Stochastic Throughput Optimization for Two-hop Systems with Finite Relay Buffers

Bo Zhou, Ying Cui, Member, IEEE, and Meixia Tao, Senior Member, IEEE

Abstract—Optimal queueing control of multi-hop networks remains a challenging problem even in the simplest scenarios. In this paper, we consider a two-hop half-duplex relaying system with random channel connectivity. The relay is equipped with a finite buffer. We focus on stochastic link selection and transmission rate control to maximize the average system throughput subject to a half-duplex constraint. We formulate this stochastic optimization problem as an infinite horizon average cost Markov decision process (MDP), which is well-known to be a difficult problem. By using sample-path analysis and exploiting the specific problem structure, we first obtain an equivalent Bellman equation with reduced state and action spaces. By using relative value iteration algorithm, we analyze the properties of the value function of the MDP. Then, we show that the optimal policy has a threshold-based structure by characterizing the supermodularity in the optimal control. Based on the threshold-based structure and Markov chain theory, we further simplify the original complex stochastic optimization problem to a static optimization problem over a small discrete feasible set and propose a low-complexity algorithm to solve the simplified static optimization problem by making use of its special structure. Furthermore, we obtain the closed-form optimal threshold for the symmetric case. The analytical results obtained in this paper also provide design insights for two-hop relaying systems with multiple relays equipped with finite relay buffers.

Index Terms—Wireless relay system, finite buffer, throughput optimization, Markov decision process, Markov chain theory, matrix update, structural results.

I. INTRODUCTION

The demand for communication services has been changing from traditional voice telephony services to mixed voice, data, and multimedia services. When data and real-time services are considered, it is necessary to jointly consider both physical layer issues such as coding and modulation as well as higher layer issues such as network congestion and delay. It is also important to model these services using queueing concepts [1].

Consider a two-hop relaying system with one source node (S), one half-duplex relay node (R) and one destination node (D) under i.i.d. on-off fading. Under conventional decode-and-forward (DF) relay protocol, the listening phase (S-R) is always followed by the retransmission phase (R-D) [2]. As a result, the system throughput (from S to D) is restricted by the instantaneous flow balance constraint, i.e., the minimum of the throughputs from S to R and from R to D. Under random link connectivity, the system throughput is non-zero only when both the S-R and the R-D link are connected.

Now consider a finite buffer at R and apply cross-layer buffered decode-and-forward (BDF) protocol to exploit the random channel connectivity and queueing [3]. Under BDF, due to buffering at R, a scheduling slot can be adaptively allocated for the S-R transmission or the R-D transmission, according to the R queue length and link quality. Then, the throughput to D can be made non-zero provided that the R-D link is connected. While the buffer at R appears to offer clear advantages, it is not clear how to design the optimal control to maximize the average system throughput given a finite relay buffer. Buffering a certain amount of bits at R can capture R-D transmission opportunity (when only R-D link is on) and improve the throughput in the future. However, buffering too many bits at R may waste S-R transmission opportunity (when only S-R link is on) due to R buffer overflow. Therefore, it remains unclear how to take advantage of the finite buffer at R to balance the transmission rates of the S-R link and R-D link so as to maximize the average system throughput.

Recently, the idea of cross-layer design using queueing concepts has been considered in the context of multi-hop networks with buffers. In [3] and [4], the authors consider the delay-optimal control for two-hop networks with infinite buffers at the source and relay. Specifically, in [4], the authors obtain a delay-optimal link selection policy for non-fading channels. Then, in [3], the authors extend the analysis to the i.i.d. on/off fading channels and show that a threshold-based link selection policy is asymptotically delay-optimal when the scheduling slot duration tends to zero. However, it is not known whether the delay-optimal policy still has a threshold-based structure. In [3], the authors consider a two-hop relaying system with general fading channels, and assume infinite backlog at the source node and an infinite buffer at the relay node. The optimal link selection policy is obtained to maximize the average system throughput without considering the stability of the relay queue. In all the above references, the relay is assumed to be equipped with an infinite buffer and the proposed algorithms cannot guarantee that the instantaneous relay queue length is below a certain threshold. However, in practical systems, buffers are finite and buffer overflow may lead to significant performance loss [6]. Therefore, it is important to consider finite relay buffers in designing optimal resource control for multi-hop networks to support data and real-time services.

Lyapunov drift approach represents a systematic way to queue stabilization problems for general multi-hop networks [7], [8]. The derived stochastic control usually does not require system statistics predict beforehand and can be easily implemented online. However, Lyapunov drift approach...
cannot properly handle buffers of finite size. References [9] and [10] modify the quadratic Lyapunov function [7] in the traditional Lyapunov drift approach to design stochastic control algorithms for multi-hop networks with infinite source buffers and finite relay buffers. In [9], the authors propose scheduling algorithms to stabilize source queues under a fixed routing design. In [10], the authors propose joint $\epsilon$-optimal flow control, routing and scheduling algorithms to maximize the throughput and show that the gap between the optimal performance and the throughput of each proposed algorithm is inversely proportional to the relay buffer size. References [11] and [12] adopt a similar modified Lyapunov function to the one in [9] and [10], and propose $\epsilon$-optimal algorithms to optimize network utility for multi-hop networks with finite source and relay buffers. The gap between the optimal performance and the performance of each proposed algorithm is also inversely proportional to the buffer size. Note that, although control algorithms for the finite buffer case can be obtained by applying the modified Lyapunov functions, they cannot achieve performance that is optimal or arbitrarily close to optimal.

On the other hand, dynamic programming represents a systematic approach to optimal queuing control problems [13], [14]. Generally, there exist only numerical solutions, which do not typically offer many design insights and are usually impractical for implementation due to the curse of dimensionality [14]. For example, in [15] and [16], the authors formulate the delay-optimal control problem for two-hop relay systems as infinite horizon average-cost Markov Decision Processes (MDPs) [14] and propose distributed numerical algorithms using approximate MDP and stochastic learning. However, the obtained algorithms may still be too complex for practical systems and do not offer many design insights. The concept of supermodularity [17] is usually applied to analyze structural properties of the optimal policies for MDPs in simple queueing systems. Most existing literature considers the structural analysis of a single queue with either controlled arrival rate or controlled departure rate [18]–[20]. To the best of our knowledge, the structural analysis for a single queue with both controlled arrival and departure rates is still unknown.

In general, the stochastic throughput maximization for multi-hop systems with fading channels and finite relay buffers is still unknown even for the case of a simple two-hop relaying system. In this paper, we shall tackle some of the technical challenges. We consider a two-hop relaying system with one source node (S), one relay node (R) and one destination node (D). S cannot transmit packets to D due to the limited coverage and has to communicate with D with the help of R via the S-R link and the R-D link. R is half-duplex and equipped with a finite buffer. We consider a discrete-time system, in which the time axis is partitioned into scheduling slots with unit slot duration. The slots are indexed by $t$ ($t = 1, 2, \ldots$).

![System model](image)

**Fig. 1:** System model.

II. SYSTEM MODEL

In this section, we elaborate on the system topology, the physical layer model and the queuing model.

A. System Topology

As illustrated in Fig. 1 we consider a two-hop relaying system with one source node (S), one relay node (R) and one destination node (D). S cannot transmit packets to D due to the limited coverage and has to communicate with D with the help of R via the S-R link and the R-D link. R is half-duplex and equipped with a finite buffer. We consider a discrete-time system, in which the time axis is partitioned into scheduling slots with unit slot duration. The slots are indexed by $t$ ($t = 1, 2, \ldots$).

B. Physical Layer Model

We model the channel fading of the S-R link and the R-D link with i.i.d. random link connectivity. This channel fading model is widely used in the literature [21]–[23]. Let $G_{s,t}, G_{r,t} \in \mathcal{G} \triangleq \{0, 1\}$ denote the link connectivity state information (CSI) of the S-R link and the R-D link at slot $t$, respectively, where 1 denotes connected and 0 not connected. Let $G_t = (G_{s,t}, G_{r,t}) \in \mathcal{G} \times \mathcal{G}$ denote the joint CSI at the $t$-th slot, where $\mathcal{G}$ denotes the joint CSI state space.

**Assumption 1 (Random Link Connectivity Model):** $\{G_{s,t}\}$ and $\{G_{r,t}\}$ are both i.i.d. over time, where in each slot $t$, the probabilities of being 1 for $G_{s,t}$ and $G_{r,t}$ are $p_s$ and $p_r$, respectively, i.e., $Pr[G_{s,t} = 1] = p_s$ and $Pr[G_{r,t} = 1] = p_r$. Furthermore, $\{G_{s,t}\}$ and $\{G_{r,t}\}$ are independent of each other.

---

1 This channel fading model is widely used in the literature [21]–[23].
We consider packet transmission. The maximum transmission rates (packet/slot) of the S-R link when \( G_{s,t} = 1 \) and the R-D link when \( G_{r,t} = 1 \) are given by \( R_s \) and \( R_r \), respectively.\(^3\) Due to the half-duplex restriction and random channel connectivity, we shall dynamically select one link to transmit at each slot.

