

حَسْنَةٌ لِرَحْمَةِ رَبِّنَا



ارزیابی کارایی سیستم‌های کامپیووتری

پروژه: شبیه‌سازی شبکه صف باز

آرمان آزادی

۱۴۰۴ زمستان

## فهرست مطالب

۴ .....	مقدمه	.۱.
۶ .....	شرح مسئله	.۲.
۱۰ .....	مدل‌سازی تحلیلی	.۳.
۱۶ .....	طراحی شبیه‌سازی رخداد گستته	.۴.
۲۰ .....	پیاده‌سازی رابط کاربری گرافیکی (GUI)	.۵.
۲۶ .....	اعتبارسنجی و مقایسه نتایج	.۶.
	۷. نتیجه‌گیری	۲۹

## ۱. مقدمه

### ۱.۱ پیش‌زمینه ارزیابی کارایی

ارزیابی کارایی (Performance Evaluation) یکی از مراحل بنیادی در مهندسی سیستم‌های کامپیوتری است. پیش از آنکه یک سیستم - مانند یک شبکه کامپیوتری، مزرعه سرورهای ابری، یا خط مونتاژ تولید - به صورت فیزیکی ساخته شود، مهندسان باید رفتار آن را تحت بارهای کاری مختلف پیش‌بینی کنند. اگر سیستمی بدون ارزیابی مناسب مستقر شود، ممکن است دچار گلوگاه‌ها، تاخیر بالا یا اشباع منابع شود که منجر به شکست‌های پرهزینه خواهد شد.

سه تکنیک اصلی برای ارزیابی کارایی وجود دارد:

۱. اندازه‌گیری (Measurement): پایش سیستم واقعی تحت بار. این روش دقیق‌ترین است اما معمولاً در فاز طراحی غیرممکن است.

۲. مدل‌سازی تحلیلی (Analytical Modeling): استفاده از فرمول‌های ریاضی (نظریه صفر، زنجیره‌های مارکوف) برای استخراج دقیق معیارهای کارایی. این روش سریع است اما بر فرضیات ساده‌ساز (مانند توزیع‌های نمایی) تکیه دارد.

۳. شبیه‌سازی (Simulation): ساخت یک مدل نرم‌افزاری که رفتار سیستم را در طول زمان تقلید می‌کند. این روش تعاملات پیچیده‌ای را که ریاضیات به سختی حل می‌کند، پوشش می‌دهد اما از نظر محاسباتی پرهزینه است.

این پروژه از هر دو روش مدل‌سازی تحلیلی و شبیه‌سازی برای مطالعه یک شبکه صفر پیچیده و باز بهره می‌برد.

### ۱.۲ مروجی بر شبیه‌سازی رخداد گسسته (DES)

شبیه‌سازی رخداد گسسته (Discrete Event Simulation) روش انتخاب شده برای این پروژه است. برخلاف شبیه‌سازی پیوسته (که از معادلات دیفرانسیل استفاده می‌کند)، DES سیستم را به عنوان دنباله‌ای از رخدادهای گسسته در زمان مدل می‌کند. وضعیت سیستم بین رخدادها بدون تغییر باقی می‌ماند.

مفاهیم کلیدی در DES عبارتند از:

- **وضعیت سیستم (System State):** مجموعه‌ای از متغیرها که سیستم را در یک لحظه زمانی توصیف می‌کنند (مثلًاً تعداد کارها در یک صفحه).
- **ساعت شبیه‌سازی (Simulation Clock):** متغیری که زمان فعلی را نشان می‌دهد. در DES، ساعت به زمان رخداد بعدی "پرش" می‌کند و دوره‌های عدم فعالیت را نادیده می‌گیرد.
- **لیست رخدادها (FEL):** یک صف اولویت‌دار که رخدادهای آینده را بر اساس زمان مرتب کرده است.
- **انباستگرهای آماری (Statistical Accumulators):** متغیرهایی که برای ردیابی داده‌ها (مانند مجموع زمان انتظار) جهت محاسبه میانگین‌ها در پایان کار استفاده می‌شوند.

### ۱.۳. اهداف پروژه

اهداف مشخص این پروژه عبارتند از :

۱. ساخت مدل: تعریف یک شبکه صف باز با چهار گره، مسیریابی احتمالی و حلقه‌های بازخورد.
۲. حل تئوری: حل دستی سیستم با استفاده از معادلات شبکه جکسون برای ایجاد یک مبنای "حقیقت پایه" (Ground Truth).
۳. پیاده‌سازی نرم‌افزاری: توسعه یک شبیه‌ساز مبتنی بر پایتون با رابط کاربری گرافیکی (GUI) برای مشاهده جریان مشتریان.
۴. تایید و اعتبارسنجی: اجرای شبیه‌سازی برای تعداد زیادی مشتری و مقایسه نتایج با مقادیر تئوری جهت تایید صحت کرد.

## ۲. شرح مسئله

### ۲.۱ توپولوژی و معماری سیستم

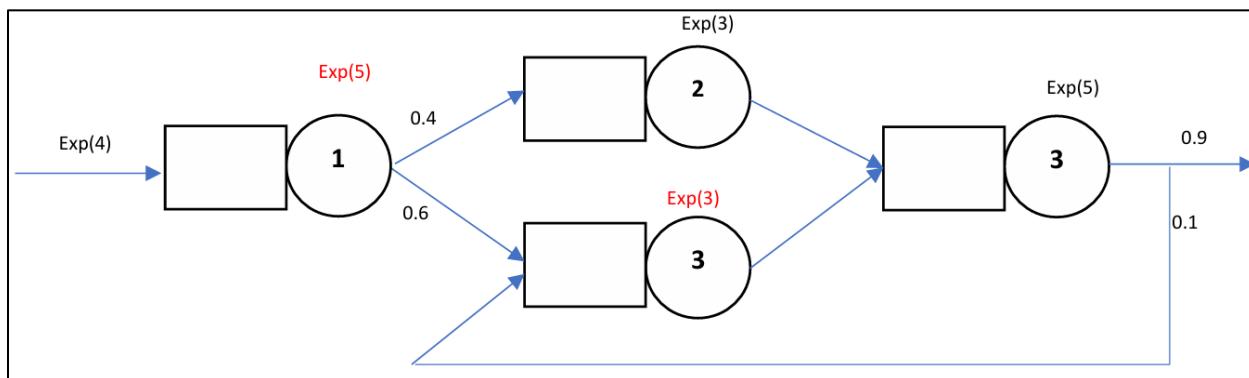
سیستم مورد مطالعه یک شبکه صف باز (Open Queueing Network) متشکل از چهار ایستگاه سرویس دهنده (گره) متمایز است (شکل ۱، شکل ۲). این سیستم "باز" نامیده می‌شود زیرا مشتریان از یک منبع خارجی وارد شده و در نهایت از سیستم خارج می‌شوند.

