MANAGEMENT SCIENCE
Vol. 50, No. 7, July 2004, pp. 949–966
ISSN 0025-1909 | EISSN 1526-5501 | 04 | 5007 | 0949

informs

DOI 10.1287/mnsc.1040.0251

Self-Interested Routing in Queueing Networks

Ali K. Parlaktürk, Sunil Kumar
Graduate School of Business, Stanford University, Stanford, California 94305
{pturk@stanford.edu, skumar@stanford.edu}

We study self-interested routing in stochastic networks, taking into account the discrete stochastic dynamics of such networks. We analyze a two-station multiclass queueing network in which the system manager chooses the scheduling rule and individual customers choose routes in a self-interested manner. We show that this network can be unstable in Nash equilibrium under some scheduling rules. We also design a nontrivial scheduling rule that negates the performance degradation resulting from self-interested routing and achieves a Nash equilibrium with performance comparable to the first-best solution.

Key words: queueing networks; routing; scheduling; Nash equilibrium; mechanism design; stability
History: Accepted by Paul Glasserman, stochastic models and simulation; received January 29, 2003. This paper was with the authors 5 months for 3 revisions.

1. Introduction

In many operations, especially in the service sector, the sequence of tasks is chosen by the customers themselves rather than the system manager. For example, in an amusement park, a customer chooses the order in which she visits various attractions. In choosing this order, she is influenced not only by factors under the system designer's control, such as the layout of the park, but also by the relative congestion levels at the various attractions, which are determined by the decisions of other customers. From the point of view of congestion, the behavior of the amusement park is a result of the complex interaction between the decisions taken by the manager of the park as well as the decisions of the various customers, where a customer's decision is based on her self-interest and her beliefs about the behavior of other customers. Recently, software packages have become available that help customers plan their routes through amusement parks providing historical wait-time statistics at various attractions in the park. Even more common examples of customers choosing their "routes" through a service operation are the call centers equipped with voice response units. Consider the situation where a customer needs to perform multiple operations, such as checking balance and transferring money. The voice response unit allows the customer to choose the order in which the multiple operations will be performed and thus choose her route, in the same way a driver chooses her route through a network of roads. While other service operations may not offer such flexibility to their customers, the rigidity of routing is not always a consequence of technological or procedural constraints. For instance, at the Department of Motor Vehicles (DMV) there is no reason that all the tasks required to process a license application should be done in one fixed order by the DMV employees. The so-called "first best" is achieved when the tasks are routed and scheduled by a single decision maker who has the sufficient "sophistication." On the other hand, it is conceivable that relaxing rigid routing schemes and allowing customers to choose their own routes through the operations would improve customer satisfaction while maintaining good system performance, perhaps even outperforming some rigid suboptimal routing schemes. After all, the natural goal of a self-interested customer, such as minimizing her individual delay, is not completely antagonistic to the goals of the system manager, such as minimizing the average customer delay (which in turn minimizes the average number of customers waiting in the system).

Despite the natural fit of self-interested routing in transportation networks and service operations, self-interested routing also may be applicable in manufacturing. For example, in a new rapid prototyping technology called Shape Deposition Manufacturing (SDM) (Pinilla and Prinz 2003), products are built by alternately shaping and depositing the material. A product of a given geometry can be made using different sequences of shaping and deposition. As a key performance metric in the prototyping industry is the response time, and as the customers tend to provide detailed specifications of the product and even the process to be used, it is conceivable that the manager of an SDM shop would allow the customers to

exploit the flexibility of SDM to their own advantage by permitting them to specify the actual sequence of deposition and shaping steps on their orders. Thus, the customers would effectively choose their "routes" through the SDM shop in their self-interest.

In this paper we model the operations described above as so-called *processing networks* that consist of networks of stations through which the customers flow. The customers require sequences of tasks corresponding to routes through the network. We study self-interested routing in these processing networks taking into account the discrete stochastic dynamics of such systems. Individual customers selfishly choose their routes through the network so as to minimize their individual expected delay, based on the announced scheduling rule and the observed state of the network. Furthermore, we study the use of scheduling rules as a way of discriminating and inducing desired behavior among the self-interested customers. The system manager chooses the scheduling rule at each station that determines the order in which the customers awaiting service will be served. The goal of the system manager is to minimize the expected number of customers waiting in the system. We study the behavior of this system in Nash equilibrium, in which there are no benefits to unilateral deviation. That is, if all customers except one follow the Nash strategy, the deviating customer would incur an expected delay that is at least as long as the delay she would incur had she chosen the Nash prescription. A consequence of this equilibrium is that the decisions made by *future* arrivals also need to be taken into account by the customer.

We study a network that is just sufficiently complex to render the interaction between the scheduling rule chosen by the system manager and the routing strategies chosen by the customers nontrivial. We consider the simplest symmetric system that consists of multiple stations, in which the customers are allowed to choose between two sequences of operations (or two routes as described in above the examples). While we model customer arrivals to the network as random, the service times are deterministic, primarily for analytical tractability. Not without loss of generality, service times are assumed to be symmetric across routes. The service time depends only on whether the task being performed is the first or the second along a route, and not on the route chosen. These assumptions allow sample-path analysis of the equilibrium. For part of the paper, we also assume some ordering in the service times (cf. Assumption 3), but we relax this assumption in §7. As will be evident in §3, the assumptions we make about our model are quite stringent. As it is not the intention of the authors to provide a general tool for the analysis or design of networks under self-interested routing and as the assumptions allow us to illustrate phenomena in a simple and direct manner, we retain them in the current form. While it is conceivable that the network shown in Figure 1 could be a simplified model of an actual operation such as SDM, the reader will be best served by thinking of this model as the simplest network model that is rich enough to reproduce the variety of interactions—both between customers as well as between processing stations—that one could expect in a general network.

An important question is whether the network under consideration, and networks in general, are always stabilized by the routing choices of smart but self-interested customers regardless of the announced nonidling scheduling policy. To this end, is it possible for the system manager to choose a scheduling rule that results in an unstable Nash equilibria? That is, could self-interested customers minimizing their individual delays cause the queue lengths to grow without bound, even when the network has sufficient capacity to handle all the customers? The answer turns out to be affirmative, which we address in §4. We study two common scheduling rules, characterize the corresponding equilibrium routing strategies followed by the customers, and show that the system behaves like an unstable $GI/D/1$ queue after some time in each of these cases.

The second issue addressed in this work is the design of a scheduling rule that negates the performance degradation from self-interested routing. At the very least, we wish to find a scheduling rule that induces a Nash equilibrium that is stable in the queueing sense. We would also like the performance of the scheduling rule in equilibrium to be as close as possible to the performance of the optimal centralized control, at least in some asymptotic regime. For the network under consideration, we propose such a scheduling rule in §5 and characterize the corresponding Nash routing strategy. The sample-path proof that the conjectured routing strategy is indeed Nash is presented in §6. Using fluid limit techniques (Dai 1995, Dai and Prabhakar 2000), we prove that the network is stable in equilibrium. Furthermore, we simulate the network and compare the performance under the proposed policy against that of commonly used policies, and the linear-programming-based lower bounds of Kumar and Kumar (1994). In addition, we provide evidence that allows us to conjecture that the proposed scheduling rule and the resulting Nash equilibrium is asymptotically optimal in heavy traffic, based on Brownian approximations (Harrison 1988, Harrison and Van Mieghem 1997). We stop short of providing a complete proof of this result. In passing, it is worth mentioning that we also provide an example of a scheduling policy that induces a Nash equilibrium that falls between the two extremes of instability and asymptotically optimal performance. Finally,

in §7 we consider extensions of our model in which we relax the assumptions on the order of service times and the network topology and provide examples of both instability as well as asymptotically optimal performance. This illustrates that the key insights of the paper are not artifacts of these assumptions and can extend to more general settings.

To summarize, our contribution in this paper is twofold. First, for the network under consideration, we establish that it can be unstable under self-interested routing for some common scheduling rules, even though it can be stabilized under centralized routing. One interpretation of this result is that these scheduling rules exacerbate the alignment "gap" between the objective of the system manager (minimizing the average number of customers waiting in the system) and the objectives of the customers (minimizing their individual delay). Second, we establish that scheduling rules can be designed to mitigate the negative influence of self-interested routing that better aligns these objectives. That is, we show that it is possible to design a mechanism, in the form of a scheduling rule, that leaves routing to the customers and yet obtains excellent performance in a system in which a simpler scheduling rule may induce instability. Moreover, we prove that the suggested scheduling rule induces a dominant routing strategy in most states of the system, which ensures robustness of the proposed policy: There are no benefits to deviation by a customer even when other customers deviate. This work illustrates that the scheduling rule determines performance of the system under self-interested routing in two parts. First, it induces a routing rule and, second, it intricately interacts with the induced routing rule that determines the overall system performance. This paper provides the first steps towards a better understanding of this interaction.

2. Related Literature

The literature involving economic and game-theoretic reasoning in queueing systems is extensive and well established. In what follows, we sketch the flavor of this literature; it is not our intention to be exhaustive.

The study of self-interested routing in networks has a long history dating back several decades (cf. Beckmann et al. 1956). It is well-known that the overall performance of a network deteriorates when the customers are allowed to choose routes in their self-interest, with the most famous result in this vein being the so-called Braess paradox (Braess 1968). Braess showed that adding a new link to a network can make every customer worse off. Kelly (1991) provides a very readable introduction to this literature, and Roughgarden and Tardos (2002) illustrate recent results in this vein. This phenomenon is a special case of a more general phenomenon where noncooperation among decentralized decision makers results in social inefficiency (Fudenberg and Tirole 1991, Dubey 1986). The papers (Beckmann et al. 1956, Braess 1968, Cohen and Kelly 1990, Kelly 1991, Roughgarden and Tardos 2002) invariably make one or more of the following three assumptions. First, they assume that the relationship between the amount of flow along a link and the resulting delay observed by the customer is given by a deterministic static relationship. In some cases this relationship is taken from the steady-state analysis of the underlying queueing dynamics (Cohen and Kelly 1990, Kelly 1991). Second, these papers assume that the self-interested customers ignore the impact of their decisions on the network and choose routes in such a way to achieve a so-called Wardrop equilibrium. Third, these papers assume that all customers at a resource are homogeneous and that the resource does not discriminate among customers, for example, via a scheduling rule. Finally, many of these papers do not study the normative question of how to induce a desired behavior among customers.

In his seminal work, Naor (1969) uses pricing as a tool to achieve social optimum and avoid performance degradation from the self-interested entry. In his model, customers decide to enter a single queue based on the observed congestion and on the toll levied. The system manager sets the toll to induce desired behavior. Other papers have explored different models for customer behavior in simple queueing systems. Mendelson and Whang (1990) assume a distribution for willingness to pay among customers and conduct a similar analysis. A similar approach to determining the socially optimal prices, along with a LIFO scheduling rule, is studied in Bell and Stidham (1983). In addition to prices, scheduling rules are employed to induce desired behavior in some of the papers. For example, Adiri and Yechiali (1974) employ a priority scheduling rule and allow arriving customers to purchase entry into the priority class at a price chosen by the system manager. A recent paper that studies the interplay of scheduling rules and pricing is Van Mieghem (2000). In most of these papers, the underlying queueing system is quite simple, usually consisting of a single server. In such simple systems, choice of routes is either unavailable or essentially trivial, and the customer's decision is based on average queue lengths, not on the current state. Furthermore, prices are used only to control admission into the system and are not used to control routing choices. In a network setting, it is not realistic to assume that different (state-dependent) prices can be set for *every* task along *every* route. For example, in a call center with a "call-back" option, it is not possible to charge different prices to customers who choose to wait in line when compared to those who want a

call back to complete the same service. Armony and Maglaras (2004) study such a system and use scheduling as a way to optimally control customer choices. In our paper, we also use scheduling policies, rather than price discrimination, to optimally control customer choices.

Our work extends the body of work cited above in many ways. First, we study self-interested routing in the context of stochastic networks, taking into account the discrete stochastic dynamics of such networks. That is, we do not model the relationship between flows and observed performance as static or deterministic. Rather, we keep track of the progress of each customer through the network. Second, we assume that the customers are aware of the impact of their own decisions on the network. This modeling framework is similar to that used by Koutsoupias and Papadimitriou (1999), where a single-resource system was studied. We study the network under symmetric Nash equilibrium in which each customer uses the same routing strategy that depends on the observed state of the system and, under the equilibrium, there are no gains to unilateral deviation. The equilibrium concept is the same as in Altman and Shimkin (1998), who consider a queueing system with no routing control. Furthermore, we do not concentrate solely on performance degradation as Beckmann et al. (1956), Braess (1968), Cohen and Kelly (1990), Kelly (1991), Roughgarden and Tardos (2002) do. Rather, we also study the possibility of inducing equilibria with excellent performance employing scheduling rules as mechanisms.

