Time Dependent Queuing

Mark S. Daskin
Department of IE/MS, Northwestern University
Evanston, IL 60208

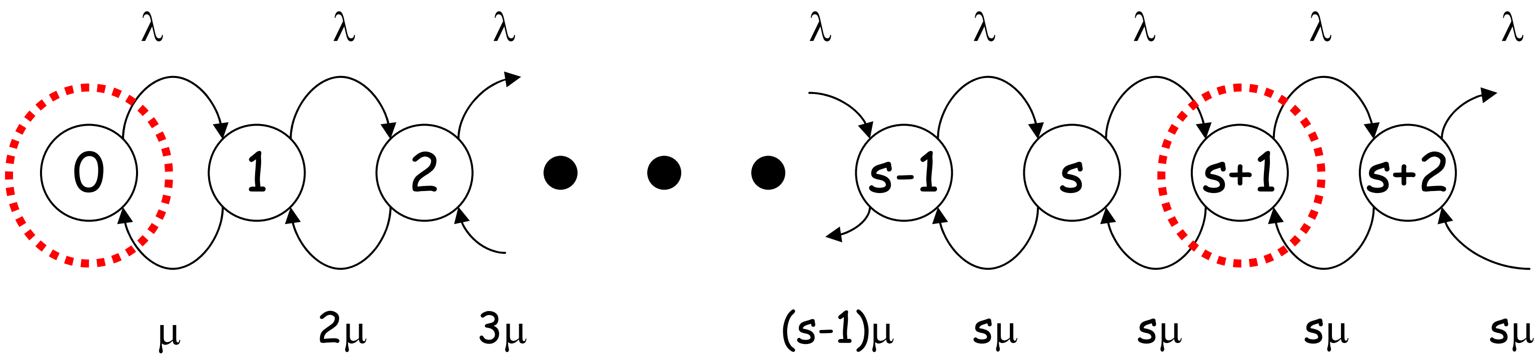
Spring, 2006

Outline

- Will look at M/M/s system
- Numerically integration of Chapman-Kolmogorov equations
- Introduction to Time Dependent Queue Analyzer

Chapman Kolmogorov Equations – M/M/s queue

State Transition Diagram



$$\frac{d\,P_0(t)}{dt} = \mu_1(t)P_1(t) - \lambda_0(t)P_0(t)$$

$$\frac{d\,P_n(t)}{dt} = \hat{\mu}_{n+1}(t)P_{n+1}(t) + \lambda_{n-1}(t)P_{n-1}(t) - \left[\hat{\mu}_n(t) + \lambda_n(t)\right]P_n(t) \qquad n = 1,\ldots$$

$$\hat{\mu}_n(t) = s_n(t)\mu_n(t)$$

First-order difference equations

$$\frac{d\,P_0(t)}{dt} = \mu_1(t)P_1(t) - \lambda_1(t)P_0(t)$$

$$\frac{P_0(t+\Delta t) - P_0(t)}{\Delta t} = \mu_1(t)P_1(t) - \lambda_1(t)P_0(t)$$

$$\boxed{P_0(t+\Delta t) = P_0(t) + \Delta t\left\{\mu_1(t)P_1(t) - \lambda_1(t)P_0(t)\right\}}$$

$$\frac{d\,P_n(t)}{dt} = \hat{\mu}_{n+1}(t)P_{n+1}(t) + \lambda_{n-1}(t)P_{n-1}(t) - \left[\hat{\mu}_n(t) + \lambda_n(t)\right]P_n(t) \qquad n = 1,\ldots$$

$$\frac{P_n(t+\Delta t) - P_n(t)}{\Delta t} = \hat{\mu}_{n+1}(t)P_{n+1}(t) + \lambda_{n-1}(t)P_{n-1}(t) - \left[\hat{\mu}_n(t) + \lambda_n(t)\right]P_n(t) \qquad n = 1,\ldots$$

$$\boxed{P_n(t+\Delta t) = P_n(t) + \Delta t\left\{\hat{\mu}_{n+1}(t)P_{n+1}(t) + \lambda_{n-1}(t)P_{n-1}(t) - \left[\hat{\mu}_n(t) + \lambda_n(t)\right]P_n(t)\right\}} \qquad n = 1,\ldots$$

You can get first order difference equations as shown on the next slide

First-order difference equations

$$P_0(t+\Delta t) = P_0(t) + \Delta t\left\{\mu_1(t)P_1(t) - \lambda_1(t)P_0(t)\right\}$$

$$P_n(t+\Delta t) = P_n(t) + \Delta t\left\{\hat{\mu}_{n+1}(t)P_{n+1}(t) + \lambda_{n-1}(t)P_{n-1}(t) - \left[\hat{\mu}_n(t) + \lambda_n(t)\right]P_n(t)\right\} \qquad n = 1,\ldots$$

Given some initial estimate of the state probabilities at time *t*,

we can use these equations to estimate the state probabilities at some time *t+Δt*

and so on….

Practical implementation

- **Make state space finite (max state=N)**
  - Adjust equation for P_N accordingly
- **Divide the day into small time slices**
  - E.g., use Δt=60 seconds or less.
- **Begin with steady state estimate of probabilities**
  - Increment s as needed to get steady state during any time slice that has $s\mu<\lambda$.

