Throughput-Optimal H-QMW Scheduling for Hybrid Wireless Networks With Persistent and Dynamic Flows

Xiaolong Lan, Yi Chen, and Lin Cai, *Senior Member, IEEE*

***Abstract*— The well-known Queue-length-based MaxWeight scheduling algorithm (QMW) has been proved to be throughput-optimal for persistent flows only, which are long-lived with infinite traffic arrival. If the flows are dynamic ones, i.e., short-lived with finite data to transmit, QMW cannot guarantee queue stability. Given future wireless networks may support both persistent machine-to-machine flows and dynamic human-to-human flows, a Flow (File) Delay based MaxWeight scheduling algorithm (F-D-MW) has been shown to be throughput-optimal. However, new flows have to suffer a long start-up latency after arriving in the system. In this work, we present the definition of the capacity region for hybrid systems with the coexistence of persistent and dynamic flows. First, when a new arrival dynamic flow classification is known, we propose an online Hybrid Queue-length-based MaxWeight (H-QMW) scheduling algorithm, and then propose a more realistic adaptive H-QMW (A-H-QMW) scheduling algorithm for the system without the knowledge of the classification of flows. We prove that H-QMW can achieve throughput-optimality for hybrid systems. Performance evaluation not only validates the throughput-optimality of H-QMW and A-H-QMW in various types of networks but also reveals that H-QMW and A-H-QMW can achieve lower start-up and total latency for dynamic flows than F-D-MW.**

***Index Terms*— Throughput-optimal scheduling, hybrid wireless networks, wireless resource management.**

I. Introduction

A SCHEDULING algorithm is *throughput-optimal* if it can always achieve the network queue stability for any traffic arrival rate vector that lies strictly within the capacity region, in which the queue stability means that the queue size will not accumulate into infinity over time.

Manuscript received March 14, 2019; revised August 2, 2019; accepted October 27, 2019. Date of publication November 12, 2019; date of current version February 11, 2020. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) and Compute Canada. The work of X. Lan was supported in part by the National Natural Science Foundation of China under Grant 61771406 and in part by the Chinese Scholarship Council (CSC). The associate editor coordinating the review of this article and approving it for publication was M. Uysal. *(Corresponding author: Lin Cai.)*

X. Lan is with Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC V8W 3P6, Canada, and also with the School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China (e-mail: xiaolonglan@uvic.ca).

Y. Chen and L. Cai are with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC V8W 3P6, Canada (e-mail: chenyi@uvic.ca; cai@uvic.ca).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TWC.2019.2951564

The pioneer works of Tassiulas and Ephremides [1]–[3] proposed the Queue-length-based MaxWeight scheduling algorithm (QMW), and proved that QMW is a throughput-optimal strategy for networks with persistent flows only. QMW prioritizes the flows with the largest product of the queue length (backlog) and the current transmission rate. It became an active research topic since the scheduling strategy of QMW is simple yet throughput-optimal. Although QMW presents desirable throughput performance, one necessary condition is that the network consists of only a fixed number of *persistent flows* which are long-lived and have continuous data injection. For machine-type applications continuously monitoring the environment, such as in the sensor networks, persistent flows may exist. However, *dynamic flows* are commonly observed for human-to-human communication applications. Dynamic flows have a finite amount of service requests upon arrival in the network, and leave the system once the demanded services are fulfilled. Since flows arrive and leave the system over time, the number of flows in the system may change from one slot to the other. The sample of applications with dynamic flows includes the on-demand video service, email/text messages transfer, web browsing, etc. In networks with dynamic flows, QMW is no longer throughput-optimal [4].