توپولوژی شبکه شامل موارد زیر می‌باشد:

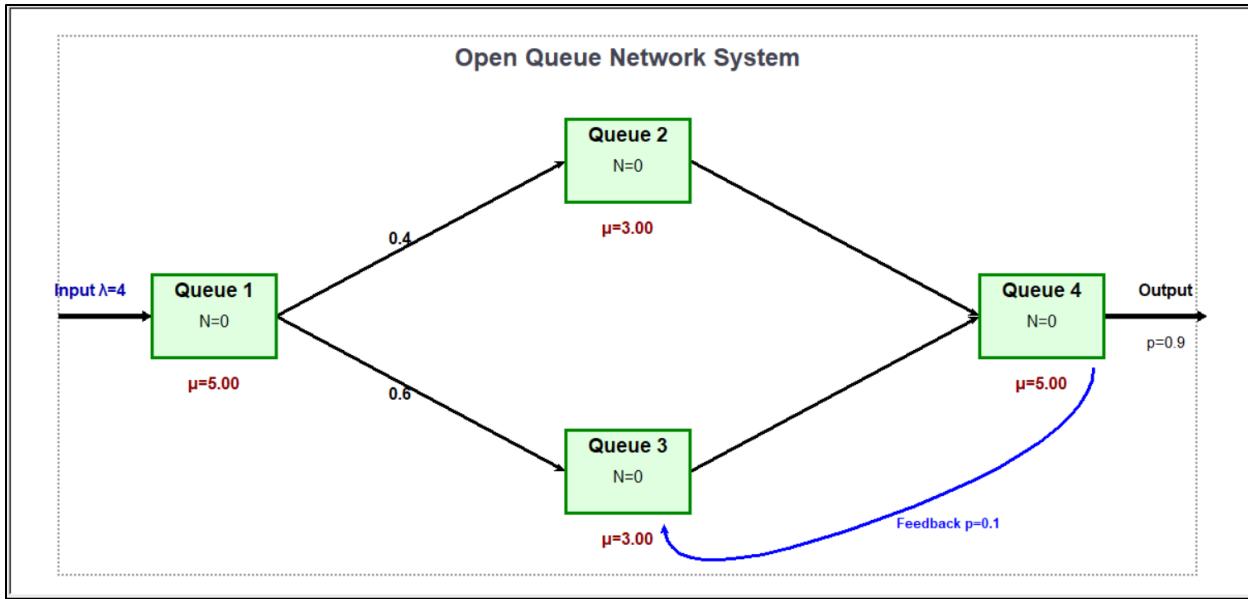
مسیرهای انشعابی: مشتریانی که از صف ۱ خارج می‌شوند می‌توانند به صف ۲ یا صف ۳ بروند.

نقاط ادغام: ترافیک خروجی از صف ۲ و صف ۳ در صف ۴ ادغام می‌شود.

حلقه بازخورد (Feedback Loop): بخشی از مشتریان که از صف ۴ خارج می‌شوند، مجدداً به صف ۳ بازگردانده می‌شوند؛ این چرخه باعث افزایش بار بر روی گره‌های پایین‌دست می‌شود.



شکل ۱



شکل ۲: قسمتی از رابط کاربری شبیه‌سازی که توپولوژی سیستم را به صورت زنده و بلادرنگ نشان می‌هد.

## ۲.۲. پارامترهای بار کاری و سرویس

این سیستم تحت فرضیات تصادفی (Stochastic) عمل می‌کند که با مدل مارکوفی سازگار است:

زمان‌های بین ورود: توزیع نمایی (فرآیند ورود پواسون).

زمان‌های سرویس: توزیع نمایی در تمام گره‌ها.

نظم صف‌بندی: سرویس‌دهی به ترتیب ورود (FCFS) در تمام صف‌ها.

ظرفیت: ظرفیت صف نامحدود (بدون مسدودسازی یا از دست رفتن بسته).

پیکربندی پارامترها:

نرخ‌های خاص استفاده شده برای تحلیل حالت پایدار به شرح زیر است (جدول ۱: نرخ‌های استفاده شده در

شبکه صف باز مسئله):

جدول ۱: نرخ‌های استفاده شده در شبکه صیف باز مسئله

توضیحات	مقدار	پارامتر
نرخ ورود از دنیای خارج	۴.۰	Lambda (source)
نرخ سرویس صف ۱	۵.۰	mu_1
نرخ سرویس صف ۲	۳.۰	mu_2
نرخ سرویس صف ۳ (بار زیاد)	۳.۰	mu_3
نرخ سرویس صف ۴	۵.۰	mu_4

### ۲.۳ منطق مسیریابی و احتمالات

جابجایی مشتریان بین صفات احتمالی است. وقتی یک مشتری سرویس خود را در یک گره به پایان می‌رساند، یک انتخاب تصادفی بر اساس ماتریس مسیریابی زیر انجام می‌شود:

از صف ۱:

$$40\% \text{ به صف ۲ می‌روند} \quad P_{1,2} = 0.4 \quad \bullet$$

$$60\% \text{ به صف ۳ می‌روند} \quad P_{1,3} = 0.6 \quad \bullet$$

از صف ۲:

$$100\% \text{ به صف ۴ می‌روند} \quad P_{2,4} = 1.0 \quad \bullet$$

از صف ۳:

$$100\% \text{ به صف ۴ می‌روند} \quad P_{3,4} = 1.0 \quad \bullet$$

از صف ۴:

• ۹۰٪ از سیستم خارج می‌شود)  $P_{4,out} = 0.9$

• ۱۰٪ به صف ۳ می‌روند)  $P_{4,3} = 0.1$

## ۳. مدل‌سازی تحلیلی

برای اعتبارسنجی شبیه‌سازی، ابتدا باید سیستم را به صورت ریاضی حل کنیم. از آنجا که فرآیند ورود پواسون و زمان‌های سرویس نمایی هستند، سیستم به عنوان یک شبکه جکسون شناخته می‌شود. قضیه جکسون به ما اجازه می‌دهد که با محاسبه نرخ ورود مؤثر ( $\lambda_i$ )، هر گره  $i$  را به عنوان یک صف مستقل  $M/M/1$  در نظر بگیریم.

