

Research Article

Delay-Aware Online Service Scheduling in High-Speed Railway Communication Systems

Shengfeng Xu, Gang Zhu, Chao Shen, Yan Lei, and Zhangdui Zhong

State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China

Correspondence should be addressed to Chao Shen; shenchao@bjtu.edu.cn

Received 24 November 2013; Accepted 12 February 2014; Published 24 March 2014

Academic Editor: Hao Shen

Copyright © 2014 Shengfeng Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We investigate the downlink service scheduling problem in relay-assisted high-speed railway (HSR) communication systems, taking into account stochastic packet arrivals and quality-of-service (QoS) requirements. The scheduling problem is formulated as an infinite-horizon average cost constrained Markov decision process (MDP), where the scheduling actions depend on the channel state information (CSI) and the queue state information (QSI). Our objective is to find a policy that minimizes the average end-to-end delay through scheduling actions under the service delivery ratio constraints. To address the challenge of centralized control and high complexity of traditional MDP approaches, we propose a distributed online scheduling algorithm based on approximate MDP and stochastic learning, where the scheduling policy is a function of the local CSI and QSI only. Numerical experiments are carried out to show the performance of the proposed algorithm.

1. Introduction

Recently, high-speed railway (HSR) systems have developed rapidly all over the world. The passengers on the train will not only enjoy the short journey but also have a high demand on multimedia services. The cellular network deployed along the rail lines can provide seamless coverage and data packets delivery. However, the data transmission rate is strictly limited due to the penetration loss in traditional HSR communication systems. As an alternative solution, the relay-assisted HSR network architecture has been proposed in [1, 2], which was considered a better choice than direct transmission in case of large penetration loss [3] and becomes a promising architecture for future broadband mobile communications in providing high data-rate services [4].

We consider a relay-assisted two-hop HSR network architecture in this work. The data packets are delivered via a relay station (RS) instead of direct transmission to achieve a high data transmission rate. The passengers send service requests when the train is moving. If a large number of services are requested, the resource contention among multiple services should be resolved and an efficient scheduling scheme should be proposed, which not only considers the highly dynamic channel due to the extremely high moving speed but can also

be implemented in a distributed manner with low complexity. In addition, the buffering at network devices, for example, content server and RS, is involved; it is thus important to consider not only the throughput performance but also the end-to-end (e2e) delay performance. Delay is a key quality-of-service (QoS) criterion for real-time multimedia services. As a result, we will focus on the delay issues for multimedia services and aim at developing a delay-aware scheduling algorithm for the relay-assisted HSR communication systems.

Many of the previous studies have been conducted to improve performance on scheduling and resource allocation in HSR communication systems. In order to support the e2e real-time data application, [5] studied a circuit domain latency model and employed a priority scheduling algorithm to estimate approximate service latency. In HSR networks with a cell array architecture, [6] proposed a scheduling and resource allocation mechanism to maximize the service rate by considering the channel variations and handover information. The optimal resource allocation problem in a cellular/infostation integrated HSR network has been investigated in [7], considering the intermittent network connectivity and multiservice demands. However, to the best of our knowledge, none of them has addressed the delay-aware scheduling in downlink relay-assisted HSR communication systems.

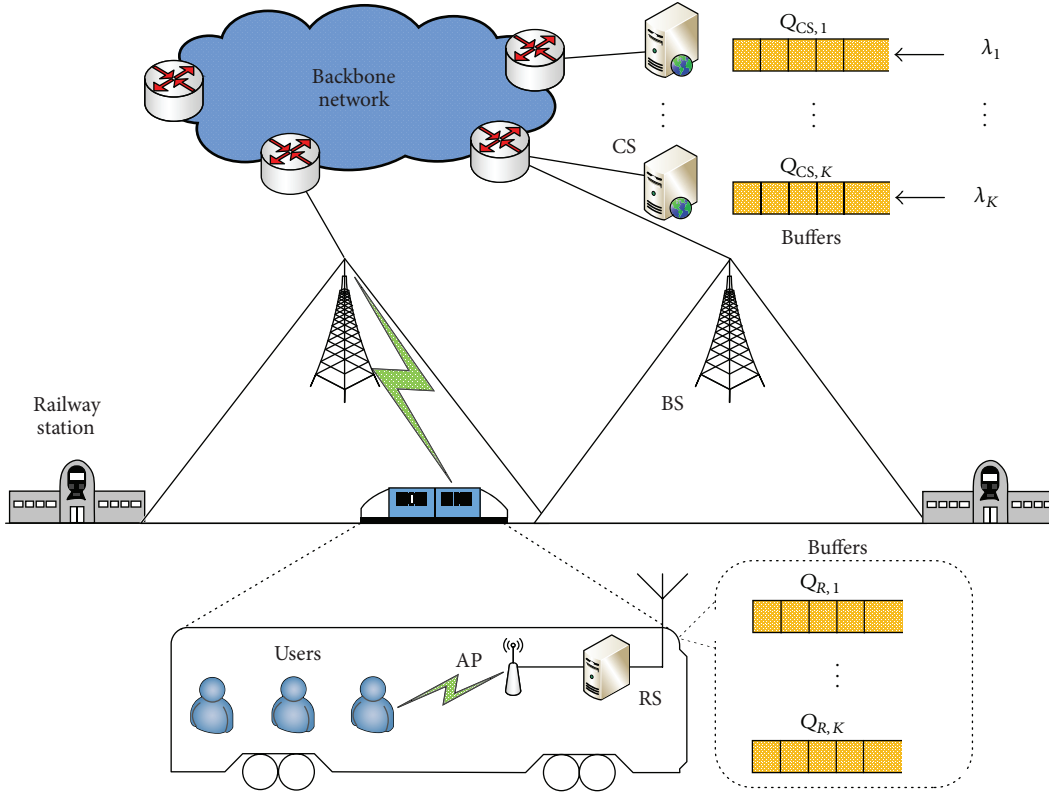


FIGURE 1: System model.

Although the two-hop HSR network architecture is simple, how to schedule multiple services for such a network when taking account of stochastic packet arrivals and QoS requirements is still an open problem. This motivates us to investigate the delay-aware downlink scheduling problem for relay-assisted HSR communication systems.

The contributions of this paper are threefold. First, the two-hop scheduling problem is formulated as an infinite-horizon average cost constrained Markov decision process (CMDP) with the objective to minimize the average e2e delay under the service delivery ratio constraints. The above CMDP problem is converted into an unconstrained MDP by Lagrange theory and then the general solution of the CMDP problem is given by traditional iterative methods. Second, since the general solution could not give a simple implementable solution due to the curse of dimensionality, in order to simplify the solution and address the challenge of centralized control, we propose a distributed online scheduling algorithm based on approximate MDP and stochastic learning. A linear combination of per-node value functions is employed to approximate the value function of the associated optimality equation. Based on the per-node value functions, a distributed two-stage scheduling policy is derived, which is a function of the local state information only. Third, simulation results show that the proposed algorithm can achieve better performance in terms of e2e delay and service delivery ratio compared to conventional schemes. Moreover, the convergence of the proposed scheduling algorithm is established through simulations.

