Proceedings of the 2018 Winter Simulation Conference
M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, eds.

A MANAGEMENT TOOL BASED ON DISCRETE EVENT SIMULATION FOR HUMANITARIAN SUPPORT

Vlademir Fazio Santos
Michel Daoud Yacoub

School of Electrical and Computer Engineering (FEEC)
University of Campinas (Unicamp)
Av. Albert Einstein, 400
Campinas-SP, 13083-852, BRAZIL

Edson L. Ursini
Paulo S. Martins

School of Technology (FT)
University of Campinas (Unicamp)
R. Paschoal Marmo. 1888
Limeira-SP, 13484-332 , BRAZIL

ABSTRACT

Humanitarian aid is material or logistical assistance provided for humanitarian purposes, typically in response to humanitarian crises including natural disasters and man-made disaster. Humanitarian assistance requiring short response time windows in almost the whole world may be subject to long queues due to managing problems, e.g., the lack of control and/or inefficient infrastructure. This work tackles such challenge by proposing a low-cost planning and managing model and method based on a discrete-event simulation mirror connected through WEB tools to a near or far management level. The usual configuration of parallel servers (for instance, supported by local $RFID$ monitoring) is implemented by a discrete-event simulation model that is validated by Jackson Networks (and vice versa). The results show a flexible model that may identify bottlenecks in advance in order to accommodate traffic flow variations.

1 INTRODUCTION

Humanitarian aid consists of material and logistic assistance to people who need help. It is usually short-term help until the long-term help by government and other institutions replaces it. The primary purpose of humanitarian aid is to save lives, reduce suffering and respect to human dignity. In one particular case in Rio de Janeiro, Brazil (Globo 2018), a crowd of more than 13,000 people were given assistance of humanitarian nature during less than half a day.

The five most relevant issues that Richardson et al. (2016) concluded in their research on inventory prepositioning for humanitarian logistics are (in order of importance): the speed of response, physical infrastructure, support services, costs and labor availability. Rateb J et al. (2016) pointed out that top level management and stakeholders from humanitarian organizations highlight the need of Define-Measure-Analyze-Improve and Control lack and instability of resource (even human resources) within short time-frame operations. Authors also identified the necessity of assistance for front liners and implementing staff in terms of knowledge, quality information and flexibility from actual implementation tasks. These issues would allow field teams to be more focused on quality and accountability. These works show us that weakness of management and inefficient infrastructure are major issues. When the involved scenario calls for logistic assistance, the structuring of the humanitarian aid in the form of queues and service stations is necessary to streamline these processes.

To fill in these gaps, this work proposes an efficient management control as a method and low cost solution, which is structured by discrete-event simulation (DES) (Banks et al. 2010) and Jackson networks (JN) (Jackson 1957) supported by $RFID$ and digital twin (DT) (Venkatapathy et al. 2017; Alam and El Saddik 2017; Schluse et al. 2018). It is not within the scope of this work to study the specific $RFID/WEB$ tools, but rather to show that, as communication tools to support the use of DES, they can allow the real time monitoring of actual operations. The JN design is a choice due to the fact that it is a well-known

978-1-5386-6572-5/18/$31.00

topology for parallel and independent servers submitted to single-class products/services. In our study, we validated it for multi-class services (each server one different service). This method allows accurate and quick decision-making for planning and managing queues, personnel and materials for actual actions during short-time operations. The *DES* model validates the *JN* and results show its flexibility in following traffic flow variations.

The remainder of this paper is organized as follows: Section 2 discusses related work and some other applications. Section 3 describes the problem statement; Section 4 introduces the proposed simulation model. Section 5 addresses our remarks and discussion. Section 6 presents our conclusions and suggestions for future work.

2 RELATED WORK AND BACKGROUND

There are many planning models for increasing throughput using *JN*s. Kim and Kim (2015) is one example applied to medical emergency care with the use of hybrid priority with pooling for patients in risk of death. In contrast, our model allows the choice of one among many objectives (throughput, maximum time, average time) control submitted to $FIFO$ discipline. Most importantly, the method allows a real-time management control in a digital mirror supported by a simple and robust structure (*JN*) for multi-class services to care people in the state of vulnerability.

Another regular managing model is for optimal inventory allocation of multi-class products in single server systems, as studied by de la Cruz and Daduna (2017). Differently, our work is focused in *DES* for estimating the best results in a multiple parallel servers system applied to multi services in mixed queuing networks.

Bitran and R (1994) show a optimization model for planning mixed queues networks with returning entities in a *JN* topology. Unlike, our work handle a *DES* real-time managing model for both operational and planning actions.

With regard to the DT use, normally there is the development of sophisticated models, e.g. the smart interaction controller for a digital twin tool offered by Alam and El Saddik (2017) with fuzzy and Bayes supported by highly capable infrastructure. We propose a *JN* as a simple, low cost digital twin mirror application to be used in the presence of poor local infrastructure.

Venkatapathy et al. (2017) propose the use of the dynamics of intra-logistics (associated to cyberphysical systems) with a system dynamics approach for the understanding of syntaxes that can lead to new services from inter-operating systems. Our focus is to provide functionality to the dynamics of intra-logistics to manage its emergent deviations.

