The Erlang-R Queue: A Model Supporting Personnel Staffing in Emergency Departments

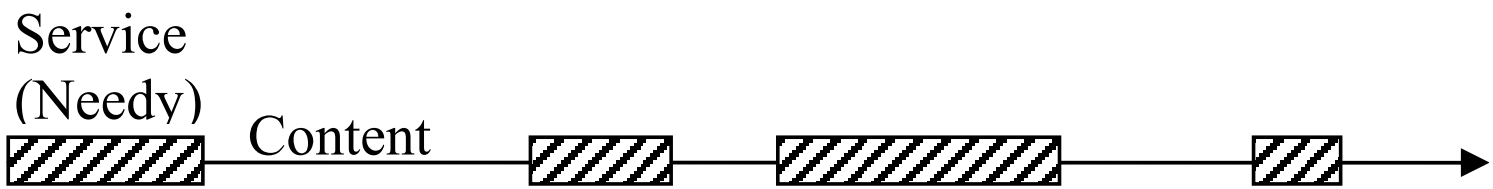
Galit Yom-Tov Avishai Mandelbaum

Industrial Engineering and Management
Technion

ORSIS, May 10-11, 2009

The Problem Studied

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.



Examples:

- Machine-Repairman (closed queueing network).
- Medical Unit model (semi-open queueing network).
- Emergency Department (open queueing network).

What is the appropriate staffing procedure? What is the significance of these cycles? Can one still use 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.

Staffing of Time-Varying Queues to Achieve Time-Stable Performance. 2007.

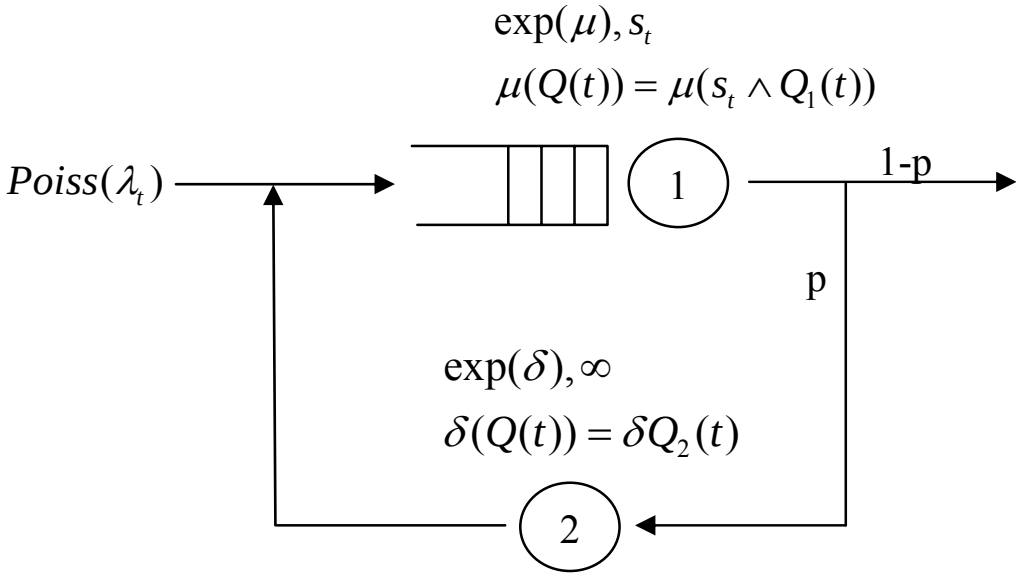
Research Outline

- Develop a queueing model that incorporates service and customers' **R**e-entrance: Erlang-**R**.
- Develop fluid and diffusion approximations.
- Develop staffing algorithm that attain pre-specified service levels.
- Validate our approach via simulation, based on hospital data.
- Understand when Erlang-R is needed.

Model Definition

The (Time-Varying) Erlang-R Queue:

- λ_t - Arrival rate of a time-varying Poisson arrival process.
- μ - Exponential service rate.
- δ - Delay rate ($1/\delta$ is the delay time between services).
- p - Probability of return to service.
- s_t - Number of servers at time t.
- $Q_i(t)$ - Number of customers in node i at time t, $i = 1, 2$.



