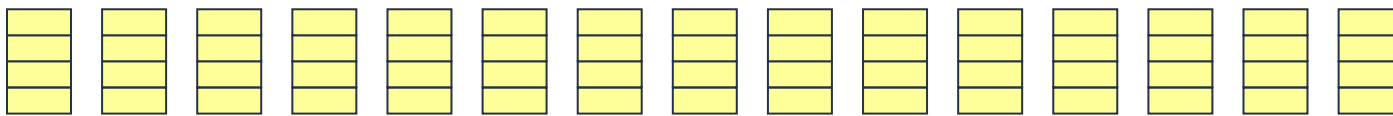
Bafer



Kanal



Bafer



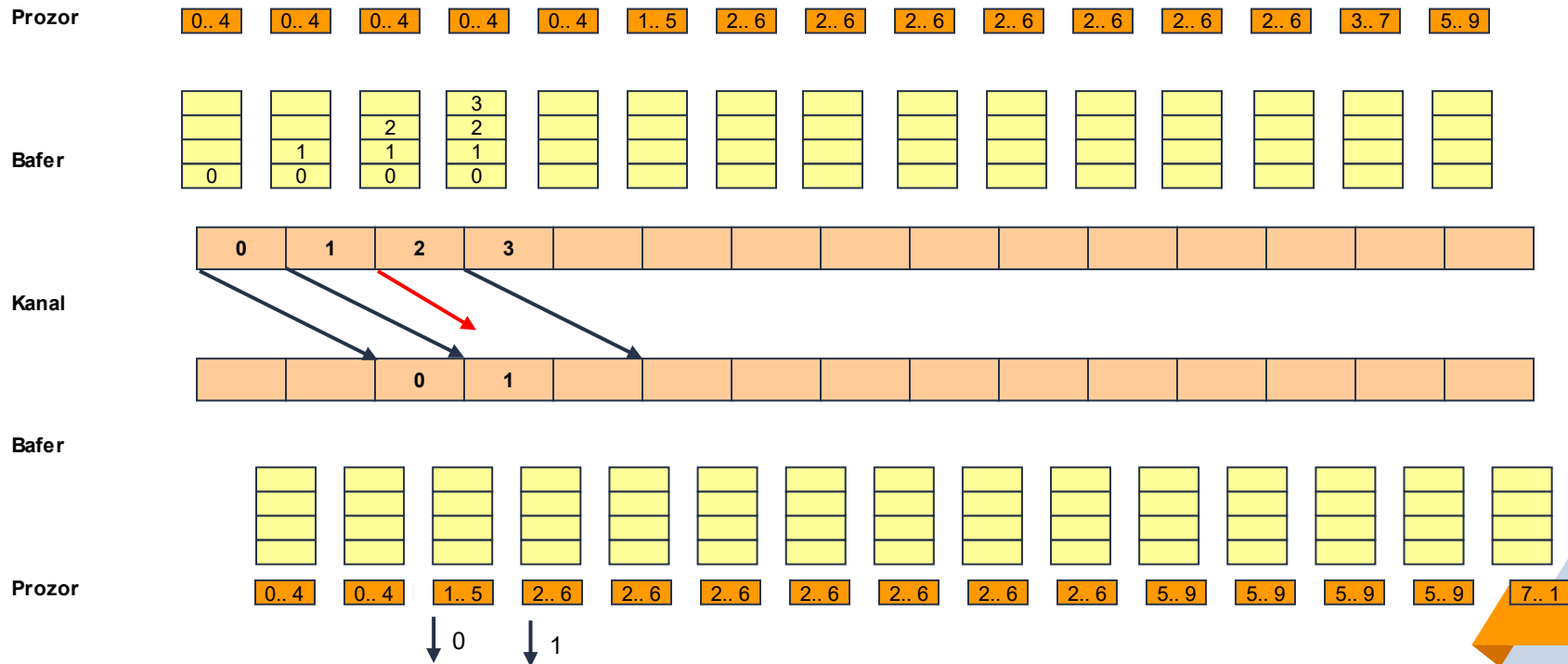
Prozor

0..4 0..4 1..5 2..6 2..6 2..6 2..6 2..6 2..6 2..6 5..9 5..9 5..9 5..9 7..1

↓ 0 ↓ 1

# Zadatak 1 – Rešenje

$$t_f = 10 \text{ ms } t_p = 20 \text{ ms} \Rightarrow a = 2$$



# Zadatak 1 – Rešenje

$$t_f = 10 \text{ ms } t_p = 20 \text{ ms} \Rightarrow a = 2$$

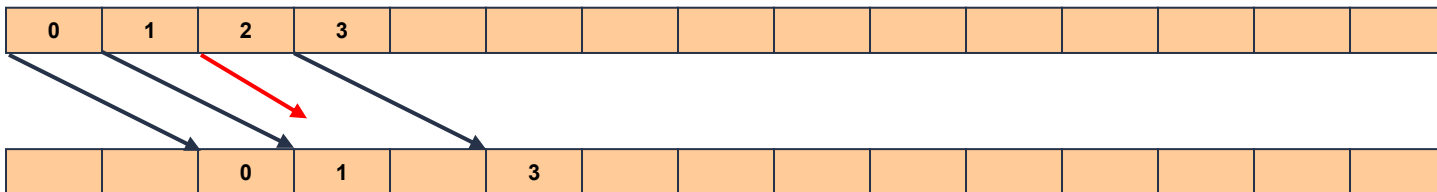
Prozor

0..4 0..4 0..4 0..4 0..4 1..5 2..6 2..6 2..6 2..6 2..6 2..6 2..6 3..7 5..9

Bafer

			3											
		2	2											
	1	1	1											
0	0	0	0											

Kanal



Bafer

					3									

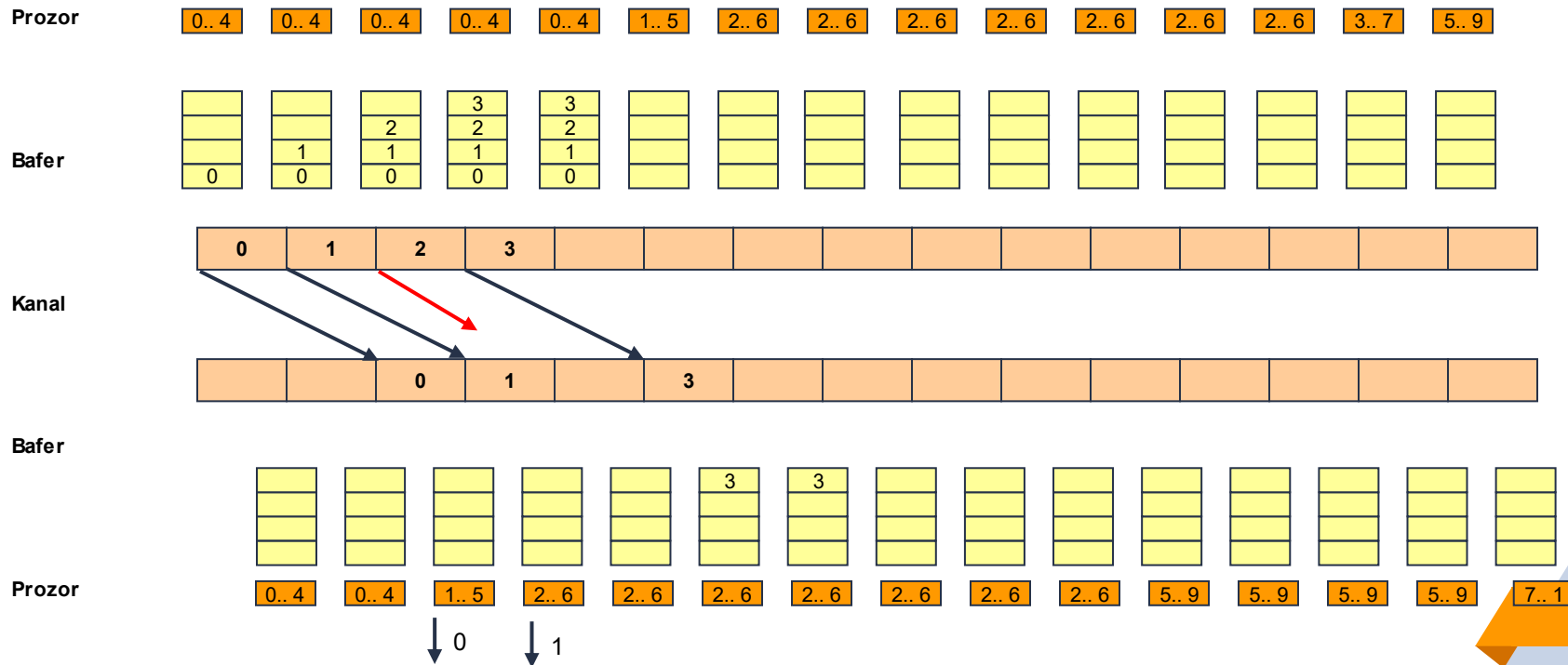
Prozor

0..4 0..4 1..5 2..6 2..6 2..6 2..6 2..6 2..6 2..6 5..9 5..9 5..9 5..9 7..1

↓ 0 ↓ 1

# Zadatak 1 – Rešenje

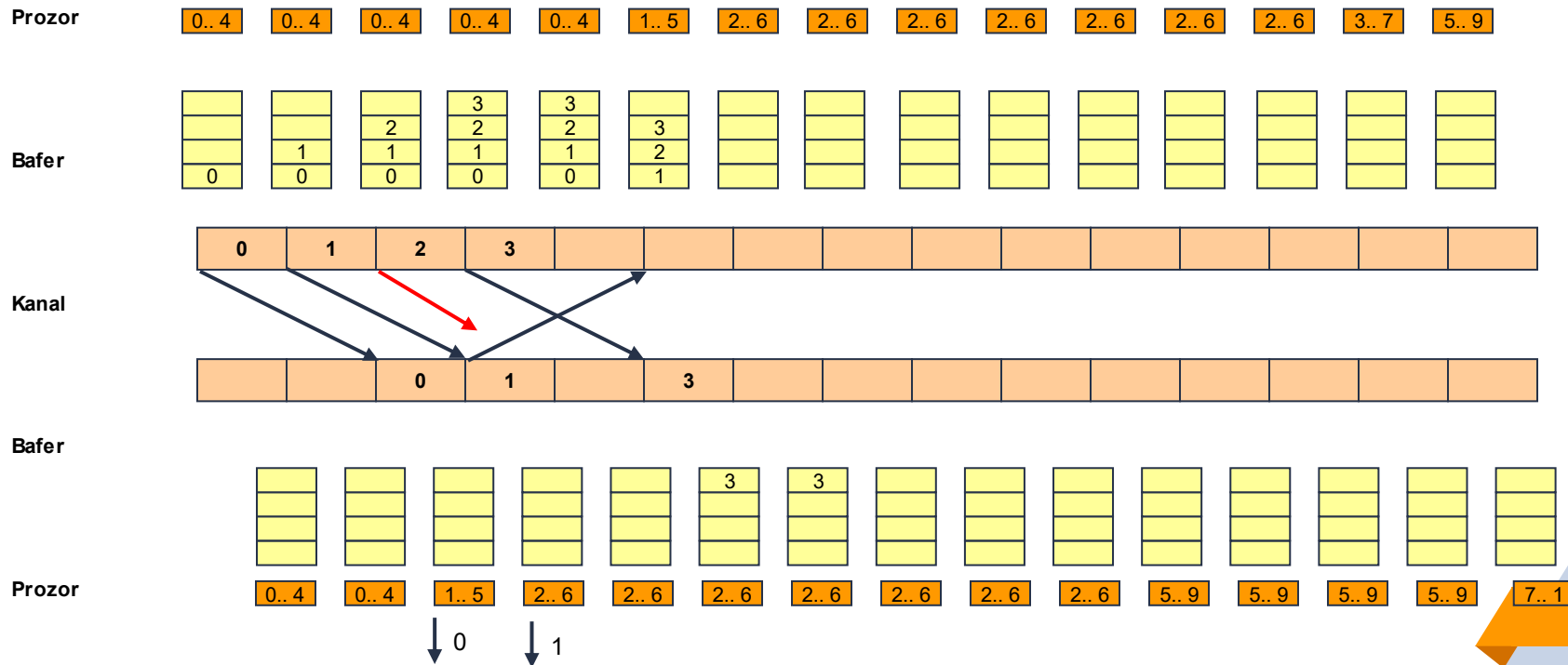
$$t_f = 10 \text{ ms } t_p = 20 \text{ ms} \Rightarrow a = 2$$





