Delay-Optimal Dynamic Mode Selection and Resource Allocation in Device-to-Device Communications - Part II: Practical Algorithm

Lei Lei *Member, IEEE*, Yiru Kuang, Nan Cheng *Student Member, IEEE*, Xuemin (Sherman) Shen *Fellow, IEEE*, Zhangdui Zhong and Chuang Lin *Senior Member, IEEE*

***Abstract*—In the Part I of the paper ("Delay-Optimal Dynamic Mode Selection and Resource Allocation in Device-to-Device Communications - Part I: Optimal Policy"), we investigated dynamic mode selection and subchannel allocation for an Orthogonal Frequency Division Multiple Access (OFDMA) cellular network with device-to-device (D2D) communications to minimize the average end-to-end delay performance under dropping probability constraint. We formulated the optimal resource control problem into an infinite horizon average reward constraint Markov decision process (CMDP), and the optimal control policy derived in Part I using the brute-force offline value iteration algorithm based on the reduced state equivalent Bellman's equation still faces the well-known curse of dimensionality problem, which limits its practical application in realistic scenarios with multiple D2D users and cellular users. In the part II of the paper, we use linear value approximation techniques to further reduce the state space. Moreover, online stochastic learning algorithm with two time scales is applied to update the value functions and Lagrangian Multipliers (LMs) based on the real-time observations of channel state information (CSI) and queue state information (QSI). The combined online stochastic learning solution converges almost surely to a global optimal solution under some realistic conditions. Simulation results show that the proposed approach achieves nearly the same performance as the offline value iteration algorithm, and outperforms the conventional CSI-only scheme and throughput-optimal scheme in stability sense.**

***Index Terms*—Device-to-Device Communication; Mode Selection; Resource Allocation; Online Stochastic Learning**

I. Introduction

In the Part I of the paper [1], we introduced the problem of optimal dynamic mode selection and resource allocation to minimize the average end-to-end delay under the constraint of packet dropping probability for network assisted device-to-device (D2D) communications [2]–[4] with bursty traffic. We considered an Orthogonal Frequency Division Multiple Access (OFDMA) system with one base station (BS), multiple D2D user equipment (UE) pairs, and cellular UEs with uplink or downlink transmission. Compared with the resource control problem in traditional cellular networks, there are a number of unique issues to address to obtain resource optimization in D2D communications, such as (1) route selection between the one-hop route of D2D link (direct over-the-air link) in D2D Mode and the two-hop route of cellular links in Cellular Mode; (2) resource allocation for D2D links and cellular links with resource reuse; (3) joint uplink and downlink resource optimization for the end-to-end performance of the two-hop route when a pair of D2D UEs works in the Cellular Mode. In order to characterize the above issues, we first developed a queuing model whose underlying system state dynamics evolves as a controlled Markov chain, where the system state includes the joint queue state of the queues at the UEs for uplink transmission and the queues at the BS for downlink transmission as well as the joint channel state of all the D2D links, cellular uplinks and cellular downlinks. Specifically, we introduced two important concepts to characterize the unique features of D2D communications. The first concept is *radio resource group* (RRG), which defines a group of links that may reuse radio resources. Therefore, the channel state of a link is a tuple including its Adaptive Modulation and Coding (AMC) states in all the RRGs that this link belongs to. The second concept is *link constraint set* of a queue to characterize the set of servers for the queue in different routes. Based on the queuing model, the delay-optimal resource control over frequency-selective fading channel with AMC scheme in the physical layer is formulated as an infinite horizon average reward constrained Markov Decision Process (CMDP) [6], [7]. In order to formulate the CMDP model, the transition kernel of the controlled Markov chain was derived, which takes into account the coupling relationship between the uplink and downlink resource allocation. Moreover, closed-form expressions for end-to-end performance metrics such as average delay and dropping probability were given as functions of steady-state probabilities of the controlled Markov chain, based on which the cost function of CMDP model was given. We utilized the Lagrangian approach to turn the CMDP problem into an unconstraint Markov Decision Process (MDP) problem, and established the strong duality result over the space of randomized policy. Moreover, we further proved the existence of an optimal policy, which is either a deterministic

Manuscript received Jan. 12, 2015; revised March 22, 2015; accepted June 8, 2015. This work was supported by the National Natural Science Foundation of China (No. 61272168, No. U1334202, No. 61472199), the State Key Laboratory of Rail Traffic Control and Safety (No. RCS2014ZT10), Beijing Jiaotong University, and the Key Grant Project of Chinese Ministry of Education (No. 313006).

L. Lei, Y. Kuang and Z. Zhong are with the State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, China.

N. Cheng and X. Shen are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada.

C. Lin is with the Department of Computer Science and Technology, Tsinghua University, Beijing, China.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2015.2444791, IEEE Transactions on Vehicular Technology

policy or a mix of two deterministic policies, equivalent to choosing independently one of two deterministic policies at each epoch by the toss of a (biased) coin. To solve the unconstraint MDP problem, we derived an equivalent Bellman's equation with reduced state space. We showed by simulations that the optimal policy derived by the brute-force offline value iteration algorithm based on the equivalent Bellman's equation achieves significant gain compared to various baselines such as the conventional CSI-only control and the throughput optimal control (MaxWeight algorithm).

It is worth noting that the complexity of the brute-force offline value iteration algorithm based on the reduced state equivalent Bellman's equation still grows exponentially with the number of users in the network, limiting its application in practical scenarios. In fact, it is well-known that there is no simple solution for the infinite horizon average reward MDP problem that delay-aware resource control belongs to, because the brute-force value iterations or policy iterations could not lead to any viable solution due to the curse of dimensionality [8]–[11]. Moreover, our problem for network assisted D2D communications is further complicated due to the unique issues listed above. For example, the channel state transition probabilities, which are used to derive the conditional expectations of cost function and queue state transition probabilities in the equivalent Bellman's equation, are very difficult to obtain when more than two links are allowed to reuse the same time-frequency resource.

In the Part II of the paper, we address the curse of dimensionality problem in solving the CMDP formulated in Part I, so that a practical algorithm with acceptable computational complexity and signaling overhead can be derived. To reduce the complexity, we obtain a delay-optimal solution using approximate dynamic programming and online stochastic learning. Specifically, we approximate the value function in the equivalent Bellman's equation by a sum of per-queue value functions. The per-queue value functions are estimated and learned using an online stochastic learning algorithm based on the real-time observations of the CSI and QSI, eliminating the need of deriving the channel state transition probabilities. Moreover, the Lagrangian Multipliers (LMs) for the constraint optimization problem are updated simultaneously with the value functions over different time scales. The optimal dynamic mode selection and resource allocation actions can be determined by an algorithm that has a similar structure with the MaxWeight algorithm in Lyapunov stability approach, with the weight determined by the per-queue value functions instead of the queue lengths. We prove the almost-sure convergence of the proposed algorithm. We also show by simulations that our proposed scheme achieves significant gain compared to various baselines such as the conventional CSI-only control and the throughput optimal control (MaxWeight algorithm). Together with Part I, this pair of works provide a general framework for the dynamic constrained optimization of mode selection and resource allocation in D2D communications under bursty traffic model, where the general form of the optimal policy and a practical algorithm with simple structure and near-optimal performance are given.

The organization of the paper is as follows. We recall the general network model for network assisted D2D communications as well as the MDP problem formulation for dynamic mode selection and resource allocation in Section II. In Section III, we derive a low complexity learning algorithm, which updates the per-queue value functions based on real-time observations of CSI and QSI, as well as a resource allocation algorithm with similar structure as the MaxWeight algorithm. In Section IV, we discuss the performance simulations. Finally, we summarize the main results in Section V.

II. Network Model and Previous Results

A. Network Model

Consider a Frequency Division Duplex (FDD) OFDMA cellular network with D2D communications capability, where there are D D2D UE pairs, C_{u} cellular UEs (CUEs) with uplink communications and C_{d} CUEs with downlink communications in a single cell. A D2D UE pair consists of a source D2D UE (src. DUE) and a destination D2D UE (dest. DUE) within direct over-the-air communications range with each other, which is formed through the various neighbor/peer/service discovery mechanisms proposed in literature. The whole uplink or downlink spectrum is divided into N_{F} equal size subchannels. A subchannel in the uplink (resp. downlink) spectrum shall be referred to as uplink (resp. downlink) subchannel in the rest of the paper. Moreover, we assume that D2D links share uplink resources with cellular uplinks. Time is slotted and each time slot has an equal length.

The above OFDMA cellular network with D2D communications can be formulated as a general network model with a set $\mathcal{N}$ of nodes and a set $\mathcal{L}$ of transmission links. Define $\mathcal{N} := \{0, 1, \ldots, N\}$, where node 0 represents the base station (BS) and nodes $1, \ldots, 2D$ represent the DUEs, nodes $2D+1, \ldots, 2D+C_{\mathrm{u}}$ represent the uplink CUEs, and nodes $2D+C_{\mathrm{u}}+1, \ldots, N = 2D+C_{\mathrm{u}}+C_{\mathrm{d}}$ represent the downlink CUEs. We use i or j to denote the index of a node within $\mathcal{N}$ (i.e., $i, j \in \mathcal{N}$) in the rest of the paper. Each transmission link represents a communication channel for direct transmission from a given node i to another node j, and is labeled by (i, j) (where $i, j \in \mathcal{N}$). All data that enter the network are associated with a particular connection which defines the source and destination of the data. Let $\mathcal{C}_{\mathrm{D}} = \{1, \ldots, D\}$, $\mathcal{C}_{\mathrm{Cu}} = \{D+1, \ldots, D+C_{\mathrm{u}}\}$, and $\mathcal{C}_{\mathrm{Cd}} = \{D+C_{\mathrm{u}}+1, \ldots, D+C_{\mathrm{u}}+C_{\mathrm{d}}\}$ represent the set of D2D connections, cellular uplink connections and cellular downlink connections, respectively. Define $\mathcal{C} := \{1, \ldots, C\} = \mathcal{C}_{\mathrm{D}} \bigcup \mathcal{C}_{\mathrm{Cu}} \bigcup \mathcal{C}_{\mathrm{Cd}}$ (with $C = D+C_{\mathrm{u}}+C_{\mathrm{d}}$) as the set of all connections in the network. We use c to denote the index of a connection within $\mathcal{C}$ (i.e., $c \in \mathcal{C}$) in the rest of the paper.

The data from connection c is transmitted hop by hop along the route(s) of the connection to its destination node. Each node i along the route(s) of connection c maintains a queue $q_i^{(c)}$ for storing its data except for the destination node, since the data is considered to exit the network once it reaches the destination. Define Θ as the set of queues in the system. We assume each queue has a finite capacity of $N_{\mathrm{Q}} < \infty$ (in number of bits or packets). The set of queues can be divided into two non-overlapping disjoint sets, i.e., uplink queues Θ_{u}

and downlink queues Θ_{d}, according to whether a queue is maintained by an UE or the BS. Define Θ_{Cu} and Θ_{Cd} as the set of queues for cellular uplink connections and cellular downlink connections, respectively, while define $\Theta_{\mathrm{D-u}}$ and $\Theta_{\mathrm{D-d}}$ as the set of uplink queues and downlink queues for D2D connections, respectively. Define the per queue link constraint set of a queue $q_i^{(c)}$ as $\mathcal{L}_i^{(c)}$, where the data from a queue $q_i^{(c)}$ can only be transmitted via links in $\mathcal{L}_i^{(c)}$. Note that there is only one link in $\mathcal{L}_i^{(c)}$ for all the queues except the uplink queues for D2D connections, i.e, $q_i^{(c)} \in \Theta_{\mathrm{D-u}}$, which can be served by either D2D link or cellular uplink.

B. Physical Layer Model

We define a Resource Reuse Group (RRG) $\mathcal{B}_u$ as the subset of links $(i,j) \in \mathcal{L}$ that can be scheduled for transmission simultaneously on any subchannel in a time slot. Let $\mathcal{U}$ represent the set of RRG indexes, $\mathcal{U}_{\mathrm{u}}$ and $\mathcal{U}_{\mathrm{d}}$ represent the subsets of RRG indexes for uplink and downlink subchannels, respectively. We use u to denote the index of a RRG within $\mathcal{U}$ (i.e., $u \in \mathcal{U}$) in the rest of the paper. For any link $(i,j) \in \mathcal{L}$, define $\mathcal{U}_{ij} := \{u|(i,j) \in \mathcal{B}_u, u \in \mathcal{U}\}$ as the index set of RRGs that contain link (i,j).

Assume that the instantaneous channel gain comprising the path loss, shadowing and fast fading effects of the wireless channel from the transmitter of node $i \in \mathcal{N}$ to the receiver of node $j \in \mathcal{N}$ on any subchannel m remains constant within a time slot and i.i.d. between time slots, the value of which at time slot t is denoted by $G_{ij,t}^{(m)}$. Let $p_{ij,t}^{(m)}$ be the transmission power of link $(i,j) \in \mathcal{L}$ on subchannel m at time slot t. Assume that every scheduled link on a downlink subchannel (resp. uplink subchannel) always transmits at constant power $P_{\max}^{\mathrm{BS}}/N_{\mathrm{F}}$, (resp. $P_{\max}^{\mathrm{UE}}/N_{\mathrm{F}}$). The SINR value of a link (i,j) on a subchannel m when RRG $\mathcal{B}_u$ is scheduled on it at time slot t can be derived as $\forall\ (i,j) \in \mathcal{B}_u$

$$SINR_{ij,t}^{(m,u)} = \frac{p_{ij,t}^{(m)} G_{ij,t}^{(m)}}{N_{ij,t}^{(m)} + \sum_{(i',j') \in \mathcal{B}_u \backslash \{(i,j)\}} p_{i'j',t}^{(m)} G_{i'j,t}^{(m)}}, \quad (1)$$

where $N_{ij,t}^{(m)}$ denotes the noise power on subchannel m at time slot t.

