

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



ارزیابی کارایی سیستم‌های کامپیوتری

پروژه: شبیه‌سازی شبکه صف باز

آرمان آزادی

زمستان ۱۴۰۴

فهرست مطالب

۱.	مقدمه.....	۴
۲.	شرح مسئله.....	۶
۳.	مدل سازی تحلیلی.....	۱۰
۴.	طراحی شبیه سازی رخداد گسسته.....	۱۶
۵.	پیاده سازی رابط کاربری گرافیکی (GUI).....	۲۰
۶.	اعتبارسنجی و مقایسه نتایج.....	۲۶
۷.	نتیجه گیری	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 صف، 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 صف پیچیده و باز بهره می‌برد.

۱.۲ مروری 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 به‌عنوان 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 (مثلاً تعداد کارها در یک صف).
- **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.
- **انباشتگرهای 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 کد.

۲. شرح مسئله

۲.۱ توپولوژی و معماری سیستم

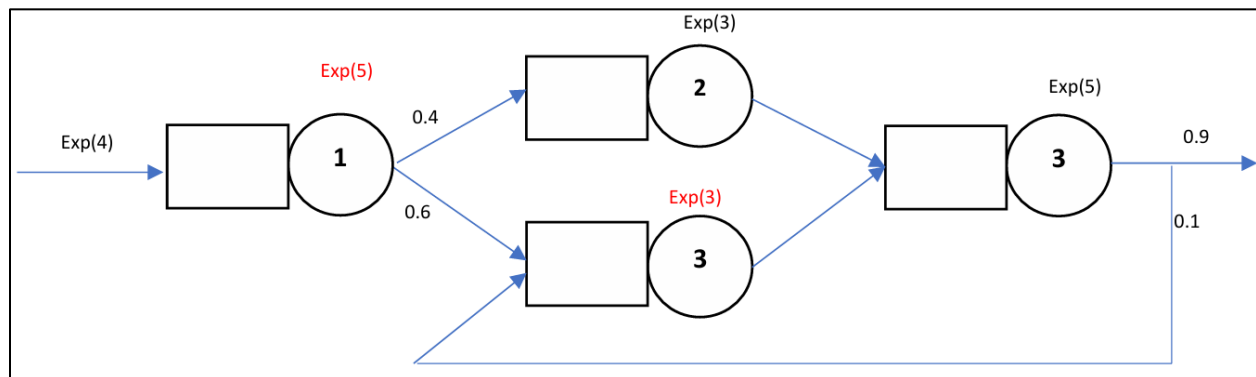
سیستم مورد مطالعه یک شبکه صف باز (Open Queueing Network) متشکل از چهار ایستگاه سرویس‌دهی (گره) متمایز است (شکل ۱، شکل ۲). این سیستم "باز" نامیده می‌شود زیرا مشتریان از یک منبع خارجی وارد شده و در نهایت از سیستم خارج می‌شوند.

توپولوژی شبکه شامل موارد زیر می‌باشد:

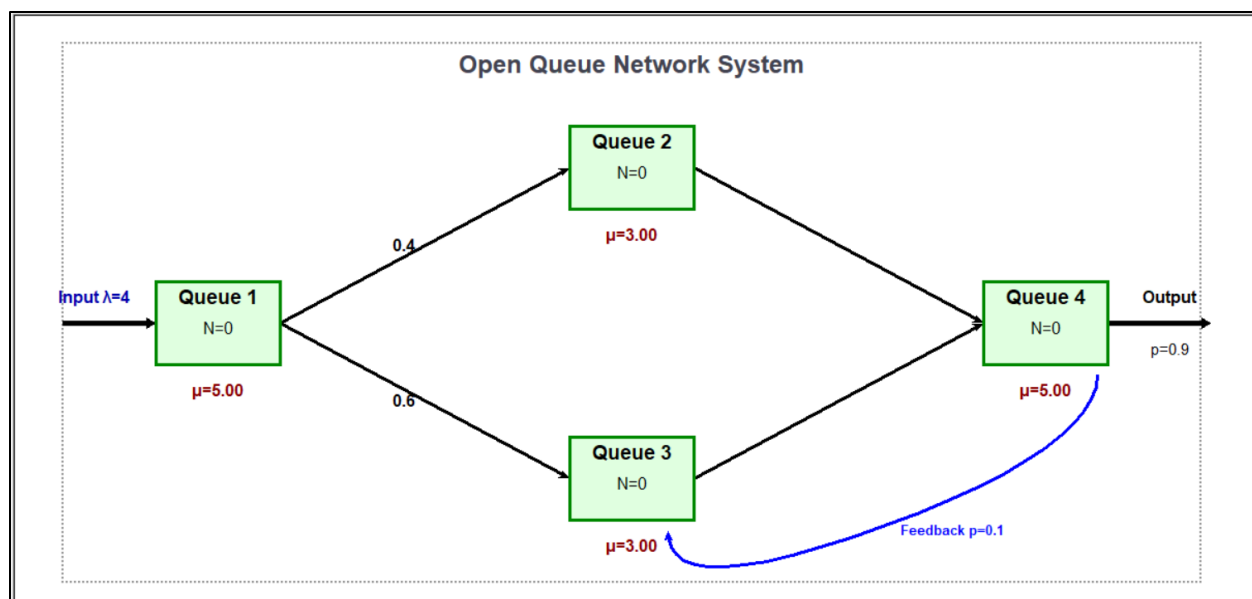
مسیرهای انشعابی: مشتریان که از صف ۱ خارج می‌شوند می‌توانند به صف ۲ یا صف ۳ بروند.

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 می‌دهد.

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 صف باز مسئله

پارامتر	مقدار	توضیحات
Lambda (source)	۴۰	نرخ ورود از دنیای خارج
mu_1	۵۰	نرخ سرویس صف ۱
mu_2	۳۰	نرخ سرویس صف ۲
mu_3	۳۰	نرخ سرویس صف ۳ (بار زیاد)
mu_4	۵۰	نرخ سرویس صف ۴

۲.۳ منطق مسیریابی و احتمالات

جابجایی مشتریان بین صف‌ها 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:

- $P_{1,2} = 0.4$ (40% به صف ۲ می‌روند)

- $P_{1,3} = 0.6$ (60% به صف ۳ می‌روند)

از صف ۲:

- $P_{2,4} = 1.0$ (100% به صف ۴ می‌روند)

از صف ۳:

- $P_{3,4} = 1.0$ (100% به صف ۴ می‌روند)

از صف ۴:

• $P_{4,out} = 0.9$ (90% از سیستم خارج می شود)

• $P_{4,3} = 0.1$ (10% به صف 3 می روند)

۳. مدل سازی تحلیلی

برای اعتبارسنجی شبیه سازی، ابتدا باید سیستم را به صورت ریاضی حل کنیم. از آنجا که فرآیند ورود پواسون و زمان های سرویس نمایی هستند، سیستم به عنوان یک شبکه جکسون شناخته می شود. قضیه جکسون به ما اجازه می دهد که با محاسبه نرخ ورود مؤثر (λ_i) ، هر گره i را به عنوان یک صف مستقل $M/M/1$ در نظر بگیریم.

۳.۱ مبانی ریاضی

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

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

که در آن:

- λ_i : کل نرخ ورود مؤثر به گره i
- γ_i : نرخ ورود خارجی به گره i
- P_{ji} : احتمال انتقال از گره j به گره i

۳.۲ استخراج گام به گام نرخ های ورود مؤثر

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

گره ۱: گره ۱ تنها ترافیک خارجی را دریافت می کند.

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

گره ۲: گره ۲ ترافیک را فقط از گره ۱ دریافت می کند.