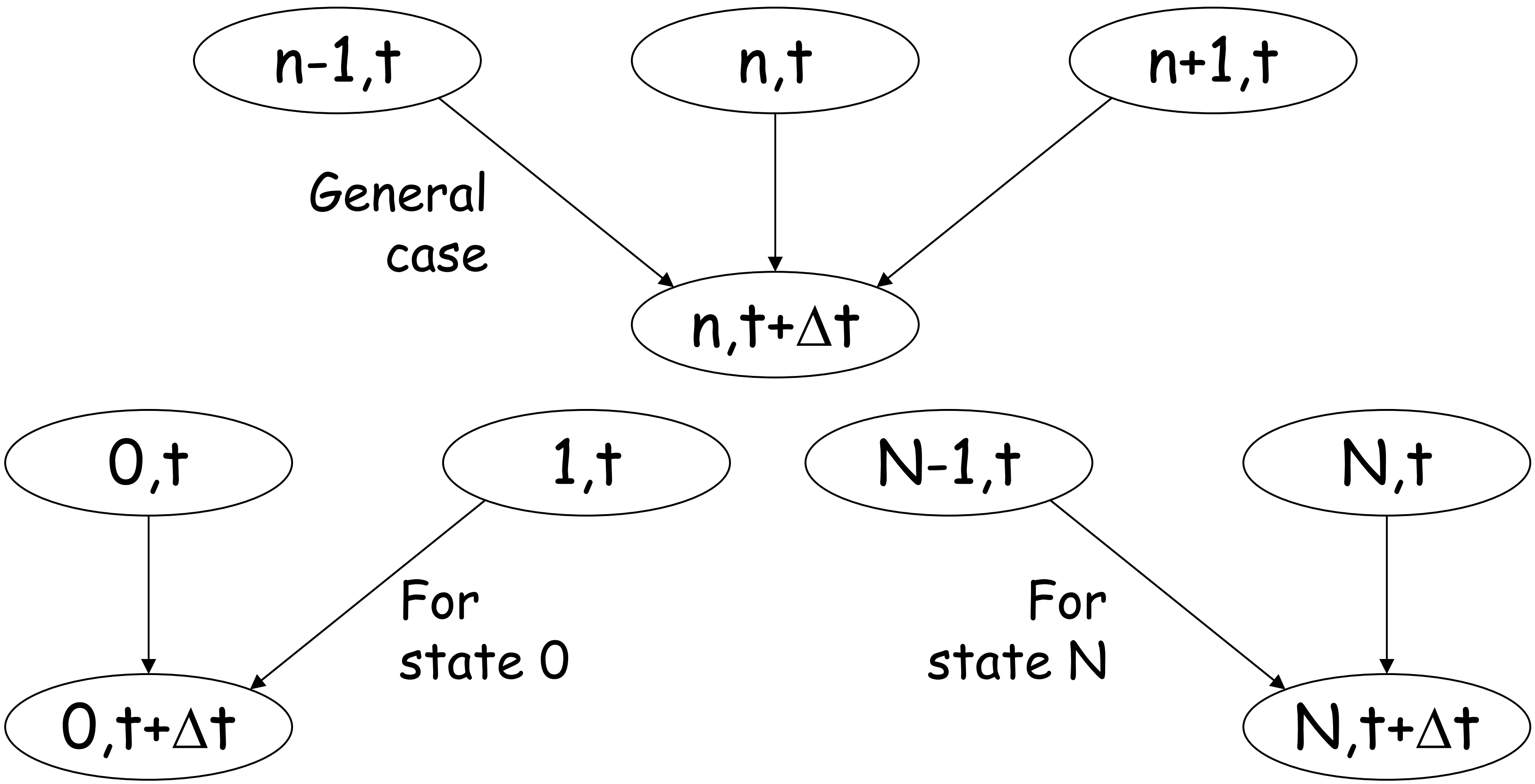
Practical implementation

- **Use fourth-order Runge-Kutta to step between time slices and not first-order Euler as shown above**
  - If any $P_n(t)$ becomes negative, set it to 0
  - Renormalize all state probabilities at each time period
  - Compute largest % change in probabilities (for prob>0.0001 for example)

