

# Распределенные алгоритмы

ЛЕКТОР: В.А. Захаров

## Лекция 4.

Коммуникационный протокол с таймерами.

Допущения о времени. Таймеры.

Устройство протокола с таймерами.

Обоснование корректности протокола с таймерами.

Модификации протокола.

# Коммуникационный протокол с таймерами

При передаче сообщений на «дальние» расстояния приходится использовать промежуточные узлы в качестве ретрансляционных станций. В этом случае возникают следующие дополнительные трудности:

# Коммуникационный протокол с таймерами

При передаче сообщений на «дальние» расстояния приходится использовать промежуточные узлы в качестве ретрансляционных станций. В этом случае возникают следующие дополнительные трудности:

- ▶ необходимо устанавливать и завершать соединения (сеансы связи);

# Коммуникационный протокол с таймерами

При передаче сообщений на «дальние» расстояния приходится использовать промежуточные узлы в качестве ретрансляционных станций. В этом случае возникают следующие дополнительные трудности:

- ▶ необходимо устанавливать и завершать соединения (сеансы связи);
- ▶ сообщения могут не только теряться, но также дублироваться и перемешиваться (изменять относительный порядок следования).

# Коммуникационный протокол с таймерами

При передаче сообщений на «дальние» расстояния приходится использовать промежуточные узлы в качестве ретрансляционных станций. В этом случае возникают следующие дополнительные трудности:

- ▶ необходимо устанавливать и завершать соединения (сеансы связи);
- ▶ сообщения могут не только теряться, но также дублироваться и перемешиваться (изменять относительный порядок следования).

Для решения этой задачи Флетчер и Ватсон предложили  $\Delta t$ -протокол сквозной передачи сообщений с использованием синхронизованных таймеров.

# Коммуникационный протокол с таймерами

Мы рассмотрим упрощенный вариант этого протокола.  
Упрощения касаются следующих четырех аспектов этого протокола.

# Коммуникационный протокол с таймерами

Мы рассмотрим упрощенный вариант этого протокола. Упрощения касаются следующих четырех аспектов этого протокола.

**Однонаправленность.** Передача данных идет только в одном направлении от процесса  $p$  к процессу  $q$ . Мы будем называть процесс  $p$  **отправителем**, а процесс  $q$  — **получателем**.



# Коммуникационный протокол с таймерами

Мы рассмотрим упрощенный вариант этого протокола. Упрощения касаются следующих четырех аспектов этого протокола.

**Однонаправленность.** Передача данных идет только в одном направлении от процесса  $p$  к процессу  $q$ . Мы будем называть процесс  $p$  **отправителем**, а процесс  $q$  — **получателем**.

**В окне приема — одно слово.** Поступивший пакет получатель доставляет по назначению только в том случае, когда его порядковый номер совпадает с ожидаемым номером.

# Коммуникационный протокол с таймерами

Мы рассмотрим упрощенный вариант этого протокола. Упрощения касаются следующих четырех аспектов этого протокола.

**Однонаправленность.** Передача данных идет только в одном направлении от процесса  $p$  к процессу  $q$ . Мы будем называть процесс  $p$  **отправителем**, а процесс  $q$  — **получателем**.

**В окне приема — одно слово.** Поступивший пакет получатель доставляет по назначению только в том случае, когда его порядковый номер совпадает с ожидаемым номером.

**Упрощенные допущения о времени.** Используется минимальное количество таймеров. Например, предполагается, что подтверждение может быть отправлено в любое время до тех пор, пока получатель поддерживает соединение открытым.

# Коммуникационный протокол с таймерами

Мы рассмотрим упрощенный вариант этого протокола. Упрощения касаются следующих четырех аспектов этого протокола.

**Однонаправленность.** Передача данных идет только в одном направлении от процесса  $p$  к процессу  $q$ . Мы будем называть процесс  $p$  **отправителем**, а процесс  $q$  — **получателем**.

**В окне приема — одно слово.** Поступивший пакет получатель доставляет по назначению только в том случае, когда его порядковый номер совпадает с ожидаемым номером.

**Упрощенные допущения о времени.** Используется минимальное количество таймеров. Например, предполагается, что подтверждение может быть отправлено в любое время до тех пор, пока получатель поддерживает соединение открытым.

**Пакеты состоят из одного слова.** В каждый пакет данных отправитель может поместить одно-единственное слово.

# Допущения о времени. Таймеры

**Таймеры** — устройства, измеряющие физическое время. Мы будем опираться на следующие основные предположения о времени и таймерах.

# Допущения о времени. Таймеры

**Таймеры** — устройства, измеряющие физическое время. Мы будем опираться на следующие основные предположения о времени и таймерах.

1. *Глобальное время.*
2. *Ограниченное время жизни пакета.*
3. *Таймеры*

# Допущения о времени. Таймеры

**Таймеры** — устройства, измеряющие физическое время. Мы будем опираться на следующие основные предположения о времени и таймерах.

1. *Глобальное время.* Все процессы системы функционируют в рамках единой глобальной шкалы времени. Каждое событие происходит мгновенно, и процессы **не могут** регистрировать те моменты времени, в которые происходят события.
2. *Ограниченное время жизни пакета.*
3. *Таймеры*

# Допущения о времени. Таймеры

**Таймеры** — устройства, измеряющие физическое время. Мы будем опираться на следующие основные предположения о времени и таймерах.

1. *Глобальное время.*
2. *Ограниченное время жизни пакета.* Время жизни пакета ограничено некоторой константой  $\mu$  (**максимальное время жизни пакета**). Если пакет был отправлен в момент времени  $\sigma$  и получен в момент времени  $\tau$ , то справедливо неравенство

$$\sigma < \tau < \sigma + \mu.$$

Если в канале произошло дублирование пакета, то каждая копия должна быть получена спустя не более  $\mu$  единиц времени после отправления исходного пакета (в противном случае копии будут утрачены).

3. *Таймеры*

# Допущения о времени. Таймеры

**Таймеры** — устройства, измеряющие физическое время. Мы будем опираться на следующие основные предположения о времени и таймерах.

1. *Глобальное время.*
2. *Ограниченное время жизни пакета.*
3. *Таймеры* **Таймер** — вещественная переменная  $X_t$ , значение которой со временем постоянно убывает (или присваивается этой переменной явным образом). Запись  $X_t^{(t)}$  обозначает значение таймера в момент времени  $t$ . Если в период времени между моментами  $t_1$  и  $t_2$  переменной  $X_t$  не было присвоено какое-либо значение, то справедливо неравенство

$$X_t^{(t_1)} - X_t^{(t_2)} = t_2 - t_1.$$



# Уточнение условий задачи коммуникации

## Поступление входных слов.

Входные слова, подлежащие отправлению, моделируются бесконечным массивом  $in_p$ . В каждый момент времени процесс  $p$  имеет доступ только к некоторой части массива.

Предполагается, что имеется некоторый процесс-генератор (агент), порождающий слова входного массива.

Процесс-отправитель  $p$  получает все новые и новые слова от агента, порождающего эти слова. Эту операцию будем называть **поступлением** слов к процессу-отправителю.

# Уточнение условий задачи коммуникации

## Поступление входных слов.

Входные слова, подлежащие отправлению, моделируются бесконечным массивом  $in_p$ . В каждый момент времени процесс  $p$  имеет доступ только к некоторой части массива. Предполагается, что имеется некоторый процесс-генератор (агент), порождающий слова входного массива.

Процесс-отправитель  $p$  получает все новые и новые слова от агента, порождающего эти слова. Эту операцию будем называть **поступлением** слов к процессу-отправителю.

## Вручение выходных слов.

Предполагается, что имеется некоторый процесс-потребитель (агент), которому должны быть доставлены слова входного массива. Процесс-получатель будет вручать эти слова процессу-потребителю посредством операции, которую условимся называть **вручением** слова.

# Задача надежного обмена информацией

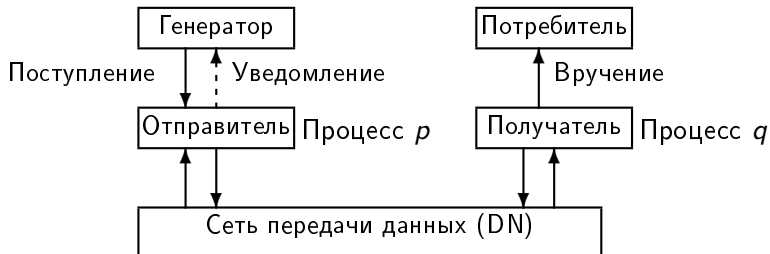


Рис.: Упрощенная архитектура сети.

# Уточнение условий задачи коммуникации

Свойства протокола.

1. Отсутствие потерь.
2. Соблюдение порядка.

# Уточнение условий задачи коммуникации

## Свойства протокола.

1. **Отсутствие потерь.** Каждое слово из массива  $in_p$  будет вручено процессом  $q$  потребителю или зарегистрировано процессом  $p$  (как «вероятно потерянное») спустя ограниченный отрезок времени с момента поступления этого слова от генератора к процессу  $p$ .
2. **Соблюдение порядка.**

# Уточнение условий задачи коммуникации

Свойства протокола.

1. **Отсутствие потерь.**
2. **Соблюдение порядка.** Слова, которые вручаются потребителю процессом  $q$ , следуют в порядке строгого возрастания номеров в массиве  $in_p$ .

# Уточнение условий задачи коммуникации

## Свойства протокола.

1. **Отсутствие потерь.** Каждое слово из массива  $in_p$  будет вручено процессом  $q$  потребителю или зарегистрировано процессом  $p$  (как «вероятно потерянное») спустя ограниченный отрезок времени с момента поступления этого слова от генератора к процессу  $p$ .
2. **Соблюдение порядка.** Слова, которые вручаются потребителю процессом  $q$ , следуют в порядке строгого возрастания номеров в массиве  $in_p$ .

# Уточнение условий задачи коммуникации

## Свойства протокола.

1. **Отсутствие потерь.** Каждое слово из массива  $in_p$  будет вручено процессом  $q$  потребителю или зарегистрировано процессом  $p$  (как «вероятно потерянное») спустя ограниченный отрезок времени с момента поступления этого слова от генератора к процессу  $p$ .
2. **Соблюдение порядка.** Слова, которые вручаются потребителю процессом  $q$ , следуют в порядке строгого возрастания номеров в массиве  $in_p$ .

**Никакой протокол не может предоставить гарантии того, что слово будет доставлено по назначению за ограниченный срок времени.**

**Почему?**



# Описание протокола с таймерами

Список констант.

**Сетевая константа:**

$\mu$  : real; (\* максимальное время жизни пакета \*)

**Константы протокола:**

$U$  : real; (\* Продолжительность периода отправления сообщения \*)

$R$  : real; (\* Продолжительность перерыва при приеме сообщения:  
 $R \geq U + \mu$  \*)

$S$  : real; (\* Продолжительность перерыва при передаче сообщения:  
 $S \geq R + 2\mu$  \*)

# Описание протокола с таймерами

Список переменных и массивов.

