

# Рачунарске мреже (вежбе - термин 2)

Контрола тока (Flow Control)

# Контрола тока

- ▶ Бешумни канали
  - ▶ Протокол за неограничен једносмерни пренос
  - ▶ Stop-and-Wait
- ▶ Бучни канали
  - ▶ Stop-and-wait ARQ (Automatic Repeat reQuest)
  - ▶ Go-Back-N-ARQ
  - ▶ Selective Repeat ARQ

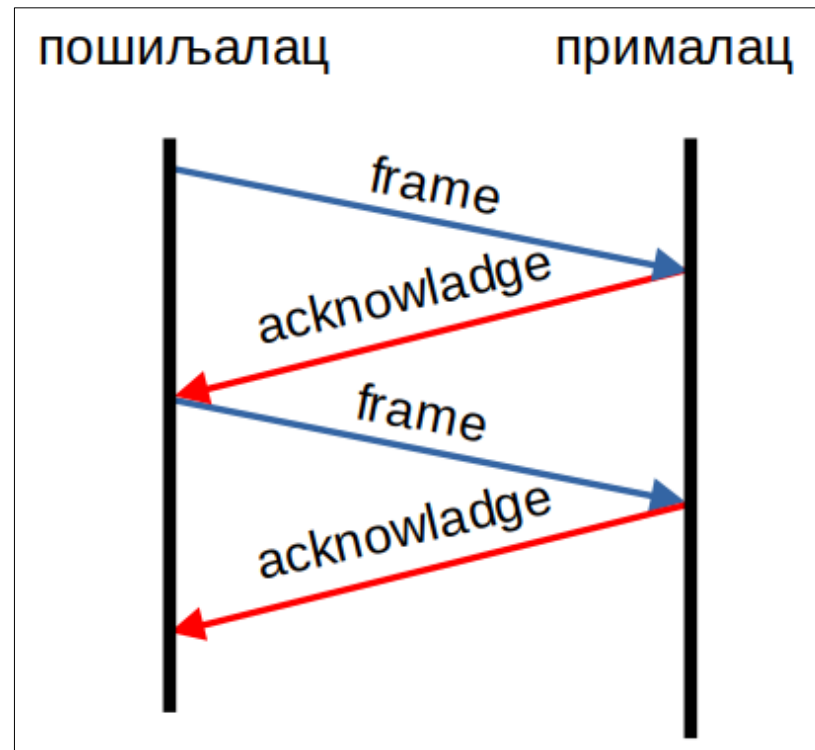
# Протокол за неограничен једносмерни пренос

```
void sender(void)
{
    frame s;           /*skladište za odlazni okvir*/
    packet buffer;    /*skladište za odlazni paket*/
    while(true)
    {
        from_network_layer(&buffer); /*preuzmi paket sa mrežnog sloja*/
        s.info = buffer;           /*umetni paket u okvir*/
        to_physical_layer(&s);     /*prosledi okvir fizičkom sloju*/
    }
}

void receiver(void)
{
    frame r;           /*skladište za dolazni okvir*/
    event_type event; /*događaj*/
    while(true)
    {
        wait_for_event(&event); /*jedina mogućnost je "frame_arrival"*/
        from_physical_layer(&r); /*preuzmi okvir*/
        to_network_layer(&r.info); /*prosledi paket iz okvira mrežnom sloju*/
    }
}
```

# Stop-and-Wait

- ▶ Stop-and-Wait је протокол Data-Link Layer-а намењен за трансмисију оквира путем бешумних канала. Пошиљалац емитује оквир и чека потврду, а тек када пријемник врати потврду да је оквир примљен, пошиљалац наставља са емитовањем наредног оквира.
- ▶ RPC - Remote Procedure Call

