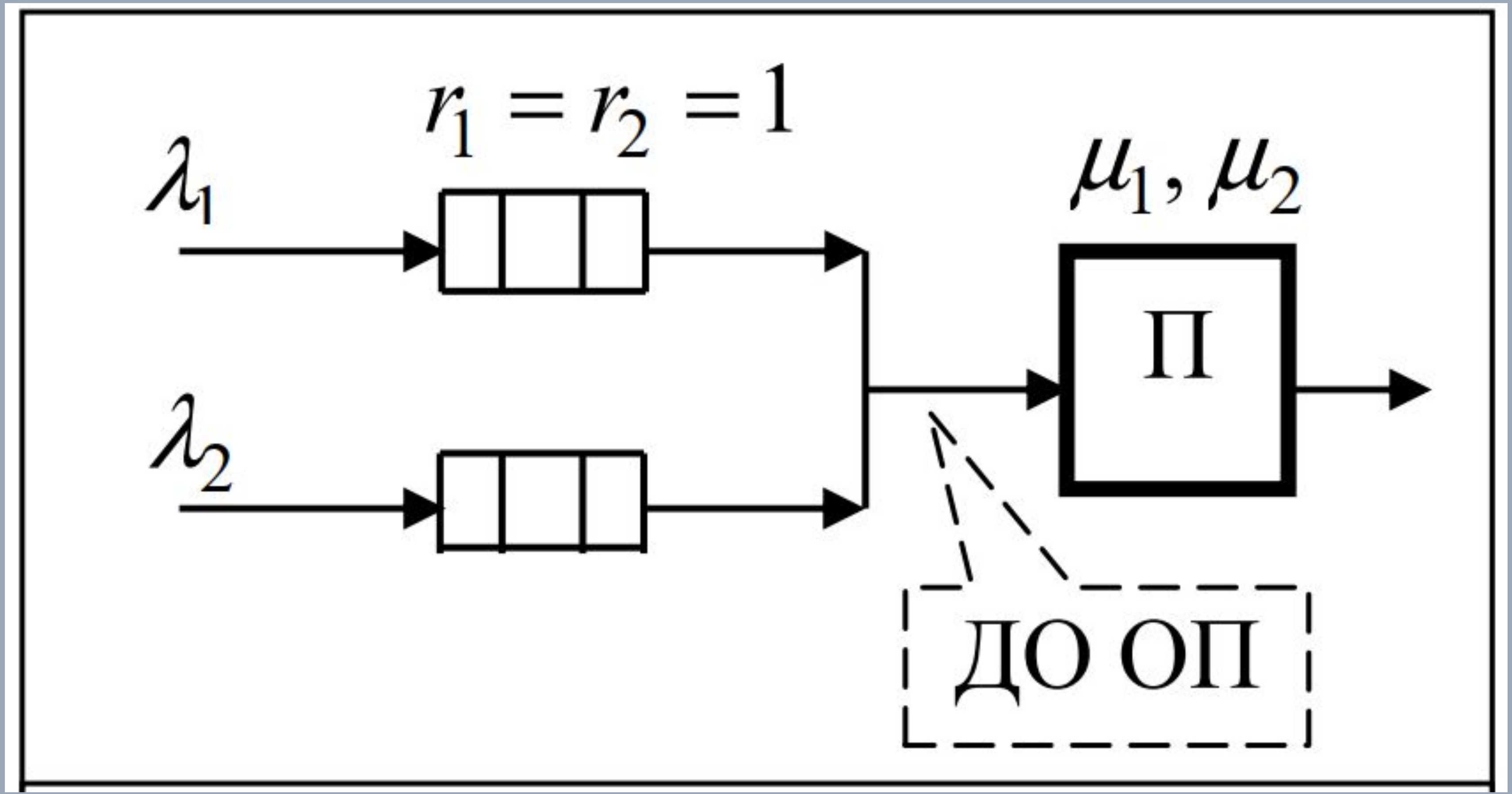
non-uniform flow requests and relative priorities



SINGLE-CHANNEL QS WITH NON-UNIFORM FLOW REQUESTS AND RELATIVE PRIORITIES

. Description of the system

11.1. The system is single-channel.

1.2. Incoming stream of applications - heterogeneous: the system enters two classes of applications.

1.3. Accumulators for orders each class - limited capacity: $r1 = r2 = 1$.

1.4. Buffering discipline - without displacing orders: if at receipt of any application in the system

class the corresponding drive is full, then the application is lost

1.4. Service discipline - with relative priorities: first class applications have priority over second class applications class.

