

Article

An Optimal Flow Admission and Routing Control Policy for Resource Constrained Networks

Essia Hamouda

J. H. Brown College of Business & Public Administration, California State University,
San Bernardino, CA 92407, USA; ehamouda@csusb.edu

Received: 5 October 2020; Accepted: 12 November 2020; Published: 17 November 2020



Abstract: Overloaded network devices are becoming an increasing problem especially in resource limited networks with the continuous and rapid increase of wireless devices and the huge volume of data generated. Admission and routing control policy at a network device can be used to balance the goals of maximizing throughput and ensuring sufficient resources for high priority flows. In this paper we formulate the admission and routing control problem of two types of flows where one has a higher priority than the other as a Markov decision problem. We characterize the optimal admission and routing policy, and show that it is a state-dependent threshold type policy. Furthermore, we conduct extensive numerical experiments to gain more insight into the behavior of the optimal policy under different systems' parameters. While dynamic programming can be used to solve such problems, the large size of the state space makes it untractable and too resource intensive to run on wireless devices. Therefore, we propose a fast heuristic that exploits the structure of the optimal policy. We empirically show that the heuristic performs very well with an average reward deviation of 1.4% from the optimal while being orders of magnitude faster than the optimal policy. We further generalize the heuristic for the general case of a system with n ($n > 2$) types of flows.

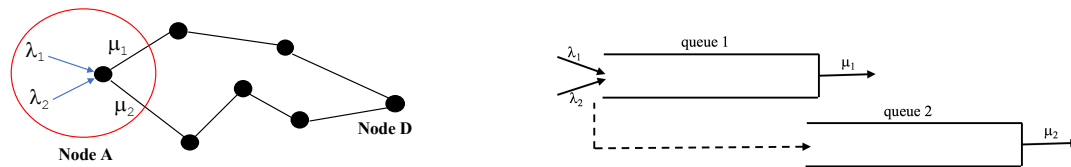
Keywords: wireless communication; performance optimization; markov decision process; energy efficiency; threshold routing; sensor networks

1. Introduction

Efficient resource utilization is a primary problem in resource constrained networks. In wireless sensor networks (WSNs) for instance the issue of energy efficiency is crucial to ensure network connectivity and quality of service. In WSNs, sensor nodes are generally deployed to transmit sensitive information in a timely manner. They rely on neighboring nodes to relay traffic to a given destination while operating on limited battery capacity. Energy is used when a node is listening, receiving, or transmitting. If a node's battery is depleted, neighboring nodes become incapable of relaying and transmitting urgent traffic through the node. More importantly, if said node belongs to the optimal path, a less efficient path will need to be computed which reduces network throughput and consumes more of the scarce resources. It is expected that some sensor networks will be deployed over large and inhospitable areas [1–6]. Since these networks may not be accessible following deployment, it is crucial that implemented admission and routing policies are resource efficient (i.e., consume minimal energy).

Consider a sensor node A with two available paths to the same destination as shown in Figure 1. We define a task as the transmission of a single flow from a relaying node to the final destination. We say that a task is successful if its corresponding flow is treated to the full extent - that is, all packets that belong to the same flow reach their final destination. Consider the scenario where sensor node A is tasked with relaying two flows. To maximize task success, it may be more efficient to treat one task to a full extent and reject or treat the second task partially than to partially treat both tasks. For instance,

suppose a node is sending information about two events to a control center (CC) simultaneously: an attack on a battlefield and a fire in another nearby region. The CC would prefer to receive full information about one flow and act on it, rather than receive partial information about both flows that would be discarded. Hence, sometimes rejecting a flow may be perceived as more beneficial than accepting a flow and partially transmitting it due to lack of resources. This saves energy consumption while transmitting a flow to the full extent.



(a) Network with two alternate paths from Node A to Node D.

(b) Node A modeled as a queuing system.

Figure 1. The queueing system at node *A* where type- $i, i \in \{1, 2\}$ arrivals join queue 1. Instead of being aborted from the system, packets are routed to queue 2; $\mu_2 \leq \mu_1$. Served at queue 1 or at queue 2, packets will reach the same destination D over alternate paths.

In this paper, we model node *A* (Figure 1) as a queueing system where accepted packets belong to two different flows. We assume that one of the flows has higher priority than the other. Once a packet of a given flow is accepted to the system, it joins queue 1 and it is guaranteed service to the full extent independent of its type. Instead of being rejected or preempted from service, packets have the option to be served at a slower server behind a second queue (queue 2). Hence, the packet is transmitted over a less rewarding path. Transmitted type- i packets are rewarded $r_i, i \in \{1, 2\}, r_1 > r_2 > 0$ if served at queue 1, and $r_3 \leq r_2$ if served at queue 2. Using the less rewarding path not only minimizes the number of rejected packets from the system, but also maximizes the chance that both flows will be treated to the full extent. Most importantly, allowing packets to be served at queue 2, allows the extension of the life of the path behind queue 1 (the efficient path).

We are interested in finding ways by which node *A* can, through its local decision policy, accept or reject and decide which path to choose to transmit packets. The objective is to minimize energy consumption and maximize the number of flows served, hence maximizing network throughput.

Such decision control mechanism is fundamental to a variety of other interesting applications. For example, consider the case where a node experiences a flood of traffic as a sign of it being compromised. A node can be subject to a SYN flood where an attacker attempts to fill the backlog queue of a victim machine's Transmission Control Protocol server (TCP) [7,8]. This results in resource depletion that renders the node unresponsive to legitimate traffic. By recognizing such flood of traffic, a node may either classify it as high priority traffic to identify the attacker and take the appropriate measures, or as low priority traffic and route it to another queue with a slower server.

This problem finds applications not only in computers and communication networks but in various other fields as well. Blockchain-based applications for instance, suffer from high computational and storage expenses, negatively impacting overall performance and scalability [9]. Therefore, work has been done to move computation and data off the chain (Off-chain). Off-chain transactions (i.e., high priority traffic) can be executed instantly and usually have low or no transaction fee. However, on-chain transactions (i.e., low priority traffic) can have a lengthy lag time depending on the network load and on the number of transactions waiting in the queue to be confirmed. Similarly, control of multi-class queueing systems has received significant attention in supply chain management and manufacturing systems ([10–12] and references therein). One of the main tools for such control problems is to characterize a performance measure of interest and use optimization methods to find the optimal control policy [13–15]. An algorithm for optimal pricing and admission control is proposed in [14,16].

In this paper, we develop a dynamic programming formulation of the Admission and Routing Control (ARC) problem, that maximizes the network throughput by extending the life (the resources) of the efficient path and thus the number of flows serviced to the full extent. We formulate the ARC problem as a Markov decision process (MDP) [17] and characterize the optimal policy under the Poisson traffic model. In particular, we show that the ARC policy that maximizes the expected reward is stationary and is a state-dependent threshold type policy. While dynamic programming can be used to solve such problems, the large size of the state space makes it untractable and too resource intensive to run on network devices and especially on wireless devices. Therefore, we propose a fast heuristic that exploits the structure of the optimal policy. Much of the computation required for our method can be done off-line, and the real-time computation requires no more than a table lookup. Furthermore, computing the parameters of the heuristic control policy is orders of magnitude faster than computing the optimal. We empirically show that the heuristic performs very well with an average reward deviation of 1.4% from the optimal while being orders of magnitude faster than the optimal policy. We further generalize the heuristic for the general case of a system with n ($n > 2$) types of flows. We believe our heuristic is general enough to be widely applicable and can be implemented in realtime. Although the method we propose applies to general network resource constraints, we consider energy limitations as our motivating application.

The rest of the paper is organized as follows. In Section 2, we provide a literature review. In Section 3, we describe the model and provide its mathematical formulation. In Section 4, we develop properties of the expected reward value function and characterize the optimal ARC policy. In Section 5, we analyze the behavior of the optimal policy through extensive numerical experiments. In Section 6, we propose a heuristic control policy, for the general case of n types of packets, and conduct extensive numerical experiments, for the case of 2 types of flows, in order to assess its performance compared to the optimal policy. In Section 7, we summarize our findings and propose future directions of this work.

2. Literature

In this section, we review the existing literature pertaining to resource constrained environments in WSNs. This is by no means exhaustive; it is only indicative of the interest and the applications.

Extensive work has been done to address the problem of energy saving with respect to WSNs. In [18–21], controlled mobility is used to extend the network lifetime. Algorithms for self-organization of sensor networks have been proposed [22,23] to minimize the risk of data loss during transmission and to maximize the battery life of individual sensors. Various coverage optimization protocols have been studied [24,25] where a number of sensor nodes are deployed to ensure adequate coverage of a region. Using a coverage optimization protocol, nodes with overlapping sensing areas are turned off to reduce energy consumption. We refer the readers to the following survey [26] of the various other energy efficient coverage techniques. Research in [19] focused on shortest path algorithms to optimize energy consumption. However, using the shortest path may lead to an increase in the ratio of lost packets [27]. In [28], research focused on scheduling sensor nodes to switch on and off, depending on the queue size, to reduce energy consumption. However, switching nodes from an idle to a busy state and vice versa has been a major portion of the power consumption [29].

Numerous works have also treated admission control policies relevant to WSNs. Several algorithms have been used by Internet routers to decide on packets admission and rejection to manage queues and minimize congestion in TCP [30]. In Tail Drop [31], when the queue reaches its maximum capacity, the newly arriving packets are rejected independent of their types. Though traffic may belong to different flows, it is not differentiated, and each packet is treated identically. When segments are lost, the TCP sender enters slow-start, which reduces throughput in that TCP session. A more severe problem occurs when segments from multiple TCP flows are dropped causing global synchronization - that is, all of the involved TCP senders enter slow-start. This problem can be mitigated by routing segments from one flow (the lower priority flow for instance) to a different queue, rather than discarding one segment from each flow. In Weighted Random Early

Detection (WRED) [32], flows are differentiated and treated depending on their type. Packets with a higher IP address precedence are less likely to be dropped than packets with a lower precedence. Thus, higher priority packets are delivered with a higher probability than lower priority packets. As in TCP, there is no guarantee that discarded packets belong to the same flow.

Queuing theory has also been widely used in network optimization [33–35]. In [36], a queueing network model was used to analyze and study the performance of a mobile WSN. In [37,38], models implementing admission control mechanisms to manage scarce radio resources in WSNs are described in the form of a queueing system with unreliable devices. Work in [39] proposed an energy saving mechanism that controls the ON/OFF state of a sensor node. A sensor node enters an OFF state (multiple fixed duration vacation periods) as soon as its queue is empty and is turned on (changes to an ON state) only when its queue size reaches a threshold value of packets. A $M/M/1$ -type queueing model with a control mechanism is proposed in [40] as a tool to reduce power consumption in WSNs. The tool switches a sensor node from an OFF mode to an ON mode only when its queue size reaches or exceeds a given size.

In order to achieve quality of service in a multi-class two parallel queue system, work in [41–43] dedicates a server to the high priority flow. In a resource constrained environment, such a model is not efficient. The dedicated server may be idle for a long time waiting, and consuming resources, for the high priority traffic to arrive while other servers may be congested. A different approach is considered in research that focuses on a single server and two classes of jobs. To minimize the sum of the holding/processing and switching costs, work in [44] switches serving between two classes of traffic. In [45] the authors consider a two-class single server preemptive priority queue where jobs can be denied admission to the system and can be aborted from service. Aborted jobs can rejoin the queue and resume service at a later time. Service does not have to restart but it continues from the step it was aborted from. In [46], an expulsion/scheduling control mechanism is proposed for a single class $M/M/2$ queueing system with non-identical servers. It was further extended in [47] to a multiple class system where job preemption is allowed.

The concept of termination control, studying a single server one-class workload model is introduced in [48]. In this work, the service of a job may be aborted before the job has received full service, and may be removed at any point in time from the queue. The authors further show that optimal threshold (acceptance and termination) policies exist.

A common characteristic of these methods is that they may find applications in areas such as workflow and assembly lines. However, they are not applicable to communication networks. One key element of communication networks is that once a packet transmission starts, it cannot be aborted and resume its service at a later time. A packet can be either fully transmitted or put back in the queue before its service starts. Moreover, in resource constrained networks, once a packet is queued or starts service, processing and computation resources are consumed. Preempting its service (transmission) and returning it to the queue for its service to restart at a later time, only consumes more of the scarce resources.

A recurrent assumption of existing work related to admission control is that low priority packets are denied admission to the queue when higher priority packets are already present. In the event they are accepted, low priority packets can be at any time rejected from the queue, or aborted from service in favor of a higher priority packet [45]. Another option in multiserver systems is to restrict sending the high priority flow over one of the available paths (generally the optimal one). These approaches may work well in certain applications such as delay tolerant networks and networks with unlimited resources. They are also applicable in general applications such as in workflow and in assembly lines. However, in resource constrained computer networks, once a packet is accepted to the system, resource (energy, computation, memory) consumption starts.

3. Model Description and Formulation

3.1. Model Description

We consider the system given in Figure 1. We model node A as a two-class queueing system and assume that type- i , $i \in \{1, 2\}$, packets arrive at node A according to independent Poisson process with arrival rate $\lambda_i \geq 0$, respectively. We assume that node A has two paths that lead to the destination D : an energy efficient path, through queue 1, and a less energy efficient path through queue 2. We assume that the service at queue 1 and at queue 2 are exponentially distributed with rates μ_1 and μ_2 respectively, where $\mu_1 \geq \mu_2$. We assume that type-1 packets have higher priority than type-2 packets hence, they are always accepted to queue 1 upon arrival. However, an admission control mechanism at node A decides to accept or reject type-2 packets. Arrivals (type-1 and accepted type-2 packets) join queue 1. At any event, arrival or service completion, the decision maker can decide to route a packet to queue 2 or to serve it at queue 1. This decision is made to give service advantage to the higher priority packets to be served at queue 1 and to be transmitted over the more energy efficient path. In summary, the system is controlled in 3-ways: (1) all type-1 packets are always accepted to the system, (2) an admission policy at node A decides to accept or reject a newly arriving type-2 packet, (3) a routing policy at node A decides to route type- i , $i \in \{1, 2\}$ packet to queue 2 or to serve it at queue 1. However, type-1 packets are always given service priority at queue 1.