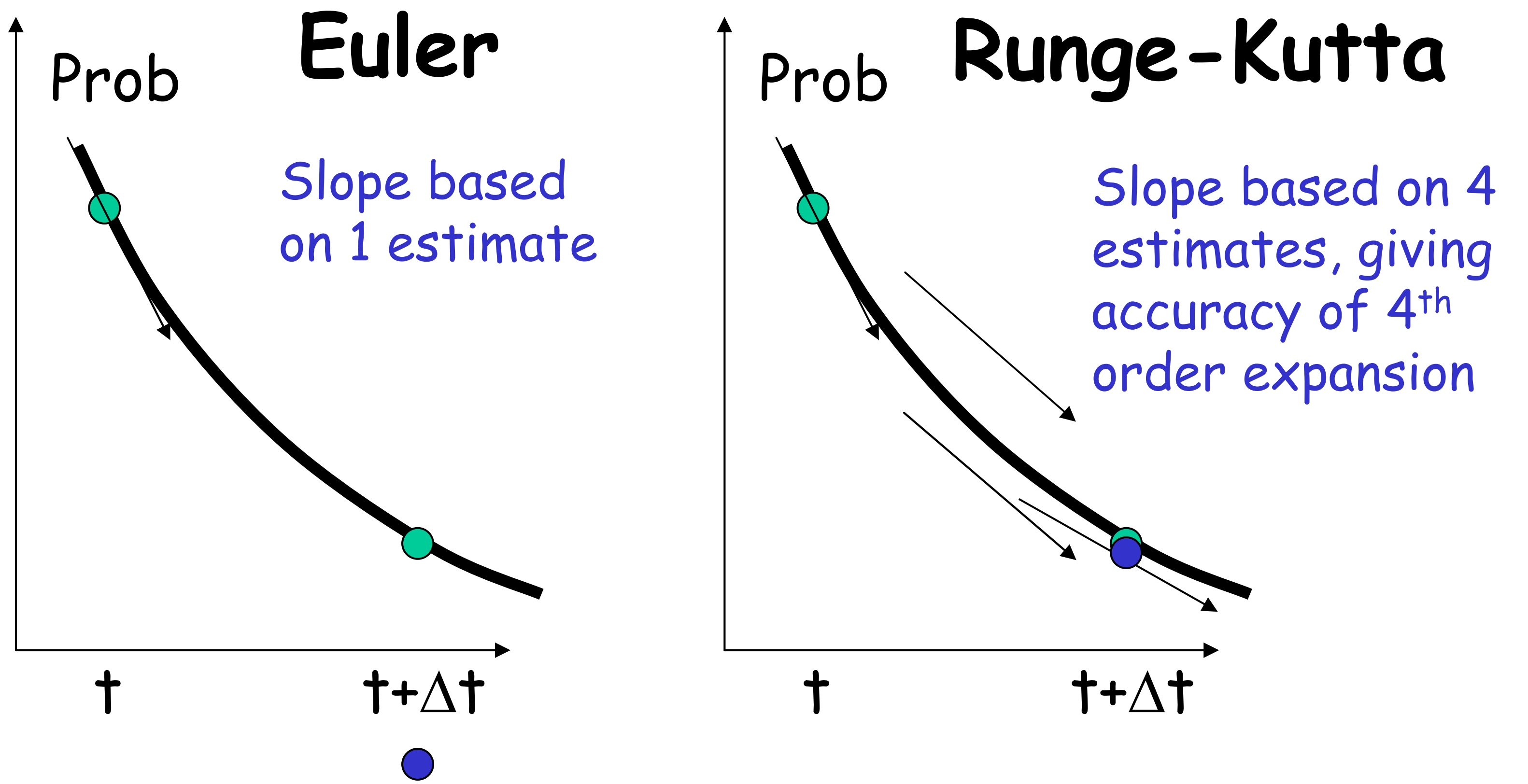
Practical implementation

- If any state probabilities go negative, start process over with smaller Δt
- If largest % change too big, cycle through probabilities again. Repeat as needed.

Which state probabilities influence $P_n(t+\Delta t)$



Euler vs 4^{th} Order Runge-Kutta



Euler vs 4th Order Runge-Kutta

Euler $$P_n(t+\Delta t) = P_n(t) + f_n[P_n(t),t]\Delta t$$

Runge-Kutta $$P_n(t+\Delta t) = P_n(t) + [k_1(t) + 2k_2(t) + 2k_3(t) + k_4(t)]/3$$

$$k_1(t) = f_n[P_n(t),t]\Delta t/2$$

$$k_2(t) = f_n[P_n(t) + k_1(t), t + 0.5\Delta t]\Delta t/2$$

$$k_3(t) = f_n[P_n(t) + k_2(t), t + 0.5\Delta t]\Delta t/2$$

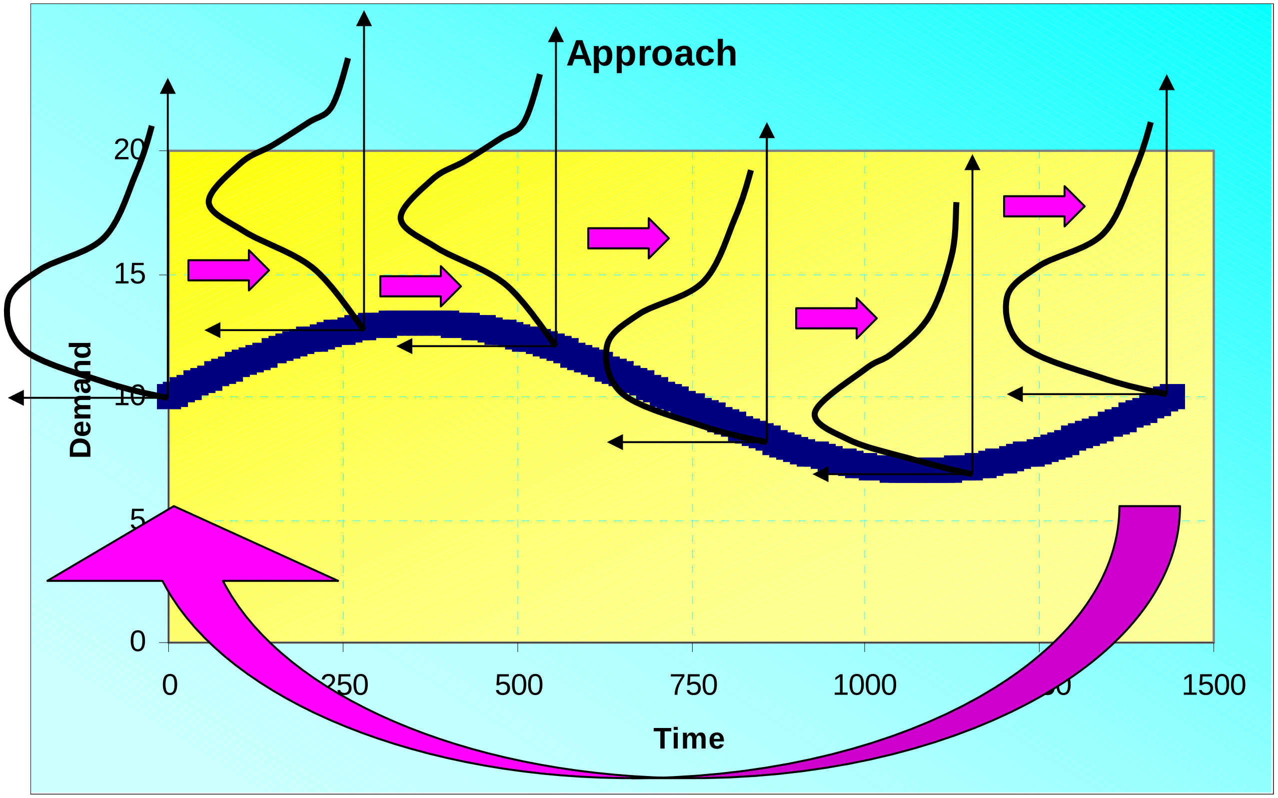
$$k_4(t) = f_n[P_n(t) + 2k_3(t), t + \Delta t]\Delta t/2$$