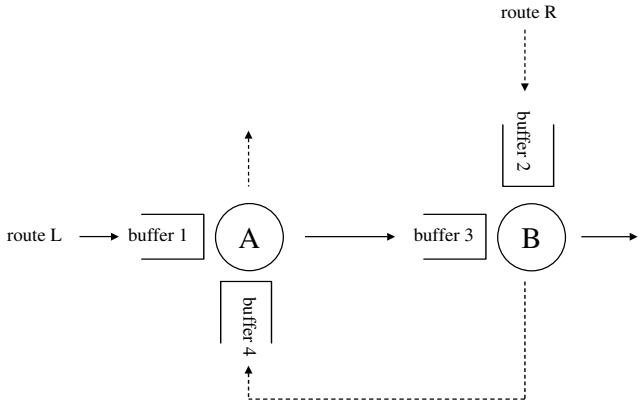
The network topology considered in this paper (cf. Figure 1) is similar to a network, the so-called the Rybko-Stolyar network (1992), with a history of interesting behavior. The preemptive static priority scheduling rule that gives priority to the customers who have fewer remaining tasks has been shown to cause instability under fixed routing (Rybko and Stolyar 1992). However, ours is the first result to establish the instability phenomenon under self-interested routing. Furthermore, we establish instability under the first-in-system-first-out (FIFO) scheduling rule, for which there are no other existing instability results. The mechanism of instability in Rybko and Stolyar (1992) is one of alternate blocking and starvation, which is described further in §4. Earlier papers (Kumar and Seidman 1990, Lu and Kumar 1991) essentially describe networks with the same phenomenon but under different assumptions. In our paper, the mechanism of instability is not one of alternate blocking and starvation, rather, it is one of traffic not splitting evenly across the two routes. Our result is consistent with, but stronger than, the result of Roughgarden and Tardos (2002) in establishing that the performance degradation under self-interested routing can be arbitrarily bad, because it establishes the actual instability. Moreover, we illustrate that the join-the-shortest-queue routing rule is unstable in this network, reinforcing the earlier results of Sharifnia (1997) and Whitt (1986).

Finally, a note on the asymptotic analysis used in the paper is in order. A problem with most queueing networks is that neither the performance under a given policy nor the choice of the optimal policy is easy to determine. This is true for the network considered in this paper as well. As a result, asymptotic techniques (Dai 1995, Harrison 1988, Harrison and Van Mieghem 1997) and bounds (Kumar and Kumar 1994) are commonly used to obtain performance guarantees and policy prescriptions. We use one such technique, namely the fluid limit analysis pioneered by Dai (1995), to establish the stability of the system under the proposed policy and use the bounds of Kumar and Kumar (1994) to provide a performance benchmark. Of late, quite a few papers have begun to study equilibria and strategic behavior in asymptotic models (Armony and Maglaras 2004; Maglaras and Zeevi 2003, 2005). In contrast to these, we characterize the strategic behavior in the exact model rather than in an asymptotic model, and only use the asymptotic analysis to evaluate performance. A recent paper in this vein is Whitt (2003), where the performance of a queueing system without routing control operating under congestion-dependent demand is studied in an asymptotic regime.

3. The Model

The Network. The queueing network consists of two servers, A and B as shown in Figure 1. The customers arrive at network according to a renewal process with rate λ. That is, the times between successive arrivals are independent and identically distributed (i.i.d.) with mean $1/\lambda$. Each arriving customer requires two stages of service, one from each server. The customer chooses between having server A carry out the first stage of service and server B carry out

Figure 1 The Network



the second (or equivalently taking route L through the network), or having server B carry out the first stage of service and server A carry out the second (or taking route R through the network). Route L customers wait in buffer 1 at server A and in buffer 3 at server B, and route R customers wait in buffer 2 at server B and in buffer 4 at server A.[1] We will call buffers 1 and 2 *entry buffers*, and buffers 3 and 4 *exit buffers*. Service times are deterministic, and the first service takes m_f time units, regardless of the server involved (either at buffer 1 or at buffer 2). The second service takes m_s time units, regardless of the server involved (either at buffer 3 or at buffer 4). Although the service times are assumed to be symmetric, it is important to note that services at servers A and B are different and not interchangeable. We treat a customer who has partially completed service as a fractional customer as follows. If (r_1, r_2, r_3, r_4) denote the residual service times and (B_1, B_2, B_3, B_4) denote the queue lengths (including the customer in service) at each of the four buffers, the *augmented queue lengths* are defined to be $Q_i = B_i - 1 + r_i/m_f$ for $i = 1, 2$ and $Q_i = B_i + r_i/m_s - r_{i-2}/m_f$ for $i = 3, 4$. As will become obvious in what follows, the vector of augmented queue lengths $Q = (Q_1, Q_2, Q_3, Q_4)$ serves as the system state descriptor. Hereafter we will simply say queue lengths to refer to the augmented queue lengths Q, for brevity.

Scheduling Rules and Service Mechanism. The system manager (SM) chooses a scheduling policy T that determines which buffer receives a server's attention at different states. We only consider stationary scheduling policies and allow for splitting of the server's effort. To be precise, $T: R_+^4 \to [0, 1]^4$, where $T_i(Q)$ denotes the fraction of the appropriate server's effort expended on buffer i when the state is Q. The set of feasible scheduling policies F is specified by

$$F = \{T: \forall Q \in R_+^4,\ T_1(Q) + T_4(Q) \le 1 \text{ and } T_2(Q) + T_3(Q) \le 1\}. \quad (1)$$

Furthermore, we require that the scheduling policy be in the set of nonidling policies D, formally defined by

$$D = \{T \in F: Q_1 + Q_4 > 0 \Rightarrow T_1(Q) + T_4(Q) = 1 \text{ and } Q_2 + Q_3 > 0 \Rightarrow T_2(Q) + T_3(Q) = 1\}. \quad (2)$$

Restricting the SM to nonidling policies (to the set D) prevents the SM from acting like a dictator. Otherwise the SM can impose severe threats, such as not to serve a customer at all when she does not conform, and by means of these threats, any behavior can trivially be induced by the SM.

[1] These buffers need not be physically separated from each other.

The customers within a buffer are processed in a FIFO manner, with service effort given only to the customer at the head of the line. A customer departs from her current buffer when she receives server effort for a period of time equal to the service requirement (either m_f or m_s). For simplicity of analysis, we assume that server B can work on the processed part of a route L customer in buffer 1 if server effort is given to buffer 3 even though the customer is not physically present in buffer 3. The same holds symmetrically for the customers on route R. This assumption is in effect only if one of the exit buffers is empty and even then it only affects the first customer. Consequently, it does not change the behavior of the system substantially, while it does improve analytical tractability considerably.

Customer's Routing Choice. An incoming customer observes the number of customers in each buffer (and, consequently, their routing choices) and the residual service times. Furthermore, the scheduling policy T is known by every customer. Based on all this information, the customer picks her route upon arrival to minimize her expected time until her departure from the network. Since the network is not nonovertaking in general, she needs to incorporate future arrivals in her decision making. So she conjectures that the future customers also play symmetric Nash equilibrium strategies and decides accordingly. Such use of the Nash equilibrium as the basis of decision making is fairly standard in economics (Fudenberg and Tirole 1991).

Let Q^j denote the vector of queue lengths observed by the jth customer, $\phi_j \in \{L, R\}$ denote her routing decision and W_j denote her sojourn time. A symmetric routing strategy is prescribed by a mapping $\Phi: Q \to \{L, R\}$. For a given scheduling policy T, a Nash routing strategy Φ_T maps the information set of an incoming customer j to one of the two routes such that

$$\begin{aligned} &E[W_j \mid T, Q^j, \phi_j = \Phi_T(Q^j), \\ &\quad \phi_{j+1} = \Phi_T(Q^{j+1}), \phi_{j+2} = \Phi_T(Q^{j+2}), \cdots] \\ &\quad \le E[W_j \mid T, Q^j, \phi_j = \overline{\Phi_T(Q^j)}, \\ &\qquad \phi_{j+1} = \Phi_T(Q^{j+1}), \phi_{j+2} = \Phi_T(Q^{j+2}), \cdots], \end{aligned} \quad (3)$$

where $\overline{\Phi_T(Q^j)}$ denotes the complementary decision of $\Phi_T(Q^j)$. In other words, there are no gains to unilateral deviation from the Φ_T Nash routing strategy by the jth customer. W_j can be explicitly computed if the future arrival sequence is known and the conditional expectation above is taken with respect to the future arrival times. This equilibrium concept is very similar to the one in Altman and Shimkin (1998). There are two issues that arise with the framework

described above. First, we do not know whether a function Φ that is not dependent on j can achieve (3) for a given scheduling rule T. Second, the choice of such a Φ for a given T need not be unique. In the subsequent examples, we establish the existence of such a Φ and address uniqueness for the specific instances of T that are considered. In these examples we present a much stronger form of equilibrium: We show that (3) holds not only in expectation but also for every possible realization of the arrival sequence.

SM's Objective. Let $\bar{Q}(T, \Phi_T)$ denote the long-run average number of customers in the network when operating under the scheduling rule T and when customers choose their routes according to the Nash routing strategy Φ_T. The SM's objective is given by

$$\min_{T \in D} \bar{Q}(T, \Phi_T).$$

That is, the SM chooses a nonidling scheduling rule that minimizes the average number of customers in the network (over all nonidling rules) when the customers choose the corresponding Nash routing strategy.

Assumptions on Parameters. We make the following assumptions on the arrival rate λ and the service times m_f and m_s.

Assumption 1. $\lambda(m_f + m_s) < 2$.

Assumption 2. $\lambda(\max(m_f, m_s)) > 1$.

Assumption 3. $m_f < m_s$.

Assumption 1 is a necessary condition for stability. Without this assumption, the SM can never achieve a finite $\bar{Q}$. Assumption 2 necessitates the splitting of traffic across both routes to achieve a finite $\bar{Q}$; that is, the servers cannot handle the traffic if every customer chooses the same route. Finally, Assumption 3 is essential for the instability examples in §4 and it is relaxed in §7.

4. Unstable Nash Equilibria

In this section we construct scheduling rules that cause $\bar{Q}$ to be infinite in Nash equilibrium. In fact, under the chosen rules $Q(t) \to \infty$ as $t \to \infty$ with probability 1. One such rule is the preemptive static priority rule that gives priority to the exit buffers 3 and 4 over the entry buffers 1 and 2, respectively. Let $T^{\text{PEB}} \in D$ denote the priority-to-exit-buffer (PEB) scheduling rule, formally defined by

$$T_i^{\text{PEB}}(Q) = \left(1 - I_{\{Q_{5-i}>0\}}\right) I_{\{Q_i>0\}} \quad i = 1, 2,$$
$$T_i^{\text{PEB}}(Q) = I_{\{Q_i>0\}} \quad i = 3, 4. \tag{4}$$

T^{PEB} is a greedy rule that myopically tries to remove customers from the system at the fastest possible rate.[2] Furthermore it corresponds to the so-called shortest remaining process time rule (cf. Panwalker and Iskander 1977) in this network. The T^{PEB} scheduling rule induces a unique Nash routing strategy that is characterized in the following lemma.

Lemma 1. *T^{PEB} induces the (dominant) Φ^{SEB} routing strategy in which the arriving customers join the server with the smaller-exit-buffer* (SEB). *To be precise,*

$$\Phi^{\text{SEB}}(Q) = \begin{cases} L & \text{if } Q_4 < Q_3 \text{ or } Q_4 = Q_3,\ Q_1 \le Q_2 \\ R & \text{if } Q_4 > Q_3 \text{ or } Q_4 = Q_3,\ Q_1 > Q_2 \end{cases}.$$

Proof. We claim that joining the route with the larger exit buffer is the *dominant* strategy, meaning that a customer's best action is independent of other customers' actions (including the future arrivals). For a given sample path suppose that a customer arrives at $t = 0$ and observes the queue lengths $Q(0)$. In the rest of the proof, $Q(0)$ is abbreviated as Q.