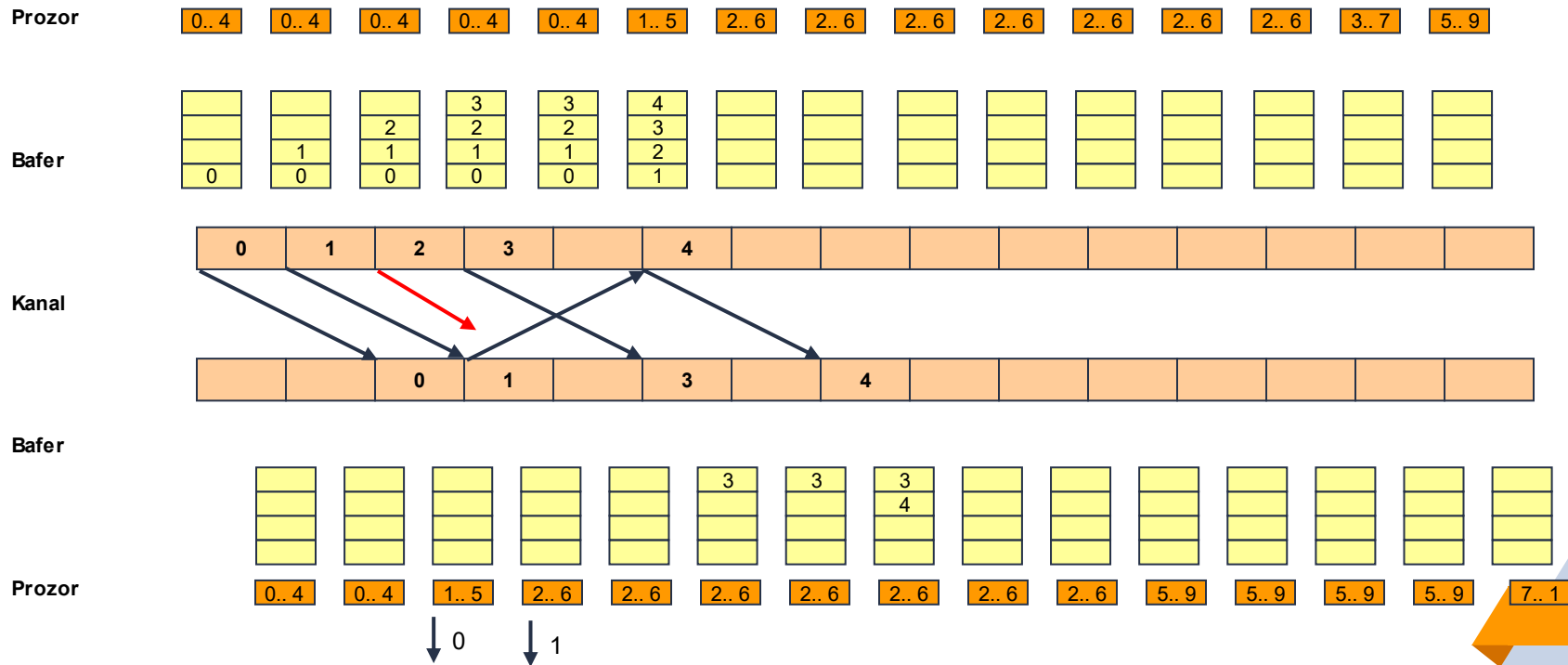
# Zadatak 1 – Rešenje

$$t_f = 10 \text{ ms } t_p = 20 \text{ ms} \Rightarrow a = 2$$



# Zadatak 1 – Rešenje

$$t_f = 10 \text{ ms } t_p = 20 \text{ ms} \Rightarrow a = 2$$



# Zadatak 1 – Rešenje

$$t_f = 10 \text{ ms } t_p = 20 \text{ ms} \Rightarrow a = 2$$

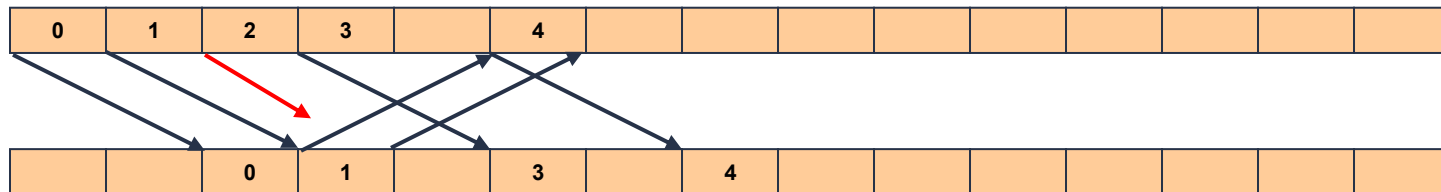
Prozor

0..4 0..4 0..4 0..4 0..4 1..5 2..6 2..6 2..6 2..6 2..6 2..6 2..6 3..7 5..9

Bafer

			3	3	4									
		2	2	2	3	4								
	1	1	1	1	2	3								
0	0	0	0	0	1	2								

Kanal



Bafer

					3	3	3							
							4							

Prozor

0..4 0..4 1..5 2..6 2..6 2..6 2..6 2..6 2..6 5..9 5..9 5..9 5..9 7..1

↓ 0 ↓ 1

# Zadatak 1 – Rešenje

$$t_f = 10 \text{ ms } t_p = 20 \text{ ms} \Rightarrow a = 2$$

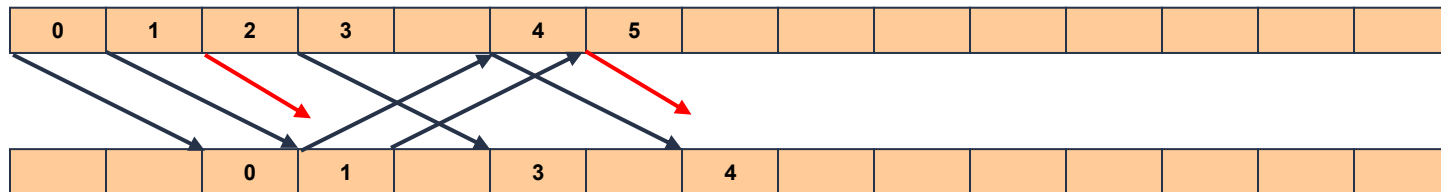
Prozor

0..4 0..4 0..4 0..4 0..4 1..5 2..6 2..6 2..6 2..6 2..6 2..6 2..6 3..7 5..9

Bafer

			3	3	4	5								
		2	2	2	3	4								
	1	1	1	1	2	3								
0	0	0	0	0	1	2								

Kanal



Bafer

					3	3	3							
							4							

Prozor

0..4 0..4 1..5 2..6 2..6 2..6 2..6 2..6 2..6 5..9 5..9 5..9 5..9 7..1

↓ 0  
↓ 1

# Zadatak 1 – Rešenje

$$t_f = 10 \text{ ms } t_p = 20 \text{ ms} \Rightarrow a = 2$$

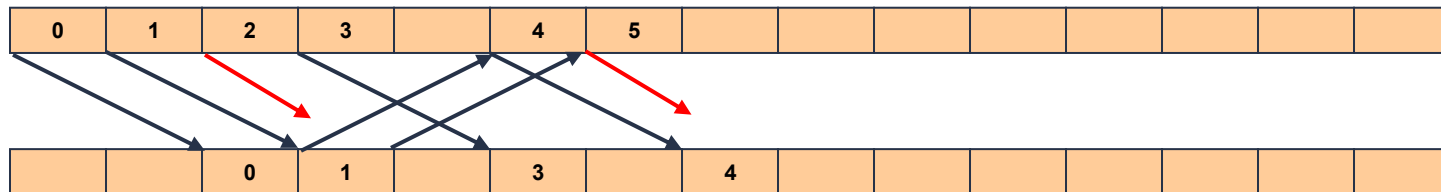
Prozor

0..4 0..4 0..4 0..4 0..4 1..5 2..6 2..6 2..6 2..6 2..6 2..6 2..6 3..7 5..9

Bafer

			3	3	4	5	5							
		2	2	2	3	4	4							
	1	1	1	1	2	3	3							
0	0	0	0	0	1	2	2							

Kanal



Bafer

					3	3	3	3						
							4	4						

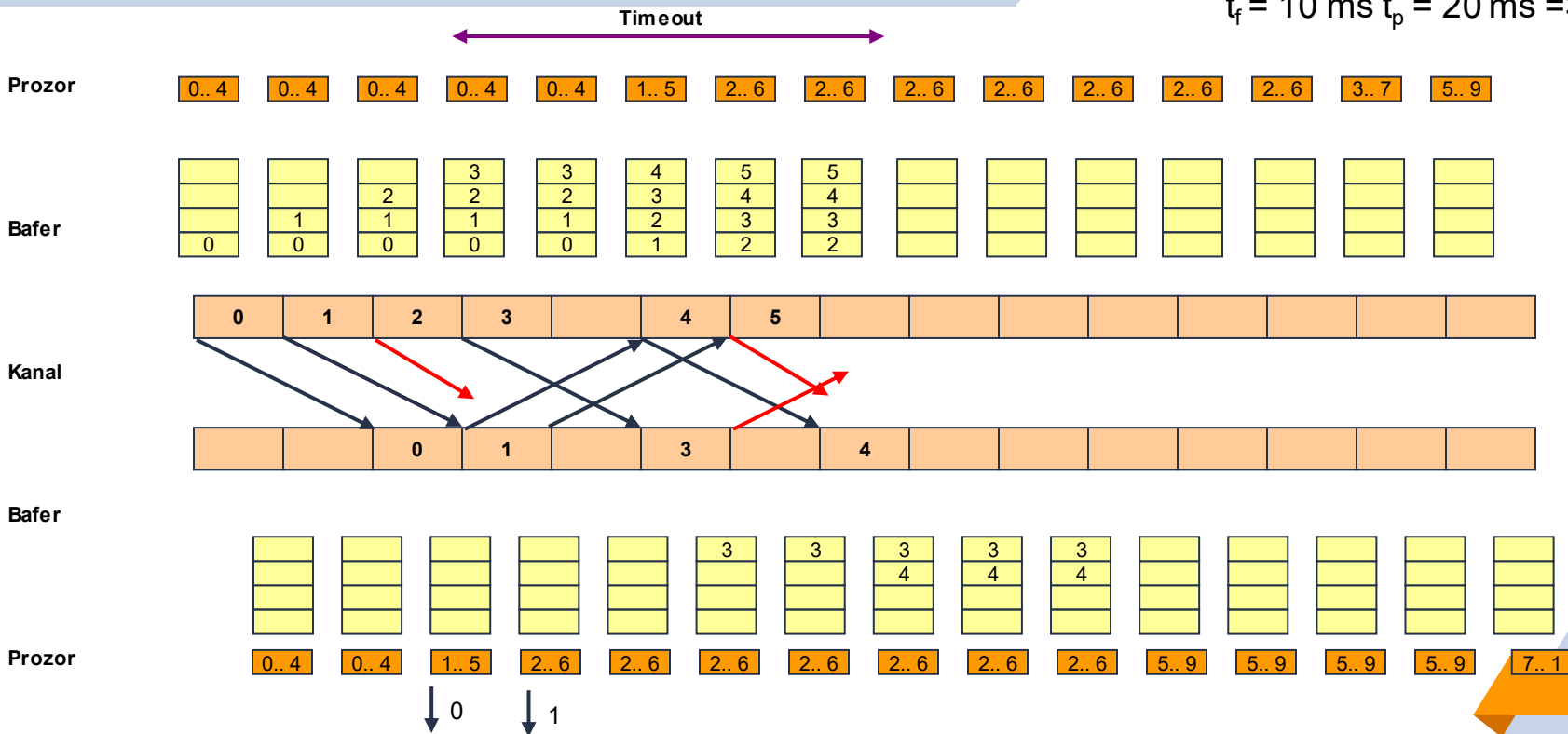
Prozor

0..4 0..4 1..5 2..6 2..6 2..6 2..6 2..6 2..6 2..6 5..9 5..9 5..9 5..9 7..1

↓ 0  
↓ 1

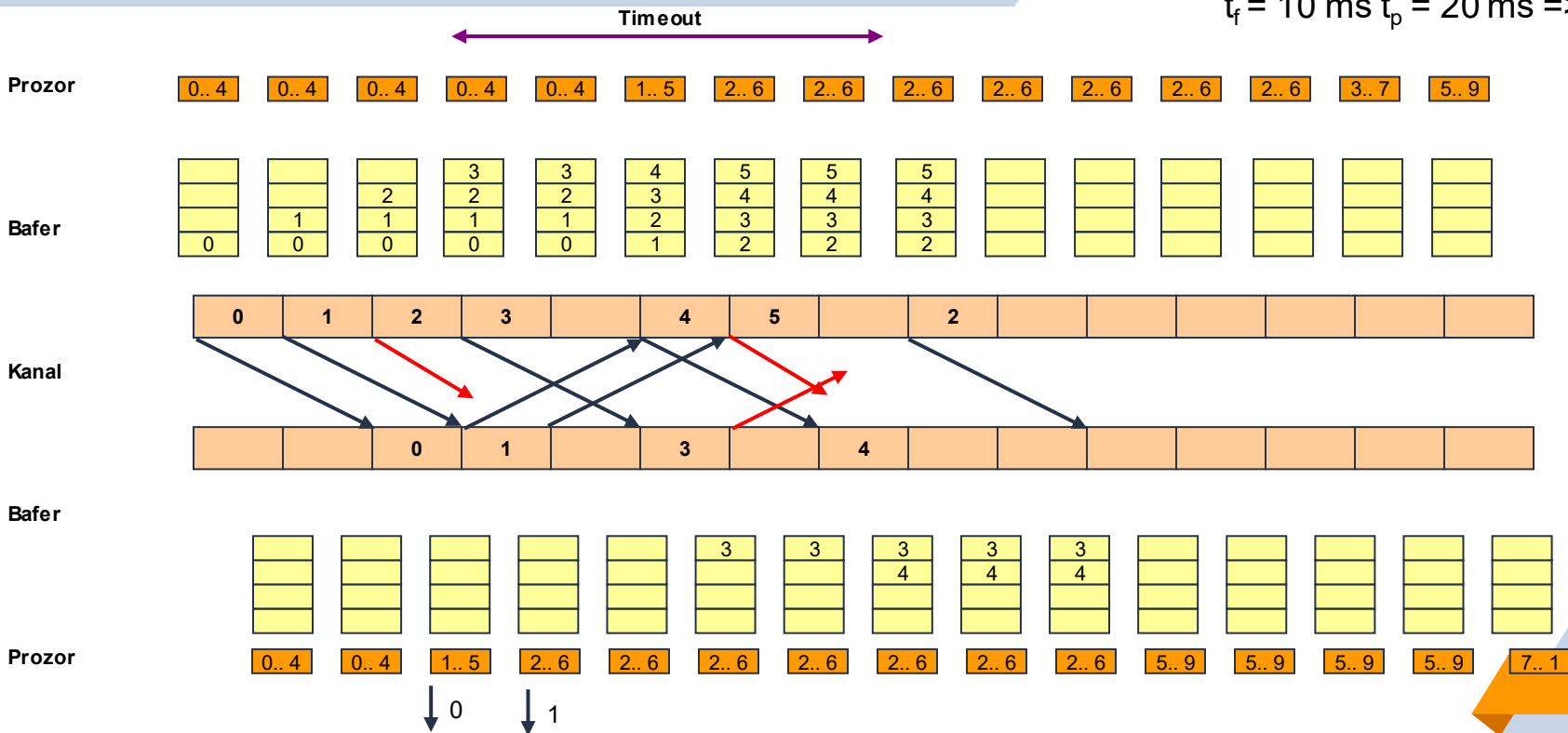
# Zadatak 1 – Rešenje

$$t_f = 10 \text{ ms } t_p = 20 \text{ ms} \Rightarrow a = 2$$



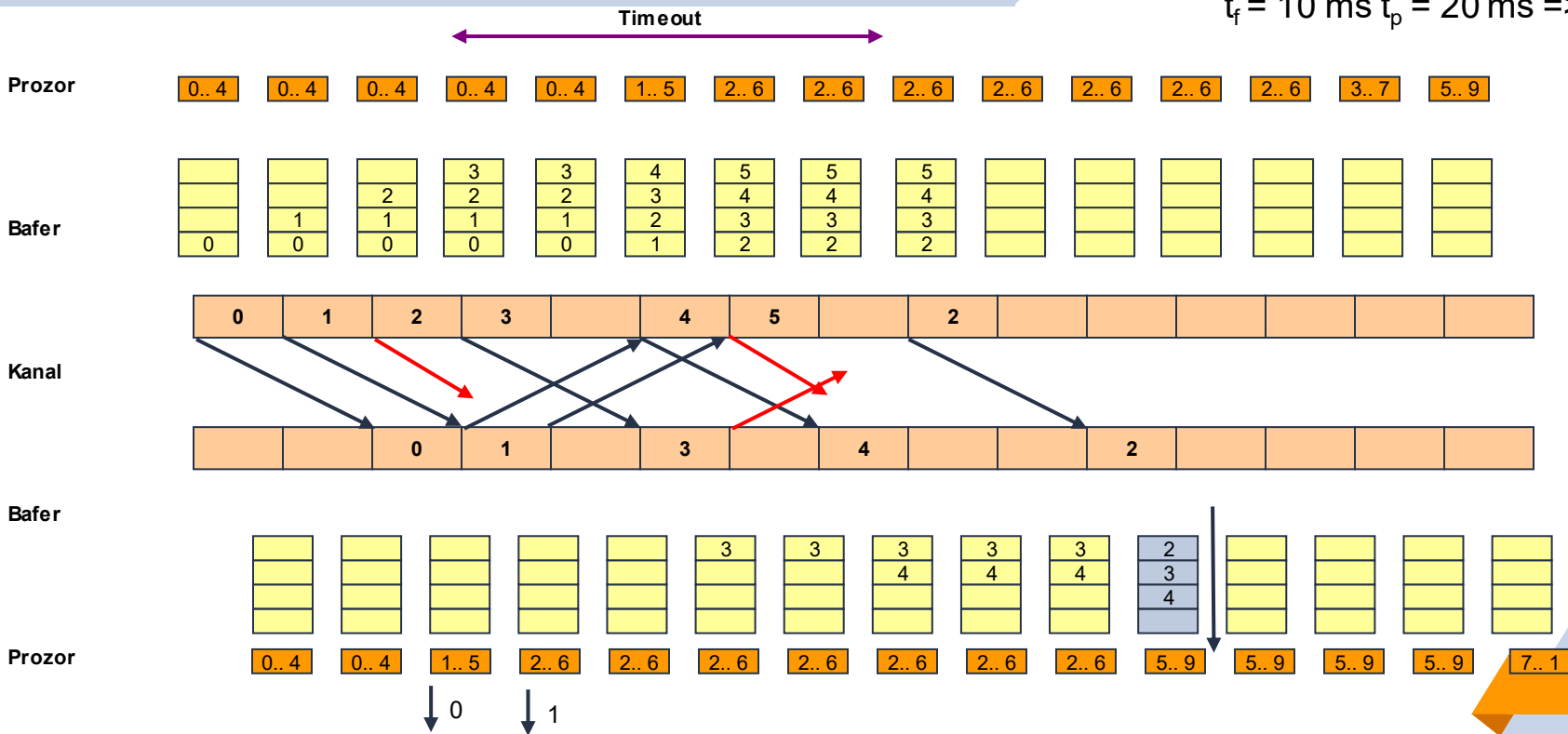
# Zadatak 1 – Rešenje

$$t_f = 10 \text{ ms } t_p = 20 \text{ ms} \Rightarrow a = 2$$



# Zadatak 1 – Rešenje

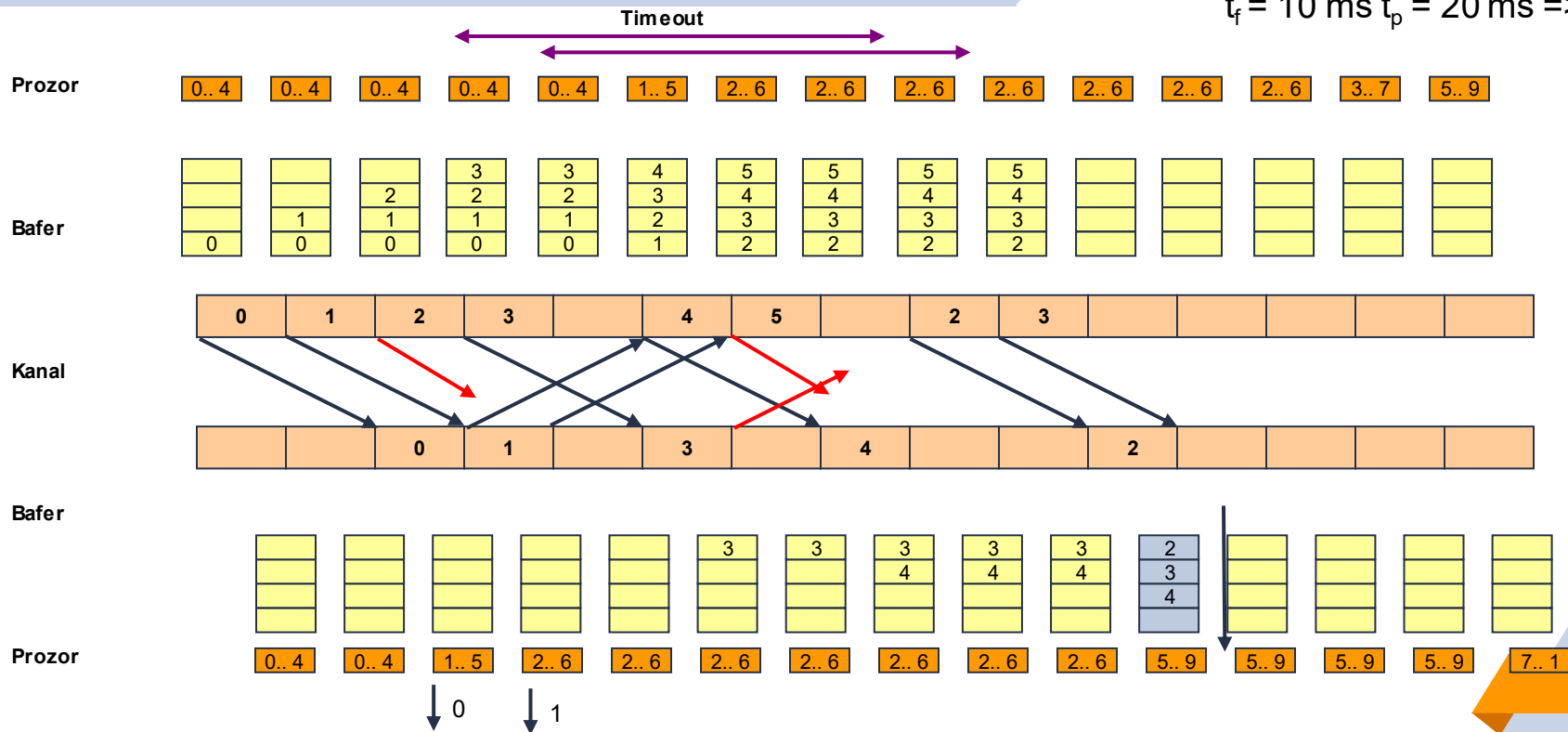
$$t_f = 10 \text{ ms } t_p = 20 \text{ ms} \Rightarrow a = 2$$