$$f_0(t) = \mu_1(t)P_1(t) - \lambda_1(t)P_0(t)$$

$$f_n(t) = \hat{\mu}_{n+1}(t)P_{n+1}(t) + \lambda_{n-1}(t)P_{n-1}(t) - [\hat{\mu}_n(t) + \lambda_n(t)]P_n(t) \qquad n = 1, \ldots N-1$$

$$f_N(t) = \lambda_{N-1}(t)P_{N-1}(t) - \hat{\mu}_N(t)P_N(t)$$

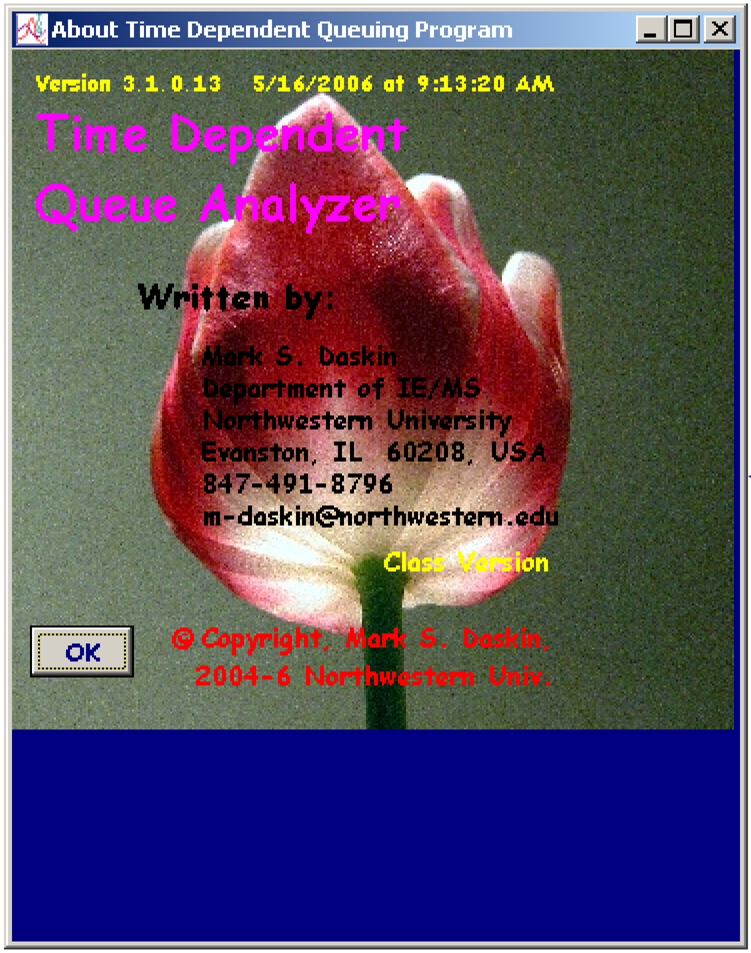
Graphical representation of solution procedure