Although there have been a few solutions to design scheduling algorithms for the systems with dynamic flows [5], [6], the majority of the existing work considered the networks with flows of one type only, i.e., either persistent or dynamic flows. However, the coexistence of persistent and dynamic flows cannot be ignored in practice. In future wireless systems, both machine-to-machine and human-to-human applications share the same spectrum. The approach of separating the two types of flows and scheduling them independently is not the best choice, because separating the resources for two types of flows will result in a lower multiplexing gain and efficiency. The existing scheduling algorithm for hybrid systems, Flow (or File) Delay based MaxWeight scheduling (F-D-MW) [6], [25]–[30], can achieve throughput-optimality, but suffer a long start-up latency for dynamic flows. Furthermore, the implementation of delay-based schedulers is difficult in practice. It motivates us to investigate alternative optimal scheduling design for hybrid systems with both persistent and dynamic flows. The contributions of this paper are three-fold.

- First, we present the definition of the capacity region for the hybrid system with the coexistence of persistent and dynamic flows.
- Second, when the classification of new arrival dynamic flows can be known, we design an online Hybrid Queue-length based MaxWeight (H-QMW) scheduling algorithm with channel rate variations in the hybrid networks, which has been proven to achieve throughput-optimality. Moreover, we propose a more realistic adaptive H-QMW (A-H-QMW) scheduling algorithm, in which the system does not need to know the classification of dynamic flow. H-QMW and A-H-QMW are easy-to-implement online algorithms by counting the queue length rather than tracking the latency of each flow.
- Third, through the performance evaluation, we verify that the proposed H-QMW and A-H-QMW scheduling algorithms can stabilize the system with traffic rates within the capacity region, i.e., H-QMW and A-H-QMW are throughput-optimal. Simulation results show that H-QMW and A-H-QMW can achieve the lower start-up and total latency for dynamic flows than F-D-MW. The results also show that the approach of separating the two types of flows and scheduling them independently cannot ensure the stability of hybrid systems. Furthermore, it is shown that the off-line MR scheduling algorithm, throughput-optimal for dynamic flows, is not throughput-optimal when the dynamic flows coexist with persistent flows.

The rest of this paper is organized as follows. Section II introduces the related work. Section III presents the system model, including the definition of the system capacity for hybrid systems. In Section IV, the H-QMW scheduling algorithm for hybrid systems is proposed, and its throughput-optimality is studied. Then a more realistic A-H-QMW scheduling algorithm is presented. Performance evaluation is presented in Section V, followed by the concluding remarks and further research issues in Section VI.

II. Related Work

QMW has been extensively studied in the literature, such as the delay performance [7], energy consumption [8], and fairness [9], etc. The application of QMW has been found in a wide range of research areas such as maximum secure information delivery [10], smart grids [11], and wireless sensor networks [12]. Besides QMW, queue length based throughput-optimal scheduling has other variations [13]–[17]. It has been revealed that all these schedulers are not throughput-optimal if the system consists of dynamic flows [4], [18]. The authors of [4] also proposed a scheduling algorithm to stabilize the systems of dynamic flows, which is an off-line scheduler requiring the knowledge of the channel profile. Subsequent works have developed various scheduling algorithms that were proved to be throughput-optimal for dynamic flows [19], [20]. For example, the MaxRate scheduling algorithm (MR) was studied in [19], which always selects the flows in the system when they are associated with their maximum possible transmission rates. MR is an off-line scheduling algorithm because it needs to know the channel profile to know when the maximum transmission rate is reached. An online alternative was proposed in the same paper with the introduction of a learning period to know what could be the best channel condition. MR has been considered as the benchmark of the throughput-optimal scheduling algorithms for systems with dynamic flows. However, its performance in hybrid systems with the coexistence of persistent and dynamic flows is unknown. Other works focused on the distributed implementation of throughput-optimal schedulers [21]–[24].