C. Queueing Model

We assume that S has infinite backlog (i.e., always has data to transmit) and consider a finite buffer of size \( N_r \) (in number of packets) at R. Assume \( N_r > R_s, R_r \). The finite buffer at R is used to hold the packet flow from S. We consider the buffered decode-and-forward (BDF) protocol\(^4\) to exploit the potential benefit of buffering at R under random channel connectivity. Specifically, according to BDF, (i) S can transmit packets to R when the S-R link is connected, and R decodes and stores the packets from S in its buffer; (ii) R can transmit the packets in its buffer to D when the R-D link is connected. Using the buffer at R and BDF, we can dynamically select the S-R link or the R-D link to transmit to exploit the potential benefit of buffering at R under random channel connectivity. Assume the buffer at R and BDF, we can

III. Problem Formulation

In this section, we first introduce the control policy and formulate the stochastic throughput optimization problem.

A. Control Policy and Queue Dynamics

For notation convenience, we denote \( \chi_t = (Q_t, G_t) \in \chi = \mathcal{Q} \times \mathcal{G} \) as the system state at the \( t \)-th slot. Let \( a_{s,t} \in \{0, 1\} \) and \( a_{r,t} \in \{0, 1\} \) denote whether the S-R link or the R-D link is scheduled, respectively, in the \( t \)-th slot, where 1 denotes scheduled and 0 otherwise. Let \( u_{s,t} \in \{0, 1, \ldots, R_s\} \) and \( u_{r,t} \in \{0, 1, \ldots, R_r\} \) denote the transmission rates of \( S \) and \( R \) in the \( t \)-th slot, respectively. Given an observed system state \( \chi \), the link selection action \( (a_s, a_r) \in \{0, 1\}^2 \) and the transmission rate control action \( (u_s, u_r) \in \{0, 1, \ldots, R_s\} \times \{0, 1, \ldots, R_r\} \) are determined according to a stationary policy defined below.

**Definition 1 (Stationary Policy):** A stationary link selection and transmission rate control policy \( \Omega = (\Omega_s, \Omega_r) \) is a mapping from the system state \( \chi = (Q, G) \) to the link selection action \( (a_s, a_r) \) and the transmission rate control action \( (u_s, u_r) \), where \( \Omega_s(\chi) = (a_s, a_r) \) and \( \Omega_r(\chi) = (u_s, u_r) \) satisfy the following constraints.

1. \( a_s, a_r \in \{0, 1\} \);
2. \( a_s + a_r \leq 1 \) (orthogonal link selection);
3. \( (a_s, a_r) = (0, 0), \ G = (0, 0) \) (link not connected);
4. \( u_s \in \{0, 1, \ldots, \min\{R_s, N_r - Q\}\} \) (departure rate at S);
5. \( u_r \in \{0, 1, \ldots, \min\{R_r, Q\}\} \) (departure rate at R).

Note that, we consider the finite buffer at R and do not allow packet drop at R, since packet drop will waste system resources and will not contribute to the system throughput. This can be seen from the constraint in 4). Therefore, given a stationary control policy \( \Omega \) defined in Definition 1\(^5\), the queue dynamics at R is given by

\[
Q_{t+1} = Q_t + a_s,t u_{s,t} - a_r,t u_{r,t}, \ orall t = 1, 2, \cdots. \tag{1}
\]

B. MDP Formulation

From Assumption 1 and the queue dynamics in (1), we can see that the induced random process \( \{\chi_t\} \) under policy \( \Omega \) is a Markov chain with the following transition probability

\[
\Pr[\chi_{t+1} = \chi_t, \Omega(\chi_t)] = \Pr[G_{t+1} = G_t] \Pr[Q_{t+1} = Q_t | Q_t, G_t]. \tag{2}
\]

For a given stationary unchain policy \( \Omega \), the average system throughput is given by

\[
\tilde{R}^\Omega = \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[r(\chi_t, \Omega(\chi_t))], \tag{3}
\]

where \( r(\chi_t, \Omega(\chi_t)) = a_{r,t} u_{r,t} \) is the per-stage reward (i.e., the departure rate at R at slot \( t \), indicating the number of packets delivered by the two-hop relaying system) and the expectation is taken w.r.t. the measure induced by the policy \( \Omega \).

We wish to find an optimal link selection and transmission rate control policy \( \Omega^* \) to maximize the average system throughput \( \tilde{R}^\Omega \) in (3).

**Problem 1 (Stochastic Throughput Optimization):**

\[
\tilde{R}^* = \lim_{\Omega \to \infty} \max \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[r(\chi_t, \Omega(\chi_t))]. \tag{4}
\]

Note that Problem 1 is an infinite horizon average cost MDP, which is well-known to be a difficult problem \(^{14}\). While dynamic programming represents a systematic approach for MDP, there generally exist only numerical solutions. These solutions do not typically offer many design insights, and are usually impractical for implementation in practical systems due to the curse of dimensionality \(^{14}\).

Fig. 2 illustrates in the following, how we shall address the above challenges to solve Problem 1. Specifically, in Sections IV and V, we shall analyze the properties of the optimal policy. Based on these properties, we shall simplify Problem 1 to a static optimization problem (Problem 2) and develop a low-complexity algorithm (Algorithm 3) to solve it. Finally,
we shall obtain the corresponding static optimization problem (Problem 5) for the symmetric case and derive its closed-form optimal solution.

IV. STRUCTURE OF OPTIMAL POLICY

In this section, we first obtain an equivalent Bellman equation based on reduced state and action spaces. Then, we show that the optimal policy has a threshold-based structure.

A. Optimality Equation

By exploiting some special structures in our problem, we obtain the following equivalent Bellman equation by reducing the state and action spaces. By solving the Bellman equation, we can obtain the optimal policy to Problem 1.

Theorem 1 (Equivalent Bellman Equation): (i) The optimal transmission rate control policy $\Omega^*_\mu$ is given by

$$\Omega^*_\mu(\chi) = \left\{ \begin{array}{ll}
\min\{R_s, N_r - Q\}, & \min\{R_r, Q\}, & \forall \chi \in \chi. \\
\end{array} \right. \quad (5)$$

(ii) There exists $\theta(\{V(\cdot)\})$ satisfying the following equivalent Bellman equation:

$$\theta + V(Q) = \bar{p}_s \bar{p}_r V(Q) + p_s \bar{p}_r V(\min\{Q + R_s, N_r\}) + p_s p_r \max\left\{ a_r \min\{Q, R_r\} \right\} + V(\min\{N_r - Q, R_s\} - a_r \min\{Q, R_r\}), \quad (6)$$

where $\left\lfloor x \right\rfloor^+ \triangleq \max\{x, 0\}$, $\bar{p}_s = 1 - p_s$, $\bar{p}_r = 1 - p_r$, and $\bar{a}_r = 1 - a_r$. $\theta$ is the optimal value to Problem 1 for all initial state $\chi \in \chi$ and $V(\cdot)$ is called the value function.

(iii) The optimal link selection policy $\Omega^*_\alpha$ is given by

$$\Omega^*_\alpha(\chi) = \left\{ \begin{array}{ll}
(0, 0), & G = (0, 0); \\
(0, 1), & G = (0, 1); \\
(1, 0), & G = (1, 0); \\
(\alpha^*_\alpha(Q), \alpha^*_\alpha(Q)), & G = (1, 1). \\
\end{array} \right. \quad (7)$$

where

$$\alpha^*_\alpha(Q) = \arg\max_{a_r} \left\{ a_r \min\{Q, R_r\} \right\} + V(\min\{N_r - Q, R_s\} - a_r \min\{Q, R_r\}), \quad (8)$$

and $\alpha^*_\alpha(Q) = 1 - \alpha^*_\alpha(Q)$.

Proof: Please see Appendix A.

Remark 1 (Reduction of State and Action Spaces): The Bellman equation in (6) is defined over the QSI state space. Thus, the system state space $Q \times G$ in Definition 1 is reduced to the QSI state space $Q$. The action space reduction can be observed by comparing Definition 1 with (5) and (7).

Note that the closed-form optimal transmission rate control policy $\Omega^*_\mu$ has already been obtained in (5). The optimal link selection policy $\Omega^*_\alpha$ is determined by the policy $\alpha^*_\alpha$ in (8). Thus, we only need to consider the optimal link selection for $G = (1, 1)$. In the following, we also refer to $\alpha^*_\alpha$ as the optimal link selection policy. To obtain the optimal policy, it remains to characterize $\alpha^*_\alpha$. From Theorem 1 we can see that $\alpha^*_\alpha$ depends on the QSI state $Q$ through the value function $V(\cdot)$. Obtaining $V(\cdot)$ involves solving the equivalent Bellman equation in (6) for all $Q$. There is no closed-form solution in general [14]. Brute force solutions such as value iteration and policy iteration are usually impractical for implementation [14] and do not yield many design insights. Therefore, it is desirable to study the structure of $\alpha^*_\alpha$.

B. Threshold Structure of Optimal Link Selection Policy

To further simplify the problem and obtain design insights, we study the structure of the optimal link selection policy. Most existing literature considers the structural analysis of a single queue with either controlled arrival rate or departure rate. In our case, we control both the arrival and departure rates of the relay queue. Therefore, the structural analysis in our case is more challenging.

First, by the relative value iteration algorithm [19], we can iteratively prove the following properties of the value function.