Medical emergency procedures (entity is the patient, services are jobs, independent or not) is another example as showed by Xu et al. (2014), can use $IoT-based$ methods where a chain of simultaneous procedures with responses coming through the Internet can guarantee patients' lives. However, besides a planning model our work also deals with a managing tool for operational decision making.

Some examples of practical processes illustrated by Fig. 1 as a description of a wide variety of scenarios with different types of operation, independents or not that can be executed simultaneously in different entities. A first actual example is a local process of loading trucks (entity is the truck, commodities are jobs) with the need to carry all product types as a whole cargo where commodities (corn, beans, rice, etc.) can be randomly laden in trucks by servers availability.

3 PROBLEM STATEMENT

This type of intra-logistics Gunther et al. (2008) operation assists people in need in an environment of multiple humanitarian services and they have to wait in very long queues to be cared for, or they must face intense competition for some assistance. People arrive at a supporting location ($Fig.1$) with a set of assistance needs from e.g. receptionists, nurses, medical doctors, dentists, and lawyers. Local infrastructure is usually poor; it needs material support and it also receives operational directions from volunteers that need to be guided too.

A people identification process is proposed for *Welcome* & *screening* assistance by the use of a regular passive *RFID* devices (for each individual) that can be completed with presence sensors control (e.g., *Arduino* based).

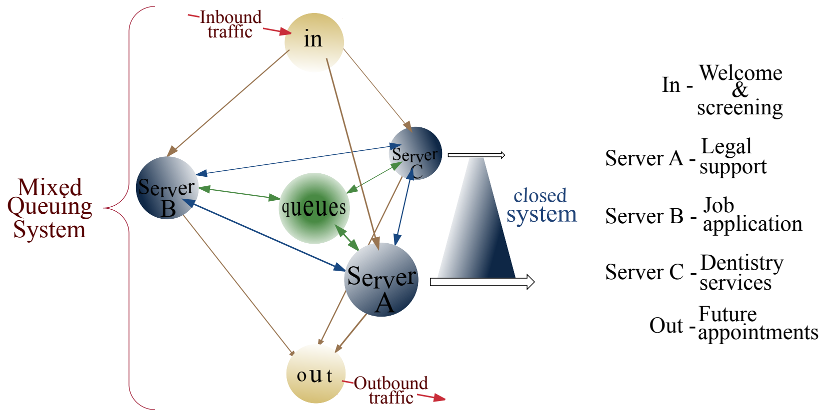

Figure 1: General system model. A general view of a three servers system. It can be extended as *n* server system.

The arrivals and exits in any of the system *assistance points* (service stations) use *RFID* sensors to quickly identify quantities and time intervals which allows online supply of system statistics to support an intelligent system of a digital management interface. We assume that after an initial reception an individual already has a list of requested necessities. He/she has to hold for the next service in waiting rooms to be attended by availability to all the services. A person with no more necessities to be attended is evaluated and, in general, he/she is free to go away or to be back in a return schedule.

We ran a set of simulated experiments that assumes three different services for people that arrive on a daily basis, where the demand is organized and served by order of arrival. A full assistance has three services, each one with a service distribution time in a general design of one server for one node what can be easily extended for one node with many parallel servers. The servers simultaneously work in this class of large-scale processes, whose sequence of steps can be random, performed by a parallel network what often form long queues and cause loss and damage (Adan and Resing 2001; Bitran and R 1994; Eeckhout 2016).

This is a class of problems of provision of large-scale services (it can also be transportation and/or transactions) in a multi-class process of queuing networks. It is a topology of Mixed Queues with a closed queue as a core where entities are distributed by chance and dynamically for all servers without preemption (Bitran and R 1994).

The following are the assumptions of the proposed model:

1. server queues have no space constraints;
2. the scheduling policy is first-in-first-out (FIFO);
3. entities may randomly start from any free server;
4. both the arrival $(1/\lambda)$ and the service time $(1/\mu_i)$ of individuals/entities are Markovian;
5. the provision of services are independent from each other and there is no precedence relationships between individuals;
6. one server per service station, and each server has the same probability of receiving an entity;
7. each server provides a distinct service;

8. an entity may only enter the system upon the exiting of another one, i.e. Under high traffic, the system operates under a statistical equilibrium. This means that an entity may only have access to a server once this same server releases another entity (Bitran and R 1994);.
9. each individual has to be served by a complete set of servers before leaving the system (Fig. 3) (i.e. the probability p of going from a server to the next available one is 100%);

The system with these nine characteristics may be modeled as JN. We designed a DES model (Subsection 4.2) which was analytically validated by JN and vice-versa. As a consequence, the system traffic logically behaves as a model of n serial queues, as confirmed by a comparison between the analytical and simulation models (Section 6). Once the model is validated, it is amenable to other distributions and system settings. We derive some simple equations (Subsection 4.1) for outbound traffic that analytically corroborates the simulation results and vice-versa.

In addition to monitoring processes, the proposed model may also be able to support planning and management of operations by dealing with relatively high traffic flow, in local operations or at distance, with limited number of available servers and very long queues. For example, as shown in Fig. 8, as the traffic increases, the JN model is able to correspondingly adjust its traffic behavior. Specifically, when one server overloads causing the formation of queues, the other servers can accommodate the excess load. Therefore, the capacity for load balancing is implicit in the JN model. This feature allows that the JN model may be used not only for static planning and dimensioning but also for online, dynamic management of an operation.