When $Q_3 = Q_4$ and $Q_1 \le Q_2$, the servers initially serve the exit buffers until they are drained to zero. Thereafter, server efforts are split such that the entry buffers are drained at the same rate, keeping the exit buffers empty and hence joining route L leads to a shorter sojourn time. Similarly, when $Q_3 = Q_4$ and $Q_1 > Q_2$, joining route R leads to a shorter sojourn time.

Now suppose that $Q_3 > Q_4$. Again, the servers initially serve the exit buffers and, as soon as buffer 4 becomes empty, the network behaves like a tandem queue of buffers 1 and 3 until buffer 3 becomes empty. This is the consequence of the scheduling rule, the fact that $m_f < m_s$ and the assumption that server B can serve partially processed customers in buffer 1 (that is, when a customer receives her first service, she also enters the exit buffer on her chosen route). Thus sojourn time of the customer is equal to $(Q_3 + Q_1 + 1)m_s$ when she joins route L. On the other hand, when she joins route R, the sojourn time is greater than or equal to $(Q_3 + Q_1 + Q_2 + 1)m_s$, with equality if no further arrival occurs. Clearly the sojourn time is longer when the customer joins route R. This analysis is independent of the decisions of other customers (past and future arrivals). Hence Φ^{SEB} is indeed the dominant strategy. □

The Nash equilibrium is unique because dominant strategies are employed. Note that Φ^{SEB} is equiva-

[2] When $\min(Q_1, Q_2) > 0$ and $Q_3 = Q_4 = 0$, to be consistent with (4) and the assumption that partially served customers in the entry buffers can be processed as if they were in their exit buffers, the server efforts are split as $T_1 = T_2 = m_s/(m_f + m_s)$ and $T_3 = T_4 = m_f/(m_f + m_s)$ so that the exit buffers stay empty and the entry buffers are served at the same rate. In all other cases, (4) holds as stated.

lent to joining the *route with the larger exit buffer*. For example, if $Q_3 > Q_4$, then the arriving customer joins buffer 1, i.e., takes route L. The result is intuitive. Because the exit buffers are prioritized, the customer joins the server with the smaller exit buffer to minimize her waiting time in her entry buffer. Under the $T^{\text{PEB}} - \Phi^{\text{SEB}}$ combination the network is unstable as stated in the next theorem.

THEOREM 2. *Under the combination of the T^{PEB} scheduling rule and the consequent Φ^{SEB} Nash routing strategy,*

$$\mathbf{P}\left\{\frac{\max(Q_3(t), Q_4(t))}{t} \to 0 \text{ as } t \to \infty\right\} = 0.$$

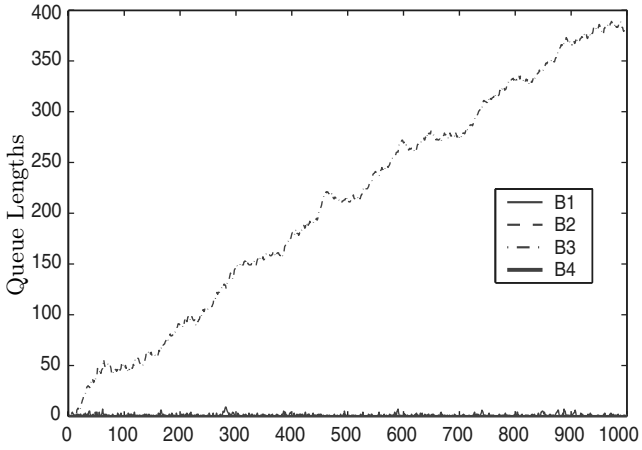
PROOF. We show that on every sample path, all arrivals join the same route after some finite time. This implies that one of the buffers 3 or 4 behaves like an overloaded unstable $GI/D/1$ queue because of Assumptions 2 and 3, and the result follows.

Suppose $Q_2(0) = Q_4(0) = 0$ and $Q_1(0), Q_3(0) \geq 0$. It is straightforward to verify that all arrivals join route L from $t = 0$ onwards and either $Q_3 > Q_4$ or $Q = 0$ (network is empty) at any time. Now suppose $Q_4(0) > Q_3(0)$. The exit buffers are served together as long as $Q_4 > Q_3 > 0$. Hence, either $Q_4 > Q_3$ for all times or $Q_4 = 0$ (and $Q_3(0) = 0$) at some finite time. But $Q_4 = 0$ requires $Q_2 = 0$ because $m_f < m_s$. Therefore, either all arrivals join route R from $t = 0$ onwards (that is, $Q_4 > Q_3$ holds for all times), or buffer 4 empties and $Q_4 = Q_2 = 0$ at some finite time, which takes us back to the previous case. Similarly, one can start with $Q_3(0) > Q_4(0)$ or $Q_3(0) = Q_4(0)$ and show that either all arrivals join the same route from $t = 0$ onwards, or we get back to one of the cases above in finite time. □

We want to point out that the symmetry of service times is not necessary for the instability result of Theorem 2 as is evident from the proof of Lemma 1. As long as $m_1 < m_3$ and $m_2 < m_4$, the Nash routing strategy is to join the server with the smaller exit buffer weighted by the service times and the instability result follows from Assumption 2.

Some discussion on Theorem 2 is in order. Although the queues blow up under the specified policy, it is still in the interest of the customer to stick to her Nash routing strategy. If the customer deviates from the Nash routing strategy, it will certainly take longer for her to depart from the network. In fact, she might never depart from the network. Theorem 2 is derived under the rather stringent assumption of deterministic service times. The reader may be tempted to believe that the instability phenomenon is an artifact of synchronization between various operations in the network. This is not the case. Figure 2 displays the result of a simulation under the $T^{\text{PEB}} - \Phi^{\text{SEB}}$ combination carried out with exponential service times and parameters that satisfy Assumptions 1–3. The system

Figure 2 **A Simulation Trace Under the $T^{\text{PEB}} - \Phi^{\text{SEB}}$ Combination with Exponential Server Times and Poisson Arrivals** ($m_f = 0.5$, $m_s = 1.33$, $\lambda = 1$)



is evidently unstable. The proof of instability of the network operating under the $T^{\text{PEB}} - \Phi^{\text{SEB}}$ combination and random i.i.d. service times whose means satisfy Assumptions 1–3 can be established. However, when the service times are not deterministic, Φ^{SEB} is not necessarily a Nash routing strategy induced by the T^{PEB} scheduling rule.

To the reader familiar with the Rybko-Stolyar (RS) network (Rybko and Stolyar 1992), the choice of T^{PEB} as the scheduling rule may appear naive. Assumptions 2 and 3 correspond to the parameter choices that cause the instability phenomenon in the RS network. However, the RS network differs from our network essentially in that an arriving customer randomly picks her route using a fair coin rather than choosing the route in her self-interest. T^{PEB} is identical to the rule that causes instability in the RS network. However, the mechanism of instability is quite different in the RS network. In our network a server runs out of capacity because of selfish routing choices of customers, whereas in the RS network instability is a result of alternate blocking and starvation. To be more specific, in the RS network the high-priority buffer 3 blocks the low-priority buffer 2 at the same server from getting service, and this leads to starvation in buffer 4. Furthermore, the customers accumulate in buffer 2 as a result of arrivals. Eventually when the high-priority buffer 3 empties, buffer 2 starts "flooding" downstream buffer 4 which is also a high-priority buffer. This in turn, causes buffer 4 to block the low-priority buffer 1 at the same server from getting service, starving buffer 3 of customers. With careful choice of parameters, this cycle can be made to repeat. Because one high-priority buffer is always starved and hence forced to idle, the work in the system at the end of each cycle can be made larger than the work at the beginning of the cycle, resulting in instability.

Instability in our network is not limited to the T^{PEB} scheduling rule. One can construct other scheduling rules that induce instability in our network, despite not having been proven unstable in the RS network. One such rule is the *first-in-system-first-out* (FISFO), or Global FIFO rule, which gives preemptive priority at each server to the customer who arrived earliest to the network among all waiting customers.[3] To precisely specify the FISFO rule, one needs to extend the domain of mapping T^{FISFO} from R^4 to $R^Z \times R^4$, because one needs to keep track of the arrival time a_j of every customer j in addition to the queue lengths. However, the range of T^{FISFO} can be kept the same as before ($[0,1]^4$) and the definitions (1) and (2) of the feasible scheduling policies continue to hold. The defining feature of T^{FISFO} is $T_i^{\text{FISFO}} = 1$ if and only if there is a customer j in buffer i such that $a_j = \min\{a_l\}$ for all customers l in buffers i and $5-i$. Although we do not completely characterize the Nash routing strategy induced by T^{FISFO}, we provide the following partial characterization that is sufficient to establish that T^{FISFO} results in the same form of instability as T^{PEB}.

THEOREM 3. *Under* T^{FISFO}, *the* (*dominant*) *routing strategy is to choose* $\Phi(Q) = L$ *when* $Q_4 = Q_2 = 0$ *and* $Q_3 > 0$. *Similarly, the* (*dominant*) *strategy is to choose* $\Phi(Q) = R$ *when* $Q_3 = Q_1 = 0$ *and* $Q_4 > 0$. *That is, in these set of states any Nash routing strategy coincides with* Φ^{SEB}. *Consequently,*

$$\mathbf{P}\left\{\frac{\max(Q_3(t), Q_4(t))}{t} \to 0 \text{ as } t \to \infty \;\middle|\; Q(0) = 0\right\} = 0.$$

The proof of this theorem is essentially same as the proof of Theorem 2 and is omitted.

We want to point out that the instability is not intrinsic to the T^{FISFO} scheduling rule, rather it is a consequence of the choices made by rational customers, as evidenced by the fact that T^{FISFO} is stable under a simple centralized routing scheme that determines customers' routes by fair coin tosses.

As before, one can verify (using simulation) that the $T^{\text{FISFO}} - \Phi^{\text{SEB}}$ combination is unstable even when the deterministic assumptions are relaxed. When the network starts empty, T^{PEB} and T^{FISFO} result in the same trajectories, because they induce the same Φ^{SEB} routing policy. However, it is important to notice that these scheduling policies would result in different trajectories under a different centralized routing policy, such as fair coin tossing.

Finally, as an interesting aside we consider the behavior under the well-known, centralized join-the-shortest-queue (JSQ) routing rule, which routes customers to the server that has a smaller total number of customers. In the set of states with $Q_3 > Q_1 \geq Q_2 = Q_4 = 0$, an incoming customer is routed to route L and the network stays in the same set of states with $Q_3 > Q_1 \geq Q_2 = Q_4 = 0$. Hence, all arrivals are routed to route L. So, the system behaves like a tandem queue of buffers 1 and 3, and buffer 3 goes unstable independent of the nonidling scheduling rule employed. The nontrivial question that remains is whether one can construct a scheduling rule that induces stable "good" behavior in this network. This is topic of the next section.

5. Achieving Good Nash Equilibria

First-Best Solution. The optimal policy is not known for this network, as is common in many networks, even with centralized routing or when arrivals to the two routes are exogenously generated. The problem of determining the optimal policy is both analytically and computationally difficult as discussed by Papadimitriou and Tsitsiklis (1996). However, from the symmetry of the network, we expect that there exists an optimal routing and scheduling rule that is symmetric. In particular, the average queue lengths in the entry buffers should be equal, and those of exit buffers should also be equal under this optimal policy.

Maglaras (1998) provides a policy that is optimal in an asymptotic sense (in the so-called fluid limit) when arrivals to the two routes are exogenously generated. The suggested policy keeps the exit buffers as small as possible, holding most of the work in entry buffers. It gives priority to both exit buffers, aiming to decrease queue lengths myopically, unless an exit buffer is "small," in which case the entry buffer that feeds the small exit buffer is prioritized.

Workload Regulating Scheduling Rule. It is shown in §4 that giving blind priority to exit buffers is not a good idea when customer choose their own routes. All customers eventually prefer one route under T^{PEB} and the other route is not used at all, once a certain state is reached. A similar phenomenon occurs under T^{FISFO} as well. The key driver of performance degradation in these cases is the following. The scheduling rule favors the more congested route and makes it attractive for arriving customers to join. The system moves further away from the symmetric ideal with each arriving customer. Therefore, any scheduling rule that is intended to minimize performance degradation should favor the less congested route at times, to make it attractive for arriving customers and restore symmetry as much as possible. Restoring symmetry alone does not suffice, however. The scheduling rule needs to minimize idleness and remove work from the system efficiently. We would like the scheduling rule to keep the exit buffers small

[3] This differs from the standard FIFO rule in which priority is given to the customer who arrived earliest at the *server*.

