The Erlang-R Queue: Time-Varying QED Queues with Reentrant Customers in Support of Healthcare Staffing

Galit Yom-Tov    Avishai Mandelbaum

Industrial Engineering and Management
Technion

MSOM Conference, June 2010
The Problem Studied

Problems in Emergency Departments:

- Hospitals do not manage patients flow.
- Long waiting times in the ED for physicians, nurses, and tests.
  => Deterioration in medical state.
- Patients leave ED without being seen or abandon during their stay.
  => Patient return in severe state.

We use **Service Engineering** approach to reduce these effects.
Can we determine the **number of physicians (and nurses)** needed to improve patients flow, and control the system in balance between service quality and efficiency?
Standard assumption in service models: service time is continuous.
But we find systems in which: service is dis-continuous and customers re-enter service again and again.

What is the appropriate staffing procedure?
What is the significance of the re-entering customers?
What is the implication of using simple Erlang-C models for staffing?
Standard assumption in service models: service time is continuous.
But we find systems in which: service is dis-continuous and customers re-enter service again and again.

What is the appropriate staffing procedure?
What is the significance of the re-entering customers?
What is the implication of using simple Erlang-C models for staffing?
Related Work

Mandelbaum A., Massey W.A., Reiman M.
*Strong Approximations for Markovian Service Networks.* 1998.

Massey W.A., Whitt W.
*Networks of Infinite-Server Queues with Nonstationary Poisson Input.* 1993.

Green L., Kolesar P.J., Soares J.
*Improving the SIPP Approach for Staffing Service Systems that have Cyclic Demands.* 2001.

Jennings O.B., Mandelbaum A., Massey W.A., Whitt W.
*Server Staffing to Meet Time-Varying Demand.* 1996.

Feldman Z., Mandelbaum A., Massey W.A., Whitt W.
The (Time-Varying) Erlang-R Queue:

- $\lambda_t$ - Arrival rate of a time-varying Poisson arrival process.
- $\mu$ - Service rate.
- $\delta$ - Delay rate ($1/\delta$ is the delay time between services).
- $p$ - Probability of return to service.
- $s_t$ - Number of servers at time $t$.

**Model Definition**

**The Erlang-R Queue**

Motivation

The Erlang-R Queue

Results

Model Definition

Staffing Time-Varying Erlang-R Queue
Patients Arrivals to an Emergency Department

Galit Yom-Tov, Avishai Mandelbaum
The Erlang-R Queue
Staffing: Determine $s_t, t \geq 0$

- Based on the QED-staffing formula:

$$s = R + \beta \sqrt{R}, \quad \text{where } R = \lambda E[S]$$

- In time-varying environments: $s(t) = R(t) + \beta \sqrt{R(t)}$, where $\beta$ is chosen according to the steady-state QED.

- Two approaches to calculate the time-varying offered load ($R(t)$):
  - **PSA / SIPP (lag-SIPP)** - divide the time-horizon to planning intervals, calculate average arrival rate and steady-state offered-load for each interval, then staff according to steady-state recommendation (i.e., $R(t) \approx \bar{\lambda}(t)E[S]$).
  - **MOL/IS** - assuming no constraints on number of servers, calculate the time-varying offered-load. For example, in a single service system:

$$R(t) = E[\int_{t-S_e}^t \lambda(u)du] = E[\lambda(t - S_e)]E[S].$$
Staffing: Determine $s_t$, $t \geq 0$

- Based on the QED-staffing formula:
  \[ s = R + \beta \sqrt{R}, \quad \text{where } R = \lambda E[S] \]

- In time-varying environments: $s(t) = R(t) + \beta \sqrt{R(t)}$, where $\beta$ is chosen according to the steady-state QED.

- Two approaches to calculate the time-varying offered load ($R(t)$):
  - **PSA / SIPP (lag-SIPP)** - divide the time-horizon to planning intervals, calculate average arrival rate and steady-state offered-load for each interval, then staff according to steady-state recommendation (i.e., $R(t) \approx \bar{\lambda}(t)E[S]$).
  - **MOL/IS** - assuming no constraints on number of servers, calculate the time-varying offered-load. For example, in a single service system:
    \[ R(t) = E[\int_{t-S_e}^{t} \lambda(u)du] = E[\lambda(t - S_e)]E[S]. \]
Motivation

The Erlang-R Queue

Results

Model Definition

Staffing Time-Varying Erlang-R Queue

Staffing: Determine $s_t, t \geq 0$

- Based on the QED-staffing formula:
  \[ s = R + \beta \sqrt{R}, \quad \text{where } R = \lambda E[S] \]

- In time-varying environments: $s(t) = R(t) + \beta \sqrt{R(t)}$, where $\beta$ is chosen according to the steady-state QED.

- Two approaches to calculate the time-varying offered load ($R(t)$):
  - **PSA / SIPP (lag-SIPP)** - divide the time-horizon to planning intervals, calculate average arrival rate and steady-state offered-load for each interval, then staff according to steady-state recommendation (i.e., $R(t) \approx \bar{\lambda}(t)E[S]$).
  - **MOL/IS** - assuming no constraints on number of servers, calculate the time-varying offered-load. For example, in a single service system:
    \[ R(t) = E[\int_{t-S_e}^{t} \lambda(u) du] = E[\lambda(t - S_e)]E[S]. \]
The Offered-Load

Offered-Load in Erlang-R = The number of busy servers (or the number of customers) in a corresponding \((M_t/M/\infty)^2\) network.