Along with the queue length based scheduling algorithms, a F-D-MW [25]–[30] has been shown to be throughput-optimal for the persistent flows as well. F-D-MW gives the priority to the flows (packets) with the largest product of the delay and the current transmission rate in the system. F-D-MW has been investigated widely in the subsequent works. Reference [31] studied the network utility maximization with F-D-MW in wireless systems. Reference [32] developed the delay based back-pressure throughput-optimal scheduler for multihop wireless networks. Considering flow-level dynamics, [6] showed that F-D-MW is also throughput-optimal by applying it to the systems of dynamic flows. Reference [33] revealed that F-D-MW can be applied to the hybrid system with the presence of both persistent and dynamic flows. There are two problems remaining to be explored. First, the channel rate variation is not considered in [33]. Second, the delay performance of F-D-MW is not desirable. By adopting F-D-MW, new flows in the system may suffer a long start-up latency after arriving in the system, which is not compatible with the existing transport layer protocols such as TCP [34] and not desirable for many dynamic-flow applications.

Different from the existing work, we investigate an on-line H-QMW scheduling algorithm based on the current queue length and transmission rate. We give the analysis of the throughput-optimality of the proposed H-QMW algorithm with channel rate variations in hybrid networks that have both persistent and dynamic flows.

III. System Model

We consider a downlink wireless system with two types of flows, persistent flows and dynamic flows, which share the channel resources. For each type of flow, there can be multiple classes of flows, which makes the system a heterogeneous one. Within each class, the flows have independent and identically distributed (i.i.d.) traffic arrivals and the i.i.d channel, i.e., with the same channel rate distribution. Denote by M and K the number of classes of persistent and dynamic flows, respectively.

A. Arrival Model

As shown in Fig. 1, each persistent flow in the system is long-lived and has continuous traffic arrival, so it has an infinite amount of data to transmit when $t \to \infty$. Each class of dynamic flows has continuous flow arrivals, while each dynamic flow has a finite amount of data to transmit upon its arrival in the system, and leaves the system once its buffer is empty. We focus on the flows that are backlogged in the system which have buffered data to transmit. Let $Q_{ij}^{(p)}(t)$ and $Q_{ij}^{(d)}(t)$

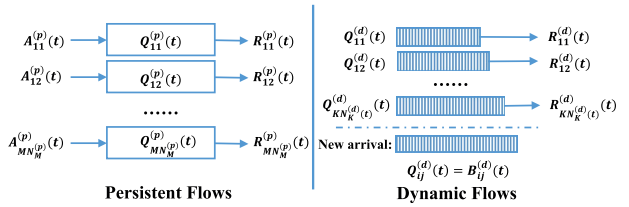


Fig. 1. Hybrid systems with the coexistence of persistent and dynamic flows.

denote the residual queued bits waiting for transmission of the j-th *persistent* flow of class-i and the j-th *dynamic* flow of class-i at the beginning of time slot t, respectively. All the persistent flows arrive during time slot $t = 0$ and never leave the system. For dynamic flows, since we have the departure of old flows and the arrival of new flows, the j-th dynamic flow of class-i may change from one time slot to the other. With each class of the existing dynamic flows, the flow index j is determined according to its arrival time. Let $\mathcal{N}^{(p)}$ and $\mathcal{N}^{(d)}(t)$ denote the set of persistent flows and dynamic flows at time slot t, respectively. Let $N^{(p)} = |\mathcal{N}^{(p)}| = \sum_{i=1}^{M} N_i^{(p)}$ denote the number of persistent flows in the system, where $N_i^{(p)}$ is the number of persistent flows of class-i. The total number of dynamic flows at the beginning of time slot t is $N^{(d)}(t) = |\mathcal{N}^{(d)}(t)| = \sum_{i=1}^{K} N_i^{(d)}(t)$, where $N_i^{(d)}(t)$ is the number of class-i dynamic flows that are backlogged in the system at time slot t.

For the j-th persistent flow of class-i, the amount of arrival data in one slot is denoted by $A_{ij}^{(p)}(t)$ with the mean of $\lambda_i^{(p)} = \mathbb{E}[A_{ij}^{(p)}(t)]$. Let $\alpha_i^{(p)}$ denote the number of persistent flows of class-i in the system, and thus $\lambda^{(p)} = \sum_{i=1}^{M} \alpha_i^{(p)} \lambda_i^{(p)}$ is the average amount of arrival data from all of the persistent flows in one slot.