### ۳.۱ مبانی ریاضی

معادله بنیادی برای حل شبکه‌های جکسون، معادله توازن جریان (Flow Balance Equation) است:

$$\lambda_i = \gamma_i + \sum_{j=1}^M \lambda_j P_{ji}$$

که در آن:

- $\lambda_i$ : کل نرخ ورود مؤثر به گره  $i$
- $\gamma_i$ : نرخ ورود خارجی به گره  $i$
- $P_{ji}$ : احتمال انتقال از گره  $j$  به گره  $i$

### ۳.۲ استخراج گام‌به‌گام نرخ‌های ورود مؤثر

ما معادله توازن جریان را برای هر یک از چهار گره اعمال می‌کنیم.

گره ۱: گره ۱ تنها ترافیک خارجی را دریافت می‌کند.

$$\lambda_1 = \lambda_{source} = 4$$

گره ۲: گره ۲ ترافیک را فقط از گره ۱ دریافت می‌کند.

$$\lambda_2 = \lambda_1 \times P_{1,2}$$

$$\lambda_2 = 4 \times 0.4 = 1.6$$

گره ۴: گره ۴ تمام ترافیک گره ۲ و گره ۳ را دریافت می‌کند.

$$\lambda_4 = \lambda_2 + \lambda_3$$

(ما هنوز نمی‌توانیم این معادله را حل کنیم زیرا  $\lambda_3$  مجهول است.)

گره ۳: گره ۳ ترافیک را از گره ۱ و بازخورد را از گره ۴ دریافت می‌کند.

$$\lambda_3 = (\lambda_1 \times P_{1,3}) + (\lambda_4 \times P_{4,3})$$

$$\lambda_3 = (4 \times 0.6) + (\lambda_4 \times 0.1)$$

$$\lambda_3 = 2.4 + 0.1\lambda_4$$

حل دستگاه معادلات: اکنون یک دستگاه دو معادله‌ای برای  $\lambda_3$  و  $\lambda_4$  داریم:

$$\lambda_3 = 2.4 + 0.1\lambda_4$$

$$\lambda_4 = 1.6 + \lambda_3$$

با جایگذاری:

$$\lambda_4 = 1.6 + (2.4 + 0.1\lambda_4)$$

$$\lambda_4 = 4.0 + 0.1\lambda_4$$

$$0.9\lambda_4 = 4.0$$

$$\lambda_4 = \frac{4.0}{0.9} \approx 4.4444$$

اکنون  $\lambda_3$  را می‌یابیم:

$$\lambda_3 = 2.4 + 0.1(4.4444) \approx 2.8444$$

نرخ‌های مؤثر نهایی:

$$\lambda_1 = 4.000$$

$$\lambda_2 = 1.600$$

$$\lambda_3 = 2.844$$

$$\lambda_4 = 4.444$$

### ۳.۳ محاسبه تئوری معیارهای کارایی

با استفاده از فرمول‌های استاندارد  $M/M/1$  برای هر گره  $i$ :

بهره‌وری: •

$$\rho_i = \lambda_i / \mu_i$$

میانگین تعداد در سیستم:

$$L_i = \rho_i / (1 - \rho_i)$$

میانگین زمان در سیستم:

$$W_i = 1 / (\mu_i - \lambda_i)$$

: ۱ صف

$$(\mu = 5)$$

$$\rho_1 = 4/5 = 0.80$$

$$L_1 = 0.8/0.2 = 4.00$$

: ۲ صف

$$(\mu = 3)$$

$$\rho_2 = 1.6/3 = 0.5333$$

$$L_2 = 0.5333/0.4667 = 1.143$$

صف ۳ (گره با بار زیاد):

$$(\mu = 3)$$

$$\rho_3 = 2.8444/3 = 0.9481$$

$$L_3 = 0.9481/(1 - 0.9481) = 18.286$$

نکته: بهره‌وری در اینجا حدود ۹۵٪ است که منجر به اندازه صف میانگین بسیار بزرگی می‌شود.

صف ۴:

$$(\mu = 5)$$

$$\rho_4 = 4.4444/5 = 0.8889$$

$$L_4 = 0.8889/0.1111 = 8.000$$

#### ۳.۴ تجمعی کل سیستم

برای یافتن کارایی شبکه به عنوان یک کل:

- تعداد کل در شبکه (N):

$$N = \sum L_i = 4.0 + 1.143 + 18.286 + 8.0 = 31.429$$

• زمان پاسخ سیستم (R): با استفاده از قانون لیتل (N = λ<sub>source</sub> R):

$$R = N/4 = 31.429/4 = 7.857 \text{ seconds}$$

#### ۳.۵ پیاده‌سازی حل تحلیل با کد

در برنامه نوشته شده با زبان پایتون، علاوه بر شبیه‌سازی، قسمتی برای حل تحلیلی سیستم نیز در نظر گرفته شده است که در همان محیط گرافیکی بتوان نتیجه تحلیلی را با نتیجه شبیه‌سازی مقایسه کرد.

```

416     # 2. Analytical Calculations (Exact)
417     lam = LAMBDA_SOURCE_RATE
418     # Exact Flow Equations
419     L1 = lam
420     L2 = 0.4 * L1
421     L4 = L1 / 0.9
422     L3 = 0.6 * L1 + 0.1 * L4
423
424     eff_lam = {1: L1, 2: L2, 3: L3, 4: L4}
425
426     ana_metrics = {}
427     total_L_ana = 0
428
429     for i in range(1, 5):
430         l = eff_lam[i]
431         m = MU_RATES[i]
432         rho = l / m
433         L_val = rho / (1 - rho)
434         Lq_val = L_val - rho
435         W_val = 1 / (m - l)
436         Wq_val = W_val - (1/m)
437
438         ana_metrics[i] = {'L': L_val, 'Lq': Lq_val, 'W': W_val, 'Wq': Wq_val, 'Rho': rho}
439         total_L_ana += L_val
440
441     sys_R_ana = total_L_ana / lam
442

