Optimum Scheduling and Memory Management in Input Queued Switches with Finite Buffer Space

Saswati Sarkar*†

Abstract

This paper addresses scheduling and memory management in input queued switches with finite input buffers, with the objective of minimizing packet loss. The framework and algorithms proposed here apply to buffer constrained wireless networks as well. The scheduling problem has been extensively addressed under the assumption of infinite input buffers. We study the finite buffer case here which arises in practice. The introduction of memory constraint significantly complicates the problem. The optimal strategies for infinite buffer case no longer apply and become strictly suboptimal in presence of memory limitations. We present closed form optimal strategies which minimize packet loss in 2×2 switches with equal arrival rates for all streams. We identify certain characteristics of the optimal strategy for arbitrary arrival rates, and use these properties to design a near optimal heuristic. We obtain a lower bound for the cost associated with packet loss for $N \times N$ switches. We use this lower bound to design a heuristic for $N \times N$ switches, arbitrary N and show numerically that this heuristic performs close to optimal. The policies presented here reduce packet loss by about 25% as compared to the optimal strategy for the infinite buffer case.

1 Introduction

We consider resource allocation in input queued switches with finite memory in the input adapters. The performance objective is to minimize packet loss due to memory overflow. To the best of our knowledge this problem has not been addressed in input queued switches with memory constraints, even though its counterpart with the infinite buffer assumption has been the subject of extensive research. The finite memory constraint introduces significant additional complications, and as we discuss later the optimum strategies are significantly different from the infinite buffer case. For example, if the buffers are assumed to be infinite then the challenge is to decide the packet scheduling so as to attain the desired performance objective. However, in the finite buffer case, additionally one needs to decide whether or not to accept an arriving packet and which packet to drop in case of a memory overflow. This decision has a significant impact on the packet loss performance. Besides the packet scheduling must adapt to the memory constraints.

We first briefly review the existing literature in optimum scheduling in input queued switches. Karol *et al* [7] showed that under FIFO scheduling the throughput is limited to 58.5% of the

*S. Sarkar is with the Departments of Electrical and Systems Engineering and Computer and Information Sciences with the University of Pennsylvania. Her email is swati@ee.upenn.edu

†The work of Saswati Sarkar was supported in part by NSF grant ANI01-06984.

switching capacity on account of head of line blocking. Mckeown *et al.* [8] presented the maximum weighted matching based scheduling which attains the maximum possible throughput. We will refer to this policy as MM in the rest of the paper. Tassiulas *et al.* [15] presented a maximum difference in backlog based scheduling which attains the maximum possible throughput in any network of constrained queues. The constraints there apply to input queued switches as well. However, all these generic results apply to the case where the queues have infinite buffer space, and consequently they have not addressed memory management at all. Next, we discuss how the resource management problem differs in input queued switches when input buffers are assumed to have finite memory.

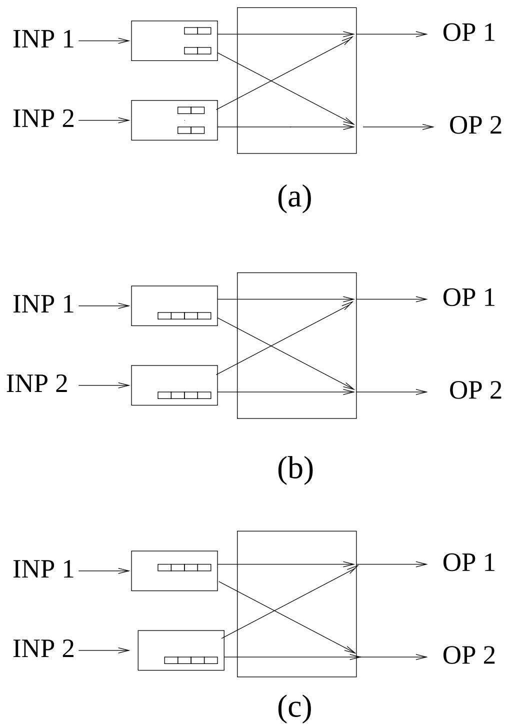

Figure 1: The figure shows a 2×2 switch with three different buffer occupancies. The buffer occupancies are $\vec{x} = (2, 2, 2, 2)$ (Figure (a)), $\vec{x} = (0, 4, 0, 4)$ (Figure (b)) and $\vec{x} = (4, 0, 0, 4)$ (Figure (c)). Only one packet can be transferred to the output at a time in Figure (b), while two packets can be transferred simultaneously in the other two cases.

The resource management problem has two components when the input buffers have finite memory: (a)packet scheduling and (b)memory management. The first decides which packets can be scheduled without violating the switching constraints imposed by the input queued switch. Input queueing introduces several scheduling inter-dependencies, and allows only certain scheduling patterns. The scheduling should be such that an input or output port is used for the transfer of only one packet at a time. Different packets can be simultaneously transferred to the outputs as long as these do not share the same input or output port. Memory management determines the packet acceptance policy, namely which incoming packet will be accepted, and which packet should be dropped in case of memory overflow (the choice is between incoming packets and existing packets). These decisions will depend on buffer occupancy and possibly arrival and service statistics. More importantly, the packet scheduling and memory management decisions must be taken in conjunction, and will depend on each other. For example, only one packet can be transferred to the output side if all outstanding packets are waiting at the same input or are destined to the same output. However, many more packets can be transferred if the outstanding packets do not have common inputs and outputs. Refer to figure 1 for an illustration. Memory management can be utilized efficiently to ensure that fewer outstanding packets share input and output ports. We discuss this in details later. To the best of our knowledge, neither problem has been addressed in presence of memory constraints. Interestingly, we observed that the MM scheduling strategy [8] which maximizes throughput in input queued switches with infinite buffers is strictly suboptimal in terms of throughput in presence of memory constraints.

Memory management has been addressed for other types of switch architectures, e.g., shared memory switches[1, 2, 6, 13]. However, scheduling is not an issue in these switches. This is because multiple packets can be simultaneously transferred to an output from different inputs. The outputs can simultaneously serve packets independent of each other. Memory is the only shared resource in these switches. Thus the problem of deciding a jointly optimal scheduling and memory management strategy does not arise, and the memory management strategies need not be designed to cater to scheduling dependencies in these switches.

Similar scheduling and memory management problems arise in optical switches [17], and these are seriously constrained in memory. Also, mathematically the scheduling and memory management problems in input queued switches are equivalent to those in finite memory nodes in wireless ad hoc networks [14]. Thus scheduling policies which attain maximum throughput in wireless networks maximize the throughput in networks of input queued switches [8, 15]. The significance of this observation is that *mathematically the problem we address in this paper forms the core of several applications of practical utility, and the solution of this problem is expected to apply to all these problems.* We elaborate this aspect later (pp. 6).

The objective of this paper is to provide a theoretical basis for the problem of jointly optimum scheduling and memory management. The investigation yields several apparently counter-intuitive results as will be described later. First we mathematically model the problem and show its close connection with the bandwidth and memory management problems in wireless adhoc networks in section 2. We present a markov decision process based framework for computing the optimum strategy in section 3. Markov decision process based computations are time and memory intensive, and the computations become intractable even for switches with moderate buffer sizes. However, using this framework we present a closed form computationally simple optimal scheduling and memory management strategy for input queued switches with 2 inputs and 2 outputs under the assumption of equal arrival rates for all input-output streams in section 4.1. The optimal scheduling transfers as many packets as possible at all times. The optimal memory management uses a packet acceptance policy which balances the number of outstanding packets for the different outputs. This in turn allows the scheduling to transfer several packets to the output and reduces the buffer occupancy and thus the packet loss in the switch. We obtain several features of the optimal strategy in 2×2 switches for arbitrary arrival rates in section 4.2, and using these properties we design a heuristic strategy (SOP). Numerical computation shows that SOP performs close to the optimal. The numerical results will also demonstrate that the MM scheduling which is known to be optimal in absence of memory constraint [8] has 25% more packet loss than SOP in many cases. We present a lower bound for the MDP cost function for the general case of switches with N inputs and N outputs in section 5, and suggest a heuristic strategy (BCT) which minimizes this lower bound. We present numerical performance evaluation to demonstrate that the packet loss experienced by BCT is close to the minimum possible value. In addition, BCT performs substantially better than MM, and in many cases reduces the packet loss by 40% or more as compared to MM. We discuss implementation challenges and future research directions in context of specific applications such as input queued switches, wireless networks and optical switches in section 6. The proofs of the technical results can be found in the appendix.

2 Network Model

We consider an input queued switch with N inputs and N outputs (Figures 1 and 2). The size of the input buffers are $B_1, \ldots, B_N$. Incoming packets have predetermined outputs and the arrival processes are independent Poisson. The arrival rate of packets at input i for output j is λ_{ij}, $i = 1, \ldots, N$, $j = 1, \ldots, N$. Packets are stored in the respective input buffers until they are transferred to the intended outputs. Queue length of packets waiting at input i for output j at time t is $x_{ij}(t)$. Clearly $\sum_{j=1}^{N} x_{ij}(t) \leq B_i$. Packets have exponential duration with the service rate for the packets at the ith input and jth output being μ_{ij}. For simplicity we will assume equal buffer sizes and equal service rates, i.e., $B_i = B$ for all i and $\mu_{ij} = \mu$ for all i, j. We will also assume that all packets which start service at the same time end service at the same time as well. Thus duration of a specific schedule is exponential with rate μ. Arrivals preempt the service, i.e., the switch chooses a new matching after an arrival. These assumptions simplify the mathematical framework and the analysis substantially, without altering the nature of the results. The assumptions can be justified from the observations that switch performance is a function of the arrival rates, service rates and the buffer sizes, and unequal buffer sizes and service rates have the same effect on the performance as unequal arrival rates. We will study the effect of unequal arrival rates. We relax many of these assumptions in simulation, e.g., we consider the effect of unequal packet sizes and heavy tailed arrival processes on the design (Figure 11).

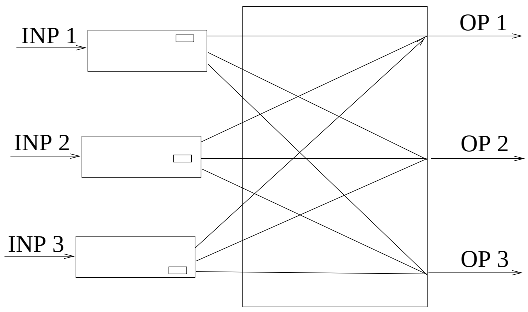

Figure 2: The figure shows a switch with 3 inputs and 3 outputs. Here, each input has only one packet waiting for transfer to the output, $x_{11} = x_{22} = x_{33} = 1, x_{ij} = 0$, if $i \neq j$. The input-output pairs $1-1, 2-2, 3-3$ constitute a matching. Similarly, $1-2, 2-1, 3-3$ is another example matching. Either matching can be scheduled, but the first transfers 3 packets while the second transfers only one packet as the $1-2$ and $2-1$ queues are empty ($x_{12} = x_{21} = 0$).

An input-output pair can be involved in transfer of only one packet at a time, i.e., while input i is transferring a packet to output j, input i can not transfer any other packet and output j can not receive any other packet. The scheduling problem is to decide which packets are transferred from the input to the output at any time t. Clearly the transfers must constitute a matching* at any time t. There can be several possible matchings and the scheduling problem at any time is to choose the appropriate matching. Refer to Figure 2 for examples of valid scheduling. Every time a packet transfer is completed from input i to output j the queue length x_{ij} decreases by one.

The memory management problem is to decide how the input buffers should be shared by packets intended for different outputs. When a new packet arrives at an input, one of the following actions can be taken: (a)the packet is accepted (b)the packet is rejected and (c)the packet is accepted while some other packet waiting at the input is dropped. The last action is commonly referred to as "push-out" (following the terminology of [1, 2, 13] in a different context). If a packet arrives when the input queue is full, then one of the last two actions must be taken. The memory management problem is to choose the appropriate course of action when a new packet arrives. Every time a new packet is accepted in input i for output j, the queue length x_{ij} increases by 1. If a packet waiting at input i intended for output j is dropped, x_{ij} decreases by one.

The objective is to choose the scheduling and the memory management scheme so as to minimize the average packet loss. A packet loss happens every time an incoming packet is rejected or an existing packet is dropped. Throughput of the switch is the average number of packets transferred from the inputs to the outputs. Throughput maximization and loss minimization are equivalent objectives, as the sum of throughput and loss rates is equal to the sum of the arrival rates. Thus a strategy which minimizes loss maximizes throughput and vice versa.

Now we argue how the finite memory assumption complicates the problem. Firstly, the issue of memory management arises only because of the finite buffer assumption. Packet loss never happens and thus packets can always be accepted for infinite buffers. However, in presence of finite buffers, which is the case in practice, packet loss happens whenever an incoming packet finds the buffer full, and thus packet acceptance and scheduling decisions need to be sophisticated so as to minimize this loss. Secondly, a scheduling which maximizes the system throughput under the infinite buffer assumption will not maximize the throughput (and hence minimize the loss) in the finite buffer case in general. This can be explained as follows. A policy is said to maximize the system throughput under the infinite buffer case if it "stabilizes" the system under maximum possible load, i.e., the policy must "stabilize" the system for all arrival rates $(\lambda_{11}, \dots \lambda_{NN})$ such that$\sum_{i=1}^{N} \lambda_{ij} < \mu$ and $\sum_{j=1}^{N} \lambda_{ij} < \mu \ \forall \ i, j$ [8, 14]. A system is said to be stable as long as the queues become empty ($x_{ij}(t) = 0 \forall i, j$) infinitely often. Note that for stability it is not important *how soon* the queues empty, but it is sufficient that the system reaches the all zero state infinitely often. However, in the finite buffer case, the average time taken to empty the queues affects the performance of the system. Consider a simple example to illustrate this fact. Consider a switch with 2 inputs and 2 outputs ($N = 2$), $\lambda_{11} = \lambda_{22} = 0.1$, $\lambda_{12} = \lambda_{21} = 0$ $\mu = 0.4$ $\forall i, j$. Consider two different policies. The first is work conserving and serves the packets in FIFO order at the output terminals. The second serves the packets in FIFO order but takes a vacation after serving every packet. The duration of the vacation is an exponential random variable with average 2.5 unit. Both of these policies stabilize†

*In graph-theoretic notions, a matching is a collection of edges which do not share a node. In the switching context a matching is a collection of input output pairs i, j which do not have a common input or output.

†The $1-1$ and $2-2$ streams are served independent of each other in both cases. The service rates are greater

the system and attain the same throughput under infinite buffer assumption (system throughput is 0.2 for both policies). However, the average duration of the busy period for the second is at least twice that for the first, and the packet loss is higher in the second for any finite buffer length. The bottomline is that the infinite buffer assumption offers the system significant latitude, while in presence of memory constraints the scheduling strategies need to use resource more carefully. Thus not all policies which optimize the performance under infinite buffer assumption will do so in the finite buffer case. In fact, as discussed before, we will demonstrate that the MM scheduling which maximizes system throughput under infinite buffer assumption [8] does not maximize throughput in the finite buffer case.

Finally, we point out the similarities between resource allocations in input queued switches and wireless adhoc networks. Incidentally, future adhoc networks are expected to consist of small, light weight terminals like PDAs, palmtops or laptops which must perform the functions of end nodes as well as intermediate routers. Thus these nodes will be constrained in both memory and bandwidth, and hence efficient usage of both resources via scheduling and memory management is necessary for meeting the performance objectives. Consider an adhoc network with nodes $1, \ldots, N$. Node i has buffer B_i. Every node has a locally unique frequency ‡, but only one radio unit. A node can be involved in at most one transmission at a time in the role of either a transmitter or a receiver on account of the single radio constraint. Several transmissions can proceed simultaneously as long as every node is involved in at most one transmission. At any time the successful transmissions must constitute a matching, and the scheduling problem is to decide the appropriate matching like in input queued switches§. The memory management problem is to decide whether to accept an incoming packet at a node and also whether an existing packet should be pushed out by a new packet. This is the same as the memory management problem in input queued switches. We conclude that the scheduling and the memory management problems are similar in both cases. This further motivates the investigation of the scheduling and the memory management problem in either case, as the solution for one may be used in the other case as well.

The optimal scheduling and memory management policy is the one which minimizes the average packet loss. We present a generic markov decision process(MDP) [11, 12] based technique for computing the optimal loss rates and the optimal scheduling and memory management decisions in section 3. However, the computations become intensive with increase in B or N. This happens because the computation needs several iterations and each iteration has a complexity $\Omega\left(B^{N^2}\right)$.¶ For example, for a 2×2 switch with $B = 50$, each iteration involves computations with 10^6 variables. The bottomline is that the generic computation technique does not scale with the buffer size. However, we will use the general results obtained from this MDP framework to compute closed form optimal strategies for the simple case of 2×2 switches where all arrival streams have equal arrival rates ($\lambda_{ij} = \lambda$ for all i, j). Subsequently we will use the insight obtained from the optimal

than the arrival rates in both cases (the service rate is $1/(2.5 * 2)$ in the second case), and thus the system is stable in both cases.

‡Several current day networks have locally unique frequencies, e.g., Bluetooth networks.

§In wireless adhoc network a node may only be able to transmit packets to a subset of other nodes as the rest may be out of its transmission range. Similarly, in input queued switches an input node may not transmit packets to all output nodes, e.g., λ_{ij} may equal zero for certain pairs i, j.

¶A computation is said to have $\Omega(n)$ complexity for input size n if it requires at least cn steps for some constant c and all large n.

strategy for this specific case and certain properties deduced from the MDP framework to design near optimal heuristics for the more general cases of 2×2 switches with arbitrary arrival rates and $N \times N$ switches with $N > 2$.

3 A Generic Framework for Computing the Optimal Strategy

We present a markov decision process (MDP)[11, 12] based framework for computing the optimal strategy. Refer to [11, 12] for details on markov decision processes. At any time t the system is characterized by the system state vector $\vec{x}(t) = (x_{11}(t), \ldots, x_{ij}(t), \ldots, x_{NN}(t))$ consisting of the queue lengths of the packets waiting at different inputs for transfer to outputs. Any state $\vec{x}$ must satisfy the following restrictions, x_{ij} are nonnegative integers and $\sum_{j=1}^{N} x_{ij} \leq B$ since the buffer for input i can store a maximum of B packets. Thus the system has a finite number of states. There is a transition to a new state whenever a packet transmission ends or a new packet arrives. The next state is determined by the system policy and the event which triggers the transition, i.e., arrival or departure.

Packet loss takes place whenever the arriving packets find the input buffer full (loss may take place even otherwise depending on the packet drop strategy). Since the objective is to minimize the packet loss, the system should avoid reaching the loss occurring states as much as possible. This can be attained by regulating the state transitions via the scheduling and memory management strategy. *A strategy or a "policy" $\vec{\pi}$ is a rule which associates certain scheduling and memory management decisions with each state.* The scheduling decision specifies which queues are scheduled for packet transfer at different states. We assume that at least one packet is served as long as the system is non-empty, i.e., the total number of packets is nonzero ("non-idling system"). The memory management decision specifies which incoming packets are accepted/rejected/pushed out at different states. *A policy is said to be markovian if the decisions depend only on the current state and not on the previous states. A policy is said to be stationary if the decisions depend only on the system state and not on the time, i.e., the decisions are the same as long as the system states are the same irrespective of the decision times. A policy is said to be deterministic if the decisions are deterministic functions of the current state.* In general, the decisions can be non-stationary, random and history dependent. However, stationary, deterministic and markovian policies are easy to implement, and fortunately there exists a stationary, deterministic and markovian policy which minimizes the packet loss[11]. We thus restrict ourselves to these policies only. Under such policies the system evolves in accordance with a markov process when the arrival is poisson and the service exponential. The scheduling and memory management rule for a stationary, deterministic and markovian policy $\vec{\pi}$ is a function of the system state and is denoted $\vec{\pi}(\vec{x})$. Different policies assign different decisions with the states, and these different decisions lead to different state transitions and hence different packet loss rates. Since $\mu > 0$, and the system has a finite number of states, it follows from the non-idling assumption that the all-zero state $\vec{0}$ ($\vec{0} = (0, \ldots, 0)$) is accessible from any other state $\vec{x}$ irrespective of the choice of the policy.

We describe the state transitions under a given policy π more formally. Let $S(\vec{x})$ be the set of possible next states when the current state is $\vec{x}$ and the next event is completion of packet transmission. Let $\mathcal{M} = \{\vec{M}_1, \vec{M}_2, \ldots \vec{M}_P\}$ be the collection of all schedulings, where an element $\vec{M}$ is described as follows: $\vec{M} = (m_{11}, \ldots, m_{ij}, \ldots, m_{NN})$ where $m_{ij} = 1$ if the queue ij is scheduled under $\vec{M}$ and $m_{ij} = 0$ otherwise. Each of the elements $\vec{M}$ represents a matching and $\mathcal{M}$ is actually a collection of all possible matchings. Now, $S(\vec{x}) = \{\vec{y} : \vec{y} = \vec{x} - \vec{M}, \vec{M} \in \mathcal{M}, y_{ij} \geq 0 \forall i, j\}$. A policy π chooses the appropriate matching and this decides the next state in this set. We denote $s_\pi(\vec{x})$ as the next state when the current state is $\vec{x}$, the next event is completion of packet transmission and the policy is π. The state transition occurs at the rate μ.

Let $A(\vec{x}, i, j)$ be the set of possible next states when the current state is $\vec{x}$ and the next event is arrival of a packet at the ith input for the jth output. The state transition happens at the rate λ_{ij}. The new packet can be accepted, rejected or accepted while dropping some other packet at the same input. Thus $\vec{y} \in A(\vec{x}, i, j)$ if either (a)$\vec{y} = \vec{x}$ or (b)$y_{ij} = x_{ij} + 1$, $y_{lm} = x_{lm}$, if $(l, m) \neq (i, j)$ or (c)$y_{ij} = x_{ij} + 1$, $y_{ik} = x_{ik} - 1$ for some $k \neq j$ such that $x_{ik} > 0$, $y_{lm} = x_{lm}$, $(l, m) \neq (i, j), (i, k)$. If input buffer i is full, (i.e., $\sum_{k=1}^{N} x_{ik} = B$), case (b) is ruled out as a new packet can be accepted only by pushing out an old packet from input i. A policy π chooses the appropriate packet acceptance/rejection/push-out decision and thereby decides the next state in this set. We denote $a_\pi(\vec{x}, i, j)$ as the next state when the current state is $\vec{x}$, the next event is arrival of a packet at the ith input for the jth output and the policy is π. Here we assume that a packet can be pushed out only from an input which has a new arrival. However, more generally packets can be pushed out from other inputs as well, and more than one packet can be pushed out. We show later that the performance objective of minimizing packet loss rules out these unnecessary packet rejections. Thus we do not consider these options now.