Fluid and Diffusion Approximations for Erlang-R

We scale by η: $\lambda_t \to \eta\lambda_t$ and $s_t \to \eta s_t$.

Theorem: Fluid (FSSLN) and Diffusion (FCLT) Approximations

As $\eta \to \infty$,

$$\frac{Q^\eta(t)}{\eta} \to Q^{(0)}(t), \quad \textit{and} \quad \sqrt{\eta}\left[\frac{Q^\eta(t)}{\eta} - Q^{(0)}(t)\right] \stackrel{d}{=} Q^{(1)}(t),$$

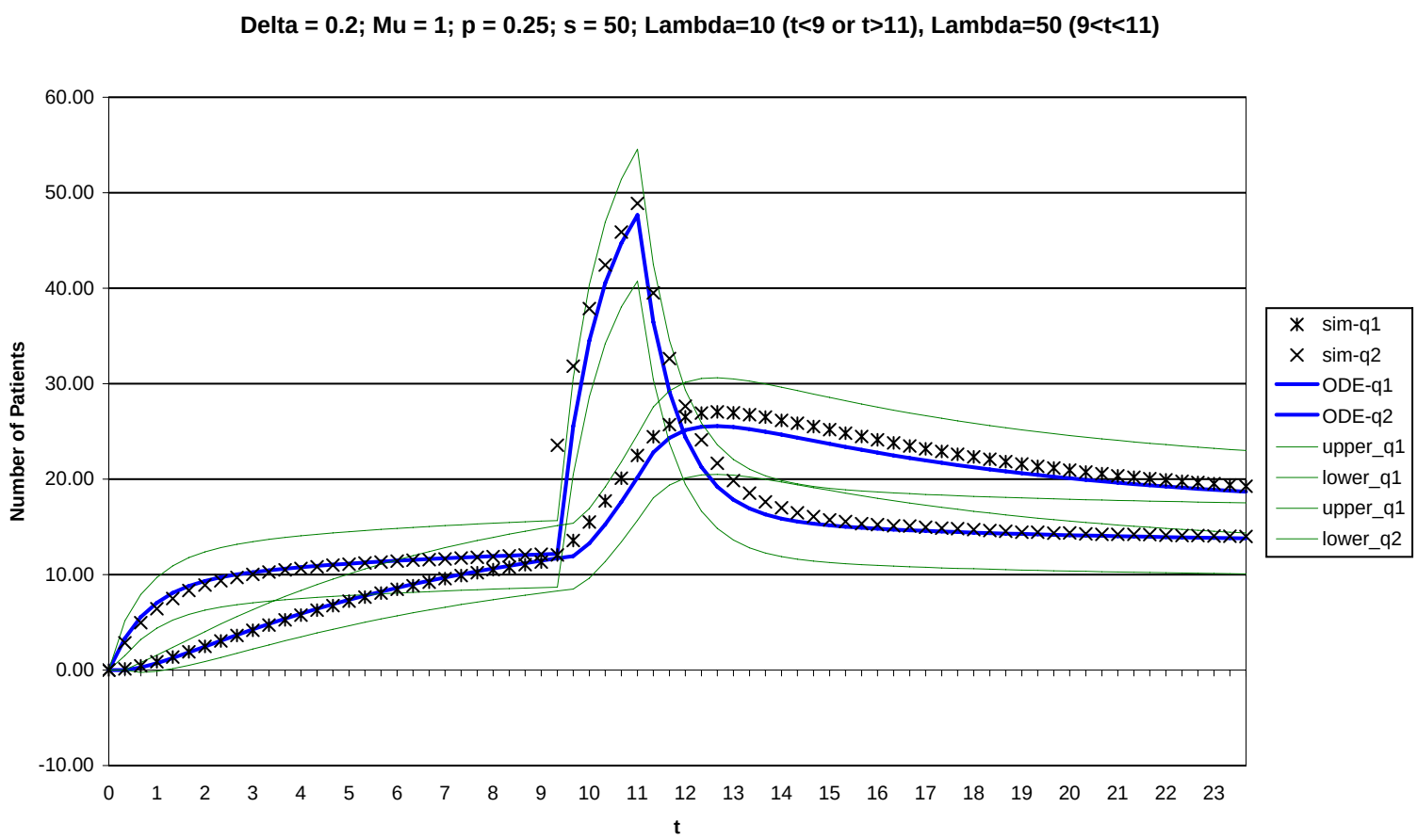
where $Q^{(0)}(t)$ is the solution of the following ODE:

$$Q_1^{(0)}(t) = Q_1^{(0)}(0) + \int_0^t \left(\lambda_\tau - \mu_\tau\left(Q_1^{(0)}(\tau) \wedge s_\tau\right) + \delta_\tau Q_2^{(0)}(\tau)\right) d\tau$$

$$Q_2^{(0)}(t) = Q_2^{(0)}(0) + \int_0^t \left(p\mu_\tau\left(Q_1^{(0)}(\tau) \wedge s_\tau\right) - \delta_\tau Q_2^{(0)}(\tau)\right) d\tau,$$

and $Q^{(1)}(t)$ is the solution of a SDE (Stochastic Differential Equation).

Example of Fluid and Diffusion Approximations for Erlang-R



Staffing: Determine s_t, $t \geq 0$

- Based on the QED-staffing formula:

$$s = R + \beta\sqrt{R}$$

- Two approaches:
  - 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 time-varying offered load. For example in Erlang-C: $R(t) = E[\int_{t-S}^{t} \lambda(u)du = E[\lambda(t-S)]E[S]$.

Staff according to the square-root formula where $R(t)$ replaces R: $s(t) = R(t) + \beta\sqrt{R(t)}$.

The Offered-Load

Offered-Load in Erlang-R queue = The number of busy servers (or the number of customers) in a corresponding $(M_t/M/\infty)^2$ network. $R(t)$ is denoted by the following two expressions:

$$R_1(t) = E\left[\int_{t-S_1}^{t} \lambda_u + \delta Q_2^{(0),\infty}(u)du\right]$$

$$R_2(t) = E\left[\int_{t-S_2}^{t} p\mu Q_1^{(0),\infty}(u)du\right]$$

where $Q^{(0),\infty}(t)$ is the solution of the following Fluid ODE:

$$\frac{d}{dt}Q_1^{(0),\infty}(t) = \lambda_t + \delta Q_2^{(0),\infty}(t) - \mu Q_1^{(0),\infty}(t),$$

$$\frac{d}{dt}Q_2^{(0),\infty}(t) = p\mu Q_1^{(0),\infty}(t) - \delta Q_2^{(0),\infty}(t).$$

Staffing: Determine s_t, $t \geq 0$

MOL Algorithm for Erlang-R:

- Solve differential equations for $Q^{\infty}(t)$.
- Calculate time-varying offered load - $R(t)$.
- Staff according to square-root formula: $s(t) = R(t) + \beta\sqrt{R(t)}$, where β is chosen according to the steady-state Halfin-Whitt formula.