Lemma 1 (Properties of Value Function): The value function $V(Q)$ satisfies the following properties:

1) $V(Q)$ is monotonically non-decreasing in $Q$;
2) $V(Q + 1) - V(Q) \leq 1$, $Q \in \{0, 1, \ldots, N_r - 1\}$;
3) $V(Q + R_s + R_r + 1) - V(Q + R_s + R_r) \leq V(Q + 1) - V(Q)$, $Q \in \{0, 1, \ldots, N_r - (Q + R_s + R_r) + 1\}$.

Proof: Please see Appendix B. ■

Next, define the state-action reward function $J(Q, a_r)$ as follows.

$$J(Q, a_r) \triangleq \bar{p}_s \bar{p}_r V(Q) + p_s \bar{p}_r V(\min\{Q + R_s, N_r\}) + p_s p_r \left[a_r \min\{Q, R_r\} + V(\min\{N_r - Q, R_s\} - a_r \min\{Q, R_r\}) \right], \quad (9)$$

Note that $J(Q, a_r)$ is related to the R.H.S. of the Bellman equation in (6). By applying the relative value iteration algorithm [14], we can show that the state-action reward function $J(Q, a_r)$ is supermodular in $(Q, a_r)$, i.e.,

$$J(Q + 1, 1) - J(Q + 1, 0) \geq J(Q, 1) - J(Q, 0). \quad (10)$$

By [17] Lemma 4.7.1, supermodularity is a sufficient condition for the monotone policies to be optimal. Thus, we have the following theorem.

A function $f(x, y): X \times Y \to \mathbb{R}$ is supermodular in $(x, y)$ if $f(x + 1, y + 1) - f(x + 1, y) \geq f(x, y + 1) - f(x, y)$ [17].
optimal transmission rate control in (5) and a threshold-based link selection policy for \( G = (1,1) \) has the threshold-based structure, i.e.,
\[
\alpha_r(Q) = \begin{cases} 
1, & \text{if } Q > Q^*_t; \\
0, & \text{otherwise.} 
\end{cases} 
\]

\( Q^*_t \) is the optimal threshold.

Proof: Please see Appendix C.

**Remark 2 (Interpretation of Theorem 2):** By Theorem 2, we know that when \( G = (1,1) \), it is optimal to schedule the S-R link if \( Q > Q^*_t \) and to schedule the R-D link otherwise. The intuition is as follows. When the relay queue length is large (\( Q > Q^*_t \)), the S-R transmission opportunities may be wasted when \( G = (1,0) \) due to the overflow of the finite R buffer. Therefore, when \( Q > Q^*_t \), we should reduce the relay queue length when \( G = (1,1) \). When the relay queue length is small (\( Q \leq Q^*_t \)), the R-D transmission opportunities may be wasted when \( G = (0,1) \), as there are not enough packets left to transmit. Therefore, when \( Q \leq Q^*_t \), we should schedule the S-R link. These design insights also hold for two-hop relaying systems with multiple relays equipped with finite relay buffers.

V. OPTIMAL SOLUTIONS FOR GENERAL CASE

In this section, we first obtain a simplified static optimization problem for Problem 1 by making use of the properties of the optimal policy in Theorems 1 and 2. Then, based on the special structure, we develop a low-complexity algorithm to solve the static optimization problem.

A. Recurrent Class

By the structure of the optimal policy in Theorems 1 and 2, we can restrict our attention to the optimal transmission rate control in (5) and a threshold-based link selection policy \( \alpha_r \) for \( G = (1,1) \), i.e.,
\[
\alpha_r(Q) = \begin{cases} 
1, & \text{if } Q > Q^*_t; \\
0, & \text{otherwise.} 
\end{cases} 
\]
where \( Q^*_t \) is the threshold. In the following, we use \( \{Q_t\} \) to denote the relay queue state process under the policies in (5) and (12). \( \{Q_t\} \) is a stationary Discrete-Time Markov Chain (DTMC), the transition probabilities of which are determined by the threshold \( Q^*_t \) and the statistics of the CSI (i.e., \( p_s \) and \( p_r \)). The transition diagram of this DTMC is illustrated in Fig. 3.

Next, we study the steady-state probabilities of \( \{Q_t\} \). Let \( R_s/R_r = a/b \) such that \( a \) and \( b \) are two positive integers having no factors in common. Denote
\[
R = R_s/a = R_r/b. 
\]
There exist a positive integer \( n \) and \( l \in \{0,1,\cdots,R-1\} \) such that
\[
N_r = nR + l. 
\]
Using the Bézout’s identity, we obtain the following lemma which characterizes the recurrent class of \( \{Q_t\} \).

**Lemma 2 (Recurrent Class):** For any \( R_s, R_r, N_r \) under the optimal transmission rate control in (5) and a threshold-based link selection policy in (12) with any \( Q_{th} \in Q \), the recurrent class \( C \) of \( \{Q_t\} \) is given by
\[
C = \begin{cases} 
\{0,R,2R,\cdots,nR\}, & \text{if } l = 0; \\
\{0,R,2R,\cdots,nR,l,l+R,l+R,\cdots,N_r\}, & \text{if } l \neq 0. 
\end{cases} 
\]
where \( R \) is given by (13) and \( l,n \) satisfy (14). The size of \( C \) is \(|C| = (l+1)(n+1)|\).

Proof: Please see Appendix D.

Note that \( C \subseteq Q \) and \( C \) is the same for any \( Q_{th} \in Q \) (\(|Q| = N_r + 1 \)). For \( \{Q_t\} \), the steady-state probability of each transient state (\( \notin C \)) is zero \( [24] \). The ergodic throughput only depends on the average departure rates and steady-state probabilities of the recurrent states (\( \in C \)).

B. Equivalent Problem

We first consider a threshold-based policy in (12) with the threshold chosen from \( C \) instead of \( Q_{th} \). We wish to find the optimal threshold \( q_{th} \in C \) to maximize the ergodic system throughput (i.e., the ergodic reward of \( \{Q_t\} \)). Later, in Lemma 3, we shall show the relationship between \( q_{th} \in C \) and \( Q_{th} \in Q \).

Given \( q_{th} \), we can express the transition probability from \( i \) to \( j \) as \( p_{i,j}(q_{th}) \), where \( i,j \in C \). Let \( \mathbf{P}(q_{th}) = (p_{i,j}(q_{th}))_{i,j \in C} \) and \( \pi(q_{th}) = (\pi_i(q_{th}))_{i \in C} \) denote the transition probability matrix and the steady-state probability row vector of the recurrent class \( C \), respectively. Note that \( \mathbf{P}(q_{th}) \) is fully determined by \( q_{th} \) and the statistics of the CSI (i.e., \( p_s \) and \( p_r \)), and can be easily obtained, as illustrated in Fig. 3. By the Perron-Frobenius theorem \( [24] \), \( \pi(q_{th}) \) can be computed from the following system of linear equations.

\[
\begin{bmatrix} 
\pi(q_{th}) \mathbf{P}(q_{th}) = \pi(q_{th}) \\
\|\pi(q_{th})\| = 1 
\end{bmatrix} 
\]

Let \( r_i(q_{th}) \) denote the average departure rate at state \( i \in Q \) under the threshold \( q_{th} \), which is given by
\[
r_i(q_{th}) = \begin{cases} 
\tilde{p}_sp_r \min\{i,R_i\}, & \text{if } i \leq q_{th}; \\
\rho_r \min\{i,R_i\}, & \text{otherwise.} 
\end{cases} 
\]
Let \( \mathbf{r}(q_{th}) = (r_i(q_{th}))_{i \in C} \) denote the average departure rate column vector of the recurrent class \( C \). Therefore, the ergodic system throughput can be expressed as \( \pi(q_{th}) \mathbf{r}(q_{th}) \).

\(^5\)When \( l \neq 0 \), the Markov chain is similar and omitted.
Now, we formulate a static optimization problem to maximize the ergodic system throughput as below.

**Problem 2 (Equivalent Optimization Problem):**

\[ \hat{p}^* = \max_{q_{th} \in C} \pi(q_{th}) r(q_{th}). \]  

(17)

Let \( q_{th}^* \in C \) denote the optimal solution to Problem 2. Note that \( Q_{th}^* \in Q \) denote the optimal threshold to Problem 1. The following lemma summarizes the relationship between Problem 1 and Problem 2.

**Lemma 3 (Relationship between Problem 1 and Problem 2):**

The optimal values to Problems 1 and 2 are the same, i.e., \( \hat{R}^* = \hat{r}^* \). Any threshold \( Q_{th}^* \in \{q_{th}, q_{th} + 1, \ldots, q_{th} + R - 1\} \) is optimal to Problem 1.

**Proof:** By Lemma 2, Fig. 3 and (16), any threshold \( Q_{th}^* \in \{q_{th}, q_{th} + 1, \ldots, q_{th} + R - 1\} \) leads to the same \( P(q_{th}) \) and \( r(q_{th}) \), thus achieves the same ergodic throughput, where \( q_{th} \in C \). By ergodic theory, the time-average system throughput in (14) is equivalent to the ergodic system throughput in (17). Thus, the optimal control to Problem 1 can be obtained by solving Problem 2.