Учетные записи отправителя:

*Low* : integer; (\* Подтвержденные слова текущего сеанса связи \*)

*High* : integer; (\* Поступившие слова текущего сеанса связи \*)

*Ut* : array of timers; (\* Массив таймеров \*)

*St* : timer; (\* Таймер отправителя\*)

Учетные записи получателя:

*Exp* : integer; (\* Очередной ожидаемый порядковый номер \*)

*Rt* : timer; (\* Таймер получателя \*)

Коммуникационная подсистема:

*M<sub>q</sub>* : channel; (\* Пакеты данных для процесса *q* \*)

*M<sub>p</sub>* : channel; (\* Пакеты подтверждений для процесса *p* \*)

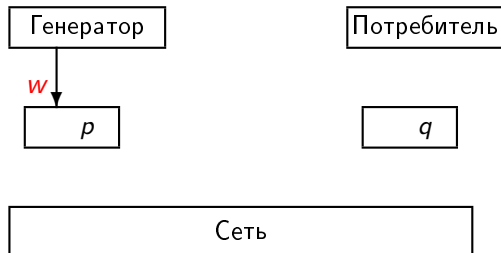
Вспомогательные переменные:

*B* : integer init 0 (\* Слова из предыдущего сеанса связи \*)

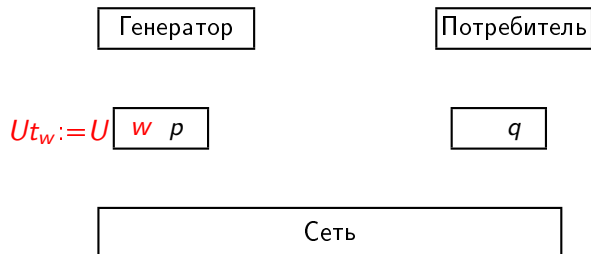
*cr* : bool init false; (\* Участие получателя в сеансе связи \*)

*cs* : bool init false; (\* Участие отправителя в сеансе связи \*)

# Сценарий работы протокола

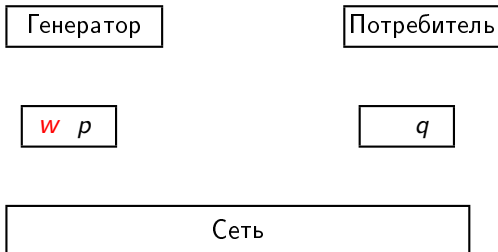


# Сценарий работы протокола

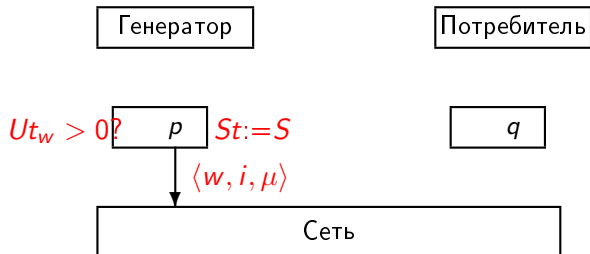


# Сценарий работы протокола

Проходит время

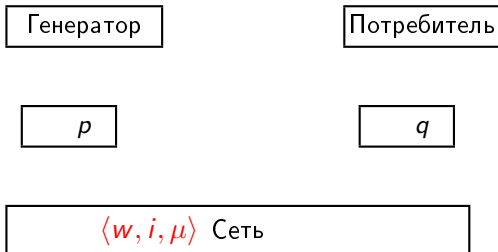


# Сценарий работы протокола

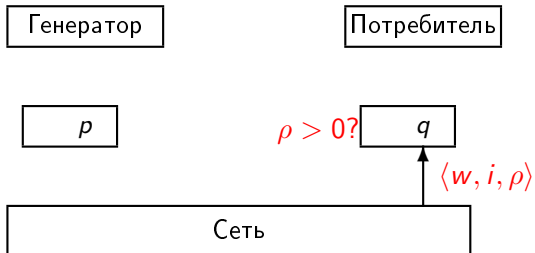


# Сценарий работы протокола

Проходит время

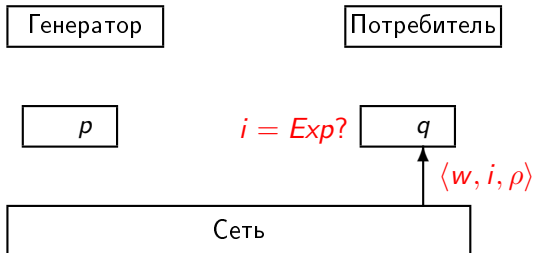


# Сценарий работы протокола

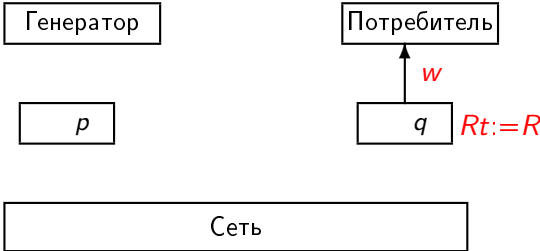




# Сценарий работы протокола

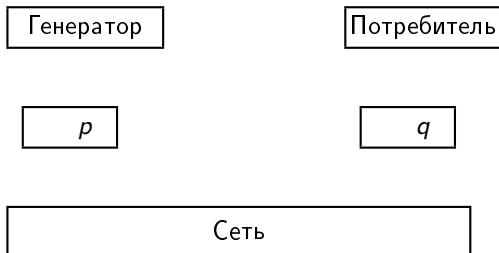


# Сценарий работы протокола

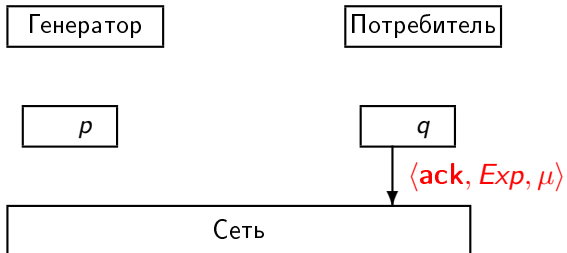


# Сценарий работы протокола

Проходит время

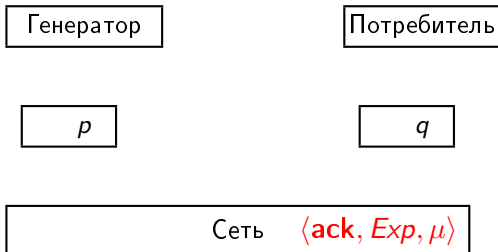


# Сценарий работы протокола

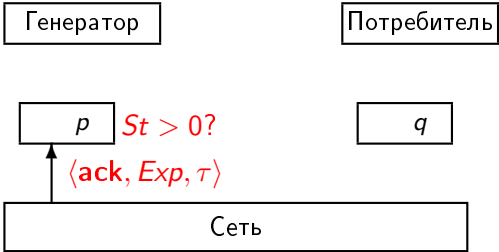


# Сценарий работы протокола

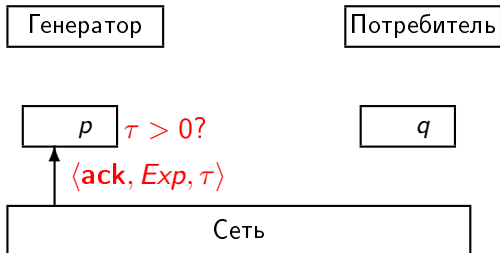
Проходит время



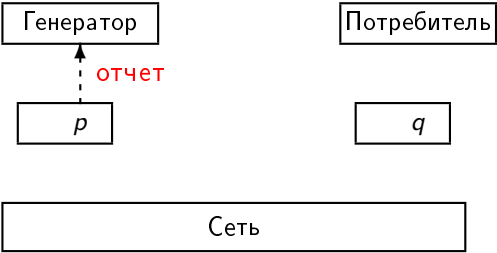
# Сценарий работы протокола



# Сценарий работы протокола



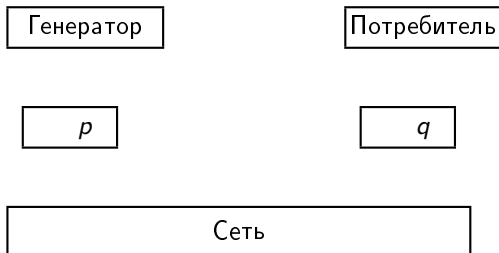
# Сценарий работы протокола



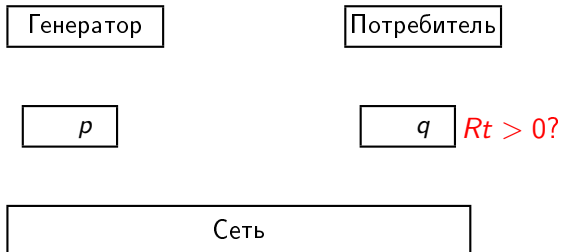


# Сценарий работы протокола

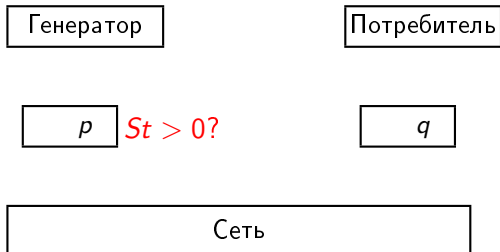
Проходит время



# Сценарий работы протокола



# Сценарий работы протокола



## Описание протокола с таймерами (процесс–отправитель)

$A_p$  : **begin** if not  $cs$  **then**  
    **begin** create( $St, High, Low$ ) ; (\*  $cs := true$  \*)  
         $Low := High := 0$  ;  $St := S$  **end** ;  
     $Ut[B + High] := U$  ;  $High := High + 1$   
**end**

$S_p$  : {  $cs \wedge Low \leq i < High \wedge Ut[B + i] > 0$  }  
**begin** send(**data**, ( $i = Low$ ),  $i$ ,  $in_p[B + i]$ ,  $\mu$ ) ;  $St := S$  **end**

$R_p$  : {  $cs \wedge \langle ack, i, \rho \rangle \in M_p$  }  
**begin** receive(**ack**,  $i, \rho$ ) ;  $Low := \max(Low, i)$  **end**

$E_p$  : {  $cs \wedge Ut[B + Low] \leq -2\mu - R$  }  
**begin**  $error[B + Low] := true$  ;  $Low := Low + 1$  **end**

$C_p$  : {  $cs \wedge St < 0 \wedge Low = High$  }  
**begin**  $B := B + High$  ; delete ( $St, High, Low$ ) **end**  
(\*  $cs := false$  \*)

# Открытие сеанса связи (процесс–отправитель)

```
Ap: (* Поступление очередного слова *)  
  begin if not cs then  
    begin(* Вначале открывается сеанс связи *)  
      create(St, High, Low) ; (* cs := true *)  
      Low := High := 0 ; St := S  
    end;  
    Ut[B + High] := U ; High := High + 1  
  end
```

---

Протокол открывает сеанс связи всякий раз, когда соединение отсутствует, но при этом к отправителю поступает некоторое слово.

Предикат *cs* имеет значение *true* , когда отправитель открыл сеанс связи. Для того, чтобы выяснить, открыт ли сеанс связи, процесс проверяет, существует ли учетная запись соединения в списке записей открытых сеансов связи.

# Открытие сеанса связи (процесс–отправитель)

```
Ap: (* Поступление очередного слова *)  
  begin if not cs then  
    begin(* Вначале открывается сеанс связи *)  
      create(St, High, Low) ; (* cs := true *)  
      Low := High := 0 ; St := S  
    end;  
    Ut[B + High] := U ; High := High + 1  
  end
```

---

Когда отправитель открывает сеанс связи, он начинает нумерацию поступивших к нему слов с 0 . Переменная *High* предназначена для подсчета количества слов, поступивших в текущем сеансе связи, а переменная *Low* служит для подсчета количества слов, подтвержденных в текущем сеансе связи. Отправитель может пересылать пакеты, порядковые номера которых расположены в интервале между двумя числами *Low* и *High* – 1 .

# Открытие сеанса связи (процесс–отправитель)

```
Ap: (* Поступление очередного слова *)  
  begin if not cs then  
    begin(* Вначале открывается сеанс связи *)  
      create(St, High, Low) ; (* cs := true *)  
      Low := High := 0 ; St := S  
    end;  
    Ut[B + High] := U ; High := High + 1  
  end
```

---

Каждое вновь поступившее слово «ставится на учет» — для него заводится отдельный таймер, в который устанавливается предельно допустимый срок отправления этого слова. Если по истечении времени  $U$  слово не будет отправлено, то оно будет считаться потерянным.

## Отправление сообщения (процесс–отправитель)

$S_p$ :(\* Отправить  $i$ -е слово текущего сеанса связи \*)  
 $\{ cs \wedge Low \leq i < High \wedge Ut[B + i] > 0 \}$   
**begin** send⟨data, ( $i = Low$ ),  $i$ ,  $in_p[B + i]$ ,  $\mu$ ⟩ ;  $St := S$  **end**

---

При открытом сеансе связи для передачи выбираются лишь те слова, номера которых расположены в диапазоне  $Low \dots High - 1$ , и соответствующие этим словам таймеры имеют положительные значения.



## Отправление сообщения (процесс–отправитель)

$$S_p: (* \text{ Отправить } i\text{-е слово текущего сеанса связи } *)$$
$$\{ cs \wedge Low \leq i < High \wedge Ut[B + i] > 0 \}$$
$$\text{begin send} \langle \text{data}, (i = Low), i, in_p[B + i], \mu \rangle ; St := S \text{ end}$$

---

Пакеты данных, которые отправляются по ходу работы рассматриваемого протокола состоят из особого бита (т.н. признака начала последовательности), порядкового номера и слова. Чтобы проводить анализ протокола, в каждом пакете выделяется четвертое (искусственное) поле, которое называется оставшимся сроком жизни пакета. В этом поле указывается максимальное время, на протяжении которого пакет может оставаться в канале до того, как он будет доставлен по назначению или потерян в силу предположения об ограниченном времени жизни пакета.