Assumptions

2. 1. Entries of two classes entering the system form the simplest flows with intensities $\lambda 1$ and $\lambda 2$, respectively.

2.2. The service times for customers of each class are distributed exponentially with intensities

$\mu 1 = 1 / b1$ and $\mu 2 = 1 / b2$, where b1 and b2 are the average durations of servicing requests of the class 1 and 2 respectively.

A stationary regime always exists in the QS, since it cannot

be endless queues.

3. Coding the states of a random process

E0: (0 / 0,0) - there are no orders in the system;

E1: (1 / 0.0) - a class 1 request is being serviced in the device;

E2: (2 / 0,0) - a class 2 request is being serviced in the device; E4: (1 / 0.1) - there is a class 1 request and one a class 2 claim is awaiting service in the second drive;

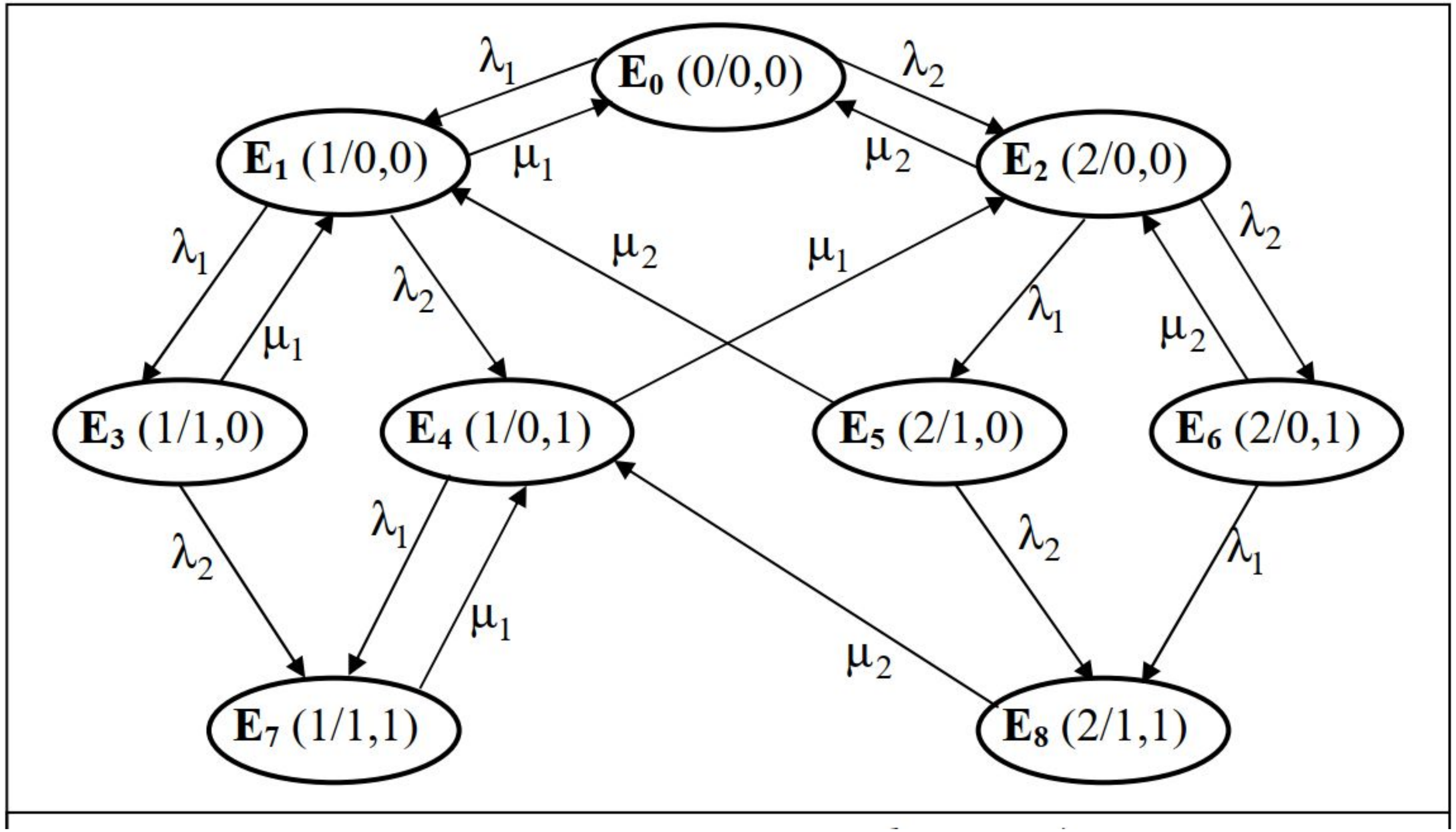
E5: (2 / 1.0) - there is a class 2 request and one a class 1 claim is awaiting service in the first drive;