The remainder of the paper is organized as follows. Section 2 provides the details of the system model and assumptions. Section 3 presents the CMDP problem formulation and discusses the general solution. In Section 4, we propose a distributed online scheduling algorithm using approximate MDP and stochastic learning. Section 5 shows the performance evaluation results obtained through simulations. Finally, Section 6 makes some conclusions.

Notations. In this paper, $|\mathcal{A}|$ is the cardinality of set \mathcal{A} . $\mathbb{E}[\cdot]$ denotes expectation. $\Pr[\cdot]$ is the probability measure. $\mathbf{I}[\cdot]$ is an indicator function. $\lfloor x \rfloor = \max\{n \in \mathbb{Z} \mid n \leq x\}$ and $\lfloor x \rfloor^+ = \max\{x, 0\}$.

2. System Model and Assumptions

A relay-assisted HSR network architecture is shown in Figure 1. The cellular network deployed along the rail line can provide seamless coverage and data packets delivery. The base stations (BSs) are connected to the backbone network via wireline links. For simplicity, we assume that the bandwidth of the links from the backbone network to the BSs is sufficiently large so that the packets can be transmitted to BSs with a negligible delay. An RS with powerful antennas is installed on the top of the train to communicate with BSs. The RS is further connected to an access point (AP) which can be accessed by the users based on wireless local area network (WLAN) technologies. Thus, the two-hop wireless link consists of the BS-RS link and the AP-users link. With this

two-hop architecture, radio signals do not need to penetrate into the carriages, and thus the radio signal penetration loss problem is resolved.

Distributed content servers are deployed in order to offload data traffic from the backbone network [8]. When a service is requested from the passengers, the data packets can be fetched from the corresponding content server (CS). To make the analysis of such network tractable, we assume each CS can provide one type of service. Notice that this can be easily extended to the situation where each CS can provide multiple types of services while one buffer is allocated to each type of service in the CS, which will be clear later on in Section 2.2. In order to simplify the protocol design for HSR applications, erasure coding based service delivery is considered [7] and the advantage is that no retransmission scheme is required for the transmission error due to a highly dynamic wireless channel condition.

Compared to the traditional cellular networks, the deterministic train trajectory in HSR networks is a unique feature [7, 9]. The train trajectory represents the location of a train at a specific time, and BS provides the service delivery if the train is under its coverage. Since the train moves on a pre-determined rail line and the velocity is relatively steady, the information of train trajectory can be obtained in advance with high accuracy so that the service packets can be delivered by the specific BS at a certain time. Therefore, this paper focuses on the delay-aware service scheduling problem regardless of which BS is used for service delivery.

2.1. Physical-Layer Model. We consider a time-slotted system for downlink service transmission with slot period T_B . When the train moves along the railway, BS and AP can transmit simultaneously without interference by operating on different frequency bands. For the communication in the BS-RS link, we consider the MAC frame structure proposed in [10] which is specifically designed for high-speed trains with a speed of up to 360 km/h. For the communication in the AP-users link, traditional WLAN standards, for example, IEEE 802.11a/b/g, are employed since passengers within the train are relatively stationary with respect to the AP.

Let $\mathbf{H}(t) = \mathbf{H}_1(t) \cup \mathbf{H}_2(t)$ be the joint CSI, where $\mathbf{H}_1(t) = \{H_{BS,R}(t), \forall k\}$ with the channel gain $H_{BS,R}$ in the BS-RS link and $\mathbf{H}_2(t) = \{H_{R,k}(t), \forall k\}$ with the channel gain $H_{R,k}$ between the AP and the user requesting service k . The channel from BS to the RS is assumed to follow the Rician distribution [11], while the channel between AP and the users can be treated as Rayleigh fading channel [3]. We assume that the channel gains remain constant during a slot duration and change across slots in an i.i.d. manner. In addition, to enhance transmission performance, the strong channel coding is used, and hence, the maximum achievable data rate is achieved by the instantaneous mutual information. For the BS-RS link with the bandwidth W_1 , the maximum achievable data rate in bit-per-second is given by

$$R_{BS} = W_1 \log_2 \left(1 + \frac{P_{BS} |H_{BS,R}|^2}{N_0} \right), \quad (1)$$

where P_{BS} is the transmit power of the BS and N_0 is the noise power. Since all the services share the common channel in the BS-RS link, they have the same data rate R_{BS} . Likewise, for the AP-users link with the bandwidth W_2 , we have the maximum achievable data rate for service k

$$R_{R,k} = W_2 \log_2 \left(1 + \frac{P_R |H_{R,k}|^2}{N_0} \right), \quad k = 1, 2, \dots, K, \quad (2)$$

where P_R is the transmit power of the AP. In addition, we assume the packets have equal size of G bits, the capacity C_{BS} can be expressed by the maximum number of packets by converting R_{BS} through a proportionality constant; that is, $C_{BS} = \lfloor R_{BS} T_B / G \rfloor$. Similarly, $C_{R,k} = \lfloor R_{R,k} T_B / G \rfloor, \forall k$.

2.2. Packet-Level Model. Assume that there are K types of services requested by the passengers. Each type of service is allocated with two buffers, one in the k th CS denoted by $\mathbf{Q}_{CS,k}$ and the other one in the RS denoted by $\mathbf{Q}_{R,k}$, for $k = 1, \dots, K$. The two buffers for each service in the CSs and RS are in tandem. Let L_Q be the maximum number of packets for the buffers in the RS and all CSs. Let $\mathbf{Q}(t) = \mathbf{Q}_1(t) \cup \mathbf{Q}_2(t)$ be the joint QSI, where $\mathbf{Q}_1(t) = \{Q_{CS,k}(t), \forall k\}$ and $\mathbf{Q}_2(t) = \{Q_{R,k}(t), \forall k\}$. Specifically, $Q_{CS,k}(t)$ and $Q_{R,k}(t)$ denote the number of packets at the beginning of slot t in the buffer of k th CS and the k th buffer in RS, respectively.

These K heterogeneous services have different packet arrival rates and QoS requirements. Data packets from the higher layer application arrive into the buffers and are queued until they are transmitted. The packet arrival process for each type of service is assumed to be i.i.d. across slots. Let $A_k(t)$ denote the number of packets arriving into the buffer in k th CS at slot t . Suppose in general, $A_k(t)$ follows the truncated Poisson distribution $f_k(b)$ with average packet arrival rate $\lambda_k = E[A_k(t)]$ for service k . The distribution $f_k(b)$ can be given as

$$f_k(b) = \exp(-\lambda_k) \frac{\lambda_k^b}{b!}, \quad b = 0, \dots, J_k, \quad (3)$$

where J_k is found assuming $f_k(J_k) \rightarrow 0$.

The average number of arriving packets at the k th CS is given by $\lambda_k(1 - P_k^d)$, where P_k^d is the dropping probability. This is the same as the average number of packets received by the corresponding buffer in RS since the two buffers are in tandem. The sufficiently large buffer and negligible dropping probability ($P_k^d \approx 0$) assumptions are considered in this paper.

2.3. MDP Model. In this paper, the service scheduling problem for the two-hop link in HSR networks is formulated as an infinite-horizon average cost CMDP. To make the analysis of the CMDP problem tractable in the sequel, it is necessary to identify the elements of MDP model in our scheduling problem. In general, an MDP model consists of five elements: decision epochs, states, actions, cost function, and state transition probability function. We describe these elements as follows.