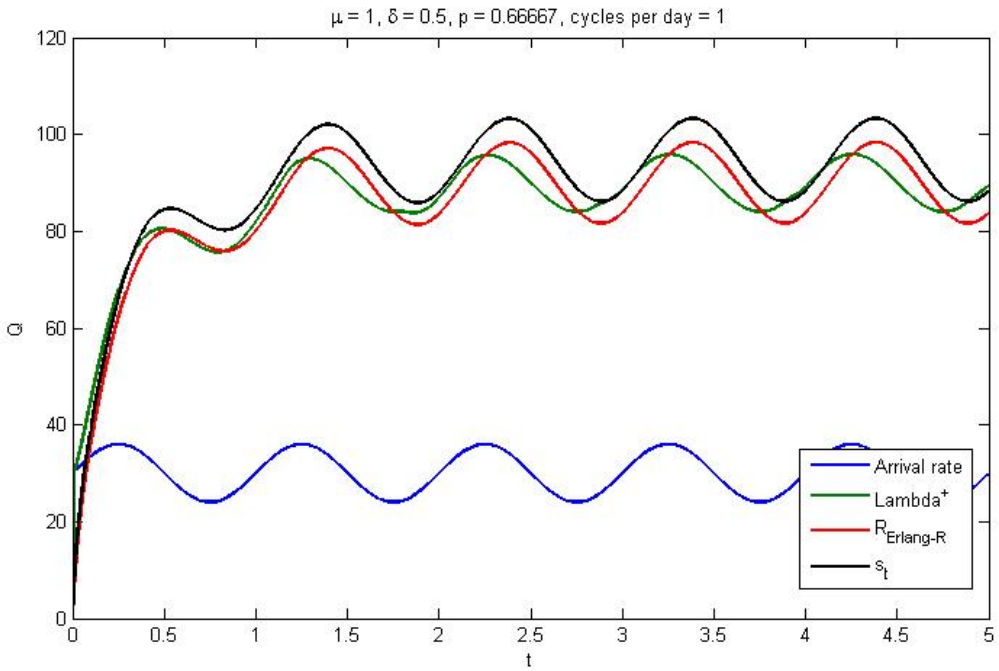
Case Study: Sinusoidal Arrival Rate

Examine the following periodic arrival rate:

$$\lambda_t = \bar{\lambda} + \bar{\lambda}\alpha sin(2\pi t/\psi)$$

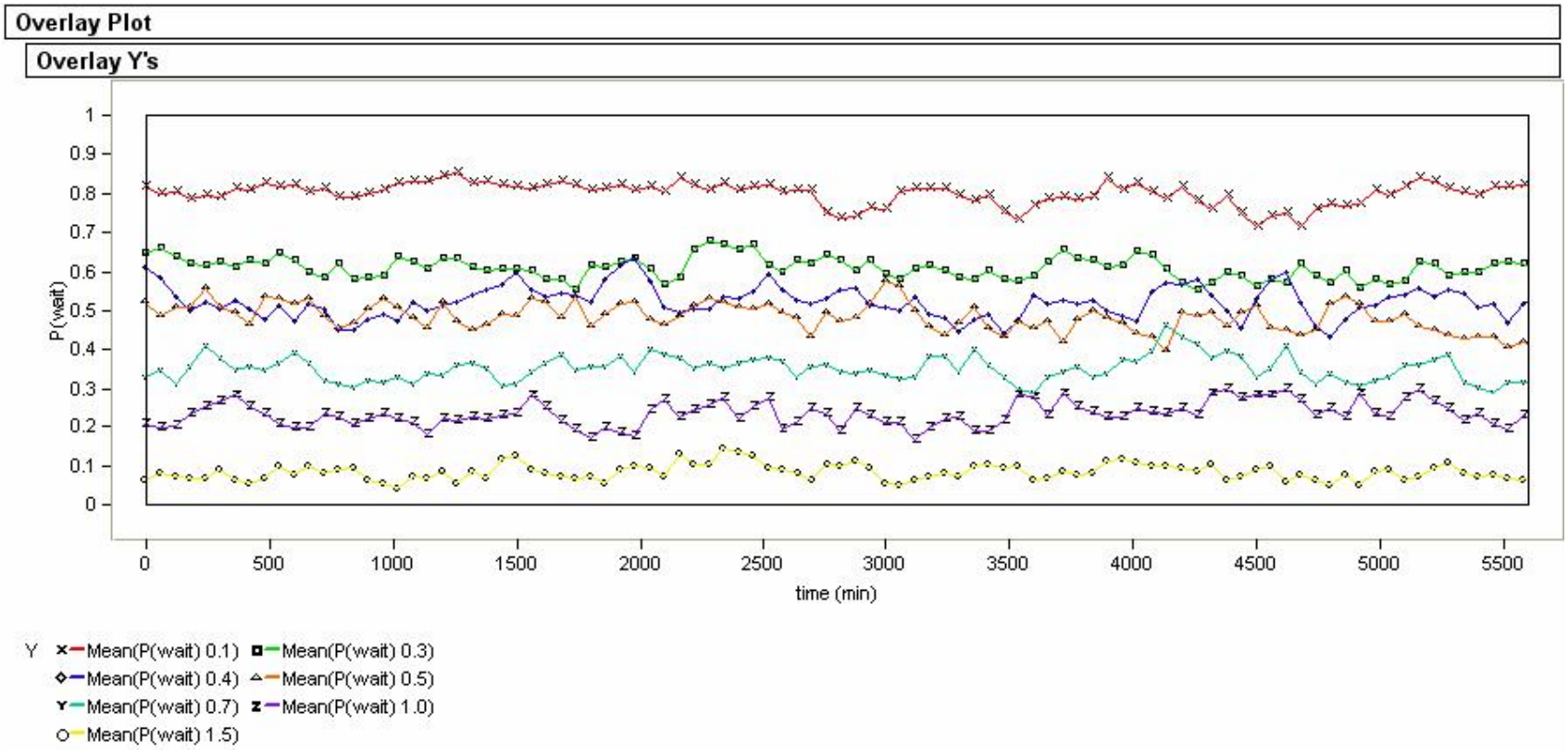
where $\bar{\lambda}$ is the average arrival rate, α is the relative amplitude ($0 < \alpha < 1$), and ψ is the period length.