E6: (2 / 0,1) - there is a class 2 request and one a class 2 claim is awaiting service in the second drive;

E7: (1 / 1,1) - a class 1 request is being serviced, and one service of each class is expected in the corresponding drives;

E8: (2 / 1,1) - a class 2 request is being serviced, and one service of each class is expected to be served in the corresponding drives.

4. Labeled transition graph of a random process



system of balance equations

$$\begin{cases} (\lambda_1+\lambda_2)\,p_0=\mu_1 p_1+\mu_2 p_2 \\ (\lambda_1+\lambda_2+\mu_1)\,p_1=\lambda_1 p_0+\mu_1 p_3+\mu_2 p_5 \\ (\lambda_1+\lambda_2+\mu_2)\,p_2=\lambda_2 p_0+\mu_1 p_4+\mu_2 p_6 \\ (\lambda_2+\mu_1)\,p_3=\lambda_1 p_1 \\ (\lambda_1+\mu_1)\,p_4=\lambda_2 p_1+\mu_1 p_7+\mu_2 p_8 \\ (\lambda_2+\mu_2)\,p_5=\lambda_1 p_2 \\ (\lambda_1+\mu_2)\,p_6=\lambda_2 p_2 \\ \mu_1 p_7=\lambda_2 p_8+\lambda_1 p_4 \\ \mu_2\,p_8=\lambda_2 p_5+\lambda_1 p_6 \\ p_0+p_1+p_2+p_3+p_4+p_5+p_6+p_7+p_8=1 \end{cases}$$

5. Calculation of the characteristics of the system

1) нагрузка: $y_1 = \lambda_1 / \mu_1 = \lambda_1 b_1; \quad y_2 = \lambda_2 / \mu_2 = \lambda_2 b_2$;

2) загрузка, создаваемая заявками, которая может трактоваться как вероятность того, что на обслуживании в приборе находится заявка класса 1 и 2 соответственно: $\rho_1 = p_1 + p_3 + p_4 + p_7; \quad \rho_2 = p_2 + p_5 + p_6 + \rho_8$;

3) среднее число заявок в очереди:

$l_1 = p_3 + p_5 + p_7 + p_8; \quad l_2 = p_4 + p_6 + p_7 + p_8$;

4) среднее число заявок в системе:

$m_1 = p_1 + 2p_3 + p_4 + p_5 + 2p_7 + p_8 = l_1 + \rho_1$;

$m_2 = p_2 + p_4 + p_5 + 2p_6 + p_7 + 2p_8 = l_2 + \rho_2$;

5) вероятность потери заявок:

$\pi_1 = p_3 + p_5 + p_7 + p_8; \quad \pi_2 = p_4 + p_6 + p_7 + p_8$;

6) производительность по каждому классу заявок (интенсивность непотерянных заявок):

$\lambda_1' = \lambda_1(1-\pi_1); \quad \lambda_2' = \lambda_2(1-\pi_2)$;

7) среднее время ожидания заявок:

$w_1 = l_1 / \lambda_1'; \quad w_2 = l_2 / \lambda_2'$

8) среднее время пребывания заявок

$u_1 = m_1 / \lambda_1' = w_1 + b; \quad u_2 = m_2 / \lambda_2' = w_2 + b$

Calculation of total flow characteristics service requests

1) total system load: $Y = y1 + y2$;

2) system load: $R = \rho 1 + \rho 2$;

3) system downtime ratio: $\eta = 1 - R$;

4) the total number of requests in all queues: $l = l1 + l2$;

5) the total number of requests in the system: $m = m1 + m2 = l + R$

6) the probability of losing orders: $\pi = \pi 1 + \pi 2$;

7) system performance (intensity of the total flow of serviced requests):) $\lambda ' = \lambda 1' + \lambda '2 = \lambda (1 – \pi)$;

8) the average waiting time for requests of the total flow: $w = (\lambda w + \lambda w /) \lambda = l / \lambda$;

9) the average sojourn time of claims of the total flow: $u = u + u ' = m' = w + b$