By Lemma 5 instead of solving Problem 1 which is a complex stochastic optimization problem, we can solve Problem 2, which is a static problem over the smaller feasible set \( C \subseteq Q \).

**C. Algorithm for Problem 2:**

Problem 2 is a discrete optimization problem over the feasible set \( C \), which can be solved in a brute-force way by computing \( \pi(q_{th}) \) for each \( q_{th} \in C \) separately, and then choose the optimal solution \( q_{th}^* = \arg \max_{q_{th} \in C} \pi(q_{th}) r(q_{th}) \). The brute-force method has high complexity and does not make use of the structure of the problem. In this part, we develop a low-complexity algorithm to solve Problem 2 by computing \( \pi(q_{th}) \) iteratively based on the special structure of \( P(q_{th}) \).

Sort the elements of \( C \) in ascending order, i.e., \( c_1, c_2, \ldots, c_{|C|} \), where \( c_k \) denotes the \( k \)-th smallest element. For notation simplicity, we use \( P(k) \) and \( \pi(k) \) to represent \( P(q_{th}) \) and \( \pi(q_{th}) \), respectively, where \( c_k = q_{th} \). In other words, instead of using \( q_{th} \in C \), each variable in \( C \) is indexed by \( k \). Denote \( A(k) = I_{|C|} - P(k)^T \).

\[
A(k) \pi(k) = 0, \\
||\pi(k)|| = 1.
\]

(19)

The steady-state probability vector \( \pi(k) \) in (19) can be obtained using partition factorization method [25] as follows. Let \( A_k(k) \) denote the submatrix formed by removing the \( (k+1) \)-th column and the \( |C| \)-th row of \( A(k) \). Accordingly, let \( K(k) \) denote the permutation matrix such that

\[ A(k) K(k)^T = \begin{bmatrix} A_k(k) & \beta(k) \\ z(i)^T & y(k) \end{bmatrix}. \]

(20)

Then, \( \pi(k) \) can be computed by the partition factorization method [25] in Algorithm 1.

**Algorithm 1 Algorithm to Compute \( \pi(k) \):**

1. Obtain \( P(k) \) and \( A(k) \) in (18).
2. Find \( K(k) \), partition \( A(k) \) into the form (20) to obtain \( A_k(k) \) and \( y(k) \).
3. Solve the subsystem for \( \hat{x}(k) \)

\[ A_k(k) \hat{x}(k) = -y(k). \]

(21)

4. Let \( x(k) = K(k) \frac{\hat{x}(k)}{1} \) and normalize \( x(k) \) to obtain \( \pi(k) \), i.e., \( \pi(k) = \frac{x(k)}{||x(k)||} \).

For each \( k = 1, 2, \ldots, |C| \), \( \hat{x}(k) \) in (21) can be obtained by Gaussian elimination. This leads to the brute-force algorithm, which computes \( \pi(k) \) for each \( k \) separately.

**Remark 3 (Computational Complexity of Brute-force Algorithm):**

The computational complexity is measured as the number of floating-point operations (flops) [26].

The computation of \( \hat{x}(k) \) for each \( k = 1, 2, \ldots, |C| \) using Gaussian elimination requires \( 2|C|(1)^3/3 \) flops [26]. Thus, to compute \( \hat{x}(k) \) for all \( k \), the brute-force algorithm requires \( 2|C|(1)^3/3 \) flops, i.e., has complexity \( O(|C|^4) \).

![Fig. 4: Illustration of \( P(k) \).](image)

The following lemma summarizes the relationship between \( A_{k+1}(k+1)^{-1} \) and \( A_k(k)^{-1} \) which directly results from this special structure of \( P(k) \).

**Algorithm 2 Relationship between \( A_{k+1}(k+1)^{-1} \) and \( A_k(k)^{-1} \):**

Let \( K(k) \) denote the permutation matrix obtained by exchanging the \( (k+1) \)-th and \( (k+2) \)-th columns of \( I_{|C|-1} \) and let \( a_{k+1}(k+1) \) and \( a_{k+2}(k) \) be the \( (k+1) \)-th column of \( A_{k+1}(k+1) \) and the \( (k+2) \)-th column of \( A_k(k) \), respectively. Then, \( A_{k+1}(k+1)^{-1} \) and \( A_k(k)^{-1} \) satisfy:

\[
A_{k+1}(k+1)^{-1} = K(k) A_k(k)^{-1}
\]

(22)

A flop is defined as one addition, subtraction, multiplication or division of two floating-point numbers.

\[ A_k(k)^{-1} \] exists because \( A_k(k) \) is a nonsingular matrix [25].
where
\[ u(k) = a_{k+1}(k+1) - a_{k+2}(k), \quad (23) \]
\[ v(k) = e_{k+1, |C|}. \quad (24) \]

**Proof:** Please see Appendix E.

Based on Lemma 4, we can compute \( \hat{x}(k) \) in (21) by Algorithm 2.

**Algorithm 2 Algorithm to Compute \( \hat{x}(k) \) in (21)**

1. if \( k = 1 \) then
2. Compute \( A_1(1)^{-1} \) using Gaussian elimination.
3. else
4. Obtain \( \hat{K}(k-1), u(k-1) \) and \( v(k-1) \) in Lemma 4.
5. Compute \( A_k(k)^{-1} \) based on \( A_{k-1}(k-1)^{-1} \) by (22).
6. end if
7. \( \hat{x}(k) = -A_k(k)^{-1}y(k). \)

**Remark 4 (Computational Complexity Algorithm 2):** By Algorithm 2 when \( k = 1 \), the computation of \( \hat{x}(k) \) in (21) requires \( 8(|C| - 1)^3/3 + 2(|C| - 1)^2 \) flops. When \( k = 2, 3, \ldots, |C| \), step 4 and 5 require \( |C| - 1, 10(|C| - 1)^2 \) and \( 2(|C| - 1)^2 \) flops, respectively, i.e., the computation of \( \hat{x}(k) \) requires \( 12(|C| - 1)^2 + |C| - 1 \) flops. Therefore, Algorithm 2 requires \( 44(|C| - 1)^3/3 + 3(|C| - 1)^2 \) flops, i.e., has complexity \( O(|C|^3) \).

By comparing Remarks 3 and 4 we can see that, to compute \( \hat{x}(k) \) for all \( k \), the complexity using Algorithm 2 \( O(|C|^3) \) is lower than that using the brute-force algorithm \( O(|C|^4) \). This is because the brute-force algorithm cannot make use of the special structure of \( P(k) \), and hence has higher computational complexity.

By implementing step 5 in Algorithm 1 using Algorithm 2 we can compute \( \pi(k) \) for all \( k \) iteratively. Therefore, we can develop Algorithm 3 to solve Problem 2.

**Algorithm 3 Algorithm to Compute \( q_{th}^* \) for Problem 2**

1. initialize \( q_{th}^* = 0, \ temp = 0. \)
2. for \( k = 1 \colon |C| \) do
3. \( q_{th}^* \leftarrow c_k. \)
4. Compute \( r(q_{th}) \) by (16).
5. Compute \( \pi(q_{th}) \) by Algorithm 1 wherein step 5 is implemented by Algorithm 2.
6. if \( \pi(q_{th})r(q_{th}) \geq \ temp \) then
7. \( \ temp \leftarrow \pi(q_{th})r(q_{th}), q_{th}^* \leftarrow q_{th}. \)
8. end if
9. end for

**VI. OPTIMAL SOLUTIONS FOR SPECIAL CASE**

In this section, we first obtain the corresponding static optimization problem for the symmetric case \((R_s = R_r = R, N_s = nR)\) and \((p_s = p_r = p)\). Then, we derive its closed-form optimal solution.

By Lemma 2, the recurrent class of \( \{Q_t\} \) is given by \( C = \{0, R, 2R, \ldots, nR\} \). Fig. 5 illustrates the corresponding transition diagram. By applying the Perron-Frobenius theorem and the detailed balance equations (24), we obtain the steady-state probability:

\[ \pi_0(m) = \frac{\bar{p}^{2m+2} - \bar{p}^{2m+1}}{\bar{p}^{2m+2} + \bar{p}^{2m+2} - 2\bar{p}^{2m+1}}, \quad (25a) \]
\[ \pi_{i+1}(m) = \begin{cases} \frac{1}{p} \pi_i(m), & 0 \leq i \leq m - 1; \\ \pi_i(m), & i = m; \\ \frac{p}{p} \pi_i(m), & m + 1 \leq i \leq n - 1. \end{cases} \quad (25b) \]

where \( \bar{p} = 1 - p \). Then, in the symmetric case, Problem 2 is equivalent to the following optimization problem.

**Problem 3 (Optimization for Symmetric Case):** In the symmetric case, any threshold

\[ Q_{th}^* \in \{ \frac{n\bar{R}}{2} - R, \frac{n\bar{R}}{2} - R + 1, \ldots, \frac{n\bar{R} + R - 1}{2} \}, \quad n \text{ odd} \]
\[ Q_{th}^* \in \{ \frac{n\bar{R}}{2} - R, \frac{n\bar{R}}{2} - R + 1, \ldots, \frac{n\bar{R} + R - 1}{2} \}, \quad n \text{ even} \quad (27) \]

achieves the optimal value to Problem 1.