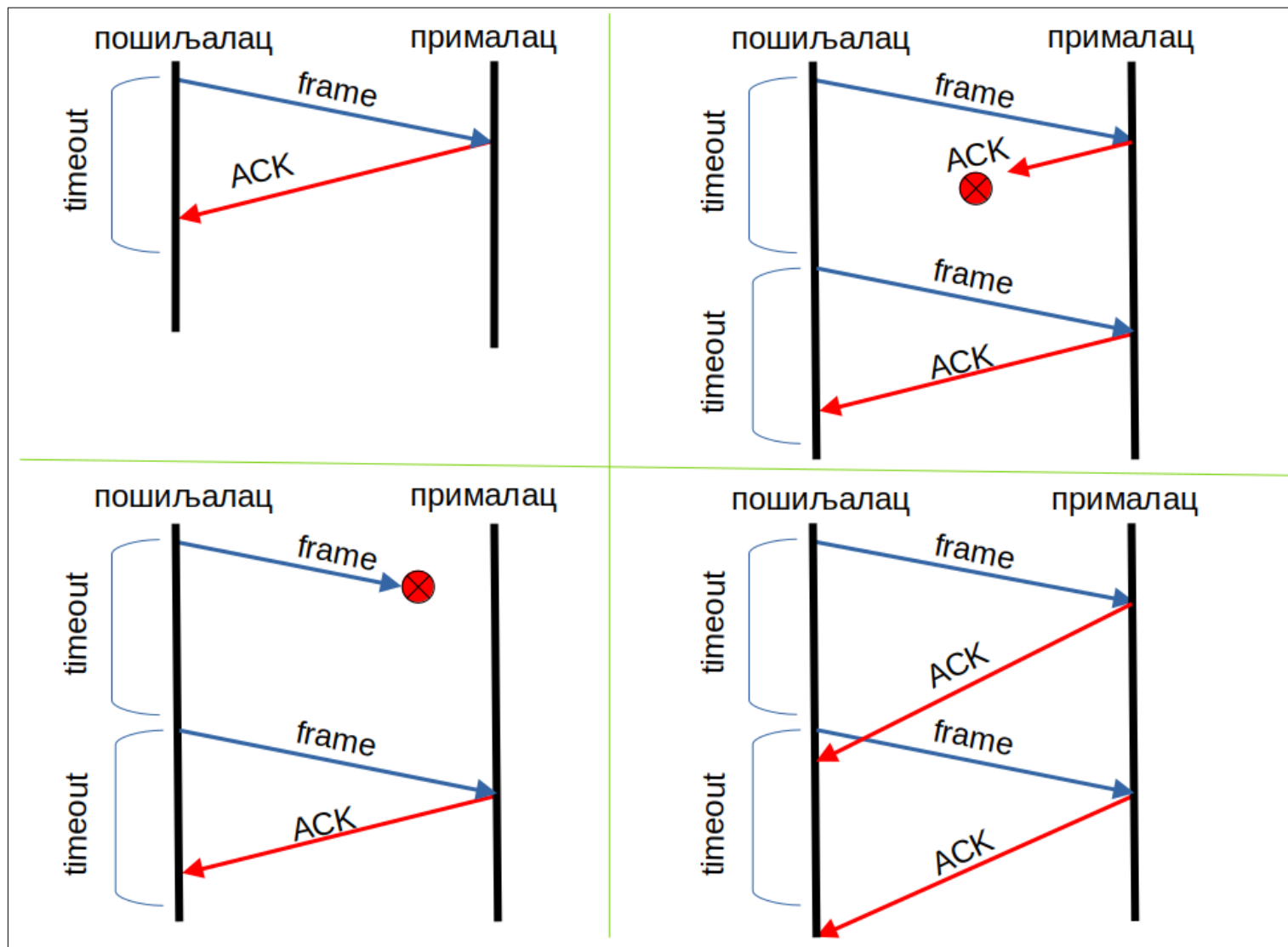


# Stop-and-Wait

```
void sender(void)
{
    frame s;           /*skladište za odlazni okvir*/
    packet buffer;    /*skladište za odlazni paket*/
    event_type event; /*jedini moguci event frame_arrival*/
    while (true)
    {
        from_network_layer(&buffer); /*preuzmi paket sa mrežnog sloja*/
        s.info = buffer;             /*umetni paket u okvir*/
        to_physical_layer(&s);       /*prosledi okvir fizičkom sloju*/
        wait_for_event(&event);
    }
}

void receiver(void)
{
    frame r,s;         /*skladište za dolazni okvir*/
    event_type event; /*događaj */
    while (true)
    {
        wait_for_event(&event); /*jedina mogućnost je "frame_arrival"*/
        from_physycal_layer(&r); /*preuzmi okvir*/
        to_network_layer(&r.info); /*prosledi paket iz okvira mrežnom sloju*/
        to_physycal_layer(&s);     /*prosledjuje potvrdu o prijemu*/
    }
}
```

# Stop-and-Wait ARQ



# Stop-and-Wait ARQ - пошилъалац

```
void sender(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);
            }
        }
    }
}
```

# Stop-and-Wait ARQ - прималац

```
void receiver(void)
{
    seq_nr next_frame_to_send;
    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);
        }
    }
}
```

