Proceedings of the 2001 Winter Simulation Conference
B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, eds.

STAGING QUEUES IN MATERIAL HANDLING AND TRANSPORTATION SYSTEMS

Kevin R. Gue

Graduate School of Business & Public Policy
Naval Postgraduate School
Monterey, CA 93943, U.S.A.

Keebom Kang

Graduate School of Business & Public Policy
Naval Postgraduate School
Monterey, CA 93943, U.S.A.

ABSTRACT

In most physical queueing applications, customers join a queue and move forward after each service, leaving room for others to join behind them. Some queues found in material handling and transportation systems do not operate like this because the queued entities (pallets or unoccupied cars, for example) are incapable of moving forward autonomously. We develop a model for the resulting *staging queue*, and give simulation results for several configurations.

1 STAGING SYSTEMS

In most finite queueing applications, a customer may join the queue as long as the number of customers in the queue is less than the number of positions in the queue. In a physical system, this is true because customers move forward after each service, leaving room for new customers to join.

Some queues in material handling and transportation systems operate differently because customers in the queue do not move forward after each service. One example occurs in rental car lots: typically there are 3–5 lanes set aside for returning cars, and arriving customers must park in the rearward-most space in one of the lanes. As each car is served, an attendant drives away the forward-most car in a lane, but cars in the rear do not advance because they are unoccupied.

Another common example is shipping areas in warehouses, which typically have lanes in front of dock doors for pallets to queue up. As the forward-most pallet is served (removed) and loaded onto a truck, the remaining pallets do not automatically advance and no additional room is made for pallets to join the queue. Crossdocks in the retail distribution industry are an extreme case, in which the entire facility (in some cases) is dedicated to staging lanes for receiving and shipping (see Figure 1). We have seen both single-stage crossdocks, with one set of staging queues, and two-stage crossdocks, with two (see Figure 2).

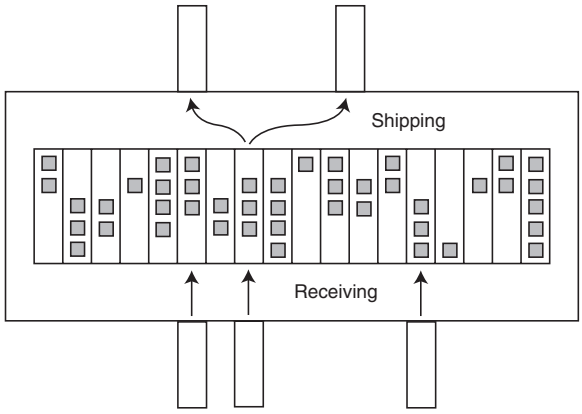

Figure 1: A single-stage crossdock. Workers on the receiving side put pallets in lanes corresponding to the receiving doors; on the shipping side, workers deliver pallets to their appropriate doors.

Bartholdi, Gue, and Kang (2001) introduce an analytical model for staging queues, which our work extends. They use their model to compare a staging queue with flow rack, which operates as a "standard" queue, and show that staging queues block more often, but not significantly so. We know of no other work on staging queues, by that or any other name. Work on crossdocking systems includes Tsui and Chang (1990), Tsui and Chang (1992), Gue (1999), Bartholdi and Gue (2000a), and Bartholdi and Gue (2000b).

We use a simulation model to investigate three areas: We consider the performance of parallel staging queues, such as might be found in a rental car return area or a container staging area in military amphibious operations (more on this later). We also investigate the behavior of tandem staging queues, which are found in some retail crossdocks. Finally, we develop results for a closed system, in which arrivals wait to join a blocked queue rather than balk. We conclude with some suggestions for designing systems of staging queues.