## Отправление сообщения (процесс–отправитель)

```
 $S_p: (* \text{ Отправить } i\text{-е слово текущего сеанса связи } *)$   
  {  $cs \wedge Low \leq i < High \wedge Ut[B + i] > 0$  }  
  begin send⟨data, ( $i = Low$ ),  $i$ ,  $in_p[B + i]$ ,  $\mu$ ⟩ ;  $St := S$  end
```

---

Признак начала последовательности используется получателем в том случае, когда пакет получен при закрытом сеансе связи, для того, чтобы решить, можно ли открыть сеанс связи (и принять содержащееся в пакете слово). Отправитель устанавливает в этом бите значение **true**, когда все предыдущие слова уже получили подтверждение или были занесены в отчет (как вероятно утраченные).

## Отправление сообщения (процесс–отправитель)

```
 $S_p: (* \text{ Отправить } i\text{-е слово текущего сеанса связи } *)$   
  {  $cs \wedge Low \leq i < High \wedge Ut[B + i] > 0$  }  
  begin send⟨data, ( $i = Low$ ),  $i$ ,  $in_p[B + i]$ ,  $\mu$ ⟩ ;  $St := S$  end
```

---

Таймер отправителя устанавливается так, чтобы избежать ситуации, при которой подтверждение поступает при закрытом сеансе связи. Для этого соединение поддерживается по меньшей мере  $S$  единиц времени после отправления очередного пакета, где  $S \geq R + 2\mu$ . Всякий раз, когда отправляется некоторый пакет, на таймере  $St$  устанавливается время  $S$ .

# Прием подтверждения (процесс–отправитель)

```
 $R_p: (* \text{ Получить подтверждение } *)$   
 $\{ cs \wedge \langle \text{ack}, i, \rho \rangle \in M_p \}$   
begin receive  $\langle \text{ack}, i, \rho \rangle$  ;  $Low := \max(Low, i)$  end
```

---

При открытом сеансе связи процесс-отправитель ожидает от процесса-получателя подтверждения об успешной передаче каждого отправленного слова.

# Прием подтверждения (процесс–отправитель)

```
 $R_p: (* \text{ Получить подтверждение } *)$   
  {  $cs \wedge \langle \text{ack}, i, \rho \rangle \in M_p$  }  
  begin receive  $\langle \text{ack}, i, \rho \rangle$  ;  $Low := \max(Low, i)$  end
```

---

В пакет с подтверждением вкладывается только следующий порядковый номер сообщения, ожидаемого процессом–получателем  $q$ . Как и в предыдущем случае, для проведения анализа мы будем считать, что в пакете с подтверждением также указывается оставшееся время жизни этого пакета.

# Прием подтверждения (процесс–отправитель)

```
 $R_p: (* \text{ Получить подтверждение } *)$   
  {  $cs \wedge \langle \text{ack}, i, \rho \rangle \in M_p$  }  
  begin receive  $\langle \text{ack}, i, \rho \rangle$  ;  $Low := \max(Low, i)$  end
```

---

После получения подтверждения изменяется положение нижней створки окна передачи  $Low$  . Номер  $i$  поступившего подтверждения означает, что все слова с номерами меньшими  $i$  уже были успешно доставлены.

## Формирование записи о потере слова (процесс–отправитель)

$E_p$ :(\* Сформировать сообщение об ошибке  
в связи с возможной потерей слова \*)

$$\{ cs \wedge Ut[B + Low] \leq -2\mu - R \}$$

**begin**  $error[B + Low] := true$  ;  $Low := Low + 1$  **end**

---

Заметим, что при поступлении слова с номером  $Low$  значение соответствующего таймера  $Ut[B + Low]$  было установлено равным  $U$  (протяженность интервала отправления). Поэтому запись о потере слова формируется в том случае, когда истекло время  $U$ , отведенное на отправление этого слова, время, отведенное на его прием  $R$ , время жизни сообщения  $\mu$  и время жизни подтверждения о получении сообщения  $\mu$ .

## Формирование записи о потере слова (процесс–отправитель)

$E_p$ :(\* Сформировать сообщение об ошибке  
в связи с возможной потерей слова \*)  
{  $cs \wedge Ut[B + Low] \leq -2\mu - R$  }  
**begin**  $error[B + Low] := true$  ;  $Low := Low + 1$  **end**

---

В записи об ошибке регистрируется, что слово с номером  $Low$  или подтверждение о его успешной доставке было потеряно, и не следует более ожидать от процесса–получателя поступления подтверждения о его доставке. Такое слово считается «вероятно потерянным».



# Заккрытие сеанса связи (процесс–отправитель)

```
Cp:(* Закреть сеанс связи *)  
  { cs ∧ St < 0 ∧ Low = High }  
  begin B := B + High ; delete(St, High, Low) end  
  (* cs := false *)
```

---

Заккрытие сеанса связи процессом–отправителем происходит после того, как

# Заккрытие сеанса связи (процесс–отправитель)

```
Cp:(* Закреть сеанс связи *)  
  {      ∧ St < 0 ∧      }  
  begin B := B + High ; delete(St, High, Low) end  
  (* cs := false *)
```

---

Заккрытие сеанса связи процессом–отправителем происходит после того, как

- ▶ истекло время отправления всех поступивших слов и

# Заккрытие сеанса связи (процесс–отправитель)

```
Cp:(* Закреть сеанс связи *)  
  {      ∧      ∧ Low = High }  
  begin B := B + High ; delete(St, High, Low) end  
  (* cs := false *)
```

---

Заккрытие сеанса связи процессом–отправителем происходит после того, как

- ▶ все поступившие слова были либо успешно доставлены, либо зарегистрированы как «вероятно потерянные».

# Закрытие сеанса связи (процесс–отправитель)

```
Cp:(* Закрыть сеанс связи *)  
  { cs ∧ St < 0 ∧ Low = High }  
  begin B := B + High ; delete(St, High, Low) end  
  (* cs := false *)
```

---

Закрытие сеанса связи процессом–отправителем происходит после того, как

- ▶ истекло время отправления всех поступивших слов и
- ▶ все поступившие слова были либо успешно доставлены, либо зарегистрированы как «вероятно потерянные».

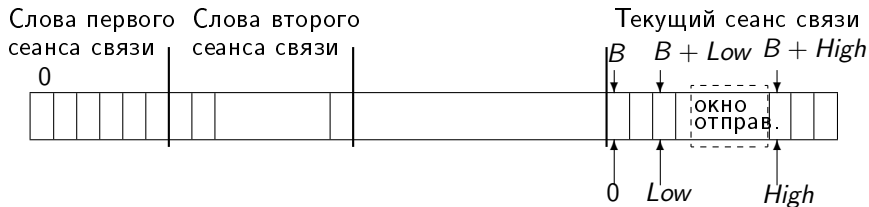
## Закрытие сеанса связи (процесс–отправитель)

```
Cp:(* Закрыть сеанс связи *)  
  { cs ∧ St < 0 ∧ Low = High }  
  begin B := B + High ; delete(St, High, Low) end  
  (* cs := false *)
```

---

$B$  — это вспомогательная переменная, которая введена только для доказательства корректности протокола. Отправитель в каждом сеансе связи проводит нумерацию слов, начиная с  $0$ , и чтобы при анализе протокола проводить различие между словами в разных сеансах связи, мы нумеруем подряд все слова. В том случае, когда отправитель дает некоторому слову номер  $i$ , «абсолютный» номер этого слова будет равен  $B + i$ , где  $B$  — это суммарное число пакетов, поступивших процессу  $p$  в предыдущие сеансы связи. При реализации протокола для переменной  $B$  памяти не отводится, и отправитель «забывает» обо всех словах  $in_p[0..B - 1]$ .

# Закрытие сеанса связи (процесс–отправитель)



Индексация слов в протоколе:

## Описание протокола с таймерами (процесс–получатель)

```
 $R_q : \{ \langle \text{data}, s, i, w, \rho \rangle \in M_q \}$   
begin receive  $\langle \text{data}, s, i, w, \rho \rangle$  ;  
  if  $cr$  then  
    if  $i = Exp$  then  
      begin  $Rt := R ; Exp := i + 1 ;$  deliver  $w$  end  
    else if  $s = true$  then  
      begin create  $(Rt, Exp) ; (* cr := true *)$   
         $Rt := R ; Exp := i + 1 ;$  deliver  $w$   
      end  
    end  
  end  
end  
 $S_q : \{ cr \}$   
begin send  $\langle \text{ack}, Exp, \mu \rangle$  end
```

# Получение сообщения (процесс–получатель)

```
Rq: (* Принять пакет данных *)  
  {  $\langle \mathbf{data}, s, i, w, \rho \rangle \in M_q$  }  
  begin receive( $\mathbf{data}, s, i, w, \rho$ ) ;  
    if cr then  
      if  $i = Exp$  then  
        begin  $Rt := R$  ;  $Exp := i + 1$  ; deliver  $w$  end  
      else if  $s = true$  then  
        begin create ( $Rt, Exp$ ) ; (*  $cr := true$  *)  
           $Rt := R$  ;  $Exp := i + 1$  ; deliver  $w$   
        end  
    end  
end
```

---

Процесс–получатель принимает поступившее сообщение.



# Получение сообщения (процесс–получатель)

```
 $R_q$ : (* Принять пакет данных *)  
  {  $\langle \mathbf{data}, s, i, w, \rho \rangle \in M_q$  }  
  begin receive  $\langle \mathbf{data}, s, i, w, \rho \rangle$  ;  
    if  $cr$  then  
      if  $i = Exp$  then  
        begin  $Rt := R$  ;  $Exp := i + 1$  ; deliver  $w$  end  
      else if  $s = true$  then  
        begin create ( $Rt, Exp$ ) ; (*  $cr := true$  *)  
           $Rt := R$  ;  $Exp := i + 1$  ; deliver  $w$   
        end  
      end  
    end
```

---

и проверяет, был ли уже открыт сеанс связи для приема сообщений.

## Получение сообщения (процесс–получатель)

```
Rq: (* Принять пакет данных *)  
  {  $\langle \mathbf{data}, s, i, w, \rho \rangle \in M_q$  }  
  begin receive  $\langle \mathbf{data}, s, i, w, \rho \rangle$  ;  
    if cr then  
      if i = Exp then  
        begin Rt := R ; Exp := i + 1 ; deliver w end  
      else if s = true then  
        begin create (Rt, Exp) ; (* cr := true *)  
          Rt := R ; Exp := i + 1 ; deliver w  
        end  
      end  
    end  
  end
```

---

Если сеанс связи был открыт, то процесс–получатель выясняет, имеет ли полученное слово ожидаемый номер *Exp* . Если номер полученного слова не совпадает с тем номером, который ожидается, то это слово игнорируется.

# Получение сообщения (процесс–получатель)

```
 $R_q$ : (* Принять пакет данных *)  
{  $\langle \mathbf{data}, s, i, w, \rho \rangle \in M_q$  }  
begin receive  $\langle \mathbf{data}, s, i, w, \rho \rangle$  ;  
  if  $cr$  then  
    if  $i = Exp$  then  
      begin  $Rt := R$  ;  $Exp := i + 1$  ; deliver  $w$  end  
    else if  $s = true$  then  
      begin create ( $Rt, Exp$ ) ; (*  $cr := true$  *)  
         $Rt := R$  ;  $Exp := i + 1$  ; deliver  $w$   
      end  
    end  
  end
```

---

Если номер полученного слова равен ожидаемому номеру  $Exp$  , то слово вручается потребителю, и ожидаемый номер увеличивается.

## Получение сообщения (процесс–получатель)

```
 $R_q$ : (* Принять пакет данных *)  
  {  $\langle \text{data}, s, i, w, \rho \rangle \in M_q$  }  
  begin receive  $\langle \text{data}, s, i, w, \rho \rangle$  ;  
    if  $cr$  then  
      if  $i = Exp$  then  
        begin  $Rt := R$  ;  $Exp := i + 1$  ; deliver  $w$  end  
      else if  $s = true$  then  
        begin create ( $Rt, Exp$ ) ; (*  $cr := true$  *)  
           $Rt := R$  ;  $Exp := i + 1$  ; deliver  $w$   
        end  
      end  
    end
```

---

Если сеанс связи не открыт, то процесс-получатель выясняет, является ли поступившее слово первым словом в передаваемой последовательности слов (проверяет признак начала последовательности  $s$ ). Если это не первое слово последовательности, то оно игнорируется.