Evolve PMFs over time

Cycle back as needed

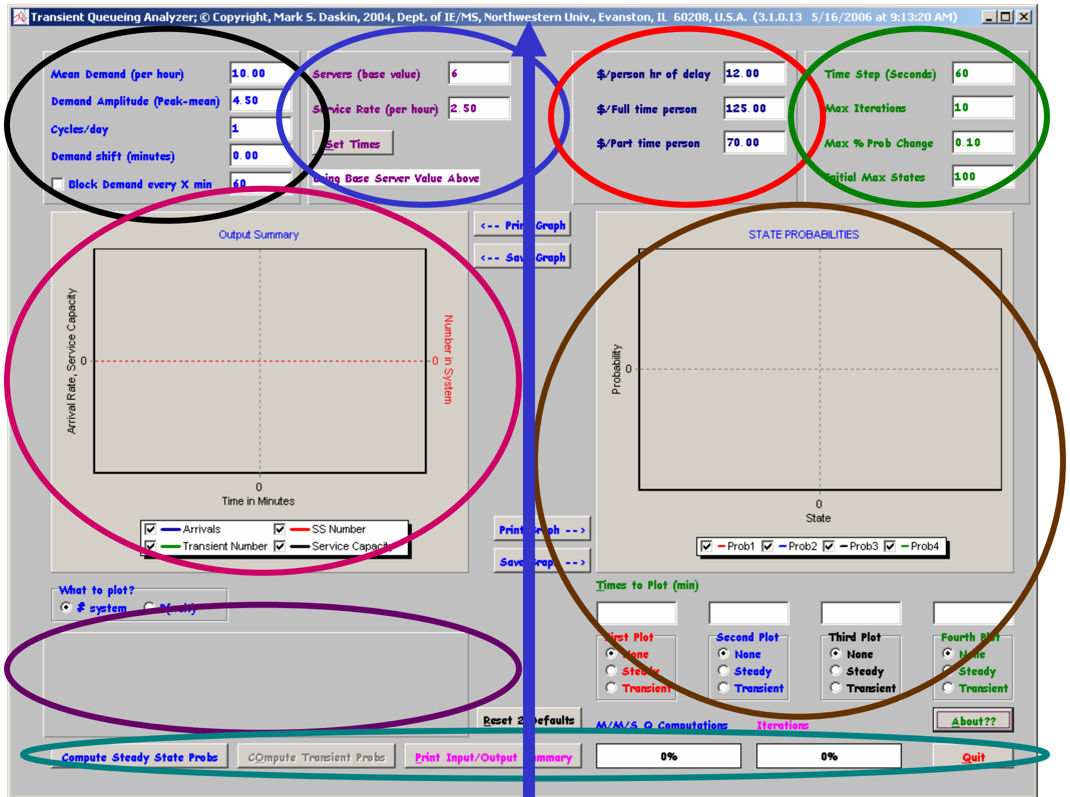
Time Dependent Queue Analyzer



Be sure your copy looks like this. Earlier versions had an error in the code.

You should see *Class Version* underneath my e-mail address.

Major blocks of inputs/outputs



Be sure your copy has the version number in the top line. Earlier versions of the code contain an error.