For dynamic flows, let $A_i^{(d)}(t) \in \{0\} \cup \mathbb{Z}^+$ denote the number of class-i dynamic flows arriving during time slot t, which is a random variable with the mean of $\alpha_i^{(d)} = \mathbb{E}[A_i^{(d)}(t)]$. The initial flow size for the j-th dynamic flow of class-i is denoted as $B_{ij}^{(d)}(t)$. For class-i dynamic flows, we assume that $B_{ij}^{(d)}(t)$ is a random variable and has a finite mean $\beta_i^{(d)} = \mathbb{E}[B_{ij}^{(d)}(t)]$. Thus, the average new data amount of class-i dynamic flows is $\lambda_i^{(d)} = \alpha_i^{(d)} \beta_i^{(d)}$, and the average amount of arrival data from all of the dynamic flows is $\lambda^{(d)} = \sum_{i=1}^{K} \alpha_i^{(d)} \beta_i^{(d)}$.

B. Channel Model

Let $R_{ij}^{(\cdot)}(t)$ denote the transmission rate of the wireless channel at time t between the j-th flow of class-i and the base station (BS). The unit of the channel rate is bit/slot. $R_{ij}^{(\cdot)}(t)$ may vary over time as a result of wireless channel fading and shadowing. For class-i flows, we assume that $R_{ij}^{(\cdot)}(t)$ is i.i.d. with finite supports, i.e., $R_{ij}^{(\cdot)}(t) \in \boldsymbol{\mathcal{R}_i}^{(\cdot)} = \{0, \mathcal{R}_{i1}^{(\cdot)}, \mathcal{R}_{i2}^{(\cdot)}, \ldots\}$, and the corresponding probability is $\{P_{i0}^{(\cdot)}, P_{i1}^{(\cdot)}, P_{i2}^{(\cdot)}, \ldots\}$ satisfying $\sum_k P_{ik}^{(\cdot)} = 1$. $\boldsymbol{\mathcal{R}_i}^{(\cdot)}$ is the set of channel rate options for class-i flows. In particular, $R_{ij}^{(\cdot)}(t) = 0$ indicates that the current signal-to-noise ratio of the j-th flow in class-i is below a certain threshold and the transmission is considered as failure, i.e., the lowest-order modulation and coding scheme cannot be supported. Different classes may have heterogeneous channel condition distributions. The maximum possible transmission rate of class-i flows is defined as $R_i^{(\cdot)\max} = \max\{0, \mathcal{R}_{i1}^{(\cdot)}, \mathcal{R}_{i2}^{(\cdot)}, \ldots\}$. In addition, in this paper, we assume that the duration of each time slot is small enough such that no resources are wasted in any time slot.

C. System Capacity Region

The capacity region of the system with a fixed number of persistent flows can be found in [35], which is different from the one with dynamic flows only [4], and thus before the transmission scheduling algorithm is investigated, the capacity region of the hybrid system with both persistent and dynamic flows should be addressed. The capacity region can be defined in terms of the traffic intensity ρ, which measures the average occupancy of the shared channel resource. In other words, ρ is the average number of time slots that are required to transmit the arrival traffic in one slot when the most efficient transmission strategy is adopted. The necessary condition for stability to be achievable is $\rho < 1$ [4]. Thus, if the average amount of arrival traffic in one slot can be transmitted is less than one time slot by the maximum possible transmission rate, there exists at least one scheduling algorithm to achieve system stability. If $\rho > 1$, on average more than one slot is required to transmit the amount of arrival data in one slot, and the residual data will accumulate into infinity over time which results in instability. From this perspective, the system capacity region is defined as $\rho < 1$, and any arrival rate vector in the capacity region should be stably transmitted by throughput-optimal algorithms.