# Получение сообщения (процесс–получатель)

```
Rq: (* Принять пакет данных *)  
  {  $\langle \mathbf{data}, s, i, w, \rho \rangle \in M_q$  }  
  begin receive  $\langle \mathbf{data}, s, i, w, \rho \rangle$  ;  
    if cr then  
      if  $i = Exp$  then  
        begin  $Rt := R$  ;  $Exp := i + 1$  ; deliver w end  
      else if  $s = true$  then  
        begin create( $Rt, Exp$ ) ; (*  $cr := true$  *)  
           $Rt := R$  ;  $Exp := i + 1$  ; deliver w  
        end  
      end  
  end
```

---

Если полученное слово является первым словом последовательности, то открывается сеанс связи, устанавливается таймер приема  $Rt$  , и полученное слово вручается потребителю.

## Отправление подтверждений (процесс–получатель)

```
 $S_q$ :(* Отправить подтверждение *)  
  {  $cr$  }  
  begin send  $\langle \text{ack}, Exp, \mu \rangle$  end
```

---

Отправляемое подтверждение свидетельствует о том, что все слова текущего сеанса связи с номерами, меньшими ожидаемого номера  $Exp$ , были успешно доставлены потребителю.

# Описание протокола с таймерами (коммуникационная среда и время)

**Loss** :  $\{ m \in M \}$  (\*  $M$  — это либо  $M_p$ , либо  $M_q$  \*)  
**begin** remove  $m$  from  $M$  **end**

**Dupl** :  $\{ m \in M \}$  (\*  $M$  — это либо  $M_p$ , либо  $M_q$  \*)  
**begin** insert  $m$  in  $M$  **end**

**Time** : (\*  $\delta > 0$  \*)  
**begin** forall  $i$  do  $Ut[i] := Ut[i] - \delta$  ;  
     $St := St - \delta$  ;  
     $Rt := Rt - \delta$  ;  
    if  $Rt \leq 0$  then delete ( $Rt$ ,  $Exp$ ); (\*  $cr := false$  \*)  
    forall  $\langle \dots, \rho \rangle \in M_p, M_q$  do  
        **begin**  $\rho := \rho - \delta$  ;  
            if  $\rho \leq 0$  then remove packet  
        **end**  
**end**  
**end**

# Коммуникационная среда

**Loss:**  $\{ m \in M \}$  (\*  $M$  — это либо  $M_p$ , либо  $M_q$  \*)  
**begin** remove  $m$  from  $M$  **end**

**Dupl:**  $\{ m \in M \}$  (\*  $M$  — это либо  $M_p$ , либо  $M_q$  \*)  
**begin** insert  $m$  in  $M$  **end**

---

Коммуникационная подсистема представлена двумя мультимножествами:  $M_p$  для пакетов, адресованных процессу  $p$ , и  $M_q$  для пакетов, адресованных процессу  $q$ . В системе есть дополнительные переходы, не связанные с действиями протоколов процессов. С их помощью моделируются ошибки потери и дублирования сообщений в каналах связи.



# Коммуникационная среда

```
Time: (*  $\delta > 0$  *)  
  begin forall  $i$  do  $Ut[i] := Ut[i] - \delta$  ;  
     $St := St - \delta$  ;  
     $Rt := Rt - \delta$  ;  
    if  $Rt \leq 0$  then delete ( $Rt, Exp$ ); (*  $cr := false$  *)  
    forall  $\langle \dots, \rho \rangle \in M_p, M_q$  do  
      begin  $\rho := \rho - \delta$  ;  
        if  $\rho \leq 0$  then remove packet  
      end  
    end  
  end
```

---

Действие **Time** приводит к уменьшению показателей всех таймеров системы на одну и ту же величину  $\delta$ , и это как раз соответствует тому, что происходит, когда два дискретных события оказываются разделенными интервалом времени протяженности  $\delta$ .

# Коммуникационная среда

```
Time: (*  $\delta > 0$  *)
  begin forall  $i$  do  $Ut[i] := Ut[i] - \delta$  ;
     $St := St - \delta$  ;
     $Rt := Rt - \delta$  ;
    if  $Rt \leq 0$  then delete ( $Rt$ ,  $Exp$ ); (*  $cr := false$  *)
    forall  $\langle \dots, \rho \rangle \in M_p, M_q$  do
      begin  $\rho := \rho - \delta$  ;
        if  $\rho \leq 0$  then remove packet
      end
    end
  end
```

---

Процесс–получатель закрывает сеанс связи, как только показание таймера получателя  $Rt$  оказывается равным 0 , сеанс связи закрывается. Это происходит, если в течение времени  $R$  , отведенного для приема ожидаемого слова, это слово не было получено.

# Корректность протокола с таймерами

## Свойства безопасности протокола.

1. **Отсутствие потерь.** Каждое слово из массива  $in_p$  будет вручено процессом  $q$  или зарегистрировано процессом  $p$  (как «вероятно потерянное») спустя ограниченный отрезок времени с момента поступления этого слова к процессу  $p$ .
2. **Соблюдение порядка.** Слова, которые вручаются процессом  $q$ , следуют в порядке строгого возрастания номеров в массиве  $in_p$ .

# Корректность протокола с таймерами

Для доказательства этих свойств безопасности воспользуемся методом инвариантов.

Установим инвариантность двух утверждений  $P_0$  (инвариант процесса–отправителя) и  $P_1$  (инвариант процесса–получателя).

# Корректность протокола с таймерами

## Теорема 4.1.

Следующее утверждение является инвариантом протокола с таймерами.

$$P_0 \equiv \quad cs \Rightarrow St \leq S \quad (1)$$

$$\wedge \quad cr \Rightarrow 0 < Rt \leq R \quad (2)$$

$$\wedge \quad \forall i < B + High : Ut[i] \leq U \quad (3)$$

$$\wedge \quad \forall \langle \dots, \rho \rangle \in M_p, M_q : 0 < \rho \leq \mu \quad (4)$$

$$\wedge \quad \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow cs \wedge St \geq \rho + \mu + R \quad (5)$$

$$\wedge \quad cr \Rightarrow cs \wedge St \geq Rt + \mu \quad (6)$$

$$\wedge \quad \langle \mathbf{ack}, i, \rho \rangle \in M_p \Rightarrow cs \wedge St > \rho \quad (7)$$

$$\wedge \quad \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow (w = in_p[B + i] \wedge i < High) \quad (8)$$

Для правильного истолкования соотношения (3), мы будем предполагать, что значение *High* равно нулю во всех тех конфигурациях, в которых отправитель не имеет соединения.

# Инвариантность утверждения $P_0$

Доказательство.

В самом начале вычисления соединения отсутствуют, нет никаких пакетов,  $B = 0$ , и поэтому утверждение  $P_0$  является истинным.

Далее рассмотрим последовательно все действия процессов  $p$  и  $q$ , и покажем выполнимость соотношения

$$\{P_0\} \longrightarrow \{P_0\}$$

# Инвариантность утверждения $P_0$

```
 $A_p$  : begin if not cs then  
        begin create( $St$ ,  $High$ ,  $Low$ ) ; (* cs := true *)  
             $Low := High := 0$  ;  $St := S$  end ;  
         $Ut[B + High] := U$  ;  $High := High + 1$  end
```

---

$$P_0 \equiv cs \Rightarrow St \leq S \quad (1)$$

$$\wedge cr \Rightarrow 0 < Rt \leq R \quad (2)$$

$$\wedge \forall i < B + High : Ut[i] \leq U \quad (3)$$

$$\wedge \forall \langle \dots, \rho \rangle \in M_p, M_q : 0 < \rho \leq \mu \quad (4)$$

$$\wedge \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow cs \wedge St \geq \rho + \mu + R \quad (5)$$

$$\wedge cr \Rightarrow cs \wedge St \geq Rt + \mu \quad (6)$$

$$\wedge \langle \mathbf{ack}, i, \rho \rangle \in M_p \Rightarrow cs \wedge St > \rho \quad (7)$$

$$\wedge \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow (w = in_p[B + i] \wedge i < High) \quad (8)$$

Соотношение (1) сохраняется, т.к. оператор присваивания, в правой части которого стоит  $St$ , всегда приводит к тому, что выполняется равенство  $St = S$ .

# Инвариантность утверждения $P_0$

$A_p$  : **begin if not cs then**

**begin** create( $St, High, Low$ ) ; (\* cs := true \*)  
 $Low := High := 0$  ;  $St := S$  **end** ;  
 $Ut[B + High] := U$  ;  $High := High + 1$  **end**

---

$$P_0 \equiv cs \Rightarrow St \leq S \quad (1)$$

$$\wedge cr \Rightarrow 0 < Rt \leq R \quad (2)$$

$$\wedge \forall i < B + High : Ut[i] \leq U \quad (3)$$

$$\wedge \forall \langle \dots, \rho \rangle \in M_p, M_q : 0 < \rho \leq \mu \quad (4)$$

$$\wedge \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow cs \wedge St \geq \rho + \mu + R \quad (5)$$

$$\wedge cr \Rightarrow cs \wedge St \geq Rt + \mu \quad (6)$$

$$\wedge \langle \mathbf{ack}, i, \rho \rangle \in M_p \Rightarrow cs \wedge St > \rho \quad (7)$$

$$\wedge \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow (w = in_p[B + i] \wedge i < High) \quad (8)$$

Соотношения (2) и (4) сохраняются, т.к. это действие не влияет на переменные этих соотношений



# Инвариантность утверждения $P_0$

```
 $A_p$  : begin if not cs then  
        begin create( $St$ ,  $High$ ,  $Low$ ) ; (* cs := true *)  
               $Low := High := 0$  ;  $St := S$  end ;  
         $Ut[B + High] := U$  ;  $High := High + 1$  end
```

---

$$P_0 \equiv cs \Rightarrow St \leq S \quad (1)$$

$$\wedge cr \Rightarrow 0 < Rt \leq R \quad (2)$$

$$\wedge \forall i < B + High : Ut[i] \leq U \quad (3)$$

$$\wedge \forall \langle \dots, \rho \rangle \in M_p, M_q : 0 < \rho \leq \mu \quad (4)$$

$$\wedge \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow cs \wedge St \geq \rho + \mu + R \quad (5)$$

$$\wedge cr \Rightarrow cs \wedge St \geq Rt + \mu \quad (6)$$

$$\wedge \langle \mathbf{ack}, i, \rho \rangle \in M_p \Rightarrow cs \wedge St > \rho \quad (7)$$

$$\wedge \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow (w = in_p[B + i] \wedge i < High) \quad (8)$$

Соотношение (3) сохраняется, т.к. перед тем, как переменная  $High$  увеличит свое значение, элементу  $Ut[B + High]$  присваивается значение  $U$ .

# Инвариантность утверждения $P_0$

$A_p$  : begin if not cs then

```
begin create(St, High, Low) ; (* cs := true *)  
  Low := High := 0 ; St := S end;  
Ut[B + High] := U ; High := High + 1 end
```

---

$$P_0 \equiv cs \Rightarrow St \leq S \quad (1)$$

$$\wedge cr \Rightarrow 0 < Rt \leq R \quad (2)$$

$$\wedge \forall i < B + High : Ut[i] \leq U \quad (3)$$

$$\wedge \forall \langle \dots, \rho \rangle \in M_p, M_q : 0 < \rho \leq \mu \quad (4)$$

$$\wedge \langle \text{data}, s, i, w, \rho \rangle \in M_q \Rightarrow cs \wedge St \geq \rho + \mu + R \quad (5)$$

$$\wedge cr \Rightarrow cs \wedge St \geq Rt + \mu \quad (6)$$

$$\wedge \langle \text{ack}, i, \rho \rangle \in M_p \Rightarrow cs \wedge St > \rho \quad (7)$$

$$\wedge \langle \text{data}, s, i, w, \rho \rangle \in M_q \Rightarrow (w = in_p[B + i] \wedge i < High) \quad (8)$$

Соотношения (5), (6) и (7) сохраняются, т.к. значение  $St$  может только увеличиться.