Now we discuss how to compute the average packet loss of any policy. The packet loss in any state for a given arrival is the total number of packets dropped including the incoming packet. This quantity is one more than the difference between the total number of packets waiting in the switch before and after the packet arrives. The packet loss experienced when the current state is $\vec{x}$, the ith input has an arrival for output j and the next state is $\vec{y}$ is denoted by the function $c(\vec{x}, \vec{y}, i, j)$, where

$$c(\vec{x}, \vec{y}, i, j) = 1 + \sum_{i=1}^{N} \sum_{j=1}^{N} (x_{ij} - y_{ij}) \tag{1}$$

The packet loss depends on the policy and the arrivals since both of these together determine the next state. For a given policy $\vec{\pi}$ the function $c(\vec{x}, \vec{y}, i, j)$ is given by $c\left(\vec{x}, a_{\vec{\pi}}(\vec{x}, i, j), i, j\right)$, and will be denoted as $c_{\vec{\pi}}(\vec{x}, i, j)$. Since we assume that not more than one packet is dropped including the incoming packet, the packet loss $c_{\vec{\pi}}(\vec{x}, i, j)$ is upper bounded by 1 for any policy $\vec{\pi}$, i, j, and state $\vec{x}$. The average packet loss under policy $\vec{\pi}$ when the system is in state x is $g_{\vec{\pi}}(\vec{x})$, where

$$g_{\vec{\pi}}(\vec{x}) = E_{i,j} c_{\vec{\pi}}(\vec{x}, i, j) = \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_{ij} c_{\vec{\pi}}(\vec{x}, i, j) \tag{2}$$

Since $c_{\vec{\pi}}(\vec{x}, i, j) \leq 1$ and $\sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_{ij} \leq 1$ this quantity is upper bounded by 1 for all $\vec{x}$. We introduce notions of discount β and "discounted loss rate" functions $J_{\beta,\vec{\pi}}(\vec{x})$ associated with every state $\vec{x}$ under a policy $\vec{\pi}$. Let $\vec{x_0}, \vec{x_1} \ldots$ be the consecutive states of the system.

$$\begin{aligned} J_{\beta,\vec{\pi}}(\vec{x}) \quad = \quad & E_{\vec{x_1}, \vec{x_2}, \ldots} \left[g_{\vec{\pi}}(\vec{x_0}) + \beta g_{\vec{\pi}}(\vec{x_1}) + \right. \\ & \left. \beta^2 g_{\vec{\pi}}(\vec{x_2}) + \ldots | \vec{x_0} = \vec{x} \right], \beta < 1, \end{aligned}$$

We explain these notions intuitively here. The 'discounted loss rate" function $J_{\beta,\vec{\pi}}(\vec{x})$ is a measure of the average loss experienced by the system if it starts from initial state $\vec{x}$ and follows the policy $\vec{\pi}$. It is a weighted sum of the average loss in the current and future states, where the average loss experienced in the kth future state is weighed by β^k, the kth power of the discount β. Intuitively, this measure of loss associates progressively lower importance with future packet loss.

Since the system has a finite number of states and the loss rates $g_{\vec{\pi}}(\vec{x})$ are upper bounded, **the packet loss probability of policy $\vec{\pi}$ is $\lim_{\beta\to 1}(1-\beta)J_{\beta,\vec{\pi}}(\vec{0})$, where $\vec{0}$ is the all-zero state**[12]. Thus the significance of the discounted loss rate is that its limiting case gives the loss probability. The discounted costs $J_{\beta,\vec{\pi}}(\vec{x})$ for all states $\vec{x}$ form the unique solution of the following system of N linear equations with N variables[11].

$$J_{\beta,\vec{\pi}}(\vec{x}) \quad = \quad g_{\vec{\pi}}(\vec{x}) + \beta\mu J_{\beta,\vec{\pi}}\left(s_{\vec{\pi}}(\vec{x})\right) + \beta \sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_{ij} J_{\beta,\vec{\pi}}\left(a_{\vec{\pi}}(\vec{x},i,j)\right) \tag{3}$$

The intuition behind this relation is that under policy $\vec{\pi}$ the discounted loss rate associated with the current state $\vec{x}$ is the sum of the average loss rate in state $\vec{x}$, $g_{\vec{\pi}}(\vec{x})$ and the average discounted loss rates of the next states. Thus the discounted loss rate experienced by the system depends on the arrival and the service rates as also on the policy which determines the next state.

The optimal policy is the one which minimizes the packet loss probability. Let Π be the set of all policies. Consider a policy $\vec{\pi}_{\beta,\mathrm{OPT}}$ which minimizes the discounted loss rate $J_{\beta,\vec{\pi}}(\vec{x})$ for all states $\vec{x}$ in the set Π. Let the minimum value of the discounted loss rate of state $\vec{x}$ be denoted as $J_{\beta,\mathrm{OPT}}(\vec{x})$. Then $J_{\beta,\mathrm{OPT}}(\vec{x})$ is the unique solution of the following equations[11]

$$J_{\beta,\mathrm{OPT}}(\vec{x}) \quad = \quad TJ_{\beta,\mathrm{OPT}}(\vec{x}) \tag{4}$$

where T is a linear operator on any function $f(\vec{x})$ defined as follows:

$$Tf(\vec{x}) \quad = \quad \min_{\vec{\pi}\in\Pi}\left(g_{\vec{\pi}}(\vec{x}) + \beta\mu f\left(s_{\vec{\pi}}(\vec{x})\right) + \beta\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_{ij} f\left(a_{\vec{\pi}}(\vec{x},i,j)\right)\right) \tag{5}$$

Thus

$$TJ_{\beta,\mathrm{OPT}}(\vec{x}) \quad = \quad \min_{\vec{\pi}\in\Pi}\left(g_{\vec{\pi}}(\vec{x}) + \beta\mu J_{\beta,\mathrm{OPT}}\left(s_{\vec{\pi}}(\vec{x})\right) + \beta\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_{ij} J_{\beta,\mathrm{OPT}}\left(a_{\vec{\pi}}(\vec{x},i,j)\right)\right) \tag{6}$$

$$\vec{\pi}_{\beta,\mathrm{OPT}}(\vec{x}) \quad = \quad \arg\min_{\vec{\pi}\in\Pi}\left(g_{\vec{\pi}}(\vec{x}) + \beta\mu J_{\beta,\mathrm{OPT}}\left(s_{\vec{\pi}}(\vec{x})\right) + \beta\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_{ij} J_{\beta,\mathrm{OPT}}\left(a_{\vec{\pi}}(\vec{x},i,j)\right)\right) \tag{7}$$

Intuitively these equations follow from the observation that at every step the policy which minimizes the discounted loss rate chooses the next state which has the minimum discounted loss rate.

Since the system has a finite number of states, and the average loss associated with each state is upper and lower bounded for each policy $\vec{\pi}$ ($0 \leq g_{\vec{\pi}}(\vec{x}) \leq 1$) and the all-zero state $\vec{0}$ is accessible from any other state $\vec{x}$ regardless of the stationary policy used, the following results hold([12] Chapter

V). **The minimum packet loss probability is given by** $\lim_{\beta\to 1}(1-\beta)J_{\beta,\mathbf{OPT}}(\vec{0})$**, where** $\vec{0}$ **is the all-zero state**. Note that using equation (2) equation (5) can be simplified as follows.

$$Tf(\vec{x}) \;=\; \sum_{i=1}^{N}\sum_{j=1}^{N} \min_{\vec{y}\in A(\vec{x},i,j)} \lambda_{ij}\left(c(\vec{x},\vec{y},i,j)+\beta f(\vec{y})\right)+\beta\mu \min_{\vec{y}\in S(\vec{x})} f\left(\vec{y}\right)) \tag{8}$$

$$s_{\vec{\pi}_{\beta},\mathrm{OPT}}(\vec{x}) \;=\; \arg\min_{\vec{y}\in S(\vec{x})} J_{\beta,\mathrm{OPT}}\left(\vec{y}\right) \tag{9}$$

$$a_{\vec{\pi}_{\beta},\mathrm{OPT}}(\vec{x},i,j) \;=\; \arg\min_{\vec{y}\in A(\vec{x},i,j)} \Big(c(\vec{x},\vec{y},i,j)+\beta J_{\beta,\mathrm{OPT}}(\vec{y})\Big) \tag{10}$$

Equations (9) and (10) specify the scheduling and packet acceptance decisions for a particular β. The strategy which minimizes average packet loss is a limiting case ($\beta \to 1$) of equations (9) and (10)([12] Chapter V). **The optimal policy** $\vec{\pi}_{\mathbf{OPT}}$ **is given by**

$$s_{\vec{\pi}_{\mathrm{OPT}}}(\vec{x}) \;=\; \lim_{\beta_n\to 1} s_{\vec{\pi}_{\beta_n},\mathrm{OPT}}(\vec{x}) \tag{11}$$

$$a_{\vec{\pi}_{\mathrm{OPT}}}(\vec{x},i,j) \;=\; \lim_{\beta_n\to 1} a_{\vec{\pi}_{\beta_n},\mathrm{OPT}}(\vec{x},i,j) \tag{12}$$

for a certain infinite sequence $\beta_1,\beta_2\ldots$ **such that** $\beta_i<1$ **for all** i **and** $\lim_{n\to\infty}\beta_n=1$.

Now we present value iteration based standard techniques for computing the solutions to the linear equations (4)[11] which will yield the optimal discounted loss rates associated with the states and thereafter the optimal strategy via equations (9) to (12). **The sequence of iterations**

$$J_{\beta,k+1}(\vec{x}) = TJ_{\beta,k}(\vec{x}), k=0,1,2,\ldots \tag{13}$$

converge to the discounted loss rates $J_{\beta,\mathbf{OPT}}(\vec{x})$ **for all states** $\vec{x}$ **irrespective of the initial choice of the** $J_{\beta,0}(\vec{x})$ **function. Thus** $\lim_{k\to\infty} J_{\beta,k}(\vec{x}) = TJ_{\beta,\mathbf{OPT}}(\vec{x})$ **for all** $\vec{x}$[11].

These iterations present a computation technique for computing the optimal loss rates and the optimal scheduling and memory management decisions. However, the iterations become computationally intensive with increase in the length of the input buffers and the number of terminals N. A $N\times N$ switch with buffer sizes B has B^{N^2} states. Thus each iteration (13) has a complexity $\Omega(B^{N^2})$. In addition, a large number of iterations may be necessary for convergence. Summarizing, this computation technique does not scale with buffer size and N. However, we use these iterations to deduce several properties of the optimal strategies in the appendix.

4 Optimal Strategy for 2×2 switches

In this section we will consider the scheduling and memory management strategies for input queued switches with 2 inputs and 2 outputs ($N=2$). We will first present the intuition behind the optimal resource allocation problem in this case. Subsequently we will present the closed form optimal strategy for symmetric traffic in subsection 4.1. Extension to the case of asymmetric traffic is considered in subsection 4.2. There we obtain certain key properties of the optimal strategy for the asymmetric traffic case, and then using these properties we design a heuristic. In subsection 4.3 we will show that this heuristic attains near optimal packet loss using numerical computations.

We first describe the optimal resource allocation problem in this case. Refer to Figure 1 for illustration. There are four different queues of packets, x_{11}, x_{12}, x_{21} and x_{22}. The queue x_{ij} stores packets at input i waiting for output j. The first two queues share the memory in input buffer 1 and the other two share the memory in input buffer 2. Arrival process is poisson for each of these queues with rates $\lambda_{11}, \lambda_{12}, \lambda_{21}$ and λ_{22} respectively and the service process is exponential at rate μ. Each buffer can store a maximum of B packets.

The pairs $[x_{11}, x_{22}]$ and $[x_{12}, x_{21}]$ can transmit packets simultaneously as the queues in these pairs do not share an input or output port. The switch can also schedule a single queue at a time. No other combination is allowed. Packet loss can be minimized if the buffer occupancy is reduced. Intuitively this happens if a pair of queues is scheduled together rather than singletons because the former removes two packets from the buffer while the latter removes only one packet from the buffer. However, if the buffer occupancy is such that one queue is empty in each pair then the switch will be forced to schedule singletons. For example, if $x_{11} = x_{21} = 0$ then the switch can schedule either x_{12} or x_{22} (Figure 1(b)). For further illustration consider the buffer occupancies in Figures 1(a) ($\vec{x} = (2, 2, 2, 2)$) and 1(b)($\vec{x} = (0, 4, 0, 4)$). Both configurations have the same queue lengths at the two inputs (4 packets) and also the total number of packets waiting for transmission is the same for both (8 packets). The switch can transmit only one packet in Figure 1(b) (from x_{12} or x_{22}) as all packets are intended for the same output (output 2). However, the switch can transmit two packets in Figure 1(a) (can schedule either the pair x_{11}, x_{22} or the pair x_{12}, x_{21} as all the queues are nonempty).

Thus a larger number of packets can be transmitted and hence packet loss can be reduced when the load is balanced across the queues. However, load should be balanced across "appropriate" queues. For example, consider the buffer occupancies in Figures 1(b) ($\vec{x} = (0, 4, 0, 4)$) and 1(c)($\vec{x} = (4, 0, 0, 4)$). Only two queues have packets in both cases and the queue lengths at the two inputs (4 packets) and the total number of packets waiting for transmission (8 packets) are the same in both. However, the switch can transfer only one packet in Figure 1(b) as discussed before, whereas two packets can be transferred in Figure 1(c) (from queues x_{11} and x_{22}) since there are packets waiting at different inputs intended for different outputs. The configuration in Figure 1(c) allows a transfer of a larger number of packets because it balances the load between appropriate queues. This can be quantified by considering the difference between the queue lengths of the queues which can be scheduled together, i.e., $|x_{11} - x_{22}|$ and $|x_{12} - x_{21}|$. These differences are $0, 0$ for Figures 1(a) and 1(c) and 4 for Figure 1(b). In general, a larger number of packets can be transferred if these differences are smaller.

Appropriate push-out policy can be used to reduce these differences. When a new packet arrives and the input buffer is full, then the packet can be rejected or accepted by pushing out an existing packet. The appropriate decision can be taken with a view to reducing this difference. Assume that $B = 4$ in Figure 1. Let a packet arrive at input 1 for output 1. In Figure 1(b) the differences between the queue lengths of the queues in the pairs remain $4, 4$ if the new packet is rejected, while the differences become $3, 3$ if the new packet is accepted while pushing out an existing packet waiting at input 1 for output 2 (in the x_{12} queue). After this replacement, both queues x_{11} and x_{22} can be scheduled transferring two packets simultaneously. In Figure 1(a) the new packet should be rejected as its acceptance using push-out will adversely affect the balance and increase the differences (packet rejection leaves the differences at $0, 0$ and push out increases the differences to $1, 1$). Summarizing, *a pair of queues should be scheduled whenever possible,* *and packet acceptance/rejection/push-out should be used judiciously to balance the load across the appropriate queues which facilitates a simultaneous transfer of two packets whenever possible.* These observations are the key behind designing optimal strategies for symmetric traffic and near-optimal heuristics for asymmetric traffic.

4.1 Optimal Strategy for Symmetric Traffic

In this subsection we will consider the symmetric traffic case. We assume that all arrival rates are equal, i.e., $\lambda_{ij} = \lambda$, for all i, j. We present the scheduling and memory management strategies which minimize the average packet loss in this case. We will refer to the optimal policy as "SOP" (**symmetric optimal policy**).

SOP Scheduling: The scheduling strategy needs to decide which queues to serve whenever the current packets finish transmission. The optimal scheduling is to serve the queues of either pair whenever possible. Let the current state be $\vec{x}$ (let $\vec{x} \neq \vec{0}$). If all the queues are nonempty ($x_{ij} > 0$ for all i, j), schedule either the $[1-1, 2-2]$ pair (x_{11}, x_{22}) or the $[1-2, 2-1]$ pair (x_{12}, x_{21}). The choice between the pairs can be arbitrary and does not affect packet loss. Now consider the case when some queues are empty. If $\min(x_{11}, x_{22}) > \min(x_{12}, x_{21}) = 0$, schedule the $[1-1, 2-2]$ pair. If $\min(x_{12}, x_{21}) > \min(x_{11}, x_{22}) = 0$, schedule the $[1-2, 2-1]$ pair. If $\min(x_{12}, x_{21}) = \min(x_{11}, x_{22}) = 0$, only one queue can be scheduled, and the longest queue is selected. The choice is arbitrary if the nonempty queues have equal lengths. If all the queues are empty ($\vec{x} = \vec{0}$), no queue is scheduled for service.

SOP Memory Management: The memory management strategy decides whether to accept an incoming packet, and if the decision is to accept the packet then whether to push out an existing packet. The optimal decision is to accept an incoming packet without any push out as long as the input buffer has space. If the input buffer is full when a new packet arrives, then the packet is either rejected or accepted while dropping an existing packet from a different queue in the same input buffer. The choice between the two is made with the objective of reducing the difference between the queue lengths of the queues which can be scheduled together. We introduce some notations for describing this part of the memory management more formally: $\text{diff}_1(\vec{x}) = x_{11} - x_{22}$, $\text{diff}_2(\vec{x}) = x_{12} - x_{21}$. Let a packet arrive at input buffer 1 for output 1 (in queue $1-1$) and let input buffer 1 be full ($x_{11} + x_{12} = B$). Then the new packet is rejected if $\text{diff}_1(\vec{x}) \geq \text{diff}_2(\vec{x}) - 1$, accepted and a packet of queue $1-2$ is dropped otherwise. Let $x_{11} + x_{12} = B$, and a new packet arrives at queue $1-2$. Then the new packet is rejected if $\text{diff}_2(\vec{x}) \geq \text{diff}_1(\vec{x}) - 1$, accepted and a packet of queue $1-1$ is dropped otherwise. Let $x_{21} + x_{22} = B$, and a new packet arrives at queue $2-1$. Then the new packet is rejected if $\text{diff}_2(\vec{x}) \leq \text{diff}_1(\vec{x}) + 1$, accepted and a packet of queue $2-2$ is dropped otherwise. Let $x_{21} + x_{22} = B$, and a new packet arrives at queue $2-2$. Then the new packet is rejected if $\text{diff}_1(\vec{x}) \leq \text{diff}_2(\vec{x}) + 1$, accepted and a packet of queue $2-1$ is dropped otherwise. An illustrative example follows.

Example 4.1: We illustrate SOP using the configurations in Figure 1. Let $B = 4$. In Figure 1(a) ($\vec{x} = (2, 2, 2, 2)$), SOP schedules either the pair $[1-1, 2-2]$ or the pair $[1-2, 2-1]$ choosing between them arbitrarily. Any incoming packet is rejected. In Figure 1(b) ($\vec{x} = (0, 4, 0, 4)$), only one queue can be scheduled. SOP schedules either $1-2$ or $2-2$ choosing between them arbitrarily since both have the same length (4 packets). At either input, an incoming packet for output 1 is accepted by pushing out a packet waiting for output 2 at the same input, and an incoming packet for output 2 is rejected. In Figure 1(c) ($\vec{x} = (4,0,0,4)$), SOP schedules the pair $1-1, 2-2$. Any incoming packet is rejected. Now let $B = 5$. The scheduling remains the same as before in each case. The memory management decision is to accept any incoming packet in all three cases.

The following theorem outlines the optimality of SOP.

Theorem 1 *SOP minimizes the average packet loss in 2×2 switches with equal arrival rates for all streams ($\lambda_{ij} = \lambda$ for all i, j).*

4.2 Resource allocation for asymmetric traffic

In this section we will consider the general case with unequal arrival rates for different arrival streams, i.e., we no longer assume that $\lambda_{ij} = \lambda$. First, we identify certain properties of the optimal scheduling and memory management strategy, and subsequently present a heuristic using these properties.

Theorem 2 *For any 2×2 input queued switch, the optimal scheduling schedules a pair of queues whenever possible, and the optimal memory management decision is to accept an incoming packet without any push out as long as the input buffer has space.*

These properties specify the strategy in certain cases. For instance, in Figure 1(c) the optimal scheduling is to serve the pair $1-1, 2-2$. Also, if $B = 5$ then the optimal memory management strategy accepts all incoming packets for all three configurations in Figure 1. However, these properties do not completely specify the optimal strategy. For example, it is not known how to choose between pairs of queues when either pair can be scheduled (e.g., Figure 1(a)), or how to choose a queue when the system state is such that only one queue can be scheduled, i.e., when $\min(x_{12}, x_{21}) = \min(x_{11}, x_{22}) = 0$ (e.g., Figure 1(b)). The packet acceptance/rejection/push-out decision when the input buffer is full is also not known in the asymmetric case (e.g., all three cases in Figure 1 with $B = 4$).

We propose SOP (Section 4.1) as a heuristic here. We justify the choice as follows. SOP satisfies the properties obtained for the optimal strategy in this general case. Besides, the scheduling and the memory management decisions of SOP strive to balance the traffic across different queues which allow the scheduling to transfer larger number of packets and thus reduce the buffer occupancy and the packet loss. Numerical computations presented in the next subsection will demonstrate that the heuristic attains near minimum packet loss.

We have obtained some other properties of the optimal strategy under the additional assumption that $\lambda_{i1} = \lambda_{i2}$ $i = 1, 2$. We still allow $\lambda_{1j} \neq \lambda_{2k}$ for any j, k i.e., unequal arrival rates at different inputs.

Theorem 3 *If $\lambda_{i1} = \lambda_{i2}$ $i = 1, 2$, then the optimal strategy has the same memory management policy as SOP. The optimal scheduling is the same as the SOP scheduling except for the case when $x_{12} = x_{22} = 0$ or $x_{11} = x_{21} = 0$.*

Under this additional assumption, the optimal strategy is known for all cases in Example 4.1.1, except the scheduling in Figure 1(b). More generally, this additional assumption fully specifies the optimal memory management scheme. The scheduling policy is also specified except that it is not known how to choose between queues when both the queues intended for one output terminal are empty, i.e., when $x_{12} = x_{22} = 0$ or $x_{11} = x_{21} = 0$ (e.g., Figure 1(b)).

4.3 Performance Evaluation using Numeric Computation

First we evaluate the performance of SOP for unequal arrival rates, using the packet loss experienced by the optimal strategy as a benchmark. We compute the average packet losses of SOP and the optimal strategy using MDP based techniques which we described in Section 3. We have observed that SOP attains near minimum average packet loss. We present numerical results for two different traffic patterns.

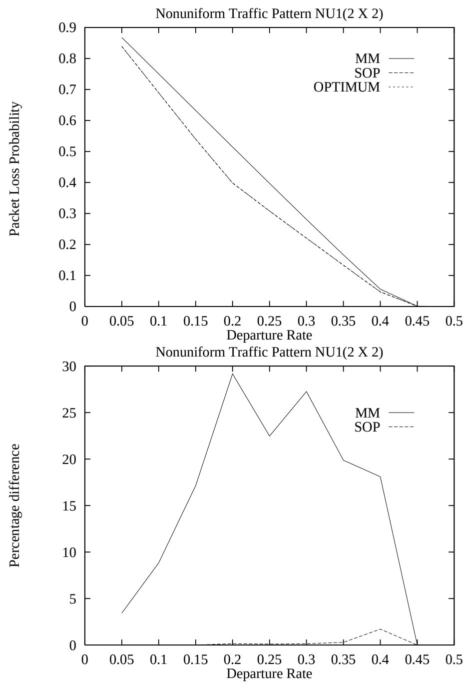

Figure 3: The figures compare the performances of SOP and MM with the optimal strategy for different departure rates (μ) in a 2×2 switch with $B = 50$ and the following arrival rate patterns: $\lambda_{11} = 3.5\lambda_{12} = 3.5\lambda_{21} = 7\lambda_{22}, \lambda_{22} = \frac{1-\mu}{15}$. Figure (a) plots the absolute values of the packet loss probabilities ($l_{\text{MM}}, l_{\text{SOP}}, l_{\text{OPTIMUM}}$). The SOP and the OPTIMUM curves can not be distinguished, while the MM curve is above both. Figure (b) plots the percentage relative difference between the loss probabilities for the SOP and optimal (curve SOP) $(100 * (l_{\text{SOP}}/l_{\text{OPT}} - 1))$, and that between MM and SOP (curve MM) $(100 * (l_{\text{MM}}/l_{\text{SOP}} - 1))$. The SOP curve is close to zero, and hence hardly visible.