We assume that AMC is used, where the SINR values are divided into K non-overlapping consecutive regions [12]. For any $k \in \{1,\ldots,K\}$, if the SINR value $SINR_{ij,t}^{(m,u)}$ of link (i,j) falls within the k-th region $[\Gamma_{k-1},\Gamma_k)$, the instantaneous data rate of link (i,j) on subchannel m when RRG $\mathcal{B}_u$, $\forall u \in \mathcal{U}_{ij}$ is scheduled is a fixed value R_k according to the selected modulation and coding scheme in this state. Obviously, $\Gamma_0 = 0$ and $\Gamma_K = \infty$. Also, we have $R_1 = 0$, i.e., no packet is transmitted in channel state 1 to avoid the high transmission error probability. Define the CSI of link (i,j) to be $\mathbf{H}_{ij,t} := \{H_{ij,t}^{(m,u)}|(m \in \{1,\ldots,N_{\mathrm{F}}\}, u \in \mathcal{U}_{ij}\}$, where $H_{ij,t}^{(m,u)}$ denotes the channel state of link (i,j) on subchannel m when RRG $\mathcal{B}_u$ is scheduled. Specifically, $H_{ij,t}^{(m,u)} = k$ if $SINR_{ij,t}^{(m,u)}$ is between $[\Gamma_{k-1},\Gamma_k)$.

C. Bursty Source Model, Queuing Dynamics, and Queuing Model

Let $A_{c,t}$ denote the amount of new connection c data[1] that exogenously arrives to its source node during time slot t. We assume that the data arrival process is i.i.d. over time slots following general distribution $f_A(n)$ with average arrival rate $\mathbf{E}[A_{c,t}] = \lambda_c$. Let $A_{i,t}^{(c)}$ denote the amount of data arrived to node i for connection c during time slot t. When $q_i^{(c)} \in \Theta_{\mathrm{u}} \bigcup \Theta_{\mathrm{Cd}}$, node i is the source node of connection c, and $A_{i,t}^{(c)} = A_{c,t}$. Otherwise, when $q_0^{(c)} \in \Theta_{\mathrm{D-d}}$, it is the second-hop queue of connection c, and $A_{0,t}^{(c)}$ depends on the data departure process of the corresponding uplink transmission on cellular uplink $((2c-1),0)$.

Let $Q_{i,t}^{(c)}$ denote the length of $q_i^{(c)}$ at the beginning of time slot t. Let $r_{i,t}^{(c)}$ be the instantaneous data rate of queue $q_i^{(c)}$ during time slot t[2], which is equal to the sum of the instantaneous data rate $r_{ij,t}$ of the scheduled link $(i,j) \in \mathcal{L}_i^{(c)}$ at time slot t. If $Q_{i,t}^{(c)}$ is less than $r_{i,t}^{(c)}$ during time slot t, padding bits shall be transmitted along with the data. However, the amount of useful data transmitted from $q_i^{(c)}$ during time slot or the throughput of $q_i^{(c)}$ is defined as

$$T_{i,t}^{(c)} = \min[Q_i^{(c)}, r_i^{(c)}]. \quad (2)$$

Moreover, the amount of useful data transmitted via link (i,j) during time slot t or the throughput of link (i,j) is defined for any link within the link constraint set of queue $q_i^{(c)} \in \Theta_{\mathrm{d}} \bigcup \Theta_{\mathrm{Cu}}$ as

$$T_{ij,t} = \min[Q_i^{(c)}, r_{ij,t}],\ \forall (i,j) \in \mathcal{L}_i^{(c)}. \quad (3)$$

For any queue $q_i^{(c)} \in \Theta_{\mathrm{D-u}}$, we assume that the data in the queue is first assigned to link $(2c-1,0)$ and then the remaining data left in the queue (if any) shall be assigned to link $(2c-1,2c)$. According to the above data assignment rule, we have that $T_{(2c-1)0,t}$ obeys (3), while $\forall q_{(2c-1)}^{(c)} \in \Theta_{\mathrm{D-u}}$

$$T_{(2c-1)(2c),t} = \min[Q_i^{(c)} - T_{(2c-1)0,t}, r_{(2c-1)(2c),t}]. \quad (4)$$

Arriving data are placed in the queue throughout the time slot t and can only be transmitted during the next time slot $t+1$. If the queue length reached the buffer capacity N_{Q}, the subsequent arriving data will be dropped. According to the above assumption, the queuing process evolves as follows:

$$Q_{i,t+1}^{(c)} = \min\left[N_{\mathrm{Q}}, \max[0, Q_{i,t}^{(c)} - r_{i,t}^{(c)}] + A_{i,t}^{(c)}\right]. \quad (5)$$

The queuing model is illustrated in Fig.2 of Part I of this work [1].

D. System state, Control Policy and State Transition Probabilities

The *global system state* of the above queuing model at time slot t can be characterized by the aggregation of the system

[1] The data can take units of bits or packets. The latter is appropriate when all the packets have fixed length.

[2] The instantaneous data rate can take units of bits/slot or packets/slot. The latter is appropriate when all the packets have fixed length and the achievable data rates are constrained to integral multiples of the packet size.

CSI and system QSI, i.e., $\mathbf{S}_t = (\mathbf{H}_t, \mathbf{Q}_t)$. The system QSI is defined as $\mathbf{Q}_t := \left\{ Q_{i,t}^{(c)} | q_i^{(c)} \in \Theta \right\}$, which is a vector consisting of the lengths of all the queues at the beginning of time slot t. The system CSI is defined as $\mathbf{H}_t := \{\mathbf{H}_{ij,t} | (i,j) \in \mathcal{L}\}$, where $\mathbf{H}_{ij,t}$ denotes the channel state of link (i,j) in time slot t as defined Section II-B.

In each time slot, an uplink (resp. downlink) subchannel can be allocated to at most one uplink (resp. downlink) RRG for uplink (resp. downlink) transmission. Let $m \in \{1, \ldots, N_{\rm F}\}$ denote the index of a subchannel, which can be either the m-th uplink subchannel or the m-th downlink subchannel. Let $x_{u,t}^{(m)} \in \{0, 1\}$ denote the subchannel allocation for RRG $\mathcal{B}_u$, $u \in \mathcal{U}$ at time slot t, where $x_{u,t}^{(m)} = 1$ if subchannel m is allocated to RRG $\mathcal{B}_u$, and $x_{u,t}^{(m)} = 0$ otherwise. We have the constraint that $\sum_{u\in\mathcal{U}_{\rm u}} x_{u,t}^{(m)} \leq 1$ and $\sum_{u\in\mathcal{U}_{\rm d}} x_{u,t}^{(m)} \leq 1$ for any $m \in \{1, \ldots, N_{\rm F}\}$. We assume that a RRG is scheduled for transmission only when all its links have non-empty queues.

A queue $q_i^{(c)}$ is scheduled in time slot t when at least one RRG $\mathcal{B}_u$ containing a link (i,j) in its link constraint set $\mathcal{L}_i^{(c)}$ is scheduled on any subchannel. Note that except for the uplink queues of D2D connections, the per-queue link constraint set of every queue contains only one link. When mode selection of a D2D connection c is performed dynamically at each time slot, the problem becomes deciding whether to schedule the D2D link $(2c-1, 2c)$ or the cellular uplink $(2c-1, 0)$ on a subchannel to serve the queue $q_{(2c-1)}^{(c)}$, which is essentially a subchannel allocation decision. Therefore, the delay-optimal dynamic mode selection and subchannel allocation problem can be solved by only considering the design of delay-optimal subchannel allocation algorithm.

In each time slot, the resource controller observes the system state $\mathbf{S}_t$ and chooses a subchannel allocation action from the set of allowable actions in the action space $\mathcal{A}_{\rm x}$. A subchannel allocation action $\mathbf{x}$ is defined as $\mathbf{x} := \left\{ x_u^{(m)} \in \{0,1\} | u \in \mathcal{U}_{\rm u} \bigcup \mathcal{U}_{\rm d}, m \in \{1, \ldots, N_{\rm F}\} \right\} \in \mathcal{A}_{\rm x}$.

A control policy prescribes a procedure for action selection in each state at all decision epoches t. We consider stationary Markovian deterministic control policies[3]. A deterministic control policy given by Ω is a mapping $\mathcal{S} \to \mathcal{A}_{\rm x}$ from the state space to the action space, which is given by $\Omega(\mathbf{S}) = \mathbf{x} \in \mathcal{A}_{\rm x}, \ \forall \mathbf{S} \in \mathcal{S}$.

Note that the instantaneous data rate $r_{i,t}^{(c)}$ is impacted by the subchannel allocation action at time slot t, i.e.,

$$r_{i,t}^{(c)} = \sum_{(i,j)\in\mathcal{L}_i^{(c)}} \sum_{m=1}^{N_{\rm F}} \sum_{u\in\mathcal{U}_{ij}} x_{u,t}^{(m)} R_{H_{ij,t}^{(m,u)}}. \qquad (6)$$

The system behavior of the above queuing model can be represented by the controlled discrete-time Markov chain (DTMC) $\{\mathbf{S}_t\}_{t=0,1,\ldots} := \{(\mathbf{H}_t, \mathbf{Q}_t)\}_{t=0,1,\ldots}$. Given a system state $\mathbf{S}_t$ and an action $\mathbf{x}$ at time slot t, the state transition probability of the DTMC is given by

$$\begin{aligned} \Pr.\{\mathbf{S}_{t+1}|\mathbf{S}_t, \mathbf{x}\} &= \Pr.\{\mathbf{H}_{t+1}|\mathbf{H}_t\}\Pr.\{\mathbf{Q}_{t+1}|\mathbf{S}_t, \mathbf{x}\} \\ &= \Pr.\{\mathbf{H}_{t+1}\}\Pr.\{\mathbf{Q}_{t+1}|\mathbf{S}_t, \mathbf{x}\}. \end{aligned} \qquad (7)$$

According to (5), the conditional probability of $Q_{i,t+1}^{(c)}$ given the system state $\mathbf{S}_t$ and an action $\mathbf{x}$ can be derived as

$$\begin{aligned} &\Pr.\{Q_{i,t+1}^{(c)}|\mathbf{S}_t, \mathbf{x}\} = \Pr.(A_{i,t}^{(c)} = n), \\ &\quad \text{if } Q_{i,t+1}^{(c)} = \min\left[N_{\rm Q}, \max[0, Q_{i,t}^{(c)} - r_{i,t}^{(c)}] + n\right], \end{aligned} \qquad (8)$$

where

$$\begin{aligned} &\Pr.(A_{i,t}^{(c)} = n) \\ &= \begin{cases} f_A(n), & \text{if } q_i^{(c)} \in \Theta_{\rm u} \bigcup \Theta_{\rm Cd}, \\ 1, & \text{if } q_i^{(c)} \in \Theta_{\rm D-d} \text{ and } n = T_{(2c-1)0,t}, \\ 0, & \text{if } q_i^{(c)} \in \Theta_{\rm D-d} \text{ and } n \neq T_{(2c-1)0,t}. \end{cases} \end{aligned} \qquad (9)$$

The queue state transition probability $\Pr.\{\mathbf{Q}_{t+1}|\mathbf{S}_t, \mathbf{x}\}$ can be derived as the product of $\Pr.\{Q_{i,t+1}^{(c)}|\mathbf{S}_t, \mathbf{x}\}$ over all queues $q_i^{(c)} \in \Theta$ as

$$\Pr.\{\mathbf{Q}_{t+1}|\mathbf{S}_t, \mathbf{x}\} = \prod_{q_i^{(c)}\in\Theta} \Pr.\{Q_{i,t+1}^{(c)}|\mathbf{S}_t, \mathbf{x}\}. \qquad (10)$$

Remark 1 (channel state transition probability)**.** *Note that we do not recall the derivation of channel state transition probability* $\Pr.\{\mathbf{H}_{t+1}|\mathbf{H}_t\} = \Pr.\{\mathbf{H}_{t+1}\}$ *in* (7) *given in Part I. This is because in order to derive the delay-optimal subchannel allocation action in Section III, we shall utilize the i.i.d. assumption of the CSI process and the stochastic approximation method to simplify the optimization problem, so that* $\Pr.\{\mathbf{H}_{t+1}\}$ *does not need to be derived. However, we would like to point out that if the i.i.d assumption of CSI process does not hold or if the objective is to determine the steady-state probabilities of the queuing model for performance evaluation,* $\Pr.\{\mathbf{H}_{t+1}|\mathbf{H}_t\}$ *needs to be derived.*

E. Optimization Problem Formulation

Our objective is to optimize the subchannel allocation policy so as to minimize the average weighted sum delay of all the connections subject to dropping probability constraints.

