

Лифт в здании

В 3-этажном здании есть один лифт, вмещающий не более 10 человек. На вход этого здания на 1-м этаже поступает поток посетителей. Первый этаж – технический, и посетители его не посещают. Они направляются на 2 и 3-й этажи с вероятностями P_2 , P_3 соответственно (полагаем $P_2+P_3=1$), пользуясь лифтом. Посетитель, попав на свой этаж, находится на нем в течение случайного времени T_p . После этого он направляется к лифту, на нем опускается на 1-й этаж и покидает здание.

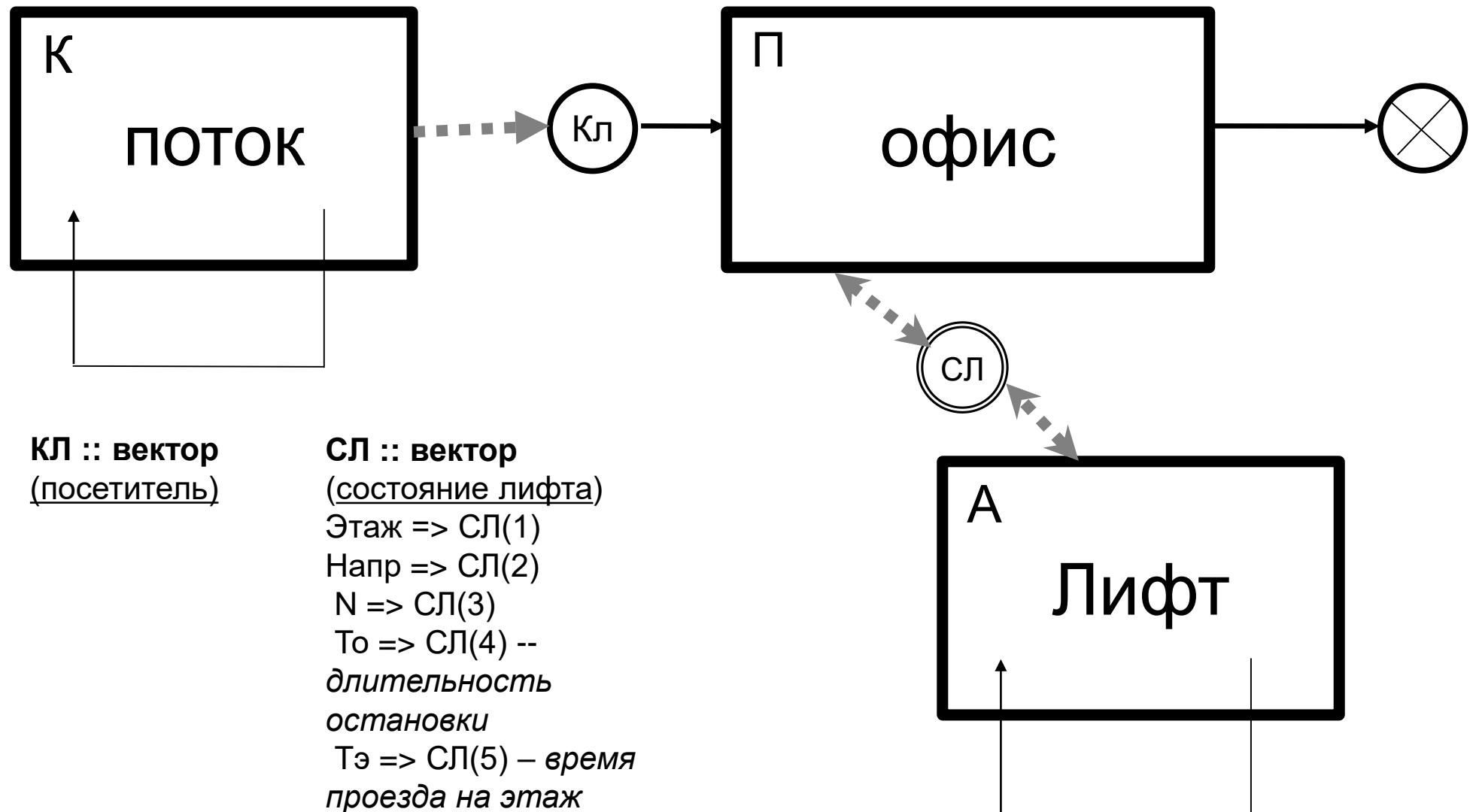
Время перемещения лифта на один этаж равно T_e .

Время посадки и высадки одного пассажира равно T_v .