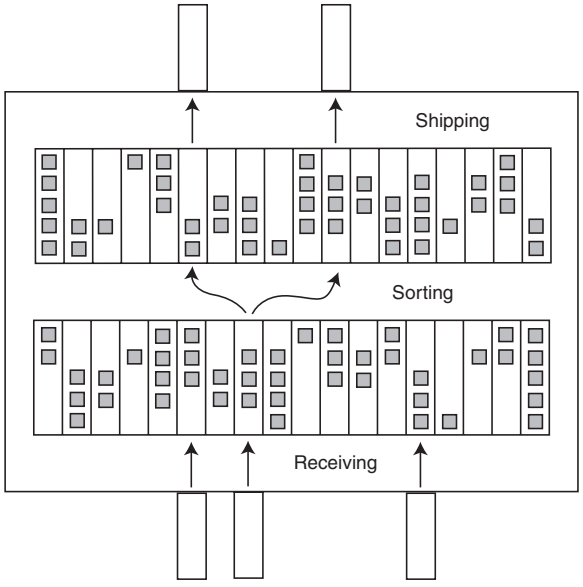

Figure 2: Representation of a two-stage crossdock. Workers put pallets in lanes corresponding to the receiving doors; a second team of workers sorts pallets into shipping lanes, from which a final team loads them onto outbound trailers.

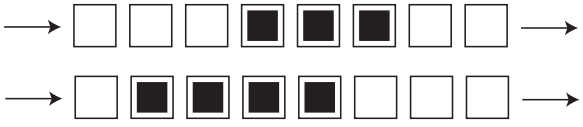

Figure 3: How a staging queue works: in the top illustration, customers (e.g. pallets or cars) occupy positions 3–5. The bottom illustation shows the system state after 2 arrivals and 1 service: positions 4–7 are occupied.

2 A MODEL FOR STAGING QUEUES

We assume that arrivals to a staging queue occupy the forward-most empty position, and servers serve the forward-most entity from the other side (see Figure 3). Note that entities in the queue must be contiguous and that the backward movement of the block of entities forms a *wave*, which propagates backward and either "breaks early" (meaning that it never reaches the last position) or "beaches" (it eventually blocks the queue until cleared).

We assume that arrivals balk if they find the queue full, even though it may not be realistic in many instances. Since analytical approaches are feasible in some cases with balking assumptions, we can compare the simulation with analytical results.

We built a simulation of staging queues with the simulation package ARENA (Kelton, Sadowski, and Sadowski 1998). We assume that interarrival and service times (from a single server) are exponentially distributed, with means $1/\lambda$ and $1/\mu$ respectively, where $\lambda + \mu = 1$. (We relax the single server assumption in Section 5.)

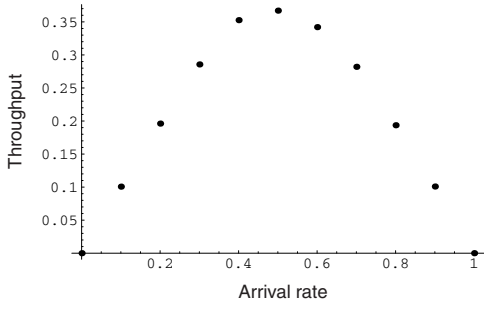

Figure 4: Throughput for a 3-position staging queue as arrival rate λ and service rate μ vary, with $\lambda + \mu = 1$.

Figure 4 shows the throughput for a 3-position staging queue as λ and μ vary. The figure suggests that throughput is greatest when λ is approximately equal to μ, and so we assume that $\lambda = \mu = 0.5$ throughout.

3 PARALLEL QUEUES

One potential application of staging queues occurs during a military amphibious operation called an *instream offload*, in which containers are transferred from ships directly to the beach and staged in a *marshalling area* (Kang and Gue 1997), from which they are transported to their final destinations. The marshalling area is essentially a two-dimensional staging queue, which we approximate with a set of parallel staging queues. How should one configure such a set? For example, is it better to have 5 queues, each with 10 positions, or 10 queues, each with 5 positions?

To gain insight, we consider a system in which arrivals and servers choose between queues at random and observe that,

Result 1 *When using a random choice rule for arrivals and the server, it is better to have fewer, long staging queues than more, short ones.*

Proof sketch. Consider a single staging queue with n positions, where n is even, arrival rate λ and service rate μ. The throughput is $\lambda_{\text{eff}} = (1 - P_n)\lambda$, where P_n is the long-run probability that a staging queue of n positions is blocked.