For the hybrid system, $\rho = \rho^{(p)} + \rho^{(d)}$, where $\rho^{(p)}$ and $\rho^{(d)}$ are the traffic intensities of persistent and dynamic flows, respectively. We discuss $\rho^{(p)}$ first. Without loss of generality, we assume that $\mathcal{R}_{i1}^{(p)} > \mathcal{R}_{i2}^{(p)} > \ldots > \mathcal{R}_{i|\boldsymbol{\mathcal{R}_i}^{(p)}|}^{(p)}$. Given a fixed probability of each channel state, if a persistent flow is scheduled only on the maximum channel rate, the queuing system may not be stabilized. For instance, there is a persistent flow with an average arrival rate of 0.7[1]. Its channel has two states with the rates of $\{1, 0.5\}$, and the probability of each state is 0.5. Thus, if we schedule the persistent flow only when the channel rate is 1, the overall service rate of this flow is 0.5 which is lower than the arrival rate. To ensure the queue stability, the flow needs to be scheduled when its channel rate is 0.5 sometimes. Therefore, the most efficient transmission scheme is that the system schedules a flow to transmit data on its n best channel states, where n is the minimum integer to support the requirement of the arrival traffic as shown in (2). Thus, the traffic intensity for the persistent flow of class-i, i.e., the minimum average number of time slots allocated to the persistent flow of class-i to achieve queue stability, can be calculated as follows:

$$\rho_i^{(p)} = \sum_{k=1}^{n-1} P_{ik}^{(p)} + \frac{\alpha_i^{(p)} \lambda_i^{(p)} - \sum_{m=1}^{n-1} P_{im}^{(p)} \mathcal{R}_{im}^{(p)}}{\mathcal{R}_{in}^{(p)}}, \qquad (1)$$

[1]We normalize the unit of arrival and service rates and then omit their units for presentation simplicity.

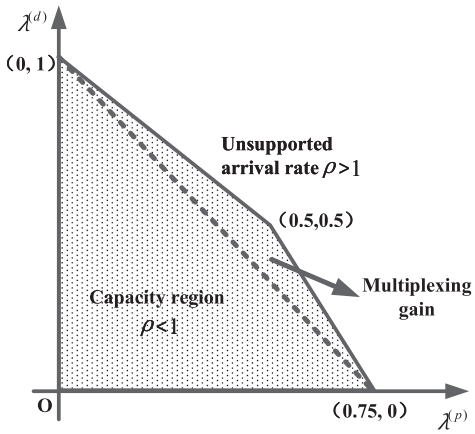


Fig. 2. Capacity region for hybrid systems when the channel rate set is $\boldsymbol{R} = \{1, 0.5\}$ with probability $\boldsymbol{P} = \{0.5, 0.5\}$.

where n satisfying

$$\sum_{k=1}^{n-1} P_{ik}^{(p)} \mathcal{R}_{ik}^{(p)} < \alpha_i^{(p)} \lambda_i^{(p)} \text{ and } \sum_{k=1}^{n} P_{ik}^{(p)} \mathcal{R}_{ik}^{(p)} \geq \alpha_i^{(p)} \lambda_i^{(p)}. \quad (2)$$

For each dynamic flow, since it has a finite amount of data to transmit, it can always wait for the channel to be in its best state to transfer data until its buffer is empty and it leaves the system. Thus, the traffic intensity of class-i dynamic flows can be given by

$$\rho_i^{(d)} = \mathbb{E}\left[\frac{\sum_{j=1}^{A_i^{(d)}(t)} B_{ij}^{(d)}(t)}{R_i^{(d)\max}}\right] = \frac{\alpha_i^{(d)} \beta_i^{(d)}}{R_i^{(d)\max}}. \quad (3)$$

The traffic intensity of the dynamic flows is $\rho^{(d)} = \sum_{i=1}^{K} \rho_i^{(d)}$.