Table 1 **Server Assignments of the T^{WR} Scheduling Rule in Representative Situations**

Q_1	Q_4	Q_2	Q_3	T_1	T_4	T_2	T_3
0	>0	0	>0	0	1	0	1
>0	0	0	>0	1	0	0	1
≥ 0	$>\theta$	≥ 0	$>\theta$	0	1	0	1
>0	$<\theta$	>0	$<\theta$	1	0	1	0
>0	$>\theta$	≥ 0	$<\theta$	1	0	0	1
0	>0	>0	$>Q_4$	0	1	1	0
0	>0	>0	$<Q_4, >0$	0	1	0	1

as in Maglaras (1998) once the initial queues have dissipated. Combining these two ideas, we propose the following workload regulating[4] scheduling rule $T^{WR} \in D$.

$$T_1^{WR}(Q) = I_{\{Q_1>0\}}\left[I_{\{Q_4=0\}} \vee I_{\{Q_3<\theta\}}\left(1 - I_{\{0<Q_4\leq Q_3\}} I_{\{Q_2=0\}}\right)\right], \quad (5)$$

$$T_2^{WR}(Q) = I_{\{Q_2>0\}}\left[I_{\{Q_3=0\}} \vee I_{\{Q_4<\theta\}}\left(1 - I_{\{0<Q_3\leq Q_4\}} I_{\{Q_1=0\}}\right)\right], \quad (6)$$

$$T_3^{WR}(Q) = I_{\{Q_3>0\}}\left[I_{\{Q_2=0\}} \vee I_{\{Q_4>\theta\}} \vee I_{\{Q_3<Q_4\leq\theta\}} I_{\{Q_1=0\}}\right], \quad (7)$$

$$T_4^{WR}(Q) = I_{\{Q_4>0\}}\left[I_{\{Q_1=0\}} \vee I_{\{Q_3>\theta\}} \vee I_{\{Q_4<Q_3\leq\theta\}} I_{\{Q_2=0\}}\right]. \quad (8)$$

Table 1 illustrates the behavior of the T^{WR} scheduling rule in representative situations.[5] The rule is parameterized by threshold θ, which is used to identify the exit buffers as big or small.

When both exit buffers are big, i.e., $Q_3, Q_4 > \theta$, they are given priority and servers process exit buffers (see (7) and (8)). This aims to decrease the number of customers in the system at the fastest rate. When one of the exit buffers, say buffer 3, is small, i.e., $Q_3 < \theta$, the entry buffer at server A, buffer 1, that feeds the small exit buffer 3, is prioritized to avoid idleness at server B (see (5)). This makes the less crowded route, route L, more attractive for an arriving customer as the entry buffer of that route, buffer 1, is prioritized at server A over the other buffer, buffer 4. This induces arrivals to split evenly and thus the WR policy avoids the instability that occurs when all arrivals choose the same route.

We want the WR policy to give rise to a Nash equilibrium of the form described in §3. We need to ensure that working on an exit buffer, say buffer 3, merely to avoid idleness at server B when the entry buffer at the same server, buffer 2, is empty, does not favor route L. To do this, if the other exit buffer, buffer 4, is smaller than buffer 3, then it is also served whenever buffer 3 is served, to treat the competing routes fairly. For example, when $\theta \geq Q_3 \geq Q_4$ and $Q_2 = 0$, server B has to serve buffer 3, and the WR policy specifies that server A serves buffer 4 (see (8)). This property of WR helps us obtain a tractable equilibrium resulting in simple and intuitive customer routing decisions.

To describe the Nash routing strategy induced by T^{WR}, we define

$$\Delta(Q) = (m_s - m_f)[Q_1 - Q_2] + m_s[Q_3 - Q_4]. \quad (9)$$

Δ can be interpreted as the degree of workload imbalance between two routes. Buffers 3 and 4 are given a weight of m_s, the service time at those buffers. Buffers 1 and 2 are given a weight of $m_s - m_f$ that takes into account the effect of parallel processing at the exit buffers.

Moreover, the sign of Δ indicates whether or not the departure time of the last customer (from the network) on route L is greater than that of the last customer on route R, if there are no future arrivals and no idling of servers before their departure. The following lemma formalizes this statement by showing that Δ stays constant until one of the servers idles. From now on we abuse notation and use $\Delta(t) \equiv \Delta(Q(t))$ and $T(t) \equiv T(Q(t))$.

Lemma 4. *If there are no arrivals to the network after $t = 0$, then $\Delta(t) = \Delta(0)$ for $t \leq \tau$ under any nonidling scheduling policy, where $\tau = \min(t\colon \min(T_1(t) + T_4(t), T_2(t) + T_3(t)) = 0)$.*

Proof. If server A is processing buffer 1 and server B is processing buffer 2, then Δ does not change because $[Q_1(t) - Q_2(t)]$ and $[Q_3(t) - Q_4(t)]$ stay constant. Similarly, if server A is processing buffer 4 and server B is processing buffer 3, Δ does not change. If server A is processing buffer 1 and server B is processing buffer 3, then $Q_1(t)$ decreases at a rate $1/m_f$ but $Q_3(t)$ increases at a rate $1/m_f - 1/m_s$; thus the rate of change in Δ is given by $-(m_s - m_f)(1/m_f) + m_s(1/m_f - 1/m_s) = 0$. Similarly, if server A is processing buffer 4 and server B is processing buffer 2, Δ stays constant. □

Lemma 4 shows that Δ changes only by arrivals and idleness. Based on the quantity Δ, we define the join-the-shorter-route (JSR) routing rule as follows,

$$\Phi^{JSR}(Q) = \begin{cases} L & \text{if } \Delta < 0 \\ R & \text{if } \Delta > 0 \\ L \text{ w.p. } 1/2 \text{ and } R \text{ w.p. } 1/2 & \text{if } \Delta = 0 \end{cases}. \quad (10)$$

Under Φ^{JSR} each arriving customer makes her routing choice based on the sign of Δ. This is equivalent to choosing the route through which the customer would depart sooner from the network (the "shorter" route in this sense), if there were no future arrivals.

[4] The name is suggested by the fact that each server attempts to make sure that there is work for the other server under this scheduling rule.

[5] For the sake of brevity, we have not specified the scheduling policy for $Q_1 > 0$, $Q_3 = \theta$ and $Q_4 \geq \theta$, or $Q_2 > 0$, $Q_4 = \theta$ and $Q_3 \geq \theta$, respectively. In these cases, server effort is split in a way that keeps buffer 3, or buffer 4 respectively, at the threshold level.

Each customer is choosing the best route myopically, in the sense of ignoring future arrivals, and this decision turns out to be optimal even in the presence of future arrivals, given that every other customer follows the same strategy. This result is formally stated in the following theorem.

THEOREM 5. *The Φ^{JSR} routing strategy is a Nash routing strategy induced by the T^{WR} scheduling rule.*

The proof of Theorem 5 is presented in the next section. The theorem does not ensure that Φ^{JSR} is the unique Nash routing strategy induced by T^{WR}. However, as can be ascertained from the following corollary, any other Nash routing strategy can deviate from Φ^{JSR} only when both of the exit buffers are below the threshold.

COROLLARY 6. *If Φ is a Nash routing strategy induced by the T^{WR} scheduling rule, then $\Phi(Q) = \Phi^{JSR}(Q)$ when* $\max(Q_3, Q_4) \geq \theta$.

The proof of Corollary 6 is discussed in §6.1. The corollary shows that any other possible equilibrium cannot show a substantial deviation in performance from $T^{WR} - \Phi^{JSR}$. Particularly, the stability of the $T^{WR} - \Phi^{JSR}$ combination leads to the stability of any other $T^{WR} - \Phi$ given that Φ is a Nash routing strategy.[6] The stability of the $T^{WR} - \Phi^{JSR}$ combination is stated in Theorem 7. Hence, we conclude that no Nash equilibrium induced by T^{WR} can be unstable. We now turn our attention to the performance of the $T^{WR} - \Phi^{JSR}$ combination.

Performance Analysis

Stability. Figure 3 presents the results of a simulation run under the $T^{WR} - \Phi^{JSR}$ combination with the same choice of parameters that led to instability under the $T^{PEB} - \Phi^{SEB}$ combination in Figure 2. As is evident from Figure 3, the combination of T^{WR} and Φ^{JSR} appears stable for this choice of parameters. Theorem 7 below formalizes this result.

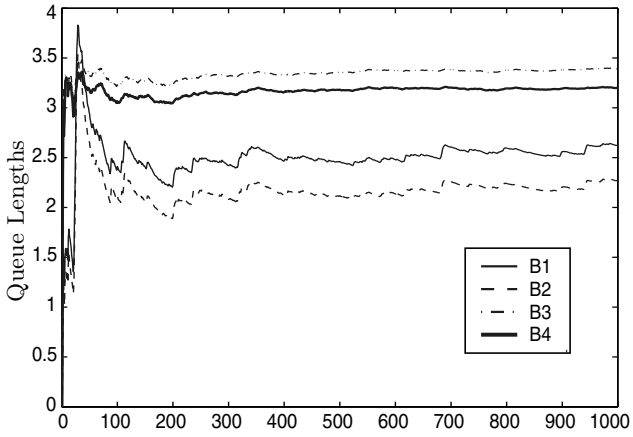
THEOREM 7. *The combination of the T^{WR} scheduling rule and the Φ^{JSR} routing strategy is* stable *in the following sense of rate stability*

$$\mathbf{P}\left\{\max_{i=1,2,3,4} \frac{Q_i(t)}{t} \to 0 \text{ as } t \to \infty\right\} = 1.$$

The proof of the theorem, which is presented in the appendix, takes advantage of the following facts: Arrivals are split evenly, and the exit buffers are held under the threshold after some finite time.

[6] Observe that under T^{WR}, an entry buffer cannot blow up while holding the corresponding exit buffer below the threshold, as T^{WR} will require the entry buffer to feed the exit buffer which will then increase above the threshold level.

Figure 3 **A Simulation Trace Under the $T^{WR} - \Phi^{JSR}$ Combination with Exponential Service Times and Poisson Arrivals** ($m_f = 0.5$, $m_s = 1.33$, $\lambda = 1$)

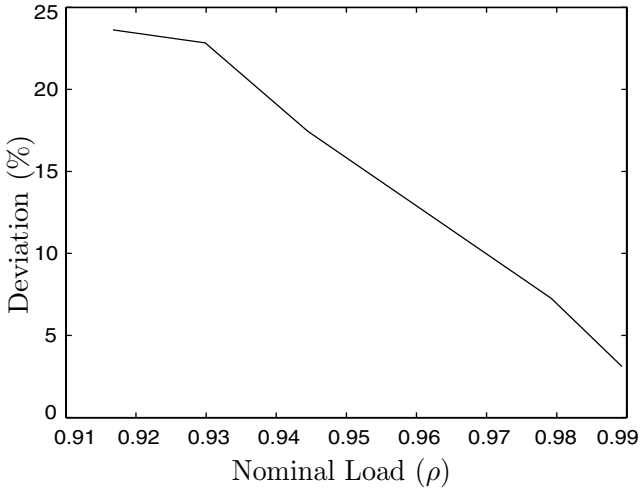


Performance Benchmarks. An important issue to explore is whether the $T^{WR} - \Phi^{JSR}$ combination has good performance in a finer sense than mere stability. In the absence of a known optimal solution, even for the case when all decisions are centralized, we evaluate the performance of the $T^{WR} - \Phi^{JSR}$ combination in two ways: (i) The network is simulated with exponential service times and the average total queue length under $T^{WR} - \Phi^{JSR}$ and commonly used centralized policies are compared. (ii) Simulation results are compared against a theoretical lower bound.

Kumar and Kumar (1994) provide a lower bound on queue lengths under exogenous routing and exponential service times. By examining the consequence of a steady state for general quadratic forms, they obtain a set of linear equality constraints on the mean value of certain random variables that determine performance of the system. Further, conservation of time and material gives an augmenting set of linear inequality constraints. We extend their analysis allowing endogenous routing chosen by the SM to get a lower bound on queue lengths in our network. This lower bound may not be attainable even by centralized policies, yet comparison of this bound with simulation results suggests that $T^{WR} - \Phi^{JSR}$ has a performance close to optimal, at least for some parameter regimes. For example, for parameter choices of $m_f = 0.65$, $m_s = 1.33$, and $\lambda = 1$, the average queue length is within 3.1% of the lower bound. Figure 4 shows the result of a series of simulations obtained by keeping m_s and λ constant but increasing m_f and hence increasing the nominal load, $\rho = (\lambda(m_f + m_s))/2$. The figure suggests that the deviation from the lower bound decreases as the nominal load on the servers increases. We will strengthen this insight further in the next subsection.