**Proof:** Please see Appendix F.

** VII. NUMERICAL RESULTS AND DISCUSSION**

In this section, we verify the analytical results and evaluate the performance of the proposed optimal solution via numerical examples. In the simulations, we choose \( p_s = p_r = 0.5 \).

**A. Threshold Structure of Optimal Policy**

Fig. 6(a) illustrates the value function \( V(Q) \) versus \( Q \). \( V(Q) \) is computed numerically using the relative value iteration algorithm [14]. It can be seen that \( V(Q) \) is increasing with \( Q \) and \( V(Q + 1) - V(Q) \leq 1 \), which verify 1) and 2) in Lemma 1 respectively. The third property of Lemma 1 can also be verified by checking the simulation points. Fig. 6(b) illustrates the function \( \Delta J(Q) = J(Q, 1) - J(Q, 0) \) versus \( Q \). Observe that \( \Delta J(Q) \) is increasing with \( Q \). This means that the state-action reward function \( J(Q, a_r) \) is supermodular in \( (Q, a_r) \), which implies the threshold-based structure of the optimal link selection [17]. This verifies Theorem 2. Moreover, for the symmetric case, \( Q_{th}^* = 10 = \frac{n\bar{R}}{2} \) verifies Lemma 5.
B. Throughput Performance

We compare the throughput performance of the proposed optimal policy (given in Theorems 1 and 2) with four baseline schemes: DOPNF [4], ADOP [5], TOP [10] and OPDU [5]. OPDU only depends on the CSI, while the other three baseline schemes depend on both of the CSI and QSI. Specifically, the threshold in DOPNF ($Q_{th}=0$) is fixed; the threshold in ADOP ($Q_{th}=R_s$) depends on $R_t$; the threshold in TOP ($Q_{th} = N_r/2$) depends on $N_r$.

Fig. 7(a) and Fig. 7(b) illustrate the average system throughput versus the maximum transmission rate and the relay buffer size, respectively, in the asymmetric case ($R_s \neq R_t$). Since DOPNF, ADOP, TOP and the proposed optimal policy depend on both of the CSI and QSI, they can achieve better throughput performance than OPDU in most cases. Moreover, as the threshold in the proposed policy also depends on $R_s$, $R_t$ and $N_r$, it outperforms all the baseline schemes. In summary, the proposed optimal policy can make better use of the system information and system parameters, and hence achieves the optimal throughput. Specifically, the performance gains of the proposed policy over DOPNF, ADOP, TOP and OPDU are up to 15%, 10%, 80% and 20%, respectively. Besides, the performance of TOP relies heavily on the choice for the parameter $R_{max}$ (the maximum admitted rate), which is not specified in [10].

(a) Throughput versus $R_s$, $N_r = 50$ packets. (b) Throughput versus $N_r$, $R_s = 3$ packets/slot.

Fig. 7: Throughput for different schemes in the asymmetric case ($R_s \neq R_t$). $R_s/R_t=3/2$. The unit of $R_{max}$ is packet/slot.

C. Computational Complexity

Table II illustrates the average Matlab computation time of different algorithms in the asymmetric case ($R_s \neq R_t$). It can be seen that, our proposed Algorithm [5] achieves the lowest computational complexity. Specifically, the numerical algorithms (i.e., policy iteration and relative value iteration) designed for the stochastic optimization problem (Problem 1) have much higher computational complexity than the algorithms (i.e., the brute-force algorithm and Algorithm 3) for the static optimization problem (Problem 2). In addition, for Problem 2, the complexity of the brute-force algorithm is higher than that of the proposed Algorithm 3 and the complexity gap between them is increasing with $|C|$ rapidly. This verifies the discussions in Remarks 4 and 5

Table II illustrates the average Matlab computation time of different algorithms in the symmetric case ($R_s = R_t$). It can be seen that, the numerical algorithms for Problem 1 have much higher computational complexity than the proposed solution for Problem 2. Note that, in the symmetric case, Problem 2 has a closed-form solution, as shown in Lemma 5. Thus, the computation time of the closed-form solution is negligible and does not change with $|C|$.

VIII. CONCLUSION

In this paper, we consider the optimal control to maximize the average system throughput for a two-hop half-duplex relay system with random channel connectivity and a finite relay buffer. We formulate the stochastic optimization problem as MDP. Then, we show that the optimal link selection policy has a threshold-based structure. Based on this structural property, we simplify the MDP to a static discrete optimization problem and propose a low-complexity algorithm to obtain the optimal threshold. Furthermore, we obtain the closed-form optimal threshold for the symmetric case.
Table I: Average Matlab computation time (sec) comparison for different algorithms in the asymmetric case ($R_s \neq R_r$). $R_s = 4$ packets/slot and $R_r = 2$ packets/slot. Policy iteration and relative value iteration are two standard numerical algorithms to solve the stochastic optimization problem (Problem 1) based on the equivalent Bellman equation in (6). The brute-force algorithm and Algorithm 3 are designed to solve the static optimization problem (Problem 2), as illustrated in Section V.B.

Table II: Average Matlab computation time (sec) comparison for different algorithms in the symmetric case ($R_s = R_r$). $R_s = R_r = 2$ packets/slot. The static optimization problem (Problem 3) has a closed-form solution, as shown in Lemma 5.

**Appendix A: Proof of Theorem 1**

First, using sample path arguments, we show that the link selection and transmission rate control policy $\Omega^* = (\Omega^*_s, \Omega^*_r)$ is optimal, where $\Omega^*_s$ and $\Omega^*_r$ satisfy the structures in (7) and (5), respectively.

Consider any stationary link selection and transmission rate control policy $\Omega = (\Omega_s, \Omega_r)$ satisfying Definition 1. Let $\{G_t\}$ be a given CSI sample path. Denote $(a_{s,t}, r_{t}, a_{r,t})$ and $(u_{s,t}, u_{r,t})$ be the link selection and transmission rate control actions at slot $t$ under $\Omega$, respectively. Let $\{Q_t\}$ be the associated trajectory of QSI which evolves according to (1) with $\{(a_{s,t}, r_{t}, a_{r,t})\}$ and $\{(u_{s,t}, u_{r,t})\}$. Denote $(a^*_{s,t}, r^*_{t}, a^*_{r,t})$ and $(u^*_{s,t}, u^*_{r,t})$ be another link selection and transmission rate control action at slot $t$, respectively. Let $\{Q^*_t\}$ be the associated trajectory of QSI which evolves according to (1) with $\{(a^*_{s,t}, r^*_{t}, a^*_{r,t})\}$ and $\{(u^*_{s,t}, u^*_{r,t})\}$, Assume $Q^*_1 = Q_1$. The relationship between $(a^*_{s,t}, r^*_{t}, a^*_{r,t})$ and $(a_{s,t}, r_{t}, a_{r,t})$ is given by

\[
(a^*_{s,t}, r^*_{t}, a^*_{r,t}) = \begin{cases} (0, 1) \text{ or } (1, 0), & \text{if } G_t = (1, 1) \\
(a_{s,t}, r_{t}, a_{r,t}) = (0, 0) & \text{otherwise} \end{cases}
\]  

(28)

$(u^*_{s,t}, u^*_{r,t})$ satisfies the structure in (5), i.e., $u^*_{s,t}$ and $u^*_{r,t}$ are the minimum of $\{R_s, N_r - Q^*_t\}$ and $\{R_r, Q^*_t\}$, respectively. We shall show that the throughput under $\{(a^*_{s,t}, r^*_{t}, a^*_{r,t})\}$ and $\{(u^*_{s,t}, u^*_{r,t})\}$ is no smaller than that under $\{(a_{s,t}, r_{t}, a_{r,t})\}$ and $\{(u_{s,t}, u_{r,t})\}$ for a given CSI sample path $\{G_t\}$. Define $\Delta_t \triangleq \sum_{r=1}^{t} (a^*_{s,t}u^*_{s,t} - a_{s,t}u_{s,t})$. It is equivalent to prove $\Delta_t \geq 0$ for all $t$. In the following, using mathematical induction, we shall show that $\Delta_{t+1} \geq 0$ and $\Delta_{t+1} + Q^*_t + 1 \geq Q^*_t + 1$ hold for all $t$. (Note that $\Delta_t + Q^*_t + 1 \geq Q^*_t + 1$ is needed to prove $\Delta_{t+1} \geq 0$.)

Consider $t = 1$. We have $\Delta_1 = a^*_{s,1}u^*_{s,1} - a_{s,1}u_{s,1}$ and $\Delta_1 + Q^*_2 - Q_2 = a^*_{s,1}u^*_{s,1} - a_{s,1}u_{s,1} + Q^*_1 + a^*_{r,1}u^*_{r,1} - a^*_{s,1}u^*_{s,1} - (Q_1 + a_{s,1}u_{s,1} - a_{r,1}u_{r,1}) = a^*_{r,1}u^*_{r,1} - a_{r,1}u_{r,1}$. To prove $\Delta_1 \geq 0$ and $\Delta_1 + Q^*_2 \geq Q_2$, we consider the following two cases.