```

## ۴. طراحی شبیه‌سازی رخداد گستته

### ۴.۱ روش‌شناسی شبیه‌سازی

شبیه‌سازی با استفاده از پایتون ساخته شده است. موتور اصلی از مکانیزم "پیشروی زمان تا رخداد بعدی" (Next-Event Time Advance) استفاده می‌کند. شبیه‌سازی یک متغیر ساعت جهانی را نگهداری می‌کند. در هر گام، موتور رخدادی را که کمترین برچسب زمانی را دارد از لیست رخدادهای آتی (FEL) حذف می‌کند، ساعت را به آن زمان جلو می‌برد و منطق مربوط به آن نوع رخداد را پردازش می‌کند.

```
34     class Event:
35         def __init__(self, time, type, customer=None, queue_id=None):
36             self.time = time
37             self.type = type
38             self.customer = customer
39             self.queue_id = queue_id
40
41         def __lt__(self, other):
42             return self.time < other.time
43
```

### ۴.۲ طراحی الگوریتم

شبیه‌سازی حول دو نوع رخداد اصلی می‌چرخد:

۱. ورود (ARRIVAL): یک مشتری وارد صف می‌شود. اگر سرور بیکار باشد، سرویس بلافاصله آغاز می‌شود (برنامه‌ریزی یک خروج). اگر سرور مشغول باشد، مشتری در صف منتظر می‌ماند.

```
133     def handle_arrival(self, event):
134         q = event.queue_id
135         self.queues[q].append(event.customer)
136         # We count arrivals, but if we reset stats later, this count resets too
137         self.stats[q]['Arrivals'] += 1
138         msg = f"Cust {event.customer.id} -> Q{q}"
139         if not self.server_busy[q]:
140             self.server_busy[q] = True
141             svc_time = random.expovariate(MU_RATES[q])
142             self.schedule(Event(self.clock + svc_time, 'DEPARTURE', event.customer, q))
143
144         return msg
```

۲. خروج (DEPARTURE): یک مشتری سرویس را تمام می‌کند. سیستم مقصد بعدی را بر اساس احتمالات تصادفی محاسبه کرده و یک ورود در صف بعدی (یا خروج از سیستم) را برنامه‌ریزی می‌کند.

```
145     def handle_departure(self, event):
146         q = event.queue_id
147         c = self.queues[q].pop(0)
148         self.stats[q]['Departures'] += 1
149         msg = f"Cust {c.id} left Q{q}"
150
151         next_q = None
152         r = random.random()
153
154         if q == 1:
155             next_q = 2 if r < PROBS['1_to_2'] else 3
156         elif q == 2:
157             next_q = 4
158         elif q == 3:
159             next_q = 4
160         elif q == 4:
161             if r < PROBS['4_exit']:
162                 next_q = 'EXIT'
163             else:
164                 next_q = 3
165
166         if next_q == 'EXIT':
167             # === WARM-UP LOGIC ===
168             if self.is_warming_up:
169                 self.warmup_count += 1
170                 msg += " -> EXITED (Warmup)"
171                 if self.warmup_count >= WARM_UP_CUSTOMERS:
172                     self.reset_stats_post_warmup()
173                     msg += " [WARM-UP DONE - STATS RESET]"
174             else:
```

#### ۴.۳ جزئیات پیاده‌سازی کد

کد پایتون از کلاس SimulationModel برای کپسوله‌سازی وضعیت استفاده می‌کند. در زیر قطعه کدی از ابتدای تعریف این کلاس آورده شده است:

```

44     class SimulationModel:
45         def __init__(self):
46             self.reset()
47
48         def reset(self):
49             self.clock = 0.0
50             self.events = []
51             self.queues = {1: [], 2: [], 3: [], 4: []}
52             self.server_busy = {1: False, 2: False, 3: False, 4: False}
53
54             # Stats
55             self.stats = {i: {
56                 'L_accum': 0, 'Q_accum': 0, 'Busy_accum': 0,
57                 'Arrivals': 0, 'Departures': 0
58             } for i in range(1, 5)}
59
60             self.completed_count = 0
61             self.total_response_time = 0
62             self.last_update = 0.0
63             self.cust_counter = 1
64
65             # Warm-up tracking
66             self.warmup_count = 0
67             self.is_warming_up = True
68             self.steady_start_time = 0.0 # Time when warm-up finishes
69
70             first_arrival = Event(random.expovariate(LAMBDA_SOURCE_RATE), 'ARRIVAL', Customer(1, 0), 1)
71             self.schedule(first_arrival)
72
73         def schedule(self, event):
74             self.events.append(event)
75             self.events.sort()

```

#### ۴.۴ مدیریت دوره گرم‌سازی (Warm-up Period)

از آنجا که شبیه‌سازی با سیستم خالی (مشتری) شروع می‌شود، معیارهای اولیه ( $L$ ,  $W$ ) دچار سوگیری (Bias) منفی هستند. برای به دست آوردن نتایج دقیق حالت پایدار، یک دوره گرم‌سازی پیاده‌سازی شده است.