Thus far, we have considered two extreme cases: the priority-to-exit-buffer (PEB) scheduling policy that

Figure 4 **Deviation in Performance of the $T^{WR} - \Phi^{JSR}$ Combination from the Theoretical Lower Bound ($\lambda = 1$, $m_s = 1.33$, and m_f Varies from 0.5 to 0.65)**



leads to instability, and the WR policy that gets close to the optimal centralized solution. An example of a scheduling policy whose performance falls between the two extremes is the priority-to-arriving-buffer (PAB) policy that gives priority to customers in the entry buffers, that is, to customers in buffer 1 over buffer 4 and to customers in buffer 2 over buffer 3. Along the lines of Lemma 1 and Theorem 5, we can establish that the Φ^{JSR} routing rule (defined in Equation (10)) is a Nash routing strategy induced by T^{PAB}. Performance of the $T^{PAB} - \Phi^{JSR}$ combination is shown in Table 2. As seen in the table, $T^{PAB} - \Phi^{JSR}$ appears stable but has a performance that is considerably worse than $T^{WR} - \Phi^{JSR}$. Thus, merely switching priority from exit buffers to entry buffers, or in other words switching from unstable T^{PEB} to T^{PAB}, is not sufficient to obtain asymptotically first-best performance. One needs to do more as in the T^{WR} policy.

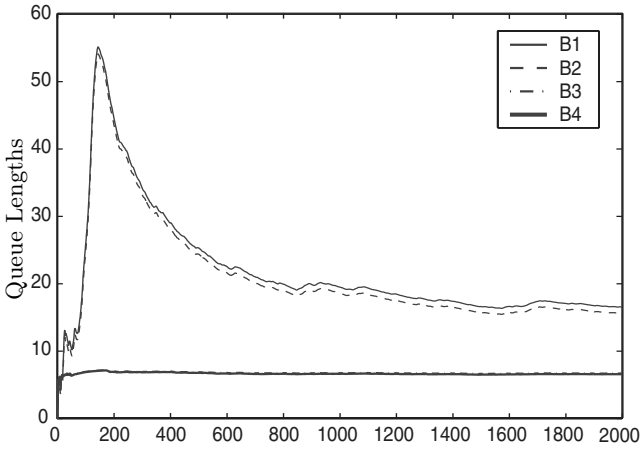
In addition, we compare performance of the $T^{WR} - \Phi^{JSR}$ combination to that of some commonly used centralized policies such as FIFO-JSQ (join-the-shorter-queue), PEB-JSQ, PAB-JSQ, FIFO-MR (Markovian routing that sends an arrival to either route based on a fair coin toss), and PAB-MR. Of course, in these cases the routing rule need not be a Nash routing strategy induced by the corresponding scheduling rule. Hence these policies are purely centralized benchmarks. $T^{FIFO} - \Phi^{JSQ}$, $T^{PEB} - \Phi^{JSQ}$, and $T^{PAB} - \Phi^{JSQ}$ yield unstable queues for this network: Queues grow without bound over time. Although $T^{FIFO} - \Phi^{MR}$ and $T^{PAB} - \Phi^{MR}$ lead to stable queues, the $T^{WR} - \Phi^{JSR}$ combination performs much better than these policies at different nominal loads tabulated in Table 2: The average total queue length under $T^{WR} - \Phi^{JSR}$ is several standard deviations below the average under $T^{PAB} - \Phi^{JSR}$, $T^{FIFO} - \Phi^{MR}$, and $T^{PAB} - \Phi^{MR}$, allowing us to establish the dominance of $T^{WR} - \Phi^{JSR}$. As discussed earlier, despite the fact that these policies route symmetrically, they do not schedule as efficiently as T^{WR}.

Table 2 **Total Queue Length Obtained Through Simulation Results and the Theoretical Lower Bound**

$m_s = 1.33$	WR-JSR	PAB-JSR	PAB-MR	FIFO-MR	Lower bound
$m_f = 0.56$					
Average	16.66	20.72	28.74	51.12	14.14
Std dev	0.76	0.88	1.13	2.79	
$m_f = 0.62$					
Average	39.92	56.92	80.24	137.86	37.58
Std dev	2.29	6.81	3.91	14.45	
$m_f = 0.65$					
Average	74.84	108.09	144.30	277.67	72.65
Std dev	7.69	16.91	25.61	49.72	

Asymptotic Analysis. As we have discussed in the previous subsection, comparison of the theoretical lower bound with the performance of proposed policy reveals that the proposed policy gets increasingly close to optimal as the nominal load on the system approaches 1 (see Figure 4). This is part of a more general phenomenon. We provide a somewhat loose discussion of this behavior below. Several salient features of the proposed policy emerge from the simulation result in Figure 5. First, the exit buffers 3 and 4 eventually stay at the threshold level θ, which is formalized in Lemma 13(iii) in the appendix. Second, most of the jobs in the system stay in the entry buffers 1 and 2, and these two buffer levels are approximately equal, which is formalized in Lemma 13(iv) in the appendix. These results yield the following. Suppose that we look at the queue lengths scaled by $1/N$ for some large number N. As stated in Lemmas 13(iii) and (iv), under the $T^{WR} - \Phi^{JSR}$ combination when scaled by $1/N$, the system eventually has equal queues in each of the

Figure 5 **A Simulation Trace Under the $T^{WR} - \Phi^{JSR}$ Combination with Threshold $\theta = 7$ and Nominal Load $\rho = 0.98$**



entry buffers and negligible queues in each of the exit buffers up to a tolerance of $o(N)$. This is exactly the asymptotically optimal behavior on the "fluid-scale" suggested by Maglaras (1998). Thus, we can conclude that our proposed policy is asymptotically optimal in the sense of Maglaras. In fact, we can go further and describe the good behavior of the $T^{\mathrm{WR}} - \Phi^{\mathrm{JSR}}$ combination using the heavy traffic approach pioneered by Harrison (1988) and Harrison and Van Mieghem (1997). Under suitable scaling, the optimal behavior of a queueing system in which the nominal load on each server is close to 1 (corresponding to $(\lambda(m_f + m_s))/2 \approx 1$ in our case) can be approximated by the solution to a frictionless ideal called the Brownian control problem. In this problem, it is possible to instantaneously move between any two states that have the same "workload," defined to be $W = (m_f + m_s)(Q_1 + Q_2) + m_s(Q_3 + Q_4)$. Since the system manager's objective would be achieved if one were to minimize $Q_1 + Q_2 + Q_3 + Q_4$, the optimal solution in the Brownian control problem is to move any configuration of queues that result in a workload W to the configuration $Q_1 = Q_2 = W/(2(m_s + m_f))$ and $Q_3 = Q_4 = 0$ and stay at that configuration thereafter, which suggests to keep all the work in the entry buffers. The $T^{\mathrm{WR}} - \Phi^{\mathrm{JSR}}$ combination eventually achieves this configuration in a suitable scale as discussed above. Thus, one would expect that the behavior of $T^{\mathrm{WR}} - \Phi^{\mathrm{JSR}}$ would be close to optimal on a suitable scale when the nominal load $(\lambda(m_f + m_s))/2 \approx 1$, as in Figure 4. Lemmas 13(iii) and (iv) in §A.1 allow us to actually prove the asymptotic optimality (under suitable scaling) of the $T^{\mathrm{WR}} - \Phi^{\mathrm{JSR}}$ combination when $(\lambda(m_f + m_s))/2 \to 1$, but we do not attempt this proof as it will take us too far afield from our primary interest, namely design of scheduling policies that induce good Nash equilibria.

6. Proof of Theorem 5

In this section we provide a detailed proof of Theorem 5. The proof will be one of the sample-path variety: For every realization (sample path) of the arrival process, we show that a unilateral deviation by a customer results in a departure time that is no sooner than the departure time under the equilibrium routing strategy, namely Φ^{JSR}. In the rest of this section we assume that an arbitrary sample path of the arrival process has been fixed, and proceed with the analysis.

Let two twin customers, Good and Bad, hereafter referred as twin G and twin B, arrive at the network simultaneously at $t = 0$ and observe the queue lengths $Q^-(0)$ (where Q^- denotes the queue lengths excluding the twins and Q includes the twins). Twin G and other customers that is, future arrivals, route themselves by Φ^{JSR} (defined in (10)) and twin B deviates from Φ^{JSR}. Without loss of generality, assume that

$$\Delta(Q^-(0)) \le 0. \tag{11}$$

So, twin G joins route L and twin B joins route R. Observe that $\Delta(Q(0)) = \Delta(Q^-(0))$. By Lemma 4, twin G would depart from the network before twin B if there were no further arrivals. We will do a sample-path analysis to conclude that twin G departs from the network before twin B even when arrivals occur for every realization of the arrival sequence. Then this rather strong result is used to establish (3) for the $T^{\mathrm{WR}} - \Phi^{\mathrm{JSR}}$ combination.

Let us call the customers present at the network (including the twins) at $t = 0$ *old customers*, and the customers that arrive at any $t > 0$, *new customers*. Let $Q^o(t) = (Q_1^o(t), Q_2^o(t), Q_3^o(t), Q_4^o(t))$ denote the queue lengths because of the old customers and $Q^n(t) = (Q_1^n(t), Q_2^n(t), Q_3^n(t), Q_4^n(t))$ denote the queue lengths because of the new customers. So, $Q(0) = Q^o(0)$, and $Q(t) = Q^o(t) + Q^n(t)$.

Let

$$\Delta^o(t) = \Delta(Q^o(t))$$

and

$$\tau_i = \inf\{t \ge 0 \colon Q_i^o(t) = 0\} \quad \text{for } i = 1, 2.$$

Hence, τ_1 and τ_2 denote the departure times of twins G and B from their entry buffers, respectively. The following corollary shows that Δ^o stays constant until a new customer is served. This observation plays an important role in the proof of the theorem.

Corollary 8. *$\Delta^o(t) = \Delta(0)$ for $t \le \tau$ under any non-idling scheduling policy, where $\tau = \inf\{t \colon [t > \tau_1, T_1(t) > 0]$ or $[t > \tau_2, T_2(t) > 0]\}$.*

Proof. $\Delta^o(0) = \Delta(0)$ by definition. No new customers have been processed up to time t for all $t \le \tau$. So Q^o is same as the queue lengths of an identical system that has not received any arrivals. Lemma 4 applies to the identical system, hence $\Delta^o(t)$ stays constant until $t \le \tau$. □

The following lemma provides the main idea of the proof of Theorem 5: The lemma gives a sufficient condition for twin G to depart from the network sooner than twin B and this is exploited in the sequel. Let τ_G and τ_B denote the departure times of twins G and B from the network, respectively.

Lemma 9. *Suppose there exists a $\tau \ge 0$, such that*

$$Q_4^o(\tau) > 0 \quad \text{and} \quad Q_3^o(t) \le Q_4^o(t) \quad \text{for all } t \ge \tau. \tag{12}$$

Then $\tau_G \le \tau_B$.

Proof. Twin G and B are both the last old customers on their routes. So, $Q_3^o(\tau_G)=0$ and $Q_4^o(\tau_B)=0$. If $\tau_G > \tau_B$ then $Q_3^o(\tau_B) > Q_4^o(\tau_B)=0$, which contradicts to the assumption (12) of the lemma. □

There are two cases to consider, depending on which twin departs from her entry buffer sooner. Proofs of all lemmas stated in these two cases are provided in the appendix.

6.1. Case 1: $(\tau_1 \le \tau_2)$

This is the case in which twin G departs from her entry buffer sooner. Figure 6 presents a snapshot of the network at $t=\tau_1$. One can verify that whenever one of the exit buffers is above the threshold, the system always ends up in Case 1. Therefore this is the case that determines the behavior of the system when an exit buffer gets large and Corollary 6 follows by the proof of this case.

Lemma 10 below establishes that there are fewer customers in front of twin G when both twin have departed from their entry buffers, i.e., at $t=\tau_2$.

