The Offered-Load in Fork-Join Networks

Applications to Staffing of Emergency Departments

Itamar Zaied
Jointly with H.Kaspi and A.Mandelbaum

June 24, 2012

Introduction - The Offered Load

The Offered Load: The Stationary Case

hours of work (= service) that arrive per hour.

Example:
$\lambda = 20$ patients/hour; $E[S] = 0.5 hours$. Offered-Load $R = 20 \cdot 0.5 = 10$ hour of work per hour.

The Offered Load of an $M_t/GI/N_t$ queue

For the $M_t/GI/N_t$ queue, the *offered load* $R = \{R(t),\ t \geq 0\}$ is given by the function $R(t) = E[L(t)]$, where $L(t)$ is the number of customers/patients (=number of busy servers) at time t, in the corresponding $M_t/GI/\infty$ **queue.**

Introduction - The Offered Load

The Offered Load: The Stationary Case

hours of work (= service) that arrive per hour.

Example:
$\lambda = 20$ patients/hour; $E[S] = 0.5 hours$. Offered-Load $R = 20 \cdot 0.5 = 10$ hour of work per hour.

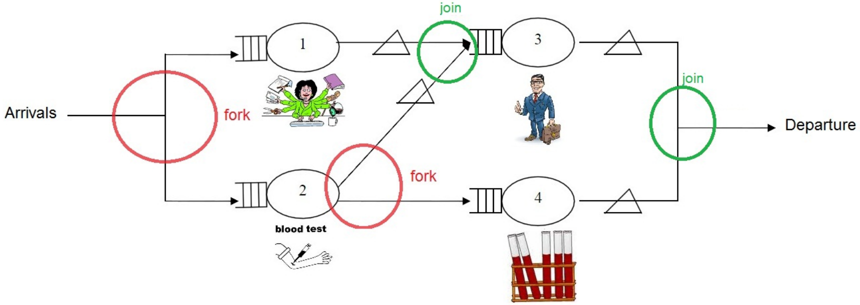
The Offered Load of an $M_t/GI/N_t$ queue

For the $M_t/GI/N_t$ queue, the *offered load* $R = \{R(t),\ t \geq 0\}$ is given by the function $R(t) = E[L(t)]$, where $L(t)$ is the number of customers/patients (=number of busy servers) at time t, in the corresponding $M_t/GI/\infty$ **queue.**

- Offered-Load has been proved to be the skeleton for staffing of time-varying systems.

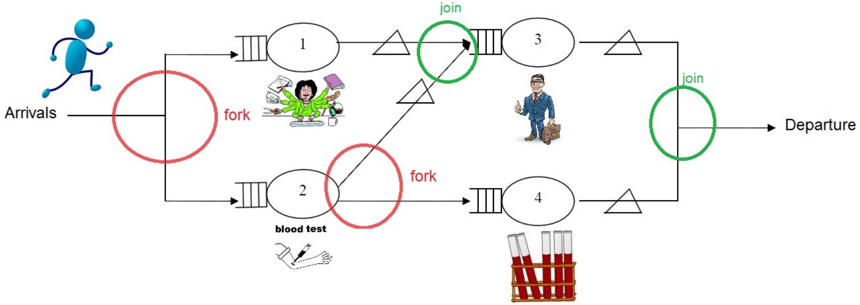
Fork-Join Networks

A fork-join network consists of service stations, which serve arriving customers simultaneously and sequentially according to pre-designed deterministic precedence constraints.



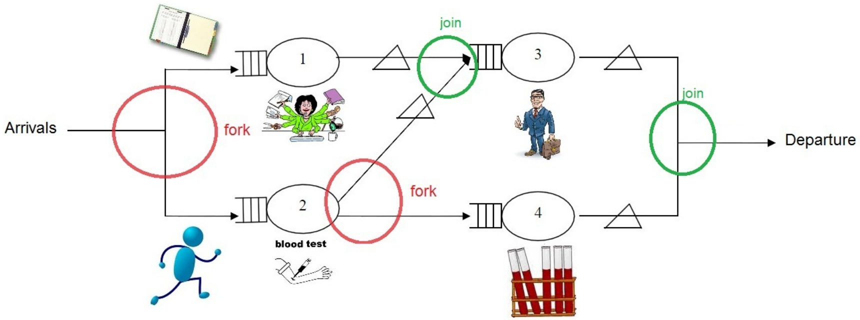
Fork-Join Networks

A fork-join network consists of service stations, which serve arriving customers simultaneously and sequentially according to pre-designed deterministic precedence constraints.



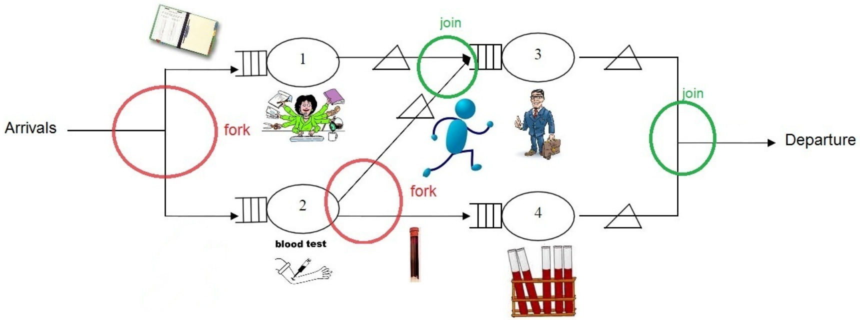
Fork-Join Networks

A fork-join network consists of service stations, which serve arriving customers simultaneously and sequentially according to pre-designed deterministic precedence constraints.