In Fig. 2, we show the capacity region for hybrid systems with the coexistence of persistent and dynamic flows. For this figure, we assume that the channel rates of all flows have two states. The channel rate set is $\boldsymbol{R} = \{1, 0.5\}$, and the corresponding probability is set to $\boldsymbol{P} = \{0.5, 0.5\}$. The x-axis and y-axis represent the arrival rate of a persistent flow and sum of the dynamic flows, respectively. The capacity region in Fig. 2 is drawn according to the traffic intensity presented in (1) and (3). The shaded part in Fig. 2 shows the arrival rate region that the system can support, i.e., $\rho < 1$. In particular, the boundary surface of the capacity region is the black solid line, i.e., $\rho = 1$. Besides, the dashed line represents the maximum arrival rate that can be supported by the approach of separating two types of flows and scheduling them independently. Since separating the resource for two types of flows leads to a lower multiplexing gain, so it cannot achieve the boundary surface of the capacity region. Furthermore, from this figure, we can find that the maximum supported arrival rate of dynamic flows is larger than that of persistent flows. This can be interpreted as follows: i) For each dynamic flow, since the amount of data to be transmitted is finite, it can always wait for being scheduled only when its channel rate achieves its maximum value. ii) Given a sufficient number of dynamic flows, the scheduler has the flexibility to choose the flow in its best channel state.

IV. Hybrid Queue-Length Based MaxWeight Scheduling

In this section, we first propose a Hybrid Queue-length based MaxWeight scheduling algorithm (H-QMW) for hybrid systems with persistent and dynamic flows, and give the analysis of throughput-optimality. Then, we propose a more realistic adaptive H-QMW (A-H-QMW) scheduling algorithm for the system without the knowledge of the classification of dynamic flows.

A. Minimum Throughput Requirement

We assume that the BS selects a flow to transmit using a resource block. Without loss of generality, we consider a time slot as one resource block. The approach can be applied in other systems where resource blocks are orthogonal in the time, frequency, and/or code domain.

Let $d_{ij}^{(\cdot)}(t)$ denote the indicator function that indicates whether the j-th flow of class-i is scheduled to transmit during slot t, which is given by

$$d_{ij}^{(\cdot)}(t) = \begin{cases} 1, & \text{if the } j\text{-th flow of class-}i \text{ is scheduled} \\ & \text{at time slot } t, \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

Let $\mathbf{d}(t) = \left\{d_{ij}^{(\cdot)}(t)\right\}$ denote the transmission scheduling policy of the BS at time slot t. To ensure the stability of all persistent and dynamic flows, the transmission scheduling policy must satisfy the following minimum throughput constraints for each flow

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} d_{ij}^{(p)}(t) R_{ij}^{(p)}(t) \geq \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} A_{ij}^{(p)}(t), \quad \forall i, j, \quad (5)$$

$$\sum_{t=t_{ij}^{(d)}+1}^{t_{ij}^{(d)}+\delta_t} d_{ij}^{(d)}(t) R_{ij}^{(d)}(t) \geq B_{ij}^{(d)}(t_{ij}^{(d)}), \forall i, j, \quad (6)$$

where $0 < \delta_t < \infty$ and $t_{ij}^{(d)}$ is the initial time that the j-th of class-i dynamic flow is added to the system. Eq. (5) indicates that for each persistent flow, the average transmission rate is no less than the average arrival traffic rate. Eq. (6) shows that for each dynamic flow, data transmission should be completed within a finite time δ_t.

In addition, at any given time slot t, since the BS can select at most one flow for transmission, the transmission scheduling policy $\mathbf{d}(t)$ satisfies the following interference constraint:

$$\sum_{i=1}^{M} \sum_{j=1}^{N_i^{(p)}} d_{ij}^{(p)}(t) + \sum_{i=1}^{K} \sum_{j\in\mathcal{N}_i^{(d)}(t)} d_{ij}^{(d)}(t) \leq 1. \quad (7)$$

B. Problem Formulation and H-QMW Scheduling Policy

In this subsection, we assume that the system can identify the class of each flow when it arrives. Our objective is to ensure the stability of all data queues by designing an optimal scheduling policy, subject to the interference constraint.