The scheduling decisions for the data packet delivery in the two-hop link have to be made slot by slot and the

instant slots are called decision epochs. Let \mathcal{S} and \mathcal{A} be the global state space and action space, respectively. $\mathcal{S} = \{\mathbf{S}^1, \mathbf{S}^2, \dots, \mathbf{S}^{|\mathcal{S}|}\} = \mathcal{Q} \times \mathcal{H}$, where \mathcal{Q} is the global QSI state space and \mathcal{H} is the global CSI state space. The global system state at slot t is denoted by $\mathbf{S}(t) = (\mathbf{H}(t), \mathbf{Q}(t))$. The action space is denoted by $\mathcal{A} = \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} = \{x_k \in \{0, 1\}, \forall k\}$ and $\mathcal{Y} = \{y_k \in \{0, 1\}, \forall k\}$. x_k and y_k are scheduling actions for service k in the BS-RS link and AP-users link, respectively. The scheduling action is set to be 1 if the corresponding service k is scheduled and is set to 0 otherwise. Moreover, the scheduling actions should satisfy $\sum_k x_k(t) = 1$ and $\sum_k y_k(t) = 1$ at any slot t .

Given a current state $\mathbf{S}(t)$, the scheduling action can be decided according to a stationary policy. A stationary scheduling policy (Π) map \mathcal{S} to \mathcal{A} , that is, $\Pi : \mathcal{S} \rightarrow \mathcal{A}$. Π is called feasible if the associated actions satisfy the constraints. We assume that the next state $\mathbf{S}(t+1)$ only depends on the current state and scheduling actions but not on the previous states; hence the process $\{\mathbf{S}(t)\}$ for a given control policy Π is Markovian with the following state transition probability function:

$$\begin{aligned} \Pr[\mathbf{S}(t+1) | \mathbf{S}(t), \Pi(\mathbf{S}(t))] \\ = \Pr[\mathbf{H}(t+1)] \Pr[\mathbf{Q}(t+1) | \mathbf{S}(t), \Pi(\mathbf{S}(t))], \end{aligned} \quad (4)$$

where the equality holds because of the independence of channel state update processes and queue state update processes. The queue state update processes at CSs and RS are given in (5) and (6), respectively. Consider

$$Q_{CS,k}(t+1) = \begin{cases} A_k(t) + [Q_{CS,k}(t) - C_{BS}(t)]^+, & \text{if } x_k = 1, \\ A_k(t) + Q_{CS,k}(t), & \text{if } x_k = 0, \end{cases} \quad (5)$$

$$\begin{aligned} Q_{R,k}(t+1) \\ = \begin{cases} [Q_{R,k}(t) - C_{R,k}(t)]^+, & \text{if } x_k = 0, y_k = 1, \\ \min\{C_{BS}(t), Q_{CS,k}(t)\} \\ + [Q_{R,k}(t) - C_{R,k}(t)]^+, & \text{if } x_k = 1, y_k = 1, \\ Q_{R,k}(t) \\ + \min\{C_{BS}(t), Q_{CS,k}(t)\}, & \text{if } x_k = 1, y_k = 0, \\ Q_{R,k}(t), & \text{if } x_k = 0, y_k = 0. \end{cases} \end{aligned} \quad (6)$$

Specifically, the dynamics of the buffers for all CSs in (5) are controlled by scheduling actions in the BS-RS link and the dynamics of the buffers in RS shown by (6) are controlled by scheduling actions in the two-hop link. Given a feasible policy Π , the induced Markov chain $\{\mathbf{S}(t)\}$ is ergodic and there exists a unique steady state distribution $\pi_{\mathcal{S}}$ [12].

By Little's law [13], for a sufficiently small dropping probability ($1 - P_k^d \approx 1$), the average time that a data packet of the service k spends in the e2e system is $(Q_{CS,k} + Q_{R,k})/\lambda_k$. Since the e2e delay in relay-assisted HSR networks is considered, the corresponding per-slot cost function is defined as

$$d(\mathbf{S}(t), \Pi(\mathbf{S}(t))) = \sum_{k=1}^K \frac{Q_{CS,k}(t) + Q_{R,k}(t)}{\lambda_k}. \quad (7)$$

2.4. Optimization Objective and Constraints. Our goal is to find an optimal policy Π^* so that the average e2e delay is minimized while satisfying the service delivery ratio constraints.

For any policy Π , the average e2e delay can be expressed as

$$\begin{aligned} \bar{D}(\Pi) &= \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \mathbb{E}^{\Pi} [d(\mathbf{S}(t), \Pi(\mathbf{S}(t)))] \\ &= \mathbb{E}^{\pi_{\mathcal{S}}} [d(\mathbf{S}(t), \Pi(\mathbf{S}(t)))] \end{aligned} \quad (8)$$

where $\mathbb{E}^{\pi_{\mathcal{S}}}$ denotes the expectation with respect to the induced steady state distribution $\pi_{\mathcal{S}}$. The average e2e delay expressed by (8) means the average time that a data packet of all the services spends when transmitted from the CSs to users. In the MDP model, the average e2e delay can be regarded as the expected cost function. Similarly, in order to indicate heterogeneous delivery ratio requirements for different types of services, the delivery ratio constraints can be given by

$$\begin{aligned} \bar{R}_k(\Pi) &= \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \mathbb{E}^{\Pi} [y_k(t) T_k(t)] \\ &= \mathbb{E}^{\pi_{\mathcal{S}}} [y_k(t) T_k(t)] \geq \alpha_k \lambda_k, \quad \forall k, \end{aligned} \quad (9)$$

where $T_k(t) = \min\{C_{R,k}(t), Q_{R,k}(t)\}$ is the maximum number of packets which can be successfully received for service k at slot t , and $\alpha_k \in (0, 1]$ denotes the delivery ratio requirement of service k , reflecting that the average delivery rate is proportional to the average arrival rate.

3. CMDP Problem Formulation

In this section, we formulate the delay-aware scheduling problem as an infinite-horizon average cost CMDP and discuss the general solution. The objective is to choose an optimal scheduling policy Π^* so that the expected cost function (8) is minimized subject to the service delivery ratio constraints (9), which is expressed by

$$\text{minimize } \bar{D}(\Pi) \quad (10a)$$

$$\text{subject to } \bar{R}_k(\Pi) \geq \alpha_k \lambda_k, \quad \forall k. \quad (10b)$$

This problem is an infinite-horizon average cost CMDP with system state space \mathcal{S} , action space \mathcal{A} , the state transition probability (4), and per-slot cost function (7).