Arrival rate, Offered load, and Staffing



Case Study: Sinusoidal Arrival Rate

Simulation results of P(wait) for various β values



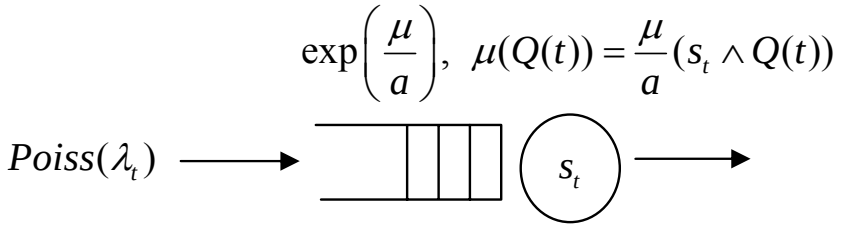
The performance measure is stable!

Are Cycles Significant? When?

Compare to concatenated services:

Define a = average number of re-entrances to service ($a = 1/(1-p)$).

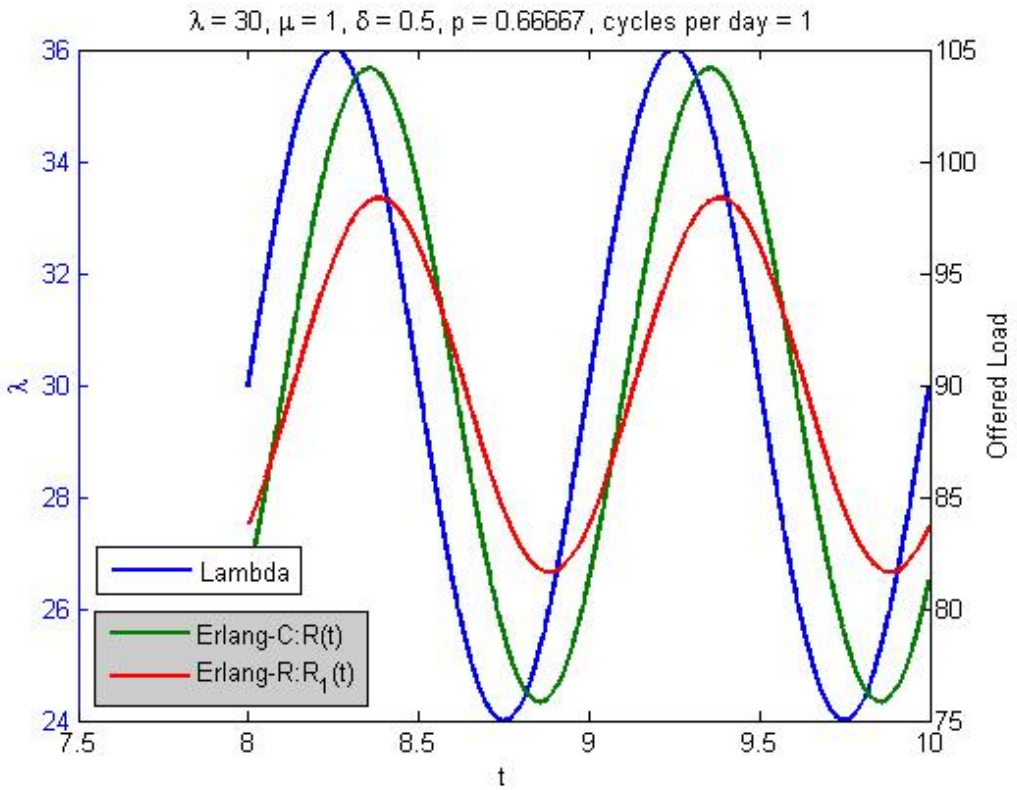
- ***Multi-service* Erlang-C:** Sum up all service times of each customer (RCCP)