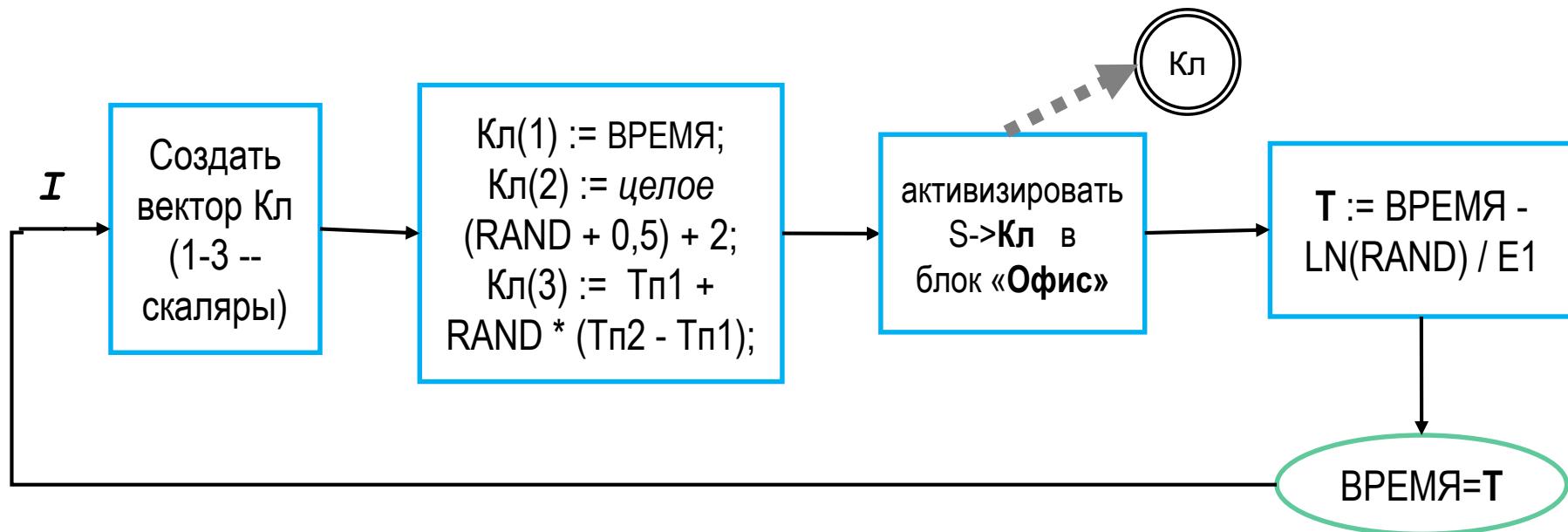
Поток пассажиров имеет показательно-распределенный характер с интенсивностью E_1 .

Время посещения T_p распределено равномерно в интервале $[T_{p1}..T_{p2}]$.