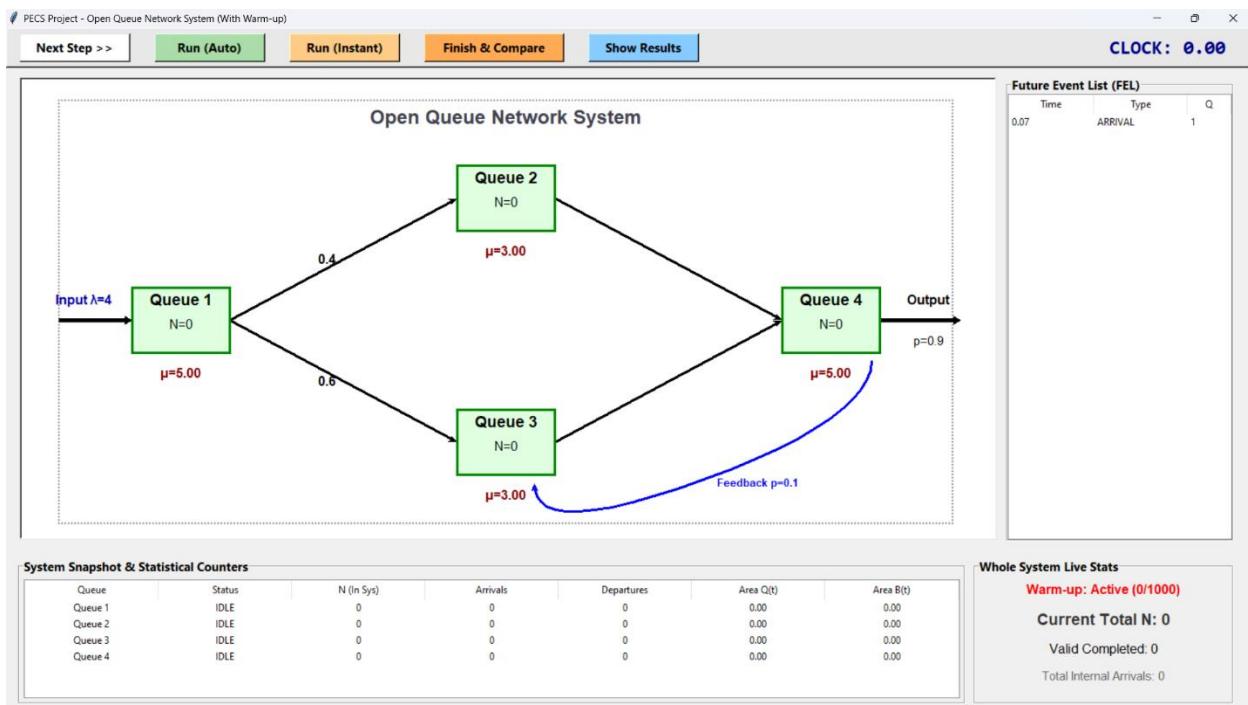
- ۱۰۰۰ مشتری اول پردازش می‌شوند تا صفات پر شوند.
- در طول این فاز، هیچ داده آماری ثبت نمی‌شود.
- به محض خروج هزارمین مشتری، تمام انباشتگرهای آماری صفر می‌شوند و شبیه‌سازی "واقعی" آغاز می‌گردد.
- در کل شبکه برای اینکه به حالت پایداری برسد برای 50,000 مشتری شبیه‌سازی می‌شود (بدون احتساب 1000 مشتری اول).

```
23 MAX_COMPLETED = 50000      # Valid customers to collect AFTER warm-up (Increased for accuracy)
24 WARM_UP_CUSTOMERS = 1000    # Number of customers to discard at start
```

```
167     # === WARM-UP LOGIC ===
168     if self.is_warming_up:
169         self.warmup_count += 1
170         msg += " -> EXITED (Warmup)"
171         if self.warmup_count >= WARM_UP_CUSTOMERS:
172             self.reset_stats_post_warmup()
173             msg += " [WARM-UP DONE - STATS RESET]"
174     else:
175         self.completed_count += 1
176         self.total_response_time += (self.clock - c.arrival_time)
177         msg += " -> EXITED"
```

## ۵. پیاده‌سازی رابط کاربری گرافیکی (GUI)

برای تعاملی و قابل مشاهده کردن شبیه‌سازی، یک رابط کاربری با استفاده از زبان پایتون و کتابخانه tkinter توسعه داده شد(شکل ۳).



شکل ۳

### ۵.۱ طراحی رابط

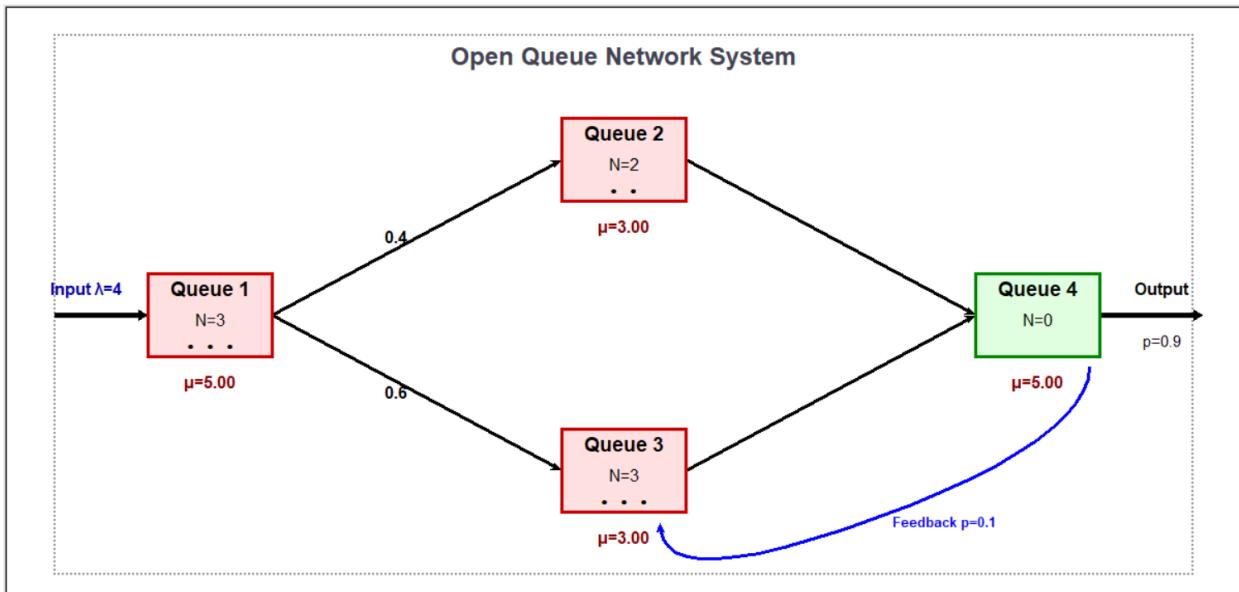
رابط کاربری به چند بخش اصلی تقسیم می‌شود:

پنل کنترل (بالا): شامل دکمه‌هایی برای کنترل جریان شبیه‌سازی ("گام بعدی"، "اجرای خودکار"، "اجرای فوری"، "پایان و مقایسه") (شکل ۴).



شکل ۴

بوم مصورسازی (وسط): نمایش گرافیکی گره‌های شبکه، اتصالات و طول صفحه‌ای فعلی(شکل ۵).



شکل ۵

داشبورد داده‌ها (پایین-چپ): نمایش لیست معیارها و شمارنده‌های آماری بلاذرنگ(شکل ۶).