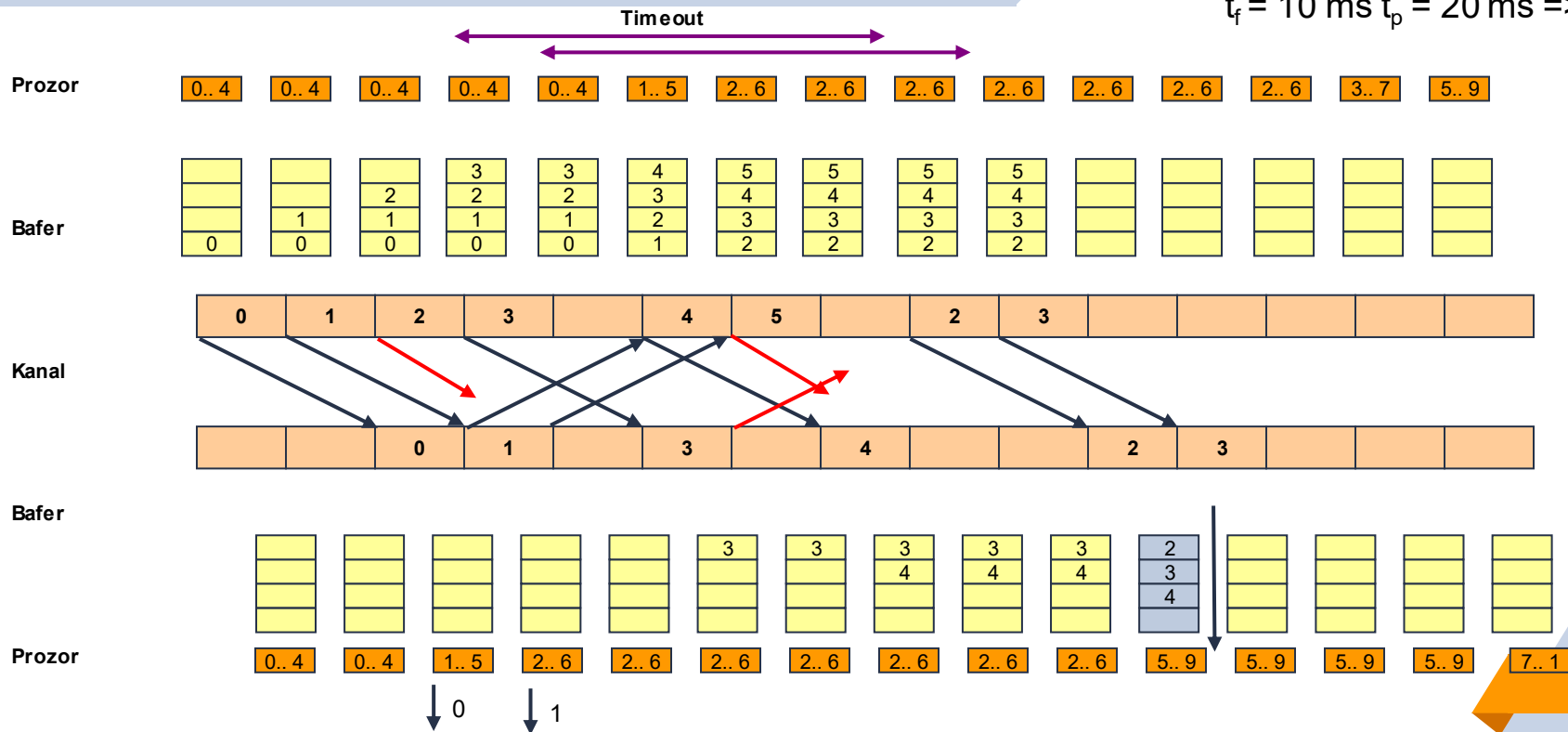
# Zadatak 1 – Rešenje

$$t_f = 10 \text{ ms } t_p = 20 \text{ ms} \Rightarrow a = 2$$



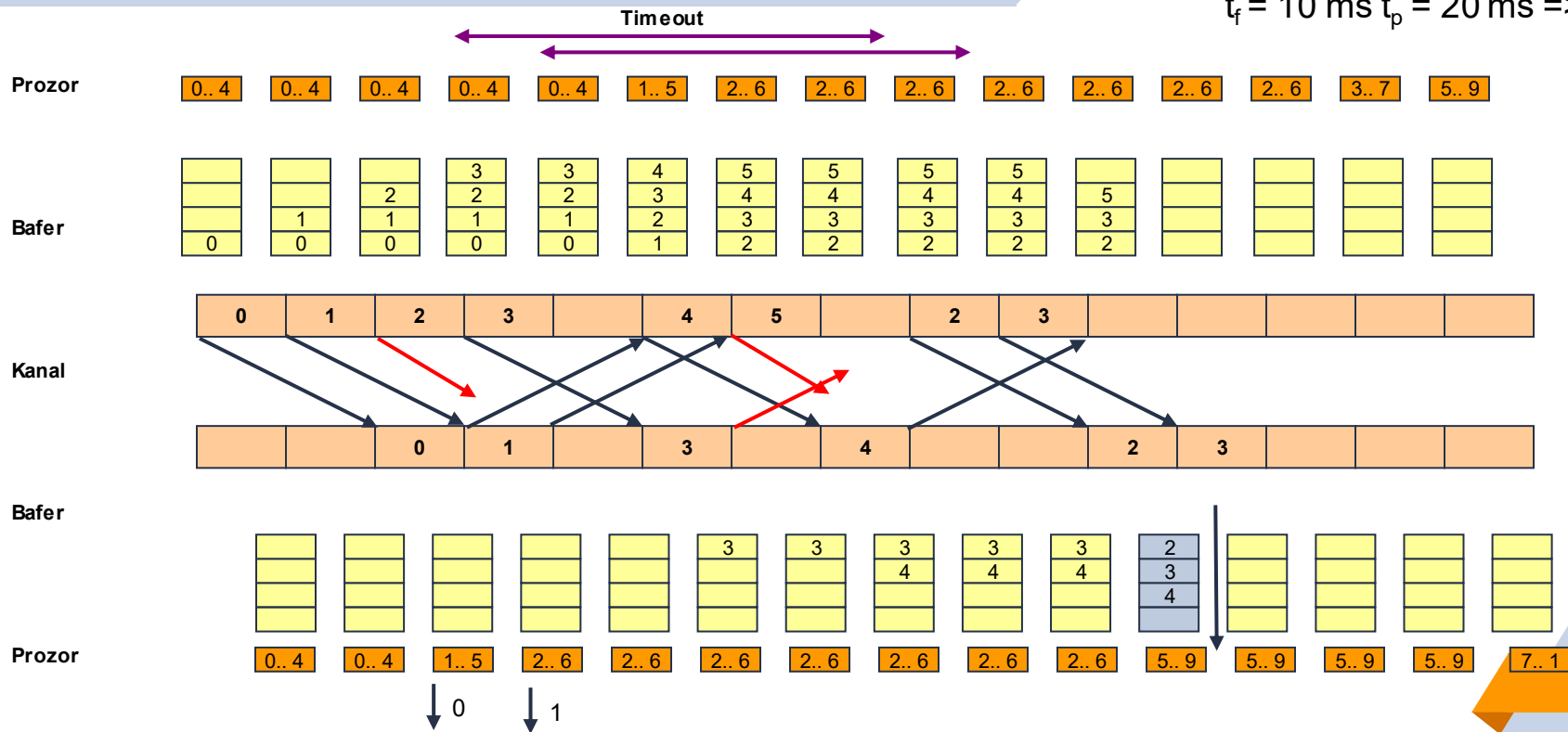
# Zadatak 1 – Rešenje

$$t_f = 10 \text{ ms } t_p = 20 \text{ ms} \Rightarrow a = 2$$



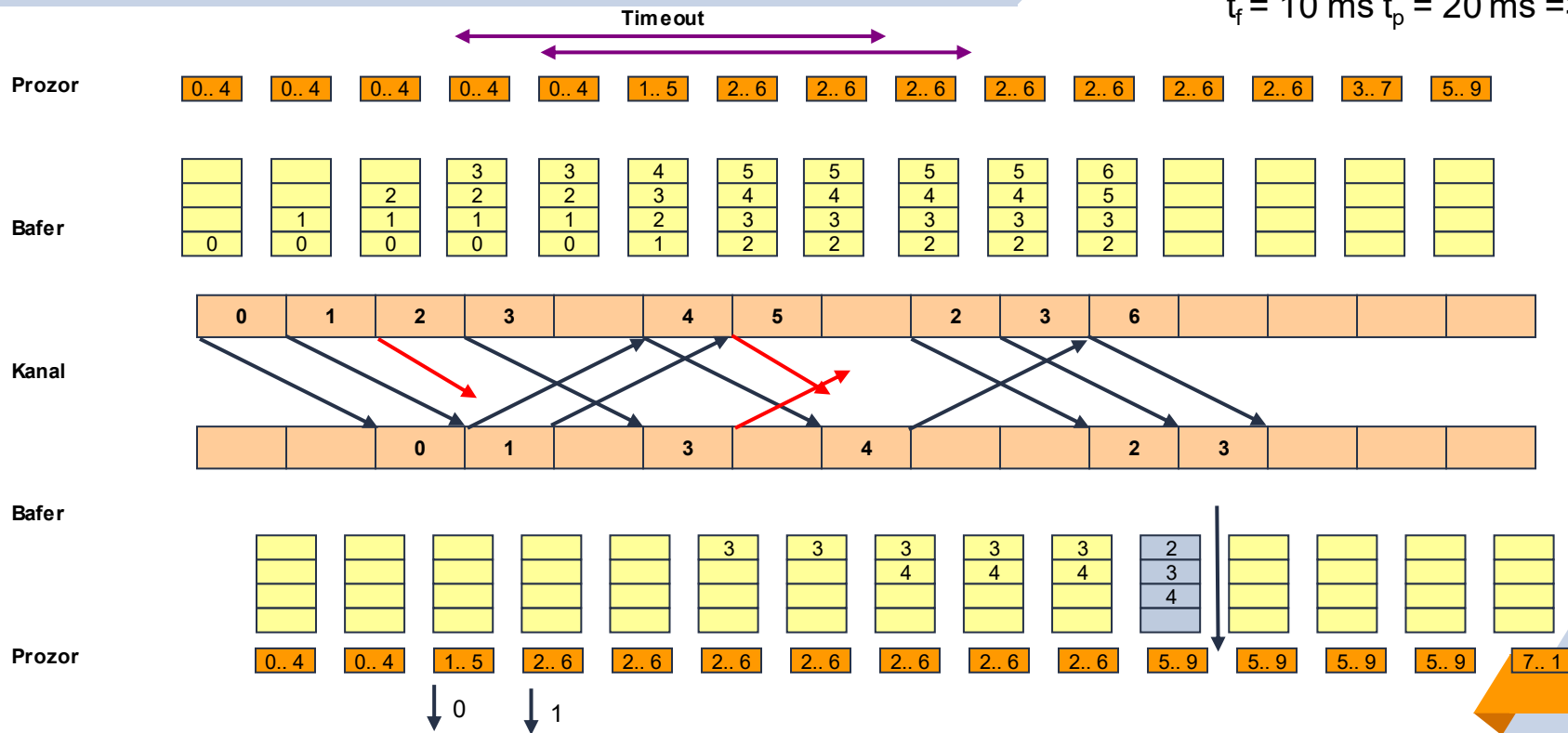
# Zadatak 1 – Rešenje

$$t_f = 10 \text{ ms } t_p = 20 \text{ ms} \Rightarrow a = 2$$



# Zadatak 1 – Rešenje

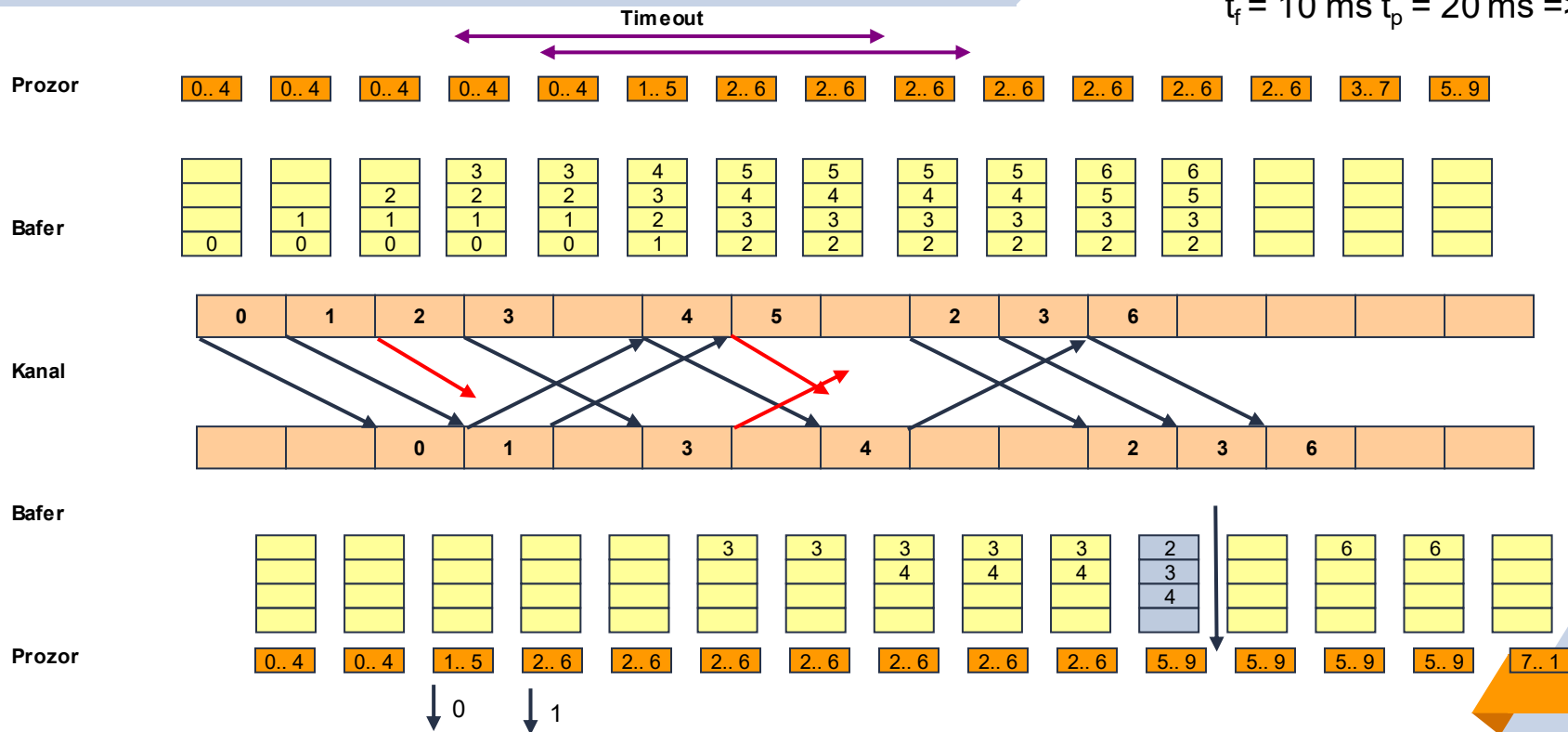
$$t_f = 10 \text{ ms } t_p = 20 \text{ ms} \Rightarrow a = 2$$





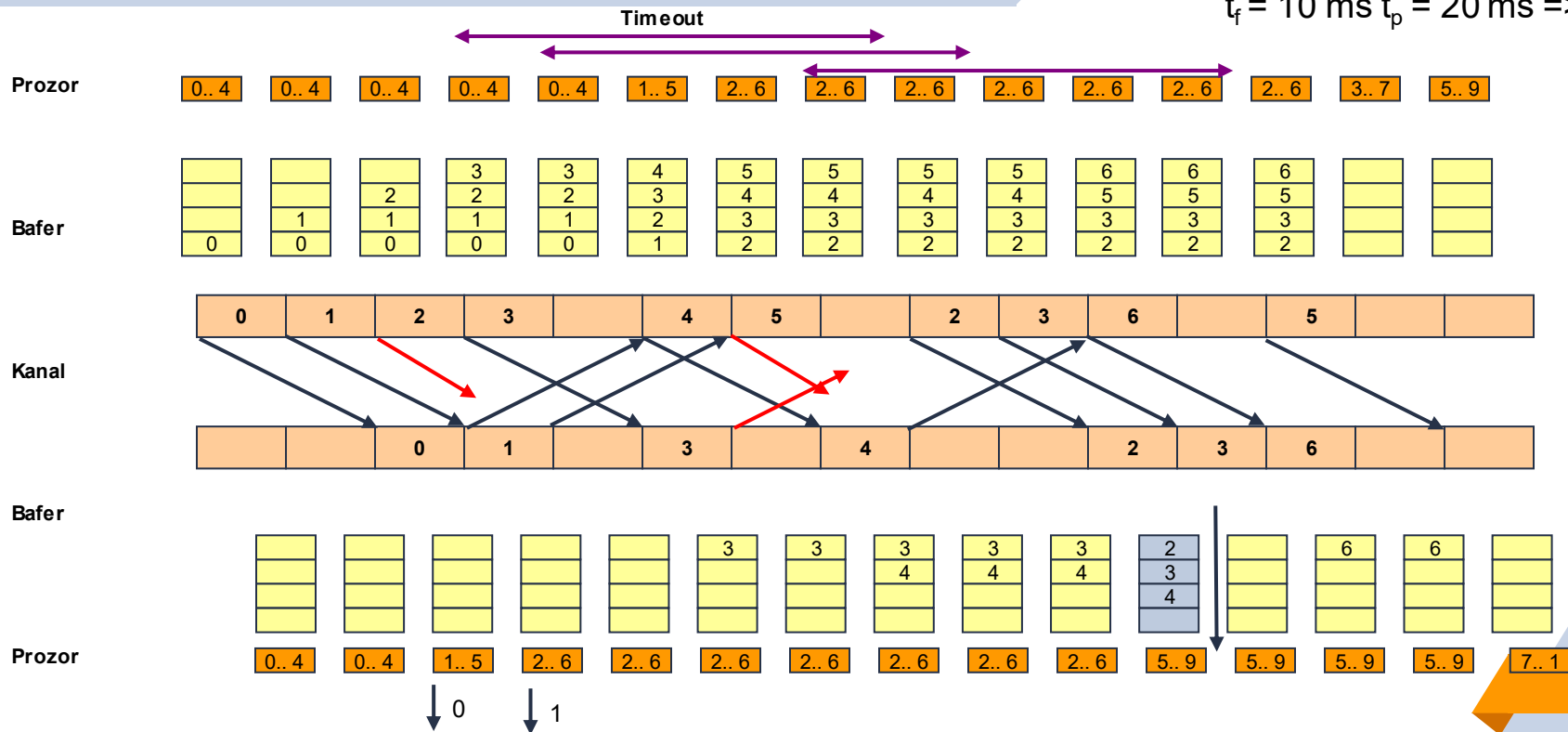
# Zadatak 1 – Rešenje

$$t_f = 10 \text{ ms } t_p = 20 \text{ ms} \Rightarrow a = 2$$



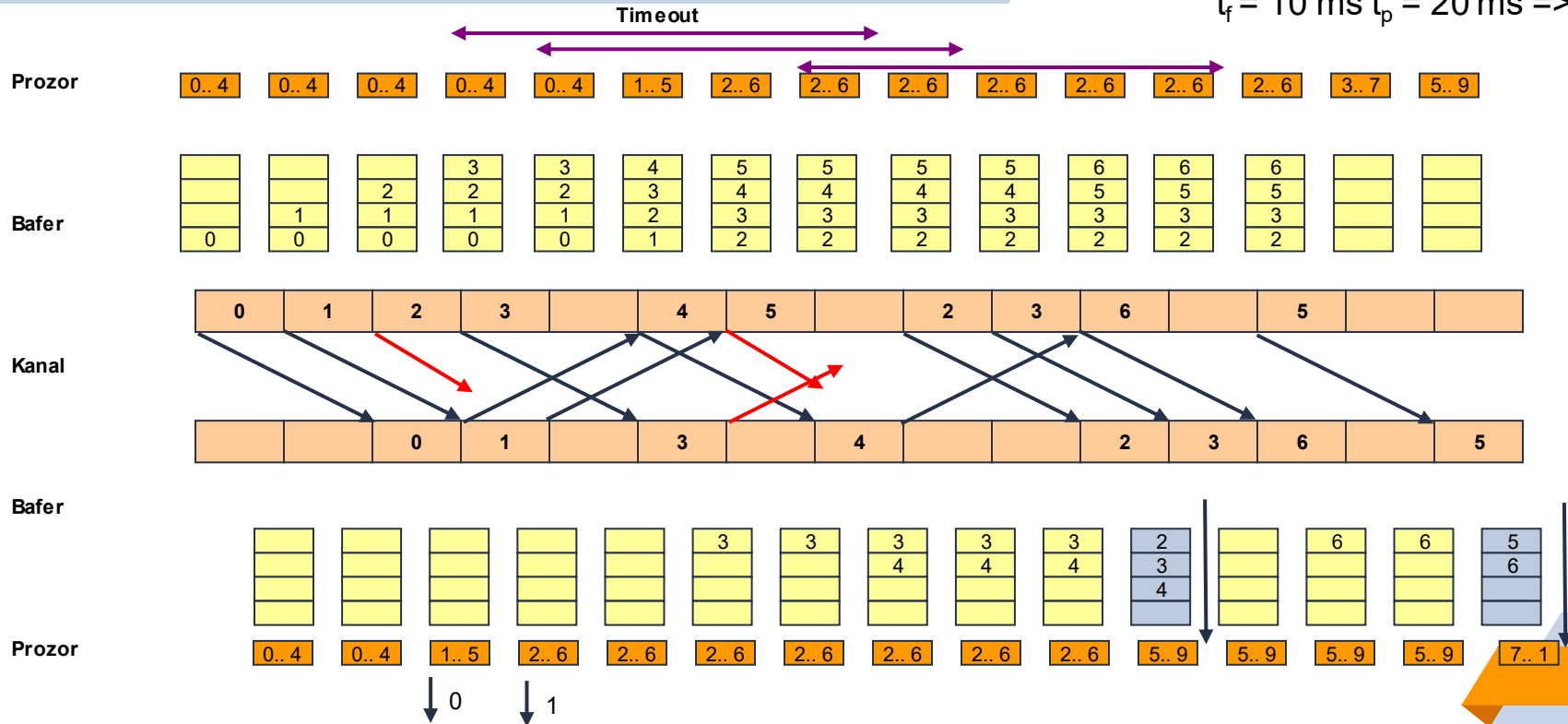
# Zadatak 1 – Rešenje

$$t_f = 10 \text{ ms } t_p = 20 \text{ ms} \Rightarrow a = 2$$



# Zadatak 1 – Rešenje

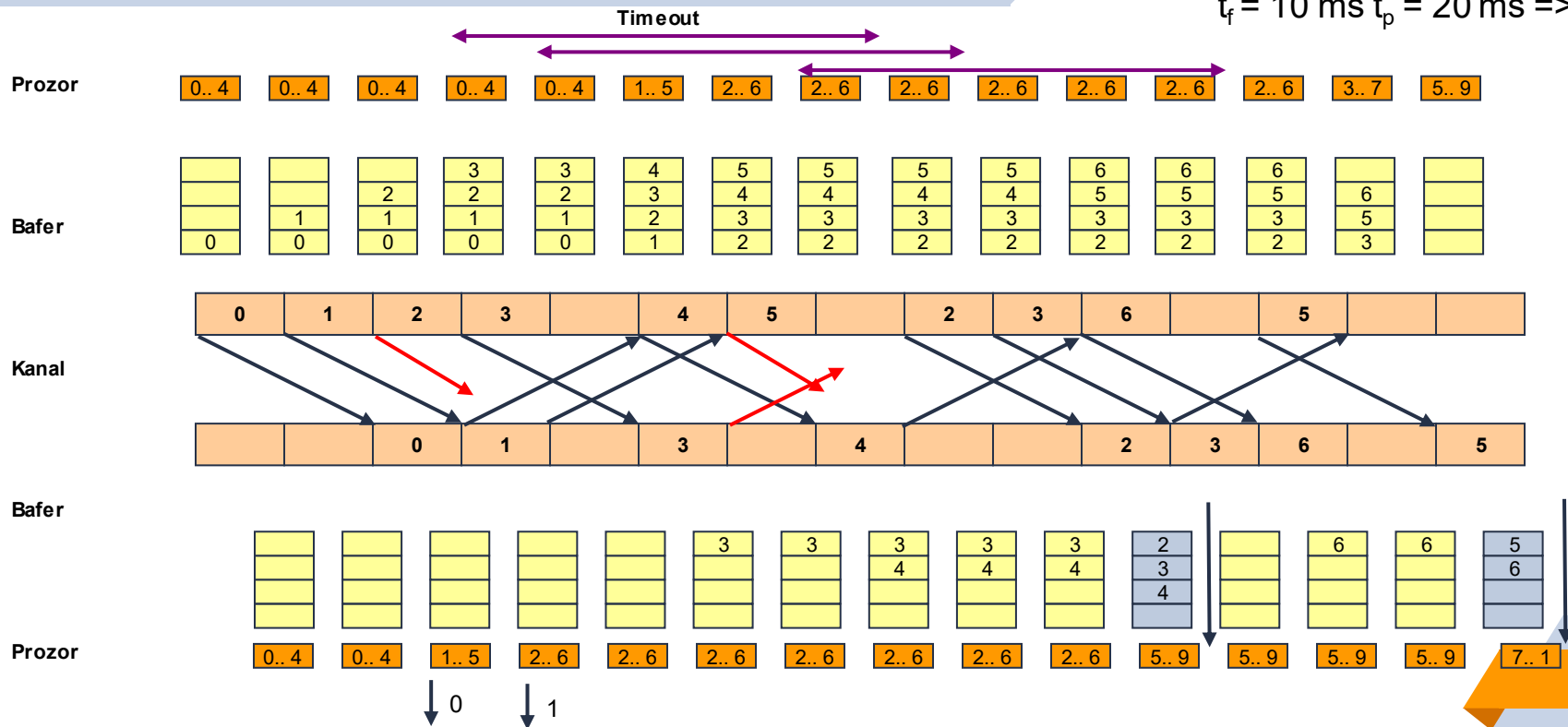
$$t_f = 10 \text{ ms } t_p = 20 \text{ ms} \Rightarrow a = 2$$