- **When $\lambda(t) \equiv \lambda$, all have the same steady state!** Hence, all measurements (such as $P(W > 0)$) are identical, though Erlang-R provides information also about the *content* state.

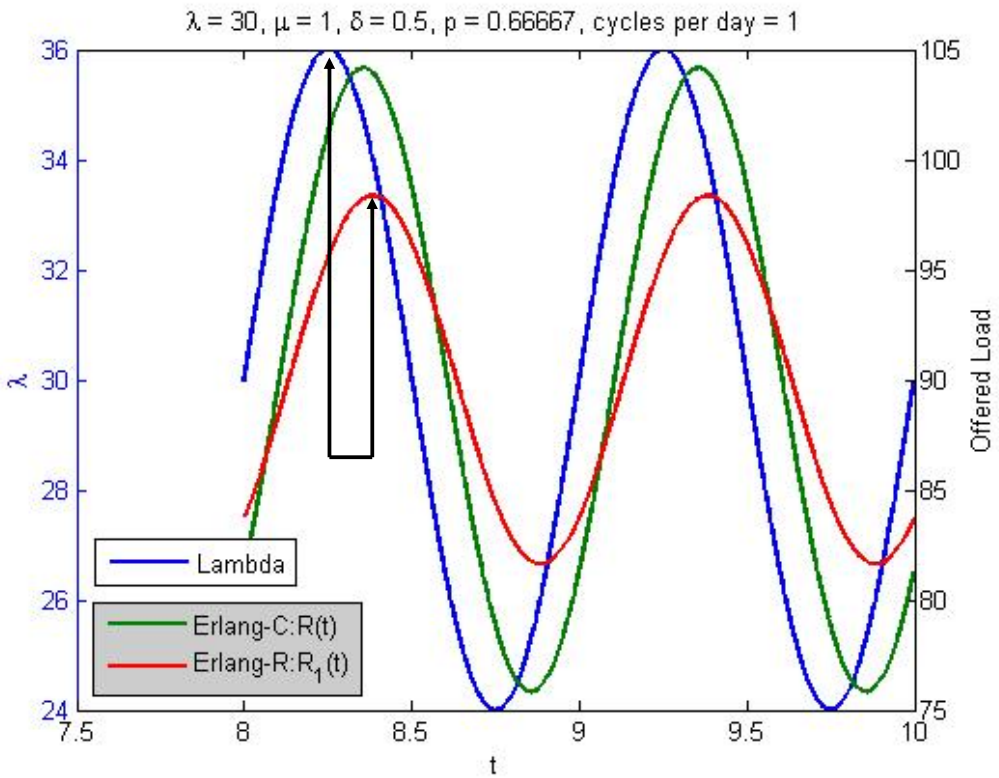
When Classic Model Fails? Time-Varying Systems

Erlang-C under- or over-estimates the offered load.