4 MODELING

Several value chains perform a similar basic flow outlined in the preceding section, (i.e. the sequential and undivided provision of services to an entity) that shows a response time with a given service time, based on previous knowledge/data of the system. For the particular case described in Fig. 1, the action conducted by the Office of the Secretary of Public Security of Rio de Janeiro City generated more than 13,000 assists to local citizens (Globo 2018). The lack of official reports with detailed data justifies modeling with characteristic mean values. The general structure starts with a *Welcome screening* service, which assigns individuals to the adequate services. Clearly, there may be a large variability of service times depending on the application, ranging from milliseconds (e.g. computer operating systems) to hours or even days/weeks (e.g. medical, military services).

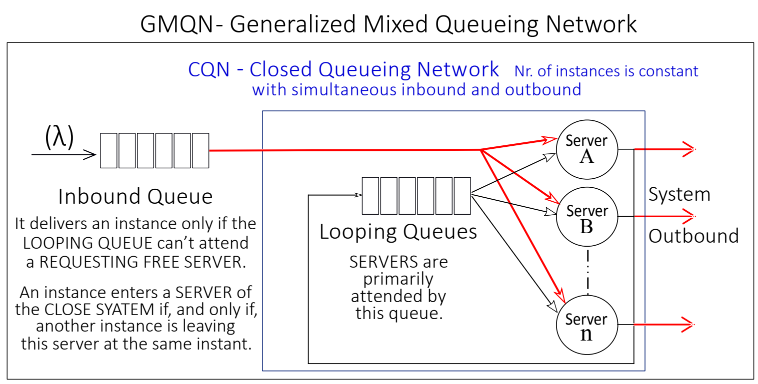

Figure 2: System schematic model − *Server A* attends with $E[B_1]$ ut, *Server B* with $E[B_2]$ ut and *Server C* with $E[B_3]$ ut. Here, a server can be seen as a node with many parallel servers. *Looping Queue* is a logic queue that highlights the set of individual queues into the system core with closed queues network.

Figure 1 may be abstracted away as shown in Fig. 2, which shows the entry through the *Inbound queue*. This queue is used to implement an admission control that is not part of the JN model. If the system is bound to overloading, incoming entities are retained in this queue or even blocked away from entry in this queue if it is already full.

Each server has its own queue and the *Looping queue* is a logical representation for this set of server queues. As mentioned earlier, an entity may enter and leave this queue and remain in the loop until it has been serviced by each and all servers.

The model shown in Fig. 2 belongs to a specific class of mixed queuing systems with a closed queuing network in its core and n servers.

4.1 Simulation and Validation

The Jackson's theorem (Jackson 1957) states that the probability of an entity moving from one node to another is $p = 1$, as shown in the traffic model $(Fig.3)$. The traffic in $Erlang\ (E)$ is described on a per-server basis ($\lambda/\mu_i < 1$ implies a stable system). Little's law (Little 1961) shows that the average residence time ($E[R]$) for each node is described as in Equation (1):

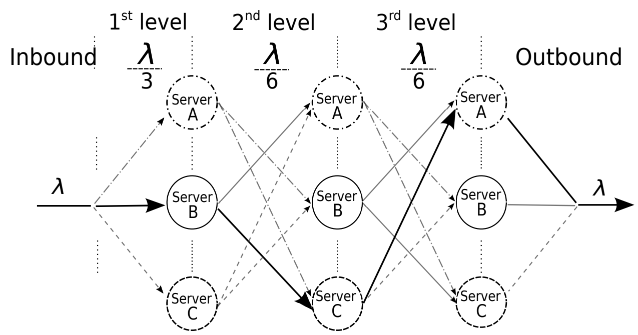

Figure 3: Equivalent traffic model for three servers $-$ *Server A* attends with $E[B_1]ut$, *Server B*, with $E[B_2]ut$ and *Server C*, with $E[B_3]ut$. Bold arrows show an equivalent traffic route $\rightarrow \lambda/3 + 2 \times \lambda/6 + \lambda/3 = \lambda$.

$$E[R_i] = 1/(\mu_i - \lambda) \tag{1}$$

The mean residence time values in each node result in a traffic that may be calculated for each server, according to the values of λ in each level shown in Fig. 3 for three servers. In Fig. 4, a *JN* that is the logical equivalent of Fig. 3, an entity traverses three different jobs (three queues) with probability 1. Each server generates an inbound traffic to all other servers.

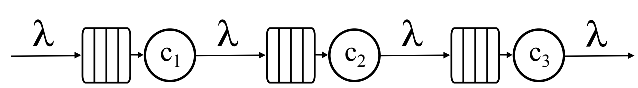

Figure 4: Three single queues equivalent system. Servers can be set in any order, e.g., server C_1 provides service with E[B_1] *ut*, C_2, with E[B_2] *ut* and C_3, with E[B_3] *ut*.