When all arrival rates are equal, there is no difference between the arrival rates of queues in the pairs which can be scheduled together, and as discussed these differences are important metrics. Thus in nonuniform traffic model, we study the cases when the differences between arrival rates of queues in these pairs are nonzero. First, we consider the case where the arrival rates are unequal for one pair, and equal for the queues in the other pair. Specifically, $\lambda_{11} = 7\lambda_{22}$, $\lambda_{12} = \lambda_{21}$ and $\lambda_{11} = 3.5\lambda_{12}$. We denote this pattern as "NU1." We compare the performances of SOP with the optimal for different values of the departure rates (μ) (Figure 3), and for each μ we choose the arrival rates such that the above ratios are satisfied and $\mu + \sum_{i=1}^{2}\sum_{j=1}^{2}\lambda_{ij} = 1$ ($\lambda_{22} = \frac{1-\mu}{15}$

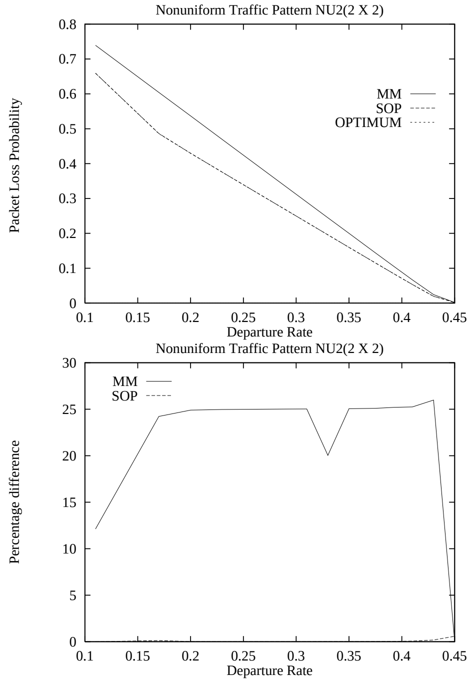

Figure 4: The figures compare the performances of SOP and MM with the optimal strategy for different departure rates (μ) in a 2×2 switch with $B = 50$ and the following arrival rate patterns: $\lambda_{11} = \lambda_{21} = 4\lambda_{12} = 4\lambda_{22}, \lambda_{22} = \frac{1-\mu}{10}$. Figure (a) plots the absolute values of the packet loss probabilities($l_{\mathrm{MM}}, l_{\mathrm{SOP}}, l_{\mathrm{OPTIMUM}}$). The SOP and the OPTIMUM curves are very close to each other and can not be distinguished, while the MM curve is above both. Figure (b) plots the percentage relative difference between the loss probabilities for the SOP and the optimal (curve SOP) $(100*(l_{\mathrm{SOP}}/l_{\mathrm{OPT}}-1))$, and that between MM and SOP (curve MM) $(100*(l_{\mathrm{MM}}/l_{\mathrm{SOP}}-1))$. The SOP curve is close to zero, and hence hardly visible.

ensure this). Figure 3 show that the performances of the two are almost identical in most cases with the difference between the loss probabilities less than 2% for all μ (the curves can not be distinguished). Next, we consider the case where the arrival rates are unequal for both pairs, while the total arrival rates are equal for both inputs, i.e, $\lambda_{11} + \lambda_{12} = \lambda_{21} + \lambda_{22}$ (this did not hold in the previous case). Here, $\lambda_{11} = 4\lambda_{22}$, $\lambda_{21} = 4\lambda_{12}$ and $\lambda_{11} = \lambda_{21}$. We denote this pattern as "NU2." Again, we vary μ and for each μ we choose the arrival rates such that the above ratios are satisfied and $\mu + \sum_{i=1}^{2}\sum_{j=1}^{2}\lambda_{ij} = 1$ ($\lambda_{22} = \frac{1-\mu}{10}$ ensure this). Figure 4 show that the performance of SOP is near optimal all through, again the curves can not be distinguished. The trends remain similar for other patterns of arrival rates which we do not exhibit on account of space constraints.

We believe that the minor difference in the packet losses of SOP and the optimal strategy is principally due to the sub-optimality of the packet acceptance/rejection/push-out decision of SOP in the general case. Consider traffic pattern NU1 as an example. Let $\vec{x}$ be the current state. Since the $1-1$ stream has significantly higher arrival rate as compared to other streams, the optimal strategy should push out the packets of the $1-2$ stream less frequently than that of the $1-1$ stream when input buffer 1 is full, i.e., $1-2$ packets should be pushed out only if $\text{diff}_1(\vec{x})$ is much smaller than $\text{diff}_2(\vec{x})$. This is because the $1-1$ stream will receive more packets than the $1-2$ stream in future. However, SOP pushes out the $1-2$ packets whenever a $1-1$ packet finds input buffer 1 full and $\text{diff}_2(\vec{x}) > \text{diff}_1(\vec{x}) + 1$, even if the difference between the two sides is small. But again, a state $\vec{x}$ where $\text{diff}_2(\vec{x}) > \text{diff}_1(\vec{x}) + 1$ is rarely reached since the $1-1$ stream has significantly high arrival rate as compared to other streams. Thus there is only slight sub-optimality. Also, depending on the arrival rates, unlike SOP, the optimal strategy may not schedule the longest queue whenever only a single queue can be scheduled. Once again, the sub-optimality is minimal.

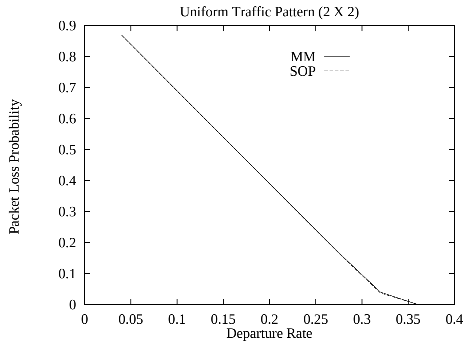

Figure 5: The figure plots the packet loss probabilities of MM and SOP for different departure rates (μ), and equal arrival rates $\lambda_{ij} = \frac{1-\mu}{4}, \forall i, j$ in a 2×2 switch with $B = 50$. Note that SOP is optimal in this case (Theorem 1).

Now we consider the MM policy which schedules packets as per a maximum weighted matching. This policy has been proposed for the infinite buffer case and attains the maximum possible throughput there [8]. Memory management is not an issue under infinite buffer assumption, and as such MM does not address it. We consider the obvious extension, whereby MM always accepts a packet whenever there is space in the buffer and always rejects a packet if the input buffer is full. Push out is not used. Average packet loss experienced by this policy can be computed using MDP based techniques (section 3). Numerical computation shows that MM has considerably higher

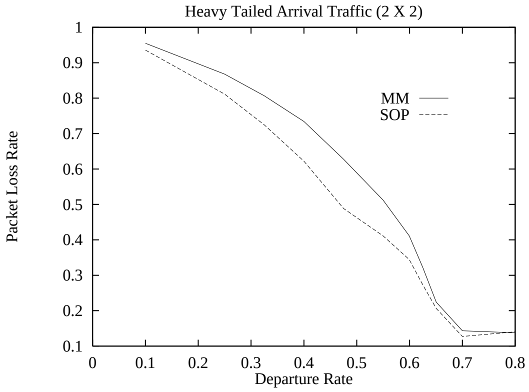

Figure 6: The figure compares the performances of SOP and MM for different departure rates (μ) in a 2×2 switch for heavy tailed arrival distribution with $B = 500$. Each arrival brings a Pareto distributed burst of packets, mean burst size 3, $\alpha = 1.24$. Arrival epochs are poisson with rates: $\lambda_{11} = 3.5\lambda_{12} = 3.5\lambda_{21} = 7\lambda_{22}$, $\lambda_{22} = \frac{1-\mu}{15}$.

packet loss than SOP for unequal arrival rates. For traffic patterns NU1 and NU2 (Figures 3 and 4 respectively), SOP decreases the packet loss by more than 25% in certain cases. Note that the difference brought about by intelligent resource allocation is pronounced for medium departure rates. For large departure rates, all policies attain low packet loss and all policies suffer from heavy packet loss in the other extreme. Thus MM and SOP perform similarly in these two extremes, while SOP substantially outperforms MM in the middle. We observed similar trends for other unequal arrival rate patterns as well. When arrival rates are equal, MM is consistently inferior to SOP for all departure rates (Figure 5). However, the difference in performance is small in this case, and thus the curves look identical in the figure. Figure 6 compares the performance of SOP and MM for bursty Poisson arrivals, with the bursts having a Pareto distribution ($F(x) \leq 1-(b/x)^{\alpha}, b = 0.6, \alpha = 1.24$). The trends are similar to those for Poisson traffic, though the packet loss rates are higher for both SOP and MM in this case.

The relative performance of MM w.r.t. SOP is much worse for nonuniform traffic (Figures 3 and 4) than for uniform traffic (Figure 5). This happens as the queues tend to be heavily dis-balanced when arrival rates are unequal, while the discrepancy is less for equal arrival rates. This load mismatch worsens the performance for MM. Thus MM experiences much heavier packet loss for non-uniform traffic than for uniform traffic. The memory management in SOP restores a balance in the queue lengths by replacing packets of more heavily loaded queues with those of the lightly loaded queues whenever possible, and this retains the performance for nonuniform traffic at the same level as for the uniform traffic case. As an example, at service rate 0.2, sum of the arrival rates ($\sum_{i=1}^{2}\sum_{j=1}^{2}\lambda_{ij}$) 0.8, MM has 39.1% and 51.5% packet loss for equal arrival rates and arrival pattern NU1 respectively, while the numbers are 38.95% and 39.87% for SOP. This indicates that SOP is robust to different traffic patterns (note that neither SOP nor MM uses statistics information in the decision process).

Finally we discuss the implications of the sub-optimality of MM. The main reason for sub-optimality of MM is that it does not use any intelligent memory management scheme which is important for performance optimization when buffers are finite. The optimal scheduling is to schedule a matching of maximum size or a "maximum matching" for both symmetric and asymmetric traffic, which is different from the maximum weighted matching MM uses. We explain MM scheduling to illustrate this. The matchings for a 2×2 switch are the single queues $[1-1], [1-2], [2-1], [2-2]$ and the pairs $[1-1, 2-2]$, $[1-2, 2-1]$. Weight of a matching is the sum of the queue lengths of the queues in the matching. MM schedules the queues which constitute the matching with the maximum weight. Let the current state be $\vec{x}$. Suppose $x_{22} = 0$, $x_{11} > x_{12} + x_{21}$ and $x_{12} > 0, x_{21} > 0$ (e.g., $\vec{x} = (5, 2, 2, 0)$). MM schedules only the queue $1-1$, while the optimum scheduling schedules the pair $1-2, 2-1$. Thus the optimum scheduling transfers a larger number of packets than MM. This reduces the buffer occupancy and hence the packet loss for the optimal strategy. MM attains optimum throughput in the infinite buffer case in spite of transferring fewer number of packets because the infinite buffer assumption offers more latitude than the finite buffer case as discussed before. Interestingly, an example provided by Mckeown *et al.* shows that for infinite buffers maximum matching scheduling is strictly suboptimal in terms of throughput [8], whereas maximum weighted matching attains optimum throughput. Apparently, it is somewhat counterintuitive that the result will be opposite in the finite buffer case, particularly since infinite buffer assumption is the limiting case of finite buffers. This apparent contradiction can be explained by two observations. (a)The example was provided for 3×3 switches while the optimal scheduling properties presented so far are for 2×2 switches. (b)The example showed that the choice of a *specific* maximum matching is suboptimal for unequal arrival rates and does not show that every maximum matching based scheduling is suboptimal. There can be several maximum matchings. The analytical results obtained here show that the optimal scheduling is to choose "some" maximum matching or rather to choose a pair of queues whenever possible. The choice between the pairs can be arbitrary for symmetric traffic, but the choice may matter for asymmetric traffic.

μ	Optimal	SOP	MM	MS
0.05	0.839162	0.839166	0.867963	0.866062
0.15	0.539816	0.539829	0.632334	0.627921
0.25	0.307703	0.308039	0.397353	0.395043
0.35	0.132880	0.133260	0.166323	0.166359
0.40	0.006317	0.007310	0.008161	0.008339

Table 1: The table shows packet loss rates for different management strategies. for different departure rates. We consider poisson arrivals with rates $\lambda_{11} = 3.5\lambda_{12} = 3.5\lambda_{21} = 7\lambda_{22}$, $\lambda_{22} = \frac{1-\mu}{15}$.

We compare the performance of SOP with some other scheduling strategies, e.g., choosing the maximum weighted matching of maximum size (MS) etc. Like MM, MS accepts packets as long as input buffer has space, and rejects incoming packets whenever the input buffer is full. It does not use any pushout. MM and MS have similar loss rates (table 1), which indicates that the performance advantage of SOP is primarily due to the load balancing attained by the memory management.

5 Resource Allocation for $N \times N$ switches for arbitrary N

We consider the scheduling and memory management problems in switches with N inputs and N outputs for arbitrary N. The objective is to minimize the packet loss. The optimal strategy can be computed using MDP (Section 3). However, no closed form solution is known for this case. Since computations using MDP are complex, we propose a computationally simple heuristic in this case and show using numerical computation that this heuristic attains low packet loss. This heuristic has been motivated by a lower bound on the MDP cost function presented in Lemma 1.

The objective of the scheduling policy is to transfer as many packets as possible so as to minimize the packet loss, and memory management decides which packets to accept, reject and push-out so as to allow the transfer of several packets in every scheduling. Consider the example shown in Figure 2. All three packets can be transferred at the same time since they are waiting at different inputs and intended for different outputs. However, if all three were waiting at the same input, only one can be transferred. Thus like in the special case for 2×2 switches we need to balance the load across appropriate queues. The SOP memory management attains this in 2×2 switches by accepting/rejecting/pushing out packets so as to reduce the difference between queue lengths of the queues which can be scheduled together($[1-1, 2-2]$ and $[1-2, 2-1]$). For $N > 2$ more than two queues can be scheduled together, e.g., $1-1, 2-2, 3-3$ can be scheduled together in the 3×3 switch showed in Figure 2. Thus the notion of reducing these differences can not be extended directly for $N > 2$.

We consider a new measure of congestion, the maximum of the total number of packets waiting at an input and waiting for an output. We denote this measure as $\text{cong}(\vec{x})$, where $\text{cong}(\vec{x}) = \max(\max_i \sum_{j=1}^{N} x_{ij}, \max_j \sum_{i=1}^{N} x_{ij})$, and $\vec{x}$ is the current state. Note that $\text{cong}(\vec{x}) = 1$ in Figure 2, $\text{cong}(\vec{x}) = 3$ if all three packets were waiting at the same input, or waiting for the same output in Figure 2. Three packets can be transferred to the outputs simultaneously in the first case, while only one packet can be transferred in the latter cases. Thus intuitively larger number of packets can be transferred when $\text{cong}(\vec{x})$ is smaller. We propose a strategy which minimizes this congestion measure. First we justify why $\text{cong}(\vec{x})$ is a measure of goodness for a state $\vec{x}$.

1. If the current state is $\vec{x}$, every packet has duration T units and there is no future arrival, then the minimum time taken to transfer all waiting packets is $T\text{cong}(\vec{x})$ [5, 16].

2. We have shown in section 3 that the optimal policy minimizes a parametrized function $J_{\beta,\text{OPT}}(\vec{x})$ referred to as the "discounted loss rate function" for each state $\vec{x}$ and all values of a certain parameter $\beta < 1$. We show here that under heavy traffic this function is lower bounded by $\text{cong}(\vec{x})$ for each state $\vec{x}$ and each $\beta < 1$.

   Lemma 1 *For all* $\beta \in [\frac{N*B}{N*B+\min(\min_i \sum_{j=1}^{N} \lambda_{ij}, \min_j \sum_{i=1}^{N} \lambda_{ij})-\mu}, 1)$, *and all states* $\vec{x}$, $J_{\beta,OPT}(\vec{x}) \geq \text{cong}(\vec{x})$ *if the total arrival rate at(intended for) each input(output) is greater than the service rate, i.e.,* $\min(\min_i \sum_{j=1}^{N} \lambda_{ij}, \min_j \sum_{i=1}^{N} \lambda_{ij}) > \mu$.

   This indicates that low packet loss can be obtained by choosing the next state so as to minimize this measure of congestion.

We present the heuristic now. The scheduling is to schedule the matching of largest size which minimizes $\text{cong}(\vec{y})$ of the next state $\vec{y}$. Note that there can be several maximum matchings or matchings of the largest size. We choose a specific maximum matching which minimizes the congestion measure of the next state. Under such a matching, the congestion measure reduces by one whenever the currently scheduled packets complete transmission. Fast algorithms can be found for computing such matchings using the theory of edge coloring of bipartite graphs [3]. The memory management policy accepts packets without pushing out any existing packet as long as the input buffer has space (in fact this is the optimal decision as shown in appendix C). We consider the acceptance decision when an incoming packet finds the input buffer full. Let the current state be $\vec{x}$. Suppose the packet is for input i and output j. We consider the total number of packets waiting for each output across all inputs. Let this number be maximum for output k among those outputs for which at least one packet wait at input i, i.e., $k = \arg\max_{m:x_{im}>0} \sum_{l=1}^{N} x_{lm}$. The incoming packet is rejected if $\sum_{l=1}^{N} x_{lj} \geq \sum_{l=1}^{N} x_{lk} - 1$, and accepted while pushing out a packet waiting for output k at input i otherwise. In other words the acceptance policy reduces the load for a more heavily loaded output at the expense of that for a lightly loaded output. Overall, the policy minimizes the congestion at both inputs and outputs, and thus we denote it as a policy which balances congestion at terminals (BCT). We present an example to illustrate this policy.

Example 5.1: Consider a 3×3 switch. Let the current state be $\vec{x}$ with $x_{11} = 2, x_{12} = x_{21} = x_{32} = 1$. All other queues are empty. Note that at most 2 packets can be transferred, and the matchings which transfer two packets are $[1-1, 3-2]$, $[1-2, 2-1]$, $[2-1, 3-2]$. The next states have congestion measures $2, 2, 3$ respectively for these choices. Thus either the first or the second matching can be selected. Suppose $B = 3$. New arrivals at inputs 2 and 3 are accepted without any push outs. Consider a new arrival at input 1 which is full. If the packet is for output 1 or 2, it is rejected. However, a new arrival for output 3 is accepted while pushing out a packet for output 1, since $\sum_{m=1}^{3} x_{m1} > \sum_{m=1}^{3} x_{m3} + 1$.

BCT applies for 2×2 switches as well, and takes the same decisions as SOP there. Weller *et al.* [16] proposed a similar scheduling strategy, and analyzed its throughput in presence of infinite buffers and pseudo-deterministic traffic arrival (a "$\alpha - S$" traffic model) [16]. Iyer *et al.* [4] showed a similar scheduling strategy to be throughput optimal in $N \times N$ switches. However, memory constraints were not considered in these, and as such the memory management component has not been proposed.

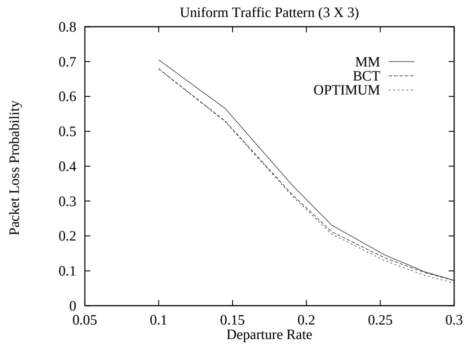

Figure 7: The figure plots the packet loss probabilities of the optimum strategy, BCT and MM for different departure rates (μ), and equal arrival rates $\lambda_{ij} = \frac{1-\mu}{9}, \forall i, j$ in a 3×3 switch with $B = 5$. The packet losses are almost identical for BCT and the optimum, while MM has a slightly higher packet loss than BCT.

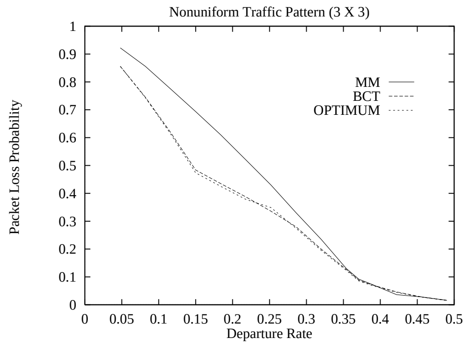

Figure 8: The figure compares the performances of the optimum strategy, BCT and MM for different departure rates (μ) in a 3×3 switch with $B = 5$ and the following arrival rate patterns: $\lambda_{11} = 9\lambda, \lambda_{ij} = \lambda, (i,j) \neq (1,1)$, $\lambda = \frac{1-\mu}{17}$. The packet losses are almost identical for BCT and the optimum, while MM has substantially higher packet loss than BCT.

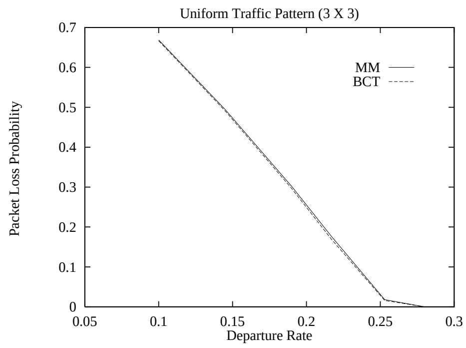

Figure 9: The figure plots the packet loss probabilities of BCT and MM for different departure rates (μ), and equal arrival rates $\lambda_{ij} = \frac{1-\mu}{9}, \forall i, j$ in a 3×3 switch with $B = 50$. MM has a slightly higher packet loss than BCT.

Average packet loss experienced by this policy can be computed using MDP based techniques (section 3). We present the numerical performance evaluations for a 3×3 switch. We first compare the performance of BCT with the optimal. However, computation of the packet loss experienced by the optimal using MDP involves several iterations, each involving B^9 operations for buffer size B. We could thus compare the performances for small values of B ($B = 5$) only. First consider the case of uniform traffic $\lambda_{ij} = \lambda, \forall i, j, \lambda = \frac{1-\mu}{9}$ for scaling purposes. Figure 7 shows the performance curves. Now consider a nonuniform traffic pattern where the arrival rate in one queue is significantly higher than others. Specifically, $\lambda_{11} = 9\lambda$, $\lambda_{ij} = \lambda$ for $(i, j) \neq (1, 1)$ and $\lambda = (1 - \mu)/17$ (the last relation scales the sum of the arrival and departure rates to 1). Figures 7 and 8 show that in both cases the packet loss of BCT is close to that of the optimum for all departure rates. MM performs inferior to BCT in both cases (the discrepancy is substantially higher for nonuniform traffic case). Investigations for other nonuniform traffic patterns indicate the same trends.

Buffer size of only 5 units may be small for drawing reliable conclusions. Thus we compare the performance of both BCT and MM via simulation for larger B, $B = 50$. Note that MDP based

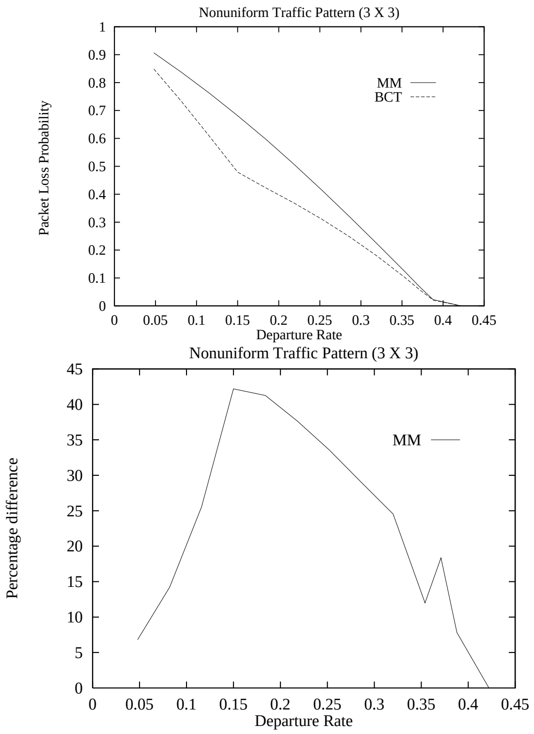

Figure 10: The figures compare the performances of BCT and MM for different departure rates (μ) in a 3×3 switch with $B = 50$ and the following arrival rate patterns: $\lambda_{11} = 9\lambda, \lambda_{ij} = \lambda, (i,j) \neq (1,1)$, $\lambda = \frac{1-\mu}{17}$. Figure (a) plots the absolute values of the packet loss probabilities (l_{MM}, l_{BCT}). Figure (b) plots the relative increase of the packet loss probability of MM w.r.t. BCT ($100 * l_{\mathrm{MM}}/l_{\mathrm{BCT}} - 1$).