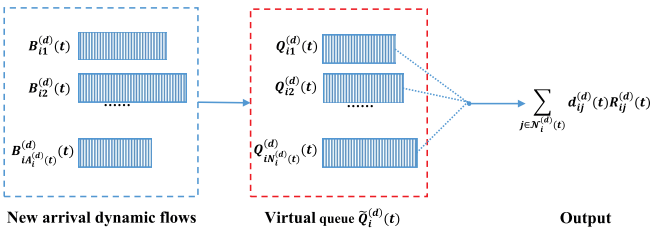


Fig. 3. Virtual queue of the class-i dynamic flow.

Since each persistent flow is long-lived and has continuous traffic arrival, and each dynamic flow has a finite amount of data to transmit upon its arrival in the system, the data queue evolution of persistent and dynamic flows can be expressed as

$$Q_{ij}^{(p)}(t+1) = \left(Q_{ij}^{(p)}(t) - d_{ij}^{(p)}(t)R_{ij}^{(p)}(t)\right)^+ + A_{ij}^{(p)}(t), \quad \forall i,j, \tag{8}$$

$$Q_{ij}^{(d)}(t+1) = \left(Q_{ij}^{(d)}(t) - d_{ij}^{(d)}(t)R_{ij}^{(d)}(t)\right)^+, \quad \forall i,j \in \mathcal{N}_i^{(d)}(t), \tag{9}$$

respectively, where $(\cdot)^+ = \max\{\cdot, 0\}$.

For the system with persistent flows only, it has continuous traffic arrival and data departure. We can use the Lyapunov optimization framework to design an efficient scheduling policy that can effectively ensure the stability of the system [35]. However, it is not suitable for systems with hybrid flows. To incorporate our analysis in the Lyapunov optimization framework, we define $\tilde{Q}_i^{(d)}(t)$ as a virtual data queue of class-i dynamic flows, which is given by

$$\begin{aligned}\tilde{Q}_i^{(d)}(t+1) &= \left(\tilde{Q}_i^{(d)}(t) - \sum_{j\in\mathcal{N}_i^{(d)}(t)} \min\left\{d_{ij}^{(d)}(t)R_{ij}^{(d)}(t), Q_{ij}^{(d)}(t)\right\}\right)^+ \\ &\quad + \sum_{j=1}^{A_i^{(d)}(t)} B_{ij}^{(d)}(t), \forall i,t,\end{aligned} \tag{10}$$

where $\tilde{Q}_i^{(d)}(t) = \sum_{j\in\mathcal{N}_i^{(d)}(t)} Q_{ij}^{(d)}(t)$. The size of $\tilde{Q}_i^{(d)}(t)$ reflects the total amount of data backlogged for class-i dynamic flows. Since the duration of each time slot is small enough such that $d_{ij}^{(d)}(t)R_{ij}^{(d)}(t) = \min\left\{d_{ij}^{(d)}(t)R_{ij}^{(d)}(t), Q_{ij}^{(d)}(t)\right\}$, i.e., no resources are wasted in any time slot. Therefore, the evolution of virtual queue can be rewritten as

$$\begin{aligned}\tilde{Q}_i^{(d)}(t+1) &= \left(\tilde{Q}_i^{(d)}(t) - \sum_{j\in\mathcal{N}_i^{(d)}(t)} d_{ij}^{(d)}(t)R_{ij}^{(d)}(t)\right)^+ \\ &\quad + \sum_{j=1}^{A_i^{(d)}(t)} B_{ij}^{(d)}(t), \quad \forall\, i,t.\end{aligned} \tag{11}$$