Problem 1. *The delay-optimal subchannel allocation design can be formulated as the constrained optimization problem*

$$\min_{\Omega} \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbf{E}^{\Omega}[g_0(\mathbf{S}_t, \Omega(\mathbf{S}_t))] \qquad (11)$$

$$\text{s.t.} \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbf{E}^{\Omega}[g_c(\mathbf{S}_t, \Omega(\mathbf{S}_t))] \leq d_{\max}, \ \forall c \in \mathcal{C},$$

where

$$g_0(\mathbf{S}, \mathbf{x}) = \frac{Q_i^{(c)}}{\lambda_c(1 - d_{\max})}, \qquad (12)$$

$$g_c(\mathbf{S}, \mathbf{x}) = \begin{cases} 1 - \frac{T_{(c+D)0}}{\lambda_c}, & \textit{if } c \in \mathcal{C}_{\rm Cu}, \\ 1 - \frac{T_{0(c+D)}}{\lambda_c}, & \textit{if } c \in \mathcal{C}_{\rm Cd}, \\ 1 - \frac{T_{(2c-1)(2c)} + T_{0(2c)}}{\lambda_c}, & \textit{if } c \in \mathcal{C}_{\rm D}. \end{cases} \qquad (13)$$

[3] In Part I, we have proven that the optimal policy is either a deterministic policy or a mix of two deterministic policies. In Part II, we focus only on deterministic policy to facilitate implementation.

For any given nonnegative LMs $\boldsymbol{\eta} = \{\eta_c | c \in \mathcal{C}\}$, we define the Lagrangian function of Problem 1 as

$$L(\Omega, \eta) = \mathbf{E}^{\pi(\Omega)}[g(\mathbf{S}, \Omega(\mathbf{S}))] + \sum_{c \in \mathcal{C}} \lambda_c \eta_c (1 - d_{\max})^2, \quad (14)$$

where

$$\begin{aligned} g(\mathbf{S}, \Omega(\mathbf{S})) = &\sum_{c \in \mathcal{C}_{\mathrm{Cu}}} \big(\omega_c Q^{(c)}_{(c+D)} - \eta_c (1 - d_{\max}) T_{(c+D)0} \big) \\ &+ \sum_{c \in \mathcal{C}_{\mathrm{Cd}}} \big(\omega_c Q^{(c)}_{0} - \eta_c (1 - d_{\max}) T_{0(c+D)} \big) \\ &+ \sum_{c \in \mathcal{C}_{\mathrm{D}}} \big(\omega_c (Q^{(c)}_{(2c-1)} + Q^{(c)}_{0}) \\ &- \eta_c (1 - d_{\max}) (T_{(2c-1)(2c)} + T_{0(2c)}) \big). \quad (15) \end{aligned}$$

Therefore, Problem 1 can be divided into the following two subproblems:

$$\textbf{Subproblem 1-1: } G(\boldsymbol{\eta}) = \min_{\Omega} L(\Omega, \boldsymbol{\eta}),$$

$$\textbf{Subproblem 1-2: } G(\boldsymbol{\eta}^*) = \max_{\boldsymbol{\eta}} G(\boldsymbol{\eta}).$$

where $G(\boldsymbol{\eta})$ is the corresponding Lagrange dual function.

Subproblem 1-1 with given LMs $\boldsymbol{\eta}^*$ can be solved by the equivalent Bellman's equation.

$$\begin{aligned} \theta + V(\mathbf{Q}^{(\grave{z})}) = &\min_{\Omega(\mathbf{Q}^{(\grave{z})})} \Big\{ g(\mathbf{Q}^{(\grave{z})}, \Omega(\mathbf{Q}^{(\grave{z})})) \\ &+ \sum_{\mathbf{Q}^{(\grave{y})} \in \mathcal{Q}} \Pr.[\mathbf{Q}^{(\grave{y})} | \mathbf{Q}^{(\grave{z})}, \Omega(\mathbf{Q}^{(\grave{z})})] V(\mathbf{Q}^{(\grave{y})}) \Big\}, \forall \mathbf{Q}^{(\grave{z})} \in \mathcal{Q}, \end{aligned} \quad (16)$$

where $V(\mathbf{Q}^{(\grave{y})}) = \mathbf{E}_{\mathbf{H}} \Big[V(\mathbf{H}, \mathbf{Q}^{(\grave{y})}) | \mathbf{Q}^{(\grave{y})} \Big] = \sum_{\mathbf{H} \in \mathcal{H}} \Pr.[\mathbf{H}] V(\mathbf{H}, \mathbf{Q}^{(\grave{y})})$ is the conditional expectation of value function $V(\mathbf{S})$ taken over the channel state space $\mathcal{H}$ given the queue state $\mathbf{Q}^{(\grave{y})}$, while $g(\mathbf{Q}^{(\grave{z})}, \Omega(\mathbf{Q}^{(\grave{z})})) = \mathbf{E}_{\mathbf{H}} \Big[g(\mathbf{H}, \mathbf{Q}^{(\grave{z})}, \Omega(\mathbf{H}, \mathbf{Q}^{(\grave{z})})) | \mathbf{Q}^{(\grave{z})} \Big]$ and $\Pr.[\mathbf{Q}^{(\grave{y})} | \mathbf{Q}^{(\grave{z})}, \Omega(\mathbf{Q}^{(\grave{z})})] = \mathbf{E}_{\mathbf{H}} \Big[\Pr.[\mathbf{Q}^{(\grave{y})} | \mathbf{H}, \mathbf{Q}^{(\grave{z})}, \Omega(\mathbf{H}, \mathbf{Q}^{(\grave{z})})] | \mathbf{Q}^{(\grave{z})} \Big]$ are conditional expectations of cost function $g(\mathbf{H}, \mathbf{Q}^{(\grave{z})}, \Omega(\mathbf{H}, \mathbf{Q}^{(\grave{z})}))$ and transition probability $\Pr.[\mathbf{Q}^{(\grave{y})} \ | \mathbf{H}, \mathbf{Q}^{(\grave{z})}, \Omega(\mathbf{H}, \mathbf{Q}^{(\grave{z})})]$ taken over the channel state space $\mathcal{H}$ given the queue state $\mathbf{Q}^{(\grave{z})}$, respectively. $\Omega(\mathbf{Q}^{(\grave{z})}) = \{\Omega(\mathbf{H}, \mathbf{Q}^{(\grave{z})}) | \forall \mathbf{H}\} \subseteq \mathcal{A}_{\mathrm{x}}$ is the partitioned actions of a policy Ω as the collection of $|\mathcal{H}|$ actions, where every action is mapped by policy Ω from a system state with given QSI $\mathbf{Q}^{(\grave{z})}$, and a different realization of CSI $\mathbf{H} \in \mathcal{H}$.

As a remark, note that equivalent Bellman's equation (16) represents a series of fixed-point equations, where the numbers of equations are determined by the possible values of value functions $V(\mathbf{Q}^{(\grave{z})})$, which is $|\mathcal{Q}|$. Therefore, we only need to solve $|\mathcal{Q}|$ instead of $|\mathcal{H}| \times |\mathcal{Q}|$ fixed-point equations with the reduced-state Bellman's equation (16). In order to solve one such fixed-point equation using value iteration, the R.H.S. of (16) has to be minimized with given value functions $V(\mathbf{Q}^{(\grave{y})})$. For this purpose, the R.H.S. of (16) can be written as

$$\min_{\Omega(\mathbf{Q}^{(\grave{z})})} \sum_{\mathbf{H} \in \mathcal{H}} \Pr.[\mathbf{H}] f(\mathbf{H}, \mathbf{Q}^{(\grave{z})}, \Omega(\mathbf{H}, \mathbf{Q}^{(\grave{z})})), \quad (17)$$

where

$$\begin{aligned} f(\mathbf{H}, \mathbf{Q}^{(\grave{z})}, \Omega(\mathbf{H}, \mathbf{Q}^{(\grave{z})})) = &\, g(\mathbf{H}, \mathbf{Q}^{(\grave{z})}, \Omega(\mathbf{H}, \mathbf{Q}^{(\grave{z})})) \\ + \sum_{\mathbf{Q}^{(\grave{y})} \in \mathcal{Q}} &\Pr.[\mathbf{Q}^{(\grave{y})} | \mathbf{H}, \mathbf{Q}^{(\grave{z})}, \Omega(\mathbf{H}, \mathbf{Q}^{(\grave{z})})] V(\mathbf{Q}^{(\grave{y})}). \quad (18) \end{aligned}$$

Since (17) is a decoupled objective function w.r.t. different CSI realizations $\mathbf{H}$ with a given queue state $\mathbf{Q}^{(\grave{z})}$, we need to obtain $|\mathcal{H}|$ optimal actions in order to achieve the minimization objective in the R.H.S. of equivalent Bellman equation (16), where every optimal action is w.r.t. a system state $(\mathbf{H}, \mathbf{Q}^{(\grave{z})})$ with given $\mathbf{Q}^{(\grave{z})}$ and a different CSI realization $\mathbf{H} \in \mathcal{H}$ that minimizes the value of $f(\mathbf{H}, \mathbf{Q}^{(\grave{z})}, \Omega(\mathbf{H}, \mathbf{Q}^{(\grave{z})}))$. This means that the control policy obtained by solving (16) is based on the system state $\mathbf{S}$ instead of only the queue state $\mathbf{Q}$.

Since the brute-force value iteration algorithm in Part I faces the curse of dimensionality problem, we will develop a solution with reduced complexity using linear value approximation and online stochastic learning in the next section.

III. Optimal Solution by Approximate MDP and Stochastic Learning

In this section, we will first assume that the optimal LMs are given and focus on the solution of Subproblem 1-1 in Section V-A and V-B. Then, in Section V-C, we use an online stochastic learning algorithm with two time scales to determine the optimal LMs.

A. Linear Value Function Approximation

In this section, we use linear value function approximation method to further reduce the state space.

First, we define the per-queue cost function as

$$g^{(c)}_i(\mathbf{S}^{(c)}_i, \Omega(\mathbf{S})) = \begin{cases} \omega_c Q^{(c)}_{2c-1} - \eta_c (1 - d_{\max}) T_{(2c-1)(2c)}, & \text{if } q^{(c)}_{2c-1} \in \boldsymbol{\Theta}_{\mathrm{D-u}}, \\ \omega_c Q^{(c)}_{c+D} - \eta_c (1 - d_{\max}) T_{(c+D)0}, & \text{if } q^{(c)}_{c+D} \in \boldsymbol{\Theta}_{\mathrm{Cu}}, \\ \omega_c Q^{(c)}_{0} - \eta_c (1 - d_{\max}) T_{0(2c)}, & \text{if } q^{(c)}_{0} \in \boldsymbol{\Theta}_{\mathrm{D-d}}, \\ \omega_c Q^{(c)}_{0} - \eta_c (1 - d_{\max}) T_{0(c+D)}, & \text{if } q^{(c)}_{0} \in \boldsymbol{\Theta}_{\mathrm{Cd}}. \end{cases} \quad (19)$$

Thus, the overall cost function is given by $g(\mathbf{S}, \Omega(\mathbf{S})) = \sum_{q^{(c)}_i \in \boldsymbol{\Theta}} g^{(c)}_i(\mathbf{S}^{(c)}_i, \Omega(\mathbf{S}))$ according to (15) Moreover, define $g^{(c)}_i(Q^{(c)}_i, \Omega(\mathbf{Q})) = \mathbf{E}_{\mathbf{H}} \Big[g(\mathbf{H}^{(c)}_i, Q^{(c)}_i, \Omega(\mathbf{H}, \mathbf{Q})) | \mathbf{Q} \Big]$ as the conditional per-queue cost function, which is equal to the conditional expectation of the per-queue cost function $g(\mathbf{S}^{(c)}_i, \Omega(\mathbf{S}))$ taken over the channel state space $\mathcal{H}$ given the queue state $\mathbf{Q}$.

Next, the linear approximation architecture for the value function $V(\mathbf{Q})$ is given by

$$\begin{aligned} V(\mathbf{Q}) = V(\{Q^{(c)}_i | q^{(c)}_i \in \boldsymbol{\Theta}\}) &\approx \sum_{q^{(c)}_i \in \boldsymbol{\Theta}} \sum_{q=0}^{N_{\mathrm{Q}}} \mathbf{I}[Q^{(c)}_i = q] \widetilde{V}^{(c)}_i(q) \\ &= \widetilde{\mathbf{V}}^T \mathbf{F}(\mathbf{Q}), \ \forall \mathbf{Q} \in \mathcal{Q}, \end{aligned} \quad (20)$$

where

$$\widetilde{\mathbf{V}} = \Big[\widetilde{\mathbf{V}}^{(c)}_i | q^{(c)}_i \in \boldsymbol{\Theta} \Big]^T, \ \widetilde{\mathbf{V}}^{(c)}_i = \Big[\widetilde{V}^{(c)}_i(0), \ldots, \widetilde{V}^{(c)}_i(N_{\mathrm{Q}}) \Big],$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2015.2444791, IEEE Transactions on Vehicular Technology

$$\mathbf{F}(\mathbf{Q}) = \left[\mathbf{I}[Q_i^{(c)} = 0], \dots, \mathbf{I}[Q_i^{(c)} = N_{\mathrm{Q}}] \mid q_i^{(c)} \in \Theta\right]^T .$$

Denote by $\widetilde{V}_i^{(c)}(q),\ q \in \{0, 1, \dots, N_{\mathrm{Q}}\}$ per-queue value function and $V(\mathbf{Q}),\ \mathbf{Q} \in \mathcal{Q}\}$ global value function in the rest of the paper. Therefore, $\widetilde{\mathbf{V}}_i^{(c)}$ and $\widetilde{\mathbf{V}}$ are the per-queue value function vectors for queue $q_i^{(c)}$ and all the queues in the network, respectively. Similarly, define the global value function vector as

$$\mathbf{V} = [V(\mathbf{Q}) | \mathbf{Q} \in \mathcal{Q}]^T .$$

As a remark, note that the number of global value functions is $|\mathcal{Q}| = (N_{\mathrm{Q}}+1)^{|\Theta|}$ in total, which grows exponentially with the number of queues. On the other hand, the number of per-queue value functions is $(N_{\mathrm{Q}}+1) \times |\Theta|$ in total, which grows linearly with the number of queues. Therefore, we can represent the $(N_{\mathrm{Q}}+1)^{|\Theta|}$ global value functions with $(N_{\mathrm{Q}}+1) \times |\Theta|$ per-queue value functions by the linear approximation architecture.