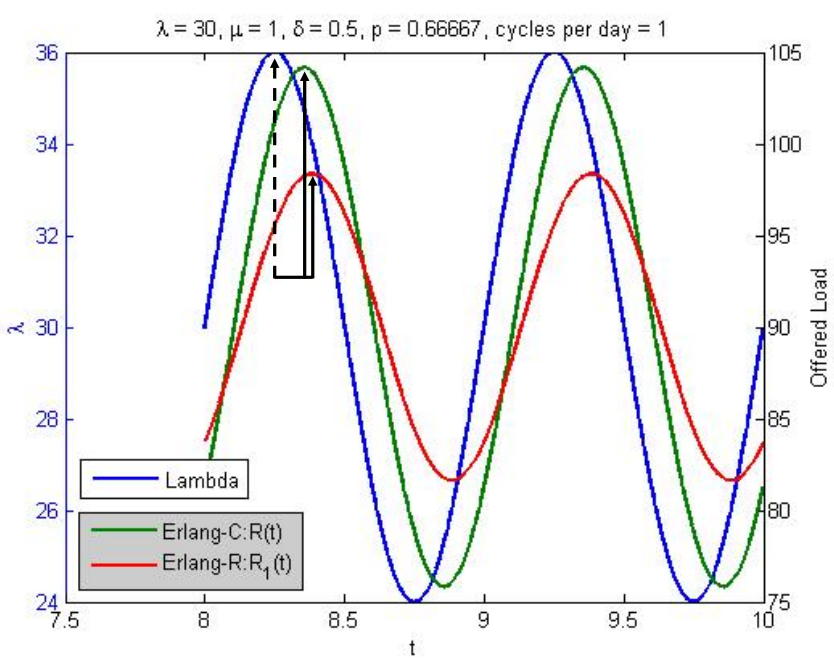
When Classic Model Fails? Time-Varying Systems

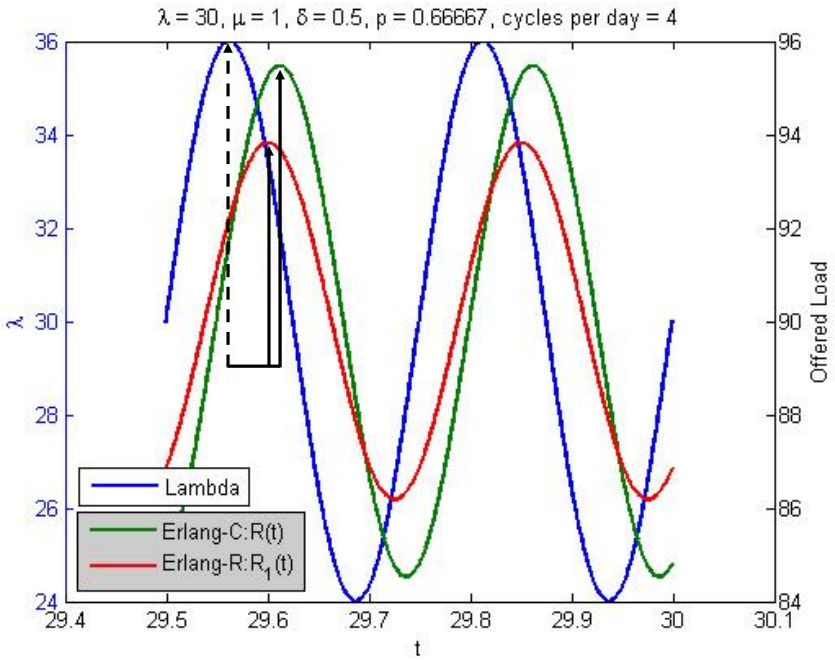
Well known that there is time-lag between peak arrival rate and offered load.



When Classic Model Fails? Time-Varying Systems

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





Why Erlang-C Does Not Fit Re-entrant Systems?