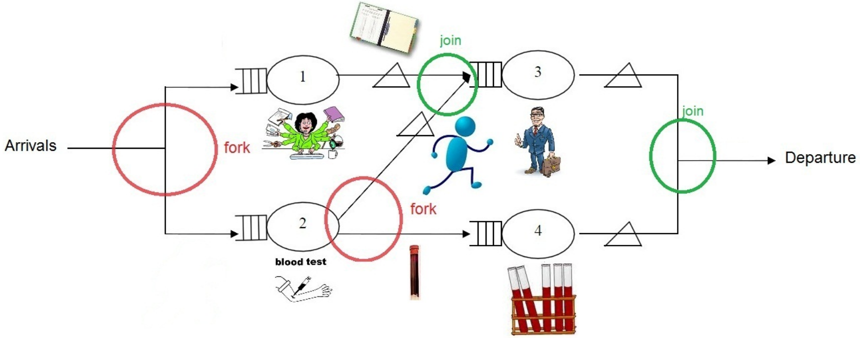
Fork-Join Networks

A fork-join network consists of service stations, which serve arriving customers simultaneously and sequentially according to pre-designed deterministic precedence constraints.



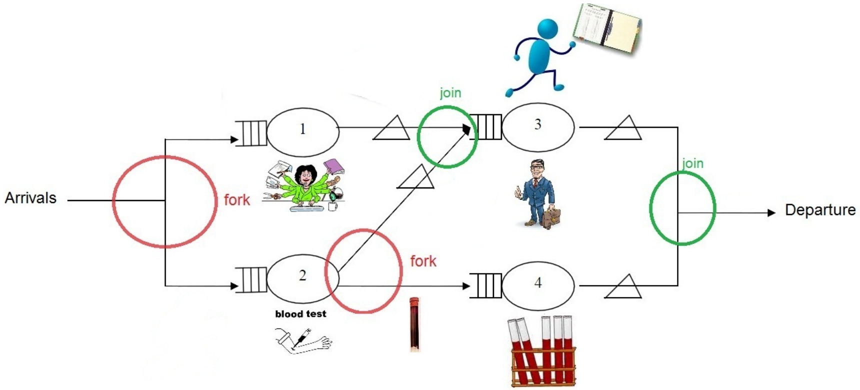
Fork-Join Networks

A fork-join network consists of service stations, which serve arriving customers simultaneously and sequentially according to pre-designed deterministic precedence constraints.



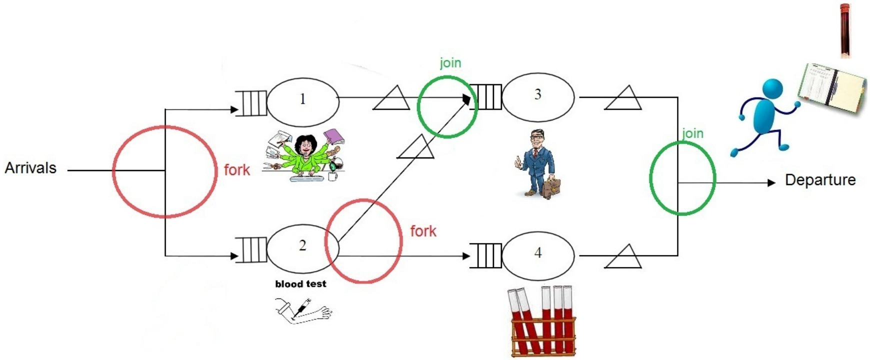
Fork-Join Networks

A fork-join network consists of service stations, which serve arriving customers simultaneously and sequentially according to pre-designed deterministic precedence constraints.

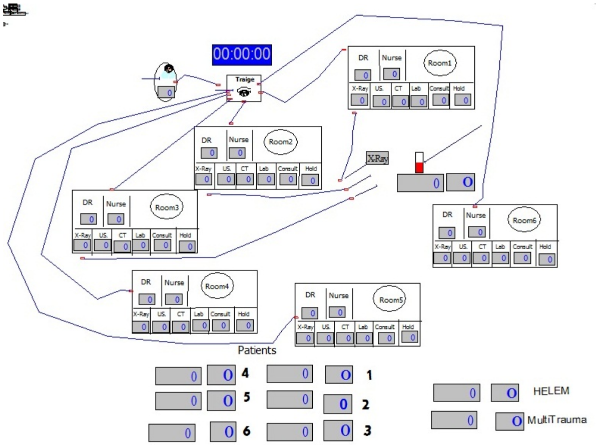


Fork-Join Networks

A fork-join network consists of service stations, which serve arriving customers simultaneously and sequentially according to pre-designed deterministic precedence constraints.



In his MSc thesis, Yariv Marmur developed a generic ED simulation, using Rockwell's software "Arena".



Offered-Load calculation using the ED simulator:

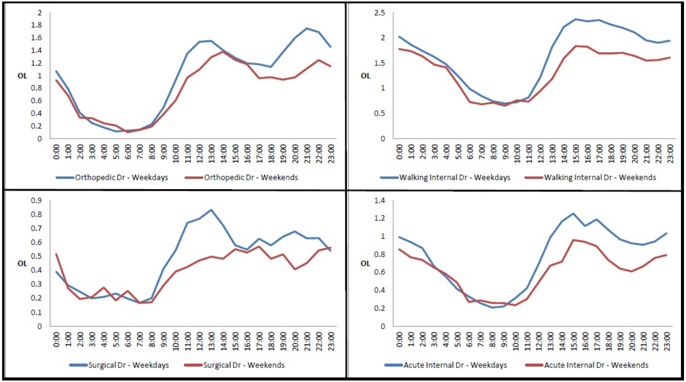
Offered-Load calculation using the ED simulator:

- In analytical models, the Offered-Load is the number of patients in the corresponding infinite-server network.

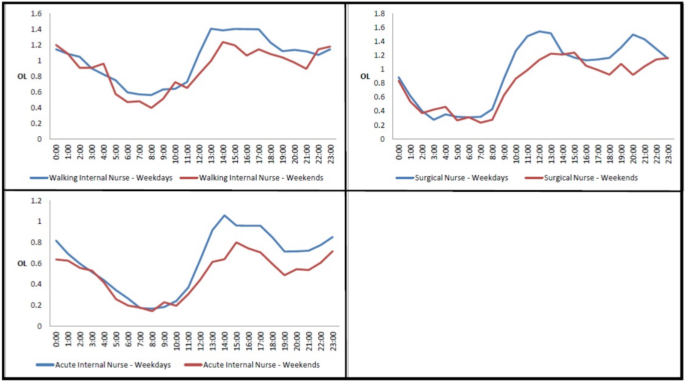
Offered-Load calculation using the ED simulator:

- In analytical models, the Offered-Load is the number of patients in the corresponding infinite-server network.
- We thus calculate the Offered-Load using the ED simulator, where we set the number of resources to be ∞ (at all the resources or only the resources under consideration).