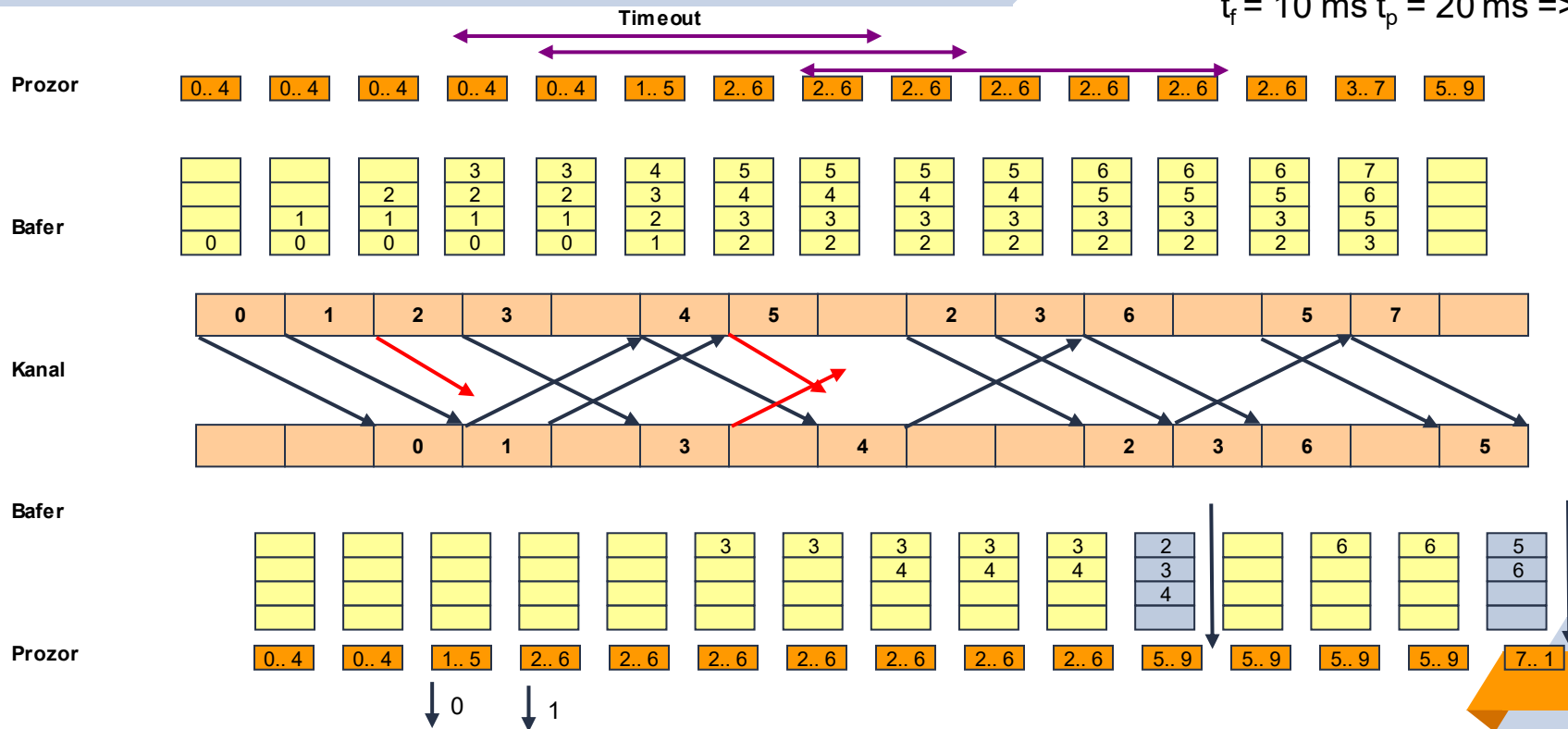
# Zadatak 1 – Rešenje

$$t_f = 10 \text{ ms } t_p = 20 \text{ ms} \Rightarrow a = 2$$



# Zadatak 1 – Rešenje

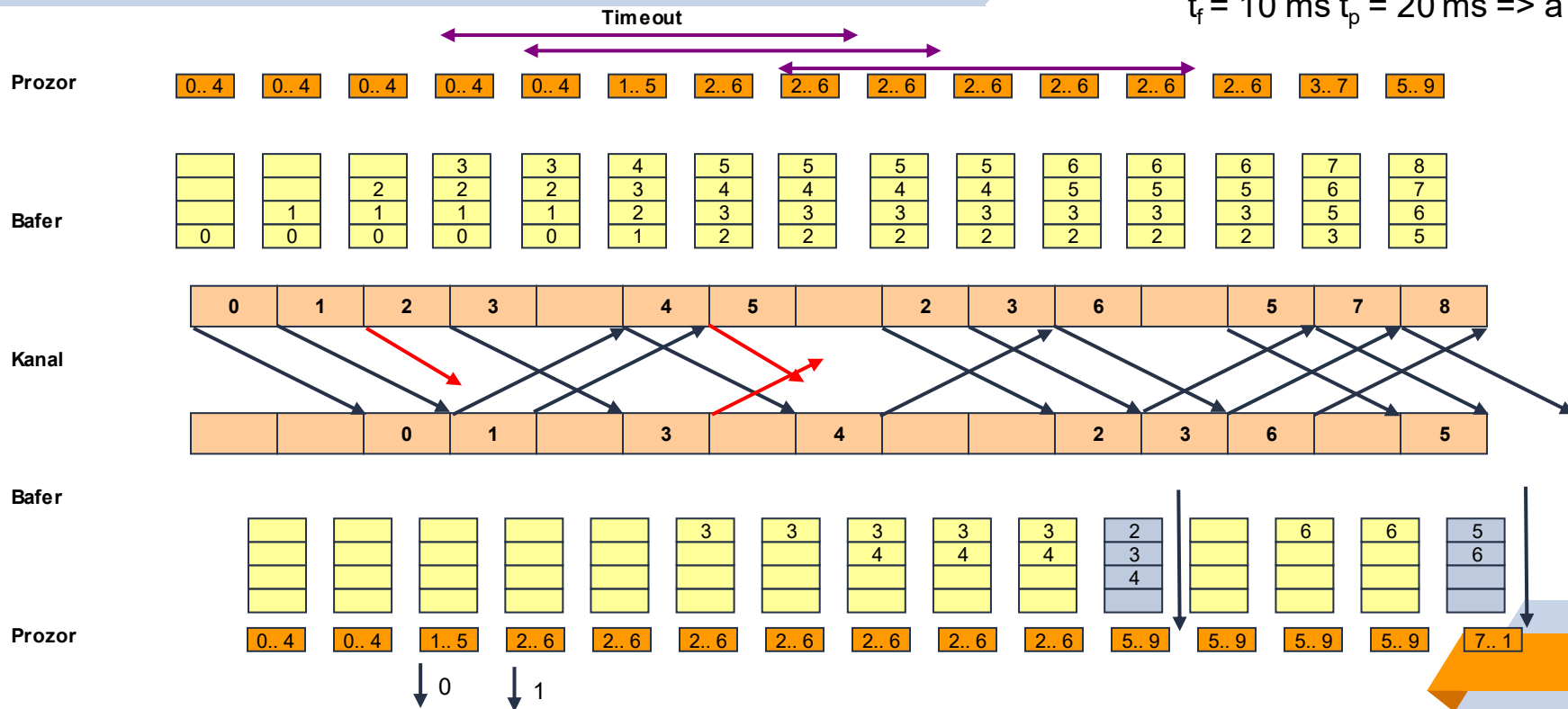
$$t_f = 10 \text{ ms } t_p = 20 \text{ ms} \Rightarrow a = 2$$





# Zadatak 1 – Rešenje

$$t_f = 10 \text{ ms } t_p = 20 \text{ ms} \Rightarrow a = 2$$



## Zadatak 2 - Rešenje

a)

$$U \geq 0.5$$

$$w / (2a + 1) \geq 0.5; w \geq (2a+1)*0.5$$

$$\Rightarrow w \geq a + 0.5 \Rightarrow a \leq 6.5$$

$$t_p = 10^6 / 2*10^8 = 5 \text{ ms}$$

$$t_f = 10^3 / v_s \geq t_p / 6.5 \Rightarrow \mathbf{v_s \leq 1.3 \text{ Mb/s}}$$

## Zadatak 2 - Rešenje

b)

0..6

0..6

1..7

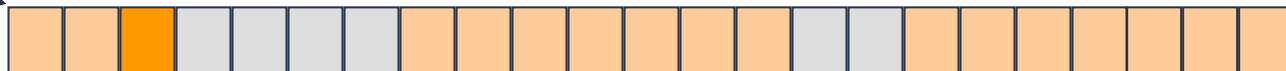
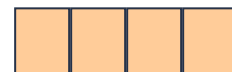
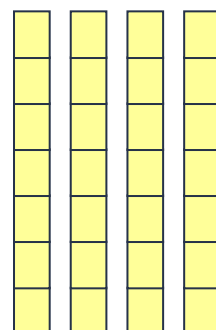
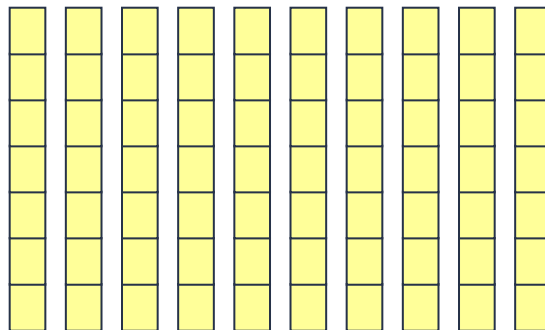
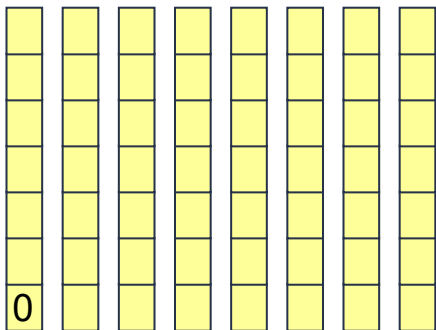
2..0

2..0

2..0

3..1

5..3



## Zadatak 2 - Rešenje

b)

0..6

0..6

1..7

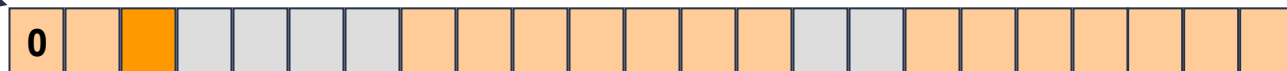
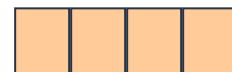
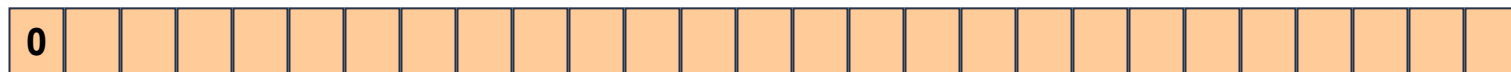
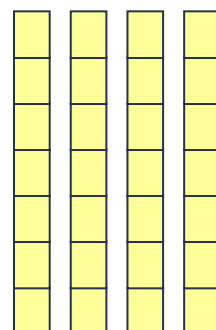
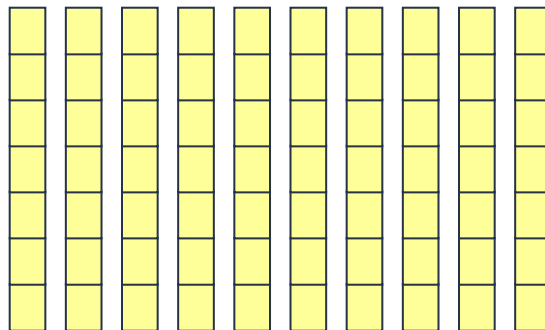
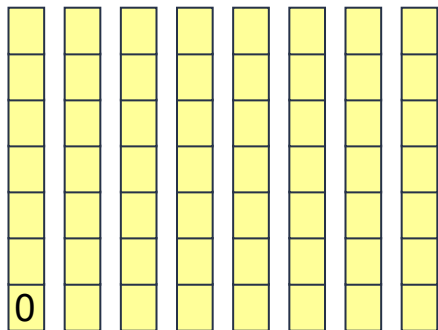
2..0

2..0

2..0

3..1

5..3



## Zadatak 2 - Rešenje

b)

0..6

0..6

1..7

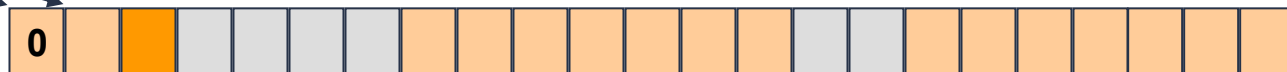
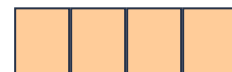
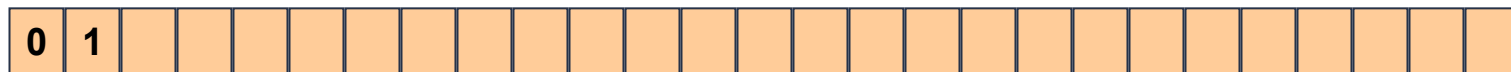
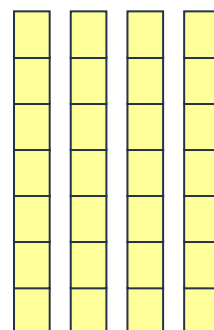
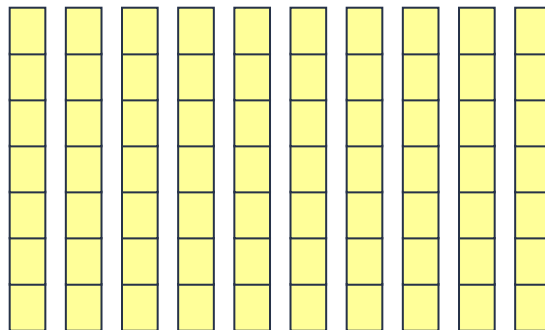
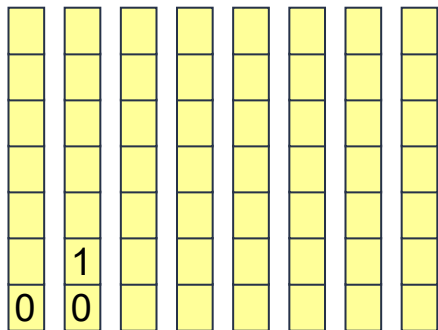
2..0

2..0

2..0

3..1

5..3





## Zadatak 2 - Rešenje

b)

0..6

0..6

1..7

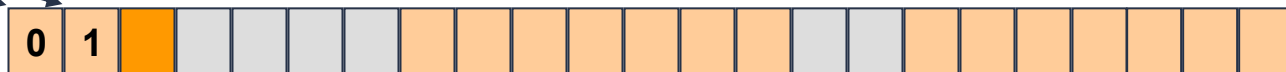
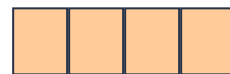
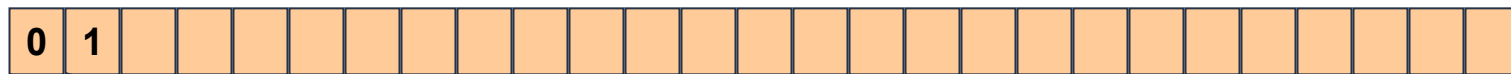
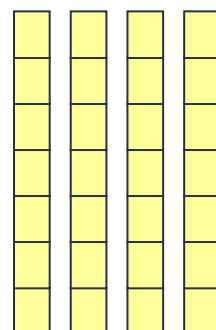
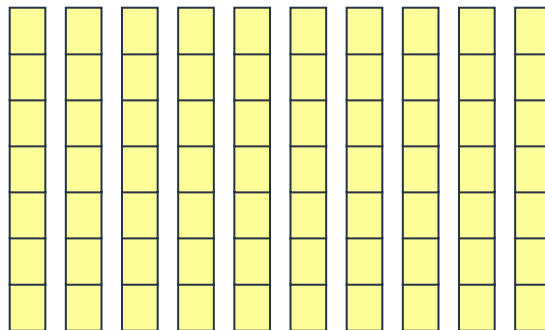
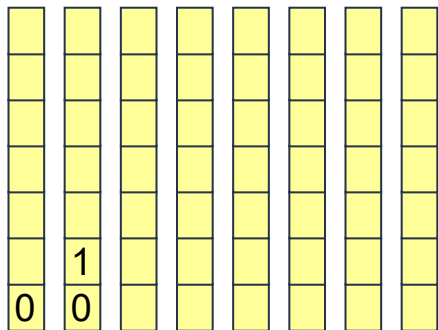
2..0

2..0

2..0

3..1

5..3



## Zadatak 2 - Rešenje

b)

0..6

0..6

1..7

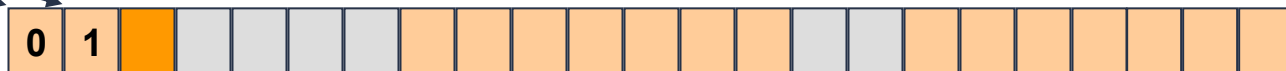
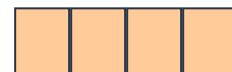
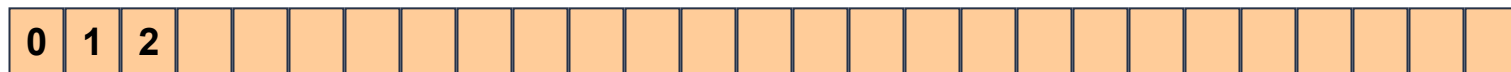
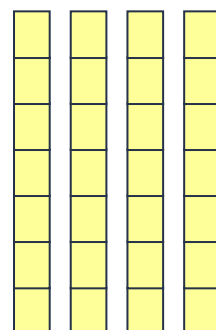
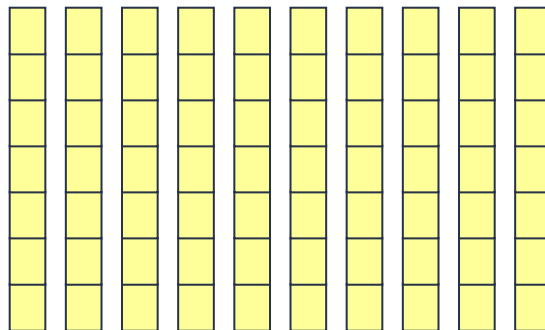
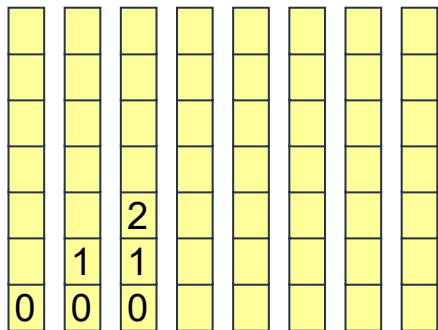
2..0