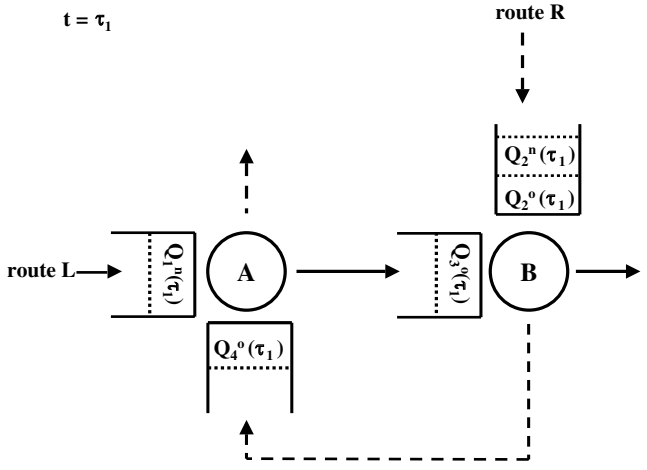
Lemma 10. $Q_3^o(\tau_2)-Q_4^o(\tau_2)=(1/m_s)(\Delta(0)-m_f Q_3^n(\tau_2))$.

Lemma 10 establishes that $Q_3^o(\tau_2) \le Q_4^o(\tau_2)$. So, if the SM serves only customers in buffers 3 and 4 after $t > \tau_2$, then clearly twin G departs from the network sooner. We bound the amount of time that the SM can serve buffer 4 but not buffer 3 under T^{WR} and derive Lemma 11, an upper bound on the difference $Q_3^o(t)-Q_4^o(t)$ for all $t \ge \tau_2$, as a function of $Q(\tau_2)$ and $Q^o(\tau_2)$.

Lemma 11. *For all* $t \ge \tau_2$,

$$Q_3^o(t)-Q_4^o(t) \le Q_3^o(\tau_2)-Q_4^o(\tau_2) + \frac{m_f}{m_s-m_f}[Q_3(\tau_2)-Q_4(\tau_2)]^+.$$

Figure 6 **The Case of $\tau_1 \le \tau_2$ at $t=\tau_1$**



By Lemmas 10 and 11 we prove that the condition (12) of Lemma 9 is satisfied. For $t \ge \tau_2$,

$$\begin{aligned}
&Q_3^o(t)-Q_4^o(t)\\
&\quad\le Q_3^o(\tau_2)-Q_4^o(\tau_2)+\frac{m_f}{m_s-m_f}[Q_3(\tau_2)-Q_4(\tau_2)]^+\\
&\quad=\max\Bigg(Q_3^o(\tau_2)-Q_4^o(\tau_2),\, Q_3^o(\tau_2)-Q_4^o(\tau_2)\\
&\qquad\qquad+\frac{m_f}{m_s-m_f}\big(Q_3^o(\tau_2)+Q_3^n(\tau_2)-Q_4^o(\tau_2)\big)\Bigg)\\
&\quad=\max\Bigg(Q_3^o(\tau_2)-Q_4^o(\tau_2),\\
&\qquad\qquad\frac{m_s\big(Q_3^o(\tau_2)-Q_4^o(\tau_2)\big)+m_f Q_3^n(\tau_2)}{m_s-m_f}\Bigg)\\
&\quad=\max\Bigg(Q_3^o(\tau_2)-Q_4^o(\tau_2),\,\frac{\Delta(0)}{m_s-m_f}\Bigg)\le 0.
\end{aligned}$$

Thus twin G departs from the network sooner. The first inequality above follows by Lemma 11. The second equality follows by the definition of τ_2 (i.e., $Q_4(\tau_2)=Q_4^o(\tau_2)$), and the last equality follows by Lemma 10.

Observe that the construction of this proof is independent of the strategies of the future arrivals. Hence Φ^{JSR} is the *dominant strategy* in Case 1. So any other Nash routing strategy should have the same routing choice in a state that leads to Case 1, in particular when one of the exit buffers is above the threshold level. Thus Corollary 6 follows.

6.2. Case 2: $(\tau_1 > \tau_2)$

This is the case in which twin B reaches her exit buffer sooner. Figure 7 presents a snapshot of the network at $t=\tau_2$. We provide analogs of Lemmas 10 and 11 in Lemma 12. Let τ_4 be the first time buffer 4 is served after $t \ge \tau_2$. To be precise, let

$$\tau_4=\inf\{t \ge \tau_2 :\, T_4(t)>0\}. \tag{13}$$

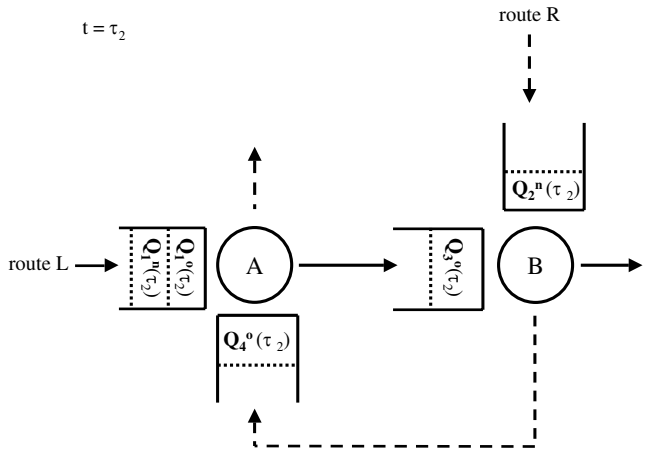
Lemma 12. (i) $Q_3^o(\tau_2)-Q_4^o(\tau_2)=\Delta(0)/m_s-((m_s-m_f)/m_s)Q_1^o(\tau_2)$.

(ii) *For all* $t \ge \tau_4$, $Q_3^o(t)-Q_4^o(t) \le Q_3^o(\tau_2)-Q_4^o(\tau_2)+((m_s-m_f)/m_s)Q_1^o(\tau_2)$.

Lemma 12(i) establishes that buffer 4 has more customers than buffer 3 when twin B reaches her exit buffer. The scheduling rule T^{WR} dictates that buffers 1 and 3 should be served in tandem until the number of customers in buffer 3 increases up to the level of buffer 4. This helps us to derive Lemma 12(ii), an upper bound on $Q_3^o(t)-Q_4^o(t)$ for all $t > \tau_4$.

Lemma 12 helps us to show that twin G has fewer customers in front of her than twin B has for all $t > \tau_4$

Figure 7 **The Case of $\tau_2 < \tau_1$ at $t = \tau_2$**



or equivalently the condition (12) of Lemma 9.[7] For $t \geq \tau_4$,

$$\begin{aligned} Q_3^o(t) - Q_4^o(t) &\leq Q_3^o(\tau_2) - Q_4^o(\tau_2) + \frac{m_s - m_f}{m_s} Q_1^o(\tau_2) \\ &= -\frac{m_s - m_f}{m_s} Q_1^o(\tau_2) + \frac{\Delta(0)}{m_s} + \frac{m_s - m_f}{m_s} Q_1^o(\tau_2) \\ &= \frac{\Delta(0)}{m_s} \leq 0. \end{aligned}$$

Thus twin B departs the network no sooner than twin G. The first inequality above follows by Lemma 12(ii), and the second equality follows by Lemma 12(i).

Thus far we have checked that on each sample path twin B departs from the network no sooner than twin G. But twin G may be imposing an externality on twin B and *vice versa*. To complete the proof of Theorem 5, we need to compare their departure times in the absence of each other. But, one can show that either there is no externality, or the twins have equal externality on each other, or the difference in externality is offset by the difference in departure times. We discuss this in a greater detail in the appendix.

7. Extensions

Thus far, we have used fairly restrictive assumptions on service times and network topology. These assumptions allowed us to deliver a crisp twofold message: Scheduling policies can induce Nash routing strategies that render the system unstable, and scheduling policies can be designed to induce Nash routing strategies that are asymptotically optimal. This message holds in greater generality than can be surmised from the earlier sections. To show this we consider two new settings in this section that involve relaxing Assumption 3 as well as the network topology of Figure 1.

We first consider the setting in which Assumption 3 is reversed but Assumptions 1 and 2 and the network topology are kept the same as in §3. That is, we assume $m_f > m_s$.[8] In this setting, we state the following equilibrium results but do not provide proofs in the interest of brevity; proofs are similar to those already provided in §§4 and 5. The PEB scheduling rule T^{PEB} induces the Nash routing strategy Φ^{JSR^c}, which is the exact opposite of the Φ^{JSR} routing. That is, if Δ defined in (9) is positive the customer chooses route L, if Δ is negative she chooses route R, and if Δ is equal to zero she flips a coin. In addition, the WR scheduling rule T^{WR} induces the routing strategy Φ^{LEB} in which customers join the *server* with the larger-exit-buffer (LEB) in the following sense. If two twin customers arrive at network at the same time and one twin chooses the route given by the Φ^{LEB} routing and the other twin chooses the opposite route, then the twin who chooses the Φ^{LEB} route exits the network before the other twin, given that future arrivals follow the Φ^{LEB} routing. This is a slightly different equilibrium concept than the Nash equilibrium of §3 because the externality of twins on each other cannot always be accounted for, in particular, when a server idles before the first twin departs from the network. Φ^{LEB} prescribes joining route L if $Q_4 > Q_3$, and joining route R if $Q_4 < Q_3$ and if $Q_4 = Q_3$ the customer joins the server with the smaller entry buffer. Furthermore, the PAB scheduling rule T^{PAB} induces the Φ^{JSR} routing defined in (10) in the same sense as T^{WR} induces Φ^{LEB}.

In terms of performance, the roles of the T^{PEB} and T^{WR} scheduling rules are reversed: While the $T^{\text{PEB}} - \Phi^{\text{JSR}^c}$ combination leads to the asymptotically first-best performance, the $T^{\text{WR}} - \Phi^{\text{LEB}}$ combination is unstable, as is the $T^{\text{PAB}} - \Phi^{\text{JSR}}$ combination. All customers join the same route under $T^{\text{PAB}} - \Phi^{\text{JSR}}$ and an entry buffer behaves like an overloaded $GI/D/1$ queue. However, instability occurs differently under $T^{\text{WR}} - \Phi^{\text{LEB}}$: In particular, not all customers join the same route, and yet the system goes unstable. Contrary to these instability examples, simulation results show that performance under the $T^{\text{PEB}} - \Phi^{\text{JSR}^c}$ combination approaches the lower bound of Kumar and Kumar (1994) as the nominal load approaches 1, as tabulated in Table 3. Furthermore, $T^{\text{PEB}} - \Phi^{\text{JSR}^c}$ eventually empties the exit buffers and keeps them empty thereafter, and the queue lengths in the entry buffers

[7] Note that $Q_3(\tau_2) = Q_3^o(\tau_2)$ and $Q_4(\tau_2) = Q_4^o(\tau_2)$ by definition of τ_2.

[8] Assumptions 1 and 2 are inconsistent if $m_f = m_s$.

Table 3 **Total Queue Length Obtained Through Simulation Results and the Theoretical Lower Bound** ($m_f > m_s$)

$m_f = 1.33$	PEB-JSR^c	PEB-MR	FIFO-MR	PAB-MR	Lower bound
$m_s = 0.56$					
Average	15.79	29.83	39.39	60.39	14.14
Std dev	0.40	0.39	1.51	2.01	
$m_s = 0.62$					
Average	39.71	80.09	106.34	166.52	37.58
Std dev	3.05	6.57	9.33	16.09	
$m_s = 0.65$					
Average	71.89	161.76	200.35	314.27	72.65
Std dev	7.94	13.51	23.39	31.96	

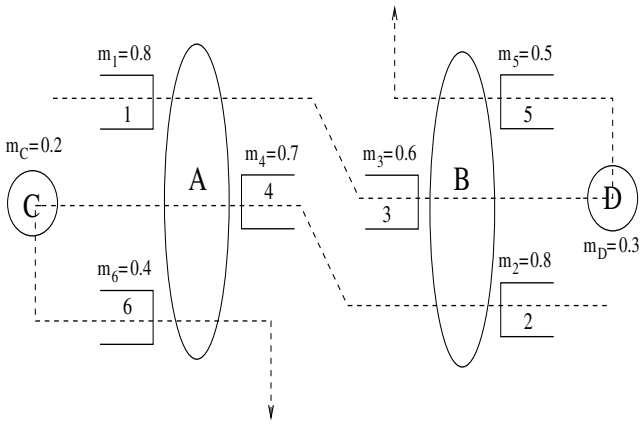
equalize eventually. As discussed in §5, this is the prescription given by the approximating Brownian control problem.[9] Thus, one can expect $T^{\text{PEB}} - \Phi^{\text{JSR}^c}$ to be asymptotically optimal under suitable scaling when $(\lambda(m_f + m_s))/2 \to 1$. In fact, it is easier to achieve the prescription of the Brownian control problem when $m_f > m_s$ because emptying the exit buffers (as done by T^{PEB}) is trivial when they are served faster. One has to work harder to get the result of Lemma 13(iii) (that is, keeping the exit buffers below threshold) when $m_f < m_s$. In addition to the lower bound of Kumar and Kumar (1994), performance of the $T^{\text{PEB}} - \Phi^{\text{JSR}^c}$ combination is compared against centralized policies as seen in Table 3. Clearly, $T^{\text{PEB}} - \Phi^{\text{JSR}^c}$ outperforms other policies.