We assume that the decision maker has complete state information, i.e., it knows the instantaneous number of packets of type- i , $i \in \{1, 2\}$ in queue 1 and the total number of packets in queue 2. Thus, the structure of the system is that of a Markovian decision process [49]. As such, we propose to formulate the admission and routing problem as a MDP and use the value iteration technique to characterize the form of the optimal policy. We formulate the problem by defining the states, the transition structure and the feasible actions.

State. The state of the system is described by the vector $\mathbf{x} = (x_1, x_2, x_3)$ where x_i , $i \in \{1, 2\}$ is the number of type- i packets in queue 1 and x_3 is the number of packets (type-1 and/or type-2) in queue 2.

Events. We distinguish three possible events: (1) the arrival of a new packet, (2) the service completion at queue 1, and (3) the service completion at queue 2.

Decisions. If the event is an arrival, then if the packet is a type-1, it is automatically accepted to queue 1 and this changes the state (x_1, x_2, x_3) into state $(x_1 + 1, x_2, x_3)$. If alternatively, it is a type-2 packet, then a decision has to be made to accept or reject the newly arrived packet. If the packet is accepted, then this changes the state (x_1, x_2, x_3) to $(x_1, x_2 + 1, x_3)$. If it is rejected, the state does not change. Next, the decision is either to serve a packet in queue 1 or to route it to queue 2. The idea here is, if there are type-1 packets in queue 1, they are given higher service priority at queue 1. Hence type-2 packets will be served at queue 1 only if there are no type-1 packets in queue 1.

One may argue, since there are two queues why not dedicate queue 1 to type-1 packets and queue 2 to type-2 packets? The main reason for not using this approach is so that type-2 packets will not be deprived from using the more efficient path when there are no type-1 packets in queue 1. This also allows type-1 packets to be served at queue 2 in the event queue 1 is heavily congested or when the server at queue 2 is idle.

Costs and rewards. Packets earn a reward upon service completion. Packets served at queue 1 receive a type-dependent reward r_i , $i \in \{1, 2\}$. Since type-1 packets have a higher priority than type-2 packets they receive a higher reward $r_1 \geq r_2 > 0$. Packets served at queue 2 receive a reward r_3 ($0 < r_3 < r_2$) independent of their type.

Packets admitted to the system are also subject to a holding and processing cost h , incurred while waiting in the queue or while being served. We assume these costs are linear in the number of packets in the system and are type independent namely, $yh \geq 0$ per unit of time when there are $y = \sum_{i=1}^3 x_i$ packets present in the queues. In addition, each time a packet is admitted to the system, type independent admission cost $c \geq 0$ is incurred. Rejecting packets is free of charge. Moreover, routing type-2 packets to queue 2 is free of charge while routing type-1 has a positive switching cost c_2 . Imposing a positive

switching cost on type-1 packets is intended to discourage these packets from being routed to queue 2, and use the less efficient path especially, when there are type-2 packets in queue 1.

For a reward to be collected and for the model to make sense, the cost incurred by a type- i packet served at queue i must be smaller than the reward it collects at the queue. When served at queue 1 a packet reward is $r_i > c + h/\mu_1$. When served at queue 2 a packet reward is $r_3 > c + h/\mu_2 + c_2 I_{\{i=1\}}$, $i \in \{1, 2\}$ where the indicator $I_{\{i=1\}} = 1$ if the packet is type-1 otherwise, it is equal to zero. As our application is related to energy optimization in sensor networks, we assume that all the costs and rewards are in units of energy. The cost h can be interpreted as the energy consumed to process and maintain a packet in the queue. The cost c is the energy consumed to receive a packet and c_2 is the energy consumed to switch or move a packet from queue 1 to queue 2. Rewards can be interpreted as the energy saved by successfully transmitting a packet compared to rejecting it.

Criterion. The objective is to maximize the expected discounted reward resulting from accepting, routing and servicing flows to completion over an infinite horizon.

Uniformization. In order to convert the continuous problem into a discrete one, we follow [50]'s uniformization technique. We adjust the transition rates of the embedded Markov chain of the system so that the transition times between decision times is a sequence of independent exponentially distributed random variables with mean $\frac{1}{\beta}$, where $\beta = \alpha + \lambda_1 + \lambda_2 + \mu_1 + \mu_2$. Then with probability $\lambda_i/\beta > 0, i = 1, 2$, a transition concerns the arrival of a type- i packet, with probability $\mu_j/\beta > 0, j = 1, 2$ concerns a service completion at queue j and with probability $\alpha/\beta > 0$, the process terminates. Without loss of generality, we scale the time line so that the rate $\beta = 1$.

Discounting. We discount future rewards at a rate $\alpha \geq 0$, (i.e., rewards at time t are multiplied by $e^{(\alpha t)}$). This is equivalent to a process that lasts an exponentially distributed time with mean $1/\alpha$ after which, there will be no more arrivals or service completions.

Note that node A in Figure 1, can be modeled as a single-shared-queue system. In this case, a routing decision can be made only when a packet reaches the head of the queue, leading to the head-of-the-line blocking (HOL) problem [51]. As such, single shared queue devices are perceived to have low performance due to the HOL blocking [52,53]. This is the main reason why network devices generally use separate queues per output port. In this work, modeling node A as a two-queue system mitigates the HOL problem, especially that the routing decision is made not only right before service but also at the arrival of a packet.

3.2. Model Formulation

In the following, we summarize and complete the model in terms of a mathematical formulation. Let $w_n(\mathbf{x})$ be the expected discounted reward of responding to an event given that the system has reached state \mathbf{x} following n state transitions starting from a randomly chosen initial state (i.e., $w_0(\mathbf{x}) = 0$ for all $\mathbf{x} \geq \mathbf{0}$, where $\mathbf{0}$ is the zero vector of dimension 3 and the inequality $\mathbf{x} \geq \mathbf{0}$ is taken component-wise).

Admission: Let $\mathcal{T}_{a_i} w_{n-1}(\mathbf{x})$ denote the expected discounted reward when an arrival of type- i packet event occurs and the system is in state \mathbf{x} . Let e_k be the k -th unit vector of dimension 3. An arrival of type-2 is accepted to the system only if the difference in reward between accepting the packet and rejecting it is positive i.e., $w_n(\mathbf{x} + e_2) - c \geq w_n(\mathbf{x})$. Recall that type-1 packets are never rejected. Thus, for $\mathbf{x} \geq \mathbf{0}$,

$$\begin{aligned}\mathcal{T}_{a_1} w_{n-1}(\mathbf{x}) &= w_{n-1}(\mathbf{x} + e_1) - c \\ \mathcal{T}_{a_2} w_{n-1}(\mathbf{x}) &= \max\{w_{n-1}(\mathbf{x} + e_2) - c, w_{n-1}(\mathbf{x})\}\end{aligned}$$

Service: When the system is in state \mathbf{x} , a service decision of a packet at queue 1 is made as follows. If it is a type-1 packet, it proceeds with no delay to service at server 1. If it is type-2 packet, it is served at server 1 only if there are no type-1 packets in queue 1. In queue 2, packets are served

on a first-come-first-serve independent of their type. We define the service operators \mathcal{T}_i at queue i , $i \in \{1, 2\}$ as follows:

$$\mathcal{T}_1 w_{n-1}(\mathbf{x}) = \begin{cases} w_{n-1}(\mathbf{x}) & \text{if } x_1 = x_2 = 0 \\ w_{n-1}(\mathbf{x} - e_1) + r_1 & \text{if } x_1 > 0 \\ w_{n-1}(\mathbf{x} - e_2) + r_2 & \text{if } x_1 = 0 \text{ and } x_2 > 0 \end{cases}$$

and

$$\mathcal{T}_2 w_{n-1}(\mathbf{x}) = \begin{cases} w_{n-1}(\mathbf{x}) & \text{if } x_3 = 0 \\ w_{n-1}(\mathbf{x} - e_3) + r_3 & \text{if } x_3 > 0 \end{cases}$$

Let $\mathcal{T}_s w_{n-1}(\mathbf{x})$ denote the expected discounted reward when the current state is \mathbf{x} and an arrival or a service completion event occurs. Note that $\mathcal{T}_s w_{n-1}(\mathbf{x})$, given by Equation (1), represents the expected reward assuming no routing to queue 2 occurred.

$$\mathcal{T}_s w_{n-1}(\mathbf{x}) = \sum_{i=1}^2 (\lambda_i \mathcal{T}_{a_i} w_{n-1}(\mathbf{x}) + \mu_i \mathcal{T}_i w_{n-1}(\mathbf{x})) - h \sum_{j=1}^3 x_j \quad (1)$$

Routing: A type-1 packet may be routed to queue 2 only if no type-2 packets are in queue 1 ($x_2 = 0$) and it is more rewarding to route the packet to queue 2 than to keep it in queue 1. However, type-2 packets can be routed to queue 2 when $x_2 > 0$ and when it is more rewarding to do so. Let $\mathcal{T}_r w_n(\mathbf{x})$ denote the expected discounted reward when the current state is \mathbf{x} and a routing decision to queue 2 is to occur.

$$\mathcal{T}_r w_{n-1}(\mathbf{x}) = \begin{cases} \max\{w_{n-1}(\mathbf{x} - e_1 + e_3) - c_2, w_{n-1}(\mathbf{x})\} & \text{if } x_2 = 0, x_1 > 0 \\ \max\{w_{n-1}(\mathbf{x} - e_2 + e_3), w_{n-1}(\mathbf{x})\} & \text{if } x_2 > 0 \end{cases} \quad (2)$$

The optimal expected discounted reward at state \mathbf{x} is given by Equation (3). It is implied that at any event, arrival or service completion, the decision maker can decide to route a packet to queue 2 or serve it at queue 1.

$$w_n(\mathbf{x}) = \max(\mathcal{T}_s w_{n-1}(\mathbf{x}), \mathcal{T}_r w_{n-1}(\mathbf{x})) \quad \text{for } x_1 + x_2 > 0 \quad (3)$$

4. Characterization of the Optimal Arc Policy

To characterize the optimal policy, we use the value iteration technique introduced in [54,55], by recursively evaluating w_n using Equation (3) for $n \geq 0$. We prove by induction that if some structural properties of the discounted reward function w_n are satisfied, then these properties are also satisfied for w_{n+1} and therefore, they hold for all $n \geq 0$. As n tends to infinity, the optimal policy converges to the unique optimal policy. This convergence result is ensured by Theorem 8.10.1 in [17]. The convergence to the optimal policy is an important result in the MDP literature. It is based on showing that the iteration from w_n to w_{n+1} is a contraction mapping as stated in Theorem 6.2.3 in [17]. This Theorem also proves that the optimal infinite horizon policy is independent of the choice of w_0 and this is why one can simply choose $w_0(\mathbf{x}) = 0$.

4.1. Reward Function Properties

Solving the optimality Equation (3) analytically is untractable. Hence, in order to characterize the structure of the optimal policy, we show that the optimal reward function satisfies a set of properties which allow us to infer the structure of the optimal policy. The properties are listed and interpreted below.

Property 1. $w_n(\mathbf{x} + e_i) - w_n(\mathbf{x}) \geq w_n(\mathbf{x} + 2e_i) - w_n(\mathbf{x} + e_i)$, for $i \in \{1, 2, 3\}$.

Property 1 implies that $w_n(\mathbf{x})$ is concave in each of the state variables x_i . In other words, it implies that the marginal reward (i.e., $w_n(\mathbf{x} + e_i) - w_n(\mathbf{x})$) of an additional packet of type- i , $i \in \{1, 2\}$ in queue 1 is non-increasing in the number of packets x_i for a fixed $x_j, j \neq i$ and fixed number of packets, x_3 in queue 2. It also implies that the marginal reward of an additional packet in queue 2 is non-increasing in the number of packets x_3 for a fixed number of packets of type- i , $i \in \{1, 2\}$ in queue 1.

Property 2. $w_n(\mathbf{x} + e_i + e_j) - w_n(\mathbf{x} + e_j) \geq w_n(\mathbf{x} + 2e_i) - w_n(\mathbf{x} + e_i)$, for $i \neq j$ and $i, j \in \{1, 2, 3\}$.

Property 2, for $i = 2$ and $j = 3$, states that the marginal reward of an additional type-2 packet in queue 1 is non-increasing in x_2 . Therefore, routing a type-2 packet to queue 2 is less rewarding than servicing it at queue 1. Similarly, Property 2, when $i = 1$ and $j = 3$, states that the marginal reward of an additional type-1 packet in queue 1 is non-increasing in x_1 . Consequently, routing a type-1 packet to queue 2 is less rewarding than servicing it at queue 1. The other cases have similar interpretations.

Property 3. $w_n(\mathbf{x} + e_j) - w_n(\mathbf{x}) \geq w_n(\mathbf{x} + e_i + e_j) - w_n(\mathbf{x} + e_i)$, for $i \neq j$ and $i, j \in \{1, 2, 3\}$.

Property 3 states that the marginal reward of an additional type- i packet is non-increasing in x_j , $i, j \in \{1, 2\}$ and $i \neq j$ for fixed x_i . Similarly, the marginal value of an additional packet in queue 1 is non-increasing in $x_j, j \in \{1, 2\}$ for fixed queue 2 size. Mathematically, Property 3 indicates that the reward value function is sub-modular.

4.2. Reward Function Bounds

Since all accepted packets are guaranteed to be served at either queue, packets will collect a reward upon service completion as long as $r_i > c + h/\mu_1$ and $r_3 > c + h/\mu_2 + c_2 I_{\{i=1\}}, i \in \{1, 2\}$. However, since the reward depends on the packet type and on the queue where the packet resides, in this subsection, we bound the reward collected by packet type. We make use of sample path approach [56] to prove the following propositions.