2..0

2..0

3..1

5..3



## Zadatak 2 - Rešenje

b)

0..6

0..6

1..7

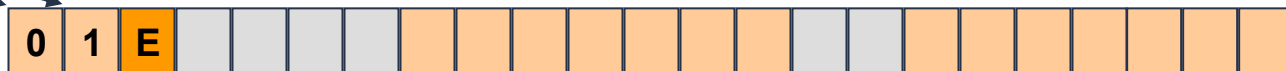
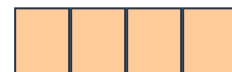
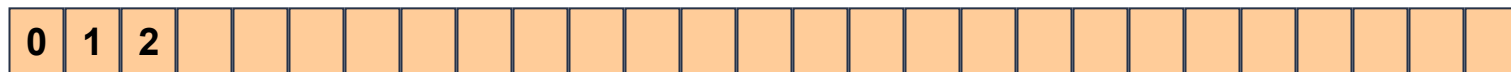
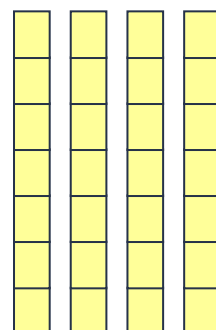
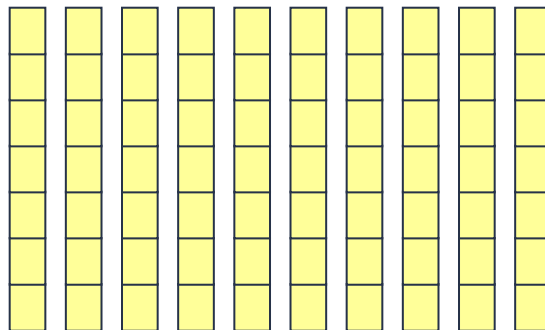
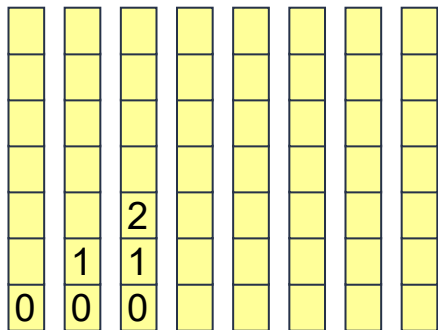
2..0

2..0

2..0

3..1

5..3



## Zadatak 2 - Rešenje

b)

0.. 6

0.. 6

1..7

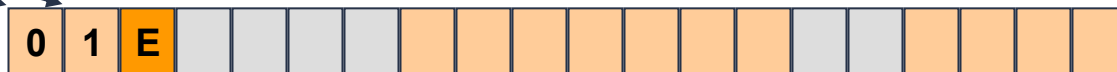
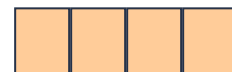
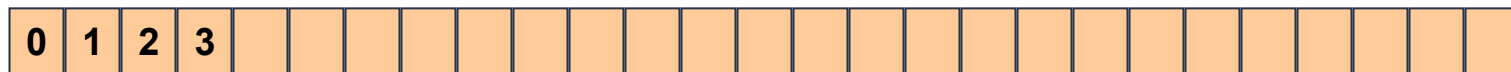
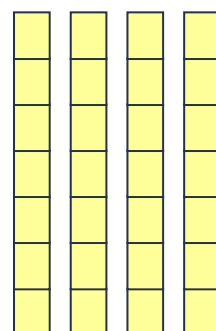
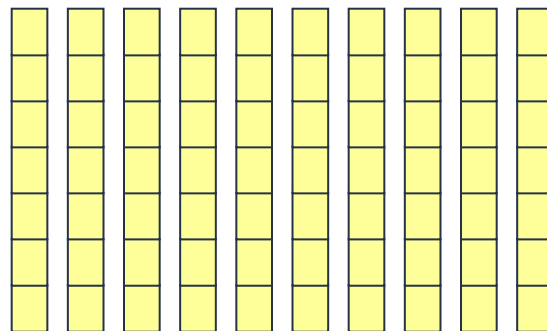
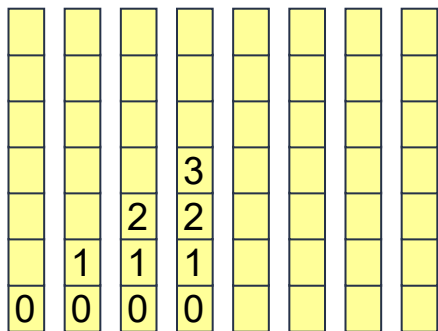
2.. 0

2.. 0

2.. 0

3.. 1

5.. 3



## Zadatak 2 - Rešenje

b)

0.. 6

0.. 6

1..7

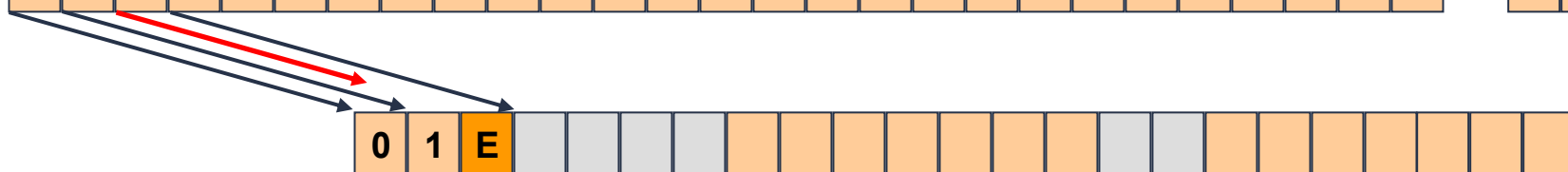
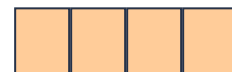
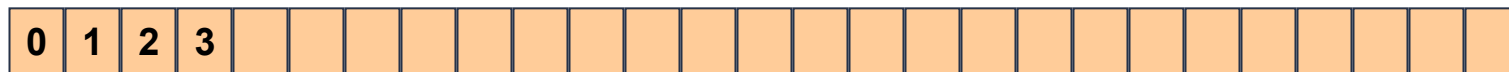
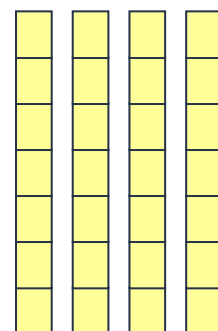
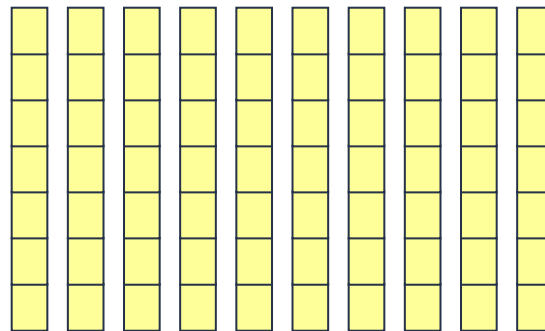
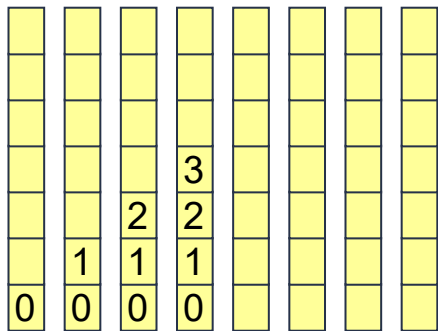
2.. 0

2.. 0

2.. 0

3.. 1

5.. 3



## Zadatak 2 - Rešenje

b)

0..6

0..6

1..7

2..0

2..0

2..0

3..1

5..3

			3				
		2	2				
	1	1	1				
0	0	0	0				



0	1	2	3																							
---	---	---	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

--	--	--	--

0	1	E	D																							
---	---	---	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

## Zadatak 2 - Rešenje

b)

0.. 6

0.. 6

1..7

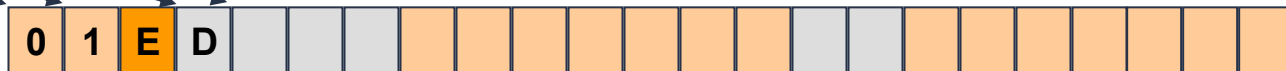
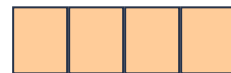
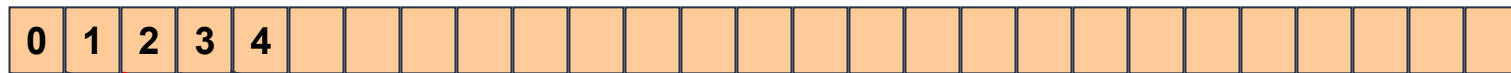
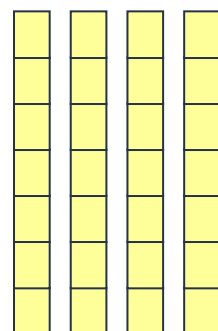
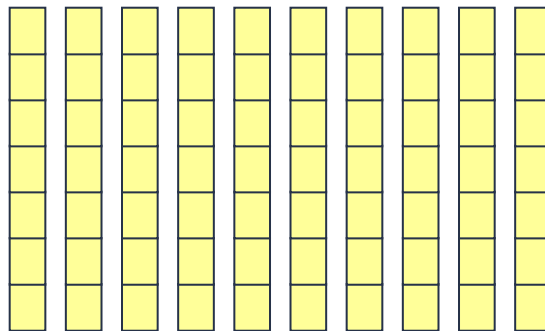
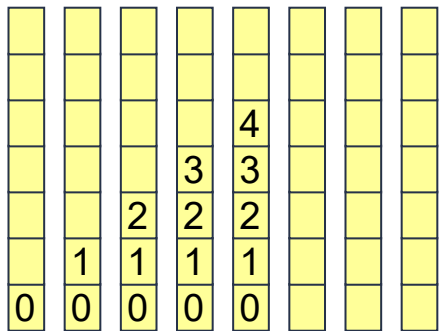
2.. 0

2.. 0

2.. 0

3.. 1

5.. 3



## Zadatak 2 - Rešenje

b)

0.. 6

0.. 6

1..7

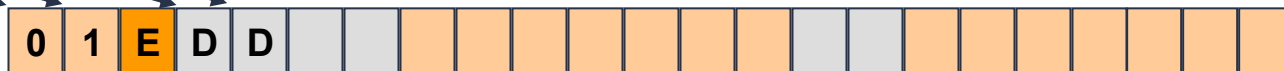
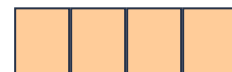
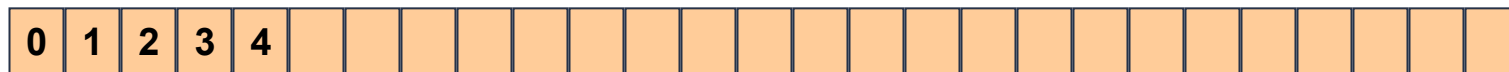
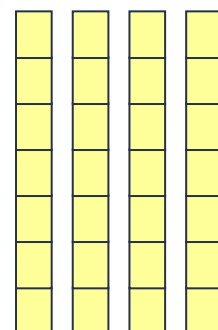
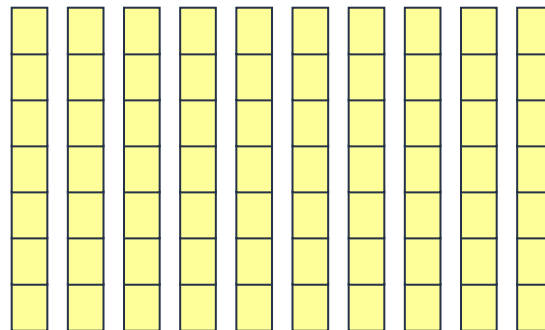
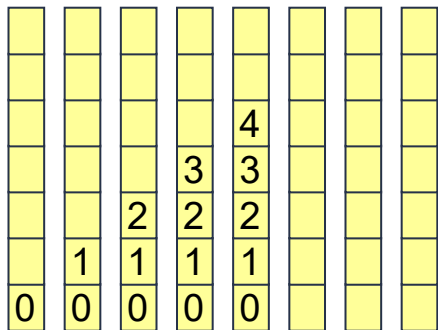
2.. 0

2.. 0

2.. 0

3.. 1

5.. 3





## Zadatak 2 - Rešenje

b)

0..6

0..6

1..7

2..0

2..0

2..0

3..1

5..3

					5		
				4	4		
			3	3	3		
		2	2	2	2		
	1	1	1	1	1		
0	0	0	0	0	0		



0	1	2	3	4	5																								
---	---	---	---	---	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

--	--	--	--

0	1	E	D	D																								
---	---	---	---	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

## Zadatak 2 - Rešenje

b)

0.. 6

0.. 6

1..7

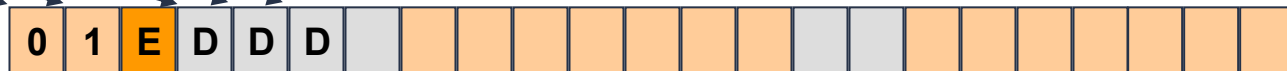
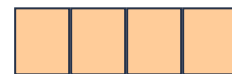
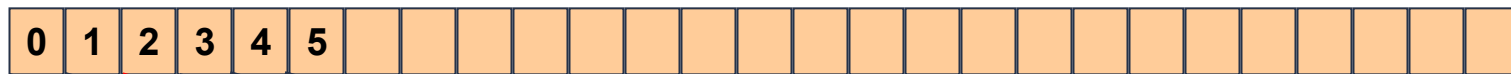
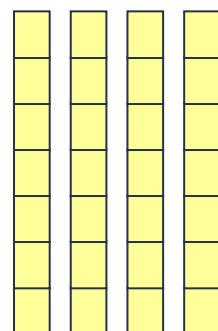
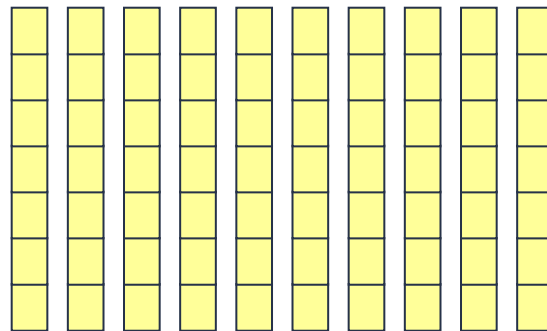
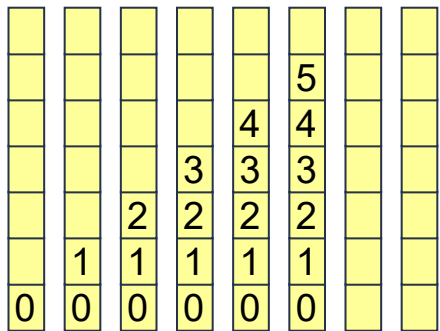
2.. 0

2.. 0

2.. 0

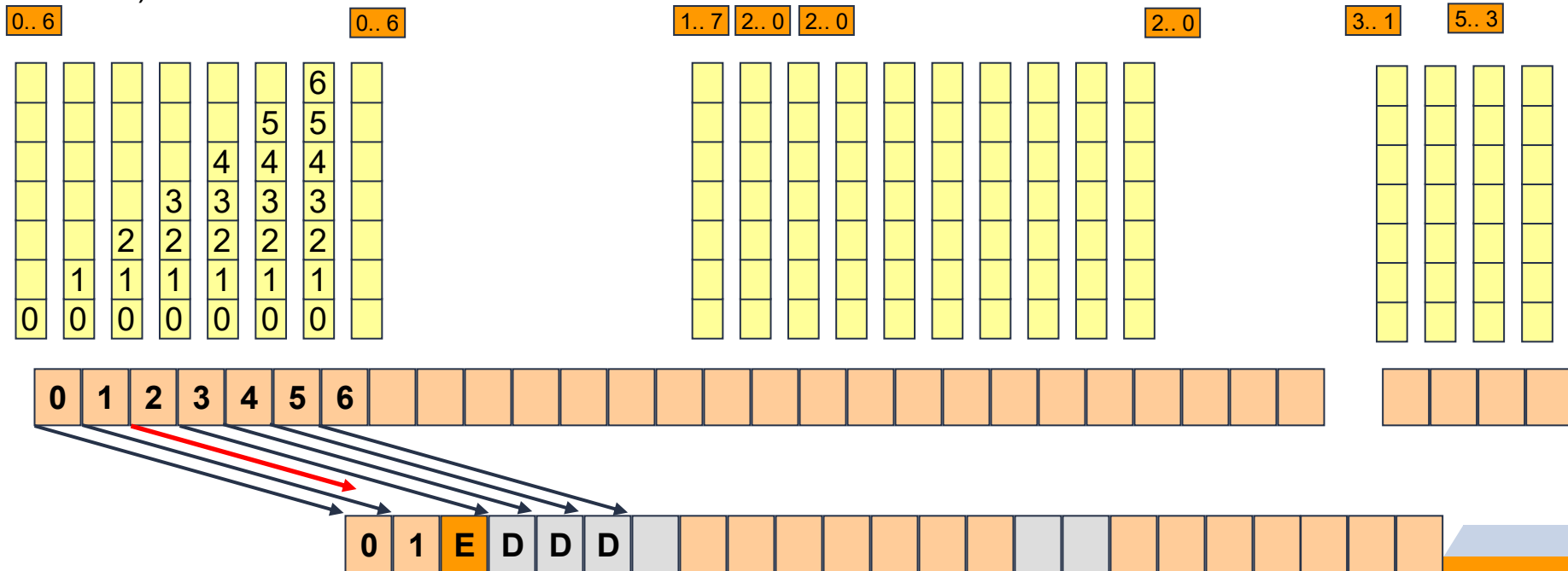
3.. 1

5.. 3



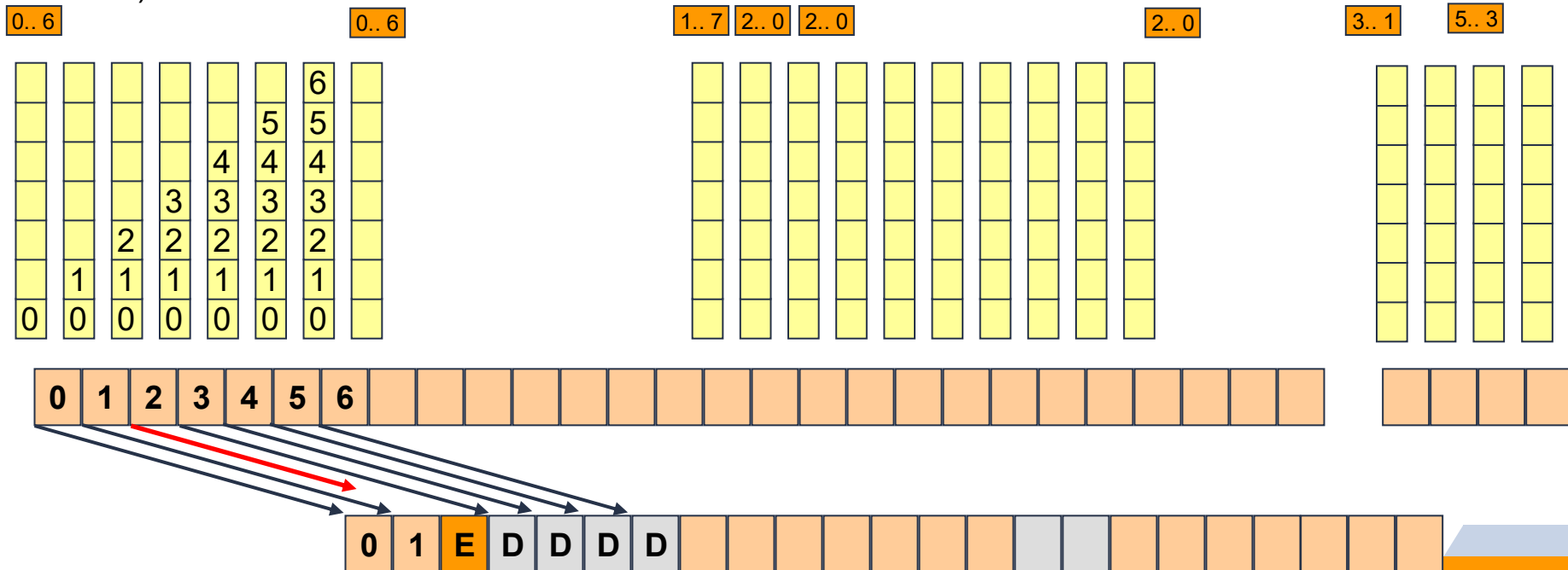
## Zadatak 2 - Rešenje

b)