Consider a second system to which entities arrive at rate λ, then randomly choose one of two parallel staging queues, each with $n/2$ positions. Randomly selecting the arrival queue retains Poisson arrivals to each queue, but at arrival rate $\lambda/2$ to each. A similar service rule yields Markovian service with rate $\mu/2$ for each queue. Throughput for the new system is $\lambda_{\text{eff}} = 2(1 - P_{n/2})(\lambda/2) = (1 - P_{n/2})\lambda$. Because the blocking probability is smaller for a longer

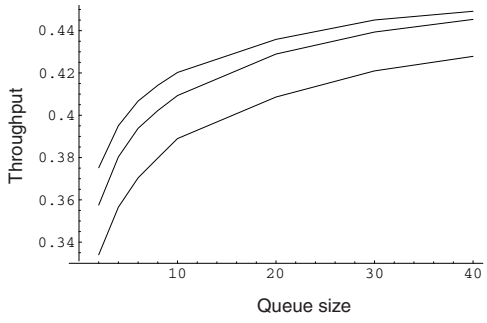

Figure 5: A comparison of parallel systems against a single queue. The single queue system curve is in the middle; the parallel system is better when using the nearest position rule (top curve) and worse when randomly selecting between queues (bottom curve).

queue, the system having a single queue of n positions has higher throughput. □

In practice, of course, arrivals do not randomly select between two staging queues. Instead, they choose the queue that seems to make the "most sense." We formalize this notion with the following *nearest position rule*: an arriving customer chooses the queue having the forward-most open position; servers serve from the queue having the forward-most occupied position.

We implemented the nearest position rule in the simulation model with the following exception. Under the nearest position rule, both queues tend to maintain approximately the same length of "wave" moving at approximately the same speed. When the last positions of both queues are occupied, arrivals are blocked until one of the queues clears. After one queue clears, the next arrival takes the first position of the empty staging queue while the last position of the other queue is still occupied. The server then serves the last positioned item instead of the nearest one (the first position in this case) to clear the second queue.

Figure 5 compares the throughput for parallel staging queues against a single queue with the same number of positions. The parallel system in which arrivals and the server randomly choose the queue has lower throughput than a single queue, as Result 1 suggests. The parallel system using the nearest position rule, however, demonstrates higher throughput than the single staging queue, which leads to the conclusion that,

Result 2 *When using the nearest position rule, it is better to have more, short staging queues than fewer, long ones.*

The result implies that rental car firms, for example, should design their lots with more short return lanes rather than fewer long ones, and military planners should configure their staging areas "wide and shallow" rather than "narrow and deep."

Also, note that in the extreme case of n parallel queues having a single position, the system of staging queues behaves exactly like a single standard queue having n-positions, only it is "turned on its side."

4 TANDEM QUEUES

In their purest form, crossdocks in the retail distribution industry transfer freight directly from inbound to outbound trailers, and the freight never touches the floor. In practice, this seldom happens and there is at least some bit of staging because either the freight needs some processing, such as labelling, or loading directly does not provide a tight enough pack for outbound vehicles. Figures 1 and 2 illustrate two models for material movement for crossdocking with staging.

For a single stage system, the firm can stage by receiving door (as in Figure 1) or stage by shipping door. The advantage to staging by receiving is that the destination need not be known when the worker unloads the freight from the trailer. This relieves the vendor of the burden of labelling pallets before shipping them. Retailers are working on relationships with vendors to achieve this level of coordination, but currently it is rare. The advantage of staging by shipping is that workers in shipping have a better view of what freight is available for loading, and so can achieve a tighter pack of freight while loading, thus reducing transportation costs in the long run.

A two-stage system achieves *both* advantages, but at what cost? To gain insight into this question, we simulated a tandem staging queue system in which departures from the first queue become arrivals to the second. We ran two scenarios: in the first, arrivals to the second queue balk if it is full; in the second, the server for the first queue is blocked until the second queue is cleared. In each scenario, we set $\lambda = \mu_1 = \mu_2 = 0.5$, where μ_1 and μ_2 are the mean service rates for queues 1 and 2 respectively.