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

Multi-service Erlang-C's offered load: $S_1 \leftrightarrow aS_1$;

$$R(t) = E\left[\lambda(t - aS_1)\right] E\left[aS_1\right]$$

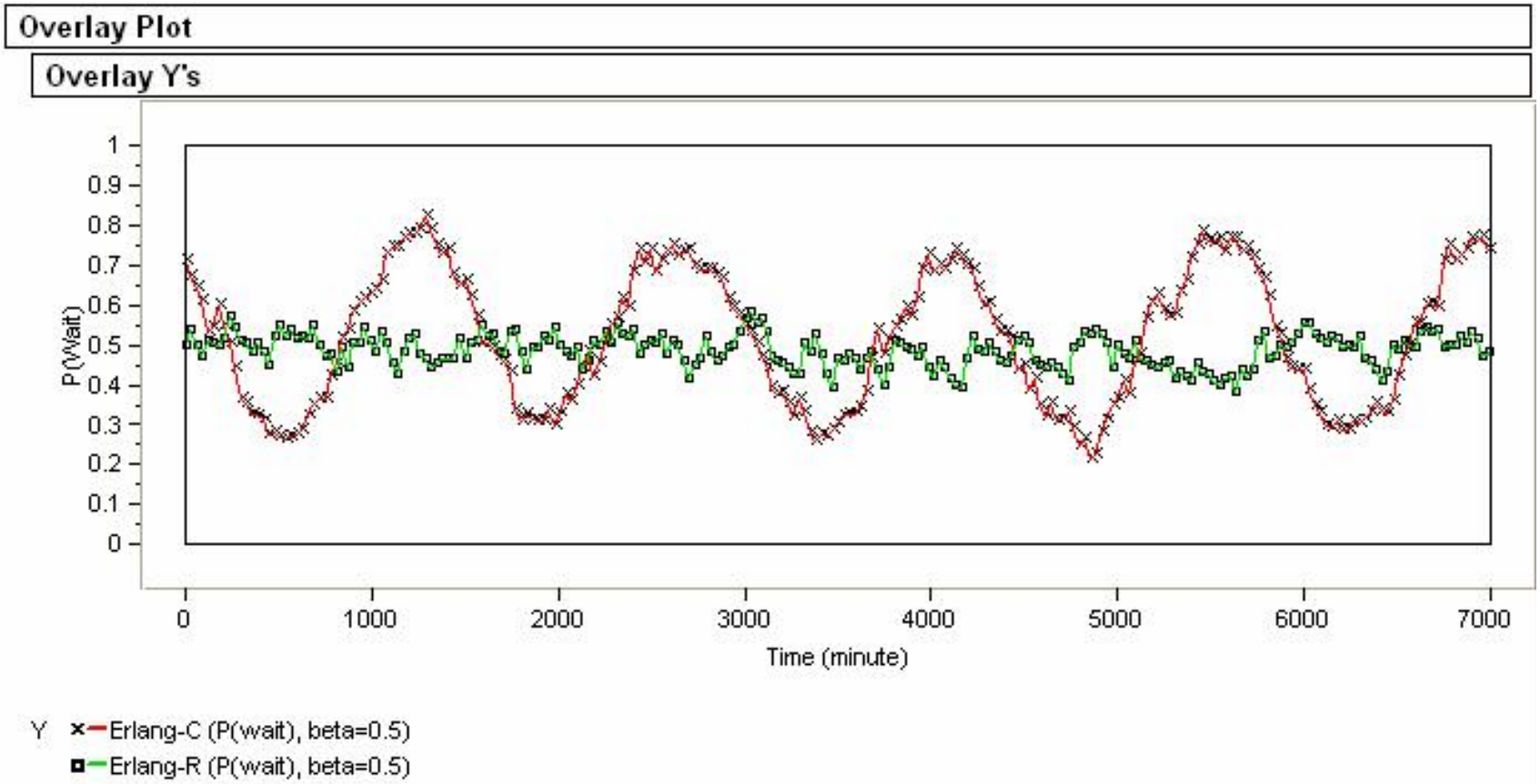
Erlang-R's offered load:

$$R_1(t) = \ldots = \sum_i p^i E[S_1] E\left[\lambda(t - (i+1)S_1 - iS_2)\right]$$

Nevertheless, Can We Use Erlang-C?

From the previous case study - sinusoidal arrival rate.

Simulation results of P(wait): Erlang-C vs. Erlang-R



Using Erlang-C's R(t), does not stabilize systems' performance.

Thank You