Схема процессной модели



ОПС алгоритма блока Поток

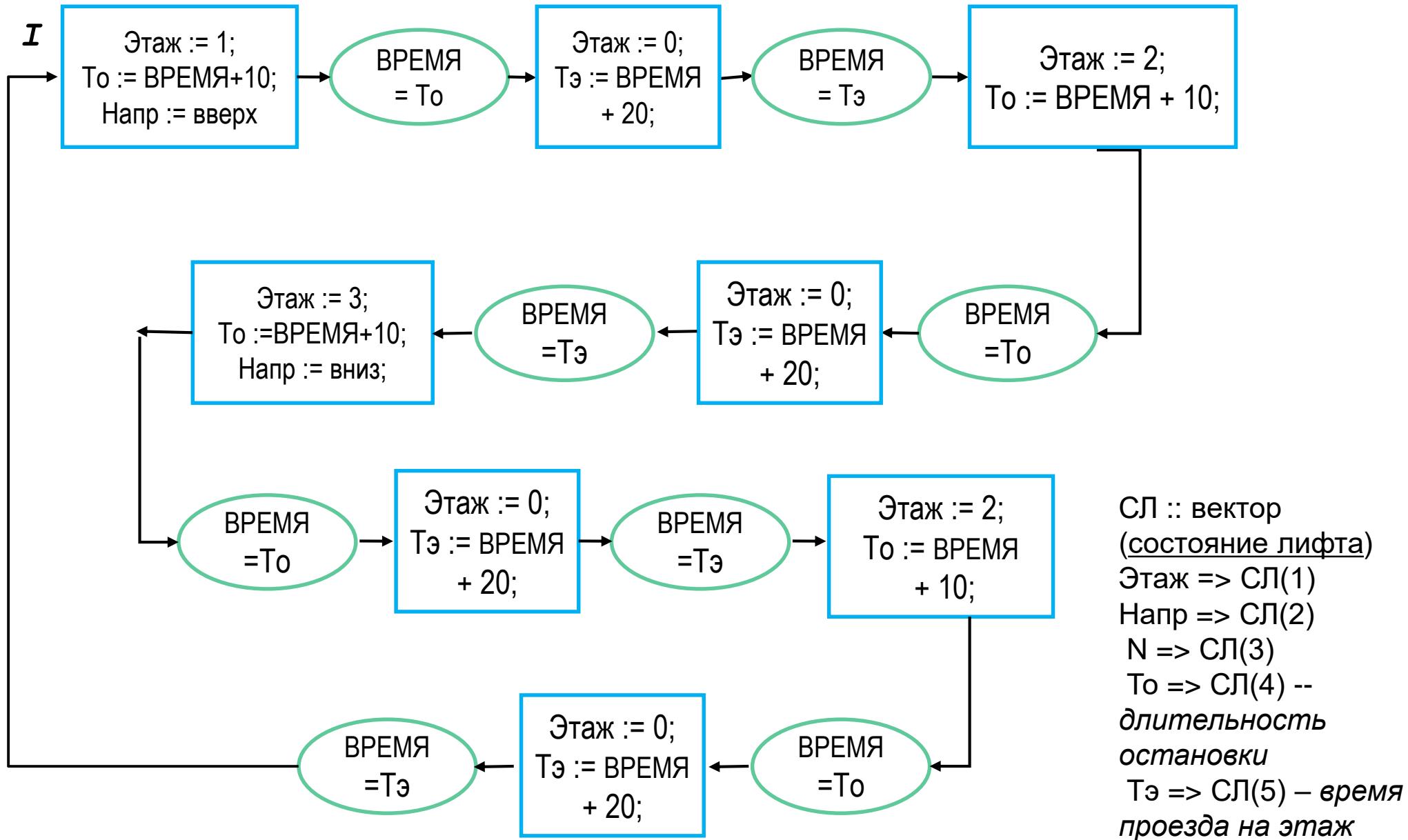


$Кл(1)$ = Время появления

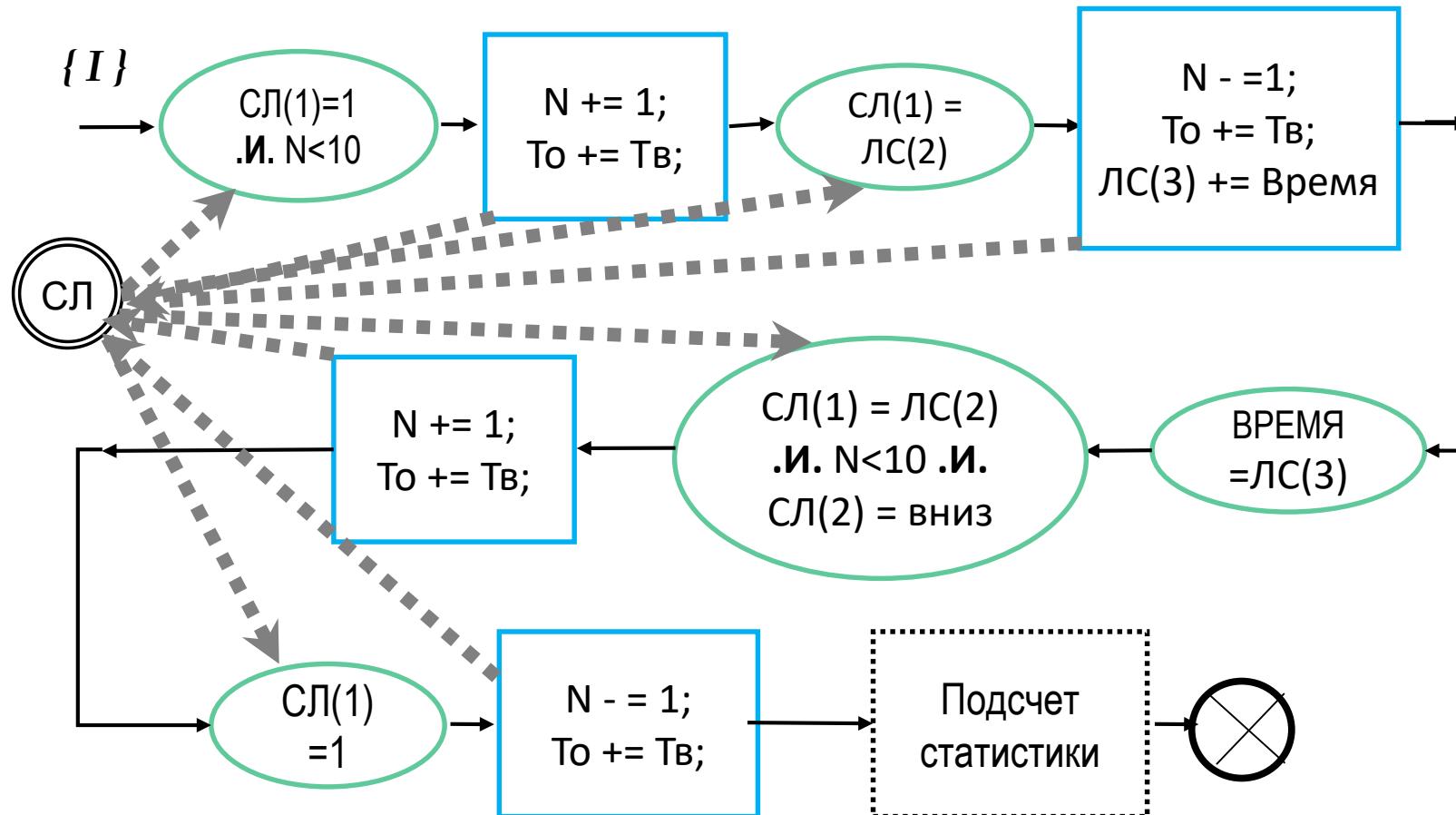
$Кл(2)$ = этаж посещения

$Кл(3)$ = длительность посещения

ОПС алгоритма агрегата Лифт



ОПС алгоритма блока Офис



СЛ :: вектор
состояние лифта
 СЛ(1) => Этаж
 СЛ(2) => Напр
 СЛ(3) => N

ЛС :: Инициатор-->вектор
 ЛС(1) => Время появления
 ЛС(2) => Этаж
 ЛС(3) => ΔT посещения

блок-контроллер ПОТОК

описание

$T_{_}$ -- скаляр; ОФИС -- внешний блок;

НАЧ -- метка; //начальное положение инициатора на метке НАЧ

все описание;

алгоритм

НАЧ: создать Кл типа вектор (1-3 -скаляры); //генерация локальной среды

Кл(1) := ВРЕМЯ;

Кл(2) := ЦЕЛОЕ (RAND+0.5)+2; // 2 или 3

Кл(3) := Тп1 + RAND * (Тп2 - Тп1);

создать S типа ссылка; S := ссылка на Кл; //подготовка инициатора