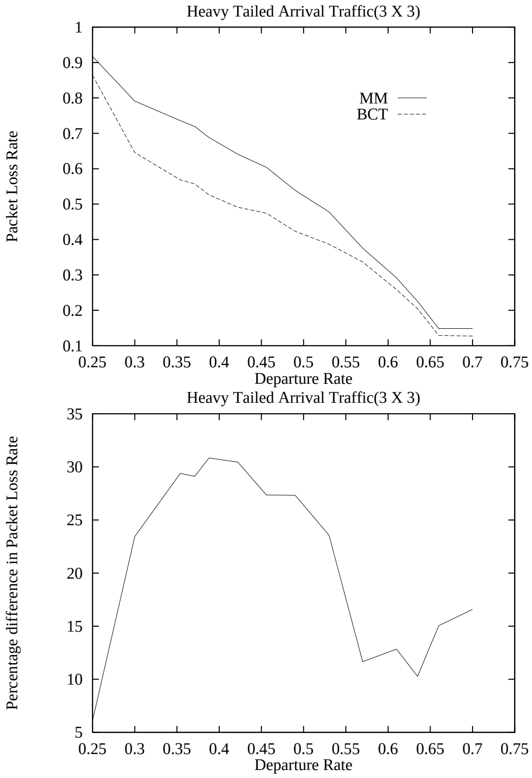

Figure 11: The figure compares the performances of BCT and MM for different departure rates (μ) in a 3×3 switch for heavy tailed arrival distribution with $B = 500$. Each arrival brings a Pareto distributed burst of packets, mean burst size 3, $\alpha = 1.24$. Arrival epochs are poisson with rates: $\lambda_{11} = 9\lambda, \lambda_{ij} = \lambda, (i,j) \neq (1,1)$, $\lambda = \frac{1-\mu}{17}$. The first figure plots the loss rates of the two strategies, while the second plots the relative increase of the packet loss probability of MM w.r.t. BCT $(100 * l_{\mathrm{MM}}/l_{\mathrm{BCT}} - 1)$.

computations need to consider 50^9 operations in each iteration, and are thus computationally infeasible. Thus we do not know the optimal packet loss rates. Nevertheless, simulation results demonstrate the improvement obtained by BCT over MM. In the uniform traffic case, MM is slightly inferior to BCT (Figure 9), while in the nonuniform traffic case the difference is substantial (Figure 10). BCT decreases the packet loss rate by more than 40% over MM (Figure 10(b)) in certain cases! Once again, we observe similar differences in performances between BCT and MM for other nonuniform traffic patterns we considered. For instance, Figure 11 shows the performance difference for bursty Poisson arrivals, with the bursts having a Pareto distribution ($F(x) \leq 1 - (b/x)^{\alpha}, b = 0.6, \alpha = 1.24$). Like in the 2×2 case, the load balancing brought about by the memory management of BCT safeguards against the detrimental effect of the difference in arrival rates, while MM has no such protection, and this explains the difference in performance for nonuniform traffic patterns. We conclude that BCT is more robust than MM.

6 Discussion and Future Research Issues

We mention potential approaches for addressing several implementation related challenges of the proposed algorithms, in context of the switching and wireless applications. We hope that this paper will entice further research required to "nail down these issues."

Real time traffic demands delay guarantees, and thus loss minimization may not be enough. Packet deadline satisfaction can be facilitated in the proposed design if the oldest packet is discarded (or replaced by a new packet) from a stream during pushout (note that the push out strategy decides the queue for push out and any packet can be discarded from the chosen queue). Another approach is to deviate from the loss optimal scheduling and serve a packet if it is close to deadline expiry.

Computing the desired matching for every scheduling decision may become computationally intensive particularly for high speed operations. Further research is required for designing a computationally simple scheduling which attains low packet loss. A possible direction is to proceed along the results obtained by Mekkittikul et al. [9] and Tassiulas [14] in simplifying the throughput optimal scheduling strategies for switches with infinite buffers.

The proposed memory management uses packet push-out which may become difficult from an implementation point of view, particularly for optical switches. An interesting topic for future research is to design a memory management strategy which optimizes packet loss under the constraint that existing packets can not be discarded. In line with the results obtained for similar objective in shared memory switches [6], the optimal strategy is likely to accept packets selectively even if the input buffer has space. We believe that a threshold based packet acceptance policy will be required.

Our objective has been to minimize packet loss. We do not address fairness issues in this paper. For example, in Figure 4 the proposed strategy may starve the queue x_{12} if the queue x_{21} has no packet, and queues x_{11}, x_{22} are heavily loaded. The fairness properties can be improved by forcing the scheduler to serve a queue if it has not received service for certain time, even at the cost of optimality. The time interval can be tuned to attain the desired tradeoff between fairness and optimality. Another possibility is to proceed in the direction of [10].

The scheduling strategy needs shared scheduling states in an input queued switch, and similarly a centralized coordination for application in wireless networks. Further research is needed to develop distributed scheduling strategies, and we believe that developing a centralized policy is the first step in that direction.

A Proof for optimality of SOP for 2×2 switches with equal arrival rates

We outline the proof of Theorem 1 here. We deduce certain properties of the optimal discounted loss rate function $J_{\beta,\mathrm{OPT}}(\vec{x})$ for all $\vec{x}$ and $\beta < 1$ using iteration (13) (Lemmas 2 to 6). The optimal strategy follows from these properties using equations (9) to (12).

Let $\vec{ei}$ be a vector with a 1 in the ith component and 0s in all other components, i.e., $\vec{e1} = (0, 0, \ldots, 1, \ldots 0)$.

Lemma 2 *For each* $i \in \{1, \ldots, 4\}$, $J_{\beta,OPT}(\vec{x}) \leq J_{\beta,OPT}(\vec{x} + \vec{ei}) \leq J_{\beta,OPT}(\vec{x}) + 1$.

Lemma 3 *Let* $\vec{x1} = (x_{11}, x_{12} - 1, x_{21} - 1, x_{22})$, $\vec{x2} = (x_{11} - 1, x_{12}, x_{21}, x_{22} - 1)$,

$$
\begin{aligned}
J_{\beta,OPT}(\vec{x1}) &\leq J_{\beta,OPT}(\vec{x} - \vec{e1}) \qquad \textit{if } \min(x_{11}, x_{12}, x_{21}) > 0 & (14)\\
J_{\beta,OPT}(\vec{x1}) &\leq J_{\beta,OPT}(\vec{x} - \vec{e4}) \qquad \textit{if } \min(x_{12}, x_{21}, x_{22}) > 0 & (15)\\
J_{\beta,OPT}(\vec{x2}) &\leq J_{\beta,OPT}(\vec{x} - \vec{e3}) \qquad \textit{if } \min(x_{11}, x_{21}, x_{22}) > 0 & (16)\\
J_{\beta,OPT}(\vec{x2}) &\leq J_{\beta,OPT}(\vec{x} - \vec{e2}) \qquad \textit{if } \min(x_{11}, x_{12}, x_{22}) > 0 & (17)
\end{aligned}
$$

Lemmas 2 and 3 hold even when the input buffers have unequal sizes (i.e., $B_1 \neq B_2$) and the arrival rates are unequal We introduce a new notation $\mathrm{queue}_1(\vec{x}) = x_{11} + x_{12}$, $\mathrm{queue}_2(\vec{x}) = x_{21} + x_{22}$,

Lemma 4 *Let* $\lambda_{11} = \lambda_{12}$, $\lambda_{21} = \lambda_{22}$ *and* $B_1 = B_2 = B$. *Consider any two states* $\vec{x}$ *and* $\vec{y}$ *with equal input buffer occupancies, i.e.,* $\mathrm{queue}_1(\vec{x}) = \mathrm{queue}_1(\vec{y})$ *and* $\mathrm{queue}_2(\vec{x}) = \mathrm{queue}_2(\vec{y})$.

1. $J_{\beta,OPT}(\vec{x}) = J_{\beta,OPT}(\vec{y})$ *if* $\mathrm{diff}_i(\mathbf{x}) = \mathrm{diff}_i(\mathbf{y})$, $\forall i$,

2. $J_{\beta,OPT}(\vec{x}) = J_{\beta,OPT}(\vec{y})$ *if* $\mathrm{diff}_1(\vec{x}) = \mathrm{diff}_2(\vec{y})$, *and* $\mathrm{diff}_2(\vec{x}) = \mathrm{diff}_1(\vec{y})$

3. $J_{\beta,OPT}(\vec{x}) \leq J_{\beta,OPT}(\vec{y})$ *if* $\mathrm{diff}_1(\vec{x}) \geq \mathrm{diff}_2(\vec{x})$, $\mathrm{diff}_1(\vec{y}) = \mathrm{diff}_1(\vec{x}) + p$, *and* $\mathrm{diff}_2(\vec{y}) = \mathrm{diff}_2(\vec{x}) - p$, *or* $\mathrm{diff}_1(\vec{x}) \leq \mathrm{diff}_2(\vec{x})$, $\mathrm{diff}_1(\vec{y}) = \mathrm{diff}_1(\vec{x}) - p$, *and* $\mathrm{diff}_2(\vec{y}) = \mathrm{diff}_2(\vec{x}) + p$, $p \geq 0$.

Lemma 5 *Let* $\lambda_{11} = \lambda_{12}$, $\lambda_{21} = \lambda_{22}$ *and* $B_1 = B_2 = B$. $J_{\beta,OPT}(x_{11}, x_{12}, x_{21}, x_{22}) = J_{\beta,OPT}(x_{21}, x_{22}, x_{11}, x_{12})$.

Lemma 6 *Let* $\lambda_{ij} = \lambda$, *and* $B_1 = B_2 = B$. *Consider two states* $\vec{x}$ *and* $\vec{y}$ *such that either*

1. $\mathrm{queue}_1(\vec{x}) \geq \mathrm{queue}_2(\vec{x})$, $\mathrm{queue}_1(\vec{y}) = \mathrm{queue}_1(\vec{x}) + 1$, *and* $\mathrm{queue}_2(\vec{y}) = \mathrm{queue}_2(\vec{x}) - 1$, $\mathrm{diff}_i(\vec{x}) < \mathrm{diff}_i(\vec{y})$, $i = 1, 2$ *or*

2. $\mathrm{queue}_2(\vec{x}) \geq \mathrm{queue}_1(\vec{x})$, $\mathrm{queue}_2(\vec{y}) = \mathrm{queue}_2(\vec{x}) + 1$, *and* $\mathrm{queue}_1(\vec{y}) = \mathrm{queue}_1(\vec{x}) - 1$, $\mathrm{diff}_i(\vec{x}) > \mathrm{diff}_i(\vec{y})$, $i = 1, 2$

Then $J_{\beta,OPT}(\vec{x}) \leq J_{\beta,OPT}(\vec{y})$.

Proof of Theorem 1: Note that $\vec{\pi}_{\beta,\mathrm{OPT}}$ specified by equations (9) and (10) is the policy which minimizes the discounted loss rate ("optimal discounted policy"). We will show that SOP is $\vec{\pi}_{\beta,\mathrm{OPT}}$ for any discount factor $\beta < 1$ and hence SOP minimizes average loss from relations (11) and (12).

From (1), (10) and Lemma 2, the optimal discounted memory management policy drops packets or rejects new arrivals only when the corresponding input buffer is full, and no more than one packet is dropped/rejected.
From (9) and Lemmas 2 and 3, the optimal discounted policy schedules a pair of queues whenever possible.
From (9) and property (1) of Lemma 4, if all the queues have packets then either pair can be scheduled, i.e., the optimal discounted scheduling does not distinguish between pairs.
Property (2) and (3) of Lemma 4, and (10) give the optimal discounted packet replacement policy when an incoming packet finds the input buffer full.
From (9), and part (2) of Lemma 4, the optimal policy is to schedule any one nonempty queue if two queues at one input are zero and the two queues at the other input have equal length. From (9), and Lemma 5, the optimal policy is to schedule any one nonempty queue if the two queues for one output are zero and the two queues for the other output have equal length. From (9), and part (3) of Lemma 4 and Lemma 6, the optimal discounted scheduling schedules the larger queue, if the system can only schedule one queue and the nonempty queues have unequal lengths. Lemma 4(5) applies to the case when the two queues at one input (for one output) are empty.$\diamond$

B Proof for Theorems 2 and 3

Proof of Theorem 2: We will show that for all $\beta < 1$, $\vec{\pi}_{\beta,\mathrm{OPT}}$ specified by equations (9) and (10) satisfies the properties mentioned in this theorem. The results follow from relations (11) and (12). Lemmas 2 and 3 hold for any 2×2 switches and arbitrary arrival rates λ_{ij}. The same properties are shown for the optimal strategy in the symmetric traffic case (Theorem 1) using Lemmas 2 and 3.$\diamond$

Lemma 4 holds under the additional assumption $\lambda_{i1} = \lambda_{i2}$ $i = 1, 2$. The additional properties in Theorem 3 follow from this lemma. The argument is similar to the proof in Theorem 1.

C Proof for properties of the optimal strategy for a $N \times N$ switch

Lemma 2 extends for a $N \times N$ switch. It follows that the optimal memory management policy accepts packets without pushing out any existing packet as long as the input buffer has space.

We prove Lemma 1 now. We introduce some notations before proceeding to the proof. Let $\mathrm{cong}_{\mathrm{inp}}(\vec{x})$ be the maximum number of packets waiting at any input, i.e., $\mathrm{cong}_{\mathrm{inp}}(\vec{x}) = \max_i \sum_{j=1}^{N} x_{ij}$. Similarly, $\mathrm{cong}_{\mathrm{op}}(\vec{x})$ be the maximum number of packets waiting for any output, i.e., $\mathrm{cong}_{\mathrm{op}}(\vec{x}) = \max_j \sum_{i=1}^{N} x_{ij}$. Clearly, $\mathrm{cong}(\vec{x}) = \max\left(\mathrm{cong}_{\mathrm{inp}}(\vec{x}), \mathrm{cong}_{\mathrm{op}}(\vec{x})\right)$.

We will use induction and iteration (13). The method is as follows for all $\beta \in [\frac{N*B}{N*B+\min(\min_i \sum_{j=1}^{N} \lambda_{ij}, \min_j \sum_{i=1}^{N} \lambda_{ij})-\mu}, 1)$. We choose initial values $J_{\beta,0}(\vec{x})$ which satisfies the lemma, e.g., $J_{\beta,0}(\vec{x}) = \text{cong}(\vec{x})$. Now we assume that $J_{\beta,k}(\vec{x})$ satisfies Lemma 1 and show that $TJ_{\beta,k}(\vec{x})$ satisfies Lemma 1. From (13), $J_{\beta,k+1}(\vec{x})$ satisfies Lemma 1 as well. It follows by induction that $J_{\beta,k}(\vec{x})$ satisfies Lemma 1 for all k. Since $J_{\beta,\text{OPT}}(\vec{x}) = \lim_{k\to\infty} J_{\beta,k}(\vec{x})$, $J_{\beta,\text{OPT}}(\vec{x})$ satisfies Lemma 1 for all $\beta \in [\frac{N*B}{N*B+\min(\min_i \sum_{j=1}^N \lambda_{ij}, \min_j \sum_{i=1}^N \lambda_{ij})-\mu}, 1)$.

For simplicity we will refer to $J_{\beta,k}(\vec{x})$ as $J(\vec{x})$.

Proof: We show that $J(\vec{x}) \geq \text{cong}_{\text{inp}}(\vec{x})$ for all $\beta \in [\frac{\max_{\vec{x}} \text{cong}_{\text{inp}}(\vec{x})}{\max_{\vec{x}} \text{cong}_{\text{inp}}(\vec{x})+\min_i \sum_{j=1}^N \lambda_{ij}-\mu}, 1)$ and all states $\vec{x}$. It can be similarly shown that $J(\vec{x}) \geq \text{cong}_{\text{op}}(\vec{x})$ for all $\beta \in [\frac{\max_{\vec{x}} \text{cong}_{\text{op}}(\vec{x})}{\max_{\vec{x}} \text{cong}_{\text{op}}(\vec{x})+\min_j \sum_{i=1}^N \lambda_{ij}-\mu}, 1)$. We omit this part for brevity. The lemma follows since $\max_{\vec{x}} \text{cong}_{\text{op}}(\vec{x}) = N*B$ and $\max_{\vec{x}} \text{cong}_{\text{inp}}(\vec{x}) = B$.

We assume that $J(\vec{x}) \geq \text{cong}_{\text{inp}}(\vec{x})$ for all $\beta \in [\frac{\max_{\vec{x}} \text{cong}_{\text{inp}}(\vec{x})}{\max_{\vec{x}} \text{cong}_{\text{inp}}(\vec{x})+\min_i \sum_{j=1}^N \lambda_{ij}-\mu}, 1)$, and all states $\vec{x}$ and show the property for $TJ(\vec{x})$. For every state $\vec{y}$ in $S(\vec{x})$, $\text{cong}_{\text{inp}}(\vec{y}) \geq \text{cong}_{\text{inp}}(\vec{x}) - 1$. This happens because the scheduling can remove at most one packet from any input. Let the input i be the most congested, i.e., it has the maximum number of packets waiting for transmission ($i = \arg\max_k \sum_{j=1}^N x_{kj}$). For every state $\vec{y}$ in $\cup_{j=1}^N A(\vec{x}, i, j)$, $\text{cong}_{\text{inp}}(\vec{y}) + c(\vec{x}, \vec{y}, i, j) \geq \text{cong}_{\text{inp}}(\vec{x}) + 1$. This can be shown as follows.

$$\begin{aligned}
\text{cong}_{\text{inp}}(\vec{y}) + c(\vec{x},\vec{y},i,j) &= 1 + \text{cong}_{\text{inp}}(\vec{y}) + \sum_{j=1}^N x_{ij} - \sum_{j=1}^N y_{ij} + \sum_{\substack{1\leq k\leq N,\\ k\neq i}} \sum_{j=1}^N (x_{kj} - y_{kj}) \text{ (from (1))} \\
&\geq 1 + \sum_{j=1}^N x_{ij} \qquad (18) \\
&= 1 + \text{cong}_{\text{inp}}(\vec{x}) \text{ (since } i = \arg\max_k \sum_{j=1}^N x_{kj}) \qquad (19)
\end{aligned}$$

Inequality (18) follows since $\text{cong}_{\text{inp}}(\vec{y}) \geq \sum_{j=1}^N y_{ij}$ and $x_{kj} \geq y_{kj}$, for all $k \neq i$ and j. The latter holds since $\vec{y} \in A(\vec{x}, i, j)$ and $k \neq i$.
For any other state $\vec{y}$ in $A(\vec{x}, l, j)$, $i \neq l$, $\text{cong}_{\text{inp}}(\vec{y}) + c(\vec{x}, \vec{y}, i, j) \geq \text{cong}_{\text{inp}}(\vec{x})$. This can be shown similar to (19). The right hand side of (18) should be replaced by $\sum_{j=1}^N x_{ij}$. This inequality holds in this case as $\text{cong}_{\text{inp}}(\vec{y}) \geq \sum_{j=1}^N y_{ij}$ and $x_{kj} \geq y_{kj}$, for all k and j, barring possibly for $k = l$ and one value of j. In this case $y_{lj} = x_{lj} + 1$.

$$\begin{aligned}
TJ(\vec{x}) &= \sum_{k=1}^N \sum_{j=1}^N \min_{\vec{y}\in A(\vec{x},k,j)} \lambda_{kj} \left(c(\vec{x},\vec{y},k,j) + \beta J(\vec{y})\right) + \beta\mu \min_{\vec{y}\in S(\vec{x})} J(\vec{y})) \\
&\geq \beta \sum_{k=1}^N \sum_{j=1}^N \min_{\vec{y}\in A(\vec{x},k,j)} \lambda_{kj} \left(c(\vec{x},\vec{y},k,j) + \text{cong}_{\text{inp}}(\vec{y})\right) + \\
&\quad \beta\mu \min_{\vec{y}\in S(\vec{x})} \text{cong}_{\text{inp}}(\vec{y})) \text{ (by induction hypothesis)}
\end{aligned}$$

$$
\begin{aligned}
&\geq \beta \sum_{j=1}^{N} \min_{\vec{y} \in A(\vec{x},i,j)} \lambda_{ij} \left(c(\vec{x},\vec{y},i,j) + \text{cong}_{\text{inp}}(\vec{y}) \right) + \\
&\quad \beta \sum_{1 \leq k \leq N, k \neq i} \sum_{j=1}^{N} \min_{\vec{y} \in A(\vec{x},k,j)} \lambda_{kj} \left(c(\vec{x},\vec{y},k,j) + \text{cong}_{\text{inp}}(\vec{y}) \right) + \\
&\quad \beta\mu \left(\text{cong}_{\text{inp}}(\vec{x}) - 1 \right) \\
&\geq \beta \sum_{j=1}^{N} \lambda_{ij} \left(1 + \text{cong}_{\text{inp}}(\vec{x}) \right) + \beta \sum_{1 \leq k \leq N, k \neq i} \sum_{j=1}^{N} \lambda_{kj} \text{cong}_{\text{inp}}(\vec{x}) + \\
&\quad \beta\mu \left(\text{cong}_{\text{inp}}(\vec{x}) - 1 \right) \\
&= \beta \text{cong}_{\text{inp}}(\vec{x}) + \beta (\sum_{j=1}^{N} \lambda_{ij} - \mu) \ (\text{since } \mu + \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_{ij} = 1) \\
&\geq \text{cong}_{\text{inp}}(\vec{x}) \text{ since } \sum_{j=1}^{N} \lambda_{ij} > \mu \text{ and } \beta \geq \frac{\max_{\vec{x}} \text{cong}_{\text{inp}}(\vec{x})}{\max_{\vec{x}} \text{cong}_{\text{inp}}(\vec{x}) + \min_i \sum_{j=1}^{N} \lambda_{ij} - \mu}
\end{aligned}
$$

$\diamond$

D Proof of Lemmas 2 to 6

We prove Lemmas 2 to 6 in this section. These lemmas have been used for proving Theorems 1 to 3. We will use induction and iteration (13). The method is similar to that for Lemma 1. Note that $J_{\beta,0}(\vec{x}) = 0$ satisfies properties in Lemmas 2 to 6. Now we assume that $J_{\beta,k}(\vec{x})$ satisfies these properties and show that $TJ_{\beta,k}(\vec{x})$ and hence $J_{\beta,k+1}(\vec{x})$ satisfies these properties. It follows by induction that $J_{\beta,k}(\vec{x})$ satisfies these properties for all k. Since $J_{\beta,\text{OPT}}(\vec{x}) = \lim_{k\to\infty} J_{\beta,k}(\vec{x})$, $J_{\beta,\text{OPT}}(\vec{x})$ satisfies these properties for all $\beta < 1$.

For simplicity we will refer to $J_{\beta,k}(\vec{x})$ as $J(\vec{x})$. We would use one index to represent the components, e.g, $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ will represent $\lambda_{11}, \lambda_{12}, \lambda_{21}, \lambda_{22}$, and state $\vec{x} = (q_1, q_2, q_3, q_4)$. From (8) and the definitions of $S(\vec{x})$ and $A(\vec{x}, i, j)$, we obtain the following relations.

$$
TJ(q_1,q_2,q_3,q_4) = \mu\phi(q_1,q_2,q_3,q_4) + \sum_{i=1}^{4} \lambda_i \theta_i(q_1,q_2,q_3,q_4) \tag{20}
$$

$$
\begin{aligned}
\phi(q_1,q_2,q_3,q_4) &= \beta \min \Big(\min_i J\left(q_1,\ldots,(q_i-1)^+,\ldots q_4\right), J\left((q_1-1)^+,q_2,q_3,(q_4-1)^+\right), \\
&\quad J\left(q_1,(q_2-1)^+,(q_3-1)^+,q_4\right) \Big)
\end{aligned} \tag{21}
$$

$$
\begin{aligned}
\theta_i(q_1,q_2,q_3,q_4) &= \min_{d_1,d_2,d_3,d_4:(\ldots(q_j-d_j),\ldots,(q_i+1-d_i),\ldots)\text{feasible}} \Big(\sum_i d_i + \\
&\quad \beta J\left(\ldots(q_j-d_j),\ldots(q_i+1-d_i),\ldots\right) \Big)
\end{aligned} \tag{22}
$$

We normalize $\mu + \sum_i \lambda_i$ as 1. Also, $\beta < 1$.