## Zadatak 2 - Rešenje

b)



## Zadatak 2 - Rešenje

b)


0..6

0..6

0.. 6

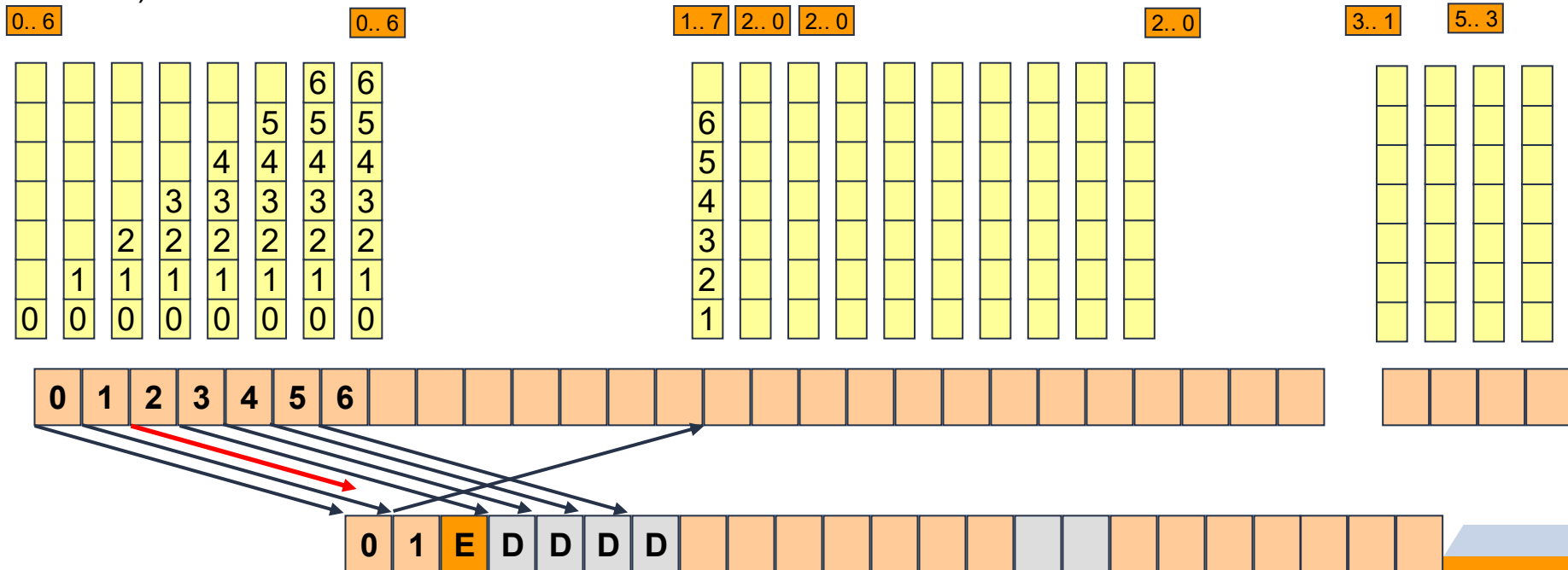
A 10x10 grid of yellow squares, representing 100 units. The grid is composed of 10 columns and 10 rows of squares. Each square is yellow with a black border. The grid is used to represent the total value of 100 units.

5.. 3

101

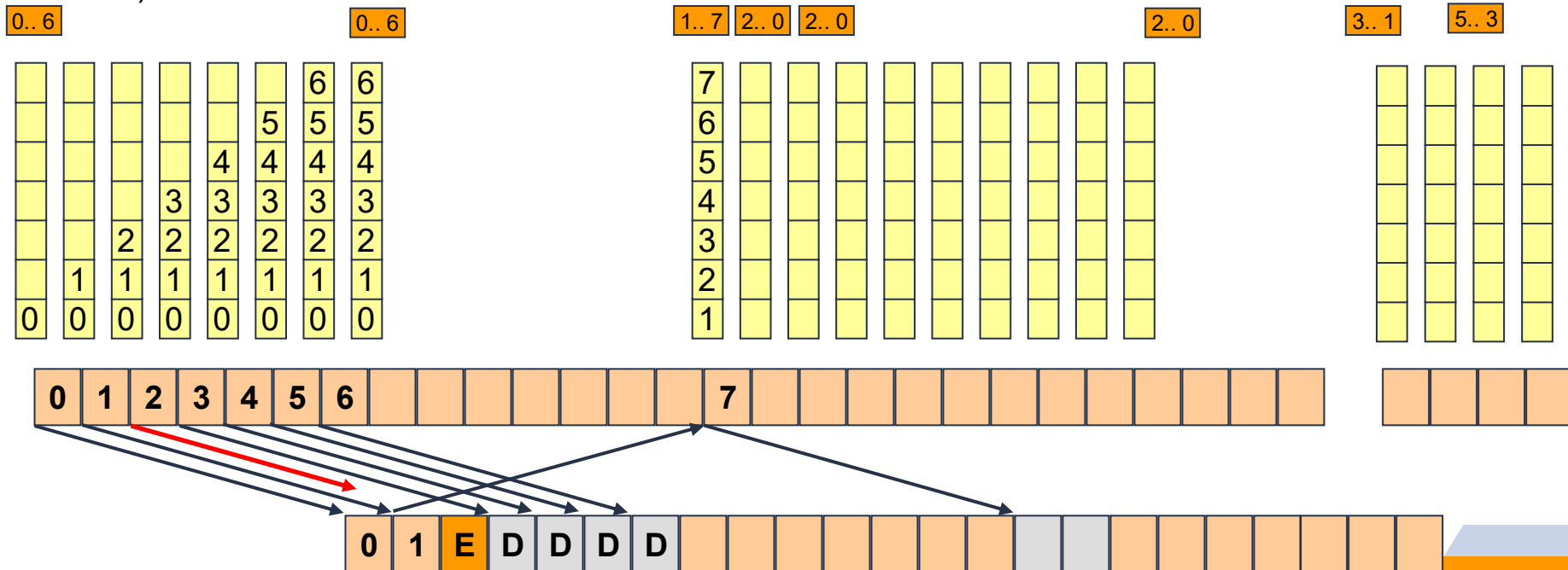
## Zadatak 2 - Rešenje

b)



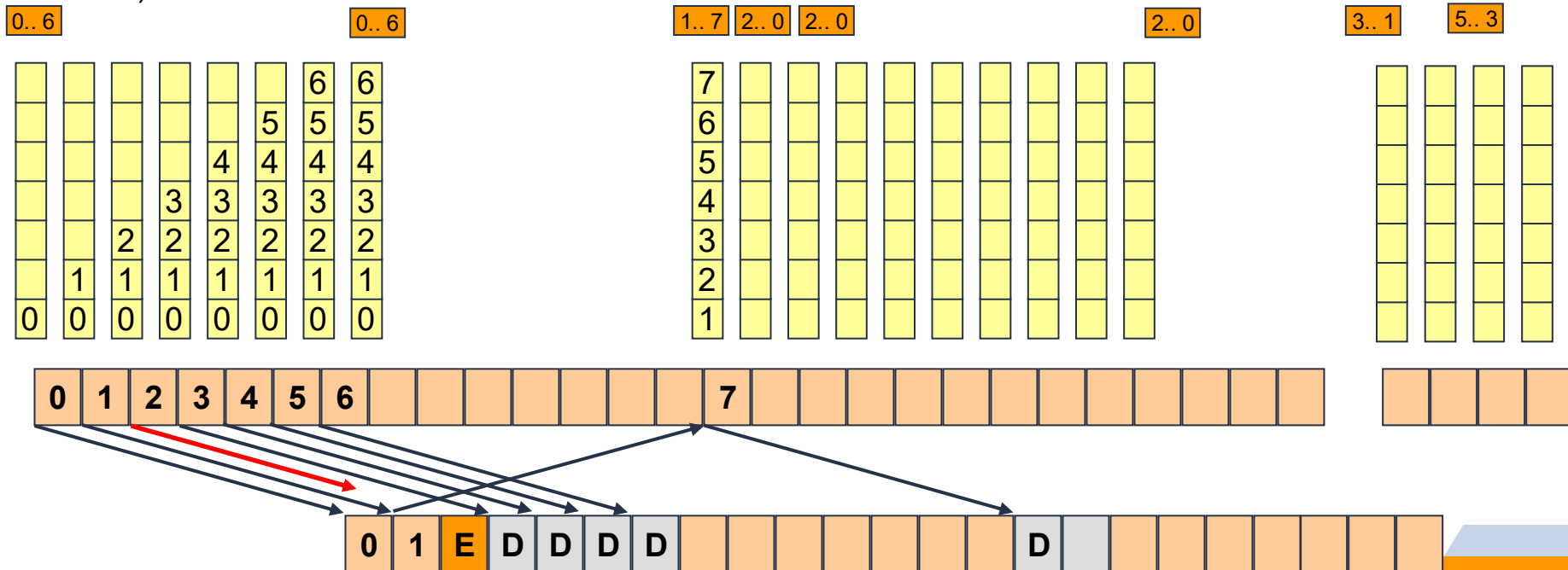
## Zadatak 2 - Rešenje

b)



## Zadatak 2 - Rešenje

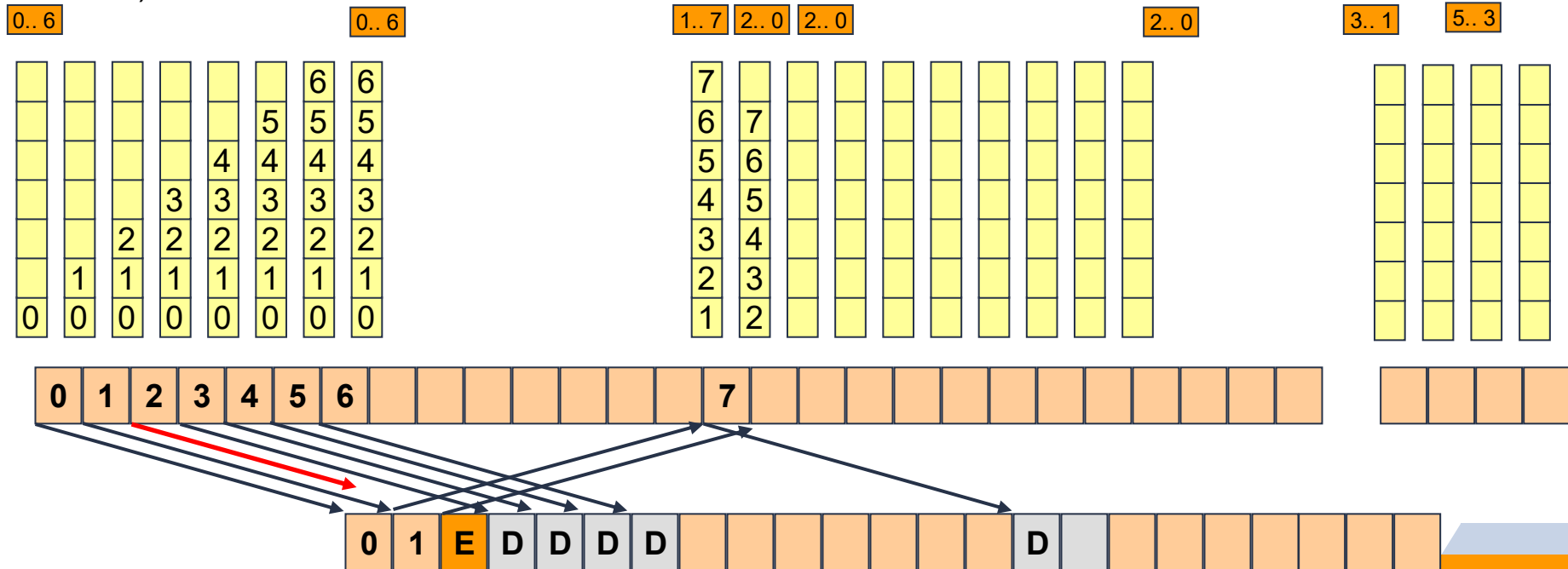
b)





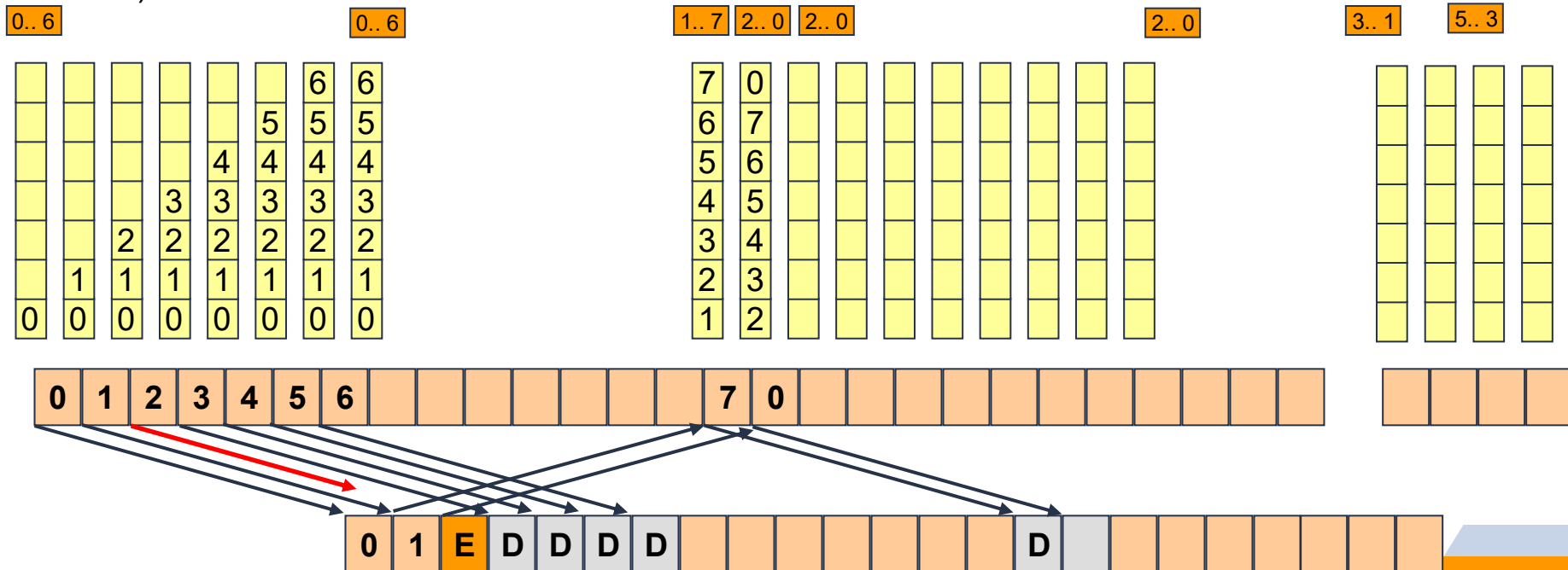
## Zadatak 2 - Rešenje

b)



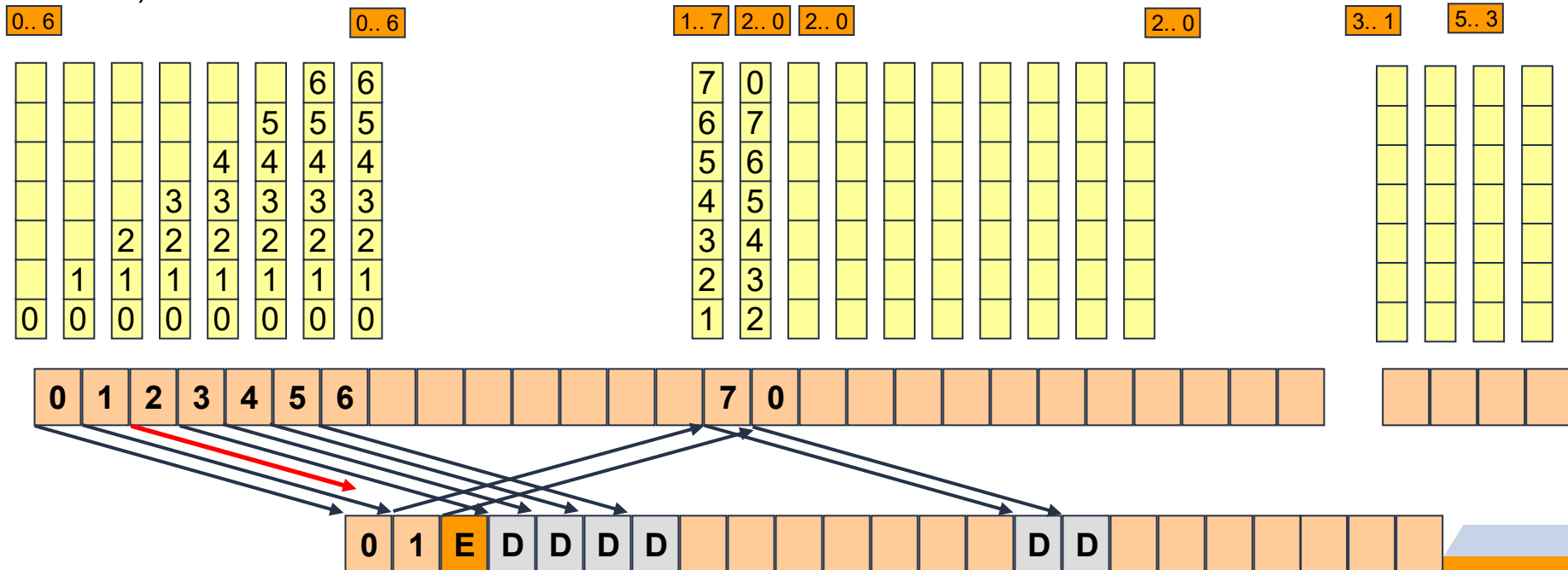
## Zadatak 2 - Rešenje

b)



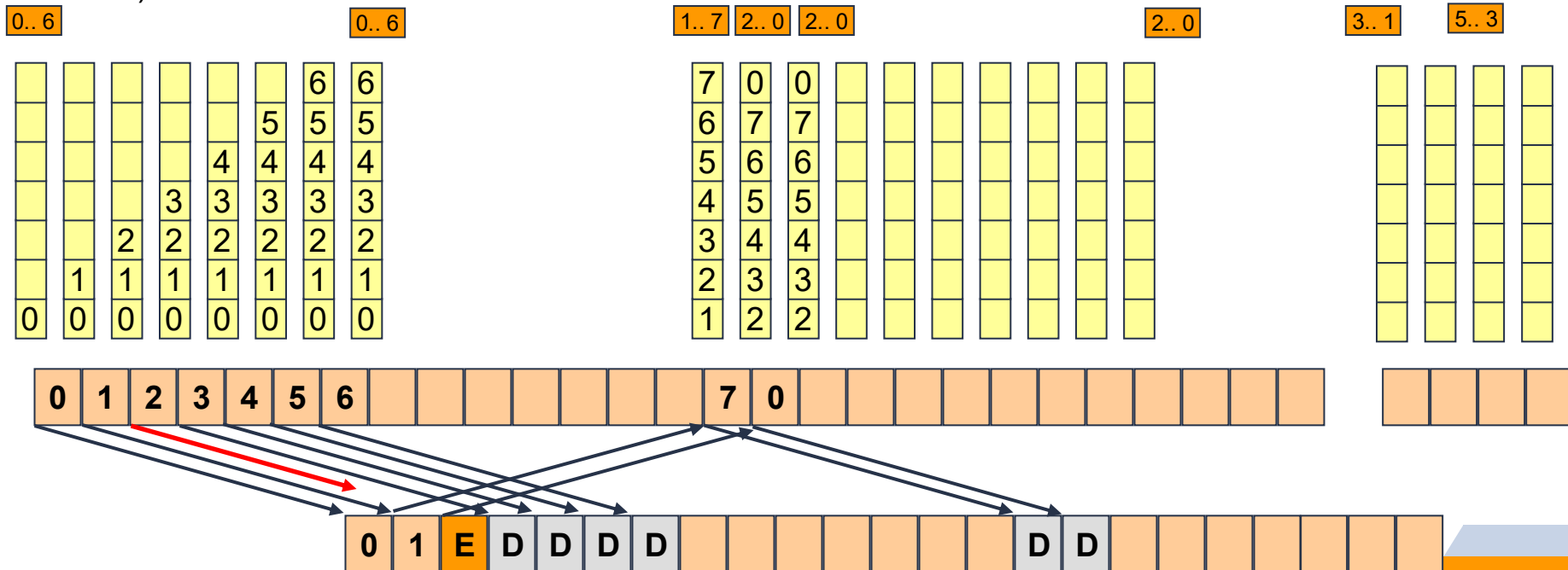
## Zadatak 2 - Rešenje

b)