3.1. Lagrangian Approach and Unconstrained MDP. The above CMDP problem can be converted into an unconstrained MDP by Lagrange theory. As demonstrated in [14, Theorem 12.7], the optimal cost and policy of the CMDP can be obtained by an unconstrained MDP and Lagrangian approach. Firstly, the Lagrangian of (10a) and (10b) is expressed as

$$\mathcal{L}(\Pi, \boldsymbol{\beta}) = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \mathbb{E}^{\Pi} [L(\mathbf{S}(t), \Pi(\mathbf{S}(t)), \boldsymbol{\beta})], \quad (11)$$

where

$$L(\mathbf{S}(t), \Pi(\mathbf{S}(t)), \boldsymbol{\beta}) = \sum_{k=1}^K \left(\frac{Q_{CS,k}(t) + Q_{R,k}(t)}{\lambda_k} - \beta_k (y_k(t) T_k(t) - \alpha_k \lambda_k) \right) \quad (12)$$

and $\boldsymbol{\beta} = [\beta_1, \beta_2, \dots, \beta_K]^T$ is the Lagrange multiplier (LM) vector. Hence, the corresponding unconstrained MDP is given by

$$G(\boldsymbol{\beta}) = \min_{\Pi} \mathcal{L}(\Pi, \boldsymbol{\beta}) = \min_{\Pi} \left\{ \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \mathbb{E}^{\Pi} [L(\mathbf{S}(t), \Pi(\mathbf{S}(t)), \boldsymbol{\beta})] \right\}, \quad (13)$$

where $G(\boldsymbol{\beta})$ gives the Lagrange dual function. As shown in [15, Theorem 2.1], there exists an LM vector $\boldsymbol{\beta} \geq \mathbf{0}$ such that Π^* minimizes $\mathcal{L}(\Pi, \boldsymbol{\beta})$ and the saddle point condition holds. Given an LM vector, the optimal policy Π^* for (13) can be obtained by solving the associated optimality equation [12, 16] as follows:

$$\theta + V(\mathbf{S}^i) = \min_{\Pi(\mathbf{S}^i)} \left\{ L(\mathbf{S}^i, \Pi(\mathbf{S}^i), \boldsymbol{\beta}) + \sum_{\mathbf{S}^j} \Pr[\mathbf{S}^j | \mathbf{S}^i, \Pi(\mathbf{S}^i)] V(\mathbf{S}^j) \right\}, \quad (14)$$

$$\forall i, j = 1, 2, \dots, |\mathcal{S}|,$$

where $V(\mathbf{S})$ is the value function of the MDP and $\Pr[\mathbf{S}^j | \mathbf{S}^i, \Pi(\mathbf{S}^i)]$ is the state transition probability which can be obtained from (4). $\theta = \min_{\Pi} \mathcal{L}(\Pi, \boldsymbol{\beta})$ is the optimal average cost per-stage and $\Pi^*(\mathbf{S}^i)$ is the optimal policy minimizing the right-hand-side of (14) at any state \mathbf{S}^i . As shown in [12], the value iteration algorithm is an efficient stable iteration algorithm to solve the optimality equation, which operates by calculating successive approximation to the value function $V(\mathbf{S})$ with computation complexity $O(|\mathcal{S}||\mathcal{S}|^2)$.

3.2. The Reduced State Optimality Equation for CMDP. The optimality equation (14) is very complicated to solve due to the huge cardinality of the system state space. In general, the channel state transition is often unavailable a priori in HSR environment. However, the statistical characteristics of the channel distribution can be obtained, which motivates us to simplify the optimality equation (14) in this subsection. By taking expectation on channel state \mathbf{H} , the reduced state optimality equation is defined as follows:

$$\theta + \bar{V}(\mathbf{Q}^i) = \min_{\Pi(\mathbf{Q}^i)} \left\{ \bar{L}(\mathbf{Q}^i, \Pi(\mathbf{Q}^i), \boldsymbol{\beta}) + \sum_{\mathbf{Q}^j} \Pr[\mathbf{Q}^j | \mathbf{Q}^i, \Pi(\mathbf{Q}^i)] \bar{V}(\mathbf{Q}^j) \right\}, \quad (15)$$

where $\bar{V}(\mathbf{Q}^i) = \mathbb{E}_{\mathbf{H}}[V(\mathbf{Q}^i, \mathbf{H}) | \mathbf{Q}^i]$ is the conditional average value function for state \mathbf{Q}^i , and $\bar{L}(\mathbf{Q}^i, \Pi(\mathbf{Q}^i), \boldsymbol{\beta}) = \mathbb{E}_{\mathbf{H}}[L(\mathbf{Q}^i, \Pi(\mathbf{Q}^i), \boldsymbol{\beta}) | \mathbf{Q}^i]$ is the conditional average cost and $\Pr[\mathbf{Q}^j | \mathbf{Q}^i, \Pi(\mathbf{Q}^i)] = \mathbb{E}_{\mathbf{H}}[\Pr[\mathbf{Q}^j | (\mathbf{Q}^i, \mathbf{H}), \Pi(\mathbf{Q}^i, \mathbf{H})]]$ is the conditional average queue state transition probability. Instead of working on the global system state, the reduced state optimality equation (15) only depends on QSI. As shown in [17], the scheduling policy obtained by solving the optimality equation in (14) is the same as that obtained by solving the reduced state optimality equation.

4. Distributed Online Scheduling Algorithm

In certain cases of practical interest, there are still three difficulties in adopting the optimal scheduling policy presented above. Firstly, solving (15) has exponential complexity. Secondly, excessive latency may not be acceptable for real-time services due to multiple iterations. Finally, the optimal policy by solving (15) is a function of global QSI, which cannot be applied for distributed implementation. To overcome the above difficulties, we adopt the key theories of MDP and stochastic approximation and propose a distributed online scheduling algorithm, which illustrates how we could utilize the techniques of approximate MDP and stochastic learning to facilitate the distributed implementation with low complexity.

4.1. Approximate MDP. To reduce the size of the state space and decentralize the service scheduling, we approximate $\bar{V}(\mathbf{Q})$ in (15) by the linear approximation of global value function, which is given below as

$$\bar{V}(\mathbf{Q}) = \sum_{k=1}^K \sum_{q=0}^{L_Q} (\bar{V}_{CS,k}(q) \mathbf{I}[Q_{CS,k} = q] + \bar{V}_{R,k}(q) \mathbf{I}[Q_{R,k} = q]), \quad (16)$$

where $\bar{V}(\mathbf{Q})$ is the global value function and $\bar{V}_{CS,k}(q)$ and $\bar{V}_{R,k}(q)$ are regarded as per-node value functions at the buffer of k th CS and the k th buffer in RS, respectively.

We note that the dimension of the value function is greatly reduced through the linear approximation. Moreover, the per-node value function can only satisfy the optimality equation (14) in some particular states $\mathcal{Q}_{\text{par}} = \{\delta_{k,q}, \phi_{k,q} | \forall k = 1, \dots, K, q = 1, \dots, L_Q\}$, where $\delta_{k,q}$ denotes \mathbf{Q}_1 with $Q_{CS,k} = q$ and $Q_{CS,k'} = 0 \forall k' \neq k$ and $\phi_{k,q}$ denotes \mathbf{Q}_2 with $Q_{R,k} = q$ and $Q_{R,k'} = 0 \forall k' \neq k$.

4.2. Distributed Scheduling Policy under MDP Approximation. Using the linear approximation in (16), we derive a distributed scheduling policy depending on the local CSI and QSI as well as the per-node value function of each buffer at RS and CSs.