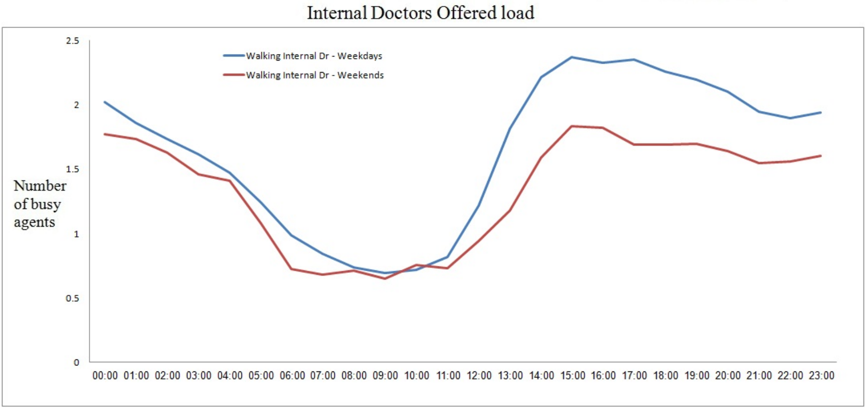
The Offered-Load of Resources: Doctors



The Offered-Load of Resources: Nurses



Zoom in on Internal Doctors



Staffing

Using the square-root staffing rule:

$$N(t) = round(R(t) + \beta \cdot \sqrt{R(t)}) \ ,$$

where:

Staffing

Using the square-root staffing rule:

$$N(t) = round(R(t) + \beta \cdot \sqrt{R(t)}) \ ,$$

where:

- $N(t)$ are the number of agents at time t.

Staffing

Using the square-root staffing rule:

$$N(t) = round(R(t) + \beta \cdot \sqrt{R(t)}) \ ,$$

where:

- $N(t)$ are the number of agents at time t.
- β is "safety staffing".

Staffing

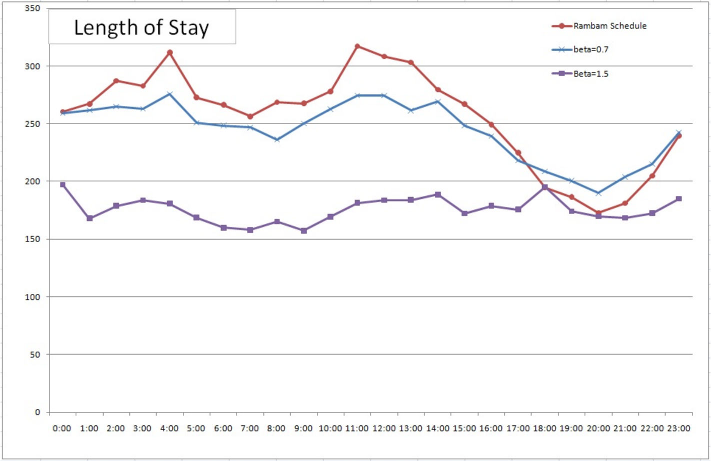
Using the square-root staffing rule:

$$N(t) = round(R(t) + \beta \cdot \sqrt{R(t)})\ ,$$

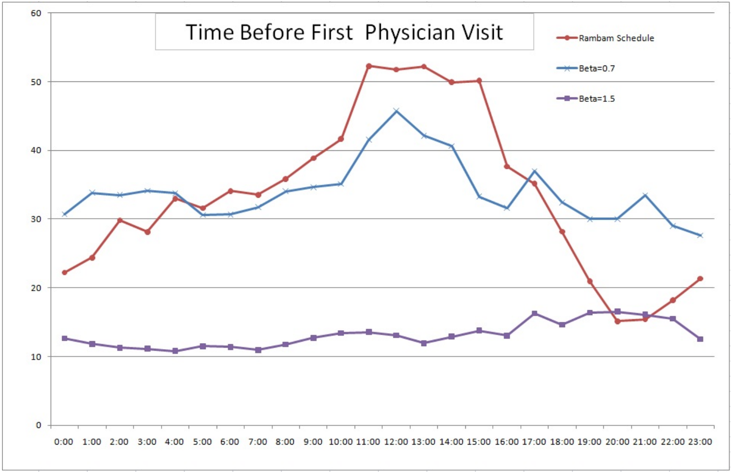
where:

- $N(t)$ are the number of agents at time t.
- β is "safety staffing".
- $round(x) = \left\{ \begin{smallmatrix} \lceil x \rceil & if & \lceil x \rceil - x \geq 0.5 \\ \lfloor x \rfloor & if & \lceil x \rceil - x < 0.5 \end{smallmatrix} \right.$.

Length of Stay

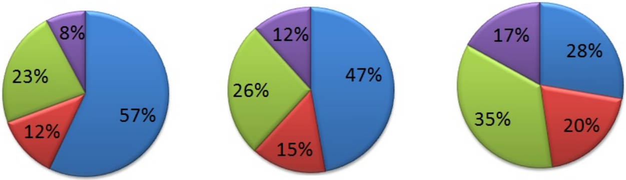


Time Up to First Physician Visit



Different β's vs Original Staffing

Anatomy of Sojourn Time

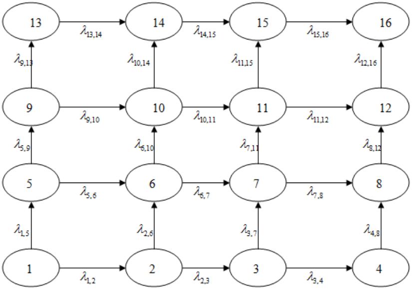


	Current Staffing	$\beta = 0.7$	$\beta = 1.5$
	LOS=7.1 hours	LOS=6 hours	LOS=4.6 hours
Daily Hours of Internal Dr:	56	63	86
Daily Hours of Internal Nurses:	48	41	68

- Queue Time — Waiting for a resource
- Service Time — Duration of Service
- Clinical Treatment Time — Waiting for a medicine to take effect, waiting for an X-Ray to develop,etc.
- Sync Time — Waiting for a "Partner"

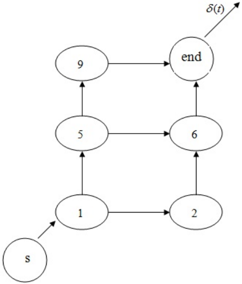
The Black Box Method

Consider a fork join network with V its set of nodes and A its set of arcs. To calculate the Offered Load of station i, one can define a new fork join network i^-, as follows:



The Black Box Method

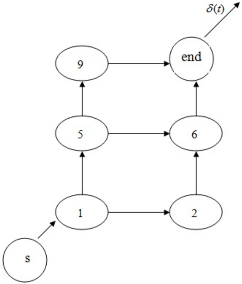
- Let V^- be all the nodes of V which have a directed path to station i.