# Инвариантность утверждения $P_0$

$A_p$  : **begin if not cs then**

**begin** create( $St, High, Low$ ) ; (\* cs := true \*)

$Low := High := 0$  ;  $St := S$  **end**;

$Ut[B + High] := U$  ;  $High := High + 1$  **end**

---

$$P_0 \equiv cs \Rightarrow St \leq S \quad (1)$$

$$\wedge cr \Rightarrow 0 < Rt \leq R \quad (2)$$

$$\wedge \forall i < B + High : Ut[i] \leq U \quad (3)$$

$$\wedge \forall \langle \dots, \rho \rangle \in M_p, M_q : 0 < \rho \leq \mu \quad (4)$$

$$\wedge \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow cs \wedge St \geq \rho + \mu + R \quad (5)$$

$$\wedge cr \Rightarrow cs \wedge St \geq Rt + \mu \quad (6)$$

$$\wedge \langle \mathbf{ack}, i, \rho \rangle \in M_p \Rightarrow cs \wedge St > \rho \quad (7)$$

$$\wedge \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow (w = in_p[B + i] \wedge i < High) \quad (8)$$

Соотношение (8) сохраняется, поскольку значение  $High$  может только увеличиться.

# Инвариантность утверждения $P_0$

$S_p$  : {  $cs \wedge Low \leq i < High \wedge Ut[B + i] > 0$  }  
**begin** send⟨**data**, ( $i = Low$ ),  $i$ ,  $in_p[B + i]$ ,  $\mu$ ⟩ ;  $St := S$  **end**

---

$$P_0 \equiv cs \Rightarrow St \leq S \quad (1)$$

$$\wedge cr \Rightarrow 0 < Rt \leq R \quad (2)$$

$$\wedge \forall i < B + High : Ut[i] \leq U \quad (3)$$

$$\wedge \forall \langle \dots, \rho \rangle \in M_p, M_q : 0 < \rho \leq \mu \quad (4)$$

$$\wedge \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow cs \wedge St \geq \rho + \mu + R \quad (5)$$

$$\wedge cr \Rightarrow cs \wedge St \geq Rt + \mu \quad (6)$$

$$\wedge \langle \mathbf{ack}, i, \rho \rangle \in M_p \Rightarrow cs \wedge St > \rho \quad (7)$$

$$\wedge \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow (w = in_p[B + i] \wedge i < High) \quad (8)$$

Соотношение (1) сохраняется, т.к.  $St$  всегда будет присвоено значение  $S$ .

# Инвариантность утверждения $P_0$

$S_p$  : {  $cs \wedge Low \leq i < High \wedge Ut[B + i] > 0$  }  
**begin** send(**data**, ( $i = Low$ ),  $i$ ,  $in_p[B + i]$ ,  $\mu$ ) ;  $St := S$  **end**

---

$$P_0 \equiv cs \Rightarrow St \leq S \quad (1)$$

$$\wedge cr \Rightarrow 0 < Rt \leq R \quad (2)$$

$$\wedge \forall i < B + High : Ut[i] \leq U \quad (3)$$

$$\wedge \forall \langle \dots, \rho \rangle \in M_p, M_q : 0 < \rho \leq \mu \quad (4)$$

$$\wedge \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow cs \wedge St \geq \rho + \mu + R \quad (5)$$

$$\wedge cr \Rightarrow cs \wedge St \geq Rt + \mu \quad (6)$$

$$\wedge \langle \mathbf{ack}, i, \rho \rangle \in M_p \Rightarrow cs \wedge St > \rho \quad (7)$$

$$\wedge \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow (w = in_p[B + i] \wedge i < High) \quad (8)$$

Соотношения (2) и (3) сохраняются, т.к. это действие не влияет на переменные этих соотношений

# Инвариантность утверждения $P_0$

$S_p$  : {  $cs \wedge Low \leq i < High \wedge Ut[B + i] > 0$  }  
**begin** send(**data**, ( $i = Low$ ),  $i$ ,  $in_p[B + i]$ ,  $\mu$ ) ;  $St := S$  **end**

---

$$P_0 \equiv cs \Rightarrow St \leq S \quad (1)$$

$$\wedge cr \Rightarrow 0 < Rt \leq R \quad (2)$$

$$\wedge \forall i < B + High : Ut[i] \leq U \quad (3)$$

$$\wedge \forall \langle \dots, \rho \rangle \in M_p, M_q : 0 < \rho \leq \mu \quad (4)$$

$$\wedge \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow cs \wedge St \geq \rho + \mu + R \quad (5)$$

$$\wedge cr \Rightarrow cs \wedge St \geq Rt + \mu \quad (6)$$

$$\wedge \langle \mathbf{ack}, i, \rho \rangle \in M_p \Rightarrow cs \wedge St > \rho \quad (7)$$

$$\wedge \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow (w = in_p[B + i] \wedge i < High) \quad (8)$$

Соотношение (4) сохраняется, т.к. каждый отправляемый пакет имеет время оставшейся жизни равное  $\mu$ .

# Инвариантность утверждения $P_0$

$S_p$  : {  $cs \wedge Low \leq i < High \wedge Ut[B + i] > 0$  }  
**begin** send(**data**, ( $i = Low$ ),  $i$ ,  $in_p[B + i]$ ,  $\mu$ ) ;  $St := S$  **end**

---

$$P_0 \equiv cs \Rightarrow St \leq S \quad (1)$$

$$\wedge cr \Rightarrow 0 < Rt \leq R \quad (2)$$

$$\wedge \forall i < B + High : Ut[i] \leq U \quad (3)$$

$$\wedge \forall \langle \dots, \rho \rangle \in M_p, M_q : 0 < \rho \leq \mu \quad (4)$$

$$\wedge \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow cs \wedge St \geq \rho + \mu + R \quad (5)$$

$$\wedge cr \Rightarrow cs \wedge St \geq Rt + \mu \quad (6)$$

$$\wedge \langle \mathbf{ack}, i, \rho \rangle \in M_p \Rightarrow cs \wedge St > \rho \quad (7)$$

$$\wedge \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow (w = in_p[B + i] \wedge i < High) \quad (8)$$

Соотношение (5) сохраняется, т.к. пакет  $\langle \dots, \mu \rangle$  отправлен,  $St$  полагается равным  $S$ , и  $S = R + 2\mu$ .

# Инвариантность утверждения $P_0$

$S_p$  : {  $cs \wedge Low \leq i < High \wedge Ut[B + i] > 0$  }  
**begin** send(**data**, ( $i = Low$ ),  $i$ ,  $in_p[B + i]$ ,  $\mu$ ) ;  $St := S$  **end**

---

$$P_0 \equiv cs \Rightarrow St \leq S \quad (1)$$

$$\wedge cr \Rightarrow 0 < Rt \leq R \quad (2)$$

$$\wedge \forall i < B + High : Ut[i] \leq U \quad (3)$$

$$\wedge \forall \langle \dots, \rho \rangle \in M_p, M_q : 0 < \rho \leq \mu \quad (4)$$

$$\wedge \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow cs \wedge St \geq \rho + \mu + R \quad (5)$$

$$\wedge cr \Rightarrow cs \wedge St \geq Rt + \mu \quad (6)$$

$$\wedge \langle \mathbf{ack}, i, \rho \rangle \in M_p \Rightarrow cs \wedge St > \rho \quad (7)$$

$$\wedge \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow (w = in_p[B + i] \wedge i < High) \quad (8)$$

Соотношения (6) и (7) сохраняются, т.к. значение  $St$  может только увеличиться.



# Инвариантность утверждения $P_0$

$S_p$  : {  $cs \wedge Low \leq i < High \wedge Ut[B + i] > 0$  }  
**begin** send(**data**, ( $i = Low$ ),  $i$ ,  $in_p[B + i]$ ,  $\mu$ ) ;  $St := S$  **end**

---

$$P_0 \equiv cs \Rightarrow St \leq S \quad (1)$$

$$\wedge cr \Rightarrow 0 < Rt \leq R \quad (2)$$

$$\wedge \forall i < B + High : Ut[i] \leq U \quad (3)$$

$$\wedge \forall \langle \dots, \rho \rangle \in M_p, M_q : 0 < \rho \leq \mu \quad (4)$$

$$\wedge \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow cs \wedge St \geq \rho + \mu + R \quad (5)$$

$$\wedge cr \Rightarrow cs \wedge St \geq Rt + \mu \quad (6)$$

$$\wedge \langle \mathbf{ack}, i, \rho \rangle \in M_p \Rightarrow cs \wedge St > \rho \quad (7)$$

$$\wedge \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow (w = in_p[B + i] \wedge i < High) \quad (8)$$

Соотношение (8) сохраняется, для нового пакета выполняются условия  $w = in_p[B + i]$  и  $i < High$ .

# Инвариантность утверждения $P_0$

$R_p$  : (\* Получить подтверждение \*)

$\{ cs \wedge \langle \text{ack}, i, \rho \rangle \in M_p \}$

**begin** receive  $\langle \text{ack}, i, \rho \rangle$  ;  $Low := \max(Low, i)$  **end**

---

$$P_0 \equiv cs \Rightarrow St \leq S \quad (1)$$

$$\wedge cr \Rightarrow 0 < Rt \leq R \quad (2)$$

$$\wedge \forall i < B + High : Ut[i] \leq U \quad (3)$$

$$\wedge \forall \langle \dots, \rho \rangle \in M_p, M_q : 0 < \rho \leq \mu \quad (4)$$

$$\wedge \langle \text{data}, s, i, w, \rho \rangle \in M_q \Rightarrow cs \wedge St \geq \rho + \mu + R \quad (5)$$

$$\wedge cr \Rightarrow cs \wedge St \geq Rt + \mu \quad (6)$$

$$\wedge \langle \text{ack}, i, \rho \rangle \in M_p \Rightarrow cs \wedge St > \rho \quad (7)$$

$$\wedge \langle \text{data}, s, i, w, \rho \rangle \in M_q \Rightarrow (w = in_p[B + i] \wedge i < High) \quad (8)$$

Действие  $R_p$  не изменяет значений ни одной из переменных, фигурирующих в формуле  $P_0$ , и изъятие пакета сохраняет соотношения (4) и (7).

# Инвариантность утверждения $P_0$

$E_p$  : {  $cs \wedge Ut[B + Low] \leq -2\mu - R$  }  
begin error[B + Low] := true ; Low := Low + 1 end

---

$$P_0 \equiv cs \Rightarrow St \leq S \quad (1)$$

$$\wedge cr \Rightarrow 0 < Rt \leq R \quad (2)$$

$$\wedge \forall i < B + High : Ut[i] \leq U \quad (3)$$

$$\wedge \forall \langle \dots, \rho \rangle \in M_p, M_q : 0 < \rho \leq \mu \quad (4)$$

$$\wedge \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow cs \wedge St \geq \rho + \mu + R \quad (5)$$

$$\wedge cr \Rightarrow cs \wedge St \geq Rt + \mu \quad (6)$$

$$\wedge \langle \mathbf{ack}, i, \rho \rangle \in M_p \Rightarrow cs \wedge St > \rho \quad (7)$$

$$\wedge \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow (w = in_p[B + i] \wedge i < High) \quad (8)$$

Действие  $E_p$  не изменяет значений ни одной из переменных, фигурирующих в формуле  $P_0$  .

# Инвариантность утверждения $P_0$

$C_p$  : {  $cs \wedge St < 0 \wedge Low = High$  }  
**begin**  $B := B + High$  ; delete ( $St, High, Low$ ) **end**  
(\*  $cs := false$  \*)

---

$$P_0 \equiv cs \Rightarrow St \leq S \quad (1)$$

$$\wedge cr \Rightarrow 0 < Rt \leq R \quad (2)$$

$$\wedge \forall i < B + High : Ut[i] \leq U \quad (3)$$

$$\wedge \forall \langle \dots, \rho \rangle \in M_p, M_q : 0 < \rho \leq \mu \quad (4)$$

$$\wedge \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow cs \wedge St \geq \rho + \mu + R \quad (5)$$

$$\wedge cr \Rightarrow cs \wedge St \geq Rt + \mu \quad (6)$$

$$\wedge \langle \mathbf{ack}, i, \rho \rangle \in M_p \Rightarrow cs \wedge St > \rho \quad (7)$$

$$\wedge \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow (w = in_p[B + i] \wedge i < High) \quad (8)$$

Соотношение (1) сохраняется, т.к. после действия  $C_p$  не выполняется его предпосылка.