**Theorem: (Massey and Whitt 1993)**

\[ R(t) = (R_1(t), R_2(t)) \] is determined by the following expression:

\[ R_i(t) = E[\lambda_i^+(t - S_{i,e})]E[S_i] \]

where,

\[ \lambda_1^+(t) = \lambda(t) + E[\lambda_2^+(t - S_2)] \]
\[ \lambda_2^+(t) = pE[\lambda_1^+(t - S_1)] \]

**Theorem:**

If service times are exponential, \( R(t) \) is the solution of the following Fluid ODE:

\[ \frac{d}{dt}R_1(t) = \lambda_t + \delta R_2(t) - \mu R_1(t), \]
\[ \frac{d}{dt}R_2(t) = p\mu R_1(t) - \delta R_2(t). \]
Case Study: Sinusoidal Arrival Rate

Periodic arrival rate: $\lambda_t = \bar{\lambda} + \bar{\lambda}\kappa \sin(\omega t)$.

$\bar{\lambda}$ is the average arrival rate, $\kappa$ is the relative amplitude, and $\omega$ is the frequency.
Case Study: Sinusoidal Arrival Rate

Simulation of $P(Wait)$ for various $\beta$ ($0.1 \leq \beta \leq 1.5$)

Performance measure is stable! ($0.15 \leq P(Wait) \leq 0.85$)
Case Study: Sinusoidal Arrival Rate

Relation between $P(\text{wait})$ and $\beta$ fits steady-state theory!
Case Study: Sinusoidal Arrival Rate

Simulation results of servers’ utilization for various $\beta$

Performance measure is stable! ($0.85 \leq Util \leq 0.98$)
Using Erlang-C’s \( R(t) \), does not stabilize systems’ performance.
**Why Erlang-C Does Not Fit Re-entrant Systems?**

**Compare R(t) of Erlang-C and Erlang-R:**

**Erlang-C offered-load (with concatenated services):**

\[
R(t) = E \left[ \lambda \left( t - \frac{1}{1 - p} S_{1,e} \right) \right] E \left[ \frac{1}{1 - p} S_1 \right]
\]

**Erlang-R offered-load:**

\[
R_1(t) = E \left[ \sum_{i=1}^{\infty} p^i \lambda \left( t - S_{1,e}^i - S_{2,e}^i - S_1,e \right) \right] E[S_1]
\]

![Diagram](image)
Erlang-C under- or over-estimates the Erlang-R offered-load.
Comparison between Erlang-C and Erlang-R

Theorem:
The ratio of amplitudes between Erlang-R and Erlang-C is given by

\[ \sqrt{\frac{(\delta^2 + \omega^2)\left((1 - p)\mu^2 + \omega^2\right)}{((\mu - i\omega)(\delta - i\omega) - p\mu\delta)((\mu + i\omega)(\delta + i\omega) - p\mu\delta)}} \]

Plot of amplitudes ratio as a function of \( \omega \)
Erlang-R over-estimate the amplitude of the offered-load. The re-entrant patients stabilize the system. Minimum ratio achieved when: $\omega = \sqrt{\delta \mu (1 - p)}$ (for example ED).
Comparison between Erlang-C and Erlang-R

Plot of the ratio of phases as a function of $\omega$

Erlang-C under- or over-estimates the time-lag.
Comparison between Erlang-C and Erlang-R

Erlang-C under- or over-estimates this time-lag depending on the period's length.

\[ \lambda = 30, \mu = 1, \delta = 0.5, p = \frac{2}{3}, \text{cycles per day} = 1 \]

\[ \lambda = 30, \mu = 1, \delta = 0.5, p = \frac{2}{3}, \text{cycles per day} = 4 \]
Small systems - Hospitals

Small systems: No of doctors range from 1 to 5

Constraints:
- Staffing resolution: 1 hour
- Minimal staffing: 1 doctor per type
- Integer values: \( s(t) = [R_1(t) + \beta \sqrt{R_1(t)}] \)

Example: \( R = 2.75 \)

<table>
<thead>
<tr>
<th>( \beta ) range</th>
<th>s</th>
<th>( P(W &gt; 0) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>[0, 0.474]</td>
<td>3</td>
<td>82.4%</td>
</tr>
<tr>
<td>(0.474, 1.055]</td>
<td>4</td>
<td>34.0%</td>
</tr>
<tr>
<td>(1.055, 1.658]</td>
<td>5</td>
<td>11.4%</td>
</tr>
<tr>
<td>(1.658, 2.261]</td>
<td>6</td>
<td>3.0%</td>
</tr>
<tr>
<td>1.658 and up</td>
<td>7</td>
<td>0%</td>
</tr>
</tbody>
</table>

\( => \) Can not achieve all performance levels!
Small systems - Hospitals

P(Wait) is stable and separable!
Conclusions

In time-varying systems where patients return for multiple services:

1. Using the MOL (IS) algorithm for staffing stabilizes performance.
2. Re-entrant patients stabilize the system.
3. Using single-service models, such as Erlang-C, is problematic in the re-entrant ED environment:
   - Time-varying arrivals
   - Transient behavior even with constant parameters
What next?

- Fluid and diffusion approximations for mass-casualty events
- QED - MOL approximations for the processes:
  - Number of customers in system
  - Virtual waiting time
- Extension: upper limit on the number of customers within the system
Motivation

The Erlang-R Queue

Results

Case Study
Analyzing of the offered load function

What next?

The semi-open Erlang-R queue

Does MOL approximation works? yes, stabilizing performance is achieved.
Is it close to M/M/s/n model? no.

Erlang-R: 

Closed Erlang-R:

IW model closed network:

Needy

Content

Delay

1-p

p

Arrivals

1

2

Arrivals: Needy:

Content:

Cleaning:

N beds

Patient is Needy

1-p

p

Patient is Content

Blocked patients

N

Galit Yom-Tov, Avishai Mandelbaum

The Erlang-R Queue
Thank You