Specifically, the scheduling policy in (15) can be obtained by solving the following optimization problem:

$$\begin{aligned}
\Pi^*(\mathbf{Q}^i) &= \arg \min_{\Pi(\mathbf{Q}^i)} \left\{ \bar{L}(\mathbf{Q}^i, \Pi(\mathbf{Q}^i), \boldsymbol{\beta}) \right. \\
&\quad \left. + \sum_{\mathbf{Q}^j} \Pr[\mathbf{Q}^j | \mathbf{Q}^i, \Pi(\mathbf{Q}^i)] \bar{V}(\mathbf{Q}^j) \right\} \\
&= \arg \min_{\Pi(\mathbf{Q}^i)} \left\{ \mathbb{E}_{\mathbf{H}} \left[\sum_{k=1}^K \left(\frac{Q_{CS,k}^i + Q_{R,k}^i}{\lambda_k} \right. \right. \right. \\
&\quad \left. \left. - \beta_k (y_k T_k - \alpha_k \lambda_k) \right) \right. \\
&\quad \left. + \sum_b f(b) \bar{V}(\mathbf{Q}_n^i) + \sum_{k=1}^K x_k \right. \\
&\quad \left. \times \left(\sum_b f(b) (\bar{V}_{CS,k}(Q_{CS,k}^i - C_{BS} + b) \right. \right. \\
&\quad \left. \left. - \bar{V}_{CS,k}(Q_{CS,k}^i + b)) \right) \right. \\
&\quad \left. + \sum_{k=1}^K x_k y_k (\bar{V}_{R,k}(Q_{R,k}^i - C_{R,k} + C_{BS}) \right. \\
&\quad \left. - \bar{V}_{R,k}(Q_{R,k}^i + C_{BS})) \right. \\
&\quad \left. + \sum_{k=1}^K (1 - x_k) y_k (\bar{V}_{R,k}(Q_{R,k}^i - C_{R,k}) \right. \\
&\quad \left. - \bar{V}_{R,k}(Q_{R,k}^i)) \right\}, \tag{17a} \\
&\tag{17b}
\end{aligned}$$

where $\mathbf{Q}_n^i = [Q_{CS,1}^i + b, \dots, Q_{CS,K}^i + b, Q_{R,1}^i, \dots, Q_{R,K}^i]^T$. Here we assume that the packet arrivals for all services follow the same distribution; that is, $f(b) = f_k(b) \forall k = 1, \dots, K$.

From the above formula derivation, we can obtain the optimal scheduling scheme by solving (17b). Since the solution to the first link depends on the solution to the second link and vice versa, the scheduling decisions in the two-hop link are coupled. One feasible solution can be obtained by enumeration, but it cannot be distributed. In order to obtain the scheduling policy in a distributed manner, motivated by the observation in [18], we split the optimization problem into two stages, which correspond to the BS-RS link and AP-users link, respectively. In the first stage, CSs solve a local MDP problem to determine the scheduling actions in the BS-RS link. In the second stage, by using the scheduling results in the first stage, RS determines the scheduling actions in AP-users link such that the e2e delay is minimized. Specifically, these two stages are described as follows.

4.2.1. Distributed Scheduling Policy in First Stage. With the given per-node value functions $\{\bar{V}_{CS,k}(q), \forall k\}$ and local queue state set $\mathbf{Q}_1^i = \{Q_{CS,k}^i, \forall k\}$, the scheduling policy in the first stage can be obtained by solving the following local MDP problem:

$$\begin{aligned}
\Pi^*(\mathbf{Q}_1^i) &= \arg \min_{\Pi(\mathbf{Q}_1^i)} \left\{ \bar{L}(\mathbf{Q}_1^i, \Pi(\mathbf{Q}_1^i), \boldsymbol{\beta}) \right. \\
&\quad \left. + \sum_{\mathbf{Q}_1^j} \Pr[\mathbf{Q}_1^j | \mathbf{Q}_1^i, \Pi(\mathbf{Q}_1^i)] \bar{V}(\mathbf{Q}_1^j) \right\} \tag{18a}
\end{aligned}$$

$$= \arg \min_{\Pi(\mathbf{Q}_1^i)} \left\{ \mathbb{E}_{\mathbf{H}_1} \left[\sum_{k=1}^K x_k F_1(k) \right] \right\}, \tag{18b}$$

where

$$\begin{aligned}
F_1(k) &= \sum_b f(b) [\bar{V}_{CS,k}(Q_{CS,k}^i - C_{BS} + b) - \bar{V}_{CS,k}(Q_{CS,k}^i + b)]. \tag{19}
\end{aligned}$$

Notice that the derivation in (18a) and (18b) is similar to that in (17a) and (17b). The local MDP problem (18a) and (18b) can be solved by the distributed computing $k_1^* = \arg \min_k \{F_1(k)\}$, which implies $x_{k_1^*} = 1$. As shown in (19), the probability density functions (PDFs) of the packet arrival processes are required to know a priori to evaluate the $F_1(k)$. However, the PDFs are often unavailable a priori in HSR communication systems, hence, developing an online scheduling algorithm without requiring known PDFs is necessary. Fortunately, due to the concept of opportunistically minimizing an expectation [19], an online algorithm can be developed by removing the expectation operator. Given the observed packet arrivals A_k for all services and the scheduling policy can be obtained by $k_1^* = \arg \min_k \{\bar{F}_1(k)\}$, where $\bar{F}_1(k) = \bar{V}_{CS,k}(Q_{CS,k}^i - C_{BS} + A_k) - \bar{V}_{CS,k}(Q_{CS,k}^i + A_k)$.

4.2.2. Distributed Scheduling Policy in Second Stage. By substituting the results from the first stage into (17a) and (17b), the scheduling policy in the second stage can be obtained by solving the following problem:

$$\Pi^*(\mathbf{Q}_2^i) = \arg \min_{\Pi(\mathbf{Q}_2^i)} \left\{ \mathbb{E}_{\mathbf{H}_2} \left[\sum_{k=1}^K y_k F_2(k) \right] \right\}, \tag{20}$$

where

$$F_2(k) = \begin{cases} \bar{V}_{R,k}(Q_{R,k}^i - C_{R,k} + C_{BS}) \\ \quad - \bar{V}_{R,k}(Q_{R,k}^i + C_{BS}) - \beta_k T_k, & \text{if } k = k_1^*, \\ \bar{V}_{R,k}(Q_{R,k}^i - C_{R,k}) \\ \quad - \bar{V}_{R,k}(Q_{R,k}^i) - \beta_k T_k, & \text{if } k \neq k_1^*. \end{cases} \tag{21}$$

Similarly, given $\{\widetilde{V}_{R,k}(q), \forall k\}$ and $\mathbf{Q}_2^i = \{Q_{R,k}^i, \forall k\}$, we can solve (20) by the distributed computing $k_2^* = \arg \min_k \{F_2(k)\}$, which implies $y_{k_2^*} = 1$.

4.3. Distributed Online Scheduling Algorithm. Based on the distributed scheduling policy in two stages, we propose a distributed online scheduling algorithm using stochastic learning, which determines the scheduling actions and the per-node value functions as well as the LMs. As the train moves from the origin station to the terminal, the online algorithm allocates the network resource to multiple services slot by slot. The detailed steps of the proposed algorithm are given as follows.

Step 1 (initialization). RS initializes the value functions $\{\widetilde{V}_{R,k}^0(q)\}$ and LMs $\{\beta_k^0\}$. Meanwhile, CSs initialize $\{\widetilde{V}_{CS,k}^0(q)\}$.