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

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

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 λ_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 λ_3 و λ_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 λ_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:

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

صف ۱:

$$(\mu = 5)$$

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

$$L_1 = 0.8/0.2 = 4.00$$

صف ۲:

$$(\mu = 3)$$

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

$$L_2 = 0.5333/0.4667 = 1.143$$

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.

$$(\mu = 5)$$

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

$$L_4 = 0.8889/0.1111 = 8.000$$

۳.۴ تجميع 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 ($N = \lambda_{\text{source}} R$):

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

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 گسسته

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     return msg
144
```


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 (۰ 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.

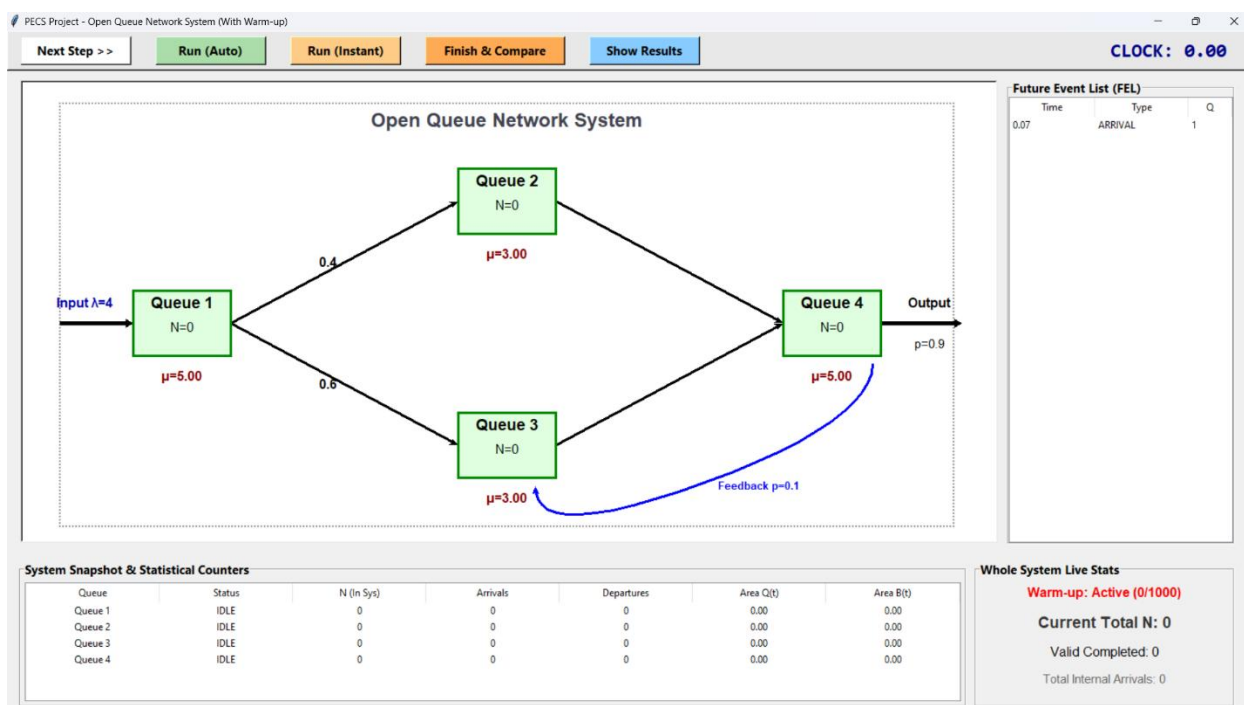
- ۱۰۰۰ مشتری اول پردازش می‌شوند تا صف‌ها پر شوند.
- در طول این فاز، هیچ داده آماری ثبت نمی‌شود.
- به محض خروج هزارمین مشتری، تمام انباشتگرهای آماری صفر می‌شوند و شبیه‌سازی "واقعی" آغاز می‌گردد.
- در کل شبکه برای اینکه به حالت پایداری برسد برای 50,000 مشتری شبیه‌سازی می‌شود (بدون احتساب 1000 مشتری اول).

```
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).