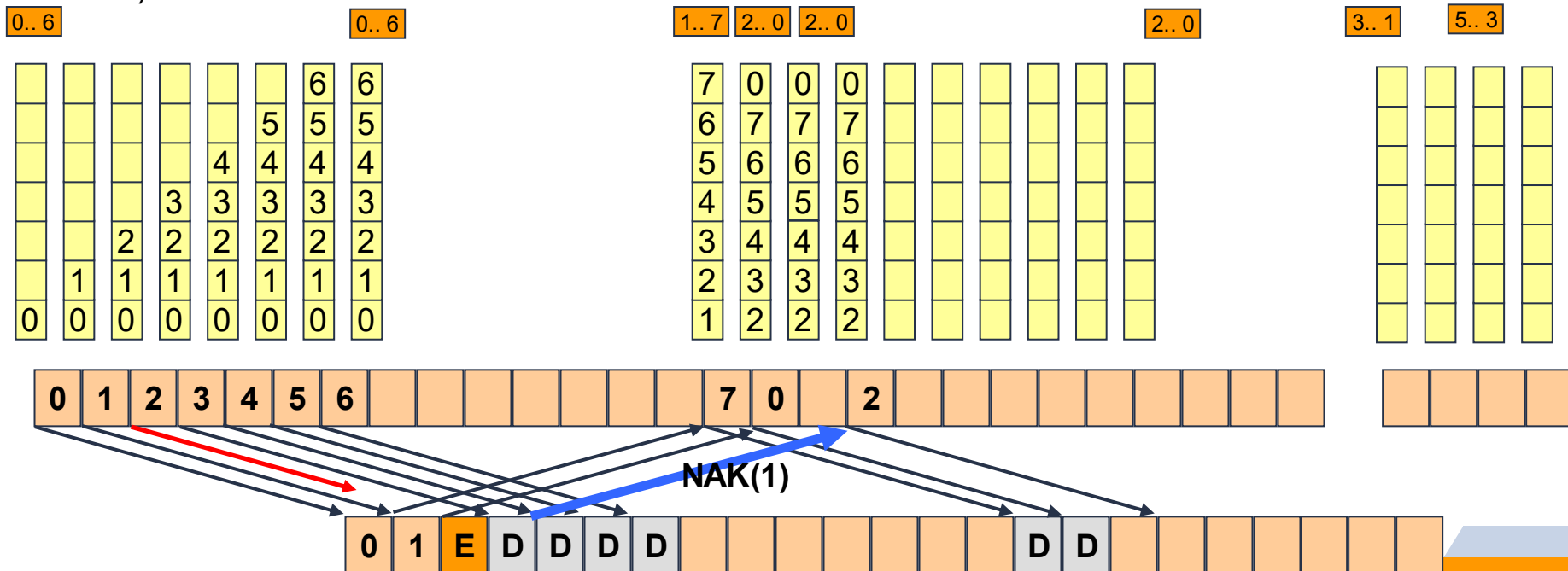
## Zadatak 2 - Rešenje

b)



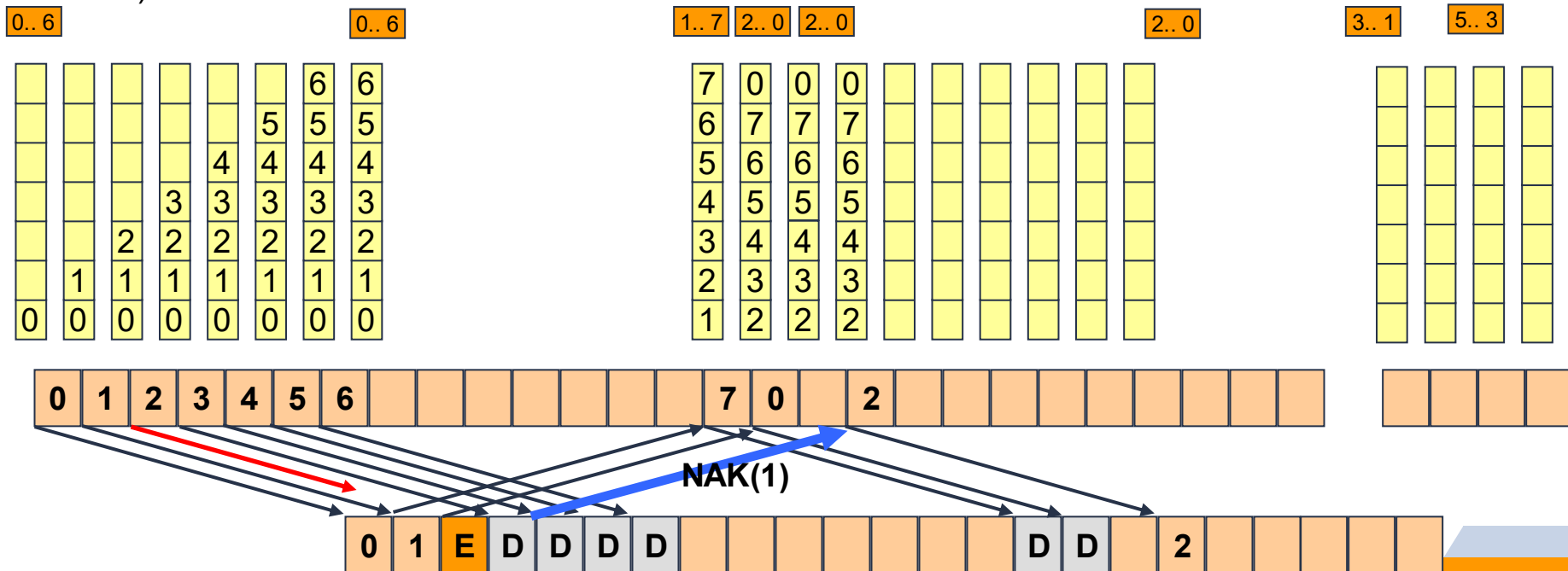
## Zadatak 2 - Rešenje

b)



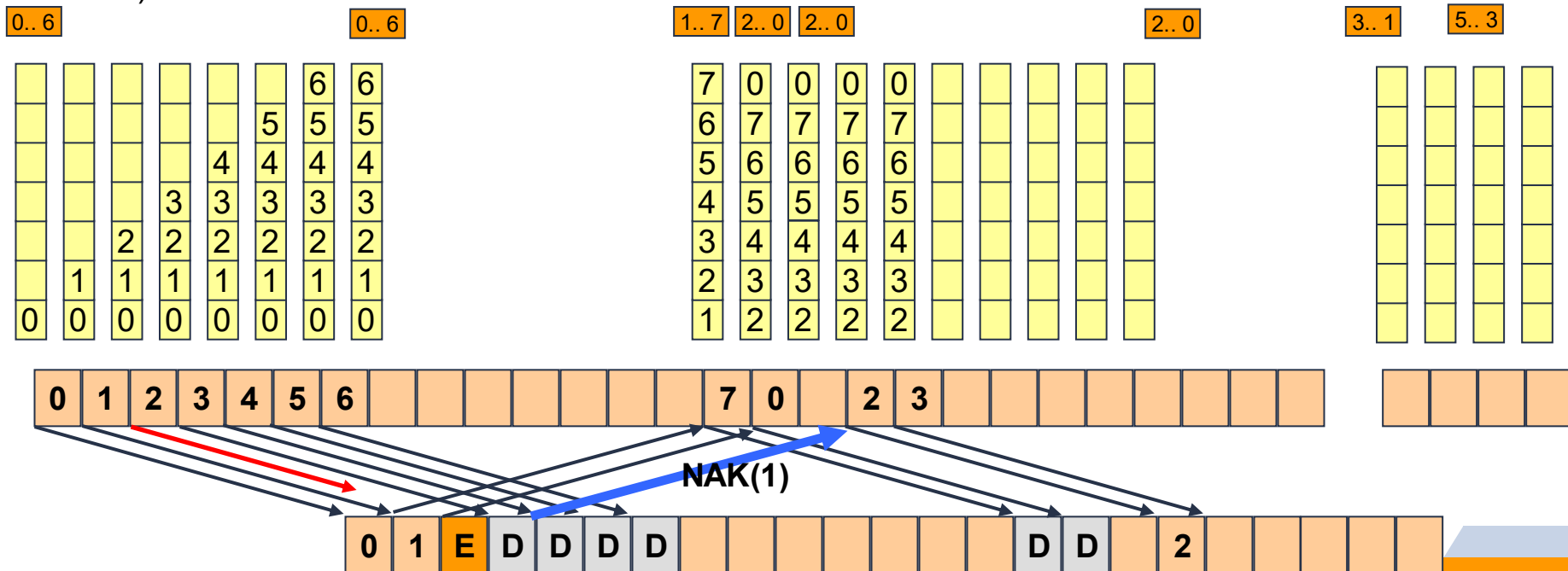
## Zadatak 2 - Rešenje

b)



## Zadatak 2 - Rešenje

b)

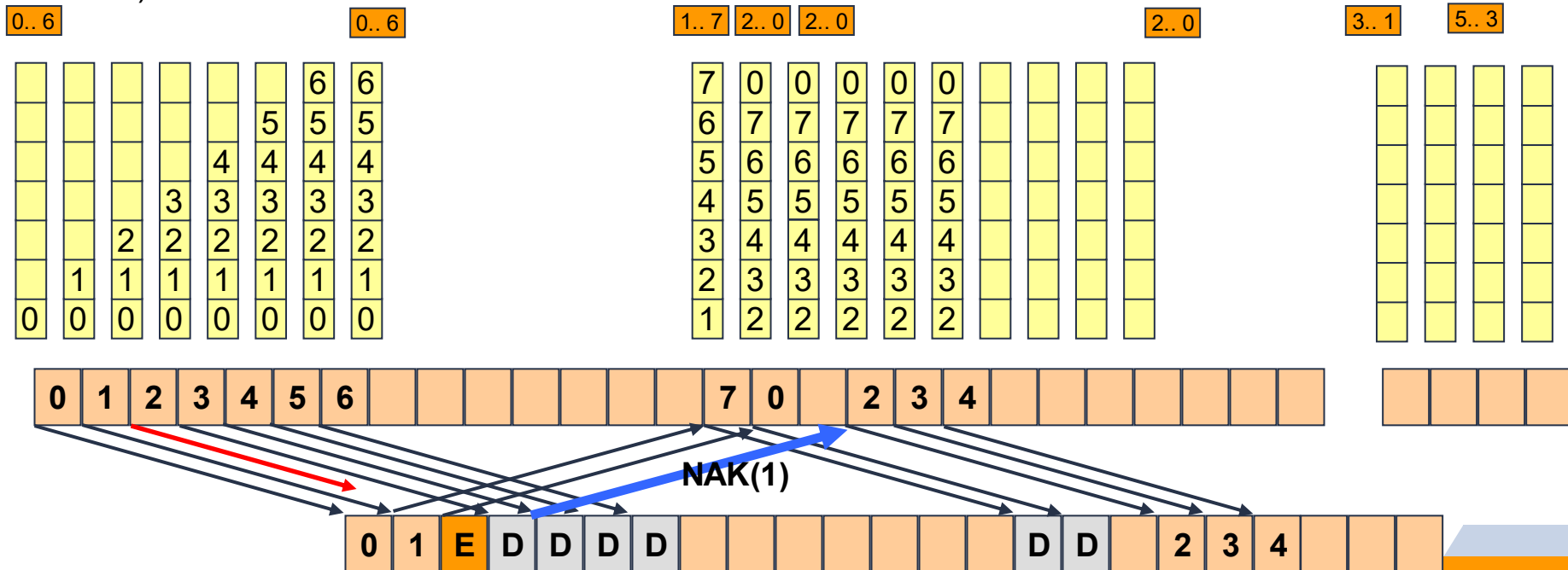






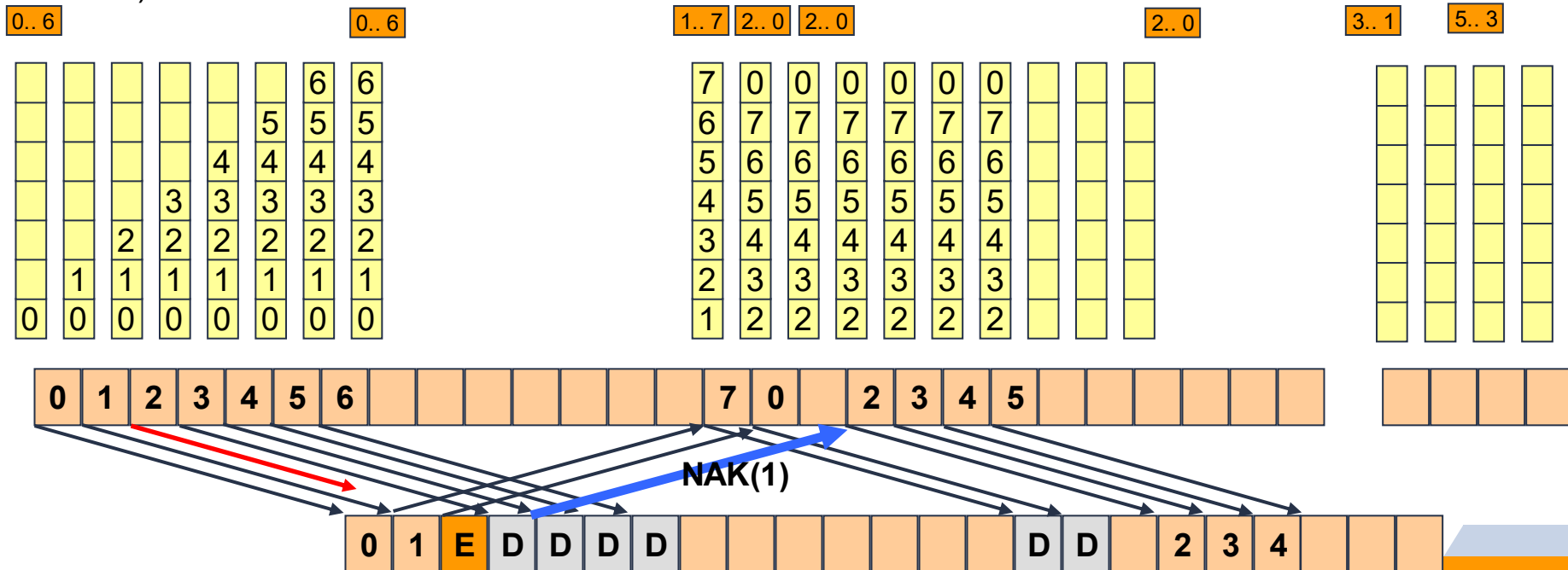
## Zadatak 2 - Rešenje

b)



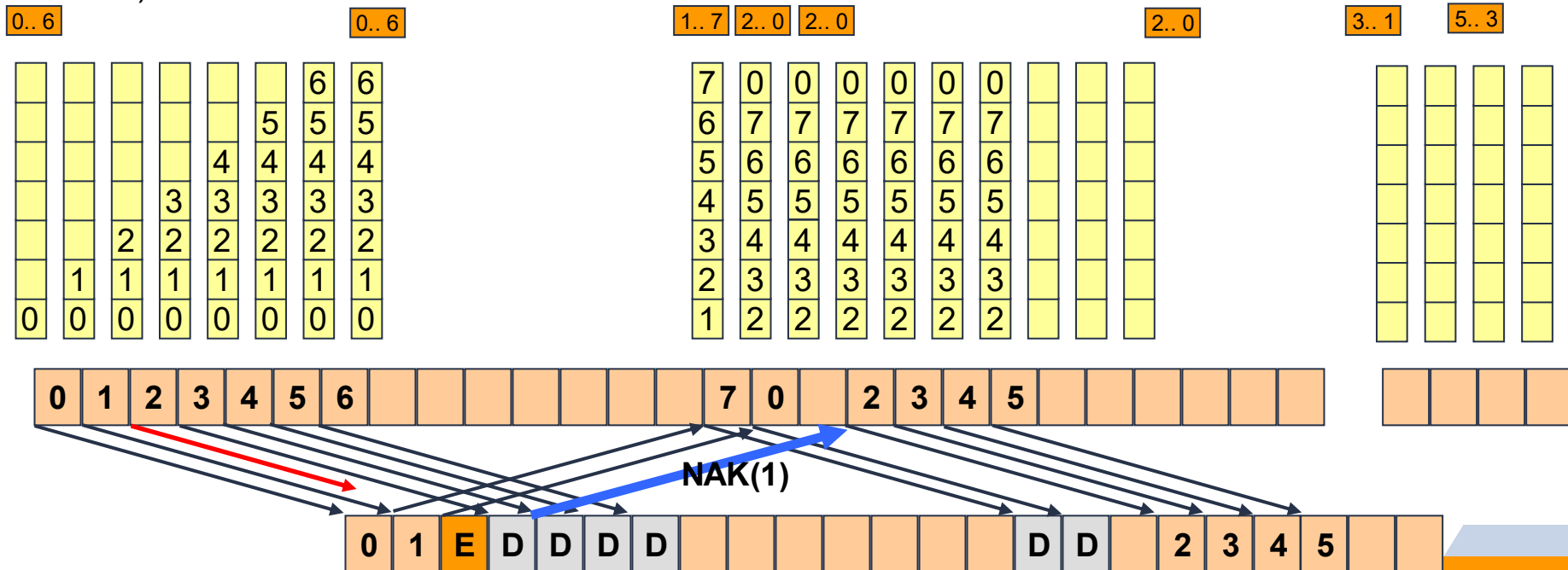
## Zadatak 2 - Rešenje

b)



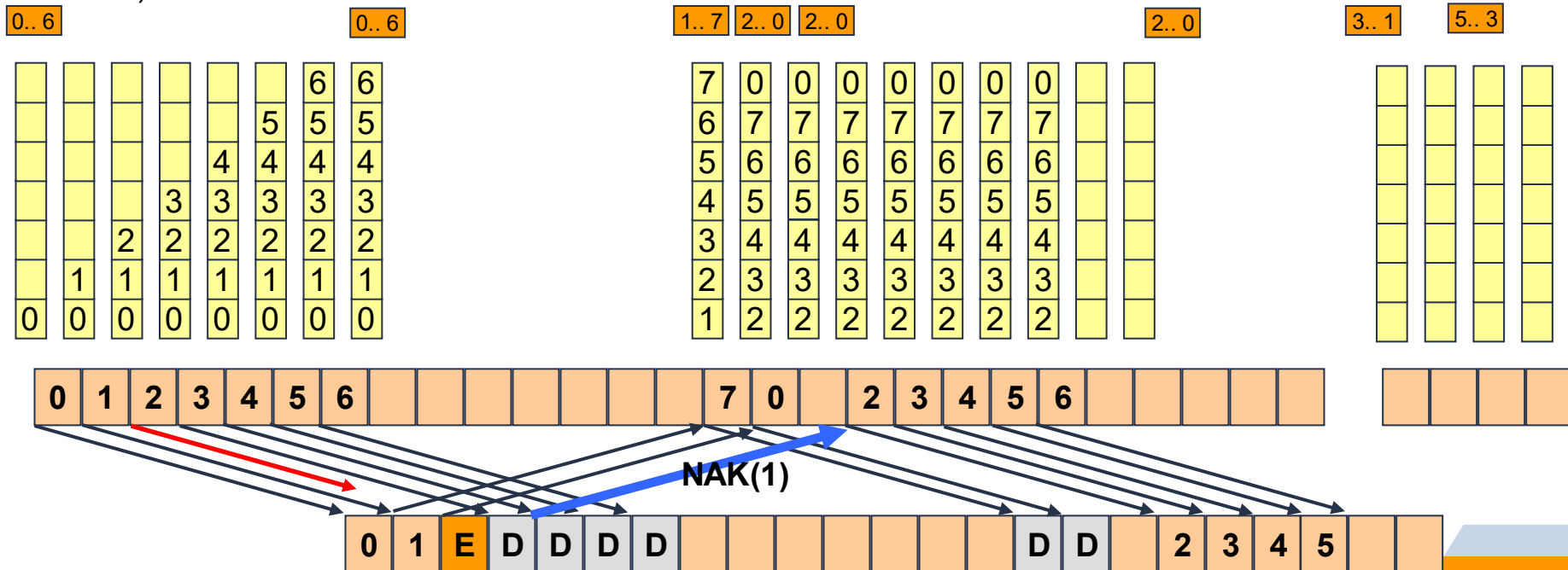
## Zadatak 2 - Rešenje

b)