Example: the network 10^-

The Black Box Method

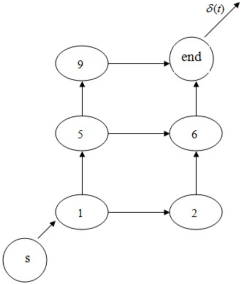
- Let V^- be all the nodes of V which have a directed path to station i.
- Let A^- be all the arcs in A which connect a node from V^- to a node from $V^- \cup i$.



Example: the network 10^-

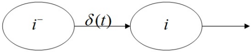
The Black Box Method

- Let V^- be all the nodes of V which have a directed path to station i.
- Let A^- be all the arcs in A which connect a node from V^- to a node from $V^- \cup i$.
- Let i^- be the fork-join network with V^- its set of nodes and A^- its set of arcs.



Example: the network 10^-

We now have a Tandem of 2 service stations, where queue i is the second queue.



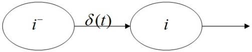
The Offered-Load at station i

The offered load at station i, is given by

$$R_i(t) = E(\lambda(t - T_i - S_i^e)) \cdot E(S_i), \quad t \geq 0,$$

where T_i is the total sojourn time of network i^- and S_i^e is the residual service time at station i.

We now have a Tandem of 2 service stations, where queue i is the second queue.



The Offered-Load at station i

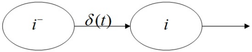
The offered load at station i, is given by

$$R_i(t) = E(\lambda(t - T_i - S_i^e)) \cdot E(S_i), \quad t \geq 0,$$

where T_i is the total sojourn time of network i^- and S_i^e is the residual service time at station i.

- Thus, finding the distribution of T_i is required for calculating the Offered-Load of station i.

We now have a Tandem of 2 service stations, where queue i is the second queue.



The Offered-Load at station i

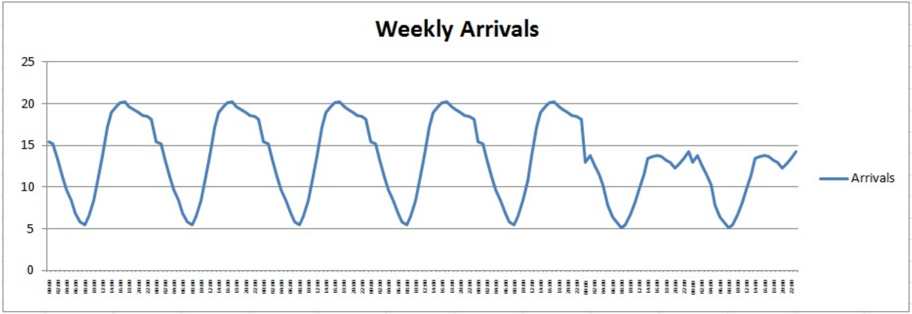
The offered load at station i, is given by

$$R_i(t) = E(\lambda(t - T_i - S_i^e)) \cdot E(S_i), \quad t \geq 0,$$

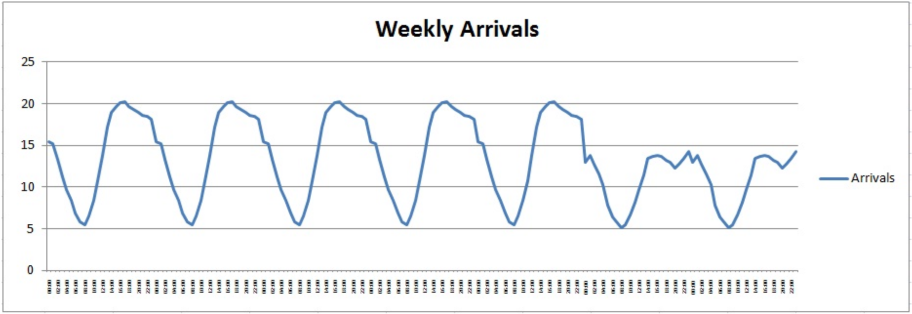
where T_i is the total sojourn time of network i^- and S_i^e is the residual service time at station i.

- Thus, finding the distribution of T_i is required for calculating the Offered-Load of station i.
- S_i^e's density function is $f_{S_i^e}(x) = \frac{1-F_{S_i(x)}}{E(S_i)}$.