Proof of Lemma 2: We will show that $J(\vec{x}) \leq J(\vec{x}+\vec{e1}) \leq J(\vec{x})+1$. The proof is similar for $\vec{ei}$, $i>1$. Let $\vec{x}=(q_1,q_2,q_3,q_4)$. We need to show that $J(q_1,q_2,q_3,q_4) \leq J(q_1+1,q_2,q_3,q_4) \leq J(q_1,q_2,q_3,q_4)+1$. We assume that this holds for $J(q_1,q_2,q_3,q_4)$ for all (q_1,q_2,q_3,q_4) and then show the result for $TJ(q_1,q_2,q_3,q_4)$ for all (q_1,q_2,q_3,q_4). It follows from this assumption and the fact that $\beta<1$, that

$$\begin{aligned}
\phi(q_1,q_2,q_3,q_4) &= \beta\min\left(J\left((q_1-1)^+,q_2,q_3,(q_4-1)^+\right),J\left(q_1,(q_2-1)^+,(q_3-1)^+,q_4\right)\right)\\
\theta_1(q_1,q_2,q_3,q_4) &= \beta J(q_1+1,q_2,q_3,q_4) \text{ if } q_1+q_2<B_1\\
\theta_1(q_1,q_2,q_3,q_4) &= 1+\beta\min\left(J(q_1,q_2,q_3,q_4),J(q_1+1,q_2-1,q_3,q_4)\right) \text{ if } q_1+q_2=B_1, q_2>0\\
\theta_1(B_1,0,q_3,q_4) &= 1+\beta J(B_1,0,q_3,q_4)
\end{aligned}$$

The functions $\theta_j(q_1,q_2,q_3,q_4)$ for $j>1$, can be constructed similarly.

Now we compare corresponding terms in $TJ(q_1,q_2,q_3,q_4)$ and $TJ(q_1+1,q_2,q_3,q_4)$.
Note that $\phi(q_1+1,q_2,q_3,q_4)=\beta\min\left(J\left(q_1,q_2,q_3,(q_4-1)^+\right),J\left(q_1+1,(q_2-1)^+,(q_3-1)^+,q_4\right)\right)$. Thus, by hypothesis,

$$\phi(q_1,q_2,q_3,q_4) \leq \phi(q_1+1,q_2,q_3,q_4) \leq \phi(q_1,q_2,q_3,q_4)+1 \tag{23}$$

By component wise comparison and hypothesis,
$\theta_j(q_1,q_2,q_3,q_4) \leq \theta_j(q_1+1,q_2,q_3,q_4) \leq \theta_j(q_1,q_2,q_3,q_4)+1$, $j>2$.

Since $q_1+q_2+1 \leq B_1$, $\theta_1(q_1,q_2,q_3,q_4)=\beta J(q_1+1,q_2,q_3,q_4)$, and $\theta_2(q_1,q_2,q_3,q_4)=\beta J(q_1,q_2+1,q_3,q_4)$. Now, we consider two cases separately, $q_1+q_2+1<B_1$, and $q_1+q_2+1=B_1$. In the former case, $\theta_1(q_1+1,q_2,q_3,q_4)=\beta J(q_1+2,q_2,q_3,q_4)$, and $\theta_2(q_1+1,q_2,q_3,q_4)=\beta J(q_1+1,q_2+1,q_3,q_4)$. Thus, by hypothesis, $\theta_j(q_1,q_2,q_3,q_4) \leq \theta_j(q_1+1,q_2,q_3,q_4) \leq \theta_j(q_1,q_2,q_3,q_4)+1$, $j \leq 2$.

Now consider the case $q_1+q_2+1=B_1$.

$$\begin{aligned}
\theta_1(q_1+1,q_2,q_3,q_4) &= 1+\beta\min\left(J(q_1+1,q_2,q_3,q_4),J(q_1+2,q_2-1,q_3,q_4)\right), \text{ if } q_2>0 &(24)\\
\theta_1(q_1+1,q_2,q_3,q_4) &= 1+\beta J(q_1+1,q_2,q_3,q_4), \text{ if } q_2=0 &(25)
\end{aligned}$$

Thus, $\theta_1(q_1+1,q_2,q_3,q_4) \leq 1+\theta_1(q_1,q_2,q_3,q_4)$.

$$\begin{aligned}
\theta_1(q_1,q_2,q_3,q_4) &= \beta J(q_1+1,q_2,q_3,q_4) &(26)\\
&\leq \beta J(q_1+1,q_2-1,q_3,q_4)+1\text{if } q_2>0, \text{ (by hypothesis)}\\
&\leq \beta J(q_1+2,q_2-1,q_3,q_4)+1 \text{ (by hypothesis)} &(27)
\end{aligned}$$

From (24), (26) and (27), $\theta_1(q_1,q_2,q_3,q_4) \leq \theta_1(q_1+1,q_2,q_3,q_4)$, if $q_2>0$. From (25) and (26), $\theta_1(q_1,q_2,q_3,q_4) \leq \theta_1(q_1+1,q_2,q_3,q_4)$, if $q_2=0$.

Next we compare $\theta_2(q_1+1,q_2,q_3,q_4)$ and $\theta_2(q_1,q_2,q_3,q_4)$ when $q_1+q_2+1=B_1$. Recall that $\theta_2(q_1,q_2,q_3,q_4)=\beta J(q_1,q_2+1,q_3,q_4)$, and

$$\theta_2(q_1+1,q_2,q_3,q_4)=1+\beta\min\left(J(q_1+1,q_2,q_3,q_4),J(q_1,q_2+1,q_3,q_4)\right) \text{ (note that } q_1+1>0) \tag{28}$$

Thus, $\theta_2(q_1+1,q_2,q_3,q_4) \leq \theta_2(q_1,q_2,q_3,q_4)+1$. Again,

$$\begin{aligned}
\theta_2(q_1,q_2,q_3,q_4) &= \beta J(q_1,q_2+1,q_3,q_4) &(29)\\
&\leq \beta J(q_1,q_2,q_3,q_4)+1 \text{ (by hypothesis)}\\
&\leq \beta J(q_1+1,q_2,q_3,q_4)+1 \text{ (by hypothesis)} &(30)
\end{aligned}$$

From (28) to (30), $\theta_2(q_1, q_2, q_3, q_4) \leq \theta_2(q_1 + 1, q_2, q_3, q_4)$. We have thus shown that

$$\theta_j(q_1, q_2, q_3, q_4) \leq \theta_j(q_1 + 1, q_2, q_3, q_4) \leq \theta_j(q_1, q_2, q_3, q_4) + 1 \forall j, \tag{31}$$

The result follows from (23) and (31) and the fact that $\mu + \sum_i \lambda_i = 1$. $\diamond$

From Lemma 2, equations (21) and (22)and the fact that $\beta < 1$, we have,

$$\begin{aligned}
\phi(q_1, q_2, q_3, q_4) &= \beta \min\left(J\left((q_1-1)^+, q_2, q_3, (q_4-1)^+\right), J\left(q_1, (q_2-1)^+, (q_3-1)^+, q_4\right)\right) && (32)\\
\theta_1(q_1, q_2, q_3, q_4) &= \beta J(q_1+1, q_2, q_3, q_4) \text{ if } q_1 + q_2 < B_1 && (33)\\
\theta_1(q_1, q_2, q_3, q_4) &= 1 + \beta \min\left(J(q_1, q_2, q_3, q_4), J(q_1+1, q_2-1, q_3, q_4)\right) \text{ if } q_1 + q_2 = B_1, q_2 > 0 && (34)\\
\theta_1(B_1, 0, q_3, q_4) &= 1 + \beta J(B_1, 0, q_3, q_4) && (35)
\end{aligned}$$

The functions $\theta_j(q_1, q_2, q_3, q_4)$ for $j > 1$, can be constructed similarly.

Proof of Lemma 3: Let $\vec{x} = (q_1, q_2, q_3, q_4)$. We will prove only the first inequality, i.e.,

$$J(q_1, q_2 - 1, q_3 - 1, q_4) \leq J(q_1 - 1, q_2, q_3, q_4) \text{ if } \min(q_1, q_2, q_3) > 0$$

The rest will follow similarly. We assume that this inequality holds for $J(q_1, q_2, q_3, q_4)$ for all (q_1, q_2, q_3, q_4) and then show the result for $TJ(q_1, q_2, q_3, q_4)$ for all (q_1, q_2, q_3, q_4).

Consider (q_1, q_2, q_3, q_4) such that $\min(q_1, q_2, q_3) > 0$. From (32) $\phi(q_1, q_2 - 1, q_3 - 1, q_4) \leq \beta J\left(q_1 - 1, q_2 - 1, q_3 - 1, (q_4 - 1)^+\right)$. Now, $\phi(q_1 - 1, q_2, q_3, q_4) = \beta \min\left(J\left(q_1 - 1, q_2 - 1, q_3 - 1, q_4\right), J\left((q_1 - 2)^+, q_2, q_3, (q_4 - 1)^+\right)\right)$. If $q_1 - 1 > 0$, then $J\left(q_1 - 1, q_2 - 1, q_3 - 1, (q_4 - 1)^+\right) \leq J\left((q_1 - 2)^+, q_2, q_3, (q_4 - 1)^+\right)$ by hypothesis. If $q_1 - 1 = 0$, then the inequality again holds from Lemma 2. Thus $J\left(q_1 - 1, q_2 - 1, q_3 - 1, (q_4 - 1)^+\right) \leq J\left((q_1 - 2)^+, q_2, q_3, (q_4 - 1)^+\right)$. Also, $J\left(q_1 - 1, q_2 - 1, q_3 - 1, (q_4 - 1)^+\right) \leq J\left(q_1 - 1, q_2 - 1, q_3 - 1, q_4\right)$ by Lemma 2. It follows that

$$\phi(q_1, q_2 - 1, q_3 - 1, q_4) \leq \phi(q_1 - 1, q_2, q_3, q_4) \tag{36}$$

We first show that $\theta_j(q_1, q_2 - 1, q_3 - 1, q_4) \leq \theta_j(q_1 - 1, q_2, q_3, q_4)$ for $j \leq 2$. We consider two cases separately: (a) $q_1 + q_2 - 1 < B_1$, and (b) $q_1 + q_2 - 1 = B_1$. In the first case, the inequalities hold from (33) and hypothesis (note that (33) is stated for θ_1 function only, but similar relations exist for θ_j functions, $j > 1$). Consider case (b) now.

$$\begin{aligned}
\theta_1(q_1, q_2 - 1, q_3 - 1, q_4) &\leq 1 + \beta J(q_1, q_2 - 1, q_3 - 1, q_4) \text{ (from (34))}\\
&\leq 1 + \beta \min\left(J(q_1, q_2 - 1, q_3, q_4), J(q_1 - 1, q_2, q_3, q_4)\right)\\
&\quad \text{(from hypothesis and Lemma 2)}\\
&= \theta_1(q_1 - 1, q_2, q_3, q_4) \text{ since } q_2 > 0 \text{ (from (34))} && (37)
\end{aligned}$$

$$\begin{aligned}
\theta_2(q_1, q_2 - 1, q_3 - 1, q_4) &= 1 + \beta \min\left(J(q_1, q_2 - 1, q_3 - 1, q_4), J(q_1 - 1, q_2, q_3 - 1, q_4)\right) && (38)\\
&\quad \text{since } q_1 > 0\\
&\leq 1 + \beta \min\left(J(q_1 - 1, q_2, q_3, q_4), J(q_1 - 2, q_2 + 1, q_3, q_4)\right)\\
&\quad \text{if } q_1 > 1 \text{ (by hypothesis)}\\
&= \theta_2(q_1 - 1, q_2, q_3, q_4) \text{ if } q_1 > 1, \text{ (from (34))} && (39)\\
\theta_2(q_1 - 1, q_2, q_3, q_4) &= 1 + \beta J(q_1 - 1, q_2, q_3, q_4) \text{ if } q_1 = 1 \text{ (note that } q_1 > 0)\\
&\geq \theta_2(q_1, q_2 - 1, q_3 - 1, q_4) \text{ if } q_1 = 1 \text{ (from (38) and Lemma 2)} && (40)
\end{aligned}$$

From (37),(39) and (40), $\theta_j(q_1, q_2-1, q_3-1, q_4) \le \theta_j(q_1-1, q_2, q_3, q_4)$ for $j \le 2$.

Now we will show that $\theta_j(q_1, q_2-1, q_3-1, q_4) \le \theta_j(q_1-1, q_2, q_3, q_4)$ for $j > 2$. We consider two cases separately: (a) $q_3+q_4 < B_2$, and (b) $q_3+q_4 = B_2$. In the first case, the inequality holds from (33) and hypothesis. Consider case (b) now.

$$\begin{aligned}
\theta_3(q_1, q_2-1, q_3-1, q_4) &= \beta J(q_1, q_2-1, q_3, q_4) \text{ (from (33))} && (41)\\
J(q_1, q_2-1, q_3, q_4) &\le 1 + J(q_1-1, q_2-1, q_3, q_4) \text{(from Lemma 2)} &&\\
&\le 1 + J(q_1-1, q_2, q_3, q_4) \text{(from Lemma 2)} && (42)\\
J(q_1, q_2-1, q_3, q_4) &\le 1 + J(q_1, q_2-1, q_3, q_4-1) \text{ if } q_4 > 0, \text{(from Lemma 2)} &&\\
&\le 1 + J(q_1-1, q_2, q_3+1, q_4-1) \text{(from hypothesis)} && (43)\\
\theta_3(q_1, q_2-1, q_3-1, q_4) &\le 1 + \beta \min\left(J(q_1-1, q_2, q_3, q_4), J(q_1-1, q_2, q_3+1, q_4-1)\right) &&\\
&\quad \text{if } q_4 > 0,\ \text{(from (41),(42),(43) and since } \beta < 1) && (44)\\
&= \theta_3(q_1-1, q_2, q_3, q_4) \text{ if } q_4 > 0,\ \text{(from (34))} &&\\
\theta_3(q_1, q_2-1, q_3-1, q_4) &\le 1 + \beta J(q_1-1, q_2, q_3, q_4) \text{ (from (41),(42) and since } \beta < 1) &&\\
&= \theta_3(q_1-1, q_2, q_3, q_4) \text{ if } q_4 = 0 \text{ (from (34))} && (45)
\end{aligned}$$

From (44) and (45), $\theta_j(q_1, q_2-1, q_3-1, q_4) \le \theta_j(q_1-1, q_2, q_3, q_4)$, for $j = 3$. We present the proof for $j = 4$ next.

$$\begin{aligned}
\theta_4(q_1, q_2-1, q_3-1, q_4) &= \beta J(q_1, q_2-1, q_3-1, q_4+1) \text{ (from (33))} && (46)\\
J(q_1, q_2-1, q_3-1, q_4+1) &\le 1 + J(q_1-1, q_2-1, q_3-1, q_4+1) \text{(from Lemma 2)} &&\\
&\le 1 + J(q_1-1, q_2, q_3-1, q_4+1) \text{(from Lemma 2)} && (47)\\
J(q_1, q_2-1, q_3-1, q_4+1) &\le 1 + J(q_1, q_2-1, q_3-1, q_4) \text{(from Lemma 2)} &&\\
&\le 1 + J(q_1-1, q_2, q_3, q_4) \text{(from hypothesis)} && (48)\\
\theta_4(q_1, q_2-1, q_3-1, q_4) &\le 1 + \beta \min\left(J(q_1-1, q_2, q_3-1, q_4+1), J(q_1-1, q_2, q_3, q_4)\right) &&\\
&\quad \text{(from (46),(47),(48) and since } \beta < 1) &&\\
&= \theta_4(q_1-1, q_2, q_3, q_4) \text{ since } q_3 > 0,\ \text{(from (34))} && (49)
\end{aligned}$$

From (49), $\theta_j(q_1, q_2-1, q_3-1, q_4) \le \theta_j(q_1-1, q_2, q_3, q_4)$ for $j = 4$.

$$\text{Thus } \theta_j(q_1, q_2-1, q_3-1, q_4) \le \theta_j(q_1-1, q_2, q_3, q_4), \forall j \qquad (50)$$

The result follows from (36) and (50). $\diamond$

From Lemmas 2, 3 and relation (32),

$$\begin{aligned}
\phi(q_1, q_2, q_3, q_4) &= \beta J(q_1-1, q_2, q_3, q_4-1) \text{ if } \min(q_1, q_4) > \min(q_2, q_3) = 0 && (51)\\
\phi(q_1, q_2, q_3, q_4) &= \beta J\left(q_1, q_2-1, q_3-1, q_4\right) \text{ if } \min(q_2, q_3) > \min(q_1, q_4) = 0 && (52)\\
\phi(q_1, q_2, q_3, q_4) &= \beta \min\left(J(q_1-1, q_2, q_3, q_4-1), J\left(q_1, q_2-1, q_3-1, q_4\right)\right), \text{ if } q_i > 0 \forall i && (53)\\
\phi(0, \ldots, q_i, \ldots, 0) &= \beta J(0, 0 \ldots, q_i - 1, 0, 0), \text{ if } q_i > 0 && (54)
\end{aligned}$$

Note that $\text{diff}_1(q_1, q_2, q_3, q_4) = q_1 - q_4$, $\text{diff}_2(q_1, q_2, q_3, q_4) = q_2 - q_3$, $\text{queue}_1(q_1, q_2, q_3, q_4) = q_1 + q_2$, $\text{queue}_2(q_1, q_2, q_3, q_4) = q_3 + q_4$, Consider two feasible states: $\mathbf{x}$ and $\mathbf{y}$ such that $\text{queue}_i(\mathbf{x}) =$

$\text{queue}_i(\mathbf{y})$, $i = 1, 2$, then $\sum_{i=1}^{2} \text{diff}_i(\mathbf{x}) = \sum_{i=1}^{2} \text{diff}_i(\mathbf{y})$. Thus if $\text{diff}_1(\mathbf{x}) = \text{diff}_1(\mathbf{y}) + p$, then $\text{diff}_2(\mathbf{x}) = \text{diff}_2(\mathbf{y}) - p$, for any p.

Proof of Lemma 4: We will first show that $J(q_1, q_2, q_3, q_4) = J(q_2, q_1, q_4, q_3)$ and refer to this property as part (0). Then we will show parts (1) to (3) using part (0). We will assume all these properties for the function $J(\mathbf{z})$ and prove these for the function $TJ(\mathbf{z})$. We assume that (a) $B_1 = B_2 = B$ and (b) $\lambda_1 = \lambda_2$ and $\lambda_3 = \lambda_4$.

Proof of part (0): We first prove property (0) for $TJ(q_1, q_2, q_3, q_4)$. Note that $\phi(q_1, q_2, q_3, q_4) = \phi(q_2, q_1, q_4, q_3)$, from part (0) of hypothesis and (32).

We will show that $\theta_1(q_1, q_2, q_3, q_4) = \theta_2(q_2, q_1, q_4, q_3)$, $\theta_2(q_1, q_2, q_3, q_4) = \theta_1(q_2, q_1, q_4, q_3)$, $\theta_3(q_1, q_2, q_3, q_4) = \theta_4(q_2, q_1, q_4, q_3)$, and $\theta_4(q_1, q_2, q_3, q_4) = \theta_1(q_2, q_1, q_4, q_3)$. The result follows since $\lambda_1 = \lambda_2$ and $\lambda_3 = \lambda_4$. We just show that $\theta_1(q_1, q_2, q_3, q_4) = \theta_2(q_2, q_1, q_4, q_3)$. The result follows similarly for others. First consider the case that $q_1 + q_2 < B$. From (33),

$$\begin{aligned}
\theta_1(q_1, q_2, q_3, q_4) &= \beta J(q_1 + 1, q_2, q_3, q_4) \\
&= \beta J(q_2, q_1 + 1, q_4, q_3) \text{ (by part (0) of hypothesis)} \\
&= \theta_2(q_2, q_1, q_4, q_3) \text{ (from (33))}
\end{aligned}$$

Now consider the case that $q_1 + q_2 = B$, and $q_2 > 0$.

$$\begin{aligned}
\theta_1(q_1, q_2, q_3, q_4) &= 1 + \beta \min\left(J(q_1 + 1, q_2 - 1, q_3, q_4), J(q_1, q_2, q_3, q_4)\right) \text{ (from (34))} \\
&= 1 + \beta \min\left(J(q_2 - 1, q_1 + 1, q_4, q_3), J(q_2, q_1, q_4, q_3)\right) \text{ (by part (0) of hypothesis)} \\
&= \theta_2(q_2, q_1, q_4, q_3) \text{ (from (34))}
\end{aligned}$$

Now consider the case that $q_1 + q_2 = B$, and $q_2 = 0$. Thus $q_1 = B$.

$$\begin{aligned}
\theta_1(B, 0, q_3, q_4) &= 1 + \beta J(B, 0, q_3, q_4) \text{ (from (35))} \\
&= 1 + \beta J(0, B, q_4, q_3) \text{ (from part (0) of hypothesis)} \\
&= \theta_2(0, B, q_4, q_3) \text{ (from (34))}
\end{aligned}$$

The result follows.

Proof of part (1): We consider two states $\mathbf{x}$ and $\mathbf{y}$ such that $\text{diff}_i(\mathbf{x}) = \text{diff}_i(\mathbf{y})$, and $\text{queue}_i(\mathbf{x}) = \text{queue}_i(\mathbf{y})$, $\forall i$. We first show that if either $\max\left(\min(x_1, x_4), \min(x_2, x_3)\right) = 0$ or $\max\left(\min(y_1, y_4), \min(y_2, y_3)\right) = 0$, then $\mathbf{x} = \mathbf{y}$, and hence the result holds trivially. Consider the case that $\max\left(\min(x_1, x_4), \min(x_2, x_3)\right) = 0$. The result holds similarly if $\max\left(\min(y_1, y_4), \min(y_2, y_3)\right) = 0$. One of the following cases must happen: (a)$x_1 = x_2 = 0$ (b)$x_3 = x_4 = 0$(c)$x_2 = x_4 = 0$ and (d)$x_1 = x_3 = 0$. For (a), $y_1 + y_2 = x_1 + x_2 = 0$. Thus $y_1 = y_2 = 0$. Also, $x_1 - x_4 = y_1 - y_4$. Thus $x_4 = y_4$. Finally, $x_2 - x_3 = y_2 - y_3$. Thus $x_3 = y_3$. Thus $\mathbf{x} = \mathbf{y}$. The result follows similarly for (b). Consider (c). Note that $x_1 = y_1 - y_4$ and $x_1 = y_1 + y_2$. It follows that $y_2 = y_4 = 0$. Thus $x_1 = y_1$ and $x_3 = y_3$. Hence, $\mathbf{x} = \mathbf{y}$. The result follows similarly for (d).

Thus we assume that $\max(\min(x_1, x_4), \min(x_2, x_3)) > 0$ and $\max(\min(y_1, y_4), \min(y_2, y_3)) > 0$. Let $\mathbf{x1} = (x_1 - 1, x_2, x_3, x_4 - 1)$, $\mathbf{x2} = (x_1, x_2 - 1, x_3 - 1, x_4)$, $\mathbf{y1} = (y_1 - 1, y_2, y_3, y_4 - 1)$ and $\mathbf{y2} = (y_1, y_2 - 1, y_3 - 1, y_4)$. Note that $\text{diff}_i(\mathbf{x1}) = \text{diff}_i(\mathbf{x2}) = \text{diff}_i(\mathbf{x}) = \text{diff}_i(\mathbf{y}) = \text{diff}_i(\mathbf{y1}) = \text{diff}_i(\mathbf{y2})$ $i = 1, 2$. Also, $\text{queue}_i(\mathbf{x1}) = \text{queue}_i(\mathbf{x2}) = \text{queue}_i(\mathbf{x}) - 1 = \text{queue}_i(\mathbf{y}) - 1 = \text{queue}_i(\mathbf{y1}) = \text{queue}_i(\mathbf{y2})$ $i = 1, 2$. From part (1) of hypothesis

$$J(\mathbf{x1}) = J(\mathbf{x2}) = J(\mathbf{y1}) = J(\mathbf{y2}) \tag{55}$$

(All of the states $\mathbf{x1}$, $\mathbf{x2}$, $\mathbf{y1}$, $\mathbf{y2}$ need not be feasible. The equality applies to the feasible ones only.)