From (16), the key issue in deriving the optimal control actions is to obtain the global value function vector $\mathbf{V}$. With linear value function approximation, we only need to obtain the per-queue value function vector $\widetilde{\mathbf{V}}$. To illustrate the structure of our solution, we first assume we could obtain the per-queue value functions via some means (e.g., via offline value iteration) and focus on deriving the optimal action under every system state to minimize the value of $f(\mathbf{H}, \mathbf{Q}^{(\grave{z})}, \Omega(\mathbf{H}, \mathbf{Q}^{(\grave{z})}))$. Define $Q_i^{(c,\grave{y})} \in \{1, \dots, N_{\mathrm{Q}}\}$ as the local queue state of queue $q_i^{(c)}$ when the global queue state is $\mathbf{Q}^{(\grave{y})}$, i.e., $\mathbf{Q}^{(\grave{y})} = \{Q_i^{(c,\grave{y})} | q_i^{(c)} \in \Theta\}$. Therefore, according to (20) we have

$$V(\mathbf{Q}^{(\grave{y})}) \approx \sum_{q_i^{(c)} \in \Theta} \widetilde{V}_i^{(c)}(Q_i^{(c,\grave{y})}). \tag{21}$$

The optimal control action is given by the following Subproblem 1-1(a).

Subproblem 1-1(a). *For given per-queue value functions $\widetilde{\mathbf{V}}$ and LMs $\boldsymbol{\eta}$, find the optimal action $\Omega^*(\mathbf{H}, \mathbf{Q}^{(\grave{z})})$ for system state $\{\mathbf{H}, \mathbf{Q}^{(\grave{z})}\}$ that minimizes the value of $f(\mathbf{H}, \mathbf{Q}^{(\grave{z})}, \Omega(\mathbf{H}, \mathbf{Q}^{(\grave{z})}))$, which can be written as equations at the bottom of previous page, where the post-action system state $Q_i^{(c,\grave{z})}(r_i^{(c)}, x)$ is defined by the following equation*

$$Q_i^{(c,\grave{z})}(r_i^{(c)}, x) = \min\left[\max\left[0, Q_i^{(c,\grave{z})} - r_i^{(c)}\right] + x, N_{\mathrm{Q}}\right],$$

and when $Q_i^{(c,\grave{z})} = 0$ and thus $r_i^{(c)} = 0$, we define

$$Q_i^{(c,\grave{z})}(x) = \min\left[x, N_{\mathrm{Q}}\right].$$

Step (22) follows from the linear value approximation structure in (21). Step (24) holds because the arrival process for any queue $q_i^{(c)} \in \Theta_{\mathrm{u}} \bigcup \Theta_{\mathrm{Cd}}$ equals the arrival process for connection c, while the arrival process for any queue $q_0^{(c)} \in \Theta_{\mathrm{D-d}}$ depends on the departure process of the previous hop. Therefore, (8) and (9) are used to replace the local queue state transition probabilities in the L.H.S. of the equality.

Remark 2 (complexity of solution based on (24))**.** *If we directly derive the optimal action based on* (24)*, for every action $\mathbf{x} \in \mathcal{A}_{\mathrm{x}}$, we need to calculate the bid term $B_{i,c}^{(\mathbf{x})}$ for every $q_i^{(c)} \in \Theta$ and derive the summation of bids over all queues. Finally, the action $\mathbf{x}^*$ with the minimum sum bid value is selected. The complexity of the above solution is $\mathcal{O}(|\mathcal{A}_{\mathrm{x}}|)$, which grows exponentially with the number of subchannels.*

In order to deal with the exponentially increasing action space with the number of subchannels, we expand $\widetilde{V}_i^{(c)}\left(Q_i^{(c,\grave{z})}(r_i^{(c)}, x)\right)$ in (24) using Taylor expansion

$$\widetilde{V}_i^{(c)}\left(Q_i^{(c,\grave{z})}(r_i^{(c)}, x)\right) = \widetilde{V}_i^{(c)}(Q_i^{(c,\grave{z})}) + (x - r_i^{(c)})\left(\widetilde{V}_i^{(c)}(Q_i^{(c,\grave{z})})\right)', \tag{25}$$

where

$$\left(\widetilde{V}_i^{(c)}(Q_i^{(c,\grave{z})})\right)' \approx \widetilde{V}_i^{(c)}(Q_i^{(c,\grave{z})} + 1)/2 - \widetilde{V}_i^{(c)}(Q_i^{(c,\grave{z})} - 1)/2.$$

$$\begin{aligned}
\Omega^*(\mathbf{H}, \mathbf{Q}^{(\grave{z})}) &= \arg\min_{\Omega} \sum_{q_i^{(c)} \in \Theta} g_i^{(c)}\left(\mathbf{H}_i^{(c)}, Q_i^{(c,\grave{z})}, \Omega(\mathbf{H}, \mathbf{Q}^{(\grave{z})})\right) + \sum_{\mathbf{Q}^{(\grave{y})} \in \mathcal{Q}} \left\{\Pr.\left[\mathbf{Q}^{(\grave{y})} | \mathbf{H}, \mathbf{Q}^{(\grave{z})}, \Omega(\mathbf{H}, \mathbf{Q}^{(\grave{z})})\right] V(\mathbf{Q}^{(\grave{y})})\right\} \\
&= \arg\min_{\Omega} \sum_{q_i^{(c)} \in \Theta} g_i^{(c)}\left(\mathbf{H}_i^{(c)}, Q_i^{(c,\grave{z})}, \Omega(\mathbf{H}, \mathbf{Q}^{(\grave{z})})\right) + \sum_{\mathbf{Q}^{(\grave{y})} \in \mathcal{Q}} \left\{\prod_{q_i^{(c)} \in \Theta} \Pr.\left[Q_i^{(c,\grave{y})} | \mathbf{H}, \mathbf{Q}^{(\grave{z})}, \Omega(\mathbf{H}, \mathbf{Q}^{(\grave{z})})\right] \sum_{q_i^{(c)} \in \Theta} \widetilde{V}_i^{(c)}(Q_i^{(c,\grave{y})})\right\} \qquad (22) \\
&= \arg\min_{\Omega} \sum_{q_i^{(c)} \in \Theta} \left(g_i^{(c)}\left(\mathbf{H}_i^{(c)}, Q_i^{(c,\grave{z})}, \Omega(\mathbf{H}, \mathbf{Q}^{(\grave{z})})\right) + \sum_{Q_i^{(c,\grave{y})}=1}^{N_{\mathrm{Q}}} \Pr.\left[Q_i^{(c,\grave{y})} | \mathbf{H}, \mathbf{Q}^{(\grave{z})}, \Omega(\mathbf{H}, \mathbf{Q}^{(\grave{z})})\right] \widetilde{V}_i^{(c)}(Q_i^{(c,\grave{y})})\right) \\
&= \arg\min_{\Omega} \sum_{q_i^{(c)} \in \Theta_{\mathrm{u}} \bigcup \Theta_{\mathrm{Cd}}} \underbrace{\left(g_i^{(c)}\left(\mathbf{H}_i^{(c)}, Q_i^{(c,\grave{z})}, \Omega(\mathbf{H}, \mathbf{Q}^{(\grave{z})})\right) + \sum_n f_A(n) \widetilde{V}_i^{(c)}\left(Q_i^{(c,\grave{z})}(r_i^{(c)}, n)\right)\right)}_{B_{i,c}^{\Omega(\mathbf{H}, \mathbf{Q}^{(\grave{z})})},\ q_i^{(c)} \in \Theta_{\mathrm{u}} \bigcup \Theta_{\mathrm{Cd}}} \qquad (23) \\
&\quad + \sum_{q_0^{(c)} \in \Theta_{\mathrm{D-d}}} \underbrace{\left(g_0^{(c)}\left(\mathbf{H}_0^{(c)}, Q_0^{(c,\grave{z})}, \Omega(\mathbf{H}, \mathbf{Q}^{(\grave{z})})\right) + \widetilde{V}_0^{(c)}\left(Q_0^{(c,\grave{z})}(r_0^{(c)}, T_{(2c-1)0})\right)\right)}_{B_{0,c}^{\Omega(\mathbf{H}, \mathbf{Q}^{(\grave{z})})},\ q_0^{(c)} \in \Theta_{\mathrm{D-d}}} \qquad (24)
\end{aligned}$$

Therefore, (24) is equivalent to (26) at the bottom of this page, where

$$B_{(i,c)}^{(m,u)} = \begin{cases} R_{H_{(2c-1)j}^{(m,u)}} \left(\left(\widetilde{V}_{2c-1}^{(c)}(Q_{2c-1}^{(c,\grave{z})}) \right)' \right. \\ \left. +\eta_c(1-d_{\max})\mathbf{I}_1 - \left(\widetilde{V}_0^{(c)}(Q_0^{(c,\grave{z})}) \right)' \mathbf{I}_2 \right), \\ \quad \text{if } q_{2c-1}^{(c)} \in \Theta_{\mathrm{D-u}}, \\ R_{H_{(c+D)0}^{(m,u)}} \left(\left(\widetilde{V}_{c+D}^{(c)}(Q_{c+D}^{(c,\grave{z})}) \right)' + \eta_c(1-d_{\max}) \right), \\ \quad \text{if } q_{c+D}^{(c)} \in \Theta_{\mathrm{Cu}}, \\ R_{H_{0(2c)}^{(m,u)}} \left(\left(\widetilde{V}_0^{(c)}(Q_0^{(c,\grave{z})}) \right)' + \eta_c(1-d_{\max}) \right), \\ \quad \text{if } q_0^{(c)} \in \Theta_{\mathrm{D-d}}, \\ R_{H_{0(c+D)}^{(m,u)}} \left(\left(\widetilde{V}_0^{(c)}(Q_0^{(c,\grave{z})}) \right)' + \eta_c(1-d_{\max}) \right), \\ \quad \text{if } q_0^{(c)} \in \Theta_{\mathrm{Cd}}, \end{cases} \tag{27}$$

and $\mathbf{I}_1 = \mathbf{I}\Big((2c-1,j) = \big((2c-1),(2c)\big)\Big)$ and $\mathbf{I}_2 = \mathbf{I}\Big((2c-1,j) = \big((2c-1),0\big)\Big)$ with $(2c-1,j) \in \mathcal{B}_u$.

Recall that any uplink (resp. downlink) subchannel $m \in \{1,\ldots,N_{\mathrm{F}}\}$ can be allocated to at most one uplink (resp. downlink) RRG. Moreover, note that the summation index m in the first term and second term of (26) represent the index of uplink subchannel and downlink subchannel, respectively. Therefore, for every m in the first term or second term, at most one $x_{u^*}^{(m)} = 1$ while all the other $x_u^{(m)} = 0$. Now Subproblem 1-1(a) becomes determining the largest $B^{(m,u)}$ for every uplink subchannel and downlink subchannel m, where for any $m = 1,\ldots,N_{\mathrm{F}}$

$$B^{(m,u)} = \begin{cases} \sum_{q_i^{(c)} \in \Theta_{\mathrm{u}}} B_{(i,c)}^{(m,u)}, & \text{if } u \in \mathcal{U}_{\mathrm{u}}, \\ B_{(0,c)}^{(m,u)}, & \text{if } u \in \mathcal{U}_{\mathrm{d}}. \end{cases} \tag{28}$$

Algorithm 1 (solution to Subproblem 1-1(a))**.** *Given per-queue value functions $\widetilde{\mathbf{V}}$ and LMs $\boldsymbol{\eta}$, based on the observed system state $\mathbf{S}_t$ at the beginning of time slot t, the optimal action for subproblem 1-1(a) is determined as*

$$x_u^{(m)} = \begin{cases} 1, & \text{if } u \in \mathcal{U}_{\mathrm{u}},\ u = \arg\max_{u'} B^{(m,u')}, \forall u' \in \mathcal{U}_{\mathrm{u}} \\ & \text{or } u \in \mathcal{U}_{\mathrm{d}},\ u = \arg\max_{u'} B^{(m,u')}, \forall u' \in \mathcal{U}_{\mathrm{d}}, \\ 0, & \text{otherwise}, \end{cases}$$
$$\forall m = 1,\ldots,N_{\mathrm{F}}. \tag{29}$$

Remark 3 (complexity of Algorithm 1)**.** *Every uplink queue needs to compute $\sum_{(i,j)\in\mathcal{L}_i^{(c)}} |\mathcal{U}_{ij}| \times N_{\mathrm{F}}$ bids, while every downlink queue needs to compute N_{F} bids. Moreover, the BS needs to find the minimum values of $|\mathcal{U}_{\mathrm{u}}| \times N_{\mathrm{F}}$ uplink $B^{(m,u)}$ values and of $|\mathcal{U}_{\mathrm{d}}| \times N_{\mathrm{F}}$ downlink $B^{(m,u)}$ values. Therefore, the overall computational complexity of Algorithm is $\mathcal{O}((|\mathcal{U}_{\mathrm{u}}| + |\mathcal{U}_{\mathrm{d}}|) \times N_{\mathrm{F}})$ and only grows linearly with the number subchannels.*