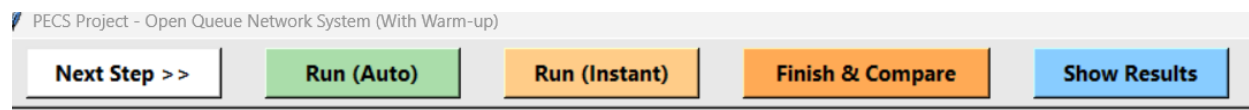


شکل ۳

۵.۱ طراحی رابط

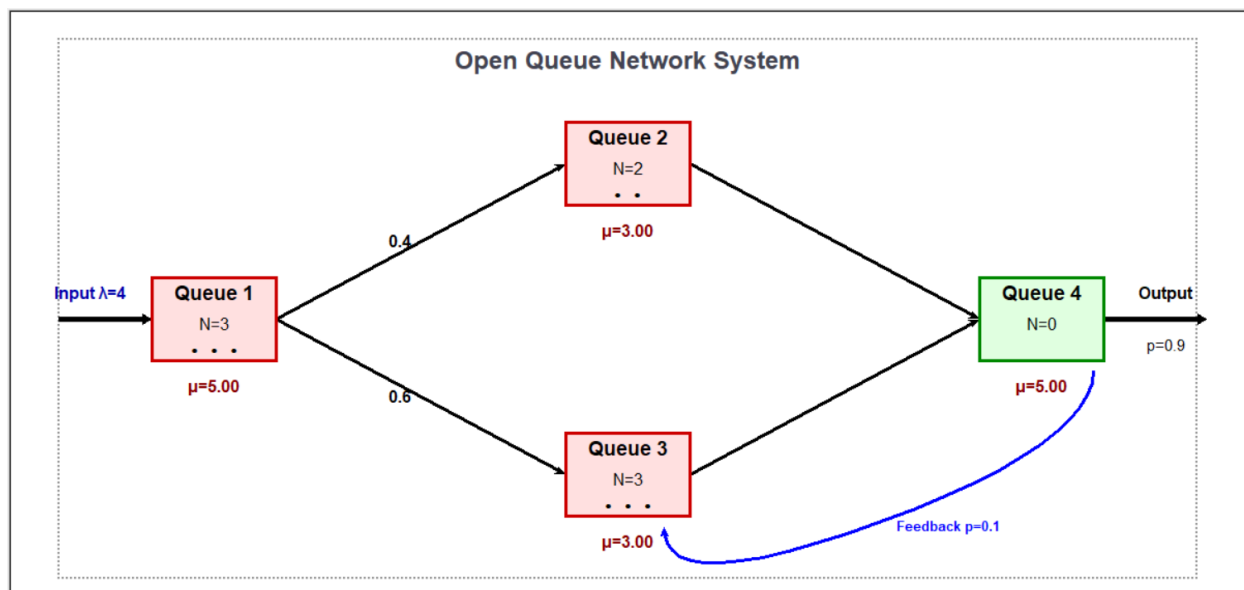
رابط کاربری به چند بخش اصلی تقسیم می‌شود:

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 صف‌های فعلی (شکل ۵).



شکل ۵

دانشبورد داده‌ها (پایین-چپ): نمایش لیست معیارها و شمارنده‌های آماری بلادرنگ (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

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

Whole System Live Stats	
Warm-up: Active (4/1000)	
Current Total N: 8	
Valid Completed: 0	
Total Internal Arrivals: 25	

شکل ۷



شکل ۸

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

CLOCK: 3.14		
Future Event List (FEL)		
Time	Type	Q
3.17	DEPARTURE	2
3.25	DEPARTURE	3
3.25	DEPARTURE	1
3.33	ARRIVAL	1