For this reason, the arrival rate in each queue is the same as the overall arrival rate λ. This observation allows us to calculate the average residence time $E[R]$ in the system through Equation (2):

$$E[R] = \sum_{i=1}^{n} E[R_i] = E[R_1] + E[R_2] + E[R_3] \tag{2}$$

It should be also noted that, from a performance viewpoint, an entity has to go through all servers and it does not matter which is the first one. As a first result we have for the numerical example $E[R] = 105$ *ut*, and with Eq. (1), it is possible to calculate a system equivalent service time μ_{eq} through Eq. (3):

$$\mu_{eq} = \frac{1}{E[R]} + \lambda \tag{3}$$

$$E[B_{eq}] = \frac{1}{\mu_{eq}}$$

$$E[B_{eq}] = \frac{E[R]}{1+\lambda\ .\ E[R]} \tag{4}$$

The use of Eq. (4) yields $E[B_{eq}] = 23.3\ ut$ for $\frac{1}{\lambda} = 30$. Otherwise,

$$\lim_{E[R]\ \to\ \infty} E[B_{eq}] = \lim_{E[R]\ \to\ \infty} \frac{E[R]}{1+\lambda\ .\ E[R]} = \frac{1}{\lambda}$$

$$\mu_{eq} > \lambda \rightarrow E[B_{eq}] < \frac{1}{\lambda} \tag{5}$$

Equation (5) implies that there is an increasing number L_{eq} of entities in the system queues when the statistical equilibrium evolves to values of $1/\lambda$ smaller than system service times. To visualize it we can observe three quantities expressed by B_{eq} in Eq. (9); a quantity numerically calculated with results of the simulation model TEX_{limit} (Eq. 6) that shows the average exit time of outbound entities, or it denotes the actual average system response time considering lost entities; and TEX (Eq. 7) that reflects the possible system response time as a maximum due to it does not consider entities losses during high traffic (Figure 8).

$$TEX_{limit} = \frac{Replication\ time}{\#\ of\ outbound\ entities} \tag{6}$$

$$TEX = \frac{Replication\ time}{\#\ of\ created\ entities} \tag{7}$$

On the other hand, $E[R]$ can be written as in Eq. 8, the sum of an equivalent time parcel due to service, $Serv_{eq} = E[B_{eq}]$, and another waiting time due to queues, Q_{eq}, in two different intervals - under system limit and over system limit.

$$E[R] = E[B_{eq}] + Q_{eq} \tag{8}$$

We use Eq. (4) and Eq. (8) to derive Eq. (9) as a relation for $E[B_{eq}]$ and Eq. (10), for L_{eq} (equivalent mean number of entities in queue).

$$E[B_{eq}] = \frac{E[R]}{1+\lambda\ .\ (E[B_{eq}]+Q_{eq})} \quad \rightarrow \quad E[B_{eq}] = \frac{E[R]}{1+A_{eq}\ +\ L_{eq}} \tag{9}$$

First, in the general case for traffic flow under the limit, it results a calculation that must consider the servers in high traffic, but not overloaded (Eq. (10)). It means servers with the maximum offered traffic in a *JN* model is as $A_{eq} =\ \lambda\ \times(E[B_1]\ +\ E[B_2]\ +\cdots+\ E[B_n])\ Erlangs$ what follows:

$$L_{eq} = (\frac{E[R]}{E[B_{eq}]} - 1)\ -\ A_{eq} \qquad If\ queue\ is\ stable, \frac{E[R]}{E[B_{eq}]} > (1\ +\ A_{eq}). \tag{10}$$

Second, during overflow traffic that means n *servers* work with a full traffic (crowded system) condition ($A_{eq} = n\ Erlangs$) in the presence of $E[B_{eq}] = 1/\lambda$. From Equation (9) during overflow traffic, the algebraic expression can be written as Eq. (11),

$$L_{eq_{hi}} = \lambda\ .\ E[R] - (n+1) \quad n = number\ of\ paralel\ servers\ (nodes) \tag{11}$$

By using some characteristic values as a numerical example (for three parallel servers), $E[B_1] = 1/\mu_1 = 20ut$, $E[B_2] = 1/\mu_2 = 15ut$ and $E[B_3] = 1/\mu_3 = 10ut$, we calculate:

$$\lambda = (1/30) \rightarrow E[R_{30}] = 105.00ut \rightarrow A_{eq_{30}} = (1/30) \times 45ut = 1.50 \rightarrow L_{eq_{30}} = 2.72$$
$$\lambda = (1/28) \rightarrow E[R_{28}] = 117.86ut \rightarrow A_{eq_{28}} = (1/28) \times 45ut = 1.61 \rightarrow L_{eq_{28}} = 3.40$$
$$\lambda = (1/26) \rightarrow E[R_{26}] = 138,37ut \rightarrow A_{eq_{26}} = (1/26) \times 45ut = 1.73 \rightarrow L_{eq_{26}} = 4.48$$
$$\lambda = (1/24) \rightarrow E[R_{24}] = 177.14ut \rightarrow A_{eq_{24}} = (1/24) \times 45ut = 1.88 \rightarrow L_{eq_{24}} = 6.50$$
$$\lambda = (1/22) \rightarrow E[R_{22}] = 285.48ut \rightarrow A_{eq_{22}} = (1/26) \times 45ut = 2.05 \rightarrow L_{eq_{22}} = 12.1$$

It is seen by Eq. 11 that as a consequence of $E[B_{eq}] > 1/\lambda$ be the limit, $L_{eq_{hi}}$ has no upper limit. The simulation experiments confirm results for traffic limit as can be seen next (Subsection 4.2).