Demand info

Service info

Cost inputs

Computational info

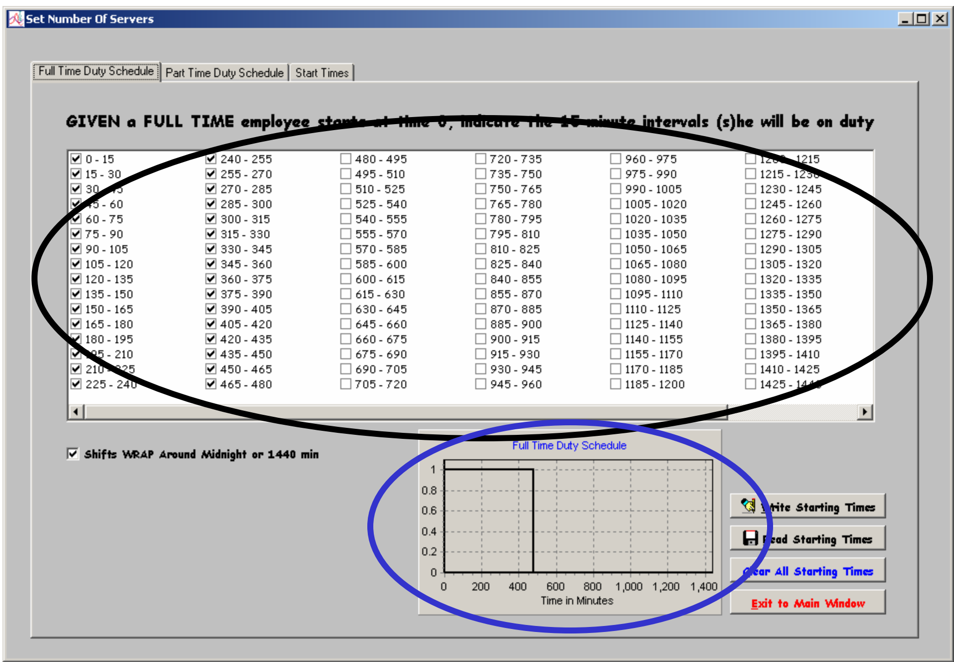
Primary graph output

State prob graphs

Summary output

Control Buttons

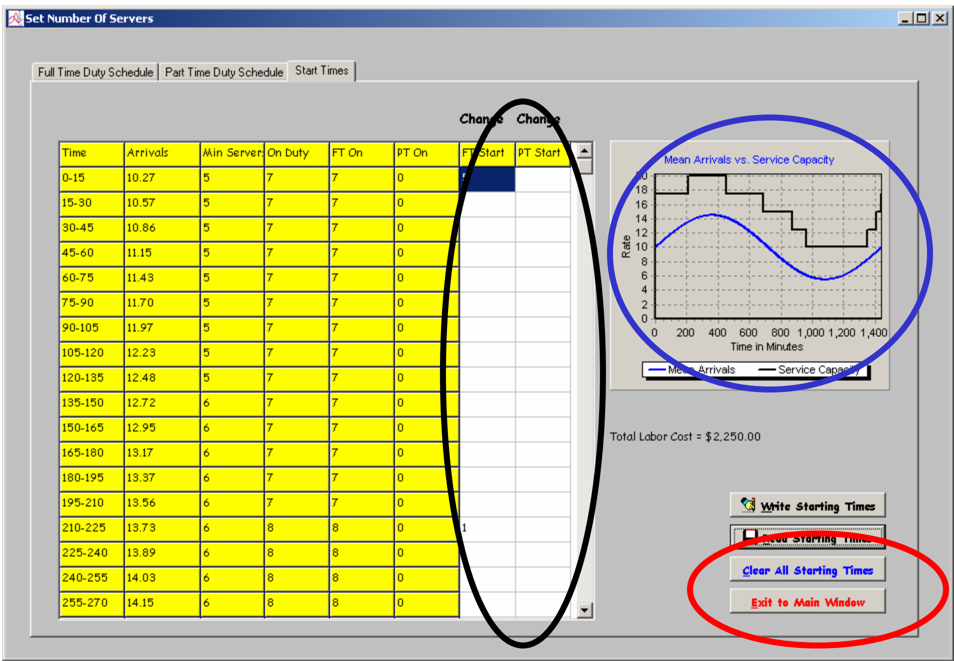
Set times form



Duty times

Graph of duty times

Start times page

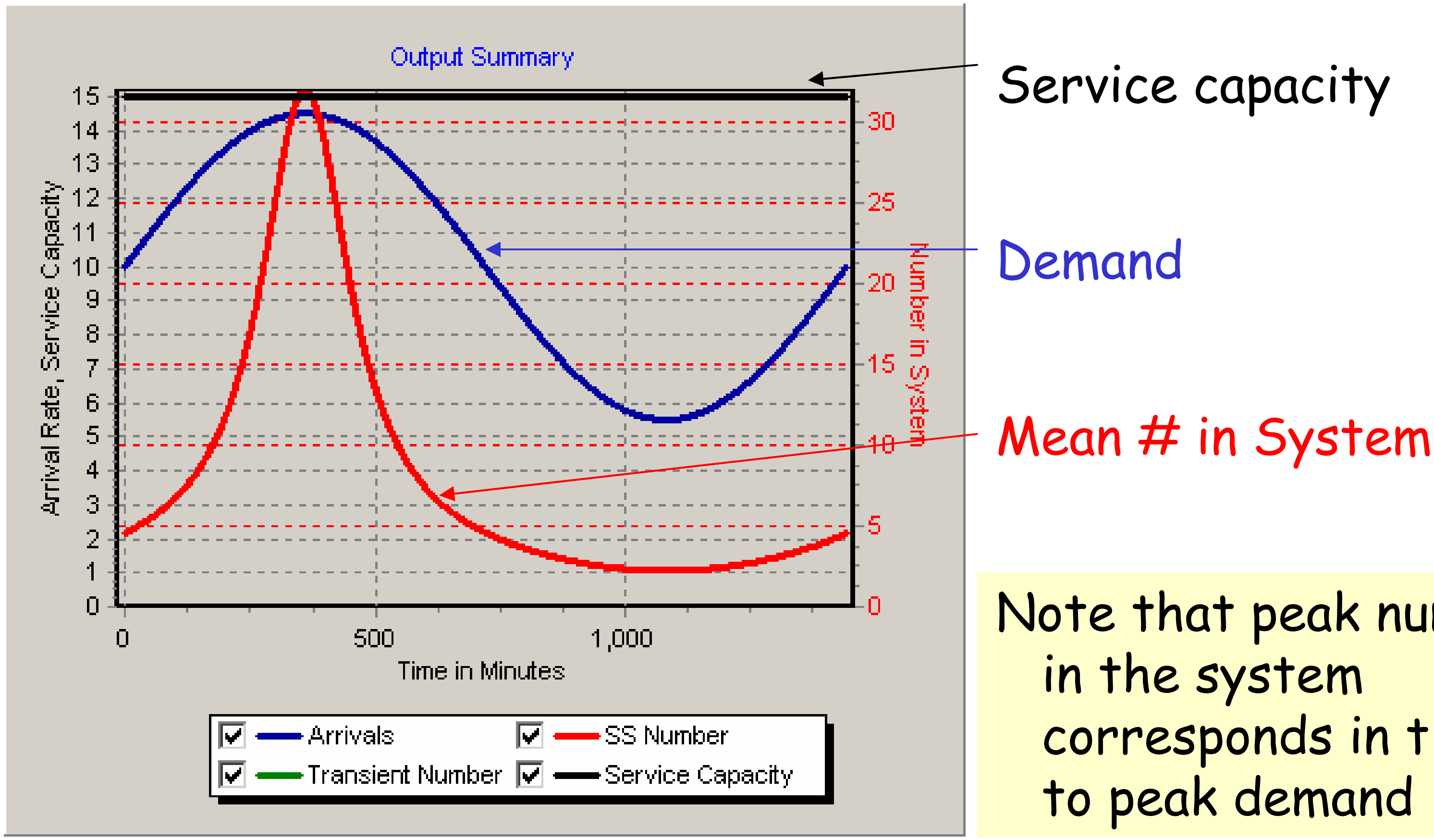


Set starting times for full time and part-time employees

Graph of demand and service capacity