Periodic Arrival Rate: Actual Arrival Rate

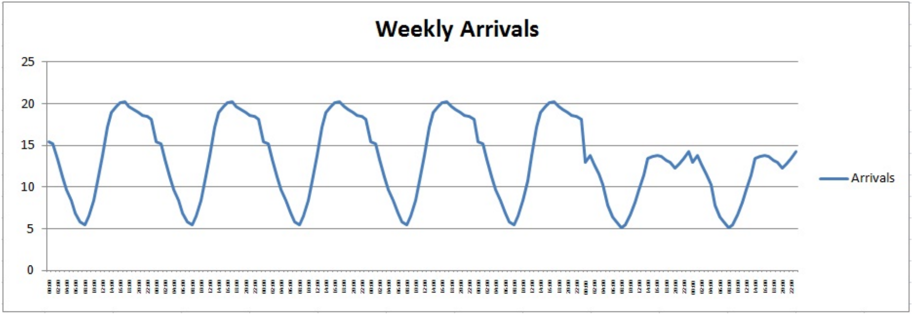


Periodic Arrival Rate: Actual Arrival Rate



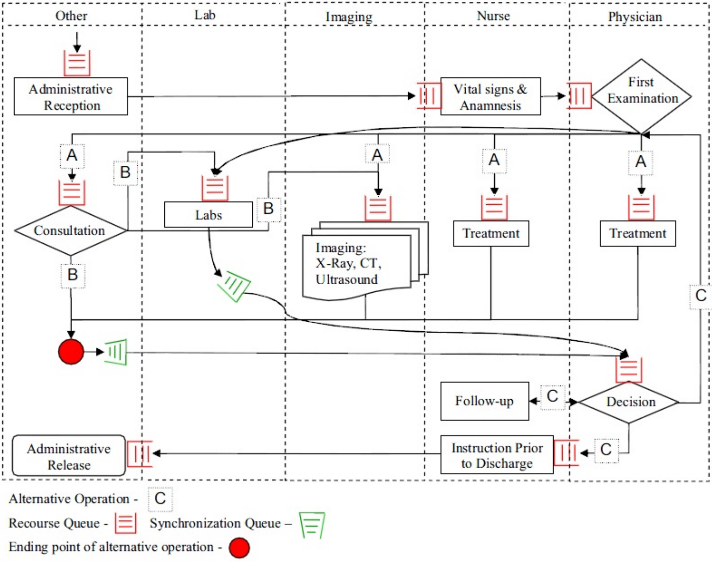
- The periodic nature of arrivals allows one to calculate the Offered-Load in an easier way: **Discrete Fourier transform.**

Periodic Arrival Rate: Actual Arrival Rate



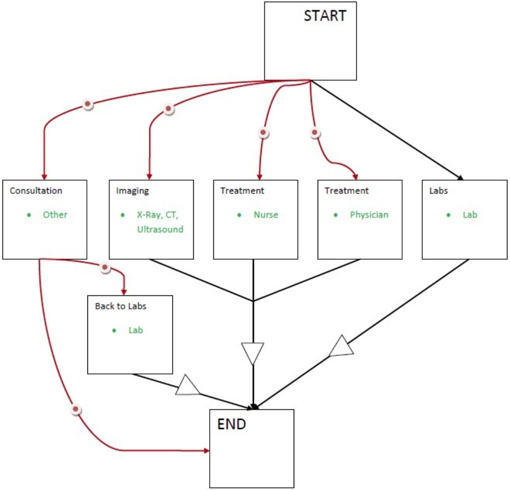
- The periodic nature of arrivals allows one to calculate the Offered-Load in an easier way: **Discrete Fourier transform.**
- The example of calculations were applied to the following fork-join network:

Periodic Arrival Rate: Actual Arrival Rate



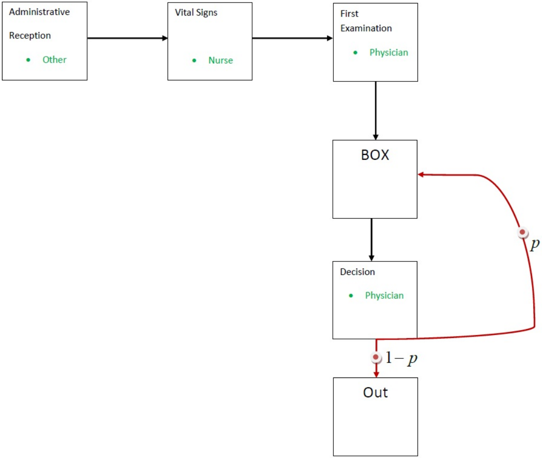
Periodic Arrival Rate: Actual Arrival Rate

To calculate the Offered Load, we disaggregate the complex queueing network into simpler smaller networks:



Periodic Arrival Rate: Actual Arrival Rate

Looking at the whole network as one station, will yield the following simple open network.



Example: The Offered-Load of Internal Drs:

1. We look at the activities that are executed by Internal Dr's: "First Examination", "Treatment" and "Decision".

Periodic Arrival Rate: Actual Arrival Rate

Example: The Offered-Load of Internal Drs:

1. We look at the activities that are executed by Internal Dr's: "First Examination", "Treatment" and "Decision".
2. We then calculate the Offered-Load of each one of the activities.

Periodic Arrival Rate: Actual Arrival Rate

Example: The Offered-Load of Internal Drs:

1. We look at the activities that are executed by Internal Dr's: "First Examination", "Treatment" and "Decision".
2. We then calculate the Offered-Load of each one of the activities.
3. Summing up the Offered-Load of the activities will give us the Offered-Load of Internal Dr's.

Periodic Arrival Rate: Actual Arrival Rate

- **Example: The Offered-Load of Internal Drs:**

$$R(t) = \sum_{j=1}^{3} R_{a_j}(t) = \sum_{j=1}^{3} \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{2\pi i}{N} kn} E(e^{-\frac{2\pi i}{N} kS}) \cdot E(S_{a_j}),$$

where a_1="First Examination", a_2="Treatment" and a_3="Decision".

Periodic Arrival Rate: Actual Arrival Rate

- **Example: The Offered-Load of Internal Drs:**

$$R(t) = \sum_{j=1}^{3} R_{a_j}(t) = \sum_{j=1}^{3} \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{2\pi i}{N} kn} E(e^{-\frac{2\pi i}{N} kS}) \cdot E(S_{a_j}),$$

  where a_1="First Examination", a_2="Treatment" and a_3="Decision".

- $x_0, ..., x_{N-1}$ are the arrival rates at times $t_0 = 0, ...t_{N-1} = N - 1$.

Periodic Arrival Rate: Actual Arrival Rate

- **Example: The Offered-Load of Internal Drs:**

$$R(t) = \sum_{j=1}^{3} R_{a_j}(t) = \sum_{j=1}^{3} \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{2\pi i}{N}kn} E(e^{-\frac{2\pi i}{N}kS}) \cdot E(S_{a_j}),$$

where a_1="First Examination", a_2="Treatment" and a_3="Decision".

- $x_0, ..., x_{N-1}$ are the arrival rates at times $t_0 = 0, ...t_{N-1} = N - 1$.
- X_k is the discrete Fourier transform of $x_0, ..., x_{N-1}$ ($X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N}kn} \quad k = 0, ..., N - 1$),
  - $n = \{k \bmod(N) | k = max\{m : m \leq t\}\}$.