Proposition 1. For all $n \geq 0$ and $\mathbf{x} \geq 0$, the difference in reward of serving a type-2 packet at queue 1 does not exceed r_2 .

$$w_n(\mathbf{x} + e_2) - w_n(\mathbf{x}) \leq r_2, x_1 = 0$$

Proof. Using a sample path analysis, let two instances Π_1 and Π_2 of the policy where Π_1 starts at state $\mathbf{x} + e_2$ and Π_2 starts at state \mathbf{x} . Π_1 will follow the actions of the optimal policy and Π_2 will copy the actions of Π_1 . An arrival to both instances changes the rewards equally (every arrival is charged a cost of c). In the event of a departure from state $\mathbf{x} + e_2$, due to service completion at queue 1, in this case, we must have $x_1 = 0$, otherwise type-1 takes priority in service, immediately afterwards Π_2 and Π_1 become identical, so a reward of r_2 is generated. The departure can also be a route to queue 2. In this case, since there is no switching cost for type-2 packets, the reward does not change. \square

Proposition 2. For all $n \geq 0$ and $\mathbf{x} \geq 0$, the difference in reward of serving a packet at queue 2 does not exceed r_3 .

$$w_n(\mathbf{x} + e_3) - w_n(\mathbf{x}) \leq r_3$$

Proof. Using a sample path analysis, let two instances Π_1 and Π_2 of the policy where Π_1 starts at state $\mathbf{x} + e_3$ and Π_2 starts at state \mathbf{x} . Π_1 will follow the actions of the optimal policy and Π_2 will copy the actions of Π_1 . A departure from both instances changes the rewards equally (every departure is

rewarded r_3 independent of the packet type). Hence, upon a departure from queue 2 at state $\mathbf{x} + e_3$, Π_2 and Π_1 become identical so the difference in reward is at most r_3 . \square

Proposition 3. For all $n \geq 0$ and $\mathbf{x} \geq 0$, the difference in reward to serve a type-1 packet at queue 1 does not exceed r_1 .

$$w_n(\mathbf{x} + e_1) - w_n(\mathbf{x}) \leq r_1$$

Proof. Using a sample path analysis, let two instances of the policy where one (Π_1) starts at state $\mathbf{x} + e_1$ and the other (Π_2) starts at state \mathbf{x} . Π_1 will follow the actions of the optimal policy and Π_2 will copy the actions of Π_1 . An arrival to both instances changes the rewards equally (every arrival is charged a cost of c). In the event of a departure from state $\mathbf{x} + e_1$, due to service completion at queue 1 (immediately afterwards Π_2 and Π_1 become identical), a reward of r_1 is generated (since type-1 takes priority over type-2, the departure will be of type-1 unless $x_1 = 0$). So the difference in reward is at most r_1 . The departure can also be a route to queue 2. Note that a routing in both instances changes the reward equally by the switching cost of $c_2 < r_1$. \square

Proposition 4. For all $n \geq 0$ and $\mathbf{x} \geq 0$, the difference in reward of serving a packet at queue 1 and at queue 2 is larger than $r_i - r_3, i \in \{1, 2\}$

$$w_n(\mathbf{x} + e_i) - w_n(\mathbf{x} + e_3) \geq r_i - r_3, \quad i \in \{1, 2\}$$

Proof. Using a sample path analysis, we first consider the case where $i = 1$ and prove $w_n(\mathbf{x} + e_1) - w_n(\mathbf{x} + e_3) \geq r_1 - r_3$. Let two instances Π_1 and Π_2 of the policy where Π_1 starts at state $\mathbf{x} + e_3$ and instance Π_2 starts at state $\mathbf{x} + e_1$. Instance Π_1 will follow the optimal policy and instance Π_2 will copy the actions of Π_1 . That is, if Π_1 routes its packet, then Π_2 routes its packets, and if Π_1 takes its packet into service, then Π_2 takes its packet into service. For both instances, while packets are still in the system, their costs and rewards are the same. Hence, the difference in reward between the two instances is zero. However, if a packet is served, a reward of r_3 in Π_1 is collected and a reward of r_1 in Π_2 is collected. Hence, the difference in reward between the two instances is $r_1 - r_3 > 0$.

The proof of the case where $i = 2$, is very similar to the above proof. It suffices to replace e_1 by e_2 and r_1 by r_2 . \square

Note that in this work, the admission cost is not as relevant as the reward as it is packet type-independent. However, the problem can be easily generalized to assigning type-dependent costs $c_{a_i} > 0, i \in \{1, 2\}, (c_{a_1} \neq c_{a_2})$. On the other hand, the switching cost c_2 is important for type-1 packets as they are charged only in the event they are routed to queue 2.

We conclude this section with the main results of the paper as illustrated in the following Theorem.

Theorem 1. There exists a stationary optimal policy for any initial state $\mathbf{x} = (x_1, x_2, x_3)$ such that:

- **Admission policy:** The optimal admission control policy is a state-dependent threshold-type, with threshold curve $\mathcal{A}(x_1, x_3)$, such that a type-2 packet is admitted to queue 1 if and only if $x_2 \leq \mathcal{A}(x_1, x_3)$ where $\mathcal{A}(x_1, x_3) = \max\{x_2 | w_n(\mathbf{x} + e_2) - w_n(\mathbf{x}) \geq c\}$.
- **Routing policy:** The optimal routing control policy is a state-dependent threshold-type, with threshold curves $\mathcal{R}_1(x_3)$ and $\mathcal{R}_2(x_1, x_3)$, such that:
 1. Type-1 packet is routed to queue 2 if and only if $x_2 = 0$ and $x_1 \geq \mathcal{R}_1(x_3)$, where $\mathcal{R}_1(x_3) = \max\{x_1 | w_n(\mathbf{x} - e_1 + e_3) - w_n(\mathbf{x}) \geq c_2\}$.
 2. Type-2 packet is routed to queue 2 if and only if $x_2 \geq \mathcal{R}_2(x_1, x_3)$, where $\mathcal{R}_2(x_1, x_3) = \max\{x_2 | w_n(\mathbf{x} - e_2 + e_3) - w_n(\mathbf{x}) \geq 0\}$.

The results of Theorem 1 also apply to the average reward criterion (see [57]). Hence, we will use the average reward criterion for all our numerical experiments as it has the advantage of not depending on the initial state. To prove the theorem, we will prove Properties 1–3. However, for ease of flow, we defer all mathematical proofs to the Appendix A.

5. Sensitivity Analysis of the Optimal Policy

In this section, we study the optimal control policy depicted in Theorem 1 and its sensitivity to the system parameters. We conduct extensive numerical experiments varying system parameters. As an illustration we consider a base case, where $\mu_1 = 0.35, \mu_2 = 0.45, \lambda_1 = 0.25, \lambda_2 = 0.25, r_1 = 80, r_2 = 60, r_3 = 30, h = 0.7, c = 5, c_2 = 0$. The optimal policy is computed using the value iteration algorithm of dynamic programming [58]. Convergence is obtained when the expected reward of successive iterations is within an accuracy of 10^{-5} . The optimal admission control policy for this system is presented in Figure 2a. The optimal action is to reject type-2 packets in all states above the switching curve (above the red line) and to accept them in all states below the curve. Similarly, the optimal routing policy for the system is presented in Figure 2b. The optimal action is to route type-2 packets to queue 2 only in states above the switching curve (above the blue and red lines). In states below the switch-curve (below the blue line), no routing is allowed. Below the red line are the only states where type-1 packets are routed to queue 2 that is, when $x_2 = 0$. We experimented with several system parameters and we obtained the same results in terms of the shape of the switching curves of the admission control policy and routing control policy.

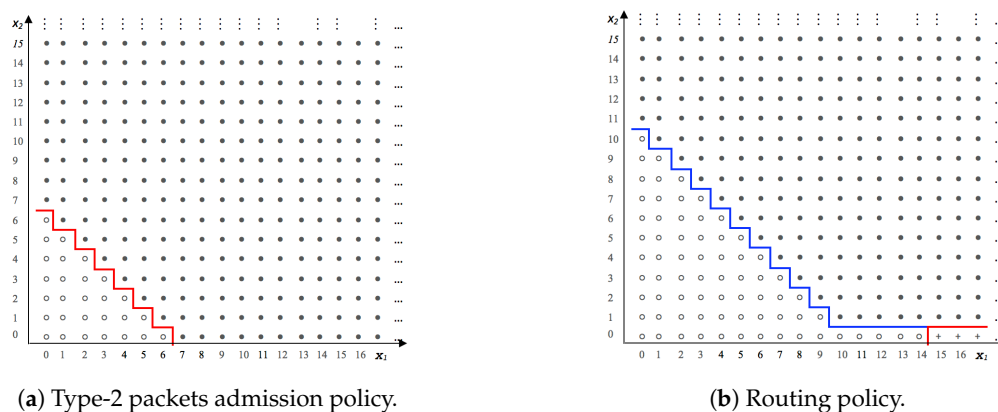


Figure 2. Optimal control policy.

In Figure 3, we superpose both control policy curves where we note that the system gives priority to type-1 packets to be served at queue 1 by routing excess type-2 packets to queue 2. All numerical results gave a straight-line switching curve with slope of -1 in the (x_1, x_2) plane for a given x_3 packets in queue 2. However, proving this result analytically is untractable as it amounts to solving the optimality Equation (3) in closed form.

We further study the effect of the system parameters on the optimal average reward for various network load values, $\rho \in \{50\%, 75\%, 95\%\}$. We isolate the effect of a particular system parameter by varying its value while holding the values of other system parameters constant.

We study the effect of increasing reward r_2 while maintaining the sum of the rewards of r_1 and r_2 constant. Figure 4a shows that the optimal average reward decreases nonlinearly as the ratio r_2/r_1 increases. This behavior can be explained as follows. As r_2 increases, the incentive for packets to be routed to queue 2 decreases. Hence, queue 1 becomes overloaded and the overall holding cost eventually becomes high affecting the optimal average reward. Note however, that under heavy network load ($\rho = 95\%$), at a certain point, routing to queue 2 becomes inevitable causing lower reward compared to a system with lower network load ($\rho = 75\%$). This also explains the crossover of

the curves corresponding to the optimal average reward curve for $\rho = 75\%$ and the one for $\rho = 95\%$ in the figure.

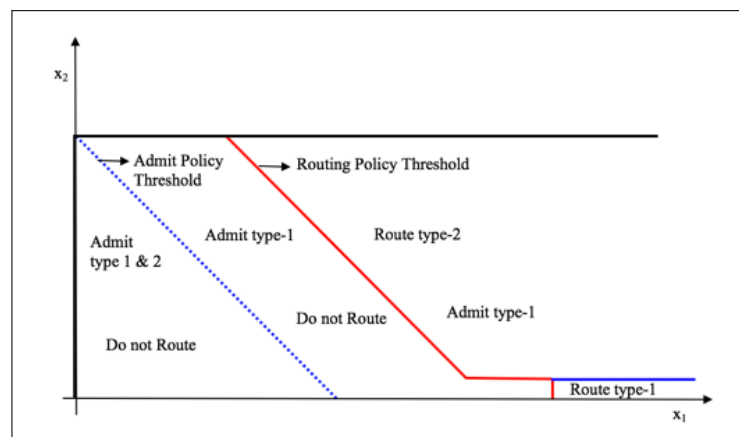
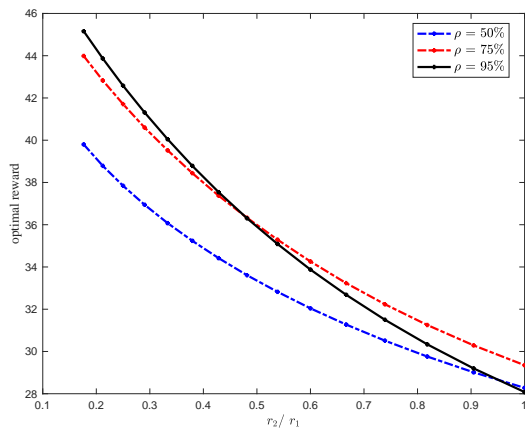


Figure 3. Admission and routing policy transient states for a given size of queue 2.

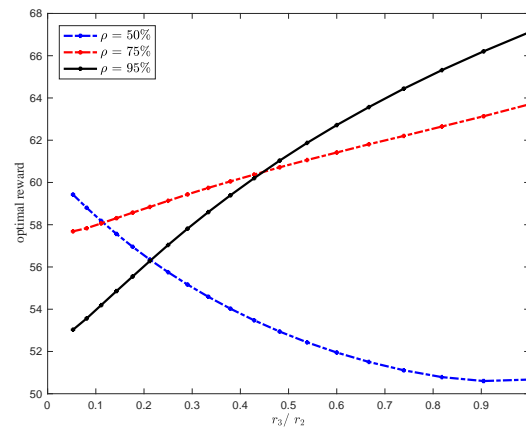
We further study the effect of increasing reward r_3 on the optimal average reward while maintaining the sum of the rewards of r_2 and r_3 constant. Figure 4b shows that the optimal average reward increases non-linearly as the ratio r_3/r_2 increases when the network load is high ($\rho = 75\%$ and $\rho = 95\%$). This increase is due to the fact that as r_3 increases, packets in queue 1 have more incentive to be routed to queue 2 especially if queue 1 has type-1 packets. As type-2 packets are routed to queue 2, more space opens up in queue 1 for type-1 packets, and more type-2 packets are admitted, hence, the optimal average reward increases. Moreover, as the network load increases, routing admitted packets to queue 2 becomes sometimes necessary. Indeed, for higher network load ($\rho = 95\%$), the optimal average reward is increasing at a faster rate compared to the optimal average reward for a network with lower load of $\rho = 75\%$. For lower network load ($\rho = 50\%$) however, as r_3 increases the optimal average reward decreases nonlinearly. This can be explained as follows: as r_3 increases and if type-1 packets are in queue 1, then type-2 packets have more incentive to be routed to queue 2, and collect a lower reward hence, the optimal average reward decreases. The graphs generated in Figure 4a,b show results when $\mu_1 = 0.45, \mu_2 = 0.35, h = 0.7, c = 5, c_2 = 0$.

Figure 4c shows the sensitivity of the optimal average reward to the arrival rates while maintaining the sum of λ_1 and λ_2 constant. The figure shows that the optimal average reward initially increases at a high rate as λ_1 increases then the rate of increase slows down. The increase continues until queue 1 becomes congested to cause type-1 packets to be routed to queue 2, hence, collect a lower reward r_3 , and decrease the optimal average reward. For high network load ($\rho = 95\%$) however, the optimal average reward eventually starts decreasing as the high load makes it necessary to increase routing packets to queue 2. Eventually, both queues saturate, leading to an increase in the holding cost, and a decrease in the optimal average reward. The graph generated in Figure 4c shows results for the following system parameters: $r_1 = 80, r_2 = 60, r_3 = 30, \mu_1 = 0.45, \mu_2 = 0.3, h = 0.7, c = 5$ and $c_2 = 0$.