(1) Consider $(a^*_{s,1}, a^*_{r,1}) = (a_{s,1}, a_{r,1})$. Since $u^*_1 = \min\{R_s, N_r - Q^*_1\}$, $u_{1,1} \in \{0, 1, \cdots, \min\{R_r, Q^*_1\}\}$ and $Q^*_1 \geq Q_1$, we have $\Delta_1 = a_{r,1}(u^*_{r,1} - u_{r,1}) \geq 0$. Since $u^*_1 = \min\{R_s, N_r - Q^*_1\}$, $u_{1,1} \in \{0, 1, \cdots, \min\{R_s, N_r - Q^*_1\}\}$ and $Q^*_1 \geq Q_1$, we have $\Delta_1 + Q^*_2 - Q_2 = a_{s,1}(u^*_{s,1} - u_{s,1}) \geq 0$. We have

\[
\Delta_t \triangleq \sum_{r=1}^{t} (a^*_{s,t}u^*_{s,t} - a_{s,t}u_{s,t}) \geq 0
\]

(29)

Consider $t > 1$. Assume $\Delta_{t-1} \geq 0$ and $\Delta_{t-1} + Q^*_t \geq Q_t$ hold for some $t > 1$. Note that,

\[
\Delta_t = \Delta_{t-1} + a^*_{r,t}u^*_{r,t} - a_{r,t}u_{r,t},
\]

(30)

To show that $\Delta_t \geq 0$ and $\Delta_t + Q^*_t + 1 \geq Q_{t+1}$ also hold, we consider the following two cases.

(1) If $(a^*_{s,t}, a^*_{r,t}) = (a_{s,t}, a_{r,t})$, we consider three cases. (i) If $(a_{s,t}, a_{r,t}) = (0, 0)$, we have $\Delta_t = \Delta_{t-1} \geq 0$ and $\Delta_t + Q^*_t + 1 = \Delta_{t-1} + Q^*_t - Q_t \geq 0$. (ii) If $(a_{s,t}, a_{r,t}) = (1, 0)$, we have $\Delta_t = \Delta_{t-1} \geq 0$. Since $u_{s,t} \in \min\{R_s, N_r - Q^*_t\}$ and $u_{r,t} = u_{r,t} \in \{0, 1, \cdots, \min\{R_s, N_r - Q^*_t\}\}$, by (28), we have $\Delta_t + Q^*_t + 1 - Q_{t+1} \geq \Delta_{t-1} + Q^*_t + 1 + \min\{R_s, N_r - Q^*_t - Q_t - \min\{R_s, N_r - Q^*_t\}\} = \min\{\Delta_{t-1} + Q^*_t + 1, R_s, N_r - Q^*_t\} \geq 0$, where the last inequality is
due to the induction hypotheses. (iii) If \((a_{s,t}, a_{r,t}) = (0, 1)\), we have \(\Delta_t + Q_{t+1}^\ast - Q_{t+1} = \Delta_{t+1} + Q_t^\ast - Q_t \geq 0\). Since \(u_{r,t}^\ast \in \{0, 1, \ldots, \min\{R_t, Q_t^\ast\}\}\), by (27), we have \(\Delta_t \geq \Delta_{t+1} + \min\{R_t, Q_t^\ast\} - \min\{R_t, Q_t\} \geq 0\), where \(\Delta_{t+1} \geq 0\) due to the induction hypotheses.

(2) If \((a_{s,t}, a_{r,t}^\ast) \neq (a_{s,t}, a_{r,t})\), by (27), we have \((a_{s,t}, a_{r,t}^\ast) = (0, 1)\) or \((0, 0)\). \((a_{s,t}, a_{r,t}^\ast) = (0, 1)\) and \(G_t = (1, 1)\). By (27) and (30), we have \(\Delta_t = \Delta_{t+1} + a_{r,t}^\ast u_{s,t}^\ast \geq 0\) and \(\Delta_t + Q_{t+1}^\ast - Q_{t+1} = \Delta_{t+1} + Q_t^\ast - Q_t + a_{s,t}^\ast u_{s,t}^\ast \geq 0\), where the two inequalities are due to the induction hypotheses.

Thus, we show that \(\Delta_t \geq 0\) and \(\Delta_t + Q_{t+1}^\ast - Q_{t+1} \geq \Delta_{t+1}\) again hold. By induction, \(\Delta_t \geq 0\) hold for all \(t\) which leads to

\[
\frac{1}{T} \sum_{t=1}^{T} a_{r,t}^\ast u_{s,t}^\ast \geq \frac{1}{T} \sum_{t=1}^{T} a_{r,t} u_{s,t}, \forall T.
\]

By taking expectation over all sample paths, \(\max\) and \(\sup\) and optimization over all link selection and transmission rate control policy space, we have \(\max_{\Omega_t} \bar{R}^t \geq \max_{\Omega_t^\ast} \bar{R}^t\), where \(\Omega^\ast = (\Omega_t^\ast, \Omega_r^\ast)\) with \(\Omega_t^\ast\) and \(\Omega_r^\ast\) satisfying the structures in (1) and (5), respectively. In the following, we can restrict our attention to the optimal stationary policy \(\Omega^\ast\).

Problem (1) is an infinite horizon average cost MDP and we consider unichain policies. By (14), there exists \((\theta, \{V(\chi)\})\) satisfying the following Bellman equation:

\[
\theta + V(\chi) = \max_{\Omega(\chi)} \left\{ r(\chi, \Omega(\chi)) + \sum_{\chi'} \Pr[\chi' | \chi, \Omega(\chi)] V(\chi') \right\}, \forall \chi, (32)
\]

where \(\theta = \bar{R}^\ast\) is the optimal value to Problem (1) for all initial state \(\chi_1 \in \chi\) and \(V(\cdot)\) is the value function. Due to the i.i.d. property of \(G\), by taking expectation over \(G\) on both sides of (32), we have

\[
\theta + V(Q) = \sum_{g \in \mathcal{G}} \Pr(G = g) \max_{\Omega(Q)} \left\{ r(\chi, \Omega(\chi)) + \sum_{Q'} \Pr[Q' | \chi, \Omega(\chi)] V(Q') \right\}, \forall Q, (33)
\]

where \(V(Q) = \mathbb{E}[V(\chi) | Q]\). Then, by the optimal link selection and transmission rate control structure in (1) and (5), the relationship between \(Q'\) and \(Q\) via (1) and the per-stage reward \(r(\chi, \Omega(\chi))\) in (5), we have (6). We complete the proof.

**APPENDIX B: PROOF OF LEMMA 1**

We prove the three properties in Lemma 1 using the relative value iteration algorithm and mathematical induction.

First, we introduce the relative value iteration algorithm (14). For each \(Q \in \mathcal{Q}\), let \(V_n(Q)\) be the value function in the \(n\)th iteration, \(n = 0, 1, \cdots\). Define

\[
J_{n+1}(Q, a_{r,n}) = \max_{a_{r,n}} \left\{ \sum_{s} \left[ \alpha_{s,n}^\ast \left(Q_n + R_s + R_r \right) \right] + \beta_{s,n} V_n(Q_n) \right\}, \forall Q_n \in \mathcal{Q}, (34a)
\]

where \(1(\cdot)\) denotes the indicator function. Note that \(J_{n+1}(Q, a_{r,n})\) is related to the R.H.S of the Bellman equation in (6). We refer to \(J_{n+1}(Q, a_{r,n})\) as the state-action reward function in the \(n\)th iteration (19). By using (34), for each \(Q\), the relative value iteration algorithm calculates \(V_{n+1}(Q)\) as

\[
V_{n+1}(Q) = \max_{a_{r,n}} J_{n+1}(Q, a_{r,n}) - \max_{a_{r,n}} J_{n+1}(Q_0, a_{r,n}), \forall n
\]

where \(Q_0 \in \mathcal{Q}\) is some fixed state. Under any initialization of \(V_0(Q)\), the generated sequence \(\{V_n(Q)\}\) converges to \(V(Q)\) (14), i.e.,

\[
\lim_{n \to \infty} V_n(Q) = V(Q), \forall Q \in \mathcal{Q}. (36)
\]

where \(V(Q)\) satisfies the Bellman equation in (6).

In the following proof, we set \(V_0(Q) = 0\) for all \(Q\). Let \(\alpha^\ast_{r,n}(Q)\) denote the control that attains the maximum of the first term in (35) in the \(n\)th iteration for all \(Q\), i.e.,

\[
\alpha^\ast_{r,n}(Q) = \arg\max_{a_{r,n}} J_{n+1}(Q, a_{r,n}), \forall Q \in \mathcal{Q}. (37)
\]

We refer to \(\alpha^\ast_{r,n}(Q)\) as the optimal policy for the \(n\)th iteration. For ease of notation, in the following, we denote \((\alpha^\ast_{r,n}(Q + R_s + R_r + 1), \alpha^\ast_{r,n}(Q + R_s + R_r), \alpha^\ast_{r,n}(Q + 1), \alpha^\ast_{r,n}(Q))\) as \((\alpha_{r,n}^1(Q), \alpha_{r,n}^2(Q), \alpha_{r,n}^3(Q), \alpha_{r,n}^4(Q))\), where \(Q \in \{0, 1, \ldots, N_r - (Q + R_s + R_r + 1)\}\). Next, we prove Lemma 1 through mathematical induction.