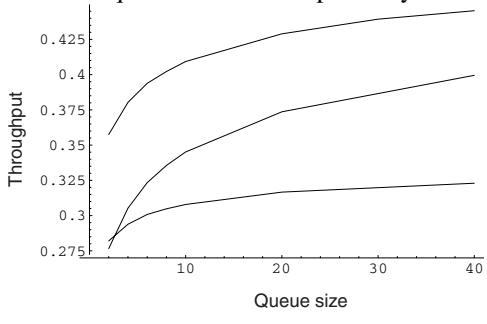

Figure 6: A comparison of tandem systems against a single queue. The single queue system (top curve) has significantly higher throughput than the tandem systems with balking or blocking (middle and bottom curves, respectively).

Figure 6 compares a single queue with n positions with tandem systems (each queue having $n/2$ positions) for the blocking and balking cases.

Result 3 *A two-stage staging system has significantly lower throughput than a single-stage system when entities block between stages.*

The implication for crossdock design is that, while a two-stage system offers the dual advantages of staging by receiving and by shipping, these advantages come at a cost of lower throughput. In practice this would be realized with higher levels of congestion as throughput increases, or with higher labor costs.

5 A CLOSED SYSTEM

5.1 Blocking

In the real applications that we know of, customers to a staging queue do not balk when a staging queue is full. Instead, they block, and perhaps wait, while the queue clears. For example, in a rental car return lot, customers may back onto the street (this has happened to one of the authors); or, in a crossdock, a forklift driver may put his load aside while he helps clear the blocked lane.

To model this phenomenon, we assume that arrivals block and wait upon finding a full queue. In the simulation, an arriving worker waits indefinitely until a queue position is available. (We used the WAIT and SIGNAL blocks in ARENA.)

The results showed that,

Result 4 *A staging queue has higher throughput when arrivals block than when they balk.*

The intuition is that after a blocked queue clears, blocked workers immediately deposit their loads into the queue and the server has work; in the case of balking, the server must wait for a next arrival.

5.2 Effects of multiple servers

In most queueing applications, a firm controls the service rate, but arrivals are exogenous. For example, a rental car firm can control the service rate of returning vehicles by staffing the service attendants, but it has no direct control over the arrival rate or distribution. In crossdocking applications, however, the firm controls both the arrivals, by allocating workers to receiving, and service, by allocating workers to shipping. Typically, managers strive to balance these two rates by allocating workers appropriately.

How does the allocation of workers to receiving and shipping affect the throughput of a staging queue? It is well-known that customers in an infinite capacity $M/M/1$ system have shorter expected cycle time (i.e., total time in the system) than in an $M/M/s$ system, when the total mean service rates are the same; or, anecdotally, it is better to have one fast server than two slow ones. We consider a similar question for staging queues with regard to throughput.

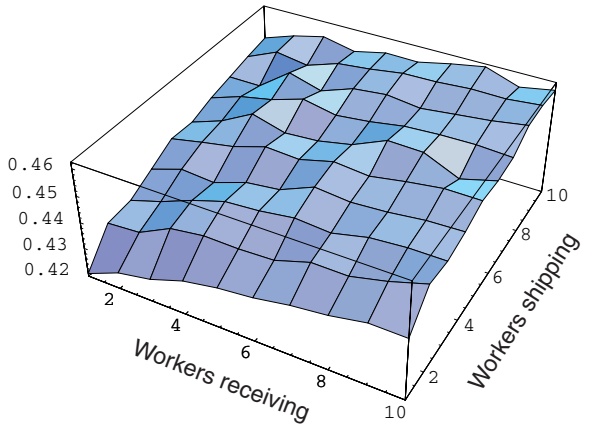

Figure 7: Throughput for different combinations of workers in receiving and shipping. Throughput increases with more workers on either side of a staging queue.