активизировать инициатор из S в блок ОФИС на метку ВХОД;

$T_{_} :=$ ВРЕМЯ - LN(RAND) / E1; //вычисление времени след.события

ждать ВРЕМЯ = $T_{_}$;

направить ИНИЦИАТОР на НАЧ;

все алгоритм;

все блок.

блок-агрегат ЛИФТ

описание

То, Напр, Этаж, Тэ -- скаляры; N – скаляр // нач.знач. 0

СТАРТ -- метка; // начальное положение инициатора на метке СТАРТ

все описание;

алгоритм

СТАРТ: Этаж := 1;

То := ВРЕМЯ + 10;

Напр := вверх;

ждать ВРЕМЯ = То;

Этаж := 0;

Тэ := ВРЕМЯ + 20;

ждать ВРЕМЯ = Тэ;

Этаж := 2;

То := ВРЕМЯ + 10;

ждать ВРЕМЯ = То;

Этаж := 0;

Тэ := ВРЕМЯ + 20;

ждать ВРЕМЯ = Тэ;

Этаж := 3;

То := ВРЕМЯ + 10;

Напр := вниз;

ждать ВРЕМЯ = То;

Этаж := 0;

Тэ := ВРЕМЯ + 20;

ждать ВРЕМЯ = Тэ;

Этаж := 2;

То := ВРЕМЯ + 10;

ждать ВРЕМЯ = То;

Этаж := 0;

Тэ := ВРЕМЯ + 20;

ждать ВРЕМЯ = Тэ;

направить ИНИЦИАТОР на СТАРТ;

все алгоритм;

все блок.