(1) We prove Property 1 by showing that for all \(n = 0, 1, \cdots\), \(V_n(Q)\) satisfies

\[
V_n(Q + 1) \geq V_n(Q), Q \in \{0, 1, \cdots, N_r - 1\}. (38)
\]

We initialize \(V_0(Q) = 0\), for all \(Q \in \mathcal{Q}\). Thus, we have \(V_0(Q + 1) = V_0(Q)\), i.e., (38) holds for \(n = 0\). Assume that (38) holds for \(n > 0\). We will prove that (38) also holds for \(n + 1\). By (35), we have

\[
V_{n+1}(Q + 1) = J_{n+1}(Q + 1, \alpha^\ast_{r,n}(Q) + 1) - \max_{a_{r,n}} J_{n+1}(Q_0, a_{r,n})
\]

(a) \(\geq J_{n+1}(Q + 1, \alpha^\ast_{r,n}(Q)) - \max_{a_{r,n}} J_{n+1}(Q_0, a_{r,n})\)

(b) \(= \sum_{s} \left[ \alpha^\ast_{s,n}(Q + 1) \left(Q + 1 + R_s + N_r\right) \right] + \beta_{s,n} V_n(Q + 1) + V_n(Q + 1 - R_r)^+\)

(c) \(+ \beta_{s,n} \{1(\alpha^\ast_{1,n}(Q) = 0) V_n(Q + 1 + R_s + N_r) \}

(d) \(+ 1(\alpha^\ast_{1,n}(Q) = 1) \left(Q + 1 + R_s\right) + V_n(Q + 1 - R_r)^+\})

- \max_{a_{r,n}} J_{n+1}(Q_0, a_{r,n}), (39)\)

We complete the proof.
where (a) follows from the optimality of $\alpha^*_n(Q)$ for $Q + 1$ in the $n$th iteration and (b) directly follows from (34b) and (35). By (34b) and (35), we also have

$$V_{n+1}(Q) = J_{n+1}(Q, \alpha^*_n(Q)) - \max_{a_{r,n}} J_{n+1}(Q_0, a_{r,n})$$

$$= \bar{p}_s \bar{p}_r V_n(Q) + p_s \bar{p}_r \min \{Q + R_s, N_r\}$$

$$+ \bar{p}_s p_r \{\min \{Q, R_r\} + V_n([Q - R_r]^+)\}$$

$$+ p_s p_r \{1 (\alpha^*_n(Q) = 0) V_n(\min \{Q + R_s, N_r\})$$

$$+ 1 (\alpha^*_n(Q) = 1) \{\min \{Q, R_r\} + V_n([Q - R_r]^+)\} \}$$

$$- \max_{a_{r,n}} J_{n+1}(Q_0, a_{r,n}).$$  \hspace{1cm} (40)

Next, we compare (39) and (40) term by term. By the facts that \(\min \{Q + 1 + R_s, N_r\} \geq \min \{Q + R_s, N_r\}, [Q + 1 - R_r]^+ \geq [Q - R_r]^+\) and \(\min \{Q + 1, R_r\} \geq \min \{Q, R_r\}\), and the induction hypothesis, we have \(V_{n+1}(Q + 1) \geq V_n(Q)\), i.e., (38) holds for \(n + 1\). Therefore, by induction, (38) holds for any \(n\). By taking limits on both sides of (38) and by (35), we complete the proof of Property 1.

(2) We prove Property 2 by showing that for all \(n = 0, 1, \ldots, V_n(Q)\) satisfies

$$V_n(Q + 1) - V_n(Q) \leq 1, Q \in \{0, 1, \ldots, N_r - 1\}. \hspace{1cm} (41)$$

We initialize \(V_0(Q) = 0\), for all \(Q \in Q\). Thus, we have \(V_0(Q + 1) - V_0(Q) = 0\), i.e., (41) holds for \(n = 0\). Assume that (41) holds for \(n > 0\). We will prove that (41) also holds for \(n + 1\). By (38) and (34b), we have,

$$V_{n+1}(Q + 1) - V_n(Q)$$

$$= J_{n+1}(Q + 1, \alpha_2^*(Q)) - J_{n+1}(Q, \alpha_1^*(Q))$$

$$= [J_{n+1}(Q + 1, \alpha_2^*(Q)) - J_{n+1}(Q, \alpha_2^*(Q))]$$

$$+ [J_{n+1}(Q, \alpha_2^*(Q)) - J_{n+1}(Q, \alpha_1^*(Q))]$$

$$\leq J_{n+1}(Q + 1, \alpha_2^*(Q)) - J_{n+1}(Q, \alpha_2^*(Q))$$

$$= p_s \bar{p}_r A_1 + p_s \bar{p}_r B_1 + \bar{p}_s p_r C_1 + p_s p_r D_1, \hspace{1cm} (42)$$

where (c) is due to \(J_{n+1}(Q, \alpha_2^*(Q)) \leq J_{n+1}(Q, \alpha_1^*(Q))\). This is because \(\alpha_1^*(Q)\) is the optimal policy for \(Q\) in the \(n\)th iteration. \(A_1, \ B_1, \ C_1\) and \(D_1\) are given as follows.

$$A_1 = V_n(Q + 1) - V_n(Q), \hspace{1cm} (43a)$$

$$B_1 = V_n(\min \{Q + 1 + R_s, N_r\}) - V_n(\min \{Q + R_s, N_r\}), \hspace{1cm} (43b)$$

$$C_1 = \min \{Q + 1, R_r\} + V_n([Q + 1 - R_r]^+)$$

$$- \min \{Q, R_r\} - V_n([Q - R_r]^+), \hspace{1cm} (43c)$$

$$D_1 = 1 (\alpha_2^*(Q) = 0) B_1 + 1 (\alpha_2^*(Q) = 1) C_1. \hspace{1cm} (43d)$$

Note that \(p_s \bar{p}_r = 1\) and \(p_r \bar{p}_r = 1\). Thus, to show \(V_{n+1}(Q + 1) - V_n(Q) \leq 1\) using (42), it suffices to show that \(A_1 \leq 1, B_1 \leq 1, C_1 \leq 1\) and \(D_1 \leq 1\). Due to the induction hypothesis, \(A_1 \leq 1\) and \(B_1 \leq 1\) hold. To prove \(C_1 \leq 1\), we consider the following two cases. (i) When \(Q \geq R_r\), we have \(C_1 = V_n(Q + 1 - R_r) - V_n(Q - R_r) \leq 1\) due to the induction hypothesis. (ii) When \(Q \leq R_r\), we have \(C_1 = 1\). Thus \(C_1 \leq 1\) holds. To prove \(D_1 \leq 1\), we consider the following two cases. (i) If \(\alpha_2^*(Q) = 0\), we have \(D_1 = B_1 \leq 1\). (ii) If \(\alpha_2^*(Q) = 1\), we have \(D_1 = C_1 \leq 1\). Thus, we can show that (41) holds for \(n + 1\). Therefore, by induction (41) holds for any \(n\). By taking limits on both sides of (41) and by (36), we complete the proof of Property 2.

(3) We prove Property 3 by showing that for all \(n = 0, 1, \ldots, V_n(Q)\) satisfies

$$V_n(Q + R_s + R_r + 1) - V_n(Q + R_s + R_r) \leq V_n(Q + 1) - V_n(Q), \hspace{1cm} Q \in \{0, 1, \ldots, N_r - (Q + R_s + R_r + 1)\}. \hspace{1cm} (44)$$

We initialize \(V_0(Q) = 0\), for all \(Q \in Q\). Thus, we have \(V_0(Q + R_s + R_r + 1) - V_0(Q + R_s + R_r) = V_0(Q + 1) - V_0(Q) = 0\), i.e., (44) holds for \(n = 0\). Assume that (44) holds for \(n > 0\). We will prove that (44) also holds for \(n + 1\). By (35), we have,
and
\[ J_{n+1} \left( Q+1, \alpha_{n+1}^*(Q) \right) - J_{n+1} \left( Q, \alpha_n^*(Q) \right) = \bar{p}_s \bar{p}_r A_1 + p_s \bar{p}_r B + \bar{p}_s p_r C_1 + p_s p_r D_1', \tag{49} \]
where \( A_1, B_1 \) and \( C_1 \) are given by \( (43a), (43b) \) and \( (43c) \), respectively, and
\[ D_1' = 1 \left( \alpha_{n+1}^*(Q) = 0 \right) B_1 + 1 \left( \alpha_n^*(Q) = 1 \right) C_1. \]

Note that, when \( Q \in \{0, 1, \ldots, N_r - (Q + R_s + R_r + 1)\} \), \( \alpha_n^*(Q) \) can be rewritten as \( B_1 = V_n(Q + 1 + R_r) - V_n(Q + R_s + 1) \).

To show that \( (44) \) holds for \( n + 1 \) using \( (47) \) and \( (49) \), it suffices to show that \( A_2 \leq A_1, B_2 \leq B_1, C_2 \leq C_1 \) and \( D_2 \leq D_1' \). Due to the induction hypothesis, \( A_2 \leq A_1 \) holds. To prove \( B_2 \leq B_1 \), we consider the following two cases. (i) When \( Q + 2 R_s + R_r \geq N_r \), we have \( B_2 - B_1 = V_n(Q + R_s) - V_n(Q + R_s + 1) \leq 0 \) as \( \alpha_n^*(Q) \) holds.