Periodic Arrival Rate: Actual Arrival Rate

- **Example: The Offered-Load of Internal Drs:**

$$R(t) = \sum_{j=1}^{3} R_{a_j}(t) = \sum_{j=1}^{3} \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{2\pi i}{N} kn} E(e^{-\frac{2\pi i}{N} kS}) \cdot E(S_{a_j}),$$

  where a_1="First Examination", a_2="Treatment" and a_3="Decision".
- $x_0, ..., x_{N-1}$ are the arrival rates at times $t_0 = 0, ...t_{N-1} = N - 1$.
- X_k is the discrete Fourier transform of $x_0, ..., x_{N-1}$ $(X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \quad k = 0, ..., N-1)$,
  • $n = \{k \text{ mod}(N) | k = max\{m : m \leq t\}\}$.
- S_a is the service time of activity a.

Periodic Arrival Rate: Actual Arrival Rate

- **Example: The Offered-Load of Internal Drs:**

$$R(t) = \sum_{j=1}^{3} R_{a_j}(t) = \sum_{j=1}^{3} \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{2\pi i}{N} kn} E(e^{-\frac{2\pi i}{N} kS}) \cdot E(S_{a_j}),$$

  where a_1="First Examination", a_2="Treatment" and a_3="Decision".

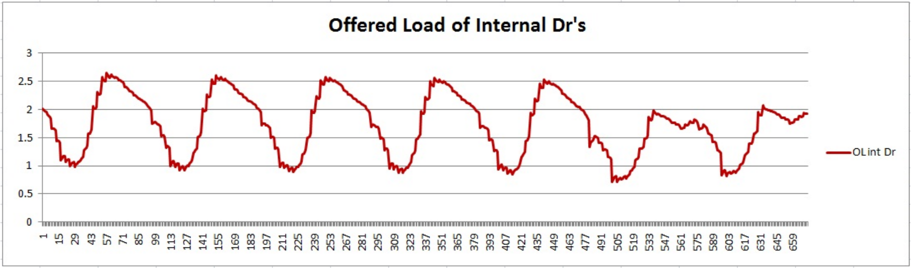
- $x_0, ..., x_{N-1}$ are the arrival rates at times $t_0 = 0, ...t_{N-1} = N - 1$.
- X_k is the discrete Fourier transform of $x_0, ..., x_{N-1}$
  $(X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \quad k = 0, ..., N-1)$,
  • $n = \{k \bmod(N) | k = max\{m : m \leq t\}\}$.
- S_a is the service time of activity a.
- S is the completion time of the network that starts at the ED entry and ends at activity a.

- **Example: The Offered-Load of Internal Drs:**

$$R(t) = \sum_{j=1}^{3} R_{a_j}(t) = \sum_{j=1}^{3} \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{2\pi i}{N} kn} E(e^{-\frac{2\pi i}{N} kS}) \cdot E(S_{a_j}),$$

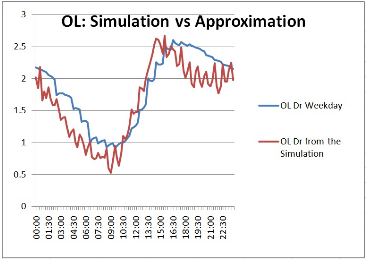
  where a_1="First Examination", a_2="Treatment" and a_3="Decision".
- $x_0, ..., x_{N-1}$ are the arrival rates at times $t_0 = 0, ...t_{N-1} = N - 1$.
- X_k is the discrete Fourier transform of $x_0, ..., x_{N-1}$
  $(X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \quad k = 0, ..., N - 1)$,
  • $n = \{k \text{ mod}(N) | k = max\{m : m \leq t\}\}$.
- S_a is the service time of activity a.
- S is the completion time of the network that starts at the ED entry and ends at activity a.
- To calculate the expression $E(e^{-\frac{2\pi i}{N} kS})$ we disaggregated the complex ED network into a few simpler subnetworks.

Periodic Arrival Rate: Actual Arrival Rate



Periodic Arrival Rate: Actual Arrival Rate

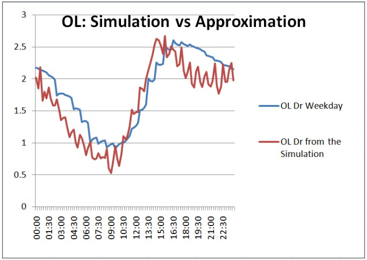
Comparison with the Offered-Load, calculated using the simulator



- The method provides an Offered-Load approximation. Its advantages are:

Periodic Arrival Rate: Actual Arrival Rate

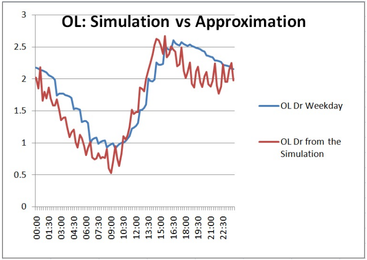
Comparison with the Offered-Load, calculated using the simulator



- The method provides an Offered-Load approximation. Its advantages are:
  1. Easy to calculate, once a calculation algorithm has been established.

Periodic Arrival Rate: Actual Arrival Rate

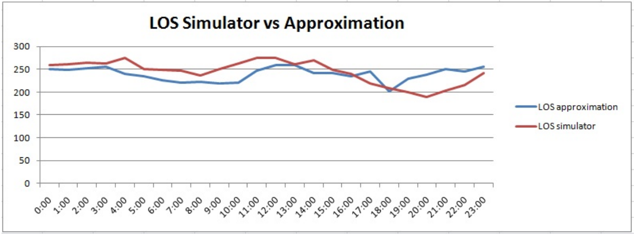
Comparison with the Offered-Load, calculated using the simulator

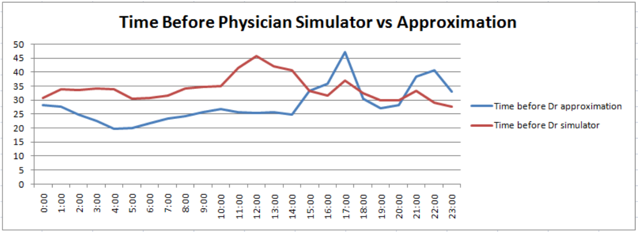


- The method provides an Offered-Load approximation. Its advantages are:
  1. Easy to calculate, once a calculation algorithm has been established.
  2. One can use this method on other fork-join networks (other ED networks in particular); instead of programming a simulator.