System Snapshot & Statistical Counters						
Queue	Status	N (In Sys)	Arrivals	Departures	Area Q(t)	Area B(t)
Queue 1	IDLE	0	49961	49967	41275.88	10300.27
Queue 2	IDLE	0	19857	19857	7195.16	6611.65
Queue 3	BUSY	5	35752	35773	170917.26	11915.72
Queue 4	BUSY	3	55630	55642	87390.49	11260.82

شکل ۶

داشبورد داده‌ها (پایین-راست): نمایش اطلاعاتی مربوط به روند شبیه‌سازی؛ اینکه آیا در مرحله گرم‌سازی هستیم یا نه، یا آنکه چند مشتری در حال حاضر در شبکه هستن و چند مشتری تا کنون شبکه را ترک کرده‌اند(شکل ۷، شکل ۸).

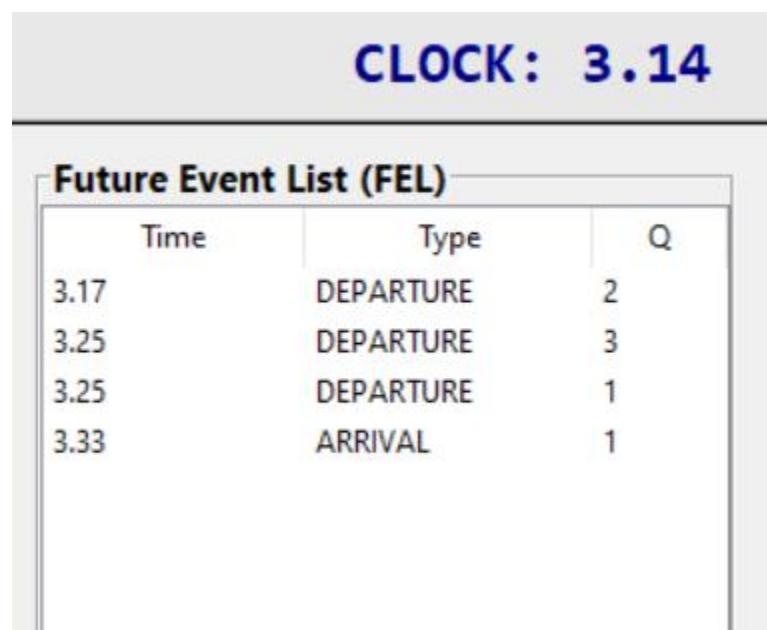


شکل ۷



شکل ۸

داشبورد لیست رخدادها و ساعت شبیه‌سازی: نمایش رخدادهای بعدی و نوع آنها برای هر کدام از صفحه‌های چهارگانه (مرتب شده بر اساس نزدیکی وقوع)؛ همچنین نمایش ساعت فعلی شبیه‌سازی (شکل ۹).



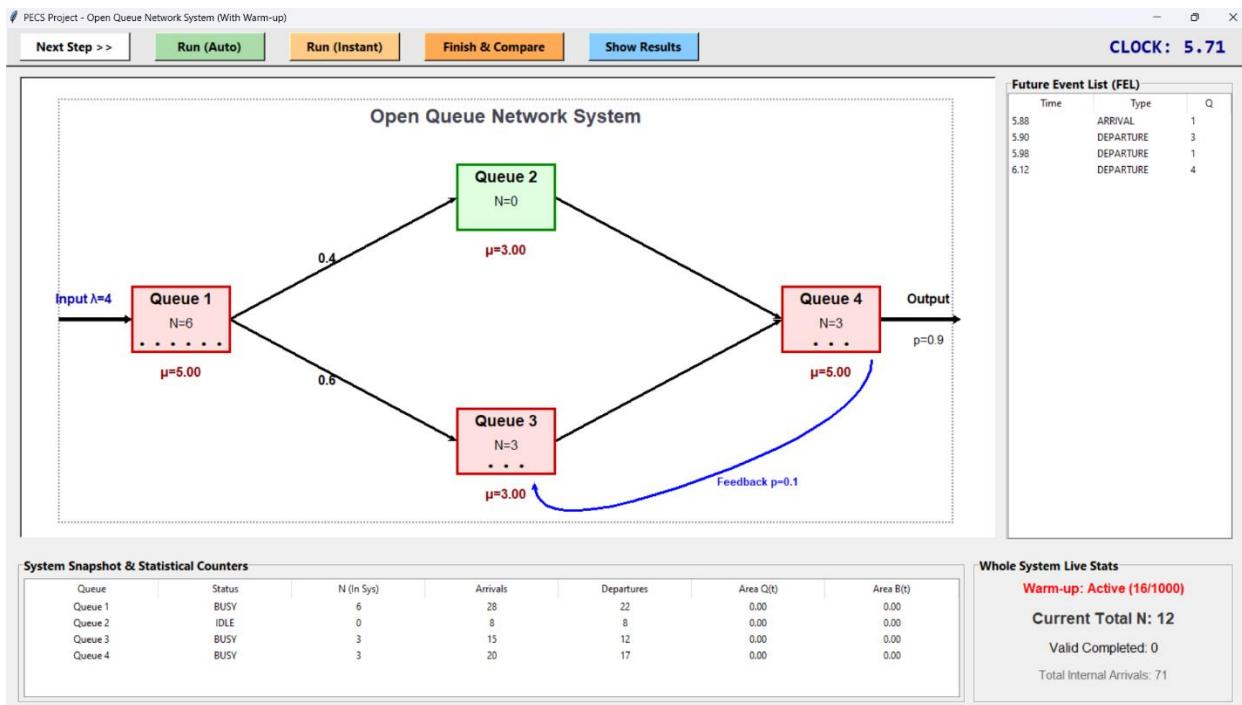
شکل ۹

## ۵.۲ مصورسازی وضعیت شبکه

- بوم نقاشی وضعیت سیستم را به صورت پویا ترسیم می‌کند (شکل ۱۰).
- گره‌ها: به صورت مستطیل نمایش داده می‌شوند. وقتی سرور مشغول است قرمز و وقتی بیکار است سبز می‌شوند.

- صف‌ها: نقاطی داخل گره ترسیم می‌شوند که نشان‌دهنده مشتریان منتظر هستند.

- بازخورد: یک فلش آبی رنگ منحنی به‌وضوح حلقه بازخورد از Q4 به Q3 را نشان می‌دهد.



شكل ۱۰

### ۵.۳ پایش آماری بلاذرنگ

پنل پایینی شفافیت کاملی از وضعیت داخلی شبیه‌سازی فراهم می‌کند. این جدول مقادیر زنده را نمایش می‌دهد:

- $N$ : تعداد مشتریان در سیستم.