блок-процессор ОФИС

описание

То, Напр, Этаж, N – скаляры блока ЛИФТ;

ВХОД -- метка; Тв – скаляр *// = 3 сек*

все описание;

алгоритм

ВХОД: **ждать** Этаж =1 .И. N<10;

N := N + 1; To := To + Тв;

ждать Этаж = ЛС(2);

N := N – 1; To := To + Тв;

ЛС(3) := ЛС(3) + ВРЕМЯ;

ждать ВРЕМЯ = ЛС(3);

ждать Этаж =ЛС(2) .И. N<10 .И. Напр =вниз;

N := N + 1; To := To + Тв;

ждать Этаж =1;

N := N – 1; To := To + Тв;

уничтожить ИНИЦИАТОР;

все алгоритм;

все блок.

*// Макрорасширение
// Обращение к локальной
// среде процесса*

ЛС (&Ак)

*// к - целое число, равное
// номеру элемента в векторе*

алгоритм:

(ИНИЦИАТОР → вектор (&Ак));

все алгоритм;

INITIAL X\$Tet,20 ;*время между этажами*

LIFT STORAGE 10 ;*ёмкость лифта*

GENERATE ,,,1,1 ;*один лифт*

Strt SAVEVALUE KYDA, "BBEPX"

SAVEVALUE Level,1

SAVEVALUE T0,5

ADVANCE 1 ;*время останова расём?*

ADVANCE (X\$T0-1)

SAVEVALUE Level,0

ADVANCE X\$Tet ;*проезд на этаж*

SAVEVALUE Level,2

SAVEVALUE T0,5

ADVANCE 1 ;*время останова расём?*

ADVANCE (X\$T0-1)

SAVEVALUE Level,0

ADVANCE X\$Tet

SAVEVALUE Level,3

SAVEVALUE T0,5

SAVEVALUE KYDA, "BHu3"

ADVANCE 1

ADVANCE (X\$T0-1)

SAVEVALUE Level,0

ADVANCE X\$Tet

SAVEVALUE Level,2

SAVEVALUE T0,5

ADVANCE 1

ADVANCE (X\$T0-1)