# Инвариантность утверждения $P_0$

$C_p$  : {  $cs \wedge St < 0 \wedge Low = High$  }  
**begin**  $B := B + High$  ; delete ( $St, High, Low$ ) **end**  
(\*  $cs := false$  \*)

---

$$P_0 \equiv cs \Rightarrow St \leq S \quad (1)$$

$$\wedge cr \Rightarrow 0 < Rt \leq R \quad (2)$$

$$\wedge \forall i < B + High : Ut[i] \leq U \quad (3)$$

$$\wedge \forall \langle \dots, \rho \rangle \in M_p, M_q : 0 < \rho \leq \mu \quad (4)$$

$$\wedge \langle data, s, i, w, \rho \rangle \in M_q \Rightarrow cs \wedge St \geq \rho + \mu + R \quad (5)$$

$$\wedge cr \Rightarrow cs \wedge St \geq Rt + \mu \quad (6)$$

$$\wedge \langle ack, i, \rho \rangle \in M_p \Rightarrow cs \wedge St > \rho \quad (7)$$

$$\wedge \langle data, s, i, w, \rho \rangle \in M_q \Rightarrow (w = in_p[B + i] \wedge i < High) \quad (8)$$

Соотношения (2),(3) и (4) сохраняются, т.к. это действие не влияет на переменные этих соотношений.

## Инвариантность утверждения $P_0$

$C_p$  : {  $cs \wedge St < 0 \wedge Low = High$  }  
begin  $B := B + High$  ; delete ( $St, High, Low$ ) end  
(\*  $cs := false$  \*)

---

$$P_0 \equiv cs \Rightarrow St \leq S \quad (1)$$

$$\wedge cr \Rightarrow 0 < Rt \leq R \quad (2)$$

$$\wedge \forall i < B + High : Ut[i] \leq U \quad (3)$$

$$\wedge \forall \langle \dots, \rho \rangle \in M_p, M_q : 0 < \rho \leq \mu \quad (4)$$

$$\wedge \langle data, s, i, w, \rho \rangle \in M_q \Rightarrow cs \wedge St \geq \rho + \mu + R \quad (5)$$

$$\wedge cr \Rightarrow cs \wedge St \geq Rt + \mu \quad (6)$$

$$\wedge \langle ack, i, \rho \rangle \in M_p \Rightarrow cs \wedge St > \rho \quad (7)$$

$$\wedge \langle data, s, i, w, \rho \rangle \in M_q \Rightarrow (w = in_p[B + i] \wedge i < High) \quad (8)$$

Действие  $C_p$  может нарушить справедливость заключений в соотношениях (5), (6) и (7), но согласно (2), (5), (6) и (7) оно применимо только тогда, когда предпосылки этих соотношений не выполняются.

## Инвариантность утверждения $P_0$

$C_p$  : {  $cs \wedge St < 0 \wedge Low = High$  }  
begin  $B := B + High$  ; delete ( $St, High, Low$ ) end  
(\*  $cs := false$  \*)

---

$$P_0 \equiv cs \Rightarrow St \leq S \quad (1)$$

$$\wedge cr \Rightarrow 0 < Rt \leq R \quad (2)$$

$$\wedge \forall i < B + High : Ut[i] \leq U \quad (3)$$

$$\wedge \forall \langle \dots, \rho \rangle \in M_p, M_q : 0 < \rho \leq \mu \quad (4)$$

$$\wedge \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow cs \wedge St \geq \rho + \mu + R \quad (5)$$

$$\wedge cr \Rightarrow cs \wedge St \geq Rt + \mu \quad (6)$$

$$\wedge \langle \mathbf{ack}, i, \rho \rangle \in M_p \Rightarrow cs \wedge St > \rho \quad (7)$$

$$\wedge \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow (w = in_p[B + i] \wedge i < High) \quad (8)$$

Действие  $C_p$  также изменяет значение переменной  $B$ , но поскольку согласно соотношениям (5) и (7) ни один пакет не находится на этапе пересылки, выполнимость соотношения (8) сохраняется.

## Инвариантность утверждения $P_0$

$R_q : \{ \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \}$   
**begin** receive  $\langle \mathbf{data}, s, i, w, \rho \rangle ;$   
**if**  $cr$  **then**  
    **if**  $i = Exp$  **then begin**  $Rt := R ; Exp := i + 1 ;$  deliver  $w$  **end**  
    **else if**  $s = true$  **then begin** create  $(Rt, Exp) ; (* cr := true *)$   
         $Rt := R ; Exp := i + 1 ;$  deliver  $w$  **end**  
**end**

---

$$cs \Rightarrow St \leq S \quad (1)$$

$$\wedge \forall i < B + High : Ut[i] \leq U \quad (3)$$

$$\wedge \forall \langle \dots, \rho \rangle \in M_p, M_q : 0 < \rho \leq \mu \quad (4)$$

$$\wedge \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow cs \wedge St \geq \rho + \mu + R \quad (5)$$

$$\wedge \langle \mathbf{ack}, i, \rho \rangle \in M_p \Rightarrow cs \wedge St > \rho \quad (7)$$

$$\wedge \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow (w = in_p[B + i] \wedge i < High) \quad (8)$$

Соотношения (1),(3-5) и (7-8) сохраняются, т.к. это действие не влияет на переменные этих соотношений.



## Инвариантность утверждения $P_0$

```
 $R_q$  : {  $\langle \mathbf{data}, s, i, w, \rho \rangle \in M_q$  }  
begin receive  $\langle \mathbf{data}, s, i, w, \rho \rangle$  ;  
if  $cr$  then  
  if  $i = Exp$  then begin  $Rt := R$  ;  $Exp := i + 1$  ; deliver  $w$  end  
  else if  $s = true$  then begin create  $(Rt, Exp)$  ; (*  $cr := true$  *)  
     $Rt := R$  ;  $Exp := i + 1$  ; deliver  $w$  end  
end
```

$$cr \Rightarrow 0 < Rt \leq R \quad (2)$$

$$\wedge \forall \langle \dots, \rho \rangle \in M_p, M_q : 0 < \rho \leq \mu \quad (4)$$

$$\wedge \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow cs \wedge St \geq \rho + \mu + R \quad (5)$$

$$\wedge cr \Rightarrow cs \wedge St \geq Rt + \mu \quad (6)$$

Соотношение (2) сохраняется, т.к. переменной  $Rt$  всегда присваивается значение  $R$  (если присваивание имеет место).

Соотношение (6) сохраняется, т.к. переменной  $Rt$  присваивается значение  $R$  только при получении пакета  $\langle \mathbf{data}, s, i, w, \rho \rangle$ , а из (4) и (5) следует, что в этом случае выполняется формула  $cs \wedge St \geq R + \mu$ .

# Инвариантность утверждения $P_0$

$S_q$  : (\* Отправить подтверждение \*)  
  {  $cr$  }  
  **begin** send  $\langle \text{ack}, Exp, \mu \rangle$  **end**

---

$$P_0 \equiv cs \Rightarrow St \leq S \quad (1)$$

$$\wedge cr \Rightarrow 0 < Rt \leq R \quad (2)$$

$$\wedge \forall i < B + High : Ut[i] \leq U \quad (3)$$

$$\wedge \forall \langle \dots, \rho \rangle \in M_p, M_q : 0 < \rho \leq \mu \quad (4)$$

$$\wedge \langle \text{data}, s, i, w, \rho \rangle \in M_q \Rightarrow cs \wedge St \geq \rho + \mu + R \quad (5)$$

$$\wedge cr \Rightarrow cs \wedge St \geq Rt + \mu \quad (6)$$

$$\wedge \langle \text{ack}, i, \rho \rangle \in M_p \Rightarrow cs \wedge St > \rho \quad (7)$$

$$\wedge \langle \text{data}, s, i, w, \rho \rangle \in M_q \Rightarrow (w = in_p[B + i] \wedge i < High) \quad (8)$$

Соотношения (1-3), (5-6) и (8) сохраняются, т.к. это действие не влияет на переменные этих соотношений.

# Инвариантность утверждения $P_0$

$S_q$  : (\* Отправить подтверждение \*)  
  {  $cr$  }  
  **begin** send  $\langle \text{ack}, Exp, \mu \rangle$  **end**

---

$$P_0 \equiv cs \Rightarrow St \leq S \quad (1)$$

$$\wedge cr \Rightarrow 0 < Rt \leq R \quad (2)$$

$$\wedge \forall i < B + High : Ut[i] \leq U \quad (3)$$

$$\wedge \forall \langle \dots, \rho \rangle \in M_p, M_q : 0 < \rho \leq \mu \quad (4)$$

$$\wedge \langle \text{data}, s, i, w, \rho \rangle \in M_q \Rightarrow cs \wedge St \geq \rho + \mu + R \quad (5)$$

$$\wedge cr \Rightarrow cs \wedge St \geq Rt + \mu \quad (6)$$

$$\wedge \langle \text{ack}, i, \rho \rangle \in M_p \Rightarrow cs \wedge St > \rho \quad (7)$$

$$\wedge \langle \text{data}, s, i, w, \rho \rangle \in M_q \Rightarrow (w = in_p[B + i] \wedge i < High) \quad (8)$$

Соотношение (4) сохраняется, т.к. при отправлении каждого пакета оставшееся время его жизни полагается равным  $\mu$ .

# Инвариантность утверждения $P_0$

$S_q$  : (\* Отправить подтверждение \*)  
  {  $cr$  }  
  begin send  $\langle \text{ack}, \text{Exp}, \mu \rangle$  end

---

$$P_0 \equiv cs \Rightarrow St \leq S \quad (1)$$

$$\wedge cr \Rightarrow 0 < Rt \leq R \quad (2)$$

$$\wedge \forall i < B + High : Ut[i] \leq U \quad (3)$$

$$\wedge \forall \langle \dots, \rho \rangle \in M_p, M_q : 0 < \rho \leq \mu \quad (4)$$

$$\wedge \langle \text{data}, s, i, w, \rho \rangle \in M_q \Rightarrow cs \wedge St \geq \rho + \mu + R \quad (5)$$

$$\wedge cr \Rightarrow cs \wedge St \geq Rt + \mu \quad (6)$$

$$\wedge \langle \text{ack}, i, \rho \rangle \in M_p \Rightarrow cs \wedge St > \rho \quad (7)$$

$$\wedge \langle \text{data}, s, i, w, \rho \rangle \in M_q \Rightarrow (w = in_p[B + i] \wedge i < High) \quad (8)$$

Соотношение (7) сохраняется, т.к. при отправлении пакета  $\langle \text{ack}, i, \rho \rangle$  в случае, когда  $cr$  имеет значение true, выполняется равенство  $\rho = \mu$ , и поэтому из соотношений (2) и (6) следует  $St > \mu$ .

# Инвариантность утверждения $P_0$

**Loss** :  $\{ m \in M \}$  (\*  $M$  — это либо  $M_p$ , либо  $M_q$  \*)  
**begin** remove  $m$  from  $M$  **end**

---

$$P_0 \equiv cs \Rightarrow St \leq S \quad (1)$$

$$\wedge cr \Rightarrow 0 < Rt \leq R \quad (2)$$

$$\wedge \forall i < B + High : Ut[i] \leq U \quad (3)$$

$$\wedge \forall \langle \dots, \rho \rangle \in M_p, M_q : 0 < \rho \leq \mu \quad (4)$$

$$\wedge \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow cs \wedge St \geq \rho + \mu + R \quad (5)$$

$$\wedge cr \Rightarrow cs \wedge St \geq Rt + \mu \quad (6)$$

$$\wedge \langle \mathbf{ack}, i, \rho \rangle \in M_p \Rightarrow cs \wedge St > \rho \quad (7)$$

$$\wedge \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow (w = in_p[B + i] \wedge i < High) \quad (8)$$

Соотношения (1-3) и (6) сохраняются, т.к. это действие не влияет на переменные этих соотношений. Соотношения (4-5) и (7-8) сохраняются, т.к. изъятие пакета может привести лишь к тому, что нарушатся предпосылки этих соотношений.