From the facts that $\max(\min(x_1, x_4), \min(x_2, x_3)) > 0$, and (51) to (53), $\max(\min(y_1, y_4), \min(y_2, y_3)) > 0$,

$$\phi(\mathbf{x}) = \begin{cases} \beta J(\mathbf{x1}) & \mathbf{x1} \text{ feasible}, \quad \text{if } \min(x_2, x_3) = 0, \\ \beta J(\mathbf{x2}), & \mathbf{x2} \text{ feasible} \quad \text{if } \min(x_1, x_4) = 0 \\ \beta \min(J(\mathbf{x1}), J(\mathbf{x2})) & \mathbf{x1}, \mathbf{x2} \text{ feasible}, \quad \text{otherwise} \end{cases} \tag{56}$$

$$\phi(\mathbf{y}) = \begin{cases} \beta J(\mathbf{y1}) & \mathbf{y1} \text{ feasible}, \quad \text{if } \min(y_2, y_3) = 0, \\ \beta J(\mathbf{y2}), & \mathbf{y2} \text{ feasible} \quad \text{if } \min(y_1, y_4) = 0 \\ \beta \min(J(\mathbf{y1}), J(\mathbf{y2})) & \mathbf{y1}, \mathbf{y2} \text{ feasible}, \quad \text{otherwise} \end{cases} \tag{57}$$

$$\text{From (55) to (57), } \phi(\mathbf{x}) = \phi(\mathbf{y}) \tag{58}$$

Next, we will show that $\theta_i(\mathbf{x}) = \theta_i(\mathbf{y}), \forall i$. We present the arguments for $i = 1$ here, and the rest of the cases will follow similarly. We consider two cases separately: (a)$\text{queue}_1(\mathbf{x}) < B$, and (b)$\text{queue}_1(\mathbf{x}) = B$. In case (a), $\text{queue}_1(\mathbf{y}) < B$. Let $\mathbf{x1} = (x_1 + 1, x_2, x_3, x_4)$, and $\mathbf{y1} = (y_1 + 1, y_2, y_3, y_4)$. From (33) $\theta_1(\mathbf{x}) = \beta J(\mathbf{x1})$ and $\theta_1(\mathbf{y}) = \beta J(\mathbf{y1})$. Note that $\text{diff}_i(\mathbf{x1}) = \text{diff}_i(\mathbf{y1})$ and $\text{queue}_i(\mathbf{x1}) = \text{queue}_i(\mathbf{y1})$ $i = 1, 2$. From part (1) of hypothesis, $J(\mathbf{x1}) = J(\mathbf{y1})$. Thus $\theta_1(\mathbf{x}) = \theta_1(\mathbf{y})$.

Now consider case (b), i.e., $\text{queue}_1(\mathbf{x}) = B$. It follows that $\text{queue}_1(\mathbf{y}) = B$. We consider four subcases: (i)$x_2 = y_2 = 0$, (ii) $x_2 = 0$, $y_2 > 0$ (iii) $x_2 > 0$, $y_2 = 0$, and (iv) $x_2 > 0$, $y_2 > 0$. In the first sub-case, from (35) $\theta_1(\mathbf{x}) = 1 + \beta J(\mathbf{x})$ and $\theta_1(\mathbf{y}) = 1 + \beta J(\mathbf{y})$. ¿From part (1) of hypothesis, $\theta_1(\mathbf{x}) = \theta_1(\mathbf{y})$. For the other three subcases, let $\mathbf{x1} = (x_1 + 1, x_2 - 1, x_3, x_4)$ and $\mathbf{y1} = (y_1 + 1, y_2 - 1, y_3, y_4)$. Note that $\text{diff}_i(\mathbf{x1}) = \text{diff}_i(\mathbf{y1})$ and $\text{queue}_i(\mathbf{x1}) = \text{queue}_i(\mathbf{y1}) = \text{queue}_i(\mathbf{x}) = \text{queue}_i(\mathbf{y})$, $i = 1, 2$. Consider subcase (ii). Here, $x_1 = B$. From (35) $\theta_1(\mathbf{x}) = 1 + \beta J(\mathbf{x})$ and $\theta_1(\mathbf{y}) = 1 + \beta \min(J(\mathbf{y}), J(\mathbf{y1}))$. We will show that $J(\mathbf{y}) \leq J(\mathbf{y1})$, and thus $\theta_1(\mathbf{y}) = 1 + \beta J(\mathbf{y})$. Hence, from part (1) of hypothesis, $\theta_1(\mathbf{x}) = \theta_1(\mathbf{y})$. Now we show that $J(\mathbf{y}) \leq J(\mathbf{y1})$. Note that $\text{diff}_2(\mathbf{y}) = \text{diff}_2(\mathbf{x}) = -x_3 \leq 0 \leq B - x_4 = x_1 - x_4 = \text{diff}_1(\mathbf{x}) = \text{diff}_1(\mathbf{y})$ (recall $x_4 \leq B$ and thus $0 \leq B - x_4$). Thus, $\text{diff}_2(\mathbf{y}) \leq \text{diff}_1(\mathbf{y})$. Also, $\text{diff}_2(\mathbf{y1}) = \text{diff}_2(\mathbf{y}) - 1$ and $\text{diff}_1(\mathbf{y1}) = \text{diff}_1(\mathbf{y}) + 1$. Also $\text{queue}_i(\mathbf{y1}) = \text{queue}_i(\mathbf{y})$, $i = 1, 2$. From part (3) of hypothesis $J(\mathbf{y}) \leq J(\mathbf{y1})$. The argument is similar for subcase (iii). For subcase (iv), $\theta_1(\mathbf{x}) = 1 + \beta \min(J(\mathbf{x}), J(\mathbf{x1}))$ and $\theta_1(\mathbf{y}) = 1 + \beta \min(J(\mathbf{y}), J(\mathbf{y1}))$. From part (1) of hypothesis $J(\mathbf{x}) = J(\mathbf{y})$ and $J(\mathbf{x1}) = J(\mathbf{y1})$. Thus, $\theta_1(\mathbf{x}) = \theta_1(\mathbf{y})$.

$$\text{Thus } \theta_i(\mathbf{x}) = \theta_i(\mathbf{y}), \forall i \tag{59}$$

From (58) and (59), $TJ(\mathbf{x}) = TJ(\mathbf{y})$. Part (1) follows.

Proof of part (2): We consider two states $\mathbf{x}$ and $\mathbf{y}$ such that $\text{queue}_i(\mathbf{x}) = \text{queue}_i(\mathbf{y})$, $\forall i$, and $\text{diff}_1(\mathbf{x}) = \text{diff}_2(\mathbf{y})$, and $\text{diff}_2(\mathbf{x}) = \text{diff}_1(\mathbf{y})$. Consider state $\mathbf{x1} = (x_2, x_1, x_4, x_3)$.

$$\text{From part (0) } TJ(\mathbf{x}) = TJ(\mathbf{x1}) \tag{60}$$

Note that $\text{queue}_i(\mathbf{x}) = \text{queue}_i(\mathbf{x1})$, $\forall i$, and $\text{diff}_1(\mathbf{x}) = \text{diff}_2(\mathbf{x1})$, and $\text{diff}_2(\mathbf{x}) = \text{diff}_1(\mathbf{x1})$. It follows that $\text{diff}_i(\mathbf{x1}) = \text{diff}_i(\mathbf{y})$ and $\text{queue}_i(\mathbf{x1}) = \text{queue}_i(\mathbf{y})$, $i = 1, 2$.

$$\text{From part (1) } TJ(\mathbf{x1}) = TJ(\mathbf{y}) \tag{61}$$

From (60) and (61), $TJ(\mathbf{x}) = TJ(\mathbf{y})$. Part (2) follows.

Proof of part (3): We consider two states $\mathbf{x}$ and $\mathbf{y}$ such that $\text{queue}_i(\mathbf{x}) = \text{queue}_i(\mathbf{y})$, $\forall i$. Also, if $\text{diff}_1(\mathbf{x}) \geq \text{diff}_2(\mathbf{x})$, then $\text{diff}_1(\mathbf{y}) = \text{diff}_1(\mathbf{x}) + p$, and $\text{diff}_2(\mathbf{y}) = \text{diff}_2(\mathbf{x}) - p$, else if $\text{diff}_1(\mathbf{x}) \leq \text{diff}_2(\mathbf{x})$, then $\text{diff}_1(\mathbf{y}) = \text{diff}_1(\mathbf{x}) - p$, and $\text{diff}_2(\mathbf{y}) = \text{diff}_2(\mathbf{x}) + p$, $p \geq 0$. We will show that $TJ(\mathbf{x}) \leq TJ(\mathbf{y})$. We show the result for the case where $\text{diff}_1(\mathbf{x}) \geq \text{diff}_2(\mathbf{x})$, $\text{diff}_1(\mathbf{y}) = \text{diff}_1(\mathbf{x}) + p$, and $\text{diff}_2(\mathbf{y}) = \text{diff}_2(\mathbf{x}) - p$, $p \geq 0$. The result follows similarly in the other case. If $p = 0$, then the result follows from part (1). So let $p > 0$.

We first show that $\phi(\mathbf{x}) \leq \phi(\mathbf{y})$. We need to consider four subcases separately:
(a) $\max(\min(x_1, x_4), \min(x_2, x_3)) > 0$ and $\max(\min(y_1, y_4), \min(y_2, y_3)) > 0$,
(b) $\max(\min(x_1, x_4), \min(x_2, x_3)) = 0$ and $\max(\min(y_1, y_4), \min(y_2, y_3)) > 0$,
(c) $\max(\min(x_1, x_4), \min(x_2, x_3)) > 0$ and $\max(\min(y_1, y_4), \min(y_2, y_3)) = 0$, and
(d)$\max(\min(x_1, x_4), \min(x_2, x_3)) = 0$ and $\max(\min(y_1, y_4), \min(y_2, y_3)) = 0$.

First consider subcase (a). Let $\mathbf{x1} = (x_1 - 1, x_2, x_3, x_4 - 1)$, $\mathbf{x2} = (x_1, x_2 - 1, x_3 - 1, x_4)$, $\mathbf{y1} = (y_1 - 1, y_2, y_3, y_4 - 1)$ and $\mathbf{y2} = (y_1, y_2 - 1, y_3 - 1, y_4)$. Note that $\text{diff}_i(\mathbf{x1}) = \text{diff}_i(\mathbf{x2}) = \text{diff}_i(\mathbf{x})$ and $\text{diff}_i(\mathbf{y}) = \text{diff}_i(\mathbf{y1}) = \text{diff}_i(\mathbf{y2})$ $i = 1, 2$. Thus $\text{diff}_1(\mathbf{xj}) \geq \text{diff}_2(\mathbf{xj})$, $\text{diff}_1(\mathbf{yk}) = \text{diff}_1(\mathbf{xj}) + p$, and $\text{diff}_2(\mathbf{yk}) = \text{diff}_2(\mathbf{xj}) - p$, $p > 0$, $j, k \in \{1, 2\}$. Also, $\text{queue}_i(\mathbf{x1}) = \text{queue}_i(\mathbf{x2}) = \text{queue}_i(\mathbf{x}) - 1 = \text{queue}_i(\mathbf{y}) - 1 = \text{queue}_i(\mathbf{y1}) = \text{queue}_i(\mathbf{y2})$ $i = 1, 2$. From part (3) of hypothesis

$$J(\mathbf{xj}) \leq J(\mathbf{yk}), j, k \in \{1, 2\}, \tag{62}$$

(All states $\mathbf{x1}$, $\mathbf{x2}$, $\mathbf{y1}$, $\mathbf{y2}$ need not be feasible. The inequality applies to the feasible ones only.)

From (51) to (53) and the fact that $\max(\min(x_1, x_4), \min(x_2, x_3)) > 0$, $\max(\min(y_1, y_4), \min(y_2, y_3)) > 0$,

$$\phi(\mathbf{x}) = \begin{cases} \beta J(\mathbf{x1}) & \mathbf{x1} \text{ feasible}, \quad \text{if } \min(x_2, x_3) = 0, \\ \beta J(\mathbf{x2}), & \mathbf{x2} \text{ feasible} \quad \text{if } \min(x_1, x_4) = 0 \\ \beta \min(J(\mathbf{x1}), J(\mathbf{x2})) & \mathbf{x1}, \mathbf{x2} \text{ feasible}, \quad \text{otherwise} \end{cases} \tag{63}$$

$$\phi(\mathbf{y}) = \begin{cases} \beta J(\mathbf{y1}) & \mathbf{y1} \text{ feasible}, \quad \text{if } \min(y_2, y_3) = 0, \\ \beta J(\mathbf{y2}), & \mathbf{y2} \text{ feasible} \quad \text{if } \min(y_1, y_4) = 0 \\ \beta \min(J(\mathbf{y1}), J(\mathbf{y2})) & \mathbf{y1}, \mathbf{y2} \text{ feasible}, \quad \text{otherwise} \end{cases} \tag{64}$$

It follows from (62) to (64) that $\phi(\mathbf{x}) \leq \phi(\mathbf{y})$.

We will use the following results for the other three subcases:

$$\max(x_1, x_3) > 0 \tag{65}$$
$$\max(x_2, x_4) > 0 \tag{66}$$
$$\max(y_1, y_3) > 0 \tag{67}$$

We prove these as follows. First let $x_1 = x_3 = 0$. Since $\text{diff}_2(\mathbf{x}) \leq \text{diff}_1(\mathbf{x})$, $x_2 \leq -x_4$. Thus $x_2 = x_4 = 0$. Thus $\text{queue}_i(\mathbf{x}) = 0$, $i = 1, 2$. It follows that $\text{queue}_i(\mathbf{y}) = 0$, $i = 1, 2$. Thus $y_i = 0$, for all i. Thus $\text{diff}_1(\mathbf{y}) = 0 = \text{diff}_1(\mathbf{x})$ which contradicts the assumption that $p > 0$. Next, assume that $x_2 = x_4 = 0$. Now, $\text{diff}_1(\mathbf{y}) = \text{diff}_1(\mathbf{x}) + p$, implies that $y_1 - y_4 = x_1 + p$. Thus $y_1 > x_1$, since $p > 0$. Thus $\text{queue}_1(\mathbf{y}) \geq y_1 > x_1 = \text{queue}_1(\mathbf{x})$, which contradicts the assumption that $\text{queue}_i(\mathbf{y}) = \text{queue}_i(\mathbf{x})$ (note that $\text{queue}_1(\mathbf{x}) = x_1$ since $x_2 = 0$). Finally let $y_1 = y_3 = 0$. Since $\text{diff}_2(\mathbf{y}) \leq \text{diff}_1(\mathbf{y})$, $y_2 \leq -y_4$. Thus $y_2 = y_4 = 0$. Thus $\text{queue}_i(\mathbf{y}) = 0$, $i = 1, 2$. It follows that $\text{queue}_i(\mathbf{x}) = 0$, $i = 1, 2$. Thus $x_i = 0$, for all i. Thus $\text{diff}_1(\mathbf{y}) = 0 = \text{diff}_1(\mathbf{x})$ which contradicts the assumption that $p > 0$.

Now consider subcase (b). We will show that subcase (b) does not occur. Note that from (65),(66) and the fact that $\max(\min(x_1, x_4), \min(x_2, x_3)) = 0$ it may be construed that one of the following two cases must happen: (i)$x_1 = x_2 = 0$(ii)$x_3 = x_4 = 0$. In the first case, $\text{queue}_1(\mathbf{x}) = 0$. It follows that $\text{queue}_1(\mathbf{y}) = 0$. Thus $y_1 = y_2 = 0$. Thus $\max(\min(y_1, y_4), \min(y_2, y_3)) = 0$, which contradicts the assumption. Now let $x_3 = x_4 = 0$. It follows that $\text{queue}_2(\mathbf{y}) = 0$. Thus $y_3 = y_4 = 0$. Thus $\max(\min(y_1, y_4), \min(y_2, y_3)) = 0$, which contradicts the assumption. Thus subcase (b) is ruled out.

Now consider subcase (c). From (67) and the fact that $\max(\min(y_1, y_4), \min(y_2, y_3)) = 0$ it may be construed that one of the following three cases must happen: (i)$y_1 = y_2 = 0$(ii)$y_3 = y_4 = 0$, (iii)$y_2 = y_4 = 0$. Arguing as for subcase (b), the first two cases imply that $\max(\min(x_1, x_4), \min(x_2, x_3)) = 0$, and thus these cases are ruled out. Thus $\mathbf{y} = (y_1, 0, y_3, 0)$, and furthermore $y_1 > 0$ and $y_2 > 0$. Otherwise, $\text{queue}_i(\mathbf{y}) = 0$ for some i and thus $\text{queue}_i(\mathbf{x}) = 0$ for some i. It would follow that either $x_1 = x_2 = 0$ or $x_3 = x_4 = 0$ which means $\max(\min(x_1, x_4), \min(x_2, x_3)) = 0$, and this contradicts the assumption for this subcase. Let $\mathbf{x1} = (x_1 - 1, x_2, x_3, x_4 - 1)$, $\mathbf{x2} = (x_1, x_2 - 1, x_3 - 1, x_4)$ and $\mathbf{y1} = (y_1 - 1, 0, y_3 - 1, 0)$. Note that $\text{diff}_1(\mathbf{y1}) = \text{diff}_1(\mathbf{y}) - 1$, and $\text{diff}_2(\mathbf{y1}) = \text{diff}_2(\mathbf{y}) + 1$. It follows that $\text{diff}_1(\mathbf{y1}) = \text{diff}_1(\mathbf{x}) + p - 1$, $\text{diff}_2(\mathbf{y1}) = \text{diff}_2(\mathbf{x}) - (p - 1)$. Also, $\text{diff}_i(\mathbf{x1}) = \text{diff}_i(\mathbf{x2}) = \text{diff}_i(\mathbf{x})$ $i = 1, 2$. It follows that $\text{diff}_1(\mathbf{y1}) = \text{diff}_1(\mathbf{xj}) + p - 1$, $\text{diff}_2(\mathbf{y1}) = \text{diff}_2(\mathbf{xj}) - (p-1)$, and $\text{diff}_1(\mathbf{xj}) \geq \text{diff}_2(\mathbf{xj})$, $j = 1, 2$. Also, $\text{queue}_i(\mathbf{x1}) = \text{queue}_i(\mathbf{x2}) = \text{queue}_i(\mathbf{x}) - 1 = \text{queue}_i(\mathbf{y}) - 1 = \text{queue}_i(\mathbf{y1})$ $i = 1, 2$. Since $p > 1$, from part (3) of hypothesis,

$$J(\mathbf{xj}) \leq J(\mathbf{y1}), j \in \{1, 2\}, \tag{68}$$

Both states $\mathbf{x1}$, $\mathbf{x2}$ need not be feasible. The inequality applies to the feasible ones only. From (32)

$$\begin{aligned} \phi(\mathbf{y}) &= \beta \min(J(y_1 - 1, 0, y_3, 0), J(y_1, 0, y_3 - 1, 0)) \\ &\geq \beta J(\mathbf{y1}) \text{ (from Lemma 2)} \end{aligned} \tag{69}$$

Since $\max(\min(x_1, x_4), \min(x_2, x_3)) > 0$ from (51) to (53)

$$\phi(\mathbf{x}) = \begin{cases} \beta J(\mathbf{x1}) & \mathbf{x1} \text{ feasible,} & \text{if } \min(x_2, x_3) = 0, \\ \beta J(\mathbf{x2}), & \mathbf{x2} \text{ feasible} & \text{if } \min(x_1, x_4) = 0 \\ \beta \min(J(\mathbf{x1}), J(\mathbf{x2})) & \mathbf{x1}, \mathbf{x2} \text{ feasible,} & \text{otherwise} \end{cases} \tag{70}$$

It follows from (68) to (70) that $\phi(\mathbf{x}) \leq \phi(\mathbf{y})$.

Now consider subcase (d). From (65),(66) and the fact that $\max(\min(x_1, x_4), \min(x_2, x_3)) = 0$ it may be construed that one of the following two cases must happen: (i)$x_3 = x_4 = 0$(ii)$x_1 = x_2 = 0$. We argue the result for case(i) only. The result can be argued similarly for case (ii). Here, $\text{queue}_2(\mathbf{x}) = 0$. It follows that $\text{queue}_2(\mathbf{y}) = 0$. Thus $y_3 = y_4 = 0$. Thus $\text{diff}_1(\mathbf{x}) = x_1$, $\text{diff}_2(\mathbf{x}) = x_2$, $\text{diff}_1(\mathbf{y}) = y_1$ and $\text{diff}_2(\mathbf{y}) = y_2$. It follows that $y_1 = x_1 + p$, $x_1 \geq x_2$, $x_2 = y_2 + p$. Let $\mathbf{x1} = (x_1 - 1, x_2, 0, 0)$, $\mathbf{x2} = (x_1, x_2 - 1, 0, 0)$, $\mathbf{y1} = (y_1 - 1, y_2, 0, 0)$ and $\mathbf{y2} = (y_1, y_2 - 1, 0, 0)$. Note that the states $\mathbf{x1}$, $\mathbf{x2}$, $\mathbf{y1}$ are feasible since $y_1 > x_1 \geq x_2 > y_2 \geq 0$. However, state $\mathbf{y2}$ need not be feasible. Here $\text{diff}_1(\mathbf{x2}) = x_1$, $\text{diff}_2(\mathbf{x2}) = x_2 - 1$, $\text{diff}_1(\mathbf{y1}) = y_1 - 1$, $\text{diff}_2(\mathbf{y1}) = y_2$, $\text{diff}_1(\mathbf{y2}) = y_1$, and $\text{diff}_2(\mathbf{y2}) = y_2 - 1$. Thus $\text{diff}_1(\mathbf{y1}) = \text{diff}_1(\mathbf{x2}) + p - 1$, $\text{diff}_2(\mathbf{y1}) = \text{diff}_2(\mathbf{x2}) - (p-1)$, $\text{diff}_1(\mathbf{y2}) = \text{diff}_1(\mathbf{x2}) + p$, $\text{diff}_2(\mathbf{y2}) = \text{diff}_2(\mathbf{x2}) - p$, $\text{diff}_1(\mathbf{x2}) \geq \text{diff}_2(\mathbf{x2})$. Also, $\text{queue}_1(\mathbf{x2}) = \text{queue}_1(\mathbf{x}) - 1 = \text{queue}_1(\mathbf{y}) - 1 = \text{queue}_1(\mathbf{yk})$ $k = 1, 2$, and $\text{queue}_2(\mathbf{x2}) = \text{queue}_2(\mathbf{yk}) = 0$, $k = 1, 2$. From part (3) of hypothesis

$$J(\mathbf{x2}) \leq J(\mathbf{yk}), k \in \{1, 2\}, \tag{71}$$

(The inequality applies to state $\mathbf{y2}$ only if it is feasible.)

$$\text{From (32), } \phi(\mathbf{y}) = \begin{cases} \beta J(\mathbf{y1}) & \text{if } y_2 = 0, \\ \beta \min\left(J(\mathbf{y1}), J(\mathbf{y2})\right) \quad \mathbf{y2} \text{ feasible}, & y_2 > 0 \end{cases} \tag{72}$$

$$\begin{aligned} \text{From (32), } \phi(\mathbf{x}) &= \beta \min\left(J(\mathbf{x1}), J(\mathbf{x2})\right) \\ &\leq \beta J(\mathbf{x2}) \end{aligned} \tag{73}$$

It follows from (71) to (73) that $\phi(\mathbf{x}) \leq \phi(\mathbf{y})$. From the valid subcases (a), (c) and (d),

$$\phi(\mathbf{x}) \leq \phi(\mathbf{y}) \tag{74}$$

Now, we will show that $\theta_i(\mathbf{x}) \leq \theta_i(\mathbf{y}), \forall i$. We present the arguments for $i = 1, 2$ here, and the rest of the cases will follow similarly. First consider $i = 1$. We consider two cases separately: (i)$\text{queue}_1(\mathbf{x}) < B$, and (ii)$\text{queue}_1(\mathbf{x}) = B$. Under (i),$\text{queue}_1(\mathbf{y}) < B$. Let $\mathbf{x1} = (x_1 + 1, x_2, x_3, x_4)$, and $\mathbf{y1} = (y_1 + 1, y_2, y_3, y_4)$. From (33) $\theta_1(\mathbf{x}) = \beta J(\mathbf{x1})$ and $\theta_1(\mathbf{y}) = \beta J(\mathbf{y1})$. Note that $\text{diff}_1(\mathbf{x1}) = \text{diff}_1(\mathbf{x}) + 1$, $\text{diff}_2(\mathbf{x1}) = \text{diff}_2(\mathbf{x})$, $\text{diff}_1(\mathbf{y1}) = \text{diff}_1(\mathbf{y}) + 1$ and $\text{diff}_2(\mathbf{y1}) = \text{diff}_2(\mathbf{y})$. It follows that $\text{diff}_1(\mathbf{y1}) = \text{diff}_1(\mathbf{x1}) + p$, $\text{diff}_2(\mathbf{y1}) = \text{diff}_2(\mathbf{x1}) - p$, and $\text{diff}_1(\mathbf{x1}) > \text{diff}_2(\mathbf{x1})$. Also, $\text{queue}_i(\mathbf{x1}) = \text{queue}_i(\mathbf{y1})$ $i = 1, 2$. From part (3) of hypothesis, $J(\mathbf{x1}) \leq J(\mathbf{y1})$. Thus $\theta_1(\mathbf{x}) \leq \theta_1(\mathbf{y})$.