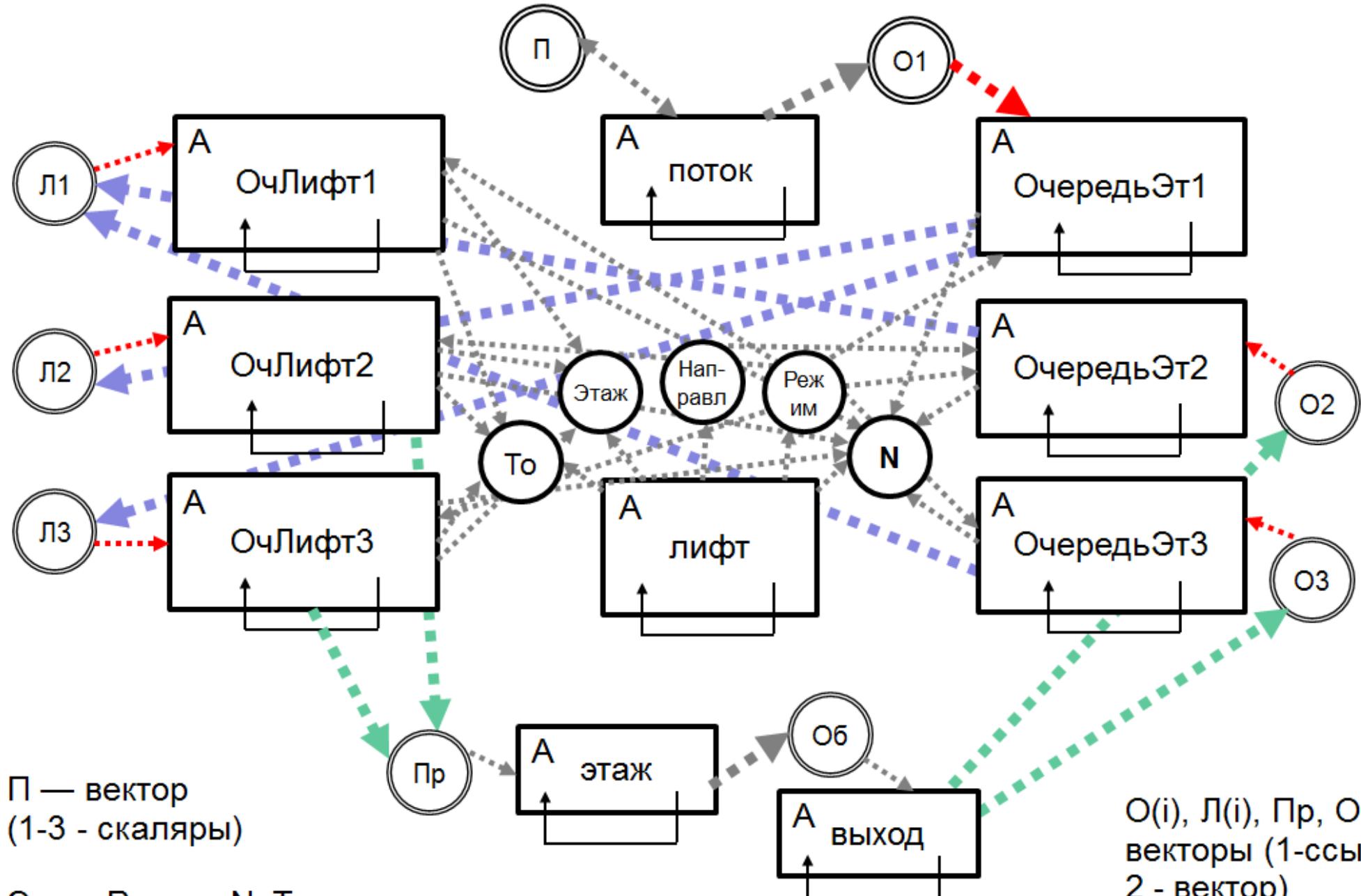
SAVEVALUE Level,0

ADVANCE X\$Tet

TRANSFER ,strt

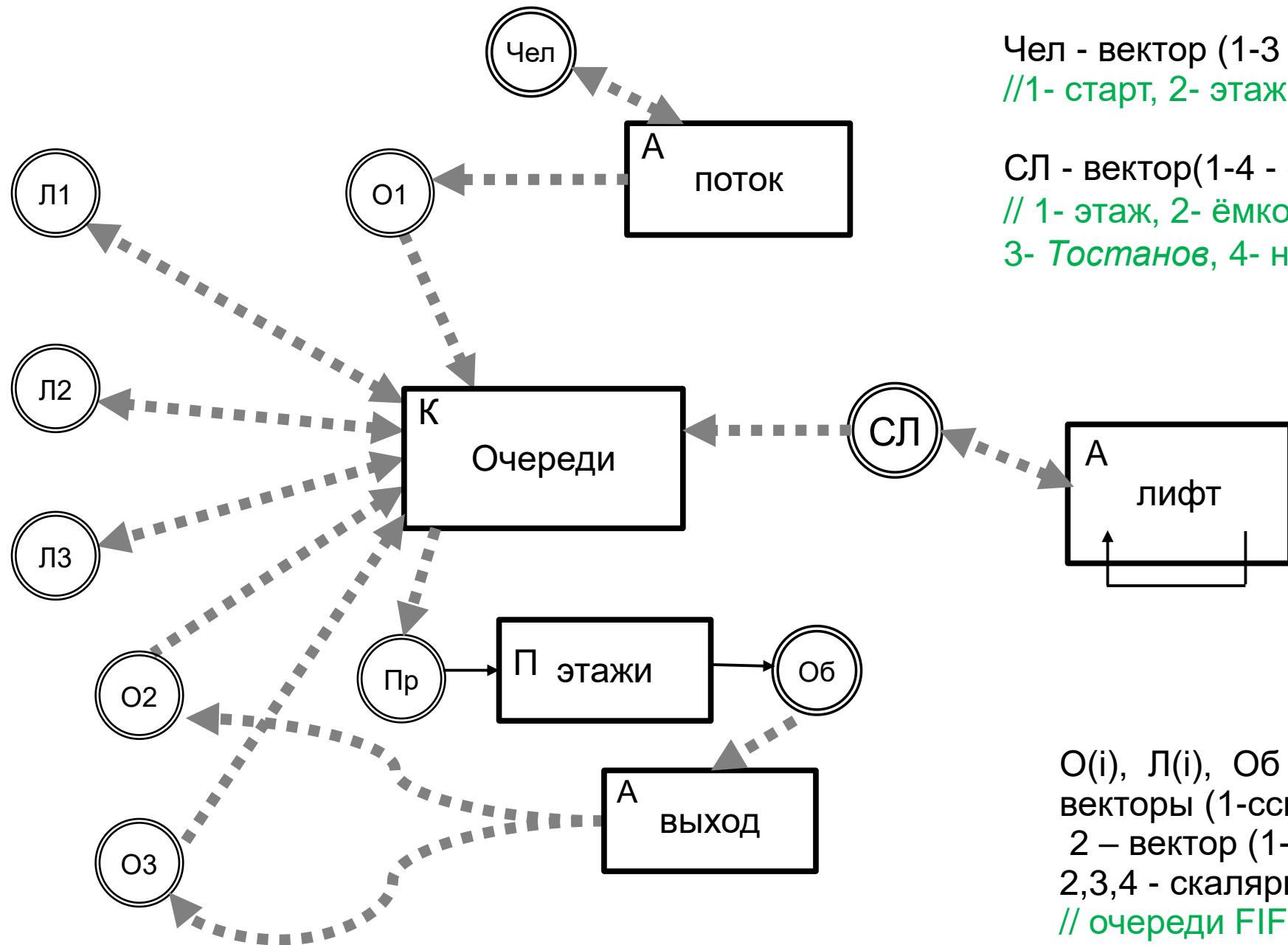
```
GENERATE (Exponential(1,0,45))
ASSIGN TimeOffice,(Uniform(1,100,1000)) ; время пребывания в офисе
ASSIGN ETAJ,(Duniform(1,2,3)) ; номер этажа клиента
QUEUE 1 ; очередь на 1м этаже
TEST E ((X$Level=1)&(S$LIFT<10)),1 ; ждём лифт и место в лифте
ENTER LIFT ; вход в лифт
SAVEVALUE T0+,3 ; время остановки лифтарастёт
DEPART 1
TEST E X$Level,P$ETAJ ; ждём свой этаж
LEAVE LIFT ; выход из лифта
SAVEVALUE T0+,3 ; время остановки лифтарастёт
ADVANCE P$TimeOffice ; время пребывания в офисе
QUEUE P$ETAJ ; очереди на своем этаже
TEST E ((X$Level=P$ETAJ)&(S$LIFT<10)&(X$KYDA="ВНиЗ")),1
ENTER LIFT ; вход в лифт
SAVEVALUE T0+,3 ; время остановки лифтарастёт
DEPART P$ETAJ
TEST E X$Level,1 ; ждём 1й этаж
LEAVE LIFT ; выход из лифта
SAVEVALUE T0+,3 ; время остановки лифтарастёт
TERMINATE
```