## Инвариантность утверждения $P_0$

**Dupl** :{  $m \in M$  } (\*  $M$  — это либо  $M_p$ , либо  $M_q$  \*)  
begin insert  $m$  in  $M$  end

---

$$P_0 \equiv cs \Rightarrow St \leq S \quad (1)$$

$$\wedge cr \Rightarrow 0 < Rt \leq R \quad (2)$$

$$\wedge \forall i < B + High : Ut[i] \leq U \quad (3)$$


$$\wedge \forall \langle \dots, \rho \rangle \in M_p, M_q : 0 < \rho \leq \mu \quad (4)$$

$$\wedge \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow cs \wedge St \geq \rho + \mu + R \quad (5)$$

$$\wedge cr \Rightarrow cs \wedge St \geq Rt + \mu \quad (6)$$

$$\wedge \langle \mathbf{ack}, i, \rho \rangle \in M_p \Rightarrow cs \wedge St > \rho \quad (7)$$

$$\wedge \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow (w = in_p[B + i] \wedge i < High) \quad (8)$$

Соотношения (1-3) и (6) сохраняются, т.к. это действие не влияет на переменные этих соотношений. Соотношения (4-5) и (7-8) сохраняются, т.к. дублирующая вставка пакета  $m$  возможна только тогда, когда пакет  $m$  уже содержится в канале; отсюда следует, что следствия соответствующих соотношений уже были выполнены уже перед самой вставкой. 

## Инвариантность утверждения $P_0$

```
Time :begin forall  $i$  do  $Ut[i] := Ut[i] - \delta$  ;  $St := St - \delta$  ;  $Rt := Rt - \delta$   
      if  $Rt \leq 0$  then delete ( $Rt$ ,  $Exp$ ); (*  $cr := false$  *)  
      forall  $\langle \dots, \rho \rangle \in M_p, M_q$  do  
        begin  $\rho := \rho - \delta$  ; if  $\rho \leq 0$  then remove packet end  
      end  
end
```

---

$$P_0 \equiv cs \Rightarrow St \leq S \quad (1)$$

$$\wedge cr \Rightarrow 0 < Rt \leq R \quad (2)$$

$$\wedge \forall i < B + High : Ut[i] \leq U \quad (3)$$

$$\wedge \forall \langle \dots, \rho \rangle \in M_p, M_q : 0 < \rho \leq \mu \quad (4)$$

$$\wedge \langle data, s, i, w, \rho \rangle \in M_q \Rightarrow cs \wedge St \geq \rho + \mu + R \quad (5)$$

$$\wedge cr \Rightarrow cs \wedge St \geq Rt + \mu \quad (6)$$

$$\wedge \langle ack, i, \rho \rangle \in M_p \Rightarrow cs \wedge St > \rho \quad (7)$$

$$\wedge \langle data, s, i, w, \rho \rangle \in M_q \Rightarrow (w = in_p[B + i] \wedge i < High) \quad (8)$$

Соотношения (1), (2) и (3) сохраняется, т.к. значения переменных  $St$ ,  $Rt$  и  $Ut[i]$  могут лишь уменьшиться, и сеанс связи получателя закрывается, когда значение  $Rt$  становится равным 0 .

## Инвариантность утверждения $P_0$

**Time** :begin forall  $i$  do  $Ut[i] := Ut[i] - \delta$  ;  $St := St - \delta$  ;  $Rt := Rt - \delta$   
if  $Rt \leq 0$  then delete ( $Rt$ ,  $Exp$ ); (\*  $cr := false$  \*)  
forall  $\langle \dots, \rho \rangle \in M_p, M_q$  do  
begin  $\rho := \rho - \delta$  ; if  $\rho \leq 0$  then remove packet end  
end

---

$$P_0 \equiv cs \Rightarrow St \leq S \quad (1)$$

$$\wedge cr \Rightarrow 0 < Rt \leq R \quad (2)$$

$$\wedge \forall i < B + High : Ut[i] \leq U \quad (3)$$

$$\wedge \forall \langle \dots, \rho \rangle \in M_p, M_q : 0 < \rho \leq \mu \quad (4)$$

$$\wedge \langle data, s, i, w, \rho \rangle \in M_q \Rightarrow cs \wedge St \geq \rho + \mu + R \quad (5)$$

$$\wedge cr \Rightarrow cs \wedge St \geq Rt + \mu \quad (6)$$

$$\wedge \langle ack, i, \rho \rangle \in M_p \Rightarrow cs \wedge St > \rho \quad (7)$$

$$\wedge \langle data, s, i, w, \rho \rangle \in M_q \Rightarrow (w = in_p[B + i] \wedge i < High) \quad (8)$$

Соотношение (4) сохраняется, т.к. значение  $\rho$  может лишь уменьшиться, и пакет изымается, когда значение его поля  $\rho$  становится равным 0 .



## Инвариантность утверждения $P_0$

```
Time :begin forall  $i$  do  $Ut[i] := Ut[i] - \delta$  ;  $St := St - \delta$  ;  $Rt := Rt - \delta$   
      if  $Rt \leq 0$  then delete ( $Rt$ ,  $Exp$ ); (*  $cr := false$  *)  
      forall  $\langle \dots, \rho \rangle \in M_p, M_q$  do  
        begin  $\rho := \rho - \delta$  ; if  $\rho \leq 0$  then remove packet end  
      end  
end
```

---

$$P_0 \equiv cs \Rightarrow St \leq S \quad (1)$$

$$\wedge cr \Rightarrow 0 < Rt \leq R \quad (2)$$

$$\wedge \forall i < B + High : Ut[i] \leq U \quad (3)$$

$$\wedge \forall \langle \dots, \rho \rangle \in M_p, M_q : 0 < \rho \leq \mu \quad (4)$$

$$\wedge \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow cs \wedge St \geq \rho + \mu + R \quad (5)$$

$$\wedge cr \Rightarrow cs \wedge St \geq Rt + \mu \quad (6)$$

$$\wedge \langle \mathbf{ack}, i, \rho \rangle \in M_p \Rightarrow cs \wedge St > \rho \quad (7)$$

$$\wedge \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \Rightarrow (w = in_p[B + i] \wedge i < High) \quad (8)$$

Действие **Time** уменьшает значения всех таймеров (и тех, которые содержатся в поле  $\rho$  пакетов) на одну и ту же величину. Поэтому сохраняются все соотношения вида

$Xt \geq Yt + C$ , где  $Xt$  и  $Yt$  — таймеры, а  $C$  — константа. 

## Инвариантность утверждения $P_0$

**Time** :begin forall  $i$  do  $Ut[i] := Ut[i] - \delta$  ;  $St := St - \delta$  ;  $Rt := Rt - \delta$   
if  $Rt \leq 0$  then delete ( $Rt, Exp$ ); (\*  $cr := false$  \*)  
forall  $\langle \dots, \rho \rangle \in M_p, M_q$  do  
begin  $\rho := \rho - \delta$  ; if  $\rho \leq 0$  then remove packet end  
end

---

$$P_0 \equiv cs \Rightarrow St \leq S \quad (1)$$

$$\wedge cr \Rightarrow 0 < Rt \leq R \quad (2)$$

$$\wedge \forall i < B + High : Ut[i] \leq U \quad (3)$$

$$\wedge \forall \langle \dots, \rho \rangle \in M_p, M_q : 0 < \rho \leq \mu \quad (4)$$

$$\wedge \langle data, s, i, w, \rho \rangle \in M_q \Rightarrow cs \wedge St \geq \rho + \mu + R \quad (5)$$

$$\wedge cr \Rightarrow cs \wedge St \geq Rt + \mu \quad (6)$$

$$\wedge \langle ack, i, \rho \rangle \in M_p \Rightarrow cs \wedge St > \rho \quad (7)$$

$$\wedge \langle data, s, i, w, \rho \rangle \in M_q \Rightarrow (w = in_p[B + i] \wedge i < High) \quad (8)$$

В соотношении (8) нет таймеров, и поэтому оно сохраняется.

# Корректность протокола с таймерами

Первое требование, которое предъявляется к рассматриваемому протоколу, состоит в том, что каждое слово должно быть рано или поздно доставлено по назначению или занесено в отчет как утраченное. Определим предикат  $Ok(i)$  следующей формулой

$Ok(i) \Leftrightarrow error[i] = true \vee$  процессу  $q$  было доставлено слово  $in_p[i]$ .

Покажем, что протокол не теряет ни одного слова, не отметив этого факта.

# Корректность протокола с таймерами

## Теорема 4.2.

Следующее утверждение является инвариантом протокола с таймерами.

$$P_1 \equiv P_0 \wedge \neg cs \implies \forall i < B : Ok(i) \quad (9)$$

$$\wedge cs \implies \forall i < B + Low : Ok(i) \quad (10)$$

$$\wedge \langle \mathbf{data}, true, l, w, \rho \rangle \in M_q \implies \forall i < B + l : Ok(i) \quad (11)$$

$$\wedge cr \implies \forall i < B + Exp : Ok(i) \quad (12)$$

$$\wedge \langle \mathbf{ack}, l, \rho \rangle \in M_p \implies \forall i < B + l : Ok(i) \quad (13)$$

Доказательство.

САМОСТОЯТЕЛЬНО.

## Корректность протокола с таймерами

Сделав дополнительное допущение, мы можем доказать теперь первую часть спецификации протокола. Без этого допущения отправитель может оказаться чересчур «ленивым» при составлении отчета о потерянных словах, а в алгоритме всего лишь указано, что сообщение о потере **не может быть** сформировано в течение  $2\mu + R$  единиц времени, следующих за окончанием интервала отправления слова, но там не сказано о том, что это сообщение рано или поздно обязано быть сформировано.

## Корректность протокола с таймерами

Сделав дополнительное допущение, мы можем доказать теперь первую часть спецификации протокола. Без этого допущения отправитель может оказаться чересчур «ленивым» при составлении отчета о потерянных словах, а в алгоритме всего лишь указано, что сообщение о потере **не может быть** сформировано в течение  $2\mu + R$  единиц времени, следующих за окончанием интервала отправления слова, но там не сказано о том, что это сообщение рано или поздно обязано быть сформировано. Поэтому сделаем дополнительное допущение о том, что действие  $E_p$  рано или поздно будет выполнено процессом  $p$  в разумные сроки, скажем, **до того**, как  $Ut[B + Low]$  станет равным  $-2\mu - R - \lambda$ .

# Корректность протокола с таймерами

Теорема 4.3. (об отсутствии потерь)

Каждое слова массива  $in_p$  либо достигает процесса  $q$  , либо отмечается процессом  $p$  как утраченное в течение  $U + 2\mu + R + \lambda$  единиц времени после поступления этого слова процессу  $p$ .

# Корректность протокола с таймерами

## Доказательство

При поступлении слова  $in_p[l]$  выполняется неравенство  $B + High > l$ , и оно продолжает выполняться далее.



# Корректность протокола с таймерами

## Доказательство

При поступлении слова  $in_p[l]$  выполняется неравенство  $B + High > l$ , и оно продолжает выполняться далее. Если сеанс связи закрывается спустя предписанный период времени после поступления слова  $in_p[l]$ , то  $B > l$ , и утверждение теоремы следует из соотношения (9):  $\neg cs \implies \forall i < B : Ok(i)$ .

# Корректность протокола с таймерами

## Доказательство

При поступлении слова  $in_p[l]$  выполняется неравенство  $B + High > l$ , и оно продолжает выполняться далее. Если сеанс связи закрывается спустя предписанный период времени после поступления слова  $in_p[l]$ , то  $B > l$ , и утверждение теоремы следует из соотношения (9):  $\neg cs \implies \forall i < B : Ok(i)$ .

Если сеанс связи остается открытым, и  $B + Low \leq l$ , то все слова с номерами из отрезке  $B + Low \dots l$  будут отмечены в отчете спустя  $2\mu + R$  единиц времени по окончании интервала времени, отведенного для отправления слова  $in_p[l]$ .

# Корректность протокола с таймерами

## Доказательство

При поступлении слова  $in_p[I]$  выполняется неравенство  $B + High > I$ , и оно продолжает выполняться далее. Если сеанс связи закрывается спустя предписанный период времени после поступления слова  $in_p[I]$ , то  $B > I$ , и утверждение теоремы следует из соотношения (9):  $\neg cs \implies \forall i < B : Ok(i)$ .

Если сеанс связи остается открытым, и  $B + Low \leq I$ , то все слова с номерами из отрезке  $B + Low \dots I$  будут отмечены в отчете спустя  $2\mu + R$  единиц времени по окончании интервала времени, отведенного для отправления слова  $in_p[I]$ . Поэтому запись в отчете будут сделана  $2\mu + R + \lambda$  единиц времени после окончания срока, отведенного для отправления сообщения, т.е. спустя  $U + 2\mu + R + \lambda$  единиц времени после поступления слова.