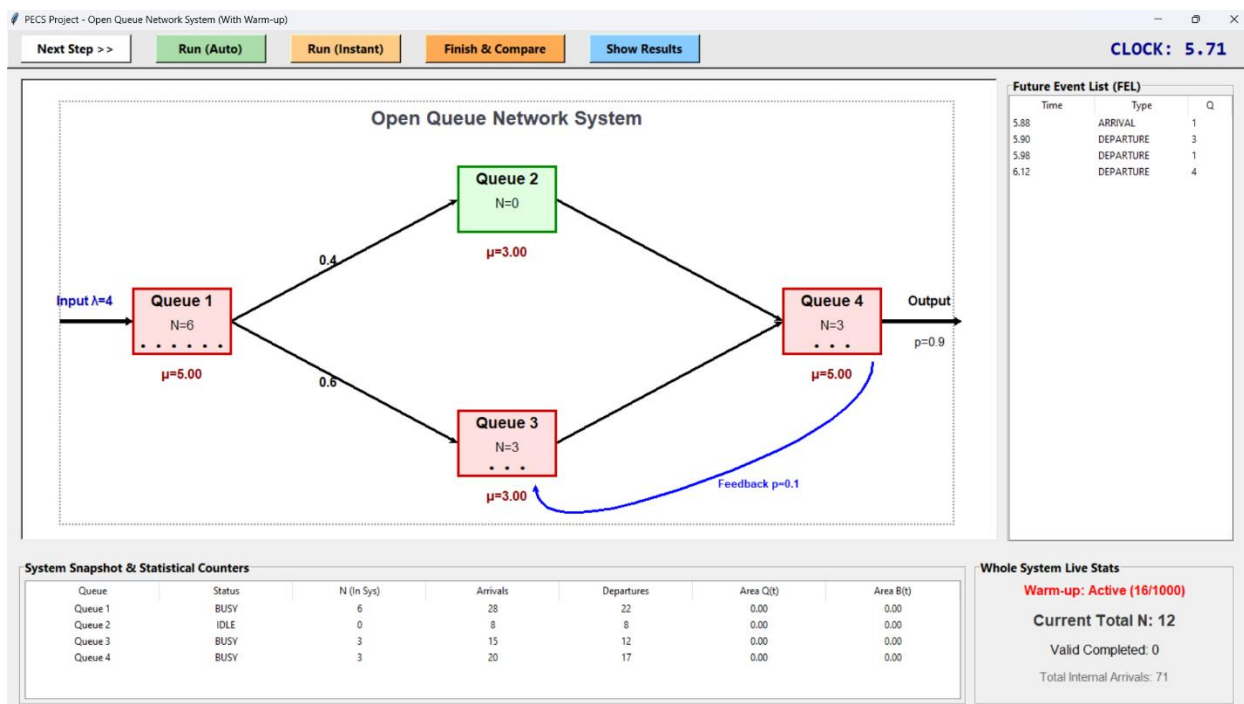
شکل ۹

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.