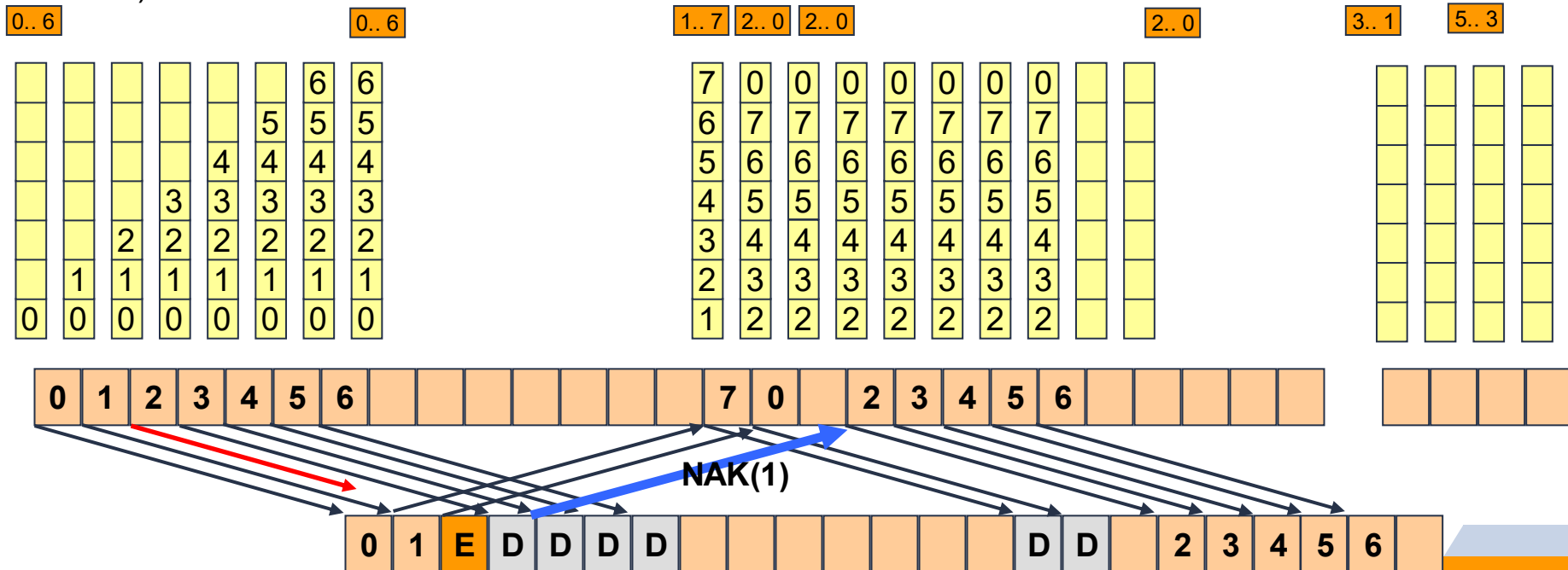
## Zadatak 2 - Rešenje

b)



## Zadatak 2 - Rešenje

b)





## Zadatak 2 - Rešenje

b)

0..6

0..6

1..7 2..0 2..0

2..0

3..1

5..3

						6	6
					5	5	5
				4	4	4	4
			3	3	3	3	3
		2	2	2	2	2	2
	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0

7	0	0	0	0	0	0	0	0	0
6	7	7	7	7	7	7	7	7	7
5	6	6	6	6	6	6	6	6	6
4	5	5	5	5	5	5	5	5	5
3	4	4	4	4	4	4	4	4	4
2	3	3	3	3	3	3	3	3	3
1	2	2	2	2	2	2	2	2	2


0	1	2	3	4	5	6							7	0		2	3	4	5	6	7	0				
---	---	---	---	---	---	---	--	--	--	--	--	--	---	---	--	---	---	---	---	---	---	---	--	--	--	--

--	--	--	--



## Zadatak 2 - Rešenje

b)

0..6

0..6

1..7 2..0 2..0

2..0

3..1

5..3

						6	6
					5	5	5
			4	4	4	4	4
		3	3	3	3	3	3
	2	2	2	2	2	2	2
	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0

7	0	0	0	0	0	0	0	0	0
6	7	7	7	7	7	7	7	7	7
5	6	6	6	6	6	6	6	6	6
4	5	5	5	5	5	5	5	5	5
3	4	4	4	4	4	4	4	4	4
2	3	3	3	3	3	3	3	3	3
1	2	2	2	2	2	2	2	2	2


0	1	2	3	4	5	6							7	0		2	3	4	5	6	7	0				
---	---	---	---	---	---	---	--	--	--	--	--	--	---	---	--	---	---	---	---	---	---	---	--	--	--	--

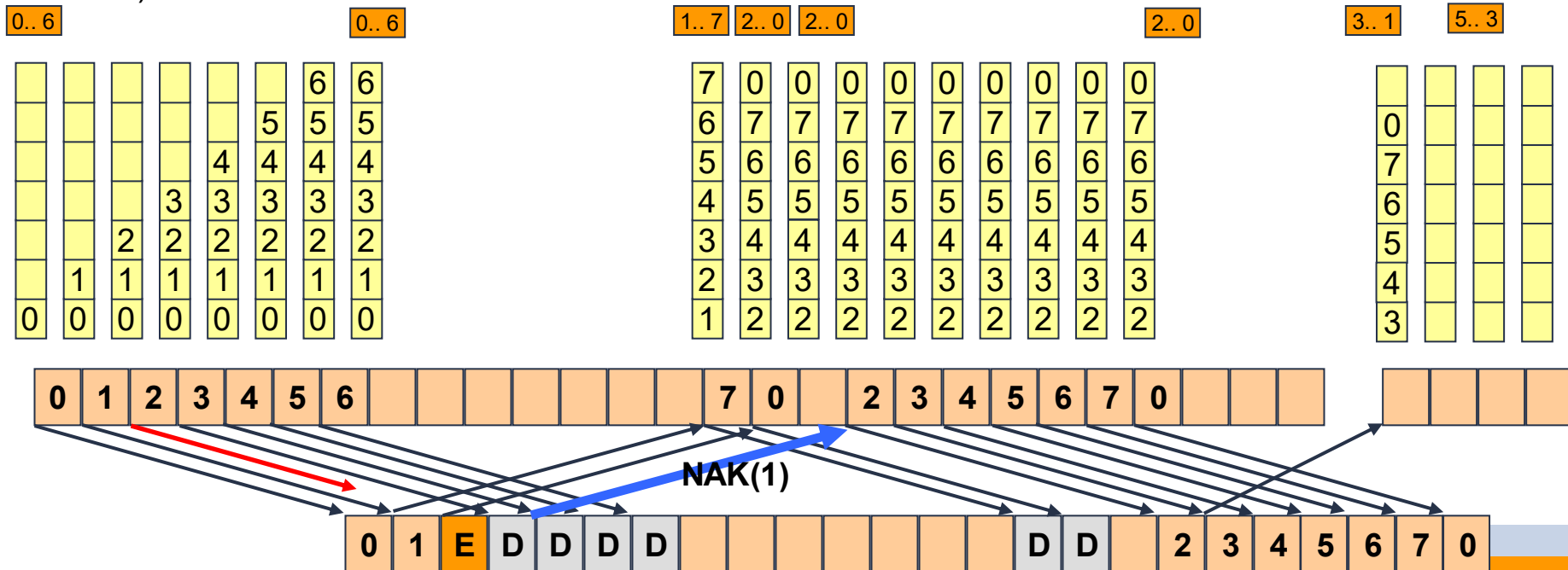
--	--	--	--





## Zadatak 2 - Rešenje

b)



## Zadatak 2 - Rešenje

b)

0..6

0..6

1..7

2..0

2..0

2..0

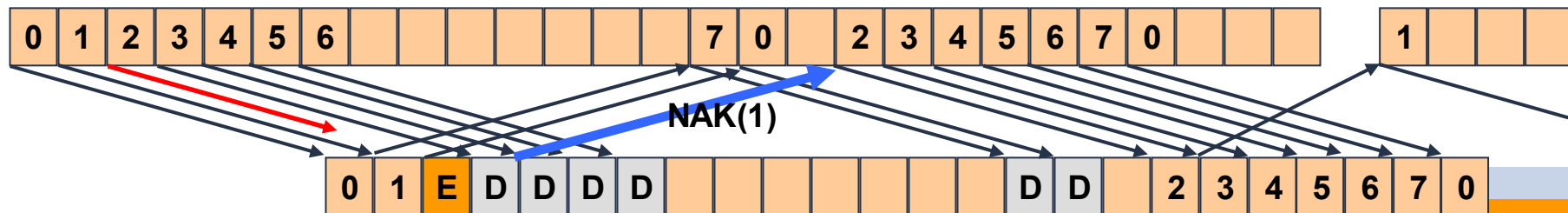
3..1

5..3

						6	6
					5	5	5
				4	4	4	4
			3	3	3	3	3
		2	2	2	2	2	2
	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0

7	0	0	0	0	0	0	0	0	0
6	7	7	7	7	7	7	7	7	7
5	6	6	6	6	6	6	6	6	6
4	5	5	5	5	5	5	5	5	5
3	4	4	4	4	4	4	4	4	4
2	3	3	3	3	3	3	3	3	3
1	2	2	2	2	2	2	2	2	2

1			
0			
7			
6			
5			
4			
3			



## Zadatak 2 - Rešenje

b)

0..6

0..6

1..7 2..0 2..0

2..0

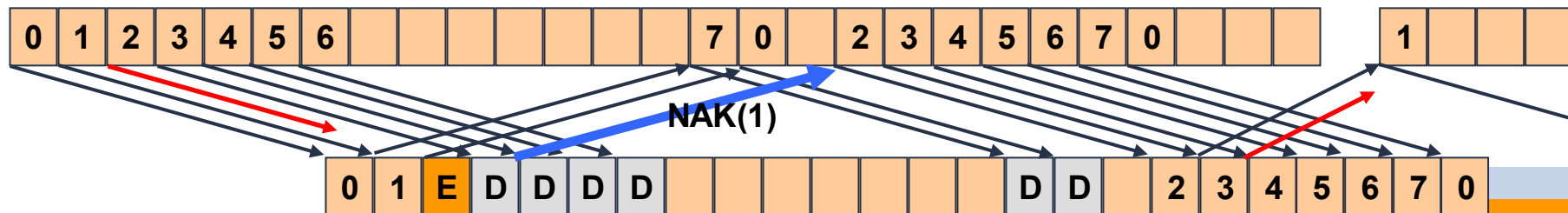
3..1

5..3

						6	6
					5	5	5
				4	4	4	4
			3	3	3	3	3
		2	2	2	2	2	2
	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0

7	0	0	0	0	0	0	0	0	0
6	7	7	7	7	7	7	7	7	7
5	6	6	6	6	6	6	6	6	6
4	5	5	5	5	5	5	5	5	5
3	4	4	4	4	4	4	4	4	4
2	3	3	3	3	3	3	3	3	3
1	2	2	2	2	2	2	2	2	2

1			
0			
7			
6			
5			
4			
3			



## Zadatak 2 - Rešenje

b)

0..6

0..6

1..7 2..0 2..0

2..0

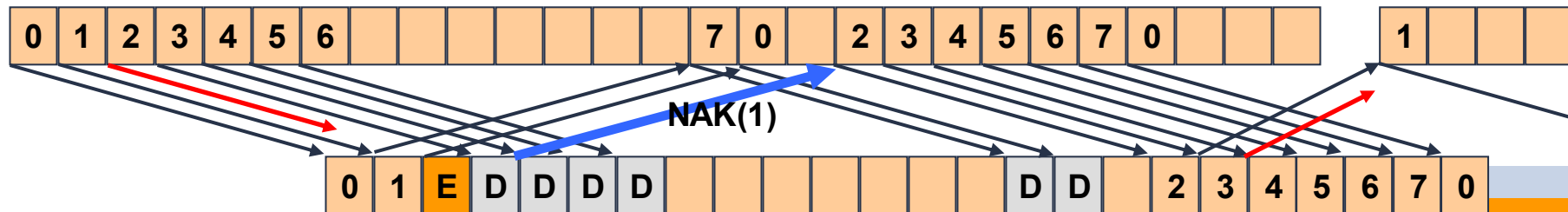
3..1

5..3

						6	6
					5	5	5
				4	4	4	4
			3	3	3	3	3
		2	2	2	2	2	2
	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0

7	0	0	0	0	0	0	0	0	0
6	7	7	7	7	7	7	7	7	7
5	6	6	6	6	6	6	6	6	6
4	5	5	5	5	5	5	5	5	5
3	4	4	4	4	4	4	4	4	4
2	3	3	3	3	3	3	3	3	3
1	2	2	2	2	2	2	2	2	2

1	1		
0	0		
7	7		
6	6		
5	5		
4	4		
3	3		



## Zadatak 2 - Rešenje

b)

0..6

0..6

1..7 2..0 2..0

2..0

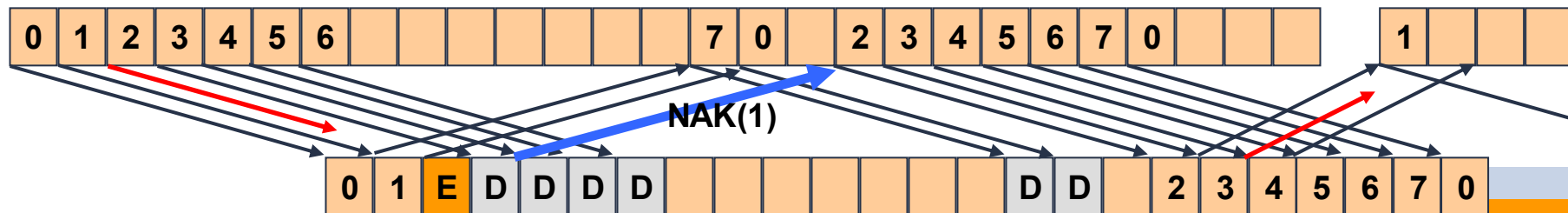
3..1

5..3

						6	6
					5	5	5
				4	4	4	4
			3	3	3	3	3
		2	2	2	2	2	2
	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0

7	0	0	0	0	0	0	0	0	0
6	7	7	7	7	7	7	7	7	7
5	6	6	6	6	6	6	6	6	6
4	5	5	5	5	5	5	5	5	5
3	4	4	4	4	4	4	4	4	4
2	3	3	3	3	3	3	3	3	3
1	2	2	2	2	2	2	2	2	2

1	1		
0	0		
7	7	1	
6	6	0	
5	5	7	
4	4	6	
3	3	5	



## Zadatak 2 - Rešenje

b)

0..6

0..6

1..7 2..0 2..0

2..0

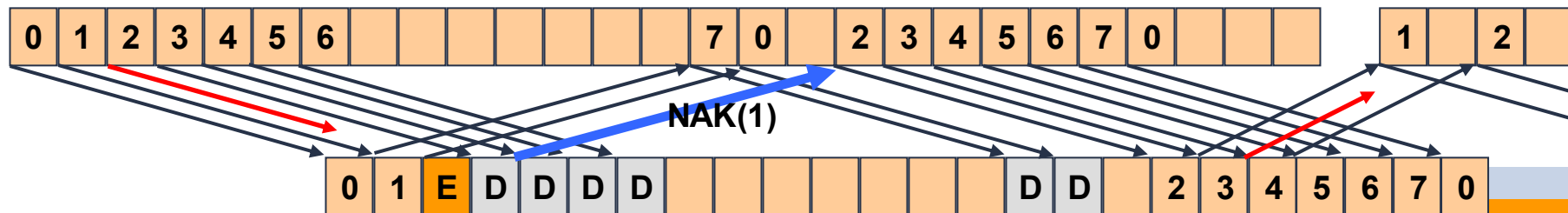
3..1

5..3

						6	6
					5	5	5
				4	4	4	4
			3	3	3	3	3
		2	2	2	2	2	2
	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0

7	0	0	0	0	0	0	0	0	0
6	7	7	7	7	7	7	7	7	7
5	6	6	6	6	6	6	6	6	6
4	5	5	5	5	5	5	5	5	5
3	4	4	4	4	4	4	4	4	4
2	3	3	3	3	3	3	3	3	3
1	2	2	2	2	2	2	2	2	2

1	1		
0	0	2	
7	7	1	
6	6	0	
5	5	7	
4	4	6	
3	3	5	



## Zadatak 2 - Rešenje

b)

0..6

0..6

1..7 2..0 2..0

2..0

3..1

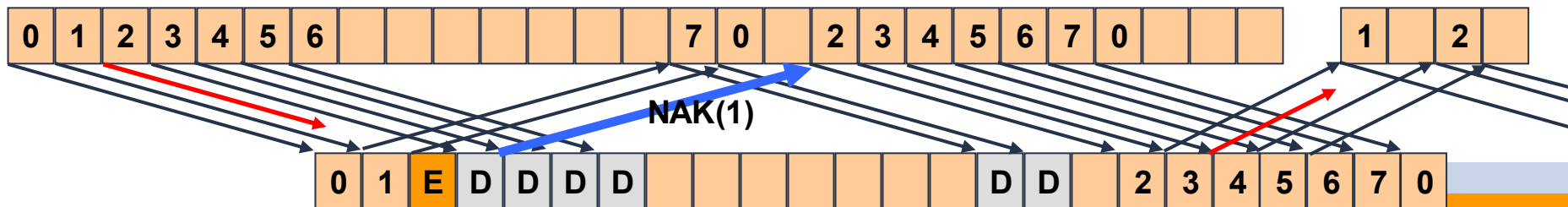
5..3

6..4

						6	6
					5	5	5
				4	4	4	4
			3	3	3	3	3
		2	2	2	2	2	2
	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0

7	0	0	0	0	0	0	0	0	0
6	7	7	7	7	7	7	7	7	7
5	6	6	6	6	6	6	6	6	6
4	5	5	5	5	5	5	5	5	5
3	4	4	4	4	4	4	4	4	4
2	3	3	3	3	3	3	3	3	3
1	2	2	2	2	2	2	2	2	2

1	1		
0	0	2	
7	7	1	2
6	6	0	1
5	5	7	0
4	4	6	7
3	3	5	6



## Zadatak 2 - Rešenje

b)

0..6

0..6

1..7 2..0 2..0

2..0

3..1

5..3

6..4

						6	6
					5	5	5
				4	4	4	4
			3	3	3	3	3
		2	2	2	2	2	2
	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0

7	0	0	0	0	0	0	0	0	0
6	7	7	7	7	7	7	7	7	7
5	6	6	6	6	6	6	6	6	6
4	5	5	5	5	5	5	5	5	5
3	4	4	4	4	4	4	4	4	4
2	3	3	3	3	3	3	3	3	3
1	2	2	2	2	2	2	2	2	2

1	1		
0	0	2	3
7	7	1	2
6	6	0	1
5	5	7	0
4	4	6	7
3	3	5	6