We simulated a crossdocking system with n workers in receiving (receivers), each having rate λ/n, and m workers in shipping (shippers), each with rate μ/m (i.e., the total rates are λ and μ, respectively). When the queue is blocked a receiver waits until the queue is cleared. A maximum of n receivers may wait in the queue. After a blocked queue clears, up to c (the queue capacity) receivers can deposit their entities.

Our results showed that,

Result 5 *For a finite staging queue, throughput is higher for a system with more receivers or more shippers.*

The intution behind the result is that a system with more servers has more customers being served on average, and so the queue is blocked less often. Less blocking means that workers in receiving are able to retrieve more loads over time and throughput is higher (see Figure 7). A system with more receivers has higher throughput because more workers are released into the queue after every blocking cycle.

6 CONCLUSIONS

While our modeling assumptions, such as exponential inter-arrival and service times, are unrealistic for many practical situations, we believe our results give some insight into the behavior of different configurations of staging queues.

For systems of parallel staging queues, it is better to have more short queues than fewer long ones, but only when using a reasonable placement and service rule. This suggests that queueing systems of empty vehicles, such as

rental car return lanes or the staging area in an automobile mixing center, should have as many queues as possible.

Our results also suggest that two-stage crossdocking systems, while having important operational advantages, do suffer significantly lower throughput than an equivalent single-stage system. The operations manager at one two-stage crossdock we visited stated that they would happily operate a single-stage system, were they able to establish the necessary information links with all of their vendors.

Finally, we found that, all other things being equal, more workers is better, both on the receiving and shipping sides of a staging queue.

REFERENCES

Bartholdi, J. J., and K. R. Gue. 2000a. Reducing Labor Costs in an LTL Crossdocking Terminal. *Operations Research* 48 (6): 823–832.

Bartholdi, J. J., and K. R. Gue. 2000b. The Best Shape for a Crossdock. *Working Paper*.

Bartholdi, J. J., K. R. Gue, and K. Kang. 2001. Staging Freight in a Crossdock. In *Proceedings of the International Conference on Industrial Engineering and Production Management*, forthcoming.

Gue, K. R. 1999. The Effects of Trailer Scheduling on the Layout of Freight Terminals. *Transportation Science* 33 (4): 419–428.

Kang, K., and K. R. Gue. 1997. Sea Based Logistics: Distribution Problems for Future Global Contingencies. In *Proceedings of the 1997 Winter Simulation Conference*, 911–916.

Kelton, W. D., R. P. Sadowski, and D. A. Sadowski. 1998. *Simulation with arena*. First ed. Boston, Massachusetts: McGraw-Hill.

Tsui, L. Y., and C.-H. Chang. 1990. A Microcomputer Based Decision Support Tool for Assigning Dock Doors in Freight Yards. *Computers in Industrial Engineering* 19:309–312.

Tsui, L. Y., and C.-H. Chang. 1992. Optimal Solution to a Dock Door Assignment Problem. *Computers & Industrial Engineering* 23:283–286.

AUTHOR BIOGRAPHIES

KEVIN GUE is Assistant Professor of Logistics in the Graduate School of Business & Public Policy at the Naval Postgraduate School, Monterey, CA. He graduated from the U.S. Naval Academy in 1985 and served as an officer in the submarine community for 5 years. He received his Ph.D. in Industrial Engineering at Georgia Tech in 1995. His research interests include warehousing, distribution, and logistics modeling. He is currently supported by the Office of Naval Research (N00014-00-WR-20244).

KEEBOM KANG is Associate Professor in the Graduate School of Business & Public Policy at the Naval Postgraduate School. He received a B.S. in Industrial Engineering from Seoul National University, an M.S. in Operations Research from the University of Texas at Austin, and a Ph.D. in Industrial Engineering from Purdue University. His research interests are in the areas of logistics and simulation modeling. He was the Director of the OR Division of the Institute of Industrial Engineering (1996-1997), Co-Editor of the *1995 Winter Simulation Conference Proceedings*, and Program Chair of the 2000 Winter Simulation Conference.