- $Q(t)$ : مساحت انباشته شده صف.

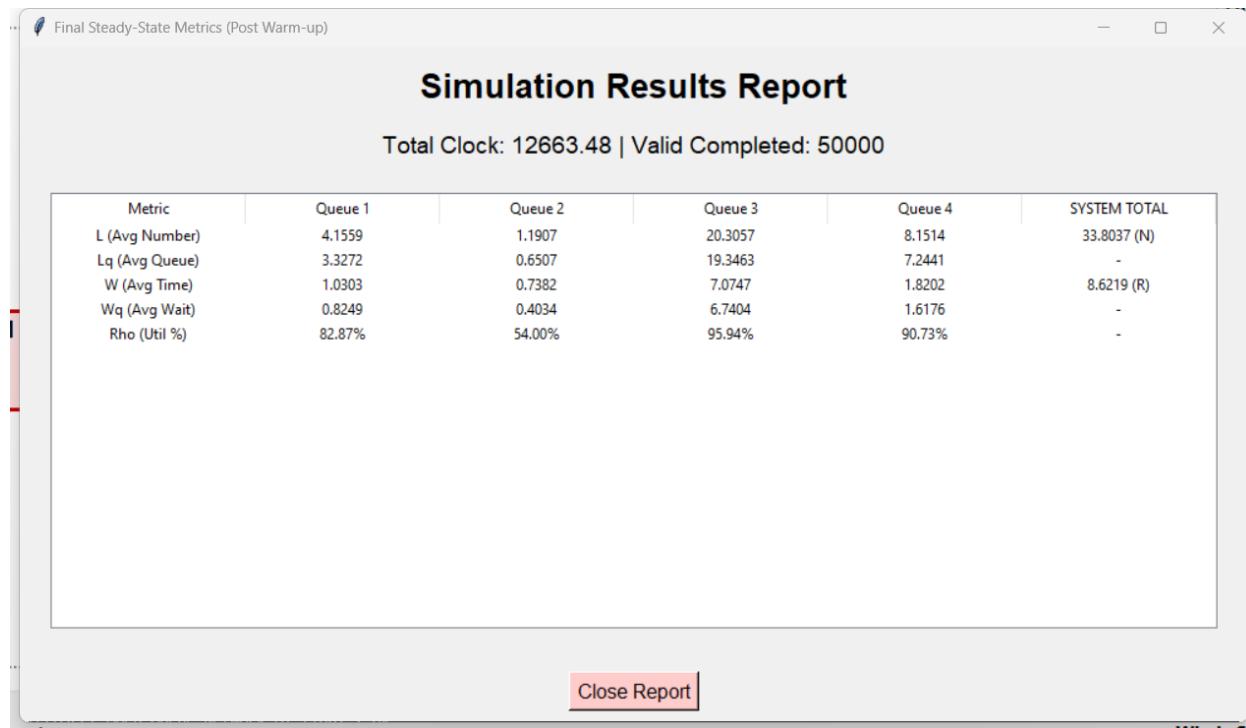
- $B(t)$ : مساحت انباشته شده زمان مشغول بودن سرور.

این امکان را به کاربر می‌دهد تا شبیه‌سازی را گام‌به‌گام دیباگ کند و صحت منطق را تایید نماید.

### ۵.۴ پنجره نتایج شبیه‌سازی

در سه حالت می‌توان به این پنجره دسترسی داشت(شکل ۱۱)،

- زدن دکمه "اجرای فوری" که در شبیه‌سازی فوراً برای تمام 50,000 مشتری انجام شده و نتایج آن نمایش داده می‌شود.



شکل ۱۱

- فشار دادن دکمه "نتایج" که نتایج شبیه‌سازی تا این لحظه از زمان را نشان می‌دهد؛ برای رسید نتایج به صورت لحظه‌ای و زنده.

- فشار دادن دکمه "پایان و مقایسه" که شبیه‌سازی را برای تمام 50,000 مشتری انجام داده و همچنین با روش حل تحلیلی معیارهای شبکه را محاسبه می‌کند؛ در یک پنجره نتایج حل تحلیلی و شبیه‌سازی را نشان می‌دهد و میزان تفاوت آنها را هم با درصد خطانشان می‌دهد. از آنجا که شبیه‌سازی از ورودی‌های تصادفی استفاده می‌کند در هر بار اجرا نتایج متفاوتی دارد و ممکن از در بعضی اجرایها نتایج آن با حل تحلیلی تفاوت معناداری داشته باشد؛ برای حل این چالش باید تعداد مشتریان شبیه‌سازی شده خیلی افزایش داده شود که شبکه به یک حالت کاملاً پایداری مشابه حل تحلیلی برسد که معمولاً زمان شبیه‌سازی را خیلی طولانی می‌کند(شکل ۱۲).

Comparison Report - Graphical View

## Analytical vs Simulation Comparison

Metric	Analytical	Simulation	Error %	Status
<b>SYSTEM WIDE METRICS</b>				
System Total N	31.4286	33.1159	5.37%	PASS
System Response R	7.8571	8.4711	7.81%	PASS
<b>QUEUE 1 METRICS</b>				
Q1 L (Avg Num)	4.0000	4.1049	2.62%	PASS
Q1 Lq (Avg Queue)	3.2000	3.2784	2.45%	PASS
Q1 W (Avg Time)	1.0000	1.0194	1.94%	PASS
Q1 Wq (Avg Wait)	0.8000	0.8141	1.77%	PASS
Q1 Rho (Util)	0.8000	0.8265	3.32%	PASS
<b>QUEUE 2 METRICS</b>				
Q2 L (Avg Num)	1.1429	1.1861	3.79%	PASS
Q2 Lq (Avg Queue)	0.6095	0.6395	4.92%	PASS
Q2 W (Avg Time)	0.7143	0.7288	2.04%	PASS
Q2 Wq (Avg Wait)	0.3810	0.3930	3.15%	PASS
Q2 Rho (Util)	0.5333	0.5466	2.49%	PASS
<b>QUEUE 3 METRICS</b>				
Q3 L (Avg Num)	18.2857	19.8470	8.54%	PASS
Q3 Lq (Avg Queue)	17.3376	18.8923	8.97%	PASS
Q3 W (Avg Time)	6.4286	6.9406	7.96%	PASS
Q3 Wq (Avg Wait)	6.0952	6.6067	8.39%	PASS
Q3 Rho (Util)	0.9481	0.9547	0.69%	PASS
<b>QUEUE 4 METRICS</b>				
Q4 L (Avg Num)	8.0000	7.9778	0.28%	PASS
Q4 Lq (Avg Queue)	7.1111	7.0705	0.57%	PASS
Q4 W (Avg Time)	1.8000	1.7779	1.23%	PASS
Q4 Wq (Avg Wait)	1.6000	1.5757	1.52%	PASS

**Close**

١٢ آذار

## ۶. اعتبارسنجی و مقایسه نتایج

آزمون نهایی یک مدل شبیه‌سازی، اعتبارسنجی آن در برابر یک راه حل تحلیلی شناخته شده است.