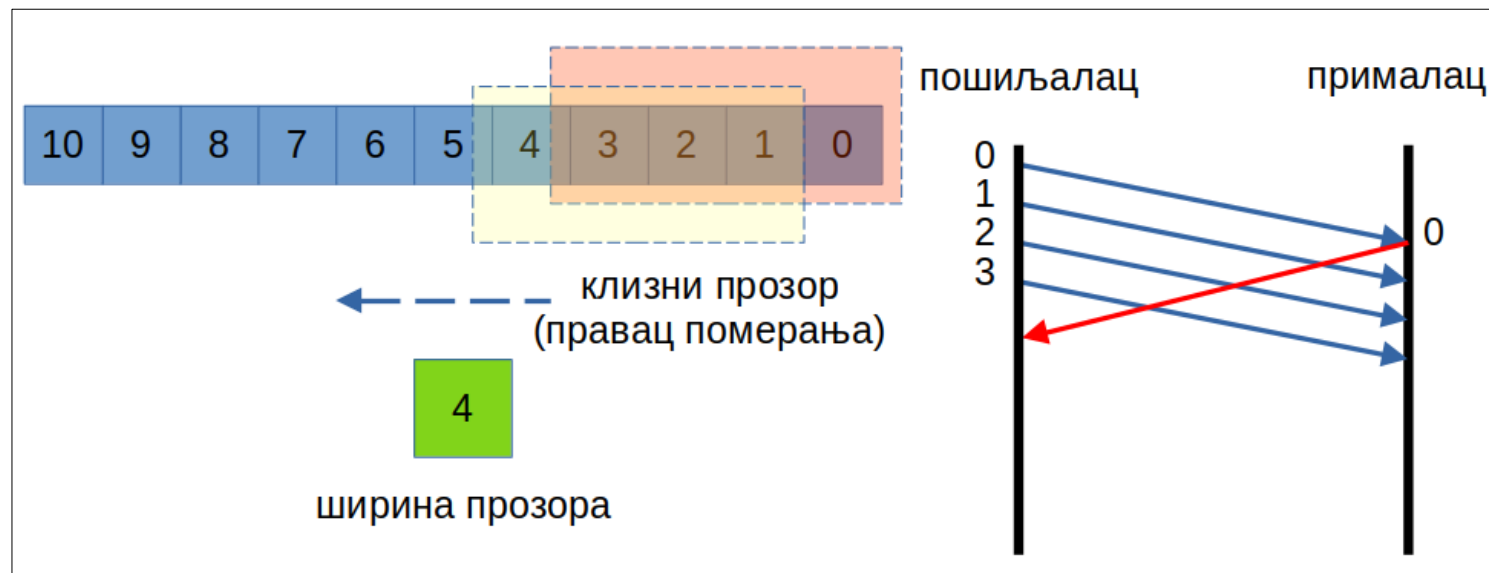


# Протокол клизног прозора (Sliding-Window)

- ▶ Разлози за побољшање Stop-and-Wait:
  - ▶ Слање само једног оквира у јединици времена
  - ▶ Слаба искоришћеност пропусног опсега канала
  - ▶ Слабо искоришћене перформансе хардвера
- ▶ Једна од метода за слање више оквира секвенцијално јесте метода клизног прозора.
- ▶ Сваки оквир је означен бројем који се назива *Sequence Number*
- ▶ Прималац одговара са АСК поруке да је прихватио пакет са редним бројем *Sequence Number*
- ▶ Након прихватања потврде пошиљалац може да пошаље нови оквир