Now consider (ii), i.e., $\text{queue}_1(\mathbf{x}) = B$. It follows that $\text{queue}_1(\mathbf{y}) = B$. Let $\mathbf{y1} = (y_1 + 1, y_2 - 1, y_3, y_4)$. Note that $\text{diff}_1(\mathbf{y1}) = \text{diff}_1(\mathbf{y}) + 1$, and $\text{diff}_2(\mathbf{y1}) = \text{diff}_2(\mathbf{y}) - 1$. It follows that $\text{diff}_1(\mathbf{y1}) = \text{diff}_1(\mathbf{x}) + p + 1$, $\text{diff}_2(\mathbf{y1}) = \text{diff}_2(\mathbf{x}) - p - 1$, and $\text{diff}_1(\mathbf{x}) \geq \text{diff}_2(\mathbf{x})$. Also, $\text{queue}_i(\mathbf{x}) = \text{queue}_i(\mathbf{y1})$ $i = 1, 2$. From part (3) of hypothesis,

$$J(\mathbf{x}) \leq J(\mathbf{y1}) \text{provided } \mathbf{y1} \text{ is feasible, i.e., if } y_2 > 0 \tag{75}$$

$$J(\mathbf{x}) \leq J(\mathbf{y}) \tag{76}$$

From (34) and (35)

$$\theta_1(\mathbf{x}) \leq 1 + \beta J(\mathbf{x}) \tag{77}$$

From (34) and (35)

$$\theta_1(\mathbf{y}) = \begin{cases} 1 + \beta J(\mathbf{y}) & \text{if } y_2 = 0, \\ 1 + \beta \min\left(J(\mathbf{y1}), J(\mathbf{y})\right) \quad \mathbf{y1} \text{ feasible, i.e.,} & \text{if } y_2 > 0 \end{cases} \tag{78}$$

From (75) to (78) $\theta_1(\mathbf{x}) \leq \theta_1(\mathbf{y})$.

Now, we will show that $\theta_2(\mathbf{x}) \leq \theta_2(\mathbf{y})$. We consider two cases separately: (i)$\text{queue}_1(\mathbf{x}) < B$, and (ii)$\text{queue}_1(\mathbf{x}) = B$. Under (i),$\text{queue}_1(\mathbf{y}) < B$. Let $\mathbf{x1} = (x_1, x_2+1, x_3, x_4)$, and $\mathbf{y1} = (y_1, y_2 + 1, y_3, y_4)$. From (33) $\theta_2(\mathbf{x}) = \beta J(\mathbf{x1})$ and $\theta_2(\mathbf{y}) = \beta J(\mathbf{y1})$. First assume that $\text{diff}_1(\mathbf{x}) > \text{diff}_2(\mathbf{x})$. Note that $\text{diff}_1(\mathbf{x1}) = \text{diff}_1(\mathbf{x})$, $\text{diff}_2(\mathbf{x1}) = \text{diff}_2(\mathbf{x}) + 1$, $\text{diff}_1(\mathbf{y1}) = \text{diff}_1(\mathbf{y})$ and $\text{diff}_2(\mathbf{y1}) = \text{diff}_2(\mathbf{y}) + 1$. It follows that $\text{diff}_1(\mathbf{y1}) = \text{diff}_1(\mathbf{x1}) + p$, $\text{diff}_2(\mathbf{y1}) = \text{diff}_2(\mathbf{x1}) - p$, and $\text{diff}_1(\mathbf{x1}) \geq \text{diff}_2(\mathbf{x1})$. Also, $\text{queue}_i(\mathbf{x1}) = \text{queue}_i(\mathbf{y1})$ $i = 1, 2$. From part (3) of hypothesis, $J(\mathbf{x1}) \leq J(\mathbf{y1})$. Thus $\theta_2(\mathbf{x}) \leq \theta_2(\mathbf{y})$.

Now assume that $\text{diff}_1(\mathbf{x}) = \text{diff}_2(\mathbf{x}) = r$. Let $\mathbf{x2} = (x_1+1, x_2, x_3, x_4)$. Thus $\text{diff}_1(\mathbf{x1}) = r$, $\text{diff}_2(\mathbf{x1}) = r+1$, $\text{diff}_1(\mathbf{x2}) = \text{diff}_1(\mathbf{x}) + 1 = r + 1$, $\text{diff}_2(\mathbf{x2}) = \text{diff}_2(\mathbf{x}) = r$. Thus, $\text{diff}_1(\mathbf{x2}) = \text{diff}_2(\mathbf{x1})$, $\text{diff}_2(\mathbf{x2}) = \text{diff}_1(\mathbf{x1})$. Also, $\text{queue}_i(\mathbf{x1}) = \text{queue}_i(\mathbf{x2})$ $i = 1, 2$. From part (2) of hypothesis, $J(\mathbf{x1}) = J(\mathbf{x2})$. Now we will show that $J(\mathbf{x2}) \leq J(\mathbf{y1})$. Note that

$$\begin{aligned}
\text{diff}_1(\mathbf{y1}) &= \text{diff}_1(\mathbf{y}) && (79)\\
&= \text{diff}_1(\mathbf{x}) + p \\
&= \text{diff}_1(\mathbf{x2}) + p - 1 \text{ (since } \text{diff}_1(\mathbf{x2}) = \text{diff}_1(\mathbf{x}) + 1) && (80)\\
\text{diff}_2(\mathbf{y1}) &= \text{diff}_2(\mathbf{y}) + 1 \\
&= \text{diff}_2(\mathbf{x}) - p + 1 \\
&= \text{diff}_2(\mathbf{x2}) - (p-1) \text{ (since } \text{diff}_2(\mathbf{x2}) = \text{diff}_2(\mathbf{x})) && (81)\\
\text{diff}_1(\mathbf{x2}) &> \text{diff}_2(\mathbf{x2}) \text{ (since } \text{diff}_1(\mathbf{x2}) = r+1, \text{diff}_2(\mathbf{x2}) = r) && (82)\\
\text{queue}_i(\mathbf{x2}) &= \text{queue}_i(\mathbf{y1}), i = 1, 2 && (83)
\end{aligned}$$

From (80) to (83), $p > 0$, part (3) of hypothesis, $J(\mathbf{x2}) \leq J(\mathbf{y1})$. It follows that

$$\begin{aligned}
\theta_2(\mathbf{x}) &= \beta J(\mathbf{x1}) \\
&= \beta J(\mathbf{x2}) \\
&\leq \beta J(\mathbf{y1}) \\
&= \theta_2(\mathbf{y})
\end{aligned}$$

Now consider case (ii), i.e., $\text{queue}_1(\mathbf{x}) = B$. It follows that $\text{queue}_1(\mathbf{y}) = B$. Note that $y_1 > 0$ in this case. (If $y_1 = 0$, $\text{diff}_1(\mathbf{y}) \leq 0$, but $\text{diff}_2(\mathbf{y}) \geq 0$, as $y_2 = B$, however $\text{diff}_1(\mathbf{y}) > \text{diff}_2(\mathbf{y})$ by assumption). Let $\mathbf{y1} = (y_1 - 1, y_2 + 1, y_3, y_4)$ ($\mathbf{y1}$ is feasible). Note that $\text{diff}_1(\mathbf{y1}) = \text{diff}_1(\mathbf{y}) - 1$, and $\text{diff}_2(\mathbf{y1}) = \text{diff}_2(\mathbf{y}) + 1$. It follows that $\text{diff}_1(\mathbf{y1}) = \text{diff}_1(\mathbf{x}) + p - 1$, $\text{diff}_2(\mathbf{y1}) = \text{diff}_2(\mathbf{x}) - (p-1)$, and $\text{diff}_1(\mathbf{x}) \geq \text{diff}_2(\mathbf{x})$. Also, $\text{queue}_i(\mathbf{x}) = \text{queue}_i(\mathbf{y1})$ $i = 1, 2$. From part (3) of hypothesis,

$$J(\mathbf{x}) \leq \min\left(J(\mathbf{y1}), J(\mathbf{y})\right) \tag{84}$$

$$\begin{aligned}
\theta_2(\mathbf{x}) &\leq 1 + \beta J(\mathbf{x}) \text{ (from (34) and (35))} && (85)\\
\theta_2(\mathbf{y}) &= 1 + \beta \min\left(J(\mathbf{y1}), J(\mathbf{y})\right) \text{ (from (34))} && (86)
\end{aligned}$$

From (84) to (86) $\theta_2(\mathbf{x}) \leq \theta_2(\mathbf{y})$.

$$\text{Thus } \theta_i(\mathbf{x}) \leq \theta_i(\mathbf{y}), \forall i \tag{87}$$

From (74) and (87), $TJ(\mathbf{x}) \leq TJ(\mathbf{y})$. Part (3) follows. $\diamond$

From property (1) of Lemma 4 if all of the queues have packets then either pair can be scheduled, i.e.,

$$\phi(q_1, q_2, q_3, q_4) = \begin{cases} \beta J(q_1 - 1, q_2, q_3, q_4 - 1) & \text{if } \min(q_1, q_4) > 0 \\ \beta J(q_1, q_2 - 1, q_3 - 1, q_4) & \text{if } \min(q_2, q_3) > 0 \end{cases} \tag{88}$$

Proof of Lemma 5: We will assume that $J(q_1, q_2, q_3, q_4) = J(q_3, q_4, q_1, q_2)$ and show that $TJ(q_1, q_2, q_3, q_4) = TJ(q_3, q_4, q_1, q_2)$. We assume that $B_1 = B_2$ and $\lambda_i = \lambda$, $\forall i$.

Note that $\phi(q_1, q_2, q_3, q_4) = \phi(q_3, q_4, q_1, q_2)$, from hypothesis and (32).

We will show that $\theta_1(q_1, q_2, q_3, q_4) = \theta_3(q_3, q_4, q_1, q_2)$, $\theta_3(q_1, q_2, q_3, q_4) = \theta_1(q_3, q_4, q_1, q_2)$, $\theta_2(q_1, q_2, q_3, q_4) = \theta_4(q_3, q_2, q_1, q_2)$, and $\theta_4(q_1, q_2, q_3, q_4) = \theta_2(q_1, q_2, q_3, q_4)$. The result follows since $\lambda_i = \lambda$, $\forall i$. We just show that $\theta_1(q_1, q_2, q_3, q_4) = \theta_3(q_3, q_4, q_1, q_2)$. The result follows similarly for others. First consider the case that $q_1 + q_2 < B$. From (33),

$$\begin{aligned} \theta_1(q_1, q_2, q_3, q_4) &= \beta J(q_1 + 1, q_2, q_3, q_4) \\ &= \beta J(q_3, q_4, q_1 + 1, q_2) \text{ (by hypothesis)} \\ &= \theta_3(q_3, q_4, q_1, q_2) \text{ (from (33))} \end{aligned}$$

Now consider the case that $q_1 + q_2 = B$, and $q_2 > 0$.

$$\begin{aligned} \theta_1(q_1, q_2, q_3, q_4) &= 1 + \beta \min\left(J(q_1 + 1, q_2 - 1, q_3, q_4), J(q_1, q_2, q_3, q_4)\right) \text{ (from (34))} \\ &= 1 + \beta \min\left(J(q_3, q_4, q_1 + 1, q_2 - 1), J(q_3, q_4, q_1, q_2)\right) \text{ (by hypothesis)} \\ &= \theta_3(q_3, q_4, q_1, q_2) \text{ (from (34))} \end{aligned}$$

Now consider the case that $q_1 + q_2 = B$, and $q_2 = 0$. Thus $q_1 = B$.

$$\begin{aligned} \theta_1(B, 0, q_3, q_4) &= 1 + \beta J(B, 0, q_3, q_4) \text{ (from (35))} \\ &= 1 + \beta J(q_3, q_4, B, 0) \text{ (from hypothesis)} \\ &= \theta_2(q_3, q_4, B, 0) \text{ (from (35))} \end{aligned}$$

The result follows. $\diamond$

Proof of Lemma 6: We will assume this property for the function $J(\mathbf{z})$ and prove it for the function $TJ(\mathbf{z})$. We prove the result for part (1). The argument follows similarly for part (2). We assume that $B_1 = B_2$ and $\lambda_i = \lambda$, $\forall i$.

First, we will prove some properties of states $\mathbf{x}$ and $\mathbf{y}$.

$$\begin{aligned} \text{diff}_i(\mathbf{x}) &= \text{diff}_i(\mathbf{y}) - 1 \forall i && (89) \\ \text{queue}_i(\mathbf{x}) &> 0 \forall i && (90) \end{aligned}$$

We first prove (89).

$$\begin{aligned} (x_1 - x_4) + (x_2 - x_3) &= (x_1 + x_2) - (x_3 + x_4) \\ &= \text{queue}_1(\mathbf{x}) - \text{queue}_2(\mathbf{x}) \end{aligned}$$

$$
\begin{aligned}
&= \text{queue}_1(\mathbf{y}) - 1 - \text{queue}_2(\mathbf{y}) - 1 \\
&= (y_1 + y_2) - 1 - (y_3 + y_4) - 1 \\
&= (y_1 - y_4 - 1) + (y_2 - y_3 - 1) \qquad (91) \\
x_1 - x_4 &\le y_1 - y_4 - 1 \text{ (since } \text{diff}_i(\mathbf{x}) < \text{diff}_i(\mathbf{y}), i = 1, 2) \qquad (92) \\
x_2 - x_3 &\le y_2 - y_3 - 1 \text{ (since } \text{diff}_i(\mathbf{x}) < \text{diff}_i(\mathbf{y}), i = 1, 2) \qquad (93)
\end{aligned}
$$

From (91) to (93)

$$
\begin{aligned}
x_1 - x_4 &= y_1 - y_4 - 1 \qquad (94) \\
x_2 - x_3 &= y_2 - y_3 - 1 \qquad (95)
\end{aligned}
$$

Thus (89) follows.

Inequality (90) follows since $\text{queue}_1(\mathbf{x}) \ge \text{queue}_2(\mathbf{x}) > \text{queue}_2(\mathbf{y}) \ge 0$.

Now we show that

$$
\begin{aligned}
&\text{If } x_1 = x_3 = 0, \text{ then } y_1 = y_3 = 0 \qquad (96) \\
&\text{If } x_2 = x_4 = 0, \text{ then } y_2 = y_4 = 0 \qquad (97)
\end{aligned}
$$

Consider (96). Let $x_1 = x_3 = 0$. From (95) $y_2 - y_3 = x_2 + 1$. Since $\text{queue}_1(\mathbf{y}) = \text{queue}_1(\mathbf{x}) + 1$, $y_1 + y_2 = x_2 + 1$. It follows that $y_1 = y_3 = 0$.

Consider (97). Let $x_2 = x_4 = 0$. From (94) $y_1 - y_4 = x_1 + 1$. Since $\text{queue}_1(\mathbf{y}) = \text{queue}_1(\mathbf{x}) + 1$, $y_1 + y_2 = x_1 + 1$. It follows that $y_2 = y_4 = 0$.

Now we will show that $\phi(\mathbf{x}) \le \phi(\mathbf{y})$. For this we first show that $\max(\min(x_1, x_4), \min(x_2, x_3)) > 0$ if $\max(\min(y_1, y_4), \min(y_2, y_3)) > 0$. Assume that $\max(\min(x_1, x_4), \min(x_2, x_3)) = 0$. From (90), either $x_1 = x_3 = 0$ or $x_2 = x_4 = 0$. From (96) and (97) $\max(\min(y_1, y_4), \min(y_2, y_3)) = 0$ in both cases.

Consider two separate cases: (i)$\max(\min(y_1, y_4), \min(y_2, y_3)) > 0$ and (ii) $\max(\min(y_1, y_4), \min(y_2, y_3)) = 0$. In the first case, from the previous argument, $\max(\min(x_1, x_4), \min(x_2, x_3)) > 0$. Let $\mathbf{x1} = (x_1 - 1, x_2, x_3, x_4 - 1)$, $\mathbf{x2} = (x_1, x_2 - 1, x_3 - 1, x_4)$ and $\mathbf{y1} = (y_1 - 1, y_2, y_3, y_4 - 1)$, and $\mathbf{y2} = (y_1, y_2 - 1, y_3 - 1, y_4)$. Note that $\text{diff}_i(\mathbf{yj}) = \text{diff}_i(\mathbf{y})$, and $\text{diff}_i(\mathbf{xk}) = \text{diff}_i(\mathbf{x})$, $i, j, k \in \{1, 2\}$. Also, $\text{queue}_i(\mathbf{xj}) = \text{queue}_i(\mathbf{x}) - 1$ and $\text{queue}_i(\mathbf{yk}) = \text{queue}_i(\mathbf{y}) - 1$ $i, j, k \in \{1, 2\}$. Thus $\mathbf{xj}$ and $\mathbf{yk}$ satisfy the condition of part (1) of the lemma for any $j, k \in \{1, 2\}$. By hypothesis,

$$
J(\mathbf{xj}) \le J(\mathbf{yk}), j, k \in \{1, 2\}, \qquad (98)
$$

All of the states $\mathbf{x1}$, $\mathbf{x2}$, $\mathbf{y1}$, $\mathbf{y2}$ need not be feasible. The inequality applies to the feasible ones only. Since $\max(\min(x_1, x_4), \min(x_2, x_3)) > 0$ and $\max(\min(y_1, y_4), \min(y_2, y_3)) > 0$ from (88)

$$
\phi(\mathbf{x}) = \begin{cases} \beta J(\mathbf{x1}) & \mathbf{x1} \text{ feasible, } \quad \text{if } \min(x_1, x_4) > 0, \\ \beta J(\mathbf{x2}), & \mathbf{x2} \text{ feasible } \quad \text{otherwise.} \end{cases} \qquad (99)
$$

$$
\phi(\mathbf{y}) = \begin{cases} \beta J(\mathbf{y1}) & \mathbf{y1} \text{ feasible, } \quad \text{if } \min(y_1, y_4) > 0, \\ \beta J(\mathbf{y2}), & \mathbf{y2} \text{ feasible } \quad \text{otherwise} \end{cases} \qquad (100)
$$

From (98) to (100) $\phi(\mathbf{x}) \le \phi(\mathbf{y})$ in this case.

We will now consider the case that $\max(\min(y_1, y_4), \min(y_2, y_3)) = 0$. Let $\mathbf{x1} = ((x_1-1)^+, x_2, x_3, (x_4-1)^+)$, $\mathbf{x2} = (x_1, (x_2-1)^+, (x_3-1)^+, x_4)$ and $\mathbf{y1} = ((y_1-1)^+, y_2, y_3, (y_4-1)^+)$, and $\mathbf{y2} = (y_1, (y_2-1)^+, (y_3-1)^+, y_4)$. From (32)

$$\begin{aligned}\phi(\mathbf{x}) &= \beta\min(J(\mathbf{x1}), J(\mathbf{x2})) &(101)\\ \phi(\mathbf{y}) &= \beta\min(J(\mathbf{y1}), J(\mathbf{y2})) &(102)\end{aligned}$$

We will show that

$$\begin{aligned}J(\mathbf{y1}) &\geq \min(J(\mathbf{x1}), J(\mathbf{x2})) \text{ if } \mathbf{y1} \neq \mathbf{y} &(103)\\ J(\mathbf{y2}) &\geq \min(J(\mathbf{x1}), J(\mathbf{x2})) \text{ if } \mathbf{y2} \neq \mathbf{y} &(104)\end{aligned}$$

$$\text{From Lemma 2, } J(\mathbf{y}) \geq J(\mathbf{yk}), k = 1, 2 \qquad (105)$$

Since $\text{queue}_1(\mathbf{y}) > \text{queue}_1(\mathbf{x}) > 0$, $y_i > 0$ for some $i \in \{1, 2\}$. Thus either $\mathbf{y1} \neq \mathbf{y}$ or $\mathbf{y1} \neq \mathbf{y2}$. Thus $\phi(\mathbf{x}) \leq \phi(\mathbf{y})$ from (101) to (105).

First we show (103). Let $\mathbf{y1} \neq \mathbf{y}$. It follows that $y_1 > 0$ or $y_4 > 0$. Assume $y_1 > 0$. Since $\max(\min(y_1, y_4), \min(y_2, y_3)) = 0$, $y_4 = 0$.

$$\text{Thus } \mathbf{y1} = (y_1 - 1, y_2, y_3, y_4) \qquad (106)$$

Let $x_3 > 0$. Consider $\mathbf{x3} = (x_1, x_2, x_3 - 1, x_4)$. It follows that $\text{diff}_1(\mathbf{y1}) = \text{diff}_1(\mathbf{y}) - 1$, $\text{diff}_2(\mathbf{y1}) = \text{diff}_2(\mathbf{y})$, $\text{diff}_1(\mathbf{x3}) = \text{diff}_1(\mathbf{x})$ and $\text{diff}_2(\mathbf{x3}) = \text{diff}_2(\mathbf{x}) + 1$. From (89), $\text{diff}_i(\mathbf{y1}) = \text{diff}_i(\mathbf{x3})$, $i = 1, 2$. Also, $\text{queue}_1(\mathbf{x3}) = \text{queue}_1(\mathbf{x})$, $\text{queue}_2(\mathbf{x3}) = \text{queue}_2(\mathbf{x}) - 1$, $\text{queue}_1(\mathbf{y1}) = \text{queue}_1(\mathbf{y}) - 1$ and $\text{queue}_2(\mathbf{y1}) = \text{queue}_2(\mathbf{y})$. It follows that $\text{queue}_i(\mathbf{y1}) = \text{queue}_i(\mathbf{x3})$.

$$\begin{aligned}J(\mathbf{y1}) &= J(\mathbf{x3}) \text{ (from part 1 of Lemma 4)}\\ &\geq J(\mathbf{x2}) \text{ (from Lemma 2)} &(107)\end{aligned}$$

Now (103) follows from (107).