Step 2 (scheduling decisions). When the train is moving, each CS and RS decide the scheduling actions in the two stages separately at the beginning of slot t . After obtaining $\mathbf{H}_1(t)$, the link capacity $C_{BS}(t)$ can be calculated. Then based on the local state information $A_k(t)$, $Q_{CS,k}(t)$, and $\{\widetilde{V}_{CS,k}^t(q)\}$, each CS calculates $\widetilde{F}_1(k)$ which is exchanged to determine the scheduling actions in the first stage. Similarly, using the scheduling results in the first stage, based on the local state information $Q_{R,k}(t)$, $\mathbf{H}_2(t)$, and $\{\widetilde{V}_{R,k}^t(q)\}$, RS determines the scheduling actions in the second stage.

Step 3 (per-node value functions and LMs update). Each CS updates $\{\widetilde{V}_{CS,k}^{t+1}(q)\}$ and the RS updates $\{\widetilde{V}_{R,k}^{t+1}(q)\}$ as well as the LMs $\{\beta_k^{t+1}\}$ according to the parameter update processes. Then let $t := t + 1$ and go to Step 2 until convergence.

The parameters are updated by the subgradient method, which is described as follows:

$$\begin{aligned} & \widetilde{V}_{CS,k}^{t+1}(q) \\ &= \widetilde{V}_{CS,k}^t(q) + \varepsilon_v^t (q + F_1'(x_k^*) - \widetilde{V}_{CS,k}^t(q)) \\ & \quad \times \mathbf{I}[\mathbf{Q}_1(t) = \delta_{k,q}], \end{aligned} \quad (22a)$$

$$\begin{aligned} & \widetilde{V}_{R,k}^{t+1}(q) \\ &= \widetilde{V}_{R,k}^t(q) + \varepsilon_v^t (q + F_2(y_k^*) - \widetilde{V}_{R,k}^t(q)) \\ & \quad \times \mathbf{I}[\mathbf{Q}_2(t) = \phi_{k,q}], \end{aligned} \quad (22b)$$

$$\beta_k^{t+1} = [\beta_k^t + \varepsilon_y^t (\alpha_k \lambda_k - y_k(t) T_k(t))]^+, \quad (22c)$$

where $\{\varepsilon_v^t\}$ and $\{\varepsilon_y^t\}$ are the step size sequences satisfying [17]: (1) $\sum_{t=0}^{\infty} \varepsilon_v^t = \infty$, $\varepsilon_v^t > 0$, $\lim_{t \rightarrow \infty} \varepsilon_v^t = 0$; (2) $\sum_{t=0}^{\infty} \varepsilon_y^t = \infty$, $\varepsilon_y^t > 0$, $\lim_{t \rightarrow \infty} \varepsilon_y^t = 0$; and (3) $\sum_{t=0}^{\infty} [(\varepsilon_v^t)^2 + (\varepsilon_y^t)^2] < \infty$. Furthermore, to enforce the convergence of the LMs and per-node value functions, ε_v^t and ε_y^t should also satisfy $\lim_{t \rightarrow \infty} \varepsilon_v^t / \varepsilon_y^t = 0$, for example, $\varepsilon_v^t = 1/(1 + t \log t)$ and $\varepsilon_y^t = 1/t$.

As shown in [20], the key idea of iteration convergence is characterized as follows. In (22a), (22b), and (22c), the updates of the per-stage value functions $\widetilde{V}_{CS,k}$ and $\widetilde{V}_{R,k}$ as well as the LMs β_k are performed using different step sizes. The update rate of the value functions is faster than that of the LMs. From the perspective of the LMs, the value functions will approximately converge to the optimal values corresponding to their current values, because they are updated at a faster time scale. Also, from the perspective of the value functions, the LMs appear to be almost constant. These two time-scale updates ensure that the value functions and the LMs converge to the optimal solution.

4.4. Implementation Issues. The distributed online scheduling algorithm runs in the three steps when the train moves from the origin station to the terminal. At each decision epoch, the scheduling actions in both stages are decided after observing each local CSI and QSI. It is worth emphasizing that the proposed algorithm is different from the iterative algorithms in static optimization problems because the data packets can be delivered during the iteration steps.

As an alternative, a table look-up method can be used for service scheduling in HSR networks. Specially, once the scheduling policy is obtained by the distributed online scheduling algorithm or other algorithms it can be stored in a table format. Each entry of the table represents the scheduling action for the given global system state including CSI and QSI. At each decision epoch, after observing the current state, the network controller looks up the table to find out the corresponding scheduling action and then executes the scheduling decision. The table look-up method is effective with low computational complexity.

Furthermore, the periodic updates in the proposed algorithm are necessary. After the proposed algorithm converges, the updates can be performed with long frequency slots or at random slots instead of at every slot. Since all the channel and queue states are realized infinitely many times during the trip, the periodic updates can ensure that the proposed algorithm also converges to the optimal solution. If the table look-up method is used, the table storing the scheduling policy can also be updated corresponding to each update in the proposed algorithm.

5. Simulation Results and Discussions

In this section, we implement the proposed scheduling algorithms using MATLAB and present simulation results to illustrate the performance of the algorithm.

5.1. Simulation Setup. For the purpose of comparison, we evaluate three related scheduling schemes as reference benchmarks. The first one is the traditional round-robin (RR) scheme which schedules services in a predetermined order. At time-slot t , the $(t \bmod (K + 1))$ th service is chosen for the two links. The second one is the greedy scheme, where service scheduling is done in a greedy method. Specifically, $x_k^* = \arg \max_k \{Q_{CS,k}(t)\}$ and $y_k^* = \arg \max_k \{\min(C_{R,k}(t), Q_{R,k}(t))\}$. The third one is the heuristic packet scheduling

algorithm proposed in [21] and used in the two links, where the services are scheduled for transmission depending on transmission rate, packet utility, and proportional fairness.

In order to better illustrate the performance of the scheduling algorithms, the suitable parameters should be set in simulations. We use a typical setting for HSR communication systems [10], with $T_B = 53 \mu\text{s}$ and $G = 240$ bits. The wireless communication in the two-hop link is established based on a carrier frequency of 2.4 GHz with bandwidth $W_1 = W_2 = 10$ MHz. The transmit power of the BS and AP is $P_{\text{BS}} = 47$ dBm and $P_R = 14$ dBm, respectively. The Rayleigh fading channel state $H_{R,k}(t)$ is sampled from the probability density function expressed as $f_H(h) = (h/\bar{\alpha}^2) \exp(-h/2\bar{\alpha}^2)$, $h \geq 0$, and $\bar{\alpha} = -2$ dB. For the Rician fading channel state, $H_{\text{BS},R}(t)$, Rician factor is 6 dB according to the Winner II project measurement results [22]. The maximum size L_Q for all buffers can be set as 50 packets such that there is no packet dropped from the buffers. A single simulation runs the algorithms for 400 slots and the results are averaged over 100 simulation runs.

5.2. Simulation Results. In our proposed scheduling algorithm, the objective is to minimize the expected cost function defined in (8), which includes the e2e delay. In the simulation results, the average e2e delay and service satisfaction are used as metrics to show the performance improvement. In addition, the convergence of the proposed distributed online scheduling algorithm is established through simulations.