4.2 Simulation Model Description

Running the simulation model built in a design of one single server per node figure 5 and setup for $FIFO$ policy (the logic of three servers in series), 100 replications of $100,000ut$ each, warm-up of $10,000ut$ and confidence level of 95%. Thus, in this way, after the validated simulation program, other tests could be performed and some other results obtained.

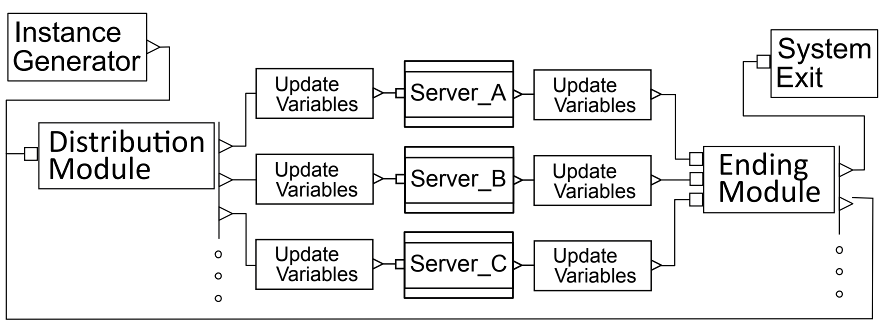

Figure 5: Simulation model. $Distribution\ Module$ draws equal probabilities to all servers driving them the requested entities. The block $Ending\ Module$ returns unfinished entities or concludes finished processes.

The simulation model was developed with an entities generator that distributes both time between arrivals and respective service times and feeds the two-phase system: 1^{st}) it controls and limits the admission of entities with $FIFO$ policy; 2^{nd}) it distributes entities for servers. The whole system contains blocks $Update\ variables$ that update data from/to entities and system what represent the inbound and outbound actions for the actual $RFID$ systems sensors.

In this example ($Fig.$ 5) product or service demanded by the entity requires, as said before, a task performed in one of three different servers. Entity can leave the system only after it passed once by each of the servers. We describe in the next paragraphs two structural modules of the simulation model - *Distribution Module* and *Ending Module*.

Distribution Module (figure 6) - It is the double-staged admission control system. In the first stage it acts as a parking-lot (*System Limit*), where instances enter the system with *FIFO* policy or leave it during overloading. It compares the number of system instances *varINLIMI* with *varLIMIT* variable (the total system capacity). If the value of *varINLIMI* represents an under capacity system, the next queued instance enters the *Smart Buffer* that together with the *Probability Equalizer* makes up the second stage.

The *Smart Buffer* selects which servers have not yet rendered and send the entity to the *Probability Equalizer*. The *Probability Equalizer* block is used to distribute traffic with equal probability for all servers that have not yet provided service to the entity. *Availability* blocks test if entity is already attended and if its routed server is free to receive it. It checks an availability variable named *varIN_X* (*X* is the server number) to open server access and test the attribute variable Ent_X to certify the correct entity to the free server. Released entities go to the *Avail* route and the denied ones go to the n/A route (not Available exit) to be returned to the *Probability Equalizer* module. Blocks *Update Variable* insert updated information to entities attributes (e.g., realized services) and system variables. This updating is even local or external (e.g., at distance) with *RFID/Internet* technologies. An important note is that in this configuration, buffers are queues that may guarantee independence from waiting entities and servers.

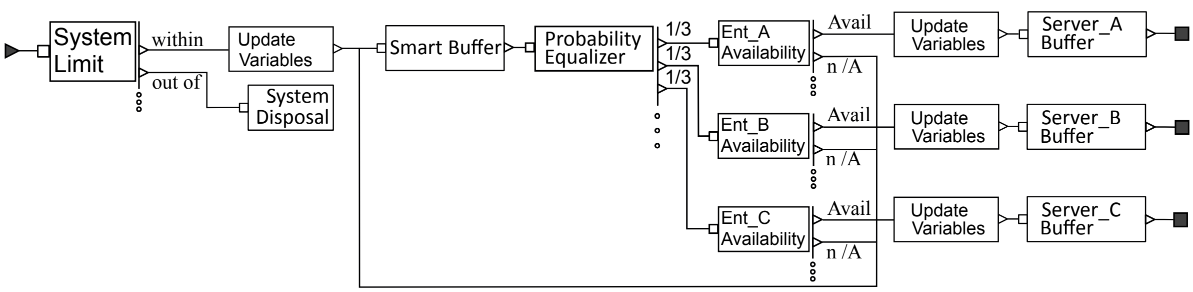

Figure 6: *Distribution Module* schema. *System Limit* disposes undesirable overflow and the logical Block *Probability Equalizer* garantees the sum of probabilities is equal to one and the same probability to each of the servers. The servers buffers are independent queues that release entities by servers' request.

Ending Module represented in figure 7 is a router that returns unfinished instances, or instances with services to be done, to the *Probability Equalizer* and sends (*Return to servers* route) those instances without services to receive to system exit. It also counts finished instances into 18 time-average intervals (the first 17 intervals distributed from $50ut$ to $4000ut$ and the last one to values higher than $4000ut$) to prepare the necessary data for a histogram that can represent the output probabilistic distribution.