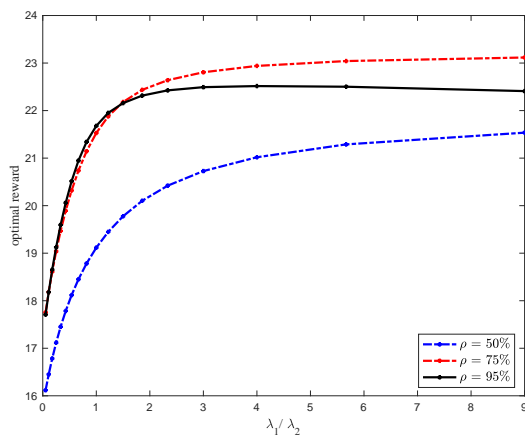
In Figure 4d, we study the effect of the service rates μ_1 and μ_2 on the optimal average reward while maintaining the sum of μ_1 and μ_2 constant. As the ratio μ_1/μ_2 increases, the optimal average reward increases and eventually levels-off for all network loads considered ($\rho \in \{50\%, 75\%, 95\%\}$). This can be explained as follows. As queue 1 service rate μ_1 increases, queue 1's length becomes shorter discouraging packets from being routed to queue 2. Hence, the optimal average reward increases. As μ_1 continues to increase relative to μ_2 , less and less routing occurs eliminating the need for queue 2 which explains the leveling-off of the optimal average reward (since the arrival rates are held constant). In practice, however, achieving a high service rate to eliminate queue 2 is rather costly. The graph generated in Figure 4d shows results for the following system parameters: $r_1 = 80, r_2 = 60, r_3 = 30, \lambda_1 = 0.35, \lambda_2 = 0.3, h = 0.7, c = 5$ and $c_2 = 0$.



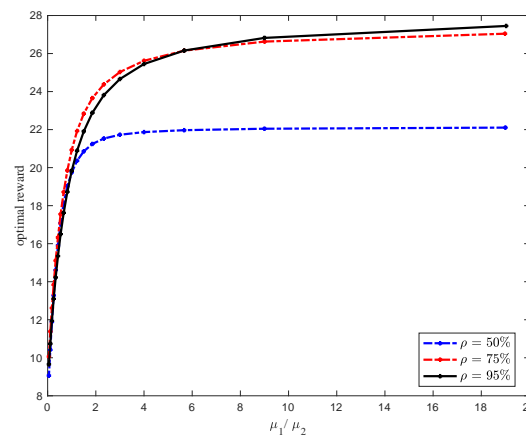
(a) Optimal reward sensitivity to r_2/r_1 .



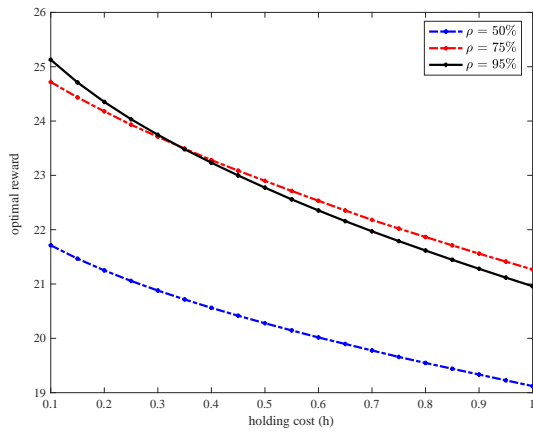
(b) Optimal reward sensitivity to r_3/r_2 .



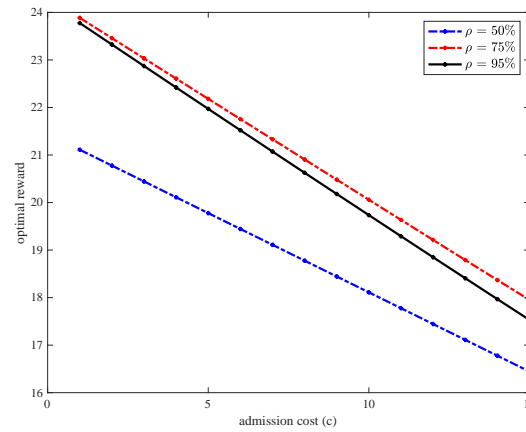
(c) Optimal reward sensitivity to λ_1/λ_2 .



(d) Optimal reward sensitivity to μ_1/μ_2 .



(e) Optimal reward sensitivity to h .



(f) Optimal reward sensitivity to c .

Figure 4. Optimal reward sensitivity to rewards r_1, r_2, r_3 , service rates μ_1, μ_2 , arrival rates λ_1, λ_2 , holding cost h and admission cost c .

Figure 4e shows that the optimal average reward is nonlinearly decreasing in the holding cost h . Figure 4f shows that the optimal average reward is linearly decreasing in the admission cost c . These results are interesting and are worth exploiting in a future work in an attempt to get an analytical expression of the reward. The graphs generated in Figure 4e,f show results for the following

system parameters: $r_1 = 80, r_2 = 60, r_3 = 30, \lambda_1 = 0.35, \lambda_2 = 0.3, c_2 = 0$ and various values of h and c respectively.

Finally, we would like to note that even though our analysis focused on the case where the switching cost is zero ($c_2 = 0$), similar results are obtained for $c_2 > 0$. As an illustration, Figure 5, shows that the optimal reward linearly decreases in c_2 . This result is expected as when the switching cost increases, type-1 packets have no incentive to be routed to queue 2. Moreover, as the network load increases, the optimal reward increases up to a point where both queues become congested. This explains why the optimal reward when $\rho = 75\%$ is higher than the optimal reward when $\rho = 95\%$ as c_2 increases. This also lead to the conclusion that there is an optimal load where the reward in maximized. The results in Figure 5 are obtained for system parameters: $r_1 = 80, r_2 = 60, r_3 = 30, \lambda_1 = 0.35, \lambda_2 = 0.3, h = 0.7$ and $c = 5$.

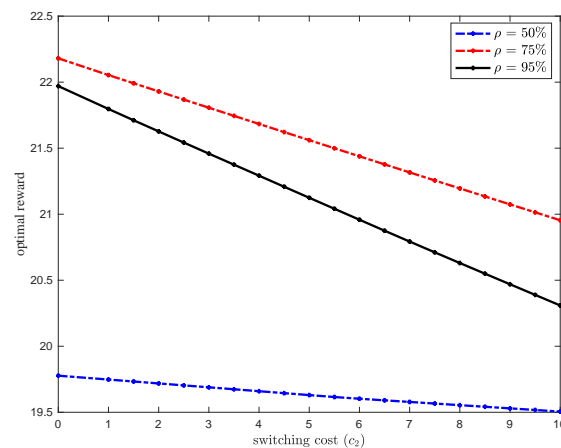


Figure 5. Optimal reward sensitivity to c_2 .

6. Heuristic Control Policy

It is well established that dynamic programming suffers from the curse of dimensionality. For our model in particular, the optimal policy is computationally untractable for systems with more than 2 types of packets (i.e., a state space with dimension greater than 3). Hence, it is too resource intensive to run on resource limited sensor devices. This motivated us to propose an efficient heuristic control policy that imitates the behavior of the optimal policy and is computationally much faster to obtain for the general case of n types of packets (see Algorithm 1). As such, we define the state of the system by the $(n + 1)$ dimensional vector $\mathbf{x} = (x_1, x_2, \dots, x_n, x_{n+1})$ where type- i packets take priority over type- j packets for $i < j$. The number of packets in queue 1 is represented by $x_1 + x_2 + \dots + x_n$ where x_i is the number of type- i packets while x_{n+1} depicts the number of packets in queue 2. The heuristic control policy is characterized by $2(n - 1)$ parameters: parameters $\mathcal{A}_i, i = 2, \dots, n$ control the admission to queue 1, while parameters $\mathcal{R}_i, i = 2, \dots, n$ control the routing to queue 2. Note that type-1 packets are always admitted to queue 1 (i.e., $\mathcal{A}_1 = \infty$). Furthermore, we have $\mathcal{A}_2 \geq \mathcal{A}_3 \geq \dots \geq \mathcal{A}_n$ and $\mathcal{R}_1 \geq \mathcal{R}_2 \geq \dots \geq \mathcal{R}_n \geq \mathcal{A}_2$. We extend the costs and reward parameters as follows: c is the admission cost; h is the holding cost; c_i is the switching cost for type- i packets ($c_1 \geq c_2 \geq \dots \geq c_n$) and $r_i, i \in \{1, 2, \dots, n\}$ is the reward of type- i packet served at queue 1 ($r_1 \geq r_2 \geq \dots \geq r_n$) and $r_{n+1} (< r_n)$ is the reward of packets served at queue 2.

At arrival of type- i packet, we use the following control policy where we define $\mathcal{I}(x) =$ the largest packet type $i \in \{1, \dots, n\}$ such that $x_i > 0$:

Algorithm 1. Proposed heuristic control policy.

```

if  $\sum_{k=1}^n x_k < \mathcal{A}_i$  then
  admit type- $i$  packet to queue 1
end if
if  $\sum_{k=1}^n x_k \geq \mathcal{A}_i$  and  $\mathcal{I}(x) \geq i$  and  $\sum_{k=1}^n x_k \geq \mathcal{R}_{\mathcal{I}(x)}$  then
  admit type- $i$  packet to queue 1 and route type- $\mathcal{I}(x)$  packet to queue 2
else
  do no admit type- $i$  packet
end if

```

In order to test the performance of the above proposed heuristic, we compare the reward generated by the heuristic to that of the optimal policy. We use the average reward criterion for this purpose. The average reward under the optimal policy is obtained using the following optimality equation:

$$w^*(\mathbf{x}) + g^* = \max\{\mathcal{T}_s w(\mathbf{x}), \mathcal{T}_r w(\mathbf{x})\} \quad (4)$$

where g^* is the optimal average reward per transition (see [58]) and $w^*(\mathbf{x})$ is the optimal differential reward, $w(\mathbf{x})$, \mathcal{T}_s and \mathcal{T}_r as defined in Section 3.2.

The average reward under the heuristic control policy (H) is defined using the following dynamic programming equation:

$$w^H(\mathbf{x}) + g^H = -h \sum_{k=1}^{n+1} x_k + \sum_{k=1}^n \lambda_i \mathcal{T}_{a_i}^H w(\mathbf{x}) + \mu_1 \mathcal{T}_1^H w(\mathbf{x}) + \mu_2 \mathcal{T}_2^H w(\mathbf{x}). \quad (5)$$

where g^H is the average reward per transition under the heuristic control policy, $w^H(\mathbf{x})$ is the differential reward under the heuristic policy and $\mathcal{T}_{a_i}^H$, \mathcal{T}_1^H and \mathcal{T}_2^H are defined as follows:

$$\mathcal{T}_{a_i}^H w(\mathbf{x}) = \begin{cases} w(\mathbf{x} + e_i) - c & \text{if } \sum_{k=1}^n x_k < \mathcal{A}_i \\ w(\mathbf{x} + e_i - e_{\mathcal{I}(\mathbf{x})} + e_{n+1}) - c_{\mathcal{I}(\mathbf{x})} & \text{if } \sum_{k=1}^n x_k \geq \mathcal{A}_i, \mathcal{I}(\mathbf{x}) \geq i \text{ and } \sum_{k=1}^n x_k \geq \mathcal{R}_{\mathcal{I}(\mathbf{x})} \\ w(\mathbf{x}) & \text{otherwise} \end{cases}$$

$$\mathcal{T}_1^H w(\mathbf{x}) = \begin{cases} w(\mathbf{x} - e_i) + r_i & \text{if } \exists i \text{ such that } x_i > 0 \text{ and } x_1 = x_2 = \dots = x_{i-1} = 0 \\ w(\mathbf{x}) & \text{otherwise} \end{cases}$$

$$\mathcal{T}_2^H w(\mathbf{x}) = \begin{cases} w(\mathbf{x} - e_{n+1}) + r_i & \text{if } x_{n+1} > 0 \\ w(\mathbf{x}) & \text{otherwise} \end{cases}$$

In the following, we compare the performance of the proposed heuristic control policy to the optimal control policy for the case of two types of packets. We examine the impact of a certain system variable by varying its value while maintaining all other variables constant. Similar to [59], we use as performance metric the reward Relative Deviation (RD) of the heuristic from the optimal. The RD , expressed in percentage, is defined as $RD = 100 \times (\psi^* - \psi^H) / \psi^*$ where ψ^* denotes the average reward rate of the optimal control policy obtained by solving Equation (4), and ψ^H denotes the average reward rate associated with the heuristic control policy obtained by solving Equation (5). Here, similar to the optimal policy, the expected reward is obtained using the value iteration algorithm with the same accuracy of 10^{-5} .

Table 1 shows a sample of 100 randomly generated system parameter values used to compute the performance of the heuristic control policy. Based on these results, it is clear that the heuristic

performs very well compared to the optimal policy. For a 95% confidence interval, the average RD is $1.40\% \pm 0.02\%$ with a range of $[0, 4.29]$.

Table 1. Heuristic performance compared to the optimal policy. RD is used as a measure of the Heuristic performance compared to the optimal policy.