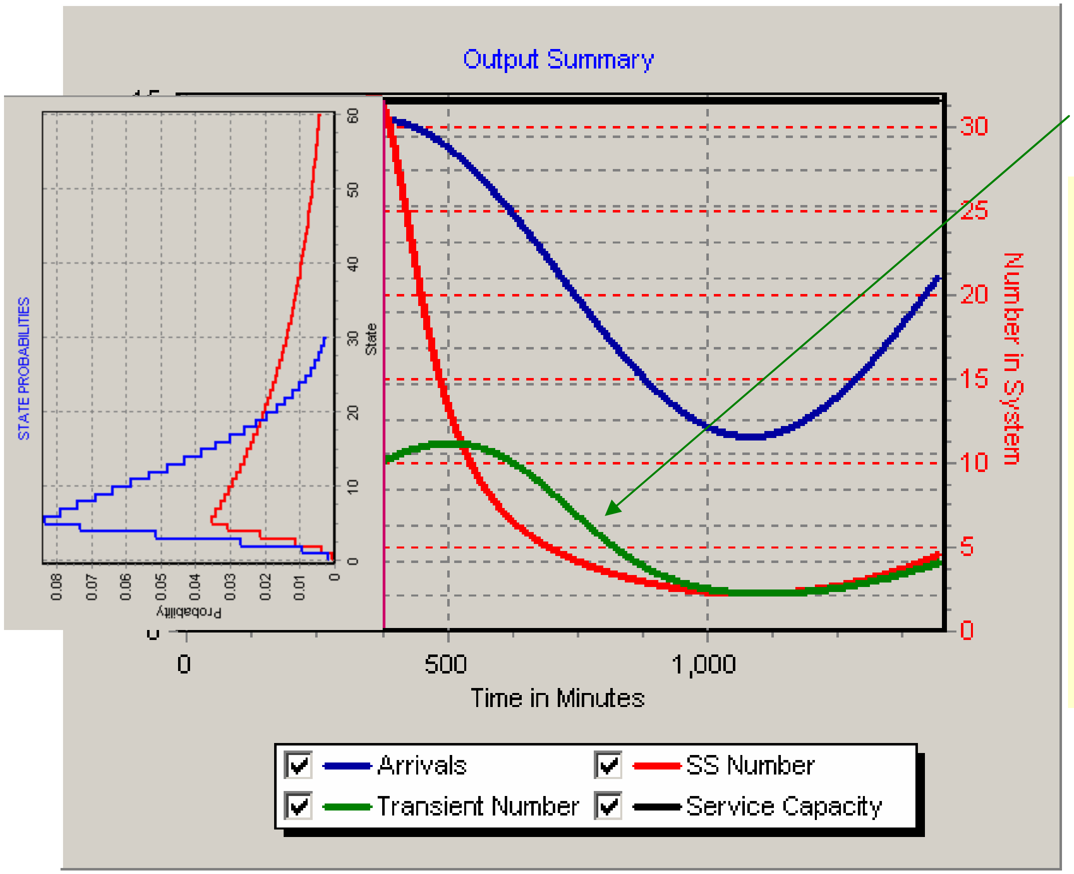
Click to return to main menu

Steady state base case output



Note that peak number in the system corresponds in time to peak demand

Time Dependent Case

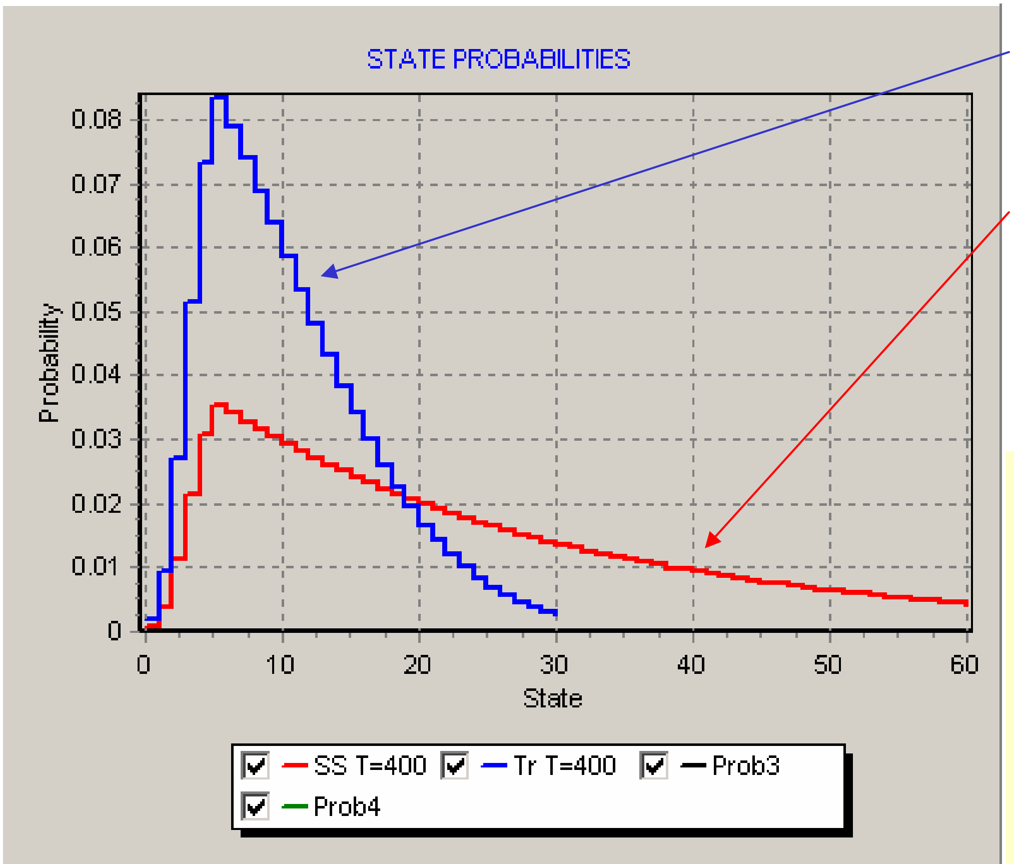


Time Dep. # in system

Note that peak number in the system under time dependent conditions is shifted to the right of the peak demand and is lower than then steady-state peak.