Remark 4 (structure of Algorithm 1)**.** *The subchannel allocation solution in* (29) *has a similar structure with the MaxWeight algorithm based on Lyapunov stability approach [13]. When the MaxWeight algorithm is applied to our network model for D2D communications, the weight for each link (i,j) at each time slot is defined as its differential backlog $W_{ij} = Q_{i,t}^{(c)} - Q_{j,t}^{(c)}$. Given the link weight, we can select a RRG u^* with maximum sum over all its links of the product of link weight W_{ij} and instantaneous data rate $R_{H_{ij}^{(m,u^*)}}$. Compared with the MaxWeight algorithm, Algorithm 1 only differs in that the weight of link (i,j) is determined by the difference in the derivatives of per-queue value functions and the LM instead of the difference in queue length.*

In the above discussion, we assume that the per-queue value function vector $\widetilde{\mathbf{V}}$ is already known in Subproblem 1-1(a) and propose Algorithm 1 in order to derive the optimal control action under every system state. However, we still have to determine $\widetilde{\mathbf{V}}$ in order to solve Subproblem 1-1. For this purpose, we let $\widetilde{V}_i^{(c)}(0) = 0$, $\forall q_i^{(c)} \in \Theta$. Therefore, according to the linear approximation architecture, among the $(N_{\mathrm{Q}}+1)^{|\Theta|}$ global value functions, there are $N_{\mathrm{Q}} \times |\Theta|$ global value functions that equal to the $N_{\mathrm{Q}} \times |\Theta|$ per-queue value functions $\{\widetilde{V}_i^{(c)}(q) | \forall\ q_i^{(c)} \in \Theta,\ q = 1,\ldots,N_{\mathrm{Q}}\}$. We refer the system states of these global value functions as *representative states*, and they share the same characteristics that only one queue is non-empty while the queue length of all the other queues are zero. The set of representative states $\mathcal{Q}_{\mathrm{R}}$ is defined as

$$\mathcal{Q}_{\mathrm{R}} = \{\mathbf{Q}_{(i,c)}^{(q)} | \forall\ q_i^{(c)} \in \Theta, q = 1,\ldots,N_{\mathrm{Q}}\},$$

where $\mathbf{Q}_{(i,c)}^{(q)} = \{Q_i^{(c)} = q, Q_{i'}^{(c')} = 0 | q_{i'}^{(c')} \in \Theta \backslash q_i^{(c)}\}$ denotes the global queue state with $Q_i^{(c)} = q \in \{1,\ldots,N_Q\}$ for queue $q_i^{(c)}$ and $Q_{i'}^{(c')} = 0$ for all the other queues $q_{i'}^{(c')} \in \Theta \backslash q_i^{(c)}$.

Therefore, given the solution of Subproblem 1-1(a), we still have to solve the following Subproblem 1-1(b) in order to solve subproblem 1-1.

Subproblem 1-1(b). *Derive the per-queue value functions $\widetilde{\mathbf{V}}$ that satisfy the following equivalent Bellman's equation under every representative state $\mathbf{Q}_{(i,c)}^{(q)} \in \mathcal{Q}_{\mathrm{R}}$*

$$\theta + \widetilde{V}_i^{(c)}(q) = \min_{\Omega} \Big\{ g_i^{(c)}(q, \Omega(\mathbf{Q}_{(i,c)}^{(q)}))$$
$$+ \sum_{q_i^{(c)} \in \Theta} \sum_{Q_i^{(c,\grave{y})}=1}^{N_{\mathrm{Q}}} \mathrm{Pr.}[Q_i^{(c,\grave{y})} | \mathbf{Q}_{(i,c)}^{(q)}, \Omega(\mathbf{Q}_{(i,c)}^{(q)})] \widetilde{V}_i^{(c)}(Q_i^{(c,\grave{y})}) \Big), \tag{30}$$

where (30) *is derived by combining* (23) *with* (16).

Remark 5 (complexity reduction due to linear value function approximation)**.** *Due to linear value function approximation,*

$$\Omega^*(\mathbf{H}, \mathbf{Q}^{(\grave{z})}) = \arg\max_{\Omega} \left(\sum_{m=1}^{N_{\mathrm{F}}} \sum_{q_i^{(c)} \in \Theta_{\mathrm{u}}} \sum_{(i,j)\in\mathcal{L}_i^{(c)}} \sum_{u \in \mathcal{U}_{ij}} x_u^{(m)} B_{(i,c)}^{(m,u)} + \sum_{m=1}^{N_{\mathrm{F}}} \sum_{q_0^{(c)} \in \Theta_{\mathrm{d}}} \sum_{(0,j)\in\mathcal{L}_0^{(c)}} \sum_{u \in \mathcal{U}_{0j}} x_u^{(m)} B_{(0,c)}^{(m,u)} \right) \tag{26}$$

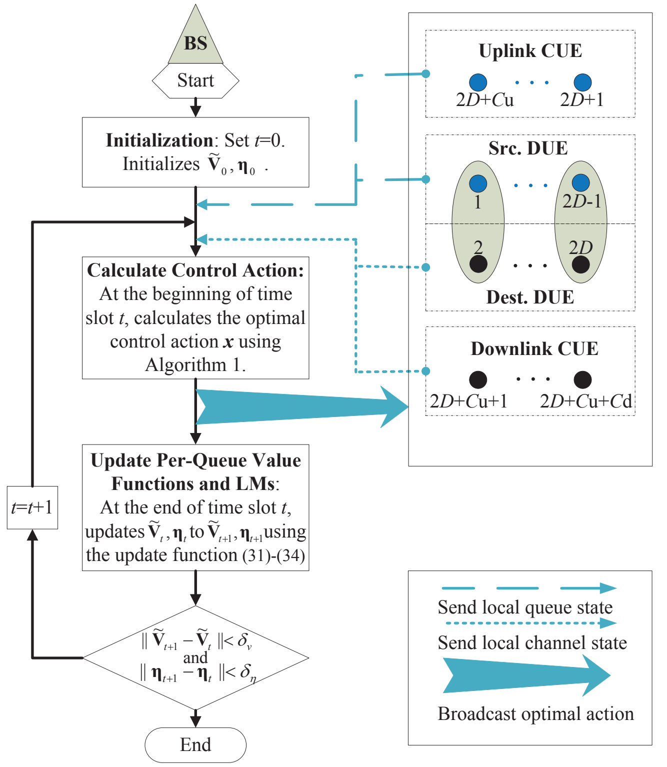
BS
Start
Initialization: Set t=0. Initializes $\widetilde{\mathbf{V}}_0, \boldsymbol{\eta}_0$.
Calculate Control Action: At the beginning of time slot t, calculates the optimal control action $\boldsymbol{x}$ using Algorithm 1.
Update Per-Queue Value Functions and LMs: At the end of time slot t, updates $\widetilde{\mathbf{V}}_t, \boldsymbol{\eta}_t$ to $\widetilde{\mathbf{V}}_{t+1}, \boldsymbol{\eta}_{t+1}$ using the update function (31)-(34)
$\| \widetilde{\mathbf{V}}_{t+1} - \widetilde{\mathbf{V}}_t \| < \delta_v$ and $\| \boldsymbol{\eta}_{t+1} - \boldsymbol{\eta}_t \| < \delta_\eta$
End
t=t+1
Uplink CUE
$2D$+Cu
$2D$+1
Src. DUE
1
$2D$-1
2
$2D$
Dest. DUE
Downlink CUE
$2D$+Cu+1
$2D$+Cu+Cd
Send local queue state
Send local channel state
Broadcast optimal action

Fig. 1. The implementation flow of Algorithm 2 with online stochastic learning (solution to Problem 1).

the following simplifications can be achieved in solving Problem 1.

- *Only* $(N_{\mathrm{Q}}+1)\times|\Theta|$ *per-queue value functions need to be stored instead of* $(N_{\mathrm{Q}}+1)^{|\Theta|}$ *global value functions.*
- *In order to determine the optimal control action in Subproblem 1-1(a) with given per-queue value functions, Algorithm 1 with a simple structure as MaxWeight algorithm can be derived with the help of Taylor's expansion.*
- *Only* $N_{\mathrm{Q}} \times |\Theta|$ *fixed point equations in* (30) *instead of* $(N_{\mathrm{Q}}+1)^{|\Theta|}$ *fixed point equations in* (16) *need to be determined in order to derive the per-queue value functions.*

B. Online Stochastic Learning

1) Solution to Problem 1: Instead of solving the equivalent Bellman's equation on the representative states (30) using offline value iteration, we will estimate $\widetilde{\mathbf{V}}$ using online stochastic learning algorithm in this section.

Remark 6 (motivation of online stochastic learning)**.** *The motivation of using online stochastic learning algorithm to update the per-queue value functions iteratively instead of using offline value iteration algorithm is that the former algorithm can solve Bellman's equation iteratively without the need of explicitly deriving the CSI probability distribution* $\Pr.[\mathbf{H}]$ *in order to calculate the "conditional cost"* $g_i^{(c)}(q,\Omega(\mathbf{Q}^{(q)}_{(i,c)}))$ *and "conditional transition probability"* $\Pr.[Q_i^{(c,\hat{y})}|\mathbf{Q}^{(q)}_{(i,c)},\Omega(\mathbf{Q}^{(q)}_{(i,c)})]$ *in* (30).

The online iterative algorithm (Algorithm 2) is given by the following, which simultaneously solves Subproblem 1-1(b) in deriving per-queue value functions and Subproblem 1-2 in deriving LMs $\boldsymbol{\eta}$. Since Algorithm 2 embeds Algorithm 1 to solve Subproblem 1-1(a), it is the complete solution for Problem 1.

Algorithm 2 (solution to Problem 1)**.** *Fig.1 illustrates the implementation flow of the overall solution with detailed steps as follows:*

- *Step 1 (**Initialization**): The per-queue value function vector* $\widetilde{\mathbf{V}}_0$ *and LM vector* $\boldsymbol{\eta}_0$ *are initialized. The subscript denote the index of time slot.*
- *Step 2 (**Calculate Control Action**): Based on the observed system state* $\mathbf{S}_t$ *and the per-queue value functions* $\widetilde{\mathbf{V}}_t$ *at time slot* t*, the optimal control action* $\mathbf{x}$ *is calculated using Algorithm 1 at the beginning of time slot* t*.*
- *Step 3 (**Update Per-Queue Value Functions and LMs**): Based on the observed system state* $\mathbf{S}_t$ *and the optimal action* $\mathbf{x}$*, the instantaneous data rate and throughput of every queue* $q_i^{(c)}$ *and its associated links* $(i,j)\in\mathcal{L}_i^{(c)}$ *are known. Based on the above information, the per-queue value functions* $\widetilde{\mathbf{V}}_t$ *and LMs* $\boldsymbol{\eta}_t$ *can be updated at the end of time slot* t *to* $\widetilde{\mathbf{V}}_{t+1}$ *and* $\boldsymbol{\eta}_{t+1}$ *using the following update function.*

$$\widetilde{V}^{(c)}_{i,t+1}(q) = \begin{cases} \left(1-\epsilon^v_{\tau_i^{(c)}(q,t)}\right)\widetilde{V}^{(c)}_{i,t}(q)+\epsilon^v_{\tau_i^{(c)}(q,t)}\Delta\widetilde{V}^{(c)}_{i,t}(q), \\ \text{if } \mathbf{Q}_t=\mathbf{Q}^{(q)}_{(i,c)}, \\ \widetilde{V}^{(c)}_{i,t}(q), \\ \text{if } \mathbf{Q}_t\neq\mathbf{Q}^{(q)}_{(i,c)}, \end{cases}$$
$$\forall q_i^{(c)}\in\Theta,\ q=1,\ldots,N_{\mathrm{Q}}, \tag{31}$$

where $\epsilon^v_{\tau_i^{(c)}(q,t)}=\sum_{t'=0}^{t}\mathbf{I}\left[\mathbf{Q}_{t'}=\mathbf{Q}^{(q)}_{(i,c)}\right]$ *and*

$$\Delta\widetilde{V}^{(c)}_{i,t}(q)=$$
$$\begin{cases} \omega_c q+\eta_c(1-d_{\max})\min\left[q,r^{(c)}_{i,t}\right] \\ +\sum_n f_A(n)\left(\widetilde{V}^{(c)}_{i,t}\left(q\left(r^{(c)}_{i,t},n\right)\right)-\widetilde{V}^{(c)}_{i,t}\left(q(n)\right)\right), \\ \text{if } q_i^{(c)}\in\Theta_{\mathrm{Cu}}\bigcup\Theta_{\mathrm{Cd}}, \\ \omega_c q+\eta_c(1-d_{\max})\min[q,r_{(2c-1)(2c),t}] \\ +\sum_n f_A(n)\left(\widetilde{V}^{(c)}_{i,t}\left(q\left(r^{(c)}_{i,t},n\right)\right)-\widetilde{V}^{(c)}_{i,t}(q(n))\right) \\ +\widetilde{V}^{(c)}_{0,t}(T_{(2c-1)0,t}), \\ \text{if } q_i^{(c)}\in\Theta_{\mathrm{D-u}}, \\ \omega_c q+\eta_c(1-d_{\max})\min\left[q,r^{(c)}_{0,t}\right]+\widetilde{V}^{(c)}_{0,t}\left(q\left(r^{(c)}_{0,t},0\right)\right), \\ \text{if } q_i^{(c)}\in\Theta_{\mathrm{D-d}}. \end{cases} \tag{32}$$