First stage of *Ending Module* checks service completion of an entity by comparing values of entities attributes *Ent_X* (*X* is the entity type). Attended entities are identified to leave the system. The second stage of the *Ending Module* classifies outbound entities into 18 residence time intervals before exiting system.

For instance, starting simulations with inbound time average $1/\lambda = 30ut$, changing to $1/\lambda = 22ut$ and $1/\lambda < 21ut$, keeping the same other parameters the simulation model runs to return the next simulation mean-time averages:

$$for\ 1/\lambda = 30ut \ \rightarrow\ TAVG_{30} = 104.85ut\ \ with\ \ HW = 1.1278ut\ \rightarrow\ for\ E[R_{30}] = 105.00ut$$

$$for\ 1/\lambda = 28ut \ \rightarrow\ TAVG_{28} = 117.03ut\ \ with\ \ HW = 1.4580ut\ \rightarrow\ for\ E[R_{28}] = 117.86ut$$

$$for\ 1/\lambda = 26ut \ \rightarrow\ TAVG_{26} = 139.29ut\ \ with\ \ HW = 2.3726ut\ \rightarrow\ for\ E[R_{26}] = 138.37ut$$

$$for\ 1/\lambda = 24ut \ \rightarrow\ TAVG_{24} = 178.84ut\ \ with\ \ HW = 5.2408ut\ \rightarrow\ for\ E[R_{24}] = 177.14ut$$

$$for\ 1/\lambda = 22ut \ \rightarrow \ TAVG_{22} = 286.75ut \ \ with \ \ HW = 13.620ut \ \rightarrow \ for\ E[R_{22}] = 285.48ut$$

$$for\ 1/\lambda = 21ut \ \rightarrow \ TAVG_{21} = 475.96ut \ \ with \ \ HW = 37.159ut \ \rightarrow \ for\ E[R_{21}] = 491.59ut$$

The developed model is thereby validated and responds perfectly, but with low losses for $1/\lambda < 21ut$. In this case, model runs to $TAVG = 475.96ut$ and $HW = 37.159ut$ for the mathematical expectation of $E[R_i] = 491.59ut$ within the upper limit of the simulated range. The equivalent traffic model in *Fig.* 2 adheres to the *JN* previously proposed. An important aspect of this work is that Jackson's analytical model validated the discrete simulation model and vice versa because there was difficulty in how best to represent the actual system. Once validated, the simulation program is ready for more results. The first one is the identification of the numeric system traffic limit. Assuming a λ range from 1/22 to 1/30 (this is regular range, in general, due to physical restrictions), the traffic $A = \lambda \times$ *average service time* in a stable condition, the upper limit for the most intense system traffic is calculated as 2.250 E ($\lambda = 1/22$). In actual operations these values must be obtained from actual system characteristics and constraints.

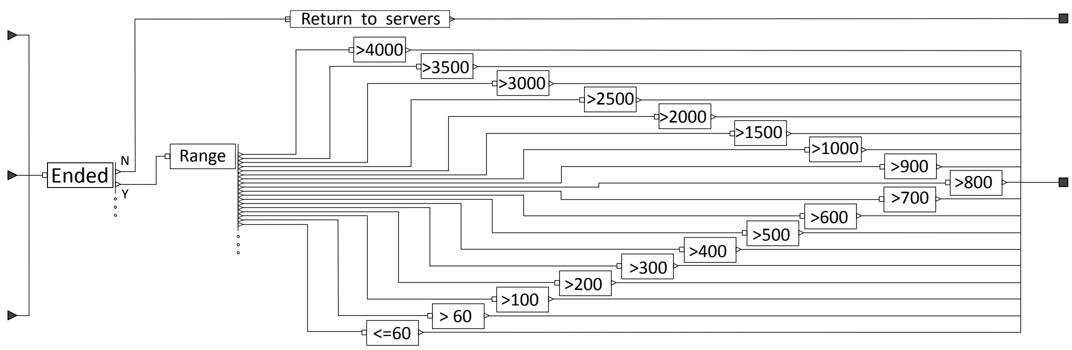

Figure 7: *Ending Module* schema. The representation shows the upper line *Return to Servers* and the time range counters in *time units*.

5 REMARKS AND DISCUSSION

Simulation experiments highlight system bottlenecks with the $TEX_{limit}\ vs\ B_{eq}$ comparison as shown in Figure 8.

As λ increases over time, TEX_{limit} value tends to $E[B_{eq}] = 23.3ut$ what means that both tend to the highest service time of servers in system ($E[B_{highest}] = E[B_1] = 20ut$).

$$\lambda \rightarrow minimum,\ TEX_{limit} \ \ and \ \ E[B_{eq}] \ \rightarrow \ E[B_{highest}]$$

In an actual action with this *management tool* supported by a simple and small RFID/Internet communication structure, the management level can update the evolving data easily monitoring λ, all servers μ_i and estimate L_{eq} and TEX to make decisions, take actions and generate staff information and reports in real time.

This simple managing method goes beyond the example provided here. Servers perform simultaneous tasks in parallel but the system behaves as if it is in series. It happens due to the fact that this system performs different tasks to different entities at the same time for many time intervals. For this case study, we suppose independent tasks for three servers in parallel, but as the simulation model is validated, it can be also extended to tasks dependent on each other.