Схема потоковой модели



Список **Об** упорядочен по **Т** посещения

Схема гибридной модели



Чел - вектор (1-3 - скаляры)
//1- старт, 2- этаж, 3- Тофис

СЛ - вектор(1-4 - скаляры)
// 1- этаж, 2- ёмкость,
3- Тостанов, 4- направлен

О(i), Л(i), Об -
векторы (1-ссылка,
2 – вектор (1-ссылка,
2,3,4 - скаляры))
// очереди FIFO //

фрагмент модели лифта

```
def run(self):  
    global eventslift, riders  
  
    stage = self.num_generator() # генератор - счетчик этажа  
    yield env.timeout(50) # подождем первых ездоков  
  
    while True:  
        level = next(stage) # Использование генератора этажа  
        self.move = False # лифт не едет  
        self.curr_floor = level # лифт на этаже  
        if (self.curr_floor in [1,MAX_LEVEL]):  
            self.going_up = (self.curr_floor==1) # лифт едет вверх?  
            print(self.env.now, ":@",self.curr_floor, "/",self.liftcab.count) # лифт приехал  
            eventslift[self.curr_floor].succeed() # создаем событие - на этаже!  
            yield env.timeout(self.Lift1Stop) # время остановки на этаже  
            eventslift[self.curr_floor] = self.env.event() # пересоздаем событие  
            print(self.env.now, ":$ ",self.curr_floor, "/",self.liftcab.count) # лифт уехал  
            self.move = True # лифт едет  
            self.curr_floor = 0 # лифт не на этаже  
            yield env.timeout(self.Lift1Time) # время переезда на след.этаж
```

фрагмент модели пассажира лифта

```
def run(self):  
    global eventslift, passengers  
    while True:  
        if not self.waiting:  
            yield eventslift[self.curr_floor] # ждем лифт на своем этаже  
            if (self.lift.going_up == (self.target_floor > self.curr_floor)) and \  
                self.lift.liftcab.count < self.lift.maxx : # если лифт туда и есть место  
                enterin = self.lift.liftcab.request() # занимаем ресурс  
                yield enterin  
                self.inside_ = True # я в лифте!  
                yield self.env.timeout(1)  
                yield eventslift[self.target_floor] # едем в лифте на целевой этаж  
                self.lift.liftcab.release(enterin) # освобождаем ресурс  
                self.inside_ = False # я вышел из лифта  
                self.curr_floor = self.lift.curr_floor  
                self.waiting = True # я не жду лифт – буду гулять  
                self.target_floor = 1 # собираюсь ехать на 1й этаж  
                yield self.env.timeout(10)  
                if self.curr_floor==1: # я вернулся на 1й этаж?  
                    return # удаляюсь из пассажиров  
            else:  
                yield self.env.timeout(1)  
        else:  
            yield self.env.timeout(self.wait) # гуляю на этаже  
            self.waiting = False # собрался ехать на лифте
```

Модель перекрёстка / сеть КА /

Модель перекрестка двух автодорог, регулируемых светофором, работающим в режиме «красный» - «зеленый». По каждой из дорог транспорт движется в одном направлении по одной полосе.