μ_1	μ_2	λ_1	λ_2	r_1	r_2	r_3	RPD (%)
0.45	0.22	0.35	0.20	91	80	54	2.70
0.44	0.43	0.29	0.16	72	64	39	0.55
0.34	0.11	0.22	0.12	63	48	26	3.02
0.41	0.35	0.28	0.16	91	89	83	0.65
0.44	0.08	0.23	0.13	45	41	40	3.38
0.39	0.17	0.26	0.15	66	63	25	2.10
0.47	0.26	0.35	0.20	60	39	28	3.86
0.81	0.75	0.35	0.20	104	68	41	0.01
0.89	0.83	0.17	0.09	87	55	48	0.00
0.66	0.32	0.08	0.05	97	70	12	0.00
0.74	0.58	0.35	0.20	100	84	31	0.21
0.52	0.44	0.35	0.20	82	47	23	1.37
0.94	0.83	0.23	0.13	66	32	32	0.58
1.05	0.72	0.53	0.89	95	51	28	1.80
1.19	0.64	0.55	0.92	90	41	37	2.00
0.72	0.58	0.39	0.65	81	40	20	3.31
1.17	0.84	0.60	1.01	90	63	6	0.10
0.89	0.74	0.81	0.49	95	70	23	1.62
0.95	0.76	0.86	0.51	36	24	19	4.27
0.99	0.69	0.84	0.50	31	12	7	2.06
1.20	0.75	0.97	0.58	63	39	12	2.06
1.15	0.65	0.90	0.54	89	62	28	1.33
0.81	0.61	0.71	0.43	79	50	25	2.55
1.08	0.67	0.87	0.52	94	56	22	1.84
1.23	0.73	0.98	0.59	57	48	31	1.40
0.81	0.56	0.68	0.41	88	66	47	1.78
1.23	0.56	0.89	0.54	95	89	86	0.88
1.05	0.41	0.73	0.44	98	65	13	0.61
1.15	0.79	0.97	0.58	76	51	7	0.84
0.86	0.26	0.35	0.20	104	83	65	0.01
0.53	0.40	0.21	0.12	76	39	16	0.02
0.95	0.73	0.17	0.10	78	54	31	0.00
0.33	0.26	0.09	0.05	93	74	59	0.01
0.79	0.57	0.12	0.07	98	55	29	0.00
0.78	0.69	0.05	0.03	61	25	17	0.00
1.03	0.82	0.56	0.74	98	91	77	0.70
1.08	0.81	0.56	0.75	59	53	10	0.55
0.71	0.61	0.40	0.53	80	47	19	1.80
0.84	0.56	0.42	0.56	69	66	50	0.85
0.83	0.43	0.38	0.51	86	51	21	0.56
0.91	0.70	0.48	0.65	99	72	11	0.42
0.81	0.52	0.40	0.53	73	19	11	2.49
0.70	0.51	0.36	0.48	43	25	10	2.22
0.60	0.55	0.34	0.46	89	41	36	4.29
0.59	0.52	0.33	0.44	84	70	64	2.57
0.65	0.31	0.29	0.38	77	37	36	2.81
0.77	0.45	0.37	0.49	52	33	32	3.32
0.80	0.59	0.42	0.56	59	45	39	2.26
0.90	0.47	0.41	0.55	79	31	25	1.65
0.96	0.59	0.47	0.62	100	83	38	0.36
0.99	0.56	0.47	0.62	85	28	21	1.63
0.43	0.41	0.25	0.34	85	77	20	1.67
0.92	0.51	0.43	0.57	73	43	36	1.35
0.90	0.55	0.43	0.58	76	66	35	0.59
0.67	0.66	0.40	0.53	59	54	21	2.23
0.88	0.48	0.41	0.54	94	46	33	0.94
0.86	0.73	0.48	0.64	98	39	37	3.33
0.70	0.45	0.34	0.46	75	55	46	1.62

Table 1. Cont.

μ_1	μ_2	λ_1	λ_2	r_1	r_2	r_3	RPD (%)
0.64	0.48	0.34	0.45	87	50	7	0.34
0.52	0.50	0.31	0.41	85	78	77	3.26
0.87	0.82	0.51	0.68	98	81	64	1.42
0.93	0.76	0.50	0.67	81	55	21	1.04
0.81	0.65	0.44	0.58	72	40	34	2.83
0.86	0.36	0.37	0.49	78	78	21	0.25
0.74	0.42	0.35	0.47	65	59	10	0.37
0.86	0.76	0.49	0.65	83	46	34	2.37
0.88	0.45	0.40	0.53	77	75	66	0.61
0.43	0.42	0.26	0.34	95	67	46	3.45
0.84	0.49	0.40	0.53	88	38	23	1.18
0.69	0.58	0.38	0.51	70	45	32	2.65
0.90	0.41	0.39	0.52	79	37	20	0.74
0.89	0.44	0.40	0.53	78	38	20	0.81
0.90	0.87	0.53	0.71	53	31	18	3.34
0.95	0.46	0.42	0.56	96	78	48	0.31
0.81	0.33	0.34	0.46	88	80	19	0.24
0.93	0.83	0.53	0.70	88	30	22	2.90
0.83	0.79	0.48	0.65	83	50	20	1.86
0.94	0.69	0.49	0.65	65	62	54	1.08
0.86	0.45	0.40	0.53	76	62	51	0.74
0.56	0.40	0.29	0.38	109	12	12	3.69
0.75	0.24	0.30	0.39	119	90	10	0.57
1.13	0.25	0.41	0.55	109	51	45	0.40
0.90	0.50	0.42	0.56	113	41	21	0.69
0.54	0.43	0.29	0.38	81	42	9	1.16
0.65	0.45	0.33	0.44	88	88	37	0.76
1.11	0.58	0.51	0.68	98	61	22	0.29
0.57	0.40	0.29	0.39	107	102	17	0.54
1.15	0.65	0.54	0.72	36	25	13	1.33
1.16	0.62	0.53	0.71	91	45	37	0.82
0.69	0.55	0.37	0.50	109	56	39	1.84
0.73	0.53	0.38	0.50	51	29	14	2.19
0.83	0.77	0.48	0.64	109	101	72	1.08
0.91	0.72	0.49	0.65	55	51	36	1.42
1.01	0.39	0.42	0.56	109	80	65	0.29
0.74	0.54	0.39	0.52	89	76	15	0.61
1.12	0.61	0.52	0.69	120	38	25	0.66
1.08	0.80	0.56	0.75	81	47	19	0.87
0.76	0.50	0.38	0.50	64	46	16	0.99
1.04	0.36	0.42	0.56	102	100	95	0.38
0.86	0.80	0.50	0.66	109	93	42	1.03

For a system with more than two types of packets, computing the optimal policy becomes untractable due to the exponential explosion of the number of states. However, computing the thresholds of this heuristic is orders of magnitude faster than computing the optimal policy. In fact, much of the computation required (i.e., computation of the system parameters) for the heuristic can be done off-line, and the real-time computation requires no more than a table lookup. The computation of the system parameters of the heuristic is approximately, and at worst equal to the number of parameters times the size of the square of the cardinality of the state space. Numerical results show that the heuristic performs very well compared to the optimal policy. For a 95% confidence interval, the average computation time is $0.045\% \pm 0.005\%$ with a range of $[10^{-4}, 9 \times 10^{-4}]$ over a sample of 100 cases.

Given that the computation time of the optimal policy scales exponentially in the state space, computing the optimal policy beyond two priority classes is untractable. For instance, consider a system with three types of packets. Even if we succeed to compute the optimal policy and the associated parameters, it will require a huge state dependent look-up table (five hyper-surfaces representing the state dependent thresholds). For the heuristic however, we will need to store only five static control parameters (i.e., two admission and three routing threshold parameters).

Finally, in practice, traffic flow (i.e., arrival rate λ) changes over time. Our model however, assumes a constant traffic flow (λ_i for flow type- i). This is by no means a limitation of our model. In fact, this issue can be approached in one of two ways: either using a transient analysis which is well documented as being untractable especially in the context of an MDP framework; or computing different policy parameters off-line for each traffic flow. These parameters would be used for the particular traffic flow in effect during deployment.

7. Conclusions

In this paper, we considered an admission and routing control problem to address the issue of resource limitation in resource constrained networks (such as WSNs). We formulated the admission and routing control problem of two types of flows where one has a higher priority than the other, as a Markov decision problem. We characterized the optimal policy and showed that it is a state-dependent threshold type policy. Furthermore, we conducted extensive numerical experiments to gain more insight into the behavior of the optimal policy under different system parameters. Due to the computational challenges of the optimal policy (curse of dimensionality) which makes it untractable and too resource intensive to run on wireless devices, we proposed a heuristic that mimics the optimal control policy. Through extensive numerical results, we showed that the heuristic performs very well. It is also orders of magnitude faster than the optimal policy. Much of the required computation can be done off-line, and the real-time computation requires no more than a table lookup. We further generalized the heuristic for the case of a system with n types of flows ($n \geq 2$).

The results presented in this work provide a first step towards a better understanding of the structure of the optimal policy. There are several avenues for future research. In particular, it would be of interest to generalize the system to multi-server queues with more than two paths leading to the same destination. We expect the problem to become considerably more difficult with each additional feature and it is not clear if the optimal policy would be tractable. A clear extension of this work is to implement and test the proposed admission and routing control policy in real resource constrained network devices.

Funding: This research received no external funding.

Conflicts of Interest: The author declares no conflict of interest.

Appendix A

In order to prove Theorem 1 we first state the following Lemma.

Lemma A1. *Let \mathcal{V} be the set of functions, v , defined on \mathcal{R}^3 such that v satisfies Properties 1–3. Then $w_n \in \mathcal{V}, \forall n \in \mathcal{Z}^+$, where \mathcal{Z}^+ is the set of non-negative integers.*

Proof. First, note that by adding Properties 2 and 3 we obtain Property 1. Therefore, it suffices to prove Properties 2 and 3. To prove these properties, we use induction on the remaining number of periods. The following is a sketch of the proof.

- *Step 1:* we observe that Properties 2 and 3 hold for $n = 0$.
- *Step 2:* we assume 2 and 3 hold for some $n \geq 0$.
- *Step 3:* we prove that 2 and 3 hold for $n + 1$.

We introduce the following notation:

$$\begin{aligned}\Delta_i w_n(\mathbf{x} + e_i) &= w_n(\mathbf{x} + e_i) - w_n(\mathbf{x}) \\ \Delta_{ij} w_n(\mathbf{x} + e_i + e_j) &= w_n(\mathbf{x} + e_i) - w_n(\mathbf{x})\end{aligned}$$

Hence, we can write

$$\begin{aligned}\max\{w_n(\mathbf{x} + e_i), w_n(\mathbf{x})\} &= w_n(\mathbf{x}) + \max\{w_n(\mathbf{x} + e_i) - w_n(\mathbf{x}), 0\} \\ &= w_n(\mathbf{x}) + \max\{\Delta_i w_n(\mathbf{x}), 0\}\end{aligned}$$

We show that \mathcal{T}_s and \mathcal{T}_r satisfy Properties 2 and 3 and thus satisfy Property 1.

Operator \mathcal{T}_s

Note that operators \mathcal{T}_{a_1} , \mathcal{T}_1 and \mathcal{T}_2 satisfy Properties 2 and 3 by induction as they do not involve any decision. Hence, we only need to show that \mathcal{T}_{a_2} satisfies Properties 2 and 3. Using the difference operator, we have:

$$T_{a_2} w_{n-1}(\mathbf{x}) = w_{n-1}(\mathbf{x}) + \max\{\Delta_2 w_{n-1}(\mathbf{x}) - c, 0\}$$

To show that \mathcal{T}_{a_2} satisfies Property 2, let

$$\begin{aligned}\text{Let } Q(\mathbf{x}) &= T_{a_2} w_{n-1}(\mathbf{x} + 2e_i) - T_{a_2} w_{n-1}(\mathbf{x} + e_i) - T_{a_2} w_{n-1}(\mathbf{x} + e_i + e_j) + T_{a_2} w_{n-1}(\mathbf{x} + e_j) \\ &= (\Delta_i w_{n-1}(\mathbf{x} + e_i) - \Delta_i w_{n-1}(\mathbf{x} + e_j)) + \max\{\Delta_j w_{n-1}(\mathbf{x} + 2e_i) - c, 0\} - \max\{\Delta_j w_{n-1}(\mathbf{x} + e_i) - c, 0\} \\ &\quad - \max\{\Delta_j w_{n-1}(\mathbf{x} + e_i + e_j) - c, 0\} + \max\{\Delta_j w_{n-1}(\mathbf{x} + e_j) - c, 0\}\end{aligned}$$

Using submodularity property, we infer that : $\Delta_j w_{n-1}(\mathbf{x} + e_i) \geq \Delta_j w_{n-1}(\mathbf{x} + e_j) \geq \Delta_j w_{n-1}(\mathbf{x} + e_i + e_j)$ and

$$\Delta_j w_{n-1}(\mathbf{x} + e_i) \geq \Delta_j w_{n-1}(\mathbf{x} + 2e_i) \geq \Delta_j w_{n-1}(\mathbf{x} + e_i + e_j).$$

We prove the property for each case.