Moreover, the LMs $\eta_{c,t}$ *of every connection* c *can be updated at the end of time slot* t *to* $\eta_{c,t+1}$ *using the following function*

$$\eta_{c,t+1}=\eta_{c,t}+\epsilon^\eta_t\Delta\eta_{c,t}, \tag{33}$$

where

$$\Delta\eta_{c,t} = \begin{cases} (1-d_{\max})(\lambda_c(1-d_{\max}) - T_{(c+D)0}), \\ \text{if } c \in \mathcal{C}_{\mathrm{Cu}}, \\ (1-d_{\max})(\lambda_c(1-d_{\max}) - T_{0(c+D)}), \\ \text{if } c \in \mathcal{C}_{\mathrm{Cd}}, \\ (1-d_{\max})(\lambda_c(1-d_{\max}) - T_{(2c-1)(2c)} - T_{0(2c)}), \\ \text{if } c \in \mathcal{C}_{\mathrm{D}}. \end{cases} \tag{34}$$

In the above equations, $(\{\epsilon_t^v\}, \{\epsilon_t^\eta\})$ *are the sequences of step sizes, which satisfy*

$$\sum_{t=0}^{\infty} \epsilon_t^v = \infty,\ \epsilon_t^v > 0,\ \lim_{t\to\infty} \epsilon_t^v = 0,$$

$$\sum_{t=0}^{\infty} \epsilon_t^\eta = \infty,\ \epsilon_t^\eta > 0,\ \lim_{t\to\infty} \epsilon_t^\eta = 0,$$

$$\sum_{t=0}^{\infty} \left[(\epsilon_t^v)^2 + (\epsilon_t^\eta)^2\right] < \infty,\ \textit{and}\ \lim_{t\to\infty} \frac{\epsilon_t^\eta}{\epsilon_t^v} = 0.$$

*Step 4 (**Termination**):If* $\|\widetilde{\mathbf{V}}_{t+1} - \widetilde{\mathbf{V}}_t\| < \delta_v$ *and* $\|\boldsymbol{\eta}_{t+1} - \boldsymbol{\eta}_t\| < \delta_\eta$*, stop; otherwise, set* $t := t+1$ *and go to Step 2.*

Remark 7 (complexity and implementation consideration of Algorithm 2)**.** *In Algorithm 2, we assume that it is centralized implemented by the BS. The BS needs to store* $(N_{\mathrm{Q}}+1)\times|\Theta|$ *per-queue value functions and C LMs. The computational complexity of Algorithm 2 at each time slot is the sum of two parts: (1) the computational complexity of determining the optimal action according to Algorithm 1, which is* $\mathcal{O}((\mathcal{U}_u + \mathcal{U}_d)\times N_{\mathrm{F}})$ *as given in Remark 3; (2) the computational complexity of updating the per-queue value functions. Note that* $\widetilde{V}_{i,t}(c)(q)$ *is only updated to a different value at any time slot t when the global queue state* $\widetilde{\mathbf{Q}}_t$ *is the representative state* $\widetilde{\mathbf{Q}}^{(q)}_{i,c}$ *according to (33). This implies that at most one per-queue value function shall be updated to a different value with computational complexity* $\mathcal{O}(N_{\mathrm{Q}}+1)$ *at any time slot, while all the other per-node value functions remain the same. Therefore, the overall computational complexity of Algorithm 2 is at most* $\mathcal{O}((N_{\mathrm{Q}}+1)+(|\mathcal{U}_{\mathrm{u}}|+|\mathcal{U}_{\mathrm{d}}|)\times N_{\mathrm{F}})$ *at each time slot, which grows linearly with buffer capacity, the number of RRGs and subchannels. The memory requirement and computational complexity of Algorithm 2 are greatly reduced compared to those of the offline value iteration algorithm. Moreover, the structure of Algorithm 2 enables distributed implementation, where the per-queue value functions are distributively maintained at nodes that maintain the corresponding queues. The optimal action in Algorithm 1 can be derived using an auction mechanism in which each src. DUE and uplink CUE sends bid values to the BS for the queues that it maintains. In this way, the computation task can be offloaded to the UEs from the BS. However, larger signaling overhead need to be involved compared to the centralized implementation.*

2) Convergence Analysis: In this section, we shall establish technical conditions for the almost-sure convergence of the online stochastic learning algorithm (Algorithm 2). Recall that the purpose of Algorithm 2 is to iteratively derive the per-queue value function vector $\widetilde{\mathbf{V}}$ in subproblem 1-1(b) and LMs $\boldsymbol{\eta}$ in subproblem 1-2, so that Problem 1 can be solved. Given $\boldsymbol{\eta}^*$, subproblem 1-1 is an unconstraint MDP problem, so learning algorithms in [14] apply to update $\widetilde{\mathbf{V}}$, which is done in Algorithm 2. However, the correct $\boldsymbol{\eta}^*$ needs to be derived. We do this by a gradient ascent in the dual (i.e., Lagrange multiplier) space in view of subproblem 1-2. Since we have two different step size sequences $\{\epsilon_t^v\}$, $\{\epsilon_t^\eta\}$, and $\{\epsilon_t^\eta\} = \mathbf{o}(\{\epsilon_t^v\})$, the LM's update is carried out simultaneously with the per-queue value function's update but over a slower timescale. Here we are using the formalism of two timescale stochastic approximation from [15]. During the per-queue value functions' update (timescale I), we have $\eta_{c,t+1} - \eta_{c,t} = \mathcal{O}(\{\epsilon_t^\eta\}) = \mathbf{o}(\{\epsilon_t^v\}), \forall c \in \mathcal{C}$ and hence the LMs appear to be quasi-static during the per-node value functions' update in (31) and (32). On the other hand, since the per-queue value functions will be updated much faster than the LMs due to $\frac{\epsilon_t^\eta}{\epsilon_t^v} \to 0$, during the LMs' update in (33) and (34) (timescale II), the 'primal' minimization carried out by the learning algorithm for MDPs in (31) and (32) is seen as having essentially equilibrated. Therefore, we will give the convergence of per-queue value function over timescale I and LMs over timescale II in Lemma 2 and Lemma 3, respectively.

Before Lemma 2 is given, we first give the relationship between the global value function vector $\mathbf{V}$ and per-queue value function vector $\widetilde{\mathbf{V}}$ in matrix form for ease of notation:

$$\mathbf{V} = \mathbf{M}\widetilde{\mathbf{V}} \text{ and } \widetilde{\mathbf{V}} = \mathbf{M}^{\dagger}\mathbf{V},$$

where $\mathbf{M} \in \mathbb{R}^{|\mathcal{Q}|\times(N_{\mathrm{Q}}+1)|\Theta|}$ with the $\check{z}$th row ($\check{z} = 1, \ldots, |\mathcal{Q}|$) equals to $\mathbf{F}^T(\mathbf{Q}^{(\check{z})})$. Therefore, the first equation above follows directly from (20). The second equation, on the other hand, uses the matrix $\mathbf{M}^{\dagger} \in \mathbb{R}^{(N_{\mathrm{Q}}+1)|\Theta|\times|\mathcal{Q}|}$ to select $N_{\mathrm{Q}}|\Theta|$ elements from $\mathbf{V}$ which correspond to the representative states. Specifically, $\mathbf{M}^{\dagger}$ has only one element of 1 in each row while all the other elements equal 0, and the position of 1 in the $(q + (id(i,c)-1)(N_{\mathrm{Q}}+1))$th row ($q \in \{1, \ldots, N_{\mathrm{Q}}\}$ and $id(i,c) \in \{1, \ldots, |\Theta|\}$ is the index of queue $q_i^{(c)}$ within set Θ) corresponds to the position of the representative state $V(\mathbf{Q}^{(q)}_{i,c})$ in the global queue state vector $\mathbf{V}$. Moreover, the position of 1 in the $(1 + N_{\mathrm{Q}} \times id(i,c))$th row corresponds to the position of global queue state with all queues being empty $V(\{Q_i^{(c)} = 0 | q_i^{(c)} \in \Theta\})$ in the global queue state vector $\mathbf{V}$.

Now the vector form of the equivalent Bellman equation (30) under all the representative states can be written as

$$\theta\mathbf{e} + \widetilde{\mathbf{V}}_{\infty}(\boldsymbol{\eta}) = \mathbf{M}^{\dagger}\mathbf{T}\left(\boldsymbol{\eta}, \mathbf{M}\widetilde{\mathbf{V}}_{\infty}(\boldsymbol{\eta})\right), \tag{35}$$

where $\mathbf{e}$ is a $(N_Q+1)|\Theta| \times 1$ vector with all elements equal to 1. The mapping $\mathbf{T}$ is defined as

$$\mathbf{T}(\boldsymbol{\eta}, \mathbf{V}) = \min_{\Omega}\left[\mathbf{g}(\boldsymbol{\eta}, \Omega) + \mathbf{P}(\Omega)\mathbf{V}\right],$$

where $\mathbf{g}(\boldsymbol{\eta}, \Omega)$ is the vector form of function $g(\mathbf{Q}, \Omega(\mathbf{Q}))$ defined in (16), and $\mathbf{P}(\Omega)$ is the matrix form of transition probability $\Pr.[\mathbf{Q}^{(\check{y})}|\mathbf{Q}^{(\check{z})}, \Omega(\mathbf{Q}^{(\check{z})})]$ defined in (16).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2015.2444791, IEEE Transactions on Vehicular Technology

Lemma 1 (convergence of per-queue value function learning over timescale I)**.** *Denote*

$$\mathbf{A}_{t-1} = (1-\epsilon^v_{t-1})\mathbf{I} + \mathbf{M}^\dagger \mathbf{P}(\Omega_t)\mathbf{M}\epsilon_{t-1},$$

$$\mathbf{B}_{t-1} = (1-\epsilon^v_{t-1})\mathbf{I} + \mathbf{M}^\dagger \mathbf{P}(\Omega_{t-1})\mathbf{M}\epsilon_{t-1},$$

where Ω_t is the unichain control policy at slot t, $\mathbf{P}(\Omega_t)$ is the transition matrix under the unichain system control policy, and $\mathbf{I}$ is the identity matrix. If for the entire sequence of control policies $\{\Omega_t\}$ there exists $\delta_\beta > 0$ and some positive integer β such that

$$[\mathbf{A}_\beta \cdots \mathbf{A}_1]_{(k,\grave{I})} \geq \delta_\beta,$$
$$[\mathbf{B}_\beta \cdots \mathbf{B}_1]_{(k,\grave{I})} \geq \delta_\beta, \forall k, \tag{36}$$

where $[\cdot]_{(k,\grave{I})}$ denotes the element in the k-th row and the $\grave{I}$-th column(where $\grave{I}$ corresponds to the queue state $\mathbf{Q}^{(\grave{I})}$ that all queues are empty), and $\delta_t = \mathcal{O}(\epsilon^v_t)$. Then the following statements are true.

1) *The update of the per-queue value function vector will converge almost surely for any given initial parameter vector $\widetilde{\mathbf{V}}_0$ and LM vector $\boldsymbol{\eta}$, i.e.,*

$$\lim_{t\to\infty} \widetilde{\mathbf{V}}_t(\boldsymbol{\eta}) = \widetilde{\mathbf{V}}_\infty(\boldsymbol{\eta}).$$

2) *The steady-state per-queue value function vector $\widetilde{\mathbf{V}}_\infty$ satisfies* (35)*.*

Proof: Refer to Appendix A. ∎

Remark 8 (interpretation of the conditions in Lemma 2)**.** *Note that $\mathbf{A}_t$ and $\mathbf{B}_t$ are related to an equivalent transition matrix of the underlying Markov chain. Equation* (36) *simply means that the system state $\mathbf{S}^I$ representing any system state where all the queue length are zero is accessible from all the system states after some finite number of transition steps. This is a very mild condition and is satisfied in most of the cases we are interested.*

Lemma 2 (convergence of LMs update over timescale II)**.** *The iteration on the vector of LMs $\boldsymbol{\eta}$ converges almost surely to the set of maxima of $G(\boldsymbol{\eta})$. Suppose the LMs converge to $\boldsymbol{\eta}^*$, then $\boldsymbol{\eta}^*$ satisfies the dropping probability constraints in Problem 1.*

Proof: Refer to Appendix B. ∎

IV. Simulation Results

In this section, we compare our proposed approximate MDP solution with online stochastic learning (Algorithm 2) to the approximate MDP solution with offline value iteration based on the Bellman's equation (30) and two other reference subchannel allocation algorithms. One is the CSI-only algorithm, in which the RRG selection is only adaptive to CSI and a subchannel is allocated to the RRG with the maximum sum over all its link transmission rates at every time slot. The other is the MaxWeight algorithm which is adaptive to both CSI and QSI as discussed in Remark 4. The offline value iteration algorithm can find the per-queue value functions and optimal policy that satisfy the Bellman's equation (30), and therefore provide the performance upper bound. However, the CSI probability distribution $\Pr.[\mathbf{H}]$ needs to be derived, which is very hard when there are more than two links in an RRG as discussed in Part I. The simulation parameter setting is the same with that in the Part I of the paper [1] except that the buffer size is set to be $N_\mathrm{Q} = 10$ packets and there are $N_\mathrm{F} = 10$ independent subchannels.