### ۶.۱ تنظیمات آزمایش

برای اطمینان از اعمال قانون اعداد بزرگ و به حداقل رساندن واریانس، شبیه‌سازی به صورت زیر پیکربندی شد:

- طول شبیه‌سازی: ۵۰,۰۰۰ مشتری تکمیل شده.
- گرم‌سازی: ۱,۰۰۰ مشتری.
- هسته تصادفی: تولیدکننده پیش‌فرض پایتون برای تولید اعداد تصادفی با نرخ نمایی (Mersenne Twister)

### ۶.۲ مقایسه نتایج

پس از اجرای شبیه‌سازی برای ۵۰,۰۰۰ مشتری، قابلیت "Finish & Compare" گزارش زیر(شکل ۱۳) را تولید کرد:

Comparison Report - Graphical View

### Analytical vs Simulation Comparison

Metric	Analytical	Simulation	Error %	Status
<strong>SYSTEM WIDE METRICS</strong>				
System Total N	31.4286	33.1159	5.37%	PASS
System Response R	7.8571	8.4711	7.81%	PASS
<strong>QUEUE 1 METRICS</strong>				
Q1 L (Avg Num)	4.0000	4.1049	2.62%	PASS
Q1 Lq (Avg Queue)	3.2000	3.2784	2.45%	PASS
Q1 W (Avg Time)	1.0000	1.0194	1.94%	PASS
Q1 Wq (Avg Wait)	0.8000	0.8141	1.77%	PASS
Q1 Rho (Util)	0.8000	0.8265	3.32%	PASS
<strong>QUEUE 2 METRICS</strong>				
Q2 L (Avg Num)	1.1429	1.1861	3.79%	PASS
Q2 Lq (Avg Queue)	0.6095	0.6395	4.92%	PASS
Q2 W (Avg Time)	0.7143	0.7288	2.04%	PASS
Q2 Wq (Avg Wait)	0.3810	0.3930	3.15%	PASS
Q2 Rho (Util)	0.5333	0.5466	2.49%	PASS
<strong>QUEUE 3 METRICS</strong>				
Q3 L (Avg Num)	18.2857	19.8470	8.54%	PASS
Q3 Lq (Avg Queue)	17.3376	18.8923	8.97%	PASS
Q3 W (Avg Time)	6.4286	6.9406	7.96%	PASS
Q3 Wq (Avg Wait)	6.0952	6.6067	8.39%	PASS
Q3 Rho (Util)	0.9481	0.9547	0.69%	PASS
<strong>QUEUE 4 METRICS</strong>				
Q4 L (Avg Num)	8.0000	7.9778	0.28%	PASS
Q4 Lq (Avg Queue)	7.1111	7.0705	0.57%	PASS
Q4 W (Avg Time)	1.8000	1.7779	1.23%	PASS
Q4 Wq (Avg Wait)	1.6000	1.5757	1.52%	PASS

Close

شکل ۱۳

### ۶.۳ تحلیل مغایرت‌ها

نتایج نشان‌دهنده دقیق بسیار بالایی هستند. اکثر معیارهای خطایی زیر ۵٪ دارند که برای یک شبیه‌سازی تصادفی عالی است.

ناهنجاری صف ۳:

بیشترین خطای در صف ۳ مشاهده می‌شود (~۶٪). این یک باگ نیست بلکه یک پدیده تئوری مورد انتظار است.

- صف ۳ دارای بهرهوری ۰.۹۵ است.
- در نظریه صف، وقتی بهرهوری به سمت ۱ میل می‌کند، واریانس طول صف منفجر می‌شود. صف ناپایدار شده و بهشدت بین خالی بودن و بسیار پر بودن نوسان می‌کند.
- برای ثبت "میانگین واقعی" چنین صف متغیری، نیاز به اجرای شبیه‌سازی بسیار طولانی‌تری (شاید میلیون‌ها مشتری) است تا نویز آماری هموار شود. با توجه به طول اجرای ۵۰,۰۰۰ تایی، انحراف ۶٪ از نظر آماری قابل قبول است.

## ۷. نتیجه‌گیری

### ۷.۱ خلاصه یافته‌ها

این پژوهه با موفقیت یک شبیه‌سازی رخداد گستته را برای یک شبکه صف باز با بازخورد طراحی، پیاده‌سازی و اعتبارسنجی کرد. مقایسه بین مدل تحلیلی شبکه جکسون و نتایج شبیه‌سازی، صحت پیاده‌سازی DES را اثبات کرد. سیستم توانست رفتارهای پیچیده‌ای مانند مسیریابی احتمالی و حلقه‌های بازخورد را با موفقیت تقلید کند.

### ۷.۲ چالش‌ها و محدودیت‌ها

چالش اصلی مدیریت شرایط بار زیاد در صفحه ۳ بود. شبیه‌سازی به وضوح نشان داد که حتی یک سیستم پایدار (دارای بهره‌وری کمتر از ۱) می‌تواند وقتی بهره‌وری از ۹۰٪ عبور می‌کند، دچار ازدحام شدید و واریانس بالا شود.

### ۷.۳ کارهای آتی

توسعه‌های آتی این کار می‌تواند شامل موارد زیر باشد:

- پیاده‌سازی توزیع‌های عمومی (G/G/1) که در آن‌ها استخراج راه حل تحلیلی دشوارتر است.
- افزودن ظرفیت محدود برای صفحه‌های مطالعه احتمالات از دست رفتن بسته (Packet Loss).
- پیاده‌سازی زمان‌بندی اولویت‌دار (مثلًاً اولویت دادن به مشتریان بازخوردهای نسبت به مشتریان جدید).