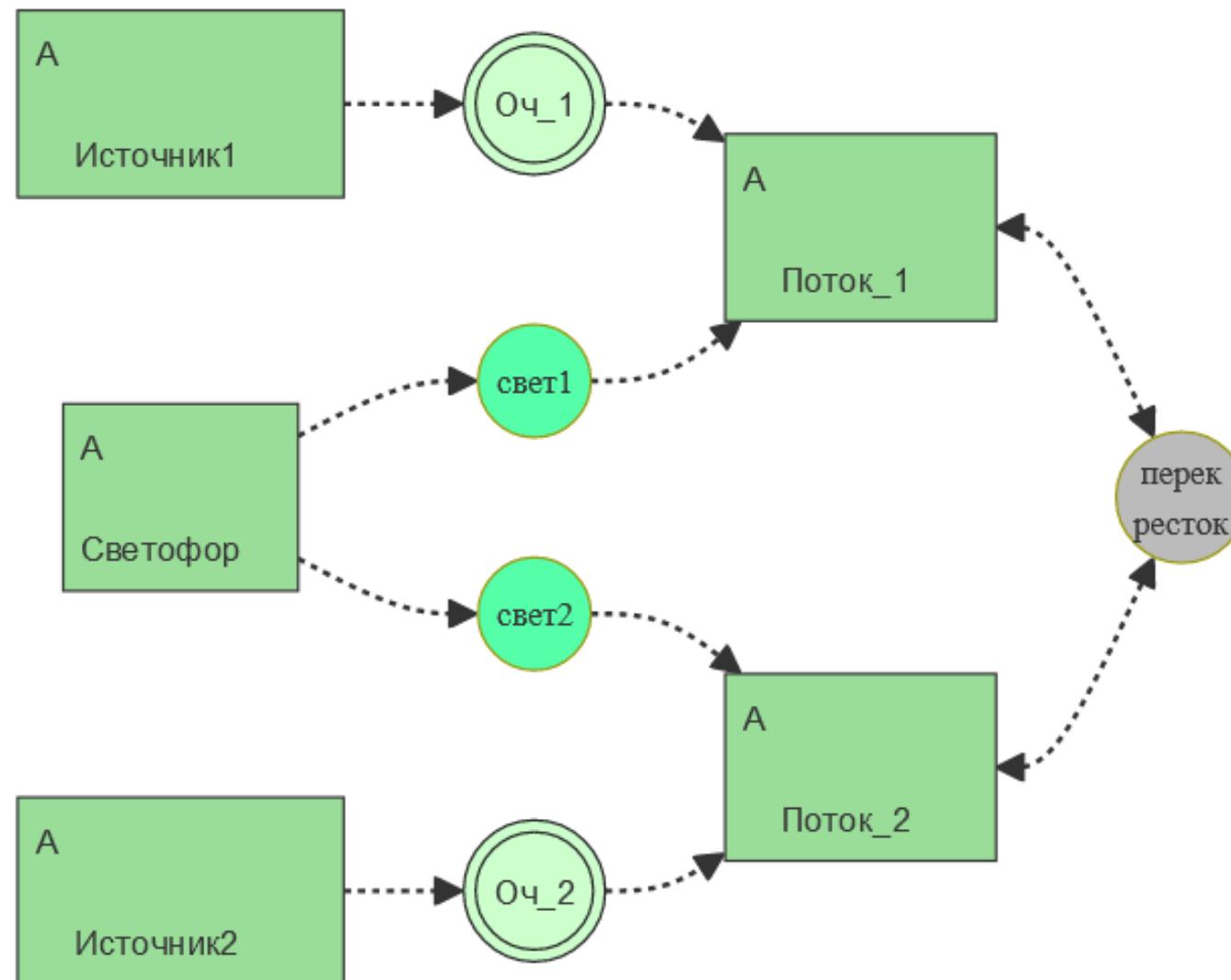


Схема гибридной модели перекрёстка