شکل ۱۰

۵.۳ پایش آماری بلادرنگ

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

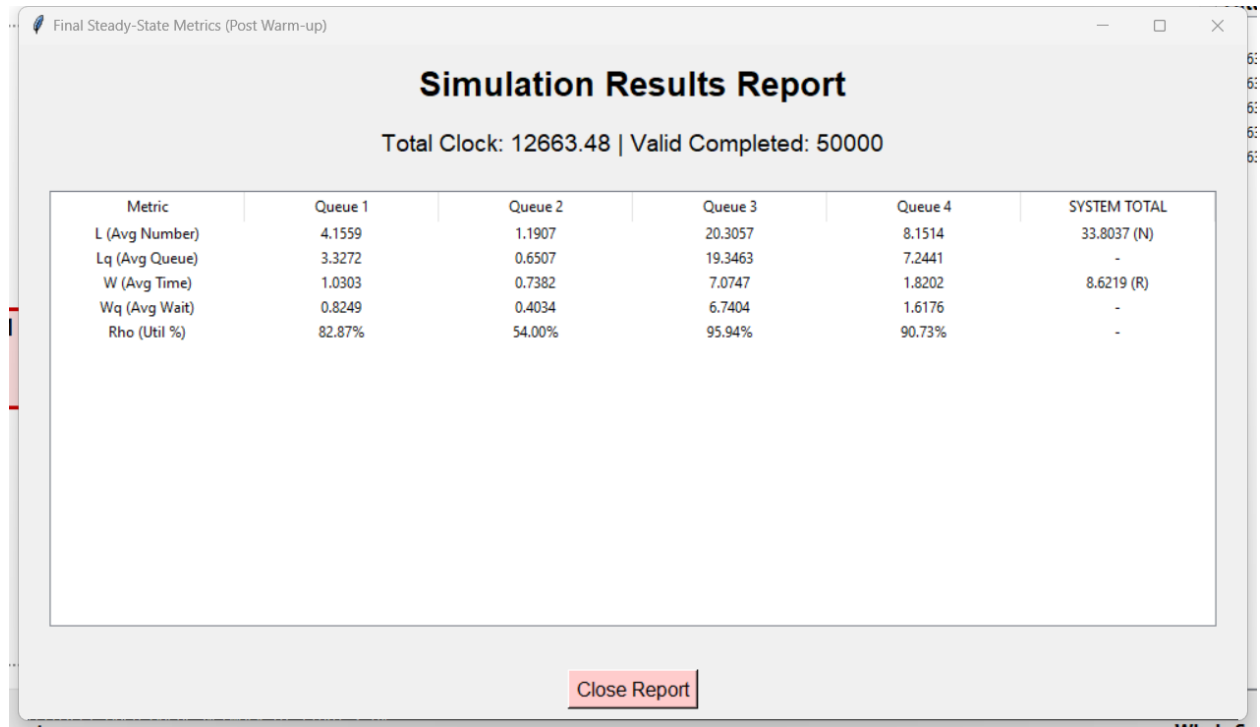
- N : تعداد مشتریان در سیستم.
- $Q(t)$: مساحت انباشته شده صف.
- $B(t)$: مساحت انباشته شده زمان مشغول بودن سرور.