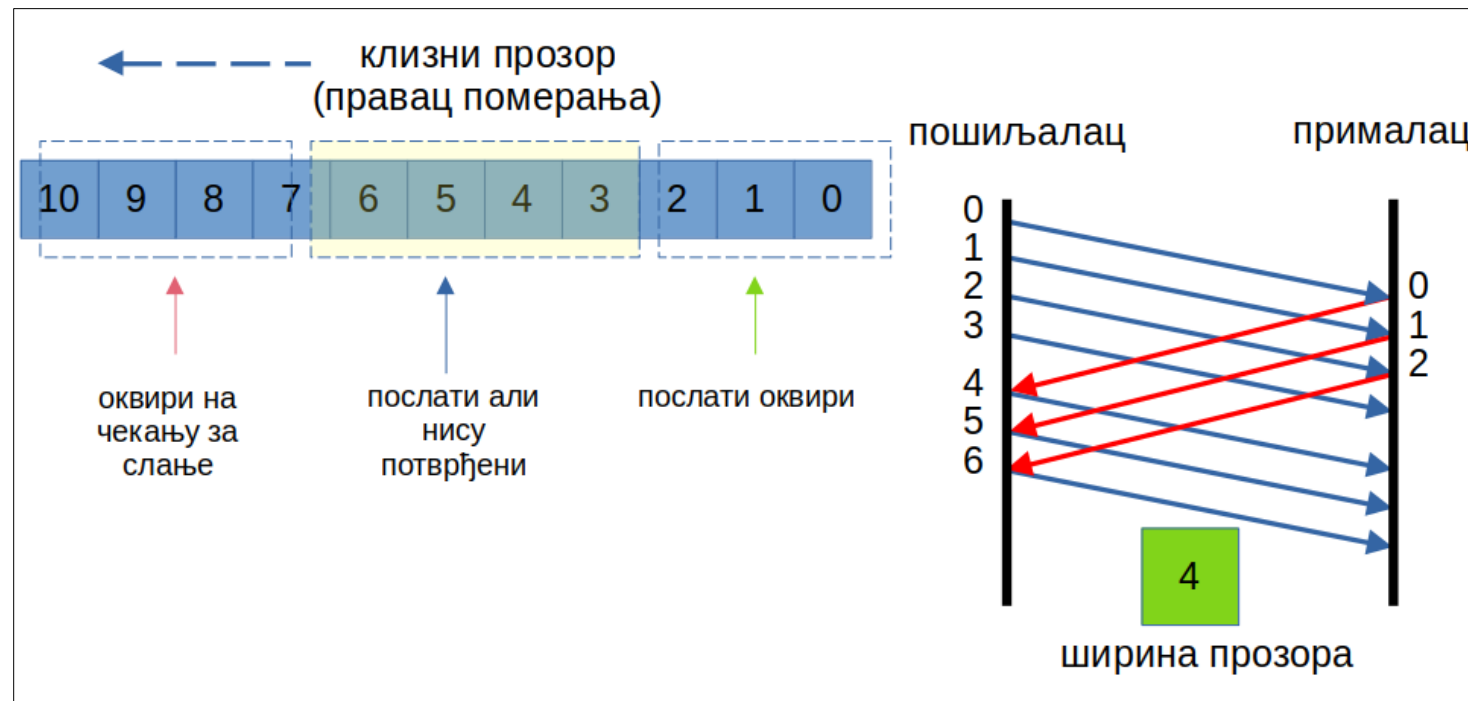
# Протокол клизног прозора - пример

- ▶ Пошиљалац треба да пошаље одређени број оквира означених редом са 0,1,2...10.
- ▶ Параметар *Windows Size* поставимо на 4, то значи да се истовремено шаље 4 оквира
- ▶ По пријему првог одговора, пошиљалац шаље оквир под редним бројем 4