Figure 2 compares the delay performance of the four scheduling schemes with different numbers of requested services. We set the delivery ratio requirement $\alpha_k = 0.8$ for all services with an equal average packet arrival rate of 5 packets/slot. It can be observed that the proposed distributed scheduling algorithm could achieve significant performance gains in average e2e delay over the other schemes. This illustrates the advantages of the proposed algorithm with distributed scheduling policy, which could effectively reduce the value of expected cost function in the two-hop HSR networks. Furthermore, we can see that, as the number of requested services increases, the average e2e delay per service for all scheduling schemes grows. This can be explained as follows. Since the capacity of the wireless channel is fixed in general, when the number of requested services increases, there are more data packets to be transmitted and less scheduling chance will be given to a certain service; then the average e2e delay per service becomes larger. Therefore, to reduce the average delay for service delivery and satisfy the QoS requirements, multi-input multioutput (MIMO) antennas can be deployed to improve the capacity of the wireless channel in HSR communication systems.

Figure 3 presents the delay performance with respect to the average packet arrival rate for the four scheduling schemes. In the simulation, there are 6 types of services supported and $\alpha_k = 0.8$ for each service. The average packet arrival rate for all services is set from 1 to 7 in order to prevent buffer overflow. From the figure, we can see that the average e2e delay per service in the proposed scheduling algorithm is smaller than the other schemes no matter how much the

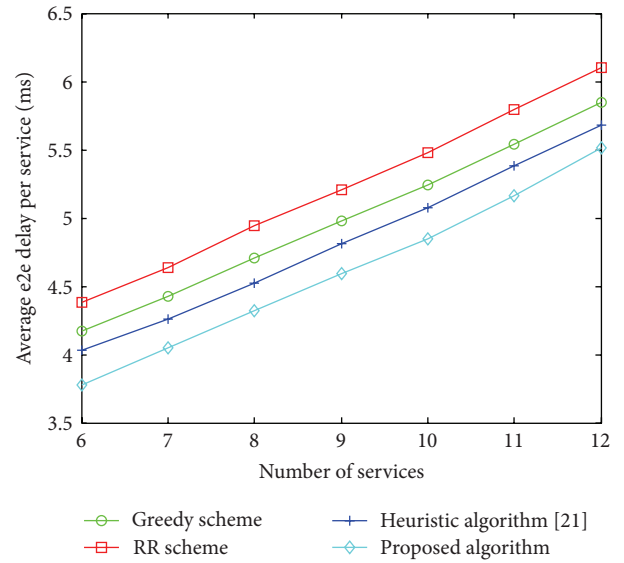


FIGURE 2: Average e2e delay per service versus the number of services.

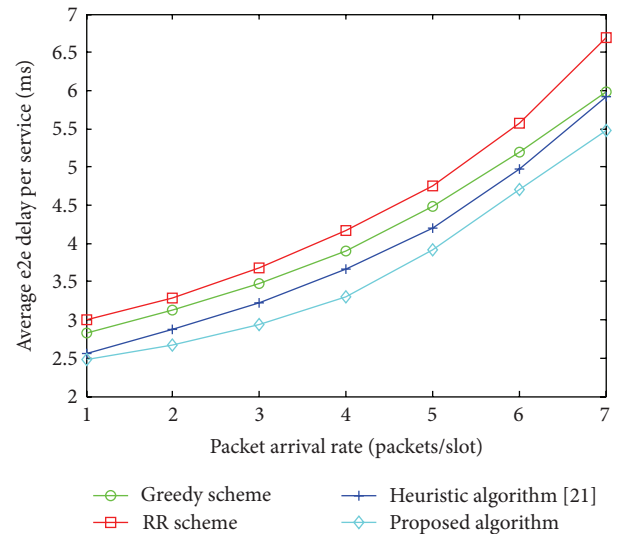


FIGURE 3: Average e2e delay per service versus packet arrival rate.

packet arrival rate is. In addition, the average delay value increases quickly with the increase of average packet arrival rate, which can be explained by Little's law. For the video services on the train, the average packet arrival rate is very high, so it is necessary to provide a large buffer or improve wireless transmission rate so as to prevent buffer overflow.

Figure 4 indicates the service delivery performance under the four scheduling schemes with $K = 6$ service types. The average packet arrival rates for all services are equal to 6 packets/slot. Based on the delivery ratio constraints (9), we define the satisfaction parameter ε_k of service k as the ratio of average throughput \bar{T}_k to $\alpha_k \lambda_k$; that is, $\varepsilon_k = \bar{T}_k / \alpha_k \lambda_k$. A larger ε_k represents the higher satisfaction of service k and $\varepsilon_k \geq 1$ implies that the delivery ratio constraint for service k is satisfied. Let the delivery ratio requirements from service 1

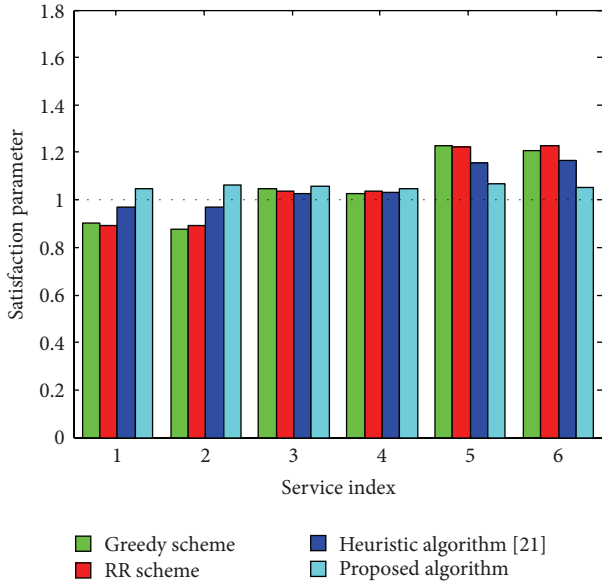


FIGURE 4: Service delivery performance.

to service K be 0.9, 0.9, 0.8, 0.8, 0.7, and 0.7. For service k , the rightmost bar shows the parameter ε_k for the proposed scheduling algorithm, and the remaining bars represent the parameter ε_k for the other three schemes. Simulation results show that the proposed distributed two-stage scheduling algorithm can achieve better service delivery performance over other schemes. This can be explained as follows. The RR scheme is fair in terms of scheduling opportunities but not channel/queue aware, the greedy scheme schedules the service with more packets in its buffer but does not consider the channel condition, and the heuristic scheduling algorithm is developed based on transmission rate, packet utility, and proportional fairness, which does not consider the heterogeneous delivery ratio requirements for different services. The same is true for both the two stages. Thus, the proposed algorithm, which is channel/queue aware and considers the service delivery ratio requirements, will be efficient on delivery ratio constraints in the long term.

Figure 5 illustrates the convergence property of the proposed distributed online scheduling algorithm. We plot the value functions of the CS for the first type of service versus scheduling slot index. It can be seen that the distributed algorithm converges quite fast and the values are extremely close to the final converged results after 200 iterations. The similar results can be obtained for per-node value functions at other CSs and RS. By comparing different curves, we can see that the per-node value function is increasing in queue backlog q . There are two reasons to explain this property. First, the value of per-slot cost function (7) grow linearly with the increase of queue backlog q . Second, the value function is positively correlated with the per-slot cost function based on (15). Furthermore, unlike the iterations in static optimization problems, the proposed scheduling algorithm is online, implying that data packets are delivered during the iteration steps.