Now consider the case $x_3 = 0$. From (90) $x_4 > 0$. From (96), if $x_1 = x_3 = 0$ then $y_1 = 0$. Since $x_3 = 0$ and $y_1 > 0$ it follows that $x_1 > 0$. Thus $\mathbf{x1} = (x_1 - 1, x_2, x_3, x_4 - 1)$. From (95) and since $x_3 = 0$, $y_2 > 0$. Thus the state $\mathbf{y3} = (y_1 - 1, y_2 - 1, y_3, y_4)$ is feasible ($y_1 > 0$ by assumption). Note that $\text{diff}_i(\mathbf{y3}) = \text{diff}_i(\mathbf{y}) - 1$ and $\text{diff}_i(\mathbf{x1}) = \text{diff}_i(\mathbf{x})$, $i = 1, 2$. From (89), $\text{diff}_i(\mathbf{y3}) = \text{diff}_i(\mathbf{x1})$, $i = 1, 2$. Also, $\text{queue}_i(\mathbf{x1}) = \text{queue}_i(\mathbf{x}) - 1$, $\text{queue}_1(\mathbf{y3}) = \text{queue}_1(\mathbf{y}) - 2$ and $\text{queue}_2(\mathbf{y3}) = \text{queue}_2(\mathbf{y})$. It follows that $\text{queue}_i(\mathbf{y3}) = \text{queue}_i(\mathbf{x1})$.

$$\begin{aligned}J(\mathbf{x1}) &= J(\mathbf{y3}) \text{ (from part (1) of Lemma 4)}\\ &\leq J(\mathbf{y1}) \text{ (from Lemma 2 and (106))} &(108)\end{aligned}$$

Now (103) follows from (108).

Assume $y_1 = 0$. It follows that $y_4 > 0$. Thus $\mathbf{y1} = (y_1, y_2, y_3, y_4 - 1)$. We will show that $x_4 > 0$. Let $x_4 = 0$. From (94) $y_1 > 0$, which is not possible. Thus $x_4 > 0$. Let $\mathbf{x4} = (x_1, x_2, x_3, x_4 - 1)$. It follows that $\text{diff}_1(\mathbf{y1}) = \text{diff}_1(\mathbf{y}) + 1$, $\text{diff}_2(\mathbf{y1}) = \text{diff}_2(\mathbf{y})$, $\text{diff}_1(\mathbf{x4}) = \text{diff}_1(\mathbf{x}) + 1$ and $\text{diff}_2(\mathbf{x4}) = \text{diff}_2(\mathbf{x})$. From (89), $\text{diff}_i(\mathbf{x4}) = \text{diff}_i(\mathbf{y1}) - 1$, $i = 1, 2$. Also, $\text{queue}_1(\mathbf{x4}) = \text{queue}_1(\mathbf{x})$, $\text{queue}_2(\mathbf{x4}) = \text{queue}_2(\mathbf{x}) - 1$, $\text{queue}_1(\mathbf{y1}) = \text{queue}_1(\mathbf{y})$ and $\text{queue}_2(\mathbf{y1}) = \text{queue}_2(\mathbf{y}) - 1$. Since

$\mathrm{queue}_1(\mathbf{x}) \geq \mathrm{queue}_2(\mathbf{x})$, $\mathrm{queue}_1(\mathbf{x4}) > \mathrm{queue}_2(\mathbf{x4})$. Also, clearly $\mathrm{queue}_1(\mathbf{y1}) = \mathrm{queue}_1(\mathbf{x4})+1$, and $\mathrm{queue}_2(\mathbf{y1}) = \mathrm{queue}_2(\mathbf{x4}) - 1$. Thus conditions of part (1) are satisfied.

$$\begin{aligned} J(\mathbf{y1}) &\geq J(\mathbf{x4}) \text{ (from hypothesis)} \\ &\geq J(\mathbf{x1}) \text{ (from Lemma 2)} \end{aligned} \tag{109}$$

Now (103) follows from (109).

Thus (103) holds in all these cases. We present the proof for (104) next.

Now, we show (104). Let $\mathbf{y2} \neq \mathbf{y}$. It follows that $y_2 > 0$ or $y_3 > 0$. Assume $y_2 > 0$. Since $\max(\min(y_1, y_4), \min(y_2, y_3)) = 0$, $y_3 = 0$.

$$\text{Thus } \mathbf{y2} = (y_1, y_2 - 1, y_3, y_4) \tag{110}$$

Let $x_4 > 0$. Consider $\mathbf{x4} = (x_1, x_2, x_3, x_4 - 1)$. It follows that $\mathrm{diff}_1(\mathbf{y2}) = \mathrm{diff}_1(\mathbf{y})$, $\mathrm{diff}_2(\mathbf{y2}) = \mathrm{diff}_2(\mathbf{y}) - 1$, $\mathrm{diff}_1(\mathbf{x4}) = \mathrm{diff}_1(\mathbf{x}) + 1$ and $\mathrm{diff}_2(\mathbf{x4}) = \mathrm{diff}_2(\mathbf{x})$. From (89), $\mathrm{diff}_i(\mathbf{y2}) = \mathrm{diff}_i(\mathbf{x4})$, $i = 1, 2$. Also, $\mathrm{queue}_1(\mathbf{x4}) = \mathrm{queue}_1(\mathbf{x})$, $\mathrm{queue}_2(\mathbf{x4}) = \mathrm{queue}_2(\mathbf{x}) - 1$, $\mathrm{queue}_1(\mathbf{y2}) = \mathrm{queue}_1(\mathbf{y}) - 1$ and $\mathrm{queue}_2(\mathbf{y2}) = \mathrm{queue}_2(\mathbf{y})$. It follows that $\mathrm{queue}_i(\mathbf{y2}) = \mathrm{queue}_i(\mathbf{x4})$.

$$\begin{aligned} J(\mathbf{y2}) &= J(\mathbf{x4}) \text{ (from part (1) of Lemma 4)} \\ &\geq J(\mathbf{x2}) \text{ (from Lemma 2)} \end{aligned} \tag{111}$$

Now (104) follows from (111).

Now consider the case that $x_4 = 0$. From (90) $x_3 > 0$. ¿From (97) if $x_2 = x_4 = 0$ then $y_2 = 0$. Since $x_4 = 0$ and $y_2 > 0$ it follows that $x_2 > 0$. Thus $\mathbf{x2} = (x_1, x_2 - 1, x_3 - 1, x_4)$. From (94) and since $x_4 = 0$, $y_1 > 0$. Thus the state $\mathbf{y3} = (y_1 - 1, y_2 - 1, y_3, y_4)$ is feasible ($y_2 > 0$ by assumption). Note that $\mathrm{diff}_i(\mathbf{y3}) = \mathrm{diff}_i(\mathbf{y}) - 1$ and $\mathrm{diff}_i(\mathbf{x2}) = \mathrm{diff}_i(\mathbf{x})$, $i = 1, 2$. From (89), $\mathrm{diff}_i(\mathbf{y3}) = \mathrm{diff}_i(\mathbf{x2})$, $i = 1, 2$. Also, $\mathrm{queue}_i(\mathbf{x2}) = \mathrm{queue}_i(\mathbf{x}) - 1$, $\mathrm{queue}_1(\mathbf{y3}) = \mathrm{queue}_1(\mathbf{y}) - 2$ and $\mathrm{queue}_2(\mathbf{y3}) = \mathrm{queue}_2(\mathbf{y})$. It follows that $\mathrm{queue}_i(\mathbf{y3}) = \mathrm{queue}_i(\mathbf{x2})$.

$$\begin{aligned} J(\mathbf{x2}) &= J(\mathbf{y3}) \text{ (from part (1) of Lemma 4)} \\ &\leq J(\mathbf{y2}) \text{ (from Lemma 2 and (110))} \end{aligned} \tag{112}$$

Now (104) follows from (112).

Assume $y_2 = 0$. It follows that $y_3 > 0$. Thus $\mathbf{y2} = (y_1, y_2, y_3 - 1, y_4)$. We will show that $x_3 > 0$. Let $x_3 = 0$. From (95) $y_2 > 0$, which is not possible. Thus $x_3 > 0$. Let $\mathbf{x3} = (x_1, x_2, x_3 - 1, x_4)$. It follows that $\mathrm{diff}_1(\mathbf{y2}) = \mathrm{diff}_1(\mathbf{y})$, $\mathrm{diff}_2(\mathbf{y2}) = \mathrm{diff}_2(\mathbf{y}) + 1$, $\mathrm{diff}_1(\mathbf{x3}) = \mathrm{diff}_1(\mathbf{x})$ and $\mathrm{diff}_2(\mathbf{x3}) = \mathrm{diff}_2(\mathbf{x})+1$. From (89) $\mathrm{diff}_i(\mathbf{x3}) = \mathrm{diff}_i(\mathbf{y2})-1$, $i = 1, 2$. Also, $\mathrm{queue}_1(\mathbf{x3}) = \mathrm{queue}_1(\mathbf{x})$, $\mathrm{queue}_2(\mathbf{x3}) = \mathrm{queue}_2(\mathbf{x}) - 1$, $\mathrm{queue}_1(\mathbf{y2}) = \mathrm{queue}_1(\mathbf{y})$ and $\mathrm{queue}_2(\mathbf{y2}) = \mathrm{queue}_2(\mathbf{y}) - 1$. Since $\mathrm{queue}_1(\mathbf{x}) \geq \mathrm{queue}_2(\mathbf{x})$, $\mathrm{queue}_1(\mathbf{x3}) > \mathrm{queue}_2(\mathbf{x3})$. Also, clearly $\mathrm{queue}_1(\mathbf{y2}) = \mathrm{queue}_1(\mathbf{x3})+1$, and $\mathrm{queue}_2(\mathbf{y2}) = \mathrm{queue}_2(\mathbf{x3}) - 1$. Thus conditions of part (1) are satisfied.

$$\begin{aligned} J(\mathbf{y2}) &\geq J(\mathbf{x3}) \text{ (from hypothesis)} \\ &\geq J(\mathbf{x2}) \text{ (from Lemma 2)} \end{aligned} \tag{113}$$

Now (104) follows from (113).

Thus (104) holds in all these cases.

$$\text{Thus } \phi(\mathbf{x}) \le \phi(\mathbf{y}) \tag{114}$$

Now we show that $\theta_i(\mathbf{x}) \le \theta_i(\mathbf{y})$. First consider $i = 1$. Let $\text{queue}_1(\mathbf{y}) < B$. Let $\mathbf{y1} = (y_1 + 1, y_2, y_3, y_4)$, $\mathbf{x1} = (x_1 + 1, x_2, x_3, x_4)$. It follows that $\text{diff}_1(\mathbf{y1}) = \text{diff}_1(\mathbf{y}) + 1$, $\text{diff}_2(\mathbf{y1}) = \text{diff}_2(\mathbf{y})$, $\text{diff}_1(\mathbf{x1}) = \text{diff}_1(\mathbf{x}) + 1$ and $\text{diff}_2(\mathbf{x1}) = \text{diff}_2(\mathbf{x})$. From (89) $\text{diff}_i(\mathbf{x1}) = \text{diff}_i(\mathbf{y1}) - 1$, $i = 1, 2$. Also, $\text{queue}_1(\mathbf{x1}) = \text{queue}_1(\mathbf{x}) + 1$, $\text{queue}_2(\mathbf{x1}) = \text{queue}_2(\mathbf{x})$, $\text{queue}_1(\mathbf{y1}) = \text{queue}_1(\mathbf{y}) + 1$ and $\text{queue}_2(\mathbf{y1}) = \text{queue}_2(\mathbf{y})$. Since $\text{queue}_1(\mathbf{x}) \ge \text{queue}_2(\mathbf{x})$, $\text{queue}_1(\mathbf{x1}) > \text{queue}_2(\mathbf{x1})$. Also, clearly $\text{queue}_1(\mathbf{y1}) = \text{queue}_1(\mathbf{x1}) + 1$, and $\text{queue}_2(\mathbf{y1}) = \text{queue}_2(\mathbf{x1}) - 1$. Thus conditions of part (1) are satisfied.

$$\begin{aligned}
J(\mathbf{y1}) &\ge J(\mathbf{x1}) \text{ (from hypothesis)} && (115)\\
\theta_1(\mathbf{x}) &= \beta J(\mathbf{x1}) \text{ (from (33))} && (116)\\
\theta_1(\mathbf{y}) &= \beta J(\mathbf{y1}) \text{ (from (33))} && (117)
\end{aligned}$$

Now $\theta_1(\mathbf{x}) \le \theta_1(\mathbf{y})$ from (115) to (117).

Now assume that $\text{queue}_1(\mathbf{y}) = B$. Thus, $\text{queue}_2(\mathbf{x}) \le \text{queue}_1(\mathbf{x}) = B - 1$. Let $\mathbf{x1} = (x_1 + 1, x_2, x_3, x_4)$, and $\mathbf{y1} = (y_1 + 1, y_2 - 1, y_3, y_4)$. It follows that

$$\begin{aligned}
\theta_1(\mathbf{x}) &= \beta J(\mathbf{x1}) \text{ (from (33))} && (118)\\
\theta_1(\mathbf{y}) &= 1 + \beta \min\left(J(\mathbf{y}), J(\mathbf{y1})\right) \text{ if } y_2 > 0, \text{ (from (34))} && (119)\\
\theta_1(\mathbf{y}) &= 1 + \beta J(\mathbf{y}), \text{ if } y_2 = 0, \text{ (from (35))} && (120)\\
J(\mathbf{x1}) &\le 1 + J(\mathbf{x}) \text{ (from Lemma 2)} &&\\
&\le 1 + J(\mathbf{y}) \text{(from hypothesis)} && (121)
\end{aligned}$$

From (118), (120) and (121) and since $\beta < 1$, $\theta_1(\mathbf{x}) \le \theta_1(\mathbf{y})$ if $y_2 = 0$. Assume that $y_2 > 0$. Let $x_2 > 0$. Let $\mathbf{x2} = (x_1 + 1, x_2 - 1, x_3, x_4)$. Thus $\mathbf{x2}$ is feasible. It follows that $\text{diff}_1(\mathbf{y1}) = \text{diff}_1(\mathbf{y}) + 1$, $\text{diff}_2(\mathbf{y2}) = \text{diff}_2(\mathbf{y}) - 1$, $\text{diff}_1(\mathbf{x2}) = \text{diff}_1(\mathbf{x}) + 1$ and $\text{diff}_2(\mathbf{x2}) = \text{diff}_2(\mathbf{x}) - 1$. From (89) $\text{diff}_i(\mathbf{x2}) = \text{diff}_i(\mathbf{y1}) - 1$, $i = 1, 2$. Also, $\text{queue}_i(\mathbf{x2}) = \text{queue}_i(\mathbf{x})$, and $\text{queue}_i(\mathbf{y1}) = \text{queue}_i(\mathbf{y})$. Thus $\text{queue}_1(\mathbf{x2}) \ge \text{queue}_2(\mathbf{x2})$, $\text{queue}_1(\mathbf{y1}) = \text{queue}_1(\mathbf{x2}) + 1$, and $\text{queue}_2(\mathbf{y1}) = \text{queue}_2(\mathbf{x2}) - 1$. Thus conditions of part (1) are satisfied.

$$\begin{aligned}
J(\mathbf{x1}) &\le 1 + J(\mathbf{x2}) \text{ (from Lemma 2)} &&\\
&\le 1 + J(\mathbf{y1}) \text{(from hypothesis)} && (122)
\end{aligned}$$

From (118), (119), (121) and (122) and since $\beta < 1$, $\theta_1(\mathbf{x}) \le \theta_1(\mathbf{y})$ if $y_2 > 0, x_2 > 0$.

Now consider the case $y_2 > 0, x_2 = 0$. Thus $x_1 = B - 1$. Since $\text{queue}_2(\mathbf{x}) \le \text{queue}_1(\mathbf{x}) = B - 1$, $x_4 \le B - 1$. Thus $\text{diff}_1(\mathbf{x}) \ge 0$. Also, $\text{diff}_2(\mathbf{x}) \le x_2 = 0$. It follows that $\text{diff}_1(\mathbf{x}) \ge \text{diff}_2(\mathbf{x})$. From (89), it follows that $\text{diff}_1(\mathbf{y}) \ge \text{diff}_2(\mathbf{y})$. Note that $\text{diff}_1(\mathbf{y1}) = \text{diff}_1(\mathbf{y}) + 1$ and $\text{diff}_2(\mathbf{y2}) = \text{diff}_2(\mathbf{y}) - 1$, and $\text{queue}_i(\mathbf{y1}) = \text{queue}_i(\mathbf{y})$. From part (3) of Lemma 4, $J(\mathbf{y}) \le J(\mathbf{y1})$. From (119),

$$\theta_1(\mathbf{y}) = 1 + \beta J(\mathbf{y}) \tag{123}$$

From (118), (121) and (123) and since $\beta < 1$, $\theta_1(\mathbf{x}) \le \theta_1(\mathbf{y})$ if $y_2 > 0, x_2 = 0$.

Thus $\theta_1(\mathbf{x}) \leq \theta_1(\mathbf{y})$ in all cases. Now, $\theta_2(\mathbf{x}) \leq \theta_2(\mathbf{y})$ can be proved similarly. Next we prove that $\theta_3(\mathbf{x}) \leq \theta_3(\mathbf{y})$. Observe that $\text{queue}_2(\mathbf{y}) < \text{queue}_2(\mathbf{x}) \leq \text{queue}_1(\mathbf{x}) < \text{queue}_1(\mathbf{y}) \leq B$. Let, $\mathbf{y3} = (y_1, y_2, y_3 + 1, y_4)$, $\mathbf{x3} = (x_1, x_2, x_3 + 1, x_4)$. Thus, from (33)

$$\begin{aligned} \theta_3(\mathbf{x}) &= \beta J(\mathbf{x3}) && (124) \\ \theta_3(\mathbf{y}) &= \beta J(\mathbf{y3}), && (125) \end{aligned}$$

Assume that $\text{queue}_2(\mathbf{x}) < \text{queue}_1(\mathbf{x})$. Observe that $\text{diff}_1(\mathbf{y3}) = \text{diff}_1(\mathbf{y})$, $\text{diff}_2(\mathbf{y3}) = \text{diff}_2(\mathbf{y}) - 1$, $\text{diff}_1(\mathbf{x3}) = \text{diff}_1(\mathbf{x})$ and $\text{diff}_2(\mathbf{x3}) = \text{diff}_2(\mathbf{x}) - 1$. From (89) $\text{diff}_i(\mathbf{x3}) = \text{diff}_i(\mathbf{y3}) - 1$, $i = 1, 2$. Also, $\text{queue}_1(\mathbf{x3}) = \text{queue}_1(\mathbf{x})$, $\text{queue}_2(\mathbf{x3}) = \text{queue}_2(\mathbf{x}) + 1$, $\text{queue}_1(\mathbf{y3}) = \text{queue}_1(\mathbf{y})$ and $\text{queue}_2(\mathbf{y3}) = \text{queue}_2(\mathbf{y}) + 1$. Since $\text{queue}_1(\mathbf{x}) > \text{queue}_2(\mathbf{x})$, $\text{queue}_1(\mathbf{x3}) \geq \text{queue}_2(\mathbf{x1})$. Also, clearly $\text{queue}_1(\mathbf{y3}) = \text{queue}_1(\mathbf{x3}) + 1$, and $\text{queue}_2(\mathbf{y3}) = \text{queue}_2(\mathbf{x3}) - 1$. Thus conditions of part (1) are satisfied. From hypothesis, $J(\mathbf{y3}) \geq J(\mathbf{x3})$. The result follows from (124) and (125).

Now assume that $\text{queue}_2(\mathbf{x}) = \text{queue}_1(\mathbf{x})$. It follows that $\text{diff}_1(\mathbf{x}) + \text{diff}_2(\mathbf{x}) = 0$. Let $\mathbf{x4} = (x_3 + 1, x_4, x_1, x_2)$. Thus using (89), $\text{diff}_1(\mathbf{x4}) = \text{diff}_1(\mathbf{x}) + 1 = \text{diff}_1(\mathbf{y})$, and $\text{diff}_2(\mathbf{x4}) = \text{diff}_2(\mathbf{x}) = \text{diff}_2(\mathbf{y}) - 1$. Recall that $\text{diff}_1(\mathbf{y3}) = \text{diff}_1(\mathbf{y})$, $\text{diff}_2(\mathbf{y3}) = \text{diff}_2(\mathbf{y}) - 1$. Thus $\text{diff}_i(\mathbf{x4}) = \text{diff}_i(\mathbf{y3})$, $i = 1, 2$. Now, $\text{queue}_1(\mathbf{x4}) = \text{queue}_2(\mathbf{x}) + 1$, $\text{queue}_2(\mathbf{x4}) = \text{queue}_1(\mathbf{x})$. Since $\text{queue}_2(\mathbf{x}) = \text{queue}_1(\mathbf{x})$, $\text{queue}_1(\mathbf{x4}) = \text{queue}_1(\mathbf{x}) + 1$, and $\text{queue}_2(\mathbf{x4}) = \text{queue}_2(\mathbf{x})$. It follows that $\text{queue}_1(\mathbf{x4}) = \text{queue}_1(\mathbf{y})$, and $\text{queue}_2(\mathbf{x4}) = \text{queue}_2(\mathbf{y}) + 1$. Recall that $\text{queue}_1(\mathbf{y3}) = \text{queue}_1(\mathbf{y})$ and $\text{queue}_2(\mathbf{y3}) = \text{queue}_2(\mathbf{y}) + 1$. It follows that $\text{queue}_i(\mathbf{y3}) = \text{queue}_i(\mathbf{x4})$. From part (1) of Lemma 4, $J(\mathbf{y3}) = J(\mathbf{x4})$. Note that $J(\mathbf{x4}) = J(\mathbf{x3})$ from Lemma 5. The result follows from (124) and (125).

Thus $\theta_3(\mathbf{x}) \leq \theta_3(\mathbf{y})$ in all cases. Now, $\theta_4(\mathbf{x}) \leq \theta_4(\mathbf{y})$ can be proved similarly.

$$\text{Thus } \theta_i(\mathbf{x}) \leq \theta_i(\mathbf{y}) \qquad (126)$$

The lemma follows from (114) and (126). $\diamond$

References

[1] M. Arpaci and J. Copeland. Buffer-management of shared memory atm switches. *IEEE Communications Surveys*, 2000.

[2] I. Cidon, L. Georgiadis, R. Guerin, and A. Khamisy. Optimal buffer sharing. *IEEE Journal on Selected Areas in Communications*, 13(7), 1995.

[3] H. Gabow and O. Kariv. Algorithms for edge coloring bipartite graphs and multigraphs. *SIAM Journal of Computing*, 11(1), February 1992.

[4] S. Iyer and N. Mckeown. Maximum size matchings and input queued switches. *Proceedings of the 40th Annual Allerton Conference on Communication, Control and Computing*, October 2002.

[5] M. Hall Jr. *Combinatorial Theory*. John Wiley and Sons, 1998.

[6] F. Kamoun and L. Kleinrock. Analysis of shared storage in a computer network node environment under general traffic conditions. *IEEE Transactions on Communications*, 28(7), July 1980.

[7] M. Karol, M. Hluchyj, and S. Morgan. Input versus output queueing on a space-division switch. *IEEE Transactions on Communications*, 35, December 1987.

[8] N. Mckeown, A. Mekkittikul, V. Anantharam, and J. Walrand. Achieving 100% throughput in an input queued switch. *IEEE Transactions on Communications*, 47(8), 1999.

[9] A. Mekkittikul and N. Mckeown. A practical scheduling algorithm to achieve 100% throughput in input-queued switches. *Proceedings of INFOCOM'1998*, April 1998.

[10] N. Ni and L. Bhuyan. Fair scheduling and buffer management in internet routers. *Proceedings of INFOCOM, NY*, June 2002.

[11] M. Putterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* John Wiley and Sons, 1998.

[12] S. Ross. *Introduction to Stochastic Dynamic Programming.* Academic Press, 1998.

[13] R. Roy and S. Panwar. Optimal space priority policies for shared memory atm systems. *Proceedings of Allerton Conference on Communication, Control, and Computing*, October 1997.

[14] L. Tassiulas. Linear complexity algorithms for maximum throughput in radio networks and input queued switchs. *Proceedings of INFOCOM'1998*, April 1998.

[15] L. Tassiulas and A. Ephremidis. Stability properties of constrained queueing systems and scheduling for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 37(12), 1992.

[16] T. Weller and B. Hajek. Scheduling nonuniform traffic in a packet switching system with small propagation delay. *Proceedings of INFOCOM'1994*, 1994.

[17] M. Yoo, C. Qiao, and S. Dixit. Qos performance of optical burst switching in ip-over-wdm networks. *IEEE Journal on Selected Areas in Communications*, 18(10), October 2000.