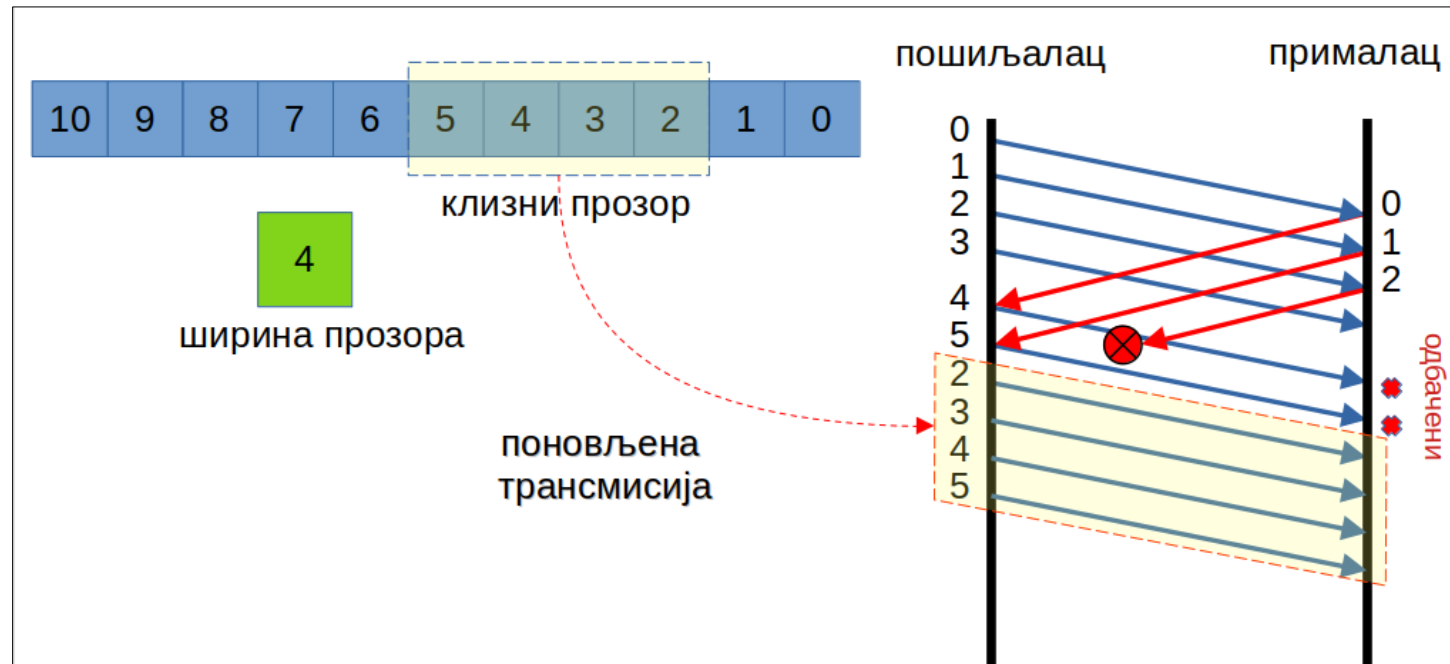
# Протокол клизног прозора - пример

- ▶ Слика приказује пример клизног прозора када су успешно потврђени оквири 0, 1 и 2, а још увек чекају потврде за оквири 3,4,5 и 6
- ▶ Оквири 7,8,9 и 10 спремни за слање и у стању су чекања.



# Go-Back-N ARQ протокол

- ▶ N у називу овог протокола означава величину прозора пошиљаоца (пример: *N-Back-4* када је величина прозора 4).
- ▶ Уколико потврда о послатом оквиру није стигла у одређеном времену, сви оквири у тренутном прозору биће ретрансмитовани



# Go-Back-N ARQ задаци

- ▶ **Задатак 1:** Рачунар А треба да пошаље поруку од 9 оквира рачунару В користећи клизни прозор величине 3 и *Go-Back-N* Error-Control стратегије. Сви оквири су спремни и одмах доступни за трансмисију. Уколико се сваки пет пакет којег шаље рачунар А изгуби (нема одговора од рачунара В), који је укупан број оквира које ће рачунар А послати рачунару В?
- ▶ **Задатак 2:** Рачунар А треба да пошаље поруку од 10 оквира рачунару В. Рачунари су се договорили да користе *Go-Back-4* протокол. Уколико се сваки шести пакет којег шаље рачунар А изгуби (нема одговора од рачунара В), који је укупан број оквира које ће рачунар А послати рачунару В?

# Селективно понављање (Selective Repeat ARQ)

- ▶ За разлику од претходног протокола, протокол селективног понављања из клизног прозора ретрансмитије само оквир за којег пошиљалац није добио одговор, уместо свих оквира из тренутног клизног прозора.
- ▶ Уколико оквир са очекујућим sequence бројем не стигне, прималац шаље *NACK* (*negative acknowledge*) пошиљаоцу. Пошиљалац ће ретрансмитовати само оквире за које је добио *NACK*.



# Селективно понављање (Selective Repeat ARQ)

- ▶ **Задатак 1:** Применом протокола селективног понављања, величина прозора између пошиљаоца и пријемника је 5. Уколико оквир `seq_num=0` врати NACK, оквир 5 трансмитује, затим оквир 1 врати NACK, оквир 2 NACK и оквир 6 се трансмитује. Како изгледа секвенца трансмисије оквира на страни пошиљаоца?
- ▶ **Задатак 2:** Рачунар А шаље 10 оквира рачунару В. Рачунари су се договорили да користе метод селективног понављања. Колико оквира ће бити трансмитовано са Рачунара А уколико је сваки 6. оквир које се шаље изгубљен или оштећен? Упоредити број трансмисија на конкретном примеру између ове методе и методе Go-Back-4 ARQ.

# Додатак

- ▶ **Задатак за колоквијум (додатни):** Рачунар А користи 32В (В - бајт) оквире за трансмисију поруке према рачунару Б користећи протокол клизног прозора. Време које је протекне од момента слања оквира до пријема потврде о пристиглом оквиру је 80 msec. Пропусни опсег канала је 128 kbps. Која је оптимална величина прозора коју рачунар А треба да одабере?



Питања?