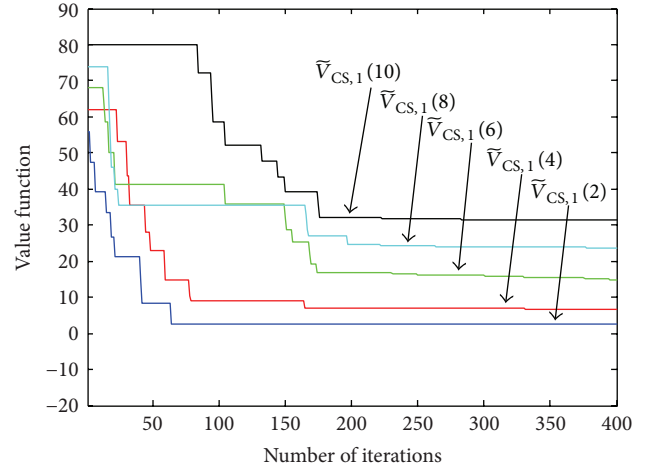


FIGURE 5: Illustration of the convergence of the proposed algorithm.

6. Conclusion

Providing passengers with multimedia services is one of the most important applications in HSR communication systems. This paper investigated delay-aware downlink service scheduling problem with stochastic packet arrivals and QoS requirements in relay-assisted HSR communication systems. We elaborate on the theory of MDP and illustrate how the approximate MDP and stochastic learning could help in obtaining low-complexity and distributed scheduling solutions. Simulation results show that the proposed algorithm outperforms other existing schemes in terms of average e2e delay and service delivery performances. Furthermore, the convergence of the proposed distributed online scheduling algorithm is shown by simulations.

For our future work, we will investigate the dynamic stochastic scheduling problem in the HSR networks using the stochastic network optimization approach. In addition, since the delay-aware is considered, motivated by [23], the design of a dynamic output feedback controller is necessary for multimedia services transmission in HSR communication systems.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the Fundamental Research Funds for the Central Universities (Grant no. 2014YJS026), the Key Projects of State Key Lab of Rail Traffic Control and Safety (nos. RCS2012ZZ004 and RCS2010ZT011), the China Postdoctoral Science Foundation (Grant no. 2013M530519), and the Natural Science Foundation of China (Grant no. U1334202).

References

- [1] D. T. Fokum and V. S. Frost, "A survey on methods for broadband Internet access on trains," *IEEE Communications Surveys and Tutorials*, vol. 12, no. 2, pp. 171–185, 2010.
- [2] J. Wang, H. Zhu, and N. J. Gomes, "Distributed antenna systems for mobile communications in high speed trains," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 4, pp. 675–683, 2012.
- [3] J. You, Z. Zhong, R. Xu, and G. Wang, "Transmission schemes for high-speed railway: direct or relay?" in *Proceedings of the IEEE International Wireless Communications and Mobile Computing Conference*, pp. 1103–1107, 2012.
- [4] L. Tian, J. Li, Y. Huang, J. Shi, and J. Zhou, "Seamless dual-link handover scheme in broadband wireless communication systems for high-speed rail," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 4, pp. 708–717, 2012.
- [5] Y. Yang, Z.-Q. Huang, Z.-D. Zhong, and X. Fu, "A study of real-time data transmission model of train-to-ground control in high-speed railways," in *Proceedings of the 72nd IEEE Vehicular Technology Conference Fall*, pp. 1–5, September 2010.
- [6] O. B. Karimi, J. Liu, and C. Wang, "Seamless wireless connectivity for multimedia services in high speed trains," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 4, pp. 729–739, 2012.
- [7] H. Liang and W. Zhuang, "Efficient on-demand data service delivery to high-speed trains in cellular/infostation integrated networks," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 4, pp. 780–791, 2012.
- [8] S. Spagna, M. Liebsch, and R. Baldessari, "Design principles of an operator-owned highly distributed content delivery network," *IEEE Communications Magazine*, vol. 51, no. 4, pp. 132–140, 2013.
- [9] H. Wu, T. Wang, and Z. Zhong, "Location updating schemes for high-speed railway cellular communication systems," *Mathematical Problems in Engineering*, vol. 2012, Article ID 802152, 15 pages, 2012.
- [10] D. H. Ho and S. Valaee, "Information raining and optimal link-layer design for mobile hotspots," *IEEE Transactions on Mobile Computing*, vol. 4, no. 3, pp. 271–284, 2005.
- [11] S. Lin, Z. Zhong, L. Cai, and Y. Luo, "Finite state Markov modelling for high speed railway wireless communication channel," in *Proceedings of the IEEE Global Communications Conference*, pp. 5421–5426, 2012.
- [12] M. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, John Wiley & Sons, Hoboken, NJ, USA, 1994.
- [13] A. Leon-Garcia, *Probability, Statistics, and Random Processes for Electrical Engineering*, Prentice Hall, 2008.
- [14] E. Altman, *Constrained Markov Decision Processes: Stochastic Modeling*, Chapman and Hall/CRC Press, London, UK, 1999.
- [15] V. S. Borkar, "An actor-critic algorithm for constrained Markov decision processes," *Systems and Control Letters*, vol. 54, no. 3, pp. 207–213, 2005.
- [16] R. Montes-de-Oca, E. Lemus-Rodríguez, and F. S. Salem-Silva, "Nonuniqueness versus uniqueness of optimal policies in convex discounted Markov decision processes," *Journal of Applied Mathematics*, vol. 2013, Article ID 271279, 5 pages, 2013.
- [17] R. Wang, V. K. N. Lau, and Y. Cui, "Queue-aware distributive resource control for delay-sensitive two-hop MIMO cooperative systems," *IEEE Transactions on Signal Processing*, vol. 59, no. 1, pp. 341–350, 2011.
- [18] M. Moghadari, E. Hossain, and L. B. Le, "Delay-optimal distributed scheduling in multi-user multi-relay cellular wireless networks," *IEEE Transactions on Communications*, vol. 61, no. 4, pp. 1349–1360, 2013.
- [19] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*, Morgan & Claypool, 2010.
- [20] F. Fu and M. van der Schaar, "Structure-aware stochastic control for transmission scheduling," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 9, pp. 3931–3945, 2012.
- [21] R. Zhou, H. N. Nguyen, and I. Sasase, "Packet scheduling for cellular relay networks by considering relay selection, channel quality, and packet utility," *Journal of Communications and Networks*, vol. 11, no. 5, pp. 464–472, 2009.
- [22] IST-WINNER D1.1.2 P. Kyösti et al., "WINNER II channel models," ver 1.1, 2007, <http://www.ist-winner.org/WINNER2-Deliverables/D1.1.2v1.2.pdf>.
- [23] G. Song and Z. Wang, "A delay partitioning approach to output feedback control for uncertain discrete time-delay systems with actuator saturation," *Nonlinear Dynamics*, vol. 74, no. 1-2, pp. 189–202, 2013.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