Will look at PMF at t=400

Comparison of PMFs at t=400

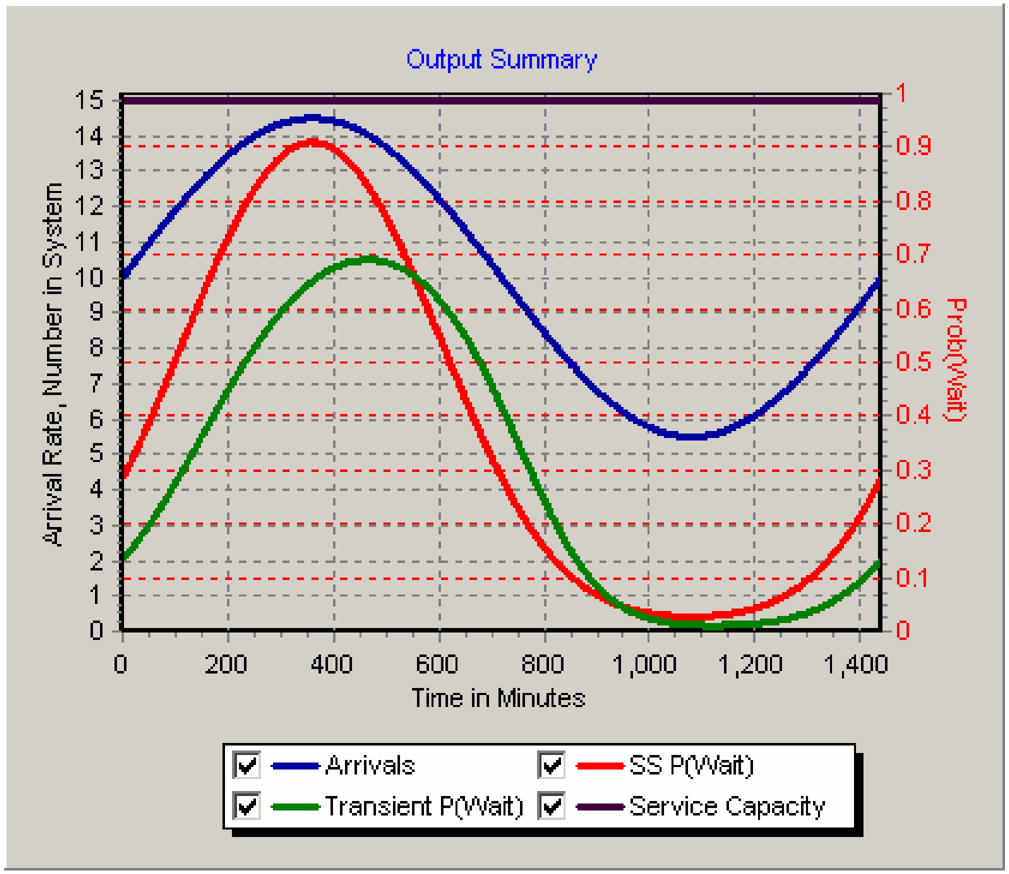


Time Dependent

Steady-state

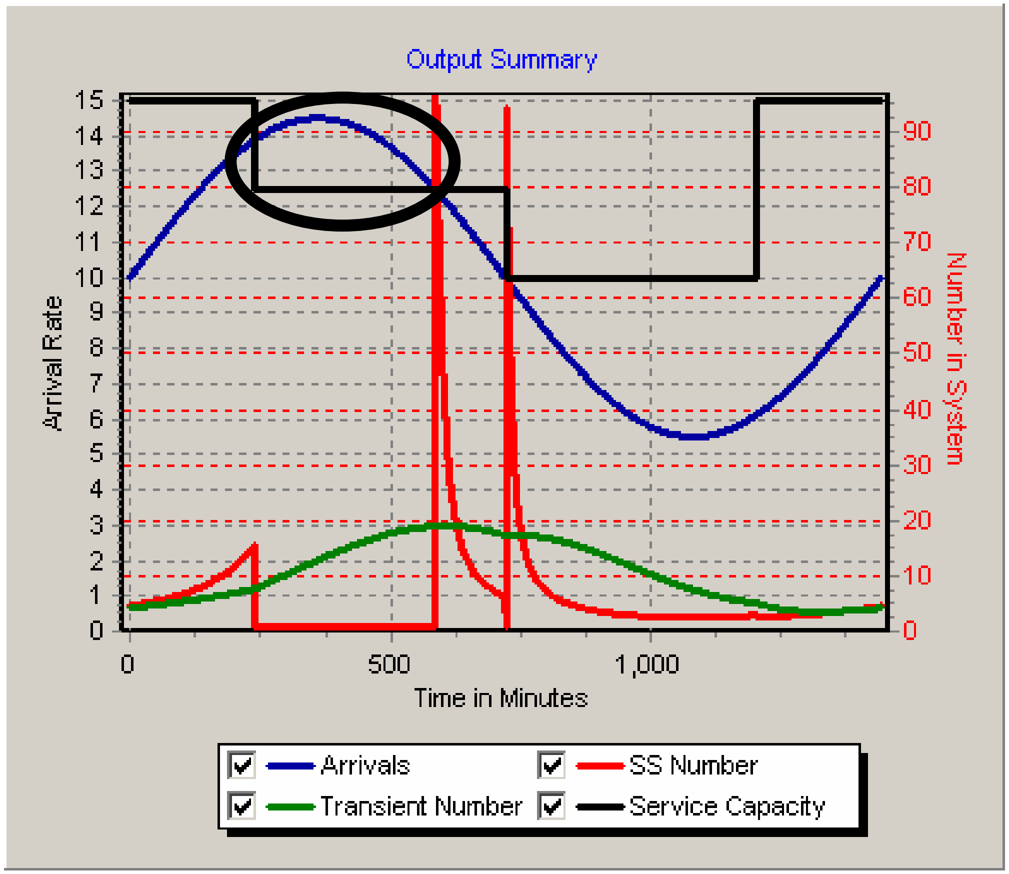
Note that time dependent probabilities are higher at low end reflecting the smaller mean at t=400.

P(wait)



Note that peak P(wait) in the system under time dependent conditions is shifted to the right of the peak demand and is lower than then steady-state peak.

More complex behavior



Note that the program does not require steady state conditions in each time slice.

Summary

- Time dependent queuing analysis is important
- Peaks are shifted to the right of steady state peaks
- Does not require steady-state conditions in all periods.