5.1 Implementation Strategy

The model discussed so far provides the theoretical framework for the scheduling of humanitarian aid across several parallel servers (or service stations). Clearly, the successful implementation of such strategy must be accompanied by modern technology that allows for the proper accounting for the process variables such as number of entities that enter the system, time stamps, and residence time among others. Two technologies that may support the implementation is *digital twin* and *RFIDs* sensoring. Specifically, with the former, the collection of information system within physical operation feeds the mirror of a digital twin design that enables the automation of a high value-added chain (Schluse et al. 2018).

Notice that it is not necessary for all levels and/or servers to be close together because, e.g., it is possible to manage these different operations remotely through a conventional Internet (Goyal and Fussell 2017). More recently, the support of IoT extended this perspective to meet new needs such as ubiquitous access, real-time analysis, availability to different platforms (Apple iOS, Google Android, Windows Phone) and spatial location (Kua et al. 2017). As each instance (an individual) must pass through all the servers to complete the process, the distance between servers is not a hindrance. There are many simple $RFID$ designs available for collecting data and the Internet basis for transferring a small set of data is also available.

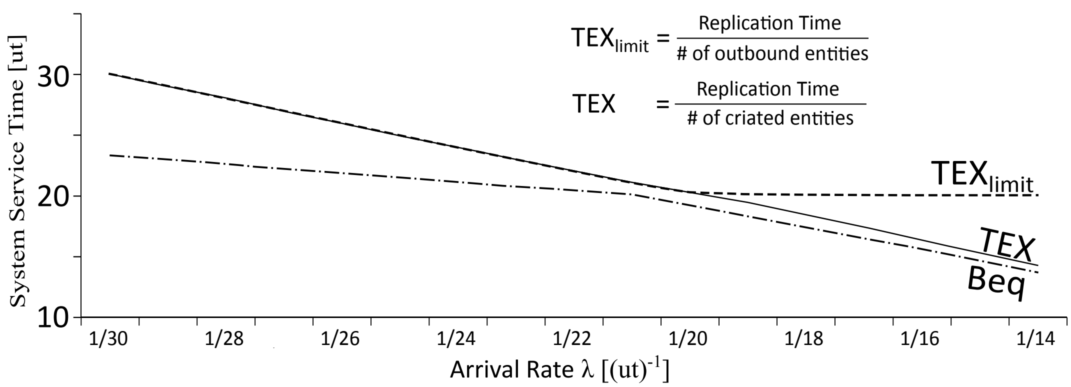

Figure 8: System service time overview. TEX_{limit} shows the actual average system response time. TEX reflects the maximum system response time. Beq is the calculated value $(Eq.(9))$ for system response time.

6 CONCLUSION

When humanitarian aid scenarios strike, preparedness is of utmost importance, including how well response teams are able to react. This work has dealt with the issue of scheduling applied to managing humanitarian aid operations. A model based on a set of parallel servers and service lines using FIFO scheduling was proposed and validated by Jackson networks. The model has shown to be flexible in that is able to adapt to different traffic rates. It may be implemented as a management tool for both planning and operational support. Simulation design with buffers instead of regular queues allows the use of any scheduling policy. In an actual action supported by RFID/Internet, it can be planned a priori, and different scenarios and solutions can be tested and adjusted in real-time basis. Because the simulation model allows the evaluation of the distribution of exit times, it is possible to monitor and interfere in the process to avoid users with extreme delays.

The Jackson network model is not just the basis for a simple and robust managing structure. It is also flexible due to its adaptive traffic characteristic. In the presence of high traffic, the system throughput tends to its maximum value. When a server is overloaded, its incoming traffic is redirected to the remaining servers, instead of being blocked on its queue. Flexibility is the consequence of the n-level buffer system, as the remaining servers work as a temporarily relief for the incoming traffic. Analyzing the simulation response curves, we clearly see from that they diverge around the slowest server service time (20*ut* in the numerical example). As usual, this *JN* simulation model is a planning tool supported by the set of equations here addressed. The equivalent service time is easily accessed and fully adheres to the model mainly in high traffic. This *JN* queuing model is designed for exponential mean-time distributions (M/M/1) but the validated simulation model fits also for other general distributions, e.g., G/G/1 and the ones presented in Section 2, in a set of real world applications this network model can be addressed.

The *DES* characteristic model supported by $WEB/RFID$ technologies may also solve some operational issues highlighted in Section 1. Quick real-time answers can be delivered by the coordination/supervisory level (local or international, specific or multi-skilled) in terms of assistance to the operational teams (intra-logistics or inventory transportation) due to the soft and simple tool structure that can run new actual scenarios in few seconds. This characteristic shows that the distance between different steps that make up a complete process is no longer a problem (Eeckhout 2016; Feki et al. 2013).

As future work, we suggest studies to understand how the *JN* structure behaves with scheduling policies different from FIFO, for instance, using SJF (Shortest Job First) with known service times. Another possibility would be to focus in the effects of space restrictions to queues in the *JN* structure by using admission control. It is important to further study the effect of the increasing number of entities in the queue, and the use of measurements to find more realistic probability distributions, different from the exponential. Application of scheduling heuristics or analytical models at various decision points by using measurements from the studied environment can be other future work.