Next, we consider extensions on the network topology. In particular, we consider the network shown in Figure 8. The analysis of the Rybko-Stolyar may tempt the reader to believe that giving priority to the faster buffer, along the lines of the classical $c - \mu$ rule, would result in an equilibrium that is at least stable if not asymptotically optimal. This is not true in general: The network shown in Figure 8 is unstable when operated under the $c - \mu$ scheduling rule that gives static priority to buffer i over buffer j if $m_i < m_j$. On inspection, the reader can verify that after a finite initial transient the network behaves as if the two fastest buffers at server A were combined into one buffer (and the same for server B), with a combined service time equal to the sum of their service times and server A (and server B) giving static priority to the combined buffer. This leads to the same instability phenomenon observed under T^{PEB} in §4. This example also shows that a larger network could exhibit instability when a subnetwork of it mimics the behavior of the examples provided.

In conclusion, the phenomena illustrated in the restrictive setting of this paper can occur in more general settings. Coming up with a systematic method for designing scheduling policies that induce Nash routing strategies that are at least stable and preferably asymptotically first best is a challenging open problem.

[9] The solution to the Brownian control problem does not depend on values of m_f and m_s.

Figure 8 **Instability Example of a Network Operating Under the $c - \mu$ Rule, Arrival Rate $\lambda = 1$**



Acknowledgments

The authors thank Haim Mendelson and Frank Kelly for their initial comments on this work. They also express their gratitude to the department editor, the associate editor, and the referees whose valuable comments helped improve the paper significantly.

Appendix

A.1. Proof of Theorem 7

We prove the theorem using the fluid limit technique pioneered by Dai (1995) and Dai and Prabhakar (2000). In particular we will use the form of the technique in Dai and Prabhakar (2000) to establish rate stability via the so-called "weak stability" of the fluid limits.

Let $N(t)$ denote the total number of arrivals to the system up to time t. Then the number of arrivals A_L to route L and A_R to route R are given by

$$A_L(t) = \sum_{n=1}^{N(t)} I_{\{\phi_n = L\}} \quad \text{and} \quad A_R(t) = \sum_{n=1}^{N(t)} I_{\{\phi_n = R\}}. \tag{A1}$$

Consider the fluid limits defined as

$$q_i(t) := \lim_{n\to\infty} \frac{1}{n} Q(nt) \quad \text{for } i = 1, 2, \ldots, 4, \tag{A2}$$

$$\tau_i(t) := \lim_{n\to\infty} \frac{1}{n} \int_0^{nt} T_i(s)\, ds \quad \text{for } i = 1, 2, \ldots, 4, \tag{A3}$$

$$a_i(t) := \lim_{n\to\infty} \frac{1}{n} A_i(nt) \quad \text{and} \quad i = L, R. \tag{A4}$$

We can argue as in Dai (1995) and Dai and Prabhakar (2000) that the limits exist almost surely, and that the limits are absolutely continuous in time and that they satisfy

$$\dot{a}_L = \lambda - \dot{a}_R, \tag{A5}$$

$$\dot{q}_1 = \dot{a}_L - \frac{\dot{\tau}_1}{m_f} \quad \text{and} \quad \dot{q}_2 = \dot{a}_R - \frac{\dot{\tau}_2}{m_f}, \tag{A6}$$

$$\dot{q}_i = -\frac{\dot{\tau}_i}{m_s} + \frac{\dot{\tau}_{i-2}}{m_f} \quad i = 3, 4, \tag{A7}$$

where $\dot{f} := df/dt$, at almost all time instants.

Lemma 13. *With probability one*
(i) $\exists T < \infty$ *such that* $|\Delta(t)| \le m_s - m_f$ *for* $t > T$;
(ii) $\lim_{t\to\infty}(A_L(t) - A_R(t))/t = 0$;
(iii) $\exists T < \infty$ *such that for* $t \ge T$, $\max(Q_3(t), Q_4(t)) \le \theta$;
(iv) $\exists T < \infty$ *such that for* $t \ge T$, $|Q_1(t) - Q_2(t)| \le 1 + m_s\theta/(m_s - m_f)$.

We postpone the proof of lemma to the end of this subsection and turn our attention to proving the theorem. For the reader unfamiliar with fluid analysis, what we are proving amounts to showing that the total queue length can increase significantly only when it is small, once the initial conditions have dissipated.

Lemma 13(ii) and the fact that $N(t)$ is a renewal process of rate λ implies $\dot{a}_L = \dot{a}_R = \lambda/2$. Lemma 13(iii) implies that $q_3 = q_4 = \dot{q}_3 = \dot{q}_4 = 0$ and Lemma 13(iv) implies that $q_1 = q_2$ and $\dot{q}_1 = \dot{q}_2$. Now suppose $q_1 > 0$ and $\dot{q}_1 > 0$, which in turn implies that $q_2 > 0$ and $\dot{q}_2 > 0$. Note that (A7) and the fact that $\dot{q}_3 = \dot{q}_4 = 0$ lead to

$$\frac{m_s}{m_f}\dot{\tau}_1 = \frac{m_s}{m_f}\dot{\tau}_2 = \dot{\tau}_3 = \dot{\tau}_4. \tag{A8}$$

(A5), (A6), the fact that $\dot{a_L} = \dot{a_R}$, and the assumption $\dot{q}_1, \dot{q}_2 > 0$ yield

$$\dot{\tau}_i < \frac{m_f}{2}\lambda \quad \text{for } i = 1, 2. \tag{A9}$$

Since the policies are nonidling,

$$\mathbf{1}_{\{q_i + q_{5-i} > 0\}}(\dot{\tau}_i + \dot{\tau}_{5-i}) = 1 \quad \text{for } i = 1, 2. \tag{A10}$$

But,

$$\begin{aligned}\dot{\tau}_i + \dot{\tau}_{5-i} &= \frac{m_s + m_f}{m_f}\dot{\tau}_i \\ &< (m_s + m_f)\frac{\lambda}{2} \\ &< 1.\end{aligned}$$

The first equality and second inequality follow by (A8) and (A9). The last inequality is due to Assumption 1, which contradicts (A10). Hence $q_i > 0$ implies $\dot{q}_i \le 0$. This in turn establishes the weak stability of the fluid limits: namely, if $q(0) = 0$, then $q(t) = 0$ for all $t > 0$. Arguing exactly as in Theorem 3 of Dai and Prabhakar (2000), we establish Theorem 7.

Proof of Lemma 13. (i) By Lemma 4, Δ changes only by idleness and arrivals. The key observation in the proof is the fact that $|\Delta(t)|$ decreases during idleness under T^{WR}. In addition, Δ increases by $m_s - m_f$ from each arrival to route L and decreases by the same amount from each arrival to route R. Arrivals join route L as long as $\Delta \le 0$ (and route R as long as $\Delta > 0$) and after $\lceil |\Delta(0)|/(m_s - m_f)\rceil$ arrivals Δ drops to $|\Delta| \le m_s - m_f$. This happens in finite time because the arrival process is a renewal process, hence $T < \infty$ almost surely. Once Δ is in $|\Delta| \le m_s - m_f$, arrivals alternate between routes[10] and Δ stays in $|\Delta| \le m_s - m_f$ for $t > T$.

(ii) A_L and A_R are formally defined in (A1). In the proof of Lemma 13(i), we have shown that arrivals alternate between two routes after some finite time T, and an arrival flips a coin to determine her route when the network is empty, i.e., $\Delta = 0$. Let C_n denote the absolute difference between number of heads and tails in n coin flips. Then,

$$|A_L(t) - A_R(t)| \le \frac{|\Delta(0)|}{m_s - m_f} + \sup_{1\le n\le N(t)} C_n.$$

The first term is the number of arrivals until Δ drops to $|\Delta| \le m_s - m_f$. The second term is an upper bound on the imbalance of coin flips. Elementary arguments about fair coin-tosses yield

$$\lim_{m\to\infty} \frac{\sup_{1\le n\le m} C_n}{m} = 0,$$

almost surely. Furthermore, $\lim_{t\to\infty} N(t)/t = \lambda$, almost surely, from which the result follows.

(iii) By part (i), for $t > T$, $Q_1(t) + Q_3(t) > 1$ implies $Q_2(t) + Q_4(t) > 0$ and $Q_2(t) + Q_4(t) > 1$ implies $Q_1(t) + Q_3(t) > 0$. This is crucial for the following proof. Note that under the WR policy, entry buffer 1 is not served when exit buffer 3 is above threshold θ.[11] The same policy applies to buffers 2 and 4. Therefore buffers 3 and 4 never receive a customer when they are above the threshold θ, hence $Q_3(t)$ and $Q_4(t)$ do not increase when they are greater than θ. One can then easily show that Q_3 and Q_4 drop below θ in finite time.

(iv) By part (i) there exists a time T_a, such that $|\Delta(t)| \le m_s - m_f$ for $t \ge T_a$ and, by part (iii), there exists a time T_b such that $|Q_3(t) - Q_4(t)| \le \theta$ for $t \ge T_b$. Then for $t \ge \max(T_a, T_b)$,

$$\begin{aligned}|Q_1(t) - Q_2(t)| &= \frac{1}{m_s - m_f}|\Delta(t) - m_s(Q_3(t) - Q_4(t))| \\ &\le \frac{|\Delta(t)|}{m_s - m_f} + \frac{m_s|Q_3(t) - Q_4(t)|}{m_s - m_f} \\ &\le 1 + \frac{m_s\theta}{m_s - m_f},\end{aligned}$$

where the first equality above follows by (9). □

A.2. Proof of Lemma 10

Figure 6 presents a snapshot of the network at $t = \tau_1$. By Corollary 8,

$$(m_s - m_f)[Q_1^o(\tau_1) - Q_2^o(\tau_1)] + m_s[Q_3^o(\tau_1) - Q_4^o(\tau_1)] = \Delta(0).$$

Consider an identical system with buffer lengths Q', such that $Q'(\tau_1) = Q(\tau_1)$, which receives the same arrival sequence as the original system. This identical system runs under a policy that deviates from the original scheduling policy only between $\tau_1 \le t \le \tau_2$, in which server B imitates

[10] The customer flips a coin to determine her route when $\Delta = 0$ (see (10)).