Comparing results: Simulator vs Approximation

For example, results of internal Drs (staffing with $\beta = 0.7$):





Project Completion Time by Continues Time Markov Chain

- Reminder: The offered load at station i, is given by

$$R_i(t) = E(\lambda(t - T_i - S_i^e)) \cdot E(S_i), \quad t \geq 0,$$

where T_i is the total sojourn time of network i^- and S_i^e is the residual service time at station i.

Project Completion Time by Continues Time Markov Chain

- Reminder: The offered load at station i, is given by

$$R_i(t) = E(\lambda(t - T_i - S_i^e)) \cdot E(S_i), \quad t \geq 0,$$

  where T_i is the total sojourn time of network i^- and S_i^e is the residual service time at station i.
- We adopt a result by Adlakha and Kulkarni [1986], to represent a fork-join network as a continues time Markov chain (activities on arcs vs activities on nodes, in our case).

Project Completion Time by Continues Time Markov Chain

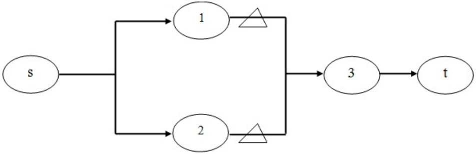
- Reminder: The offered load at station i, is given by

$$R_i(t) = E(\lambda(t - T_i - S_i^e)) \cdot E(S_i), \quad t \geq 0,$$

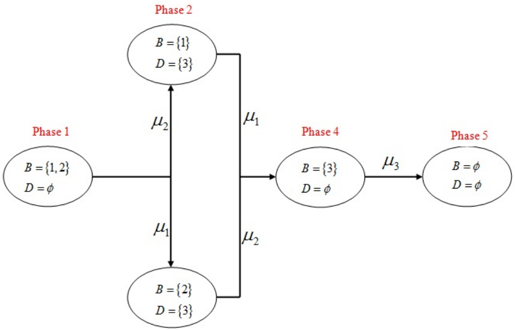
  where T_i is the total sojourn time of network i^- and S_i^e is the residual service time at station i.
- We adopt a result by Adlakha and Kulkarni [1986], to represent a fork-join network as a continues time Markov chain (activities on arcs vs activities on nodes, in our case).
- The continues time Markov chain state space consists of all the pairs (B, D) : B denote all the <u>active</u> nodes and D all the <u>dormant</u> nodes.

Project Completion Time: Example

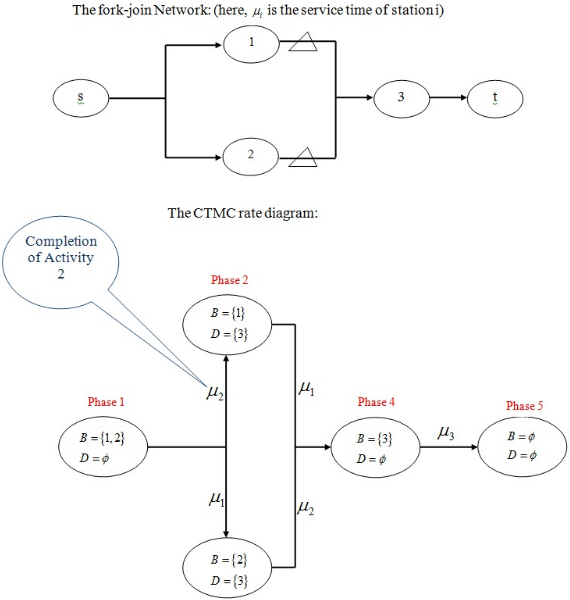
The fork-join Network: (here, μ_i is the service rate of station i)



The CTMC rate diagram:

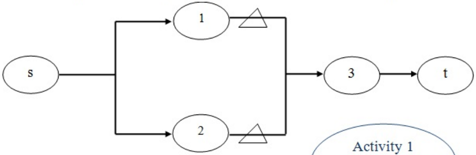


Project Completion Time: Example

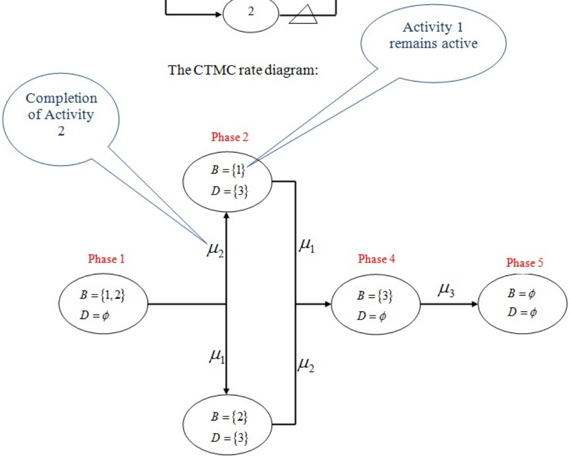


Project Completion Time: Example

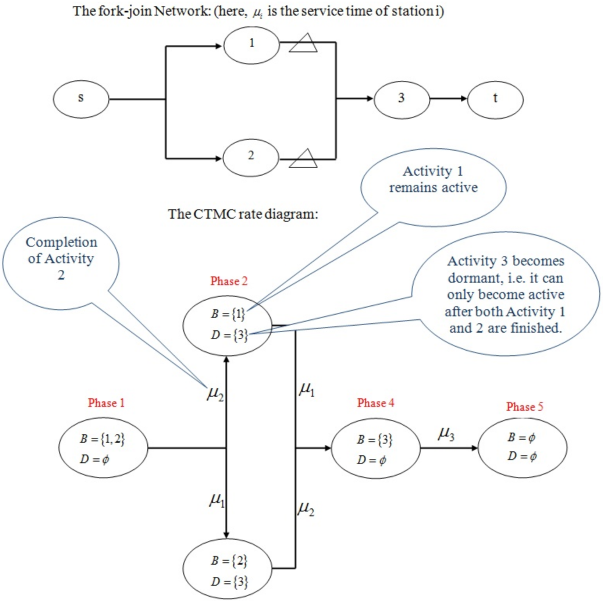
The fork-join Network: (here, μ_i is the service time of station i)



The CTMC rate diagram:



Project Completion Time: Example



Project Completion Time by Continues Time Markov Chain

The project completion time can be calculated via the next algorithm:

Backward Algorithm

Project Completion Time by Continues Time Markov Chain

The project completion time can be calculated via the next algorithm:

Backward Algorithm

- The cdf of the PERT network service time is $F(t) = p_1(t)$.