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)،

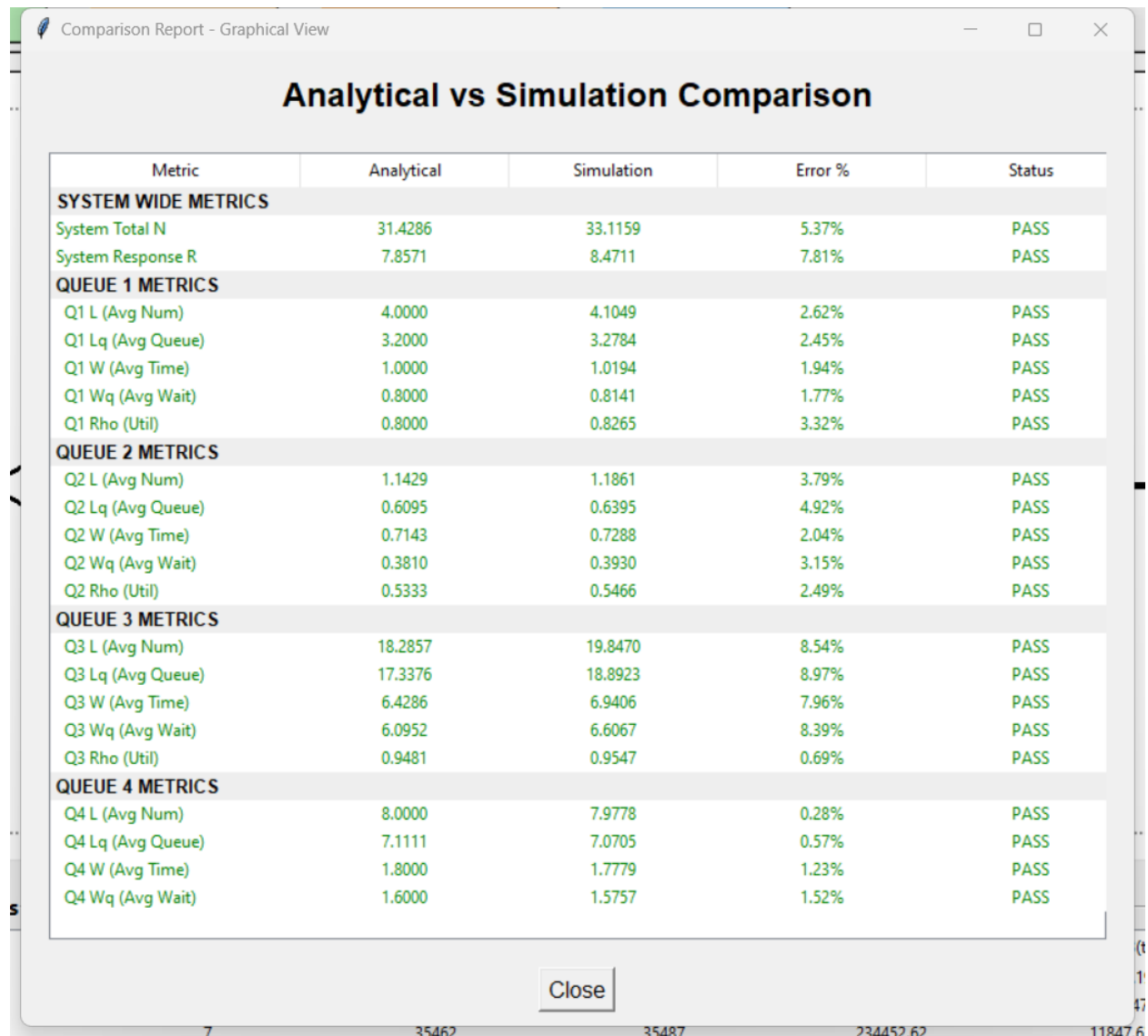
- زدن دکمه "اجرای فوری" که در شبیه‌سازی فوراً برای تمام 50,000 مشتری انجام شده و نتایج آن نمایش داده می‌شود.



Metric	Queue 1	Queue 2	Queue 3	Queue 4	SYSTEM TOTAL
L (Avg Number)	4.1559	1.1907	20.3057	8.1514	33.8037 (N)
Lq (Avg Queue)	3.3272	0.6507	19.3463	7.2441	-
W (Avg Time)	1.0303	0.7382	7.0747	1.8202	8.6219 (R)
Wq (Avg Wait)	0.8249	0.4034	6.7404	1.6176	-
Rho (Util %)	82.87%	54.00%	95.94%	90.73%	-

شکل ۱۱

- فشار دادن دکمه "نتایج" که نتایج شبیه‌سازی تا این لحظه از زمان را نشان می‌دهد؛ برای رسیدن 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 خطا نشان می‌دهد. از آنجا که شبیه‌سازی از ورودی‌های تصادفی استفاده می‌کند در هر بار اجرا نتایج متفاوتی دارد و ممکن است در بعضی اجراها 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).