REFERENCES

Adan, I., and J. Resing. 2001. *Queueing Theory. [electronic resource]*. 1st ed. Eindhoven, NB, Netherlands: Eindhoven University of Technology. Dep. of Math. and Comp. Sci., 2001.

Alam, K., and A. El Saddik. 2017. "C2PS: A digital twin architecture reference model for the cloud-based cyber-physical systems". *IEEE Access* 5:2050–2062.

Banks, J., J. Carson II, B. L. Nelson, and D. M. Nicol. 2010. *Discrete-event System Simulation*. 3rd ed. Upper Saddle River, NJ : Person Education Inc., 2010.

Bitran, G. R., and M. R. 1994. "Open Queueing Networks: Optimization and Performance Evaluation Models for Discrete Manufacturing Systems". *Production and Operations Management* 5(2):46.

de la Cruz, N.-N., and H. Daduna. 2017. "Optimal capacity allocation in a productioninventory system with base stock". *Annals of Operations Research*:1–16.

Eeckhout, L. 2016. "The Internet of Things Revolution". *IEEE Micro* 36(6):4–4.

Feki, M. A., F. Kawsar, M. Boussard, and L. Trappeniers. 2013. "The Internet of Things: The Next Technological Revolution". *Computer* 46(2):24–25.

Globo, O. 2018. "Acao social na Vila Kennedy realizou mais de 13 mil atendimentos". Online. https://oglobo.globo.com/rio/acao-social-na-vila-kennedy-realizou-mais-de-13-mil-atendimentos-22504952.

Goyal, N., and S. R. Fussell. 2017. "Intelligent Interruption Management Using Electro Dermal Activity Based Physiological Sensor for Collaborative Sensemaking". *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1(3):52:1–52:21.

Gunther, O. P., W. Kletti, and U. Kubach. 2008. *RFID in Manufacturing. [electronic resource]*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2008.

Jackson, J. R. 1957. "Networks of Waiting Lines". *Operations Research* (4):518.

Kim, S., and S. Kim. 2015. "Differentiated waiting time management according to patient class in an emergency care center using an open Jackson network integrated with pooling and prioritizing". *Annals of Operations Research* 230(1):35 – 55.

Kua, J., S. H. Nguyen, G. Armitage, and P. Branch. 2017. "Using Active Queue Management to Assist IoT Application Flows in Home Broadband Networks". *IEEE Internet of Things Journal* 4(5):1399–1407.

Little, J. D. C. 1961. "A Proof for the Queuing Formula: L = lambda/W". *Oper. Res.* 9(3):383–387.

Rateb J, S., S. Firas Izzat Mahmoud, D. Samer Eid, S. Nadia J, S. Rawan Ali, and D. Hannah. 2016. "Benchmarking of TQM practices in INGOs: a literature review". *Benchmarking: An International Journal* (1):236.

Richardson, D. A., S. Leeuw, and W. Dullaert. 2016. "Factors Affecting Global Inventory Prepositioning Locations in Humanitarian Operations-A Delphi Study". *Journal of Business Logistics* 37(1):59 – 74.

Schluse, M., M. Priggemeyer, L. Atorf, and J. Rossmann. 2018, April. "Experimentable Digital Twins 8212;Streamlining Simulation-Based Systems Engineering for Industry 4.0". *IEEE Transactions on Industrial Informatics* 14(4):1722–1731.

Venkatapathy, A. K. R., H. Bayhan, F. Zeidler, and M. ten Hompel. 2017. "Human machine synergies in intra-logistics: Creating a hybrid network for research and technologies". In *2017 Federated Conference on Computer Science and Information Systems (FedCSIS)*, 1065–1068. Prague, Czech Republic, 3 - 6 September, 2017.

Xu, B., L. D. Xu, H. Cai, C. Xie, J. Hu, and F. Bu. 2014. "Ubiquitous Data Accessing Method in IoT-Based Information System for Emergency Medical Services". *IEEE Transactions on Industrial Informatics* 10(2):1578–1586.

AUTHOR BIOGRAPHIES

VLADEMIR F. SANTOS is a doctoral candidate at School of Electrical and Computer Engineering, Unicamp. He received a MSc degree in Technology at the School of Technology, Unicamp, Brazil. He works with logistics and discrete-event simulation. His email is vlafazio@gmail.com.

EDSON L. URSINI is an Assistant Professor at Unicamp. He received a doctoral degree in Electrical Engineering from Unicamp, Brazil. His research deals with data networks, stochastic processes, simulation and performance analysis. His email is ursini@ft.unicamp.br.

PAULO S. MARTINS is an Associate Professor. He received a PhD degree in Computer Science from The University of York, UK. He works with real time systems as well as communications network design and analysis. His email is paulo@ft.unicamp.br.

MICHEL D. YACOUB received the Ph.D. degree in Electrical Engineering from The University of Essex, UK, in 1988 and joined School of Electrical and Computer Engineering, UNICAMP, in 1989, where he is currently a Full Professor. His current research interests include wireless communications. His is an associate editor of the Journal of Vehicular Transactions Technology (VTT). His email is michel@decom.fee.unicamp.br.