Project Completion Time by Continues Time Markov Chain

The project completion time can be calculated via the next algorithm:

Backward Algorithm

- The cdf of the PERT network service time is $F(t) = p_1(t)$.
- Here $p_i(t) = P(X(t) = N | X(0) = i); \quad 1 \leq i \leq N$, where

Project Completion Time by Continues Time Markov Chain

The project completion time can be calculated via the next algorithm:

Backward Algorithm

- The cdf of the PERT network service time is $F(t) = p_1(t)$.
- Here $p_i(t) = P(X(t) = N|X(0) = i); \quad 1 \le i \le N$, where
  1. The continues time Markov chain state space is numbered as $\{1, 2, ..., N\}$.

Project Completion Time by Continues Time Markov Chain

The project completion time can be calculated via the next algorithm:

Backward Algorithm

- The cdf of the PERT network service time is $F(t) = p_1(t)$.
- Here $p_i(t) = P(X(t) = N|X(0) = i); \quad 1 \leq i \leq N$, where
  1. The continues time Markov chain state space is numbered as $\{1, 2, ..., N\}$.
  2. $X(t), t \geq 0$, is the state of the project at time t.

Project Completion Time by Continues Time Markov Chain

The project completion time can be calculated via the next algorithm:

Backward Algorithm

- The cdf of the PERT network service time is $F(t) = p_1(t)$.
- Here $p_i(t) = P(X(t) = N|X(0) = i); \quad 1 \leq i \leq N$, where
  1. The continues time Markov chain state space is numbered as $\{1, 2, ..., N\}$.
  2. $X(t), t \geq 0$, is the state of the project at time t.
  3. $p_i(t)$ are given by

$$p_i'(t) = \sum_{j \leq i} q_{ij} p_j(t)$$

$$p_i(0) = \delta_{iN}, \quad 0 \leq i \leq N,$$

where $\delta_{ij} = 1$ if i=j, and 0 otherwise, and q_{ij} are taken from the infinitesimal generator matrix of the continues time Markov chain.

Project Completion Time by Continues Time Markov Chain

The project completion time can be calculated via the next algorithm:

Backward Algorithm

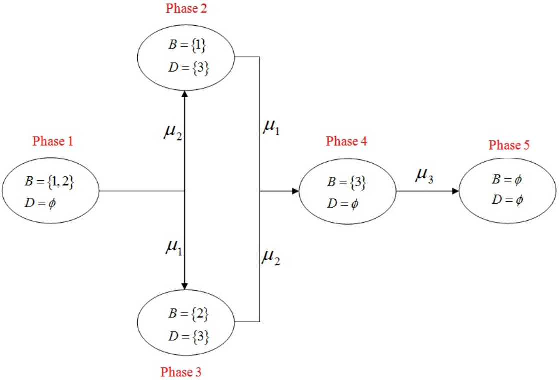
- The cdf of the PERT network service time is $F(t) = p_1(t)$.
- Here $p_i(t) = P(X(t) = N | X(0) = i); \quad 1 \leq i \leq N$, where
  1. The continues time Markov chain state space is numbered as $\{1, 2, ..., N\}$.
  2. $X(t), t \geq 0$, is the state of the project at time t.
  3. $p_i(t)$ are given by

$$p_i'(t) = \sum_{j \leq i} q_{ij} p_j(t)$$

$$p_i(0) = \delta_{iN}, \quad 0 \leq i \leq N,$$

     where $\delta_{ij} = 1$ if i=j, and 0 otherwise, and q_{ij} are taken from the infinitesimal generator matrix of the continues time Markov chain.
- We start the algorithm with $p_N(t) \equiv 1$, for $t \geq 0$ and compute $p_{N-1}(t), ..., p_1(t), p_0(t)$ recursively backward.

Project Completion Time by Continues Time Markov Chain: Example



- $P_5(t) = 1$,
- $P_3'(t) = \mu_2 \cdot P_4(t)$,
- $P_1'(t) = \mu_1 \cdot P_3(t) + \mu_2 \cdot P_2(t)$.
- $P_4'(t) = \mu_3 \cdot P_5(t)$,
- $P_2'(t) = \mu_1 \cdot P_4(t)$,

Summary

- Staffing according to the Offered Load (square root staffing) reduces patients length of stay at the ED.

Summary

- Staffing according to the Offered Load (square root staffing) reduces patients length of stay at the ED.
- Staffing according to the Offered Load stabilizes measures of performance of the emergency department over time.

Summary

- Staffing according to the Offered Load (square root staffing) reduces patients length of stay at the ED.
- Staffing according to the Offered Load stabilizes measures of performance of the emergency department over time.
- Tractable analysis of a complex network (fork-join) captures the full complexity of an emergency department.

Future Research

- Inserting *Shift Scheduling* constraints to the staffing procedure.

Future Research

- Inserting *Shift Scheduling* constraints to the staffing procedure.
- Developing Offered Load Calculation Methods for Models with Generic Service Time Distribution

Future Research

- Inserting *Shift Scheduling* constraints to the staffing procedure.
- Developing Offered Load Calculation Methods for Models with Generic Service Time Distribution
- Developing Similar Methods to EDs with Different Operational Administration

Future Research

- Inserting *Shift Scheduling* constraints to the staffing procedure.
- Developing Offered Load Calculation Methods for Models with Generic Service Time Distribution
- Developing Similar Methods to EDs with Different Operational Administration
- Time Stable Performance Measures


Questions
Thank You