شکل ۱۲

۶. اعتبارسنجی و مقایسه نتایج

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

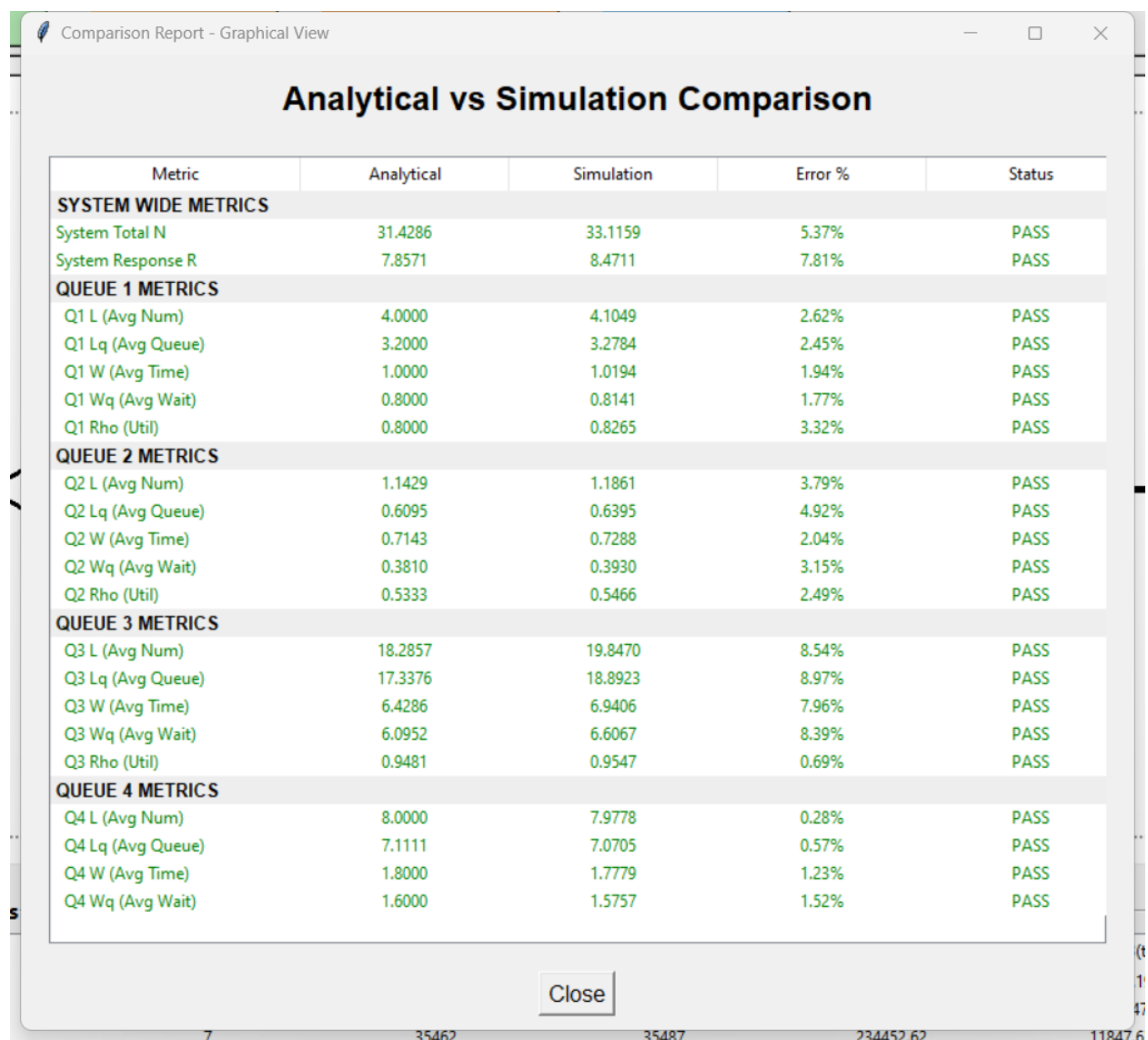
۶.۱ تنظیمات آزمایش

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:



شکل ۱۳

۶.۳ تحلیل مغایرت‌ها

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

ناهنجاری صف ۳:

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

- صف ۳ دارای بهره‌وری 0.95 است.
- در نظریه صف، وقتی بهره‌وری به سمت 1 میل می‌کند، واریانس طول صف منفجر می‌شود. صف ناپایدار شده و به شدت 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 پروژه با موفقیت یک شبیه‌سازی رخداد گسسته 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 صف ۳ بود. شبیه‌سازی به وضوح نشان داد که حتی یک سیستم پایدار (دارای بهره‌وری کمتر از 1) 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 صف‌ها جهت مطالعه احتمالات از دست رفتن بسته (Packet Loss).
- پیاده‌سازی زمان‌بندی اولویت‌دار (مثلاً اولویت دادن به مشتریان بازخوردی نسبت به مشتریان جدید).