1. case 1: assume $\Delta_j w_{n-1}(\mathbf{x} + e_i + e_j) - c \geq 0$

$$Q(\mathbf{x}) = \Delta_i w_{n-1}(\mathbf{x} + e_i + e_j) - \Delta_i w_{n-1}(\mathbf{x} + 2e_i) \leq 0, \quad \text{by the inductive hypothesis.}$$

2. case 2: assume $\Delta_j w_{n-1}(\mathbf{x} + e_j) - c \geq 0 \geq \Delta_j w_{n-1}(\mathbf{x} + e_i + e_j) - c$
and $\Delta_j w_{n-1}(\mathbf{x} + 2e_i) - c \geq 0 \geq \Delta_j w_{n-1}(\mathbf{x} + e_i + e_j) - c$

$$Q(\mathbf{x}) = \underbrace{\Delta_i w_{n-1}(\mathbf{x} + e_i + e_j) - \Delta_i w_{n-1}(\mathbf{x} + 2e_i)}_{\leq 0 \text{ by the inductive hypothesis}} + \underbrace{\Delta_j w_{n-1}(\mathbf{x} + e_i + e_j) - c}_{\leq 0 \text{ by the assumption}} \leq 0$$

3. case 3: assume $\Delta_j w_{n-1}(\mathbf{x} + e_i) - c \geq 0 \geq \Delta_j w_{n-1}(\mathbf{x} + e_j) - c \geq \Delta_j w_{n-1}(\mathbf{x} + e_i + e_j) - c$ and $\Delta_j w_{n-1}(\mathbf{x} + 2e_i) - c \geq 0$.

$$Q(\mathbf{x}) = \underbrace{\Delta_i w_{n-1}(\mathbf{x} + e_i + e_j) - \Delta_i w_{n-1}(\mathbf{x} + 2e_i)}_{\leq 0 \text{ by the inductive hypothesis}} + \underbrace{\Delta_j w_{n-1}(\mathbf{x} + e_i + e_j) - c - \Delta_j w_{n-1}(\mathbf{x} + e_j) - c}_{\leq 0 \text{ by the assumption}} \leq 0$$

4. case 4: assume $\Delta_j w_{n-1}(\mathbf{x} + e_j) - c \geq 0$ and $\Delta_j w_{n-1}(\mathbf{x} + e_i) - c \geq 0 \geq \Delta_j w_{n-1}(\mathbf{x} + 2e_i) - c \geq \Delta_j w_{n-1}(\mathbf{x} + e_i + e_j) - c$

$$Q(\mathbf{x}) = \underbrace{\Delta_i w_{n-1}(\mathbf{x} + e_i + e_j) - \Delta_i w_{n-1}(\mathbf{x} + 2e_i)}_{\leq 0 \text{ by the inductive hypothesis}} + \underbrace{\Delta_j w_{n-1}(\mathbf{x} + e_i + e_j) - c - \Delta_j w_{n-1}(\mathbf{x} + 2e_i) + c}_{\leq 0 \text{ by the assumption}} \leq 0$$

5. case 5: assume $\Delta_j w_{n-1}(\mathbf{x} + e_i) - c \geq 0 \geq \Delta_j w_{n-1}(\mathbf{x} + e_j) - c \geq \Delta_j w_{n-1}(\mathbf{x} + e_i + e_j) - c$ and $\Delta_j w_{n-1}(\mathbf{x} + e_i) - c \geq 0 \geq \Delta_j w_{n-1}(\mathbf{x} + 2e_i) - c \geq \Delta_j w_{n-1}(\mathbf{x} + e_i + e_j) - c$

$$Q(\mathbf{x}) = \underbrace{\Delta_i w_{n-1}(\mathbf{x} + e_i) - \Delta_i w_{n-1}(\mathbf{x} + e_j)}_{\leq 0 \text{ by the inductive hypothesis}} - \underbrace{\Delta_j w_{n-1}(\mathbf{x} + e_i) - c}_{\geq 0 \text{ by the assumption}} \leq 0$$

6. case 6: assume $0 \geq \Delta_j w_{n-1}(\mathbf{x} + e_i) - c \geq \Delta_j w_{n-1}(\mathbf{x} + e_j) - c \geq \Delta_j w_{n-1}(\mathbf{x} + e_i + e_j) - c$ and $0 \geq \Delta_j w_{n-1}(\mathbf{x} + e_i) - c \geq \Delta_j w_{n-1}(\mathbf{x} + 2e_i) - c \geq \Delta_j w_{n-1}(\mathbf{x} + e_i + e_j) - c$

$$Q(\mathbf{x}) = \Delta_i w_{n-1}(\mathbf{x} + e_i) - \Delta_i w_{n-1}(\mathbf{x} + e_j) \leq 0 \text{ by the inductive hypothesis.}$$

Since the set \mathcal{V} is closed under addition and multiplication by a scalar, it follows that \mathcal{T}_s satisfies Property 2.

To show that \mathcal{T}_{a_2} satisfies Property 3, let

$$\begin{aligned} Q(\mathbf{x}) &= T_{a_2} w_{n-1}(\mathbf{x} + e_i + e_j) - T_{a_2} w_{n-1}(\mathbf{x} + e_j) - T_{a_2} w_{n-1}(\mathbf{x} + e_i) + T_{a_2} w_{n-1}(\mathbf{x}) \\ &= (\Delta_i w_{n-1}(\mathbf{x} + e_j) - \Delta_i w_{n-1}(\mathbf{x})) + \max\{\Delta_j w_{n-1}(\mathbf{x} + e_i + e_j) - c, 0\} - \max\{\Delta_j w_{n-1}(\mathbf{x} + e_j) - c, 0\} \\ &\quad - \max\{\Delta_j w_{n-1}(\mathbf{x} + e_i) - c, 0\} + \max\{\Delta_j w_{n-1}(\mathbf{x}) - c, 0\} \end{aligned}$$

Using submodularity, we infer that $\Delta_j w_{n-1}(\mathbf{x}) - c \geq \Delta_j w_{n-1}(\mathbf{x} + e_i) - c \geq \Delta_j w_{n-1}(\mathbf{x} + e_j) - c \geq \Delta_j w_{n-1}(\mathbf{x} + e_i + e_j) - c$. In the following we prove the property for each case.

1. case 1: assume $\Delta_j w_{n-1}(\mathbf{x}) - c \geq \Delta_j w_{n-1}(\mathbf{x} + e_i) - c \geq \Delta_j w_{n-1}(\mathbf{x} + e_j) - c \geq \Delta_j w_{n-1}(\mathbf{x} + e_i + e_j) - c \geq 0$

$$\begin{aligned} Q(\mathbf{x}) &= (\Delta_i w_{n-1}(\mathbf{x} + e_j) - \Delta_i w_{n-1}(\mathbf{x})) + \Delta_j w_{n-1}(\mathbf{x} + e_i + e_j) - c - \Delta_j w_{n-1}(\mathbf{x} + e_j) + c \\ &\quad - \Delta_j w_{n-1}(\mathbf{x} + e_i) + c + \Delta_j w_{n-1}(\mathbf{x}) - c \\ &= \Delta_i w_{n-1}(\mathbf{x} + 2e_j) - \Delta_i w_{n-1}(\mathbf{x} + e_j) \\ &\leq 0 \text{ by the inductive hypothesis} \end{aligned}$$

2. case 2: assume $\Delta_j w_{n-1}(\mathbf{x}) - c \geq \Delta_j w_{n-1}(\mathbf{x} + e_i) - c \geq \Delta_j w_{n-1}(\mathbf{x} + e_j) - c \geq 0 \geq \Delta_j w_{n-1}(\mathbf{x} + e_i + e_j) - c$

$$\begin{aligned} Q(\mathbf{x}) &= \Delta_i w_{n-1}(\mathbf{x} + 2e_j) - \Delta_i w_{n-1}(\mathbf{x} + e_j) - \Delta_j w_{n-1}(\mathbf{x} + e_i + e_j) + c \\ &= \Delta_i w_{n-1}(\mathbf{x} + 2e_j) - \Delta_i w_{n-1}(\mathbf{x} + e_j) - \Delta_j w_{n-1}(\mathbf{x} + e_i + e_j) + c \\ &= -(\Delta_j w_{n-1}(\mathbf{x} + e_j) - c) \\ &\leq 0 \text{ by the assumption} \end{aligned}$$

3. case 3: assume $\Delta_j w_{n-1}(\mathbf{x}) - c \geq \Delta_j w_{n-1}(\mathbf{x} + e_i) - c \geq 0 \geq \Delta_j w_{n-1}(\mathbf{x} + e_j) - c \geq \Delta_j w_{n-1}(\mathbf{x} + e_i + e_j) - c$

$$Q(\mathbf{x}) = \underbrace{-(\Delta_j w_{n-1}(\mathbf{x} + e_j) - c)}_{\leq 0 \text{ from case 2}} + \underbrace{\Delta_j w_{n-1}(\mathbf{x} + e_j) - c}_{\leq 0 \text{ by the assumption}} = 0$$

4. case 4: assume $\Delta_j w_{n-1}(\mathbf{x}) - c \geq 0 \geq \Delta_j w_{n-1}(\mathbf{x} + e_i) - c \geq \Delta_j w_{n-1}(\mathbf{x} + e_j) - c \geq \Delta_j w_{n-1}(\mathbf{x} + e_i + e_j) - c$

$$Q(\mathbf{x}) = \underbrace{0}_{\text{Per case 3}} + \underbrace{\Delta_j w_{n-1}(\mathbf{x} + e_i) - c}_{\leq 0 \text{ by the assumption}} \leq 0$$

5. case 5: assume $0 \geq \Delta_j w_{n-1}(\mathbf{x}) - c \geq \Delta_j w_{n-1}(\mathbf{x} + e_i) - c \geq \Delta_j w_{n-1}(\mathbf{x} + e_j) - c \geq \Delta_j w_{n-1}(\mathbf{x} + e_i + e_j) - c$

$$Q(\mathbf{x}) = (\Delta_i w_{n-1}(\mathbf{x} + e_j) - \Delta_i w_{n-1}(\mathbf{x})) \leq 0 \text{ by the inductive hypothesis}$$

Since the set \mathcal{V} is closed under addition and multiplication by a scalar, it follows that \mathcal{T}_s satisfies Property 3.

Operator \mathcal{T}_r

$$\mathcal{T}_r w_{n-1}(\mathbf{x}) = \begin{cases} \max\{w_{n-1}(\mathbf{x} - e_1 + e_3) - c_2, w_{n-1}(\mathbf{x})\} & \text{if } x_2 = 0, x_1 \geq 0, \text{ route type 1} \\ \max\{w_{n-1}(\mathbf{x} - e_2 + e_3), w_{n-1}(\mathbf{x})\} & \text{if } x_2 > 0, \text{ route type 2} \end{cases}$$

To show that \mathcal{T}_r satisfies Property 2, we need to prove that

$$\mathcal{T}_r w_{n-1}(\mathbf{x} + 2e_i) - \mathcal{T}_r w_{n-1}(\mathbf{x} + e_i) - \mathcal{T}_r w_{n-1}(\mathbf{x} + e_i + e_j) + \mathcal{T}_r w_{n-1}(\mathbf{x} + e_j) \leq 0$$

For simplicity we prove the property for $i = 1$ and $j = 2$. The rest of the properties follow the same procedure.

1. $x_2 = 0, x_1 \geq 0$

$$\mathcal{T}_r w_{n-1}(\mathbf{x}) = w_{n-1}(\mathbf{x}) + \max\{\Delta_{-13} w_{n-1}(\mathbf{x}) - c_2, 0\} \quad (\text{A1})$$

$$\begin{aligned} \text{Let } Q(\mathbf{x}) &= \mathcal{T}_r w_{n-1}(\mathbf{x} + 2e_1) - \mathcal{T}_r w_{n-1}(\mathbf{x} + e_1) - \mathcal{T}_r w_{n-1}(\mathbf{x} + e_1 + e_2) + \mathcal{T}_r w_{n-1}(\mathbf{x} + e_2) \\ &= w_{n-1}(\mathbf{x} + 2e_1) - w_{n-1}(\mathbf{x} + e_1) - w_{n-1}(\mathbf{x} + e_1 + e_2) + w_{n-1}(\mathbf{x} + e_2) \\ &\quad + \max\{\Delta_{-13} w_{n-1}(\mathbf{x} + 2e_1) - c_2, 0\} - \max\{\Delta_{-13} w_{n-1}(\mathbf{x} + e_1) - c_2, 0\} \\ &\quad - \max\{\Delta_{-13} w_{n-1}(\mathbf{x} + e_1 + e_2) - c_2, 0\} + \max\{\Delta_{-13} w_{n-1}(\mathbf{x} + e_2) - c_2, 0\} \end{aligned}$$

Note that, by Property 2 the following inequalities hold.

$$\Delta_{-13} w_{n-1}(\mathbf{x} + e_1) - c_2 \geq \Delta_{-13} w_{n-1}(\mathbf{x} + e_1 + e_2) - c_2 \geq \Delta_{-13} w_{n-1}(\mathbf{x} + 2e_1) - c_2 \geq \Delta_{-13} w_{n-1}(\mathbf{x} + e_2) - c_2$$

We prove the property for each case.

- (a) case 1: assume $\Delta_{-13} w_{n-1}(\mathbf{x} + e_1) - c_2 \geq \Delta_{-13} w_{n-1}(\mathbf{x} + e_1 + e_2) - c_2 \geq \Delta_{-13} w_{n-1}(\mathbf{x} + 2e_1) - c_2 \geq \Delta_{-13} w_{n-1}(\mathbf{x} + e_2) - c_2 \geq 0$.

$$\begin{aligned} Q(\mathbf{x}) &= w_{n-1}(\mathbf{x} + 2e_1) - w_{n-1}(\mathbf{x} + e_1) - w_{n-1}(\mathbf{x} + e_1 + e_2) + w_{n-1}(\mathbf{x} + e_2) \\ &\quad + \Delta_{-13} w_{n-1}(\mathbf{x} + 2e_1) - \Delta_{-13} w_{n-1}(\mathbf{x} + e_1) - \Delta_{-13} w_{n-1}(\mathbf{x} + e_1 + e_2) + \Delta_{-13} w_{n-1}(\mathbf{x} + e_2) \\ &= \Delta_1 w_{n-1}(\mathbf{x} + e_3) - \Delta_1 w_{n-1}(\mathbf{x} - e_1 + e_2 + e_3) \\ &\leq 0 \text{ by the inductive hypothesis} \end{aligned}$$

- (b) case 2: assume $\Delta_{-13}w_{n-1}(\mathbf{x} + e_1) - c_2 \geq \Delta_{-13}w_{n-1}(\mathbf{x} + e_1 + e_2) - c_2 \geq \Delta_{-13}w_{n-1}(\mathbf{x} + 2e_1) - c_2 \geq 0 \geq \Delta_{-13}w_{n-1}(\mathbf{x} + e_2) - c_2$.

$$Q(\mathbf{x}) = \underbrace{\Delta_1 w_{n-1}(\mathbf{x} + e_3) - \Delta_1 w_{n-1}(\mathbf{x} - e_1 + e_2 + e_3)}_{\text{From case 1}} - (\Delta_{-13}w_{n-1}(\mathbf{x} + e_2) - c_2)$$

using the assumption, $\Delta_{-13}w_{n-1}(\mathbf{x} + e_1 + e_2) - c_2 \geq \Delta_{-13}w_{n-1}(\mathbf{x} + 2e_1) - c_2$,
and after simplification, we can express $Q(\mathbf{x})$ as,

$$Q(\mathbf{x}) \leq \underbrace{\Delta_1 w_{n-1}(\mathbf{x} + e_1) - \Delta_1 w_{n-1}(\mathbf{x} + e_2)}_{\leq 0 \text{ by concavity Property 2}} - \underbrace{(\Delta_{-13}w_{n-1}(\mathbf{x} + e_1) - c_2)}_{\geq 0 \text{ by assumption}}$$

$$\leq 0$$

- (c) case 3: assume $\Delta_{-13}w_{n-1}(\mathbf{x} + e_1) - c_2 \geq \Delta_{-13}w_{n-1}(\mathbf{x} + e_1 + e_2) - c_2 \geq 0 \geq \Delta_{-13}w_{n-1}(\mathbf{x} + 2e_1) - c_2 \geq \Delta_{-13}w_{n-1}(\mathbf{x} + e_2) - c_2$.