(ii) When \( Q + 2 R_s + R_r + 1 \leq N_r \), we have \( B_2 - B_1 = V_n(Q + 2 R_s + R_r + 1) - V_n(Q + 2 R_s + R_r) - (V_n(Q + R_s + 1) - V_n(Q + R_s)) \leq 0 \) due to the induction hypothesis. Thus, \( B_2 \leq B_1 \) holds. To prove \( C_2 \leq C_1 \), we consider two cases. (i) When \( Q \leq R_r - 1 \), we have \( C_2 - C_1 = V_n(Q + R_s + 1) - V_n(Q + R_s) - 1 \leq 0 \) as \( (41) \) holds for \( n \), (ii) When \( Q \geq R_r \), we have \( C_2 - C_1 = V_n(Q + R_s + 1) - V_n(Q + R_s + 1) \leq 0 \) due to the induction hypothesis. Thus, \( C_2 \leq C_1 \) holds. To prove \( D_2 \leq D_1' \), we consider the following four cases. (i) If \( (\alpha_n^*(Q), \alpha_n^*(Q)) = (0, 0) \), we have \( D_2 - D_1' = 0 \).

(ii) If \( (\alpha_n^*(Q), \alpha_n^*(Q)) = (1, 0) \), we have \( D_2 - D_1' = C_2 - B_1 = 0 \).

(iii) If \( (\alpha_n^*(Q), \alpha_n^*(Q)) = (1, 1) \), we have \( D_2 - D_1' = C_2 - C_1 \leq 0 \).

(iv) If \( (\alpha_n^*(Q), \alpha_n^*(Q)) = (0, 1) \), we have \( D_2 - D_1' = B_2 - C_2 \leq 0 \). Therefore, by induction \( (44) \) holds for \( n + 1 \). By taking limits on both sides of \( (44) \) and \( (46) \), we complete the proof of Property 3.

APPENDIX C: PROOF OF THEOREM

First, we show that \( J(Q, a_r) \) in \( (9) \) is supermodular in \((Q, a_r)\). By the definition of supermodularity, it is equivalent to prove \( \Delta J(Q + 1) - \Delta J(Q) \geq 0 \), where \( \Delta J(Q) = (J(Q, 1) - J(Q, 0)) / p_s p_r \). By \( (2) \), we have
\[
\Delta J(Q + 1) - \Delta J(Q) = \min \{Q + 1, R_r\} - \min \{Q, R_r\} + V(Q + 1 - R_r) - V(Q - R_r) - V(Q + 1 - R_s, N_r) + V(Q + R_s, N_r).
\tag{50}
\]
To prove \( \Delta J(Q + 1) - \Delta J(Q) \geq 0 \) using \( (50) \), we consider the following two cases.

(1) If \( N_r \leq R_s + R_r \), we consider three cases. (i) When \( Q \geq R_r \), then \( \Delta J(Q + 1) - \Delta J(Q) = V(Q + R_r) - V(Q + R_r) \geq 0 \).

(ii) When \( N_r - R_s \leq Q \leq R_r - 1 \), then \( \Delta J(Q + 1) - \Delta J(Q) = 1 \).

(iii) When \( Q \leq N_r - R_s - 1 \), then \( \Delta J(Q + 1) - \Delta J(Q) = 1 \).

(2) If \( N_r \geq R_s + R_r + 1 \), we consider the similar three cases.

APPENDIX E: PROOF OF LEMMA

For two adjacent thresholds \( c_k \) and \( c_{k+1} \), the corresponding transition probability matrices \( P(k) \) and \( P(k + 1) \) only differ in the \((k + 1)\)-th row. Thus, \( A(k) \) and \( A(k + 1) \) only differ in the \((k + 1)\)-th column. Then, by partitioning \( A(k) \) and \( A(k + 1) \) into the form \( (20) \) using the permutation matrices \( K(k) \) and \( K(k + 1) \), respectively, we obtain the corresponding submatrices \( A_k(k) \) and \( A_{k+1}(k + 1) \). By exchanging the \((k + 1)\)-th and \((k + 2)\)-th columns of \( A_k(k) \), we obtain \( A_{k+1}(k) \), i.e.,
\[
A_{k+1}(k) = A_k(k) K(k),
\tag{51}
\]
where \( K(k) \) is the corresponding permutation matrix defined in Lemma \( (44) \). Thus, \( A_{k}(k) \) and \( A_{k+1}(k + 1) \) only differ in
the \((k+1)\)-th column, and \(A_{k+1}(k+1)\) can be regarded as a rank-one update of \(A_k\). Let
\[
u(k) = a_{k+1}(k+1) - \hat{a}_{k+1}(k),
\]
(52)
where \(a_{k+1}(k+1)\) and \(\hat{a}_{k+1}(k)\) are the \((k+1)\)-column of \(A_{k+1}(k+1)\) and \(A_k\), respectively. Then, we have \(A_{k+1}(k+1) = \hat{A}_k + \nu(k)v(k)^T\), where \(v(k)\) is defined in (24). By the Sherman-Morrison formula [27], we have
\[
A_{k+1}(k+1)^{-1} = \hat{A}_k(k)^{-1} - \frac{\hat{A}_k(k)^{-1}u(k)v(k)^T\hat{A}_k(k)^{-1}}{1 + v(k)^T\hat{A}_k(k)^{-1}u(k)},
\]
(53)
By (51), we have \(\hat{A}_k(k)^{-1} = \hat{K}(k)\hat{A}_k(k)^{-1}\) and \(\hat{a}_{k+1}(k) = a_{k+2}(k)\). Thus, (52) is equivalent to (23) and (53) is equivalent to (22). We complete the proof.

APPENDIX F: PROOF OF LEMMA 5

First, we show that Problem 2 can be equivalently transformed to Problem 3. Given a CSI sample path \(\{G_t\}\), let \(\{a_{s,t}, a_{r,t}\}\) and \(\{a_{s,t}, a_{r,t}\}\) be the sequences of link selection and transmission rate actions under a policy \(\Pi\) in Definition 1 respectively. Let \(\{Q_t\}\) be the associated QSI trajectory. By (1), we have
\[
\frac{1}{T} \sum_{t=1}^{T} a_{s,t}u_{s,t} = \frac{1}{T} \sum_{t=1}^{T} a_{r,t}u_{r,t} + \frac{Q_t}{T}Q_t\text{ for all } T.
\]
Since \(Q_t, Q_t \leq N_r\), by taking expectation over all sample paths and limit sup, we have
\[
\limsup_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} a_{s,t}u_{s,t} = \limsup_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} a_{r,t}u_{r,t} = \bar{R}^\Pi.
\]
Hence, under the optimal transmission rate control in (5) and a threshold-based link selection policy in (12), the average arrival rate equals to the average departure rate. Therefore, without loss of optimality, we can also denote \(r_i(q_t)\) as the average of the average arrival and departure rates at state \(i \in \Omega\) under the threshold \(q_t\). Then, in the symmetric case, we have
\[
r_i(m) = \begin{cases} \frac{pr_i}{2}, & i = 0, n; \\ \frac{(n-p)r_i}{2}, & i = 1, \ldots, n - 1. \end{cases}
\]
(54)
where state \(i \in \{0, 1, \ldots, n\}\) represents state \(iR \in \mathcal{C}\) and \(m = q_t/R\). By (54), we obtain the ergodic system throughput
\[
\bar{r}(m) = \frac{pr_i}{2}(\frac{n}{2})^n = \frac{n}{2} p(\pi_0(0)m + \pi_n(m)).
\]
Maximizing \(\bar{r}(m)\) is equivalent to minimizing \(\pi_0(0)m + \pi_n(m)\). Therefore, by (25), we have the discrete optimization problem in Problem 3.

Next, we obtain the optimal solution to Problem 3. By change of variables in Problem 3 i.e., letting \(x = \hat{p}^m\), we have the following continuous optimization problem.
\[
\min_{x \in [0,1]} g(x) = \frac{\hat{p}^{m+1} - \hat{p}^m + \hat{p}^2x^2 - \hat{p}x^2}{(x^2 + \hat{p}^2 - 2x\hat{p})^2/2}.
\]
(55)
Letting the derivative of \(g(x)\), i.e.,
\[
g'(x) = \frac{2\hat{p}^2(x^2 - \hat{p}^2)}{(x^2 + \hat{p}^2 - 2x\hat{p})^2/2},
\]
equal to 0, we have \(x^* = \hat{p}^2\). Since \(g'(-\hat{p}) \leq 0 \in [\hat{p}, \hat{p}^2]\), and \(g'(-\hat{p}^2) \geq 0 \in [\hat{p}^2, 1], x^* = \hat{p}^2\) is the optimal solution to (55). Based on \(x^*\), we now obtain the optimal solution \(m^*\) to Problem 3. If \(n\) is odd, then \(m^* = \frac{n-1}{2}\), if \(n\) is even, \(m^* = \frac{n}{2} - 1\) or \(\frac{n}{2}\). (Note that \(\frac{n}{2} - 1\) and \(\frac{n}{2}\) achieve the same optimal value of Problem 3). Then, by Lemma 5 we have (27) which completes the proof.