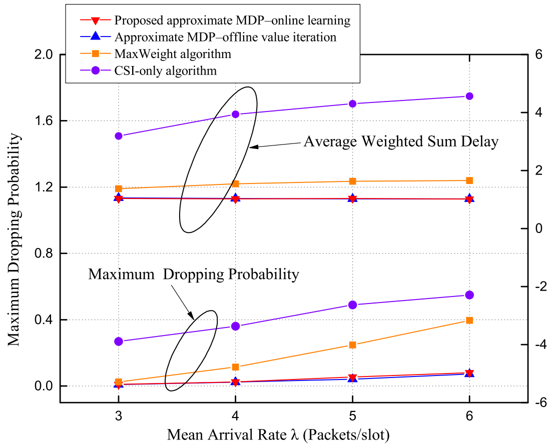

Fig. 2. The average weighted sum delay and the maximum dropping probability over all connections versus the mean arrive rate λ with $C = 3(D = C_u = C_d = 1)$ and $d_{\max} = 0.1$.

Fig.2 shows the average weighted sum delay and the maximum dropping probability over all connections versus the mean arrive rate λ for the simple network in Fig.1 of Part I [1] where the number of connections $C = 3$ (one D2D connection, one cellular uplink connection and one cellular downlink connection). The dropping probability constraint is set to $d_{\max} = 0.1$. It can be observed that the performance of the proposed approximate MDP solution with online learning is close to that of the approximate MDP solution using value iteration algorithm, where both solutions have lower average delay than the two reference subchannel allocation algorithms. Although the dropping probability of the MaxWeight algorithm is almost the same with both approximate MDP solutions in light traffic load regime, it grows significantly higher than the approximate MDP solutions when λ increases beyond 4 packets/slot. This is because the approximate MDP solutions will guarantee that the dropping probability is no larger than the constraint $d_{\max}$ if this can be achieved by any policy under the given mean arrival rate λ. As a consequence, the proposed approximate MDP approach with online learning is an effective method to reduce the complexity and achieve an optimal performance (with regard to the offline value iteration algorithm) while guaranteeing the dropping probability constraint.

Fig.3 shows the average weighted sum delay and the maximum dropping probability over all connections versus the number of connections C with $\lambda = 1$ packets/slot. The dropping probability constraint is set to $d_{\max} = 0.3$. Note that the performance of the approximate MDP solution with value iteration algorithm is not shown in Fig.3. This is because

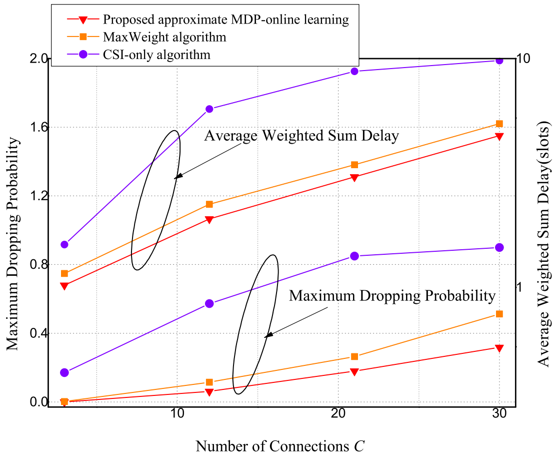

Fig. 3. The average weighted sum delay and the maximum dropping probability over all connections versus the number of connections C with $\lambda = 1$ packets/slot and $d_{\max} = 0.3$. The number of connections is $C = 3(D = C_u = C_d = 1)$, $C = 12(D = C_u = C_d = 4)$, $C = 21(D = C_u = C_d = 7)$, $C = 30(D = C_u = C_d = 10)$, respectively.

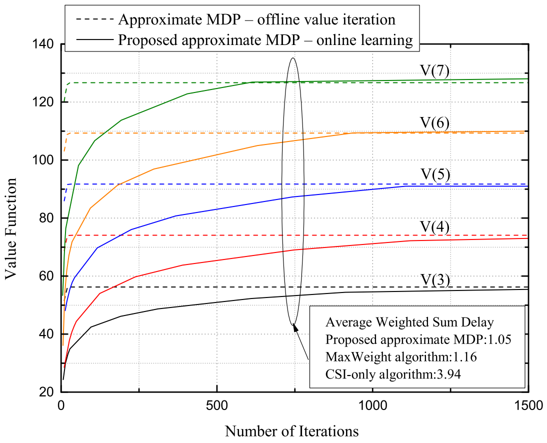

Fig. 4. Illustration of the convergence property of the proposed online stochastic learning algorithm and brute-force offline value iteration algorithm with $C = 3(D = C_u = C_d = 1)$ and $\lambda = 5$ packets/slot.

the above policy can not be derived when the number of connections becomes large, since there are some RRGs with more than two links whose CSI probability distributions are hard to obtain. It's obvious that our proposed approximate MDP solution with online learning algorithm performs better in average weighted sum delay and the maximum dropping probability than the two other reference algorithms. Among the three algorithms, the CSI-only algorithm performs the worst since it does not take the QSI into account. The dropping probability of the MaxWeight algorithm exceeds $d_{\max} = 0.3$ with the increasing number of connections, while our proposed algorithm can always keep it under the constraint. When the connection number $C = 30$, our simulation results show that compared to the MaxWeight algorithm, the average weighted sum delay and the maximum dropping probability achieved by our proposed approximate MDP-online learning algorithm are decreased by 10% and 38%, respectively.

Fig.4 shows the convergence property of the proposed online stochastic learning algorithm and brute-force offline value iteration algorithm. We plot a portion of per-queue value functions of the 3 connections versus the scheduling slot index at a mean arrive rate $\lambda = 5$ packets/slot. It can be seen that the online stochastic learning algorithm converges fast and after 1000 iterations the values are close to the final converged results. The average weighted sum delay corresponding to the average per-queue value function at the 750-th iteration is smaller than the other two reference algorithms. Moreover, it is clear that the value functions calculated online quickly approach the value functions calculated offline when the number of iteration grows.

V. Conclusion

In this pair of papers, we considered a delay-optimal dynamic mode selection and resource allocation algorithm under dropping probability constraint for network assisted D2D communications with bursty traffic arrival, which is cast into an infinite horizon average reward CMDP in the first part of this work. In the second part of this work, we addressed the issue of exponential memory requirement and computational complexity by using linear value approximation techniques to reduce the state space. Moreover, online stochastic learning algorithm with two time scales was adopted to update the value functions and LMs based on the real-time observations of CSI and QSI. The obtained solution has a simple structure with a computational complexity of $\mathcal{O}((N_{\mathrm{Q}} + 1) + (|\mathcal{U}_{\mathrm{u}}| + |\mathcal{U}_{\mathrm{d}}|) \times N_{\mathrm{F}})$, which grows linearly with the buffer capacity, number of RRGs and subchannels. We proved that under some mild conditions, the proposed approximate MDP and online stochastic learning solution converges almost surely (with probability 1) to a global optimal solution. Simulation results show that the proposed approach outperforms the conventional CSI-only scheme and throughput-optimal scheme (MaxWeight algorithm).

Appendix

A. Proof of Lemma 2

Since each representative state is updated comparably often in the asynchronous learning algorithm, quoting the conclusion in [17], the convergence property of the asynchronous update and the synchronous update is the same. Therefore, we consider the convergence of the related synchronous version for simplicity in this proof. It is easy to see that the per-queue value function vector $\widetilde{\mathbf{V}}_t$ is bounded almost surely during the iterations of the algorithm. In the following, we first introduce and prove the following lemma on the convergence of learning noise.

Lemma 3. *Define*

$$\mathbf{q}_t = \mathbf{M}^{\dagger}[\mathbf{g}(\Omega_t) + \mathbf{P}(\Omega_t)\mathbf{M}\widetilde{\mathbf{V}}_t - \mathbf{M}\widetilde{\mathbf{V}}_t - \mathbf{T}_0(\mathbf{M}\widetilde{\mathbf{V}}_t)\mathbf{e}],$$

where $\mathbf{T}_0(\mathbf{V}) = \min_{\Omega}\left[\mathbf{g}_{\grave{I}}(\Omega) + \mathbf{P}_{\grave{I}}(\Omega)\mathbf{V}\right]$ *denotes the mapping on the queue state* $\mathbf{Q}^{(\grave{I})}$*, where* $\mathbf{g}_{\grave{I}}(\Omega)$ *is the vector form of function* $g(\mathbf{Q}^{(\grave{I})}, \Omega(\mathbf{Q}^{(\grave{I})}))$*,* $\mathbf{P}_{\grave{I}}(\Omega)$ *is the matrix form* *of transition probability* $\Pr.[\mathbf{Q}^{(\grave{y})}|\mathbf{Q}^{(\grave{I})}, \Omega(\mathbf{Q}^{(\grave{I})})]$. *When the number of iterations $t \geq j \to \infty$, the procedure of update can be written as follows with probability 1:*

$$\widetilde{\mathbf{V}}_{t+1} = \widetilde{\mathbf{V}}_j + \sum_{i=j}^{t} \epsilon_i^v \mathbf{q}_i.$$

Proof: The synchronous update of per-queue value function vector can be written in the following vector form:

$$\widetilde{\mathbf{V}}_{t+1} = \widetilde{\mathbf{V}}_t + \epsilon_i^v \mathbf{M}^\dagger[\mathbf{g}(\Omega_t) + \mathbf{J}_t\mathbf{M}\widetilde{\mathbf{V}}_t - \mathbf{M}\widetilde{\mathbf{V}}_t - \mathbf{T}_0(\mathbf{M}\widetilde{\mathbf{V}}_t)\mathbf{e}],$$

where the matrix $\mathbf{J}_t$ is the matrix form of queue state transition probability $\Pr.[\mathbf{Q}^{(\grave{y})}|\mathbf{H}_{t(\grave{z})}, \mathbf{Q}^{(\grave{z})}, \Omega(\mathbf{H}_{t(\grave{z})}, \mathbf{Q}^{(\grave{z})})]$ with given $\mathbf{H}_{t(\grave{z})}$ in each row, which is the real-time observation of channel state at time slot $t(\grave{z})$ with queue state $\mathbf{Q}_{t(\grave{z})} = \mathbf{Q}^{(\grave{z})}$. Define

$$\mathbf{Y}_t = \mathbf{M}^\dagger[\mathbf{g}(\Omega_t) + \mathbf{J}_t\mathbf{M}\widetilde{\mathbf{V}}_t - \mathbf{M}\widetilde{\mathbf{V}}_t - \mathbf{T}_0(\mathbf{M}\widetilde{\mathbf{V}}_t)\mathbf{e}],$$

and $\delta\mathbf{Z}_t = \mathbf{q}_t - \mathbf{Y}_t$ and $\mathbf{Z}_t = \sum_{i=j}^{t} \epsilon_i^v \delta\mathbf{Z}_i$. The online value function estimation can be rewritten as

$$\widetilde{\mathbf{V}}_{t+1} = \widetilde{\mathbf{V}}_t + \epsilon_i^v\mathbf{Y}_t = \widetilde{\mathbf{V}}_t + \epsilon_i^v\mathbf{q}_t - \epsilon_i^v\delta\mathbf{Z}_t = \widetilde{\mathbf{V}}_j + \sum_{i=j}^{t} \epsilon_i^v\mathbf{q}_i - \mathbf{Z}_t. \tag{37}$$

Our proof of Lemma 4 can be divided into the following steps:

1) Step 1: Letting $\mathcal{F}_t = \sigma(\mathbf{V}_m, m \leq t)$, its easy to see that $\mathbf{E}[\delta\mathbf{Z}_t|\mathcal{F}_{t-1}] = 0$. Thus, $\{\delta\mathbf{Z}_t|\forall t\}$ is a Martingale difference sequence and $\{\mathbf{Z}_t|\forall t\}$ is a Martingale sequence. Moreover, $\mathbf{Y}_t$ is an unbiased estimation of $\mathbf{q}_t$ and the estimation noise is uncorrelated.
2) Step 2: According to the uncorrelated estimation error from step 1, when $j \to \infty$ we have
$$\mathbf{E}[|\mathbf{Z}_t|^2|\mathcal{F}_{j-1}] = \mathbf{E}[|\sum_{i=j}^{t} \epsilon_i^v\delta\mathbf{Z}_i|^2|\mathcal{F}_{j-1}]$$
$$= \sum_{i=j}^{t}\mathbf{E}[|\epsilon_i^v\delta\mathbf{Z}_i|^2|\mathcal{F}_{j-1}] = \widetilde{\mathbf{Z}}\sum_{i=j}^{t}(\epsilon_i^v)^2 \to 0,$$
where $\widetilde{\mathbf{Z}} \geq \max_{j \leq i \leq t}\mathbf{E}[|\delta\mathbf{Z}_i|^2|\mathcal{F}_{j-1}]$ is a bounded constant vector and the convergence of $\widetilde{\mathbf{Z}}\sum_{i=j}^{t}(\epsilon_i^v)^2$ is from the definition of sequence $\{\epsilon_i^v\}$.
3) Step 3: From step 1, $\{\delta\mathbf{Z}_t|\forall t\}$ is a Martingale sequence. Hence, according to the inequality of Martingale sequence, we have
$$\Pr[\sup_{j \leq i \leq t}|\mathbf{Z}_i| \geq \lambda|\mathcal{F}_{j-1}] \leq \frac{\mathbf{E}[|\mathbf{Z}_t|^2|\mathcal{F}_{j-1}]}{\lambda^2}, \forall\lambda > 0.$$
From the conclusion of step 2, we have
$$\lim_{j\to\infty}\Pr[\sup_{j \leq i \leq t}|\mathbf{Z}_i| \geq \lambda|\mathcal{F}_{j-1}] = 0, \forall\lambda > 0.$$
Hence, from (37), we almost surely have $\widetilde{\mathbf{V}}_{t+1} = \widetilde{\mathbf{V}}_j + \sum_{i=j}^{t}\epsilon_i^v\mathbf{q}_i$ when $j \to \infty$.