$$Q(\mathbf{x}) \leq \underbrace{\Delta_1 w_{n-1}(\mathbf{x} + e_1) - \Delta_1 w_{n-1}(\mathbf{x} + e_2) - (\Delta_{-13}w_{n-1}(\mathbf{x} + e_1) - c_2)}_{\leq 0 \text{ from previous case}} - (\Delta_{-13}w_{n-1}(\mathbf{x} + 2e_1) - c_2)$$

$$\leq \underbrace{\Delta_1 w_{n-1}(\mathbf{x} + e_1) - \Delta_1 w_{n-1}(\mathbf{x} + e_2) - (\Delta_{-13}w_{n-1}(\mathbf{x} + e_1) - c_2)}_{\leq 0 \text{ from previous case}} - \underbrace{(\Delta_{-13}w_{n-1}(\mathbf{x} + 2e_1) - c_2)}_{\leq \Delta_{-13}w_{n-1}(\mathbf{x} + e_1) - c_2 \text{ by assumption}}$$

$$\leq \Delta_1 w_{n-1}(\mathbf{x} + e_1) - \Delta_1 w_{n-1}(\mathbf{x} + e_2) - 2(\Delta_{-13}w_{n-1}(\mathbf{x} + e_1) - c_2)$$

$$\leq 0$$

- (d) case 4: assume $\Delta_{-13}w_{n-1}(\mathbf{x} + e_1) - c_2 \geq 0 \geq \Delta_{-13}w_{n-1}(\mathbf{x} + e_1 + e_2) - c_2 \geq \Delta_{-13}w_{n-1}(\mathbf{x} + 2e_1) - c_2 \geq \Delta_{-13}w_{n-1}(\mathbf{x} + e_2) - c_2$.

$$Q(\mathbf{x}) \leq \underbrace{\Delta_1 w_{n-1}(\mathbf{x} + e_1) - \Delta_1 w_{n-1}(\mathbf{x} + e_2) - 2(\Delta_{-13}w_{n-1}(\mathbf{x} + e_1) - c_2)}_{\leq 0 \text{ from previous case}} + \underbrace{\Delta_{-13}w_{n-1}(\mathbf{x} + e_1 + e_3) + c_2}_{\leq 0 \text{ by assumption}}$$

$$\leq 0$$

- (e) case 5: assume $0 \geq \Delta_{-13}w_{n-1}(\mathbf{x} + e_1) - c_2 \geq \Delta_{-13}w_{n-1}(\mathbf{x} + e_1 + e_2) - c_2 \geq \Delta_{-13}w_{n-1}(\mathbf{x} + 2e_1) - c_2 \geq \Delta_{-13}w_{n-1}(\mathbf{x} + e_2) - c_2$.

$$Q(\mathbf{x}) \leq \underbrace{\Delta_1 w_{n-1}(\mathbf{x} + e_1) - \Delta_1 w_{n-1}(\mathbf{x} + e_2) - 2(\Delta_{-13}w_{n-1}(\mathbf{x} + e_1) - c_2) + \Delta_{-13}w_{n-1}(\mathbf{x} + e_1 + e_3) + c_2}_{\leq 0 \text{ from previous case}}$$

$$+ \underbrace{\Delta_{-13}w_{n-1}(\mathbf{x} + e_1) + c_2}_{\leq 0 \text{ by assumption}}$$

$$\leq 0$$

2. $x_2 > 0$

$$\begin{aligned} \mathcal{T}_r w_{n-1}(\mathbf{x}) &= \max\{w_{n-1}(\mathbf{x} - e_j + e_3), w_{n-1}(\mathbf{x})\} \\ &= w_{n-1}(\mathbf{x}) + \max\{\Delta_{-23}w_{n-1}(\mathbf{x}), 0\} \end{aligned} \quad (\text{A2})$$

Note that $\mathcal{T}_r(\mathbf{x})$ given in Equation (A2) is a special case of $\mathcal{T}_r(\mathbf{x})$ expression given in Equation (A1) where $c_2 = 0$ and e_i is substituted by e_j , therefore the proof of Property 2 follows.

In order to prove that \mathcal{T}_r satisfies Property 3, we need to prove that

$$T_r w_{n-1}(\mathbf{x} + e_i + e_j) - T_r w_{n-1}(\mathbf{x} + e_j) - T_r w_{n-1}(\mathbf{x} + e_i) + T_r w_{n-1}(\mathbf{x}) \leq 0.$$

For simplicity we prove the property for $i = 1$ and $j = 2$. The rest of the properties follow the same procedure.

1. $x_2 = 0, x_1 \geq 0$

$$T_r w_{n-1}(\mathbf{x}) = w_{n-1}(\mathbf{x}) + \max\{\Delta_{-13} w_{n-1}(\mathbf{x}) - c_2, 0\} \tag{A3}$$

$$\begin{aligned} Q(\mathbf{x}) &= T_r w_{n-1}(\mathbf{x} + e_1 + e_2) - T_r w_{n-1}(\mathbf{x} + e_2) - T_r w_{n-1}(\mathbf{x} + e_1) + T_r w_{n-1}(\mathbf{x}) \leq 0 \\ &= w_{n-1}(\mathbf{x} + e_1 + e_2) - w_{n-1}(\mathbf{x} + e_2) - w_{n-1}(\mathbf{x} + e_1) + w_{n-1}(\mathbf{x}) \\ &\quad + \max\{\Delta_{-13} w_{n-1}(\mathbf{x} + e_1 + e_2) - c_2, 0\} - \max\{\Delta_{-13} w_{n-1}(\mathbf{x} + e_2) - c_2, 0\} \\ &\quad - \max\{\Delta_{-13} w_{n-1}(\mathbf{x} + e_1) - c_2, 0\} + \max\{\Delta_{-13} w_{n-1}(\mathbf{x}) - c_2, 0\} \end{aligned}$$

Note that, by Property 2 the following inequalities hold. $\Delta_{-13} w_{n-1}(\mathbf{x} + e_1 + e_2) - c_2 \geq \Delta_{-13} w_{n-1}(\mathbf{x} + e_2) - c_2 \geq \Delta_{-13} w_{n-1}(\mathbf{x} + e_1) - c_2 \geq \Delta_{-13} w_{n-1}(\mathbf{x}) - c_2$. We prove the property for each case.

(a) case 1: assume $\Delta_{-13} w_{n-1}(\mathbf{x} + e_1 + e_2) - c_2 \geq \Delta_{-13} w_{n-1}(\mathbf{x} + e_2) - c_2 \geq \Delta_{-13} w_{n-1}(\mathbf{x} + e_1) - c_2 \geq \Delta_{-13} w_{n-1}(\mathbf{x}) - c_2 \geq 0$

$$\begin{aligned} Q(\mathbf{x}) &= T_r w_{n-1}(\mathbf{x} + e_1 + e_2) - T_r w_{n-1}(\mathbf{x} + e_2) - T_r w_{n-1}(\mathbf{x} + e_1) + T_r w_{n-1}(\mathbf{x}) \\ &= w_{n-1}(\mathbf{x} + e_1 + e_2) - w_{n-1}(\mathbf{x} + e_2) - w_{n-1}(\mathbf{x} + e_1) + w_{n-1}(\mathbf{x}) \\ &\quad + (w_{n-1}(\mathbf{x} + e_2 + e_3) - w_{n-1}(\mathbf{x} + e_1 + e_2) - c_2) - (w_{n-1}(\mathbf{x} - e_1 + e_2 + e_3) - w_{n-1}(\mathbf{x} + e_2) - c_2) \\ &\quad - (w_{n-1}(\mathbf{x} + e_3) - w_{n-1}(\mathbf{x} + e_1) - c_2) + (w_{n-1}(\mathbf{x} - e_1 + e_3) - w_{n-1}(\mathbf{x}) - c_2) \\ &= \Delta_1 w_{n-1}(\mathbf{x} - e_1 + e_2 + e_3) - \Delta_1 w_{n-1}(\mathbf{x} - e_1 + e_3) \\ &\leq 0 \text{ by the inductive proposition of Property 3} \end{aligned}$$

(b) case 2: assume $\Delta_{-13} w_{n-1}(\mathbf{x} + e_1 + e_2) - c_2 \geq \Delta_{-13} w_{n-1}(\mathbf{x} + e_2) - c_2 \geq \Delta_{-13} w_{n-1}(\mathbf{x} + e_1) - c_2 \geq 0 \geq \Delta_{-13} w_{n-1}(\mathbf{x}) - c_2$.

$$\begin{aligned} Q(\mathbf{x}) &= T_r w_{n-1}(\mathbf{x} + e_1 + e_2) - T_r w_{n-1}(\mathbf{x} + e_2) - T_r w_{n-1}(\mathbf{x} + e_1) + T_r w_{n-1}(\mathbf{x}) \\ &= \Delta_1 w_{n-1}(\mathbf{x} - e_1 + e_2 + e_3) - \Delta_1 w_{n-1}(\mathbf{x} - e_1 + e_3) - (\Delta_{-13} w_{n-1}(\mathbf{x}) - c_2) \\ &\quad \text{Per assumption } w_{n-1}(\mathbf{x} + e_2 + e_3) - w_{n-1}(\mathbf{x} - e_1 + e_2 + e_3) \geq w_{n-1}(\mathbf{x} + e_1 + e_2) - w_{n-1}(\mathbf{x} + e_2) \\ Q(\mathbf{x}) &\leq w_{n-1}(\mathbf{x} + e_1 + e_2) - w_{n-1}(\mathbf{x} + e_2) + w_{n-1}(\mathbf{x}) - \underbrace{(w_{n-1}(\mathbf{x} + e_3) - c_2)}_{\geq w_{n-1}(\mathbf{x} + e_1) \text{ by the assumption}} \\ &\leq w_{n-1}(\mathbf{x} + e_1 + e_2) - w_{n-1}(\mathbf{x} + e_2) + w_{n-1}(\mathbf{x}) - w_{n-1}(\mathbf{x} + e_1) \\ &= \Delta_1 w_{n-1}(\mathbf{x} + e_2) - \Delta_1 w_{n-1}(\mathbf{x}) \\ &\leq 0 \text{ by the inductive hypothesis} \end{aligned}$$

(c) case 3: assume $\Delta_{-13} w_{n-1}(\mathbf{x} + e_1 + e_2) - c_2 \geq \Delta_{-13} w_{n-1}(\mathbf{x} + e_2) - c_2 \geq 0 \geq \Delta_{-13} w_{n-1}(\mathbf{x} + e_1) - c_2 \geq \Delta_{-13} w_{n-1}(\mathbf{x}) - c_2$.

$$\begin{aligned} Q(\mathbf{x}) &= \underbrace{\Delta_1 w_{n-1}(\mathbf{x} + e_2) - \Delta_1 w_{n-1}(\mathbf{x})}_{\leq 0 \text{ from previous case}} + \underbrace{(\Delta_{-13} w_{n-1}(\mathbf{x} + e_1) - c_2)}_{\leq 0 \text{ by assumption}} \\ &\leq 0 \end{aligned}$$

- (d) case 4: assume $\Delta_{-13}w_{n-1}(\mathbf{x} + e_1 + e_2) - c_2 \geq 0 \geq \Delta_{-13}w_{n-1}(\mathbf{x} + e_2) - c_2 \geq \Delta_{-13}w_{n-1}(\mathbf{x} + e_1) - c_2 \geq \Delta_{-13}w_{n-1}(\mathbf{x}) - c_2$.

$$Q_4(\mathbf{x}) = \underbrace{\Delta_1 w_{n-1}(\mathbf{x} + e_2) - \Delta_1 w_{n-1}(\mathbf{x}) + (\Delta_{-13} w_{n-1}(\mathbf{x} + e_i) - c_2)}_{\leq 0 \text{ from previous case}} + \underbrace{(\Delta_{-13} w_{n-1}(\mathbf{x} + e_j) - c_2)}_{\leq 0 \text{ by assumption}} \leq 0$$

- (e) case 5: assume $0 \geq \Delta_{-13}w_{n-1}(\mathbf{x} + e_1 + e_2) - c_2 \geq \Delta_{-13}w_{n-1}(\mathbf{x} + e_2) - c_2 \geq \Delta_{-13}w_{n-1}(\mathbf{x} + e_1) - c_2 \geq \Delta_{-13}w_{n-1}(\mathbf{x}) - c_2$.

$$Q(\mathbf{x}) = \Delta_1 w_{n-1}(\mathbf{x} + e_2) - \Delta_1 w_{n-1}(\mathbf{x}) \leq 0 \text{ by the inductive hypothesis.}$$

2. For $x_2 > 0$

$$\begin{aligned} \mathcal{T}_r w_{n-1}(\mathbf{x}) &= \max\{w_{n-1}(\mathbf{x} - e_2 + e_3), w_{n-1}(\mathbf{x})\} \\ &= w_{n-1}(\mathbf{x}) + \max\{\Delta_{-23} w_{n-1}(\mathbf{x}), 0\} \end{aligned} \quad (\text{A4})$$

Note that the expression of $\mathcal{T}_r(\mathbf{x})$ given in Equation (A4) is a special case of the one of $\mathcal{T}_r(\mathbf{x})$ given in Equation (A3) where $c_2 = 0$ and e_i is substituted by e_j , therefore $\mathcal{T}_r(\mathbf{x})$ satisfies Property 3. Finally, it is straightforward (using the induction argument) to show that $w_n(\mathbf{x}) = \max(\mathcal{T}_s w_{n-1}(\mathbf{x}), \mathcal{T}_r w_{n-1}(\mathbf{x}))$, for $x_1 + x_2 > 0$ satisfies Properties 2 and 3, hence satisfies Property 1. This completes the proof of Lemma A1.