# Корректность протокола с таймерами

## Доказательство

При поступлении слова  $in_p[I]$  выполняется неравенство  $B + High > I$ , и оно продолжает выполняться далее. Если сеанс связи закрывается спустя предписанный период времени после поступления слова  $in_p[I]$ , то  $B > I$ , и утверждение теоремы следует из соотношения (9):  $\neg cs \implies \forall i < B : Ok(i)$ .

Если сеанс связи остается открытым, и  $B + Low \leq I$ , то все слова с номерами из отрезке  $B + Low \dots I$  будут отмечены в отчете спустя  $2\mu + R$  единиц времени по окончании интервала времени, отведенного для отправления слова  $in_p[I]$ . Поэтому запись в отчете будет сделана  $2\mu + R + \lambda$  единиц времени после окончания срока, отведенного для отправления сообщения, т.е. спустя  $U + 2\mu + R + \lambda$  единиц времени после поступления слова. Поэтому  $I < B + Low$ , и рассматриваемое слово будет либо отмечено в отчете, либо доставлено по назначению согласно соотношению (10):  $cs \implies \forall i < B + Low : Ok(i)$ .  $\square$

## Корректность протокола с таймерами

Чтобы установить второе требование корректности протокола, необходимо показать, что каждое слово, которое было вручено потребителю, имеет больший порядковый номер (в массиве  $in_p$ ), чем ранее врученное слово.

# Корректность протокола с таймерами

Чтобы установить второе требование корректности протокола, необходимо показать, что каждое слово, которое было вручено потребителю, имеет больший порядковый номер (в массиве  $in_p$ ), чем ранее врученное слово. Воспользуемся записью  $pr$  для обозначения порядкового номера самого последнего из врученных потребителю слов (для удобства будем полагать, что вначале  $pr = -1$  и  $Ut[-1] = -\infty$ ).

# Корректность протокола с таймерами

## Теорема 4.4.

Следующее утверждение является инвариантом протокола с таймерами.

$$\begin{aligned} P_2 \equiv & P_1 \\ \wedge & \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \implies Ut[B + i] > \rho - \mu & (14) \\ \wedge & i_1 \leq i_2 < B + High \implies Ut[i_1] \leq Ut[i_2] & (15) \\ \wedge & cr \implies Rt \geq Ut[pr] + \mu & (16) \\ \wedge & pr < B + High \wedge (Ut[pr] > -\mu \implies cr) & (17) \\ \wedge & cr \implies B + Exp = pr + 1 & (18) \end{aligned}$$

Доказательство.

САМОСТОЯТЕЛЬНО.

# Корректность протокола с таймерами

Теорема 4.4. (о сохранении порядка вручения)

Слова, доставленные процессом  $q$  потребителю, расположены в строго возрастающем порядке в массиве  $in_p$ .

Доказательство

Прежде всего заметим, что из формулы  $P_2$  следует соотношение  $\langle \mathbf{data}, s, i_1, w, \rho \rangle \in M_q \implies (cr \vee B + i_1 > pr)$ .



# Корректность протокола с таймерами

Теорема 4.4. (о сохранении порядка вручения)

Слова, доставленные процессом  $q$  потребителю, расположены в строго возрастающем порядке в массиве  $in_p$ .

Доказательство

Прежде всего заметим, что из формулы  $P_2$  следует соотношение  $\langle \mathbf{data}, s, i_1, w, \rho \rangle \in M_q \implies (cr \vee B + i_1 > pr)$ .

Действительно, согласно соотношению

(14):  $\langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \implies Ut[B + i] > \rho - \mu$

из  $\langle \mathbf{data}, s, i_1, w, \rho \rangle \in M_q$  следует  $Ut[B + i_1] > \rho - \mu > -\mu$ .

# Корректность протокола с таймерами

Теорема 4.4. (о сохранении порядка вручения)

Слова, доставленные процессом  $q$  потребителю, расположены в строго возрастающем порядке в массиве  $in_p$ .

Доказательство

Прежде всего заметим, что из формулы  $P_2$  следует соотношение  $\langle \mathbf{data}, s, i_1, w, \rho \rangle \in M_q \implies (cr \vee B + i_1 > pr)$ .

Действительно, согласно соотношению

$$(14): \langle \mathbf{data}, s, i, w, \rho \rangle \in M_q \implies Ut[B + i] > \rho - \mu$$

из  $\langle \mathbf{data}, s, i_1, w, \rho \rangle \in M_q$  следует  $Ut[B + i_1] > \rho - \mu > -\mu$ .

Если  $B + i_1 \leq pr$ , то, учитывая, что согласно соотношению

$$(15): i_1 \leq i_2 < B + High \implies Ut[i_1] \leq Ut[i_2]$$

справедливо неравенство  $pr < B + High$ ,

получаем неравенство  $Ut[pr] > -\mu$ , из которого на основании соотношения

$$(17): pr < B + High \wedge (Ut[pr] > -\mu \implies cr)$$

следует истинность предиката  $cr$ .

# Корректность протокола с таймерами

## Доказательство (продолжение)

Предположим, что процесс  $q$  получает пакет  $\langle \mathbf{data}, s, i_1, w, \rho \rangle$  и вручает потребителю слово  $w$ .

# Корректность протокола с таймерами

## Доказательство (продолжение)

Предположим, что процесс  $q$  получает пакет  $\langle \mathbf{data}, s, i_1, w, \rho \rangle$  и вручает потребителю слово  $w$ .

Если перед получением не был открыт сеанс связи (т.е.  $cr = false$ ), то из соотношения

$$\langle \mathbf{data}, s, i_1, w, \rho \rangle \in M_q \implies (cr \vee B + i_1 > pr)$$

следует  $B + i_1 > pr$ , и поэтому слово  $w$  в массиве  $in_p$  располагалось в позиции, номер которой следует за  $pr$ .

# Корректность протокола с таймерами

## Доказательство (продолжение)

Предположим, что процесс  $q$  получает пакет  $\langle \mathbf{data}, s, i_1, w, \rho \rangle$  и вручает потребителю слово  $w$ .

Если перед получением не был открыт сеанс связи (т.е.  $cr = false$ ), то из соотношения

$$\langle \mathbf{data}, s, i_1, w, \rho \rangle \in M_q \implies (cr \vee B + i_1 > pr)$$

следует  $B + i_1 > pr$ , и поэтому слово  $w$  в массиве  $in_p$  располагалось в позиции, номер которой следует за  $pr$ .

Если сеанс связи был открыт (т.е.  $cr = true$ ), то  $i_1 = Exp$ .

Тогда, из соотношения (18):  $cr \implies B + Exp = pr + 1$  следуют равенства

$$B + i_1 = B + Exp = pr + 1,$$

из которых вытекает, что  $w = in_p[pr + 1]$ .



# Качественный анализ протокола

Хорошим считается такое решение, которое не только удовлетворяет требованиям **Отсутствия потерь** и **Сохранение порядка вручения**, но также сообщает об утрате как можно меньшего числа слов. Чтобы достичь этой цели мы должны снабдить протокол, описанный в этом разделе, некоторым механизмом, который позволял бы отправлять каждое слово повторно (на протяжении интервала отправления слова) до тех пор, пока не будет получено подтверждение о его доставке. Интервал отправления должен быть достаточно протяженным, чтобы позволить осуществлять многократную передачу определенных слов и сделать тем самым вероятность утраты слова как можно меньшей.

## Ограниченные порядковые номера

Порядковые номера, используемые в протоколе, можно ограничить, доказав утверждение, аналогичное Теореме 3.8 для симметричного протокола раздвижного окна.

## Ограниченные порядковые номера

Порядковые номера, используемые в протоколе, можно ограничить, доказав утверждение, аналогичное Теореме 3.8 для симметричного протокола раздвижного окна.

Для этого необходимо ввести допущение о том, что скорость поступления слов на вход процесса  $p$  ограничена таким образом, чтобы очередное слово поступало спустя не менее  $U + 2\mu + R$  единиц времени после поступления  $L$ -го по порядку предшествующего слова.



# Ограниченные порядковые номера

Порядковые номера, используемые в протоколе, можно ограничить, доказав утверждение, аналогичное Теореме 3.8 для симметричного протокола раздвижного окна.

Для этого необходимо ввести допущение о том, что скорость поступления слов на вход процесса  $p$  ограничена таким образом, чтобы очередное слово поступало спустя не менее  $U + 2\mu + R$  единиц времени после поступления  $L$ -го по порядку предшествующего слова.

Для этого действие  $A_p$  нужно снабдить предохранителем

$$\{(High < L) \vee (Ut[B + High - L] < -R - 2\mu)\}.$$

## Ограниченные порядковые номера

Порядковые номера, используемые в протоколе, можно ограничить, доказав утверждение, аналогичное Теореме 3.8 для симметричного протокола раздвижного окна.

Для этого необходимо ввести допущение о том, что скорость поступления слов на вход процесса  $p$  ограничена таким образом, чтобы очередное слово поступало спустя не менее  $U + 2\mu + R$  единиц времени после поступления  $L$ -го по порядку предшествующего слова.

Для этого действие  $A_p$  нужно снабдить предохранителем

$$\{(High < L) \vee (Ut[B + High - L] < -R - 2\mu)\}.$$

Тогда порядковые номера полученных пакетов данных отстают не более, чем на  $2L$  от ожидаемого номера  $Exp$ , а порядковые номера в подтверждающих сообщениях отстают не более, чем на  $L$  от номера  $High$ . Следовательно, при передаче увеличение порядковых номеров можно проводить по модулю  $2L$ .

## Несколько слов об инвариантах

Все соотношения (дизъюнкты), входящие в состав рассмотренных нами инвариантов и касающиеся пакетов, имеют следующий общий вид:

$$\forall m \in M : A(m),$$

Нетрудно убедиться, что выполнимость соотношений такого вида сохраняется при дублировании или потере пакетов.

## Несколько слов об инвариантах

Все соотношения (дизъюнкты), входящие в состав рассмотренных нами инвариантов и касающиеся пакетов, имеют следующий общий вид:

$$\forall m \in M : A(m),$$

Нетрудно убедиться, что выполнимость соотношений такого вида сохраняется при дублировании или потере пакетов.

Иногда приходится использовать инварианты более общего вида, например

$$\sum_{m \in M} f(m) = K$$

или

$$\text{предусловие} \implies \exists m \in M : A(m).$$

Предложения такого вида могут утратить истинность в случае потери или дублирования пакетов, и поэтому их нельзя использовать при доказательстве корректности алгоритмов, от которых требуется устойчивость к подобным сбоям.

# Задачи

## Задача 1.

В протоколе с таймерами отправитель может занести в отчет слово как возможно утраченное, в то время как это слово было благополучно доставлено получателю.

1. Опишите выполнение этого протокола, в ходе которого происходит подобный эффект.
2. Можно ли разработать такой протокол, в котором отправитель за ограниченное время составляет отчет об ошибке **в том и только том случае**, когда слово не доставляется получателю?

# Задачи

## Задача 2.

Предположим, что в связи с выходом из строя часового механизма, получатель не может закрыть сеанс связи вовремя. Опишите вычисление протокола с таймерами, в ходе которого слово будет утрачено, но отправитель не сможет отметить это в отчете.

## Задача 3.

Опишите такое вычисление протокола с таймерами, в ходе которого получатель открывает сеанс связи после получения пакета с порядковым номером большим нуля.

# Задачи

## Задача 4.

Проектировщик сети хотел бы воспользоваться протоколом с таймерами, но при этом желает, чтобы о возможно утраченных словах запись в отчете осуществлялась пораньше. Для этого он модифицирует действие  $E_p$  следующим образом:

```
 $E_p$ : (* Сформировать отчет об ошибке  
для возможно утраченного слова *)  
{  $Ut[B + Low] < 0$  }  
begin  $error[B + Low] := true$  ;  $Low := Low + 1$  end
```

Будет ли модифицированный таким образом протокол удовлетворять требованиям **Отсутствия потерь** и **Сохранения порядка вручения** или для этого необходимо внести также другие изменения?

Каковы, по Вашему мнению, преимущества и недостатки указанной модификации.

КОНЕЦ ЛЕКЦИИ 4.