[11] Here we are assuming that there is at least one customer on route R, which is implied by $Q_1(t) + Q_3(t) > 1$ after some $t \ge T$.

the original system but server A processes only buffer 4.[12] Note that Corollary 8 applies to identical system for $t \le \tau_2'$, and so

$$(m_s - m_f)[Q_1'^o(\tau_2') - Q_2'^o(\tau_2')] + m_s[Q_3'^o(\tau_2') - Q_4'^o(\tau_2')] = \Delta(0).$$

The first term above is zero by definition of τ_2', hence

$$m_s[Q_3'^o(\tau_2') - Q_4'^o(\tau_2')] = \Delta(0). \tag{A11}$$

Notice that $\tau_2' = \tau_2$ and $Q_3'^o(\tau_2) = Q_3^o(\tau_2)$ as they depend only on the scheduling policy of server B, which follows the same scheduling policy in both systems. In addition $Q_4'^o(\tau_2) = Q_4^o(\tau_2) - m_f Q_3^n(\tau_2)/m_s$ because server A of the original system served $Q_3^n(\tau_2)$ new customers at buffer 1 between $\tau_1 \le t \le \tau_2$, whereas server A of identical system served an additional $m_f Q_3^n(\tau_2)/m_s$ old customers at buffer 4 during the same time period. Plugging these equalities into (A11) leads to,

$$m_s[Q_3^o(\tau_2) - Q_4^o(\tau_2)] + m_f Q_3^n(\tau_2) = \Delta(0),$$

the lemma follows from the above equality. □

A.3. Proof of Lemma 11

Notice that $Q_3^o(t) - Q_4^o(t)$ for $t > \tau_2$ increases only if server A processes buffer 4 and server B processes buffer 2. During this assignment, buffer 4 increases at a rate $1/m_f - 1/m_s$. But this assignment is not feasible if $Q_3(t) \le Q_4(t)$ because of the WR scheduling policy (see (6)). So the extra time units τ that the SM can serve buffer 4 but not buffer 3 for $t > \tau_2$ cannot exceed

$$\tau = \left[\frac{Q_3(\tau_2) - Q_4(\tau_2)}{1/m_f - 1/m_s}\right]^+. \tag{A12}$$

So, for $t \ge \tau_2$

$$\begin{aligned} &Q_3^o(t) - Q_4^o(t) \\ &\quad \le Q_3^o(\tau_2) - Q_4^o(\tau_2) + \frac{\tau}{m_s} \\ &\quad = Q_3^o(\tau_2) - Q_4^o(\tau_2) + \frac{m_f}{m_s - m_f}[Q_3(\tau_2) - Q_4(\tau_2)]^+. \quad \square \end{aligned}$$

A.4. Proof of Lemma 12

By Corollary 8,

$$(m_s - m_f)[Q_1^o(\tau_2) - Q_2^o(\tau_2)] + m_s[Q_3^o(\tau_2) - Q_4^o(\tau_2)] = \Delta(0),$$

$Q_2^o(\tau_2) = 0$ by definition and as $\Delta(0) \le 0$ is assumed by (11),

$$Q_3^o(\tau_2) - Q_4^o(\tau_2) = \frac{\Delta(0)}{m_s} - \frac{m_s - m_f}{m_s} Q_1^o(\tau_2) \le 0, \tag{A13}$$

which establishes Lemma 12(i). Furthermore

$$Q_3(\tau_2) = Q_3^o(\tau_2) \quad \text{and} \quad Q_4(\tau_2) = Q_4^o(\tau_2) \tag{A14}$$

by the definition of τ_2. Given that route L is not empty, buffer 2 is processed only if buffer 4 is below the threshold (see (6)). In other words, buffer 4 receives arrivals only if $Q_4 < \theta$, therefore $Q_4(\tau_2) \le \theta$. So,

$$Q_3(\tau_2) \le Q_4(\tau_2) \le \theta.$$

Hence buffer 4 has more customers than buffer 3 when twin B reaches that buffer. But buffer 4 cannot be processed when $Q_1(t) > 0$, $Q_3(t) < Q_4(t)$ and $Q_3(t) \le \theta$ (see (8)). In addition, $Q_4(t) - Q_3(t)$ decreases only if buffers 1 and 3 are processed simultaneously and the rate of decrease is $1/m_f - 1/m_s$. Thus, by the WR scheduling policy, buffers 1 and 3 have to be processed together at least for

$$\tau = \frac{Q_4(\tau_2) - Q_3(\tau_2)}{1/m_f - 1/m_s} \tag{A15}$$

time units before buffer 4 can receive any service.[13] Observe that (A13)–(A15) lead to

$$\tau = m_f Q_1^o(\tau_2) - \frac{m_f}{m_s - m_f}\Delta(0).$$

Let $\tau' = m_f Q_1^o(\tau_2)$. Note that $\tau' < \tau$ since $\Delta(0) \le 0$. We will later show that buffer 1 has enough customers to let buffers 1 and 3 work in tandem at least for τ' time units, before buffer 4 receives any service. For the time being assume it does.

Recall that τ_4 is the first time buffer 4 is served after $t \ge \tau_2$ (see (13)). Then, by the above discussion, for $t \ge \tau_4$

$$\begin{aligned} Q_3^o(t) - Q_4^o(t) &\le Q_3^o(\tau_2) + Q_1^o(\tau_2) - Q_4^o(\tau_2) - \frac{\tau'}{m_s} \\ &= Q_3^o(\tau_2) - Q_4^o(\tau_2) + \frac{m_s - m_f}{m_s} Q_1^o(\tau_2), \end{aligned}$$

which establishes Lemma 12(ii). Because buffer 3 is served at least τ' time units more than buffer 4, the above first inequality follows. The second equality is obtained by plugging in the definition of τ'. Note that after τ' units of service all of the old customers initially present in buffer 1 at $t = \tau_2$, ($Q_1^o(\tau_2)$ number of customers) leave buffer 1 and reach buffer 3.

Now we will establish that buffer 1 has enough reserves to support τ' time unit of tandem processing at buffers 1 and 3. Clearly, if buffer 3 is served whenever buffer 1 is served, $Q_1^o(\tau_2)$ number of old customers initially present in buffer 1 is sufficient for τ' time units of tandem processing. We claim that if buffer 1 is served and buffer 3 is not served (that is, buffer 2 is served), then buffer 1 should have received enough arrivals (new customers) to complete τ' time units of tandem processing. Note that $Q_2^o(\tau_2) = 0$ by the definition of τ_2. Hence only the new customers are served in buffer 2 after $t > \tau_2$, and we show that buffer 1 receives more arrivals (new customers) than buffer 2. Therefore, whenever buffers 1 and 2 are served simultaneously, buffer 1 should have received an arrival (a new customer), which helps it to reserve $Q_1^o(\tau_2)$ number of customers for tandem processing. All that remains to complete the proof is to show that buffer 1 receives more arrivals than buffer 2. The network does not idle until one of the twins departs from the network, so Lemma 4 applies in that $\Delta(t)$ changes only by arrivals. If arrivals do occur, they join route L until $\Delta(t)$ exceeds zero, and thereafter arrivals alternate between routes. Therefore, buffer 1 indeed receives more arrivals than buffer 2. □

[12] As will be seen in (A11), there is enough work in buffer 4 to make this feasible.

[13] If $Q_3(t) < Q_4(t) = \theta$, then buffers 1 and 3 are served immediately and uninterruptedly until $Q_3 = Q_4$.

A.5. Discussion of Externality

We only need to consider the externality caused by a twin when she is in her entry buffer. To see this, consider the following. For twin G to impose any externality when she is in her exit buffer, twin B needs to be in her entry buffer, but even compensating twin B for the externality will not help twin B to depart first. We have three cases to consider.

(i) None of exit buffers exceed the threshold before twin G departs. In this case the twins have equal amount of externality ($=m_f$) on each other.

(ii) The exit buffer of twin G exceeds threshold. In this case twin G has no externality on twin B. Because the scheduling policy offsets the amount of time spent on twin G as follows, if twin G were not processed at all at buffer 1, then buffer 4 would decrease by m_f/m_s, buffer 3 would decrease by 1. So $Q_3 - Q_4$ decreases by $1 - m_f/m_s$. That requires an additional $(1/m_f - 1/m_s)(1 - m_f/m_s) = m_f$ time units of tandem processing at buffers 1 and 3, which offsets the service time of twin G at buffer 1. The term $1/m_f - 1/m_s$ is the rate at which the number of customers increases in buffer 3 when buffers 1 and 3 are served in tandem.

(iii) The exit buffer of twin G does not exceed the threshold but the exit buffer of twin B does. In this case the externality is offset by the difference in departure times, because the route of twin G gets priority over the route of twin B at both servers as soon as the exit buffer of twin B reaches threshold.

References

Adiri, I., U. Yechiali. 1974. Optimal priority purchasing and pricing decisions in nonmonopoly and monopoly queues. *Oper. Res.* **22** 1051–1066.

Altman, E., N. Shimkin. 1998. Individual equilibrium and learning in processor sharing systems. *Oper. Res.* **46** 776–784.

Armony, M., C. Maglaras. 2004. On customer contact centers with a callback option: Customer decision, sequencing rules, and system design. *Oper. Res.* **52**(2) 271–292.

Beckmann, M., C. B. McGuire, C. B. Winsten. 1956. *Studies in the Economics of Transportation*. Yale University Press, New Haven, CT.

Bell, C. E., S. Stidham. 1983. Individual versus social optimization in the allocation of customers to alternative servers. *Management Sci.* **29** 831–839.

Braess, D. 1968. Über ein paradoxon der verkehrsplanung. *Unternehmensforschung* **12** 258–268.

Cohen, J. E., F. P. Kelly. 1990. A paradox of congestion in a queuing network. *J. Appl. Probab.* **27** 730–734.

Dai, J. G. 1995. On positive Harris recurrence of multiclass queueing networks: A unified approach via fluid limit models. *Ann. Appl. Probab.* **5** 49–77.

Dai, J. G., B. Prabhakar. 2000. The throughput of data switches with and without speedup. *Proc. IEEE INFOCOM 2002*, Vol. 2, 556–564.

Dubey, P. 1986. Inefficiency of Nash equilibria. *Math. Oper. Res.* **11**(1) 1–8.

Fudenberg, D., J. Tirole. 1991. *Game Theory*. MIT Press, Cambridge, MA.

Harrison, J. M. 1988. Brownian models of queueing networks with heterogeneous customer populations. W. Fleming, P. L. Lions, eds. *Stochastic Differential Systems, Stochastic Control Theory and Applications*. Springer, New York, 147–186.

Harrison, J. M., J. A. Van Mieghem. 1997. Dynamic control of Brownian networks: State-space collapse and equivalent workload formulations. *Ann. Appl. Probab.* **7** 747–771.

Kelly, F. P. 1991. Network routing. *Philos. Trans. Roy. Soc. London A* **337** 343–367.

Koutsoupias, E., C. Papadimitriou. 1999. Worst-case equilibria. *Proc. 16th Annual Sympos. Theoret. Aspects Comput. Sci. Lecture Notes in Computer Science*, Vol. 1563. Springer, Berlin, Germany, 404–413.

Kumar, S., P. R. Kumar. 1994. Performance bounds for queueing networks and scheduling policies. *IEEE Trans. Automatic Control* **39**(8) 1600–1611.

Kumar, P. R., T. I. Seidman. 1990. Dynamic instabilities and stabilization methods in distributed real-time scheduling of manufacturing systems. *IEEE Trans. Automatic Control* **35** 289–298.

La, R. J., V. Anantharam. 2002. Optimal routing control: A repeated game approach. *IEEE Trans. Automatic Control* **47**(3) 437–450.

Lu, C. H., P. R. Kumar. 1991. Distributed scheduling based on due dates and buffer priorities. *IEEE Trans. Automatic Control* **36**(12) 1406–1416.

Maglaras, C. 1998. Dynamic control of stochastic processing networks: A fluid model approach. Ph.D. thesis, Department of Electrical Engineering, Stanford University, Stanford, CA.

Maglaras, C., A. Zeevi. 2003. Pricing and capacity sizing for systems with shared resources: Approximate solutions. *Management Sci.* **49**(8) 1018–1038.

Maglaras, C., A. Zeevi. 2005. Pricing and design of differentiated services: Approximate analysis and structural insights. *Oper. Res.* Forthcoming.

Mendelson, H., S. Whang. 1990. Optimal incentive-compatible priority pricing for the $M/M/1$ queue. *Oper. Res.* **38** 870–883.

Naor, P. 1969. The regulation of queue size by levying tolls. *Econometrica* **37** 15–24.

Panwalker, S. S., W. Iskander. 1977. A survey of scheduling rules. *Oper. Res.* **25**(1) 45–61.

Papadimitriou, C. H., J. N. Tsitsiklis. 1996. The complexity of optimal queueing network control. *Math. Oper. Res.* **24**(2) 293–305.

Pinilla, J. M., F. B. Prinz. 2003. Lead time reduction through flexible routing: Application to shape deposition manufacturing. *Internat. J. Production Res.* **41**(13) 2957–2973.

Ridemax.com. 2004. www.ridemax.com.

Roughgarden, T., E. Tardos. 2002. How bad is selfish routing? *J. ACM* **49**(2) 236–259.

Rybko, A. N., A. L. Stolyar. 1992. Ergodicity of stochastic processes describing the operations of open queueing networks. *Problems Inform. Transmission* **28** 199–220.

Sharifnia, A. 1997. Instability of the join-the-shortest-queue and FCFS policies in queuing systems and their stabilization. *Oper. Res.* **45**(2) 309–314.

Van Mieghem, J. V. 2000. Price and service discrimination in queuing systems: Incentive compatibility of $Gc\mu$ scheduling. *Management Sci.* **46**(9) 1249–1267.

Whitt, W. 1986. Deciding which queue to join: Some counterexamples. *Oper. Res.* **34**(1) 55–62.

Whitt, W. 2003. How multiserver queues scale with growing congestion dependent demand. *Oper. Res.* **51**(4) 531–542.