□

Proof of Theorem 1. For a given n we obtain an optimal policy Π_n . It can be shown that there exists an optimal stationary policy Π^* such that $\Pi^* = \lim_{n \rightarrow \infty} \Pi_n$. This is true because $w_n(\mathbf{x})$ converges to an optimal time-independent discounted reward, w , for any $\alpha > 0$ (see [58]).

The admission policy structure is due to Property 1. Indeed the latter indicates that the difference $w_n(\mathbf{x} + e_2) - w_n(\mathbf{x})$ is non increasing in x_2 . Thus $w_n(\mathbf{x})$ has a unique maximum at $\mathcal{A}(x_1, x_3)$ as defined in the theorem.

The routing policy structure for type-2 packets is due to Property 2. The latter indicates that the difference $w_n(\mathbf{x} - e_2 + e_3) - w_n(\mathbf{x})$ is non decreasing in x_2 . Therefore, there exists a unique threshold $\mathcal{R}_2(x_1, x_3)$. Similarly, for type-1 packets, Property 3 implies that there exists a unique threshold $\mathcal{R}_1(x_3)$ (since $x_2 = 0$ in this case). $\mathcal{R}_1(x_3)$ and $\mathcal{R}_2(x_1, x_3)$ are defined in the theorem.

This completes the proof of the theorem. □

References

- Puccinelli, D.; Haenggi, M. Wireless sensor networks: Applications and challenges of ubiquitous sensing. *IEEE Circuits Syst. Mag.* **2005**, *5*, 19–31. [[CrossRef](#)]
- Rabby, M.K.M.; Alam, M.S. A priority based energy harvesting scheme for charging embedded sensor nodes in wireless body area networks. *PLoS ONE* **2019**, *14*, e0214716. [[CrossRef](#)] [[PubMed](#)]
- Anguita, D.; Brizzolara, D.; Parodi, G. Building an Underwater Wireless Sensor Network Based on Optical: Communication: Research Challenges and Current Results. In Proceedings of the 2009 Third International Conference on Sensor Technologies and Applications, Glyfada, Greece, 18–23 June 2009; pp. 476–479.
- Hassan, J. Future of Applications In Mobile wireless Sensor Networks. In Proceedings of the 2018 1st International Conference on Computer Applications Information Security (ICCAIS), Riyadh, Kingdom of Saudi Arabia, 4–6 April 2018; pp. 1–6.
- Nishikawa, Y.; Sasamura, T.; Ishizuka, Y.; Sugimoto, S.; Iwasaki, S.; Wang, H.; Fujishima, T.; Fujimoto, T.; Yamashita, K.; Suzuki, T.; et al. Design of stable wireless sensor network for slope monitoring. In Proceedings of the 2018 IEEE Topical Conference on Wireless Sensors and Sensor Networks (WiSNet), Anaheim, CA, USA, 14–17 January 2018; pp. 8–11.

6. Sun, Z.; Akyildiz, I.F. Connectivity in Wireless Underground Sensor Networks. In Proceedings of the 2010 7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), Boston, MA, USA, 21–25 June 2010; pp. 1–9.
7. Bogdanoski, M.; Shuminoski, T.; Risteski, A. Analysis of the SYN flood DoS attack. *Int. J. Comput. Netw. Inf. Secur.* **2013**, *5*, 1–11. [[CrossRef](#)]
8. Oncioiu, R.; Simion, E. Approach to Prevent SYN Flood DoS Attacks in Cloud. In Proceedings of the 2018 International Conference on Communications (COMM), Bucharest, Romania, 14–16 June 2018; pp. 447–452.
9. Eberhardt, J.; Tai, S. On or Off the Blockchain? Insights on Off-Chaining Computation and Data. In Proceedings of the European Conference on Service-Oriented and Cloud Computing, Oslo, Norway, 27–29 September 2017; pp. 3–15. [[CrossRef](#)]
10. Gupta, M.K.; Hemachandra, N.; Venkateswaran, J. Some parametrized dynamic priority policies for 2-class M/G/1 queues: completeness and applications. *arXiv* **2018**, arXiv:1804.03564.
11. Dimitris, B.; Niño-Mora, J. Optimization of Multiclass Queueing Networks with Changeover Times via the Achievable Region Approach: Part II, the Multi-Station Case. *Math. Oper. Res.* **1999**, *24*, 331–361.
12. Hassin, R.; Puerto, J.; Fernandez, F.R. The use of relative priorities in optimizing the performance of a queueing system. *Eur. J. Oper. Res.* **2009**, *193*, 476–483. [[CrossRef](#)]
13. Gupta, M.K.; Hemachandra, N. On 2-moment completeness of non pre-emptive, non anticipative work conserving scheduling policies in some single class queues. In Proceedings of the 2015 13th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), Mumbai, India, 25–29 May 2015; pp. 267–274.
14. Rawal, A.; Kavitha, V.; Gupta, M.K. Optimal surplus capacity utilization in polling systems via fluid models. In Proceedings of the 2014 12th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), Hammamet, Tunisia, 12–16 May 2014; pp. 381–388.
15. Li, C.; Neely, M.J. Delay and rate-optimal control in a multi-class priority queue with adjustable service rates. In Proceedings of the 2012 Proceedings IEEE INFOCOM, Orlando, FL, USA, 25–30 March 2012; pp. 2976–2980.
16. Sinha, S.K.; Rangaraj, N.; Hemachandra, N. Pricing surplus server capacity for mean waiting time sensitive customers. *Eur. J. Oper. Res.* **2010**, *205*, 159–171. [[CrossRef](#)]
17. Puterman, M.L. *Markov Decision Processes*; John Wiley and Sons Inc.: Hoboken, NJ, USA, 2005.
18. Gouvy, N.; Hamouda, E.; Mitton, N.; Zorbas, D. Energy efficient multi-flow routing in mobile Sensor Networks. In Proceedings of the 2013 IEEE Wireless Communications and Networking Conference (WCNC), Shanghai, China, 7–10 April 2013; pp. 1968–1973.
19. Zhang, D.; Li, G.; Zheng, K.; Ming, X.; Pan, Z. An Energy-Balanced Routing Method Based on Forward-Aware Factor for Wireless Sensor Networks. *IEEE Trans. Ind. Inform.* **2014**, *10*, 766–773. [[CrossRef](#)]
20. Zhang, D.G.; Liu, S.; Zhang, T.; Liang, Z. Novel unequal clustering routing protocol considering energy balancing based on network partition and distance for mobile education. *J. Netw. Comput. Appl.* **2017**, *88*, 1–9. [[CrossRef](#)]
21. Zhang, D.; Zheng, K.; Zhang, T.; Wang, X. A novel multicast routing method with minimum transmission for WSN of cloud computing service. *Soft Comput.* **2014**, *19*, 1817–1827. [[CrossRef](#)]
22. Dressler, F. *Self-Organization in Sensor and Actor Networks*; John Wiley and Sons Ltd.: Chichester, UK, 2007.
23. Pathan, A. *Security of Self-Organizing Networks: MANET, WSN, WMN, VANET*; CRC Press: Boca Raton, FL, USA, 2011.
24. Gu, X.; Yu, J.; Yu, D.; Wang, G.; Lv, Y. ECDC: An energy and coverage-aware distributed clustering protocol for wireless sensor networks. *Comput. Electr. Eng.* **2014**, *40*, 384–398. [[CrossRef](#)]
25. Le, N.T.; Jang, M. Energy-efficient coverage guarantees scheduling and routing strategy for wireless sensor networks. *Int. J. Distrib. Sens. Netw.* **2015**, *11*, 612383. [[CrossRef](#)]
26. More, A.; Raisinghani, V. A survey on energy efficient coverage protocols in wireless sensor networks. *J. King Saud Univ. Comput. Inf. Sci.* **2017**, *29*, 428–448. [[CrossRef](#)]
27. Alghamdi, T. Energy efficient protocol in wireless sensor network: optimized cluster head selection model. *Telecommun. Syst.* **2020**, *74*, 331–345. [[CrossRef](#)]
28. Wang, Z.; Chen, Y.; Liu, B. A sensor node scheduling algorithm for heterogeneous wireless sensor networks. *Int. J. Distrib. Sens. Netw.* **2019**, *15*. [[CrossRef](#)]

29. Shih, E.; Cho, S.; Lee, F.S.; Calhoun, B.H.; Chandrakasan, A. Design Considerations for Energy-Efficient Radios in Wireless Microsensor Networks. *J. VLSI Signal Process. Syst. Signal Image Video Technol.* **2004**, *37*, 77–94. [[CrossRef](#)]
30. Santhi, V.; Natarajan, A.M. Active Queue Management Algorithm for TCP Networks Congestion Control. *Eur. J. Sci. Res.* **2011**, *54*, 245–257.
31. Comer, D.E. *Internetworking with TCP/IP*, 5th ed.; Prentice Hall: Upper Saddle River, NJ, USA, 2006.
32. Floyd, S.; Jacobson, V. Random Early Detection (RED) gateways for Congestion Avoidance. *IEEE/ACM Trans. Netw.* **1993**, *1*, 397–413. [[CrossRef](#)]
33. Robertazzi, T.G. *Computer Networks and Systems: Queuing Theory and Performance Evaluation-Queuing Theory and Performance Evaluation*, 3rd ed.; Springer: Berlin/Heidelberg, Germany, 2012.
34. Kleinrock, L. *Queuing Systems, Volume 2: Computer Applications*; Wiley: Hoboken, NJ, USA, 1976.
35. Kleinrock, L. *Queuing Systems, Volume 1: Theory*; Wiley: Hoboken, NJ, USA, 1975.
36. Lenin, R.B.; Ramaswamy, S. Performance analysis of wireless sensor networks using queuing networks. *Ann. Oper. Res.* **2015**, *233*, 237–261. [[CrossRef](#)]
37. Adou, Y.; Markova, E.; Gudkova, I. Performance Measures Analysis of Admission Control Scheme Model for Wireless Network, Described by a Queuing System Operating in Random Environment. In Proceedings of the 2018 10th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), Moscow, Russia, 5–9 November 2018; pp. 1–5.
38. Borodakiy, V.Y.; Samouylov, K.E.; Gudkova, I.A.; Ostriкова, D.Y.; Ponomarenko-Timofeev, A.A.; Turlikov, A.M.; Andreev, S.D. Modeling unreliable LSA operation in 3GPP LTE cellular networks. In Proceedings of the 2014 6th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), St. Petersburg, Russia, 6–8 October 2014; pp. 390–396.
39. Kempa, W.M. Analytical Model of a Wireless Sensor Network (WSN) Node Operation with a Modified Threshold-Type Energy Saving Mechanism. *Sensors* **2019**, *19*, 3114. [[CrossRef](#)]
40. Ghosh, S.; Unnikrishnan, S. Reduced power consumption in wireless sensor networks using queue based approach. In Proceedings of the 2017 International Conference on Advances in Computing, Communication and Control (ICAC3), Mumbai, India, 1–2 December 2017.
41. Moy, J. Open Shortest Path First (OSPF). Available online: <https://tools.ietf.org/html/rfc2328> (accessed on 11 November 2020)
42. Hedrick, C. *Routing Information Protocol (RIP)*; Rutgers University: Camden, NJ, USA, 1988.
43. Zhang, X.; Zhou, Z.; Cheng, D. Efficient path routing strategy for flows with multiple priorities on scale-free networks. *PLoS ONE* **2017**, *12*, e0172035. [[CrossRef](#)] [[PubMed](#)]
44. Groenevelt, R.; Koole, G.; Nain, P. On the bias vector of a two-class preemptive priority queue. *Math. Methods Oper. Res.* **2002**, *55*, 107–120. [[CrossRef](#)]
45. Brouns, G.A.; van der Wal, J. Optimal threshold policies in a two-class preemptive priority queue with admission and termination control. *Queueing Syst.* **2006**, *54*, 21–33. [[CrossRef](#)]
46. Xu, S. A duality approach to admission and scheduling controls of queues. *Queueing Syst. Theory Appl.* **1994**, *16*, 272–300. [[CrossRef](#)]
47. Righter, R. Expulsion and scheduling control for multiclass queues with heterogeneous servers. *Queueing Syst. Theory Appl.* **2000**, *34*, 289–300. [[CrossRef](#)]
48. Brouns, G.; van der Wal, J. Optimal threshold policies in a workload model with a variable number of service phases per job. *Math. Methods Oper. Res.* **2003**, *58*, 483–501. [[CrossRef](#)]
49. Puterman, M. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*; John Wiley and Sons Inc.: Hoboken, NJ, USA, 1994.
50. Lippman, A.L. Applying a new device in the optimization of exponential queueing systems. *Oper. Res.* **1975**, *23*, 687–710. [[CrossRef](#)]
51. Khorov, E.; Kiryanov, A.; Loginov, V.; Lyakhov, A. Head-of-line blocking avoidance in multimedia streaming over wireless networks. In Proceedings of the 2014 IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC), Washington DC, USA, 2–5 September 2014; pp. 1142–1146. [[CrossRef](#)]
52. Elhafsi, E.H.; Molle, M.; Manjunath, D. On the application of forking nodes to product-form queueing networks. *Int. J. Commun. Syst.* **2008**, *21*, 135–165. [[CrossRef](#)]

53. Kumar, A.; Manjunath, D.K.J. *Communication Networking: An Analytical Approach*; Morgan Kaufman: San Francisco, CA, USA, 2004.
54. Bellman, R. *Dynamic Programming*; Princeton University Press: Princeton, NJ, USA, 1957.
55. Howard, R. *Dynamic Programming and Markov Processes*; Massachusetts Institute of Technology Press: Cambridge, UK, 1960.
56. Liu, Z.; Nain, P.; Towsley, D. Sample path methods in the control of queues. *Queueing Syst.* **1995**, *21*, 293–335. [[CrossRef](#)]
57. Weber, R.R.; Stidham, S. Optimal control of service rates in networks of queues. *Adv. Appl. Probab.* **1987**, *19*, 202–218. [[CrossRef](#)]
58. Bertsekas, D.P. *Dynamic Programming and Optimal Control*, 2nd ed.; Athena scientific: Belmont, MA, USA, 2001; Volume 2.
59. ElHafsi, M.; Fang, J.; Hamouda, E. Optimal production and inventory control of multi-class mixed backorder and lost sales demand class models. *Eur. J. Oper. Res.* **2020**. [[CrossRef](#)]

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).