■

Moreover, the following lemma is about the limit of sequence $\{\mathbf{q}_t\}$.

Lemma 4. *Suppose the following two inequalities are true for $t = m, m+1, ..., m+n$:*

$$\mathbf{g}(\Omega_t) + \mathbf{P}(\Omega_t)\mathbf{M}\widetilde{\mathbf{V}}_t \leq \mathbf{g}(\Omega_{t-1}) + \mathbf{P}(\Omega_{t-1})\mathbf{M}\widetilde{\mathbf{V}}_t, \tag{38}$$

$$\mathbf{g}(\Omega_{t-1}) + \mathbf{P}(\Omega_{t-1})\mathbf{M}\widetilde{\mathbf{V}}_{t-1} \leq \mathbf{g}(\Omega_t) + \mathbf{P}(\Omega_t)\mathbf{M}\widetilde{\mathbf{V}}_{t-1}, \tag{39}$$

then we have

$$\lim_{t\to+\infty}\mathbf{q}_t = 0.$$

Proof: From (38) and (39), we have

$$\begin{aligned}\mathbf{q}_t &= \mathbf{M}^\dagger[\mathbf{g}(\Omega_t) + \mathbf{P}(\Omega_t)\mathbf{M}\widetilde{\mathbf{V}}_t - \mathbf{M}\widetilde{\mathbf{V}}_t - \omega_t\mathbf{e}]\\ &\leq \mathbf{M}^\dagger[\mathbf{g}(\Omega_{t-1}) + \mathbf{P}(\Omega_{t-1})\mathbf{M}\widetilde{\mathbf{V}}_t - \mathbf{M}\widetilde{\mathbf{V}}_t - \omega_t\mathbf{e}],\end{aligned}$$

$$\begin{aligned}\mathbf{q}_{t-1} &= \mathbf{M}^\dagger[\mathbf{g}(\Omega_{t-1}) + \mathbf{P}(\Omega_{t-1})\mathbf{M}\widetilde{\mathbf{V}}_{t-1} - \mathbf{M}\widetilde{\mathbf{V}}_{t-1} - \omega_{t-1}\mathbf{e}]\\ &\leq \mathbf{M}^\dagger[\mathbf{g}(\Omega_t) + \mathbf{P}(\Omega_t)\mathbf{M}\widetilde{\mathbf{V}}_{t-1} - \mathbf{M}\widetilde{\mathbf{V}}_{t-1} - \omega_{t-1}\mathbf{e}],\end{aligned}$$

where $\omega_t = \mathbf{T}_0(\mathbf{M}\widetilde{\mathbf{V}}_t)$. According to Lemma 4, we have

$$\widetilde{\mathbf{V}}_t = \widetilde{\mathbf{V}}_{t-1} + \epsilon_{t-1}^v\mathbf{q}_{t-1}.$$

Therefore

$$\begin{aligned}\mathbf{q}_t &\geq [(1-\epsilon_{t-1}^v)\mathbf{I} + \mathbf{M}^\dagger\mathbf{P}(\Omega_t)\mathbf{M}\epsilon_{t-1}]\mathbf{q}_{t-1} + \omega_{t-1}\mathbf{e} - \omega_t\mathbf{e}\\ &= \mathbf{A}_{t-1}\mathbf{q}_{t-1} + \omega_{t-1}\mathbf{e} - \omega_t\mathbf{e},\end{aligned}$$

$$\begin{aligned}\mathbf{q}_t &\leq [(1-\epsilon_{t-1}^v)\mathbf{I} + \mathbf{M}^\dagger\mathbf{P}(\Omega_{t-1})\mathbf{M}\epsilon_{t-1}]\mathbf{q}_{t-1} + \omega_{t-1}\mathbf{e} - \omega_t\mathbf{e}\\ &= \mathbf{B}_{t-1}\mathbf{q}_{t-1} + \omega_{t-1}\mathbf{e} - \omega_t\mathbf{e}.\end{aligned}$$

Thus, we have

$$\mathbf{A}_{t-1}\cdots\mathbf{A}_{t-\beta}\mathbf{q}_{t-\beta} - C_1\mathbf{e} \leq \mathbf{q}_t \leq \mathbf{B}_{t-1}\cdots\mathbf{B}_{t-\beta}\mathbf{q}_{t-\beta} - C_1\mathbf{e}$$
$$\Rightarrow (1-\delta_\beta)(\min\mathbf{q}_{t-\beta})\mathbf{e} \leq \mathbf{q}_t + C_1\mathbf{e} \leq (1-\delta_\beta)(\max\mathbf{q}_{t-\beta})\mathbf{e}$$
$$\Rightarrow \begin{cases}\min\mathbf{q}_t + C_1 \geq (1-\delta_\beta)\min\mathbf{q}_{t-\beta}\\ \max\mathbf{q}_t + C_1 \leq (1-\delta_\beta)\max\mathbf{q}_{t-\beta}\end{cases}$$
$$\Rightarrow \max\mathbf{q}_t - \min\mathbf{q}_t \leq (1-\delta_\beta)(\max\mathbf{q}_{t-\beta} - \min\mathbf{q}_{t-\beta})$$
$$\Rightarrow |\mathbf{q}_t^k| \leq \max\mathbf{q}_t - \min\mathbf{q}_t \leq C_2(1-\delta_\beta), \forall k.$$

Then we have

$$0 \leq |\mathbf{q}_{m+n}^k| \leq C_3\prod_{i=0}^{\lfloor n/\beta\rfloor-1}(1-\delta_{m+i\beta}) = 0, \forall k, \tag{40}$$

where the first step is due to conditions on matrix sequence $\mathbf{A}_t$ and $\mathbf{B}_t$, $\min\mathbf{q}_t$ and $\max\mathbf{q}_t$ denote the minimum and maximum elements in $\mathbf{q}_t$, respectively, $\mathbf{q}_t^k$ denotes the kth element of the vector $\mathbf{q}_t$, $|\mathbf{q}_t^k| \leq \max\mathbf{q}_t - \min\mathbf{q}_t$ is due to $\min\mathbf{q}_t \leq 0$, and C_1,C_2 and C_3 are constants. According to the property of sequence $\{\epsilon_t^v\}$, we have

$$\lim_{t\to+\infty}\prod_{i=0}^{\lfloor t/\beta\rfloor-1}(1-\epsilon_{i\beta}) = 0.$$

And note that $\delta_t = \mathcal{O}(\epsilon_t^v)$, from (40), we have

$$\lim_{t\to+\infty}\mathbf{q}_t^k = 0, \forall k.$$

Summarize the conclusions above, we have

$$\lim_{t\to+\infty}\mathbf{q}_t = 0.$$

■

Therefore, (35) is straightforward when $\mathbf{q}_t \to 0$. This completes the proof.

B. Proof of Lemma 3

Due to the separation of time scale, the primal update of the per-node value function converges to $\widetilde{\mathbf{V}}_\infty(\boldsymbol{\eta})$ with respect to current LM $\boldsymbol{\eta}$ [15]. By Lemma 4.2 in [18], $G(\boldsymbol{\eta})$ is a concave and continuously differentiable except at finitely many points where both right and left derivatives exist. Since subchannel allocation policy is discrete, we have $\Omega^*(\boldsymbol{\eta}) = \Omega^*(\boldsymbol{\eta} + \triangle_\eta)$, that is, $\bigtriangledown_\eta = \frac{\Omega^*(\boldsymbol{\eta}+\triangle_\eta)-\Omega^*(\boldsymbol{\eta})}{\triangle_\eta} = 0$, therefore

$$\frac{\partial G(\boldsymbol{\eta}_t)}{\partial \boldsymbol{\eta}_t} = \begin{cases} \mathbf{E}^{\Omega^*(\boldsymbol{\eta}_t)}[(1-d_{\max})(\lambda_c(1-d_{\max}) - T_{(c+D)0})], \\ \text{if } c \in \mathcal{C}_{\mathrm{Cu}}, \\ \mathbf{E}^{\Omega^*(\boldsymbol{\eta}_t)}[(1-d_{\max})(\lambda_c(1-d_{\max}) - T_{0(c+D)})], \\ \text{if } c \in \mathcal{C}_{\mathrm{Cd}}, \\ \mathbf{E}^{\Omega^*(\boldsymbol{\eta}_t)}[(1-d_{\max})(\lambda_c(1-d_{\max}) - T_{(2c-1)(2c)} - T_{0(2c)})], \\ \text{if } c \in \mathcal{C}_{\mathrm{D}}, \end{cases}$$

where $\Omega^*(\boldsymbol{\eta}_t) = \arg\min_\Omega G(\boldsymbol{\eta}_t)$. Using standard stochastic approximation theorem [14], the dynamics of the LM update equation in (33) can be represented by the following ordinary differential equation (ODE):

$$\boldsymbol{\eta}_t^{'} = \begin{cases} \mathbf{E}^{\Omega^*(\boldsymbol{\eta}_t)}[(1-d_{\max})(\lambda_c(1-d_{\max}) - T_{(c+D)0})], \\ \text{if } c \in \mathcal{C}_{\mathrm{Cu}}, \\ \mathbf{E}^{\Omega^*(\boldsymbol{\eta}_t)}[(1-d_{\max})(\lambda_c(1-d_{\max}) - T_{0(c+D)})], \\ \text{if } c \in \mathcal{C}_{\mathrm{Cd}}, \\ \mathbf{E}^{\Omega^*(\boldsymbol{\eta}_t)}[(1-d_{\max})(\lambda_c(1-d_{\max}) - T_{(2c-1)(2c)} - T_{0(2c)})], \\ \text{if } c \in \mathcal{C}_{\mathrm{D}}. \end{cases}$$

Therefore, we show that the above ODE can be expressed as $\boldsymbol{\eta}_t^{'} = \bigtriangledown G(\boldsymbol{\eta}_t)$. As a result, the above ODE will converge to $\bigtriangledown G(\boldsymbol{\eta}_t) = 0$, which corresponds to (33). This completes the proof.

REFERENCES

[1] L. Lei, Y. Kuang, C. Nan, X. Shen, Z. Zhong, and C. Lin, "Delay-optimal dynamic mode selection and resource allocation in device-to-device communications - part I: optimal policy".

[2] G. Fodor *et al.*, "Design aspects of network assisted device-to-device communications," *IEEE Commun. Mag.*, vol. 50, no. 3, pp. 170-177, March 2012.

[3] L. Lei, Z. Zhong, C. Lin, and X. Shen, "Operator controlled device-to-device communications in LTE-advanced networks," *IEEE Wireless Mag.*, vol. 19, no. 3, pp. 96-104, June 2012.

[4] L. Song, D. Niyato, Z. Han, and E. Hossain, Wireless device-to-device communications and networks. Cambridge University Press, UK, 2015.

[5] H. Nishiyama, M. Ito, and N. Kato, "Relay-by-smartphone: realizing multi-Hop device-to-device communications," *IEEE Wireless Commun.*, vol. 52, no. 4, pp. 56-65, Apr. 2014.

[6] M. L. Puterman, Markov decision processes: discrete stochastic dynamic programming. New York, NY., USA: Wiley, 2005.

[7] D. P. Bertsekas, Dynamic programming and optimal control, 3rd ed. Massachusetts: Athena Scientific, 2007.

[8] Y. Cui, V. K. N. Lau, and R. Wang, "A survey on delay-aware resource control for wireless systems-large deviation theory, stochastic lyapunov drift, and distributed stochastic learning," *IEEE Trans. Info. Theory*, vol. 58, no. 3, pp. 1677-1701, March 2012.

[9] R. Wang and V. K. Lau, "Delay-aware two-hop cooperative relay communications via approximate MDP and stochastic learning," *IEEE Trans. Info. Theory*, vol. 59, no. 11, pp. 7645-7670, Nov. 2013.

[10] M. Moghaddari, E. Hossain, and L. B. Le, "Delay-optimal distributed scheduling in multi-user multi-relay cellular wireless networks," *IEEE Trans. Wireless Commun.*, vol. 61, no. 4, pp. 1349-1360, April 2013.

[11] G. Naddafzadeh-Shirazi, K. Peng-Yong and T. Chen-Khong, "Distributed Reinforcement Learning Frameworks for Cooperative Retransmission in Wireless Networks," *IEEE Trans. Veh. Technol.*, vol. 59, no. 8, pp. 4157-4162, Oct. 2010.

[12] K. Zheng, F. Liu, L. Lei, C. Lin, and Y. Jiang, "Stochastic Performance Analysis of a Wireless Finite-State Markov Channel," *IEEE Trans. Wireless Communications*, vol. 12, no. 2, pp. 782-793, 2013.

[13] L. Georgiadis, M. J. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks", *Foundations and Trends in Networking*, vol. 1, no. 1, pp. 1-144, 2006.

[14] H. J. Kushner and G. G. Yin, Stochastic approximation and optimization of random systems, 2nd ed. New York: Springer, 2003

[15] V. S. Borkar, "Stochastic approximation with two time scales," *Syst. Control Lett.*, vol. 29, pp. 291-294, 1997.

[16] Q. Liu, S. Zhou, and G. B. Giannakis, "Queueing with adaptive modulation and coding over wireless links: cross-layer analysis and design," *IEEE Trans. Wireless Commun.*, vol. 50, no. 3, pp. 1142-1153, May 2005.

[17] V. S. Borkar, "Asynchronous stochastic approximation," *SIAM J. Control Optim.*, vol. 36, pp. 840-851, 1998.

[18] V. S. Borkar, "An actor-critic algorithm for constrained Markov decision processes," *Syst. Control Lett.*, vol. 54, pp. 207-213, 2005.