Fig. 3 presents the queue evolution of virtual queue for class-i dynamic flows. As shown in Fig. 3, the virtual queue of class-i dynamic flows, $\tilde{Q}_i^{(d)}(t)$, can be regarded as a queue for a persistent flow with continuous traffic arrival, in which $\sum_{j=1}^{A_i^{(d)}(t)} B_{ij}(t)$ can be considered as continuous data arrival rate of $\tilde{Q}_i^{(d)}(t)$, and $\sum_{j\in\mathcal{N}_i^{(d)}(t)} d_{ij}^{(d)}(t)R_{ij}^{(d)}(t)$ can be treated as the output of $\tilde{Q}_i^{(d)}(t)$. Therefore, based on these virtual queues, the systems with hybrid flows can be approximated as systems with virtual-persistent flows. Next, we will design an efficient scheduling policy for systems with hybrid flows using the Lyapunov framework.

Lemma 1: If all the persistent flow queues and virtual queues are rate stable, i.e.,

$$\lim_{T\to\infty} \frac{Q_{ij}^{(p)}(T)}{T} = \lim_{T\to\infty} \frac{\tilde{Q}_i^{(d)}(T)}{T} = 0, \forall i,j, \tag{12}$$

then the minimum throughput constraints can be satisified.

Proof: Based on the queue evolution in (8) and (11), we have

$$Q_{ij}^{(p)}(t+1) \geq Q_{ij}^{(p)}(t) - d_{ij}^{(p)}(t)R_{ij}^{(p)}(t) + A_{ij}^{(p)}(t), \quad \forall i,j,t, \tag{13}$$

$$\tilde{Q}_i^{(d)}(t+1) \geq \tilde{Q}_i^{(d)}(t) - \sum_{j\in\mathcal{N}_i^{(d)}(t)} d_{ij}^{(d)}(t)R_{ij}^{(d)}(t) + \sum_{j=1}^{A_i^{(d)}(t)} B_{ij}^{(d)}(t). \tag{14}$$

By summing the above equations over T time slots, dividing it by T and taking $\lim_{T\to\infty}$ on both sides, we have

$$\lim_{T\to\infty} \frac{Q_{ij}^{(p)}(T) - Q_{ij}^{(p)}(0)}{T} \geq \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1}\left\{A_{ij}^{(p)}(t) - d_{ij}^{(p)}(t)R_{ij}^{(p)}(t)\right\}, \tag{15}$$

$$\lim_{T\to\infty} \frac{\tilde{Q}_i^{(d)}(T) - \tilde{Q}_i^{(d)}(0)}{T} \geq \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1}\left\{\sum_{j=1}^{A_i^{(d)}(t)} B_{ij}^{(d)}(t) - \sum_{j\in\mathcal{N}_i^{(d)}(t)} d_{ij}^{(d)}(t)R_{ij}^{(d)}(t)\right\}. \tag{16}$$

Since $Q_{ij}^{(p)}(t)$ and $\tilde{Q}_i^{(d)}(t)$ are rate stable and the initial states are zero, we can obtain

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} A_{ij}^{(p)}(t) \leq \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} d_{ij}^{(p)}(t)R_{ij}^{(p)}(t), \tag{17}$$

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1}\sum_{j=1}^{A_i^{(d)}(t)} B_{ij}^{(d)}(t) \leq \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1}\sum_{j\in\mathcal{N}_i^{(d)}(t)} d_{ij}^{(d)}(t)R_{ij}^{(d)}(t). \tag{18}$$

Eq. (18) indicates that all the dynamic flows of class-i can be sent out if $\tilde{Q}_i^{(d)}(t)$ is rate stable, i.e., (6) can be satisfied. ■

Lemma 1 indicates that for systems with hybrid flows, if a scheduling policy can ensure that all the persistent flow queues and virtual queues are stable, this scheduling policy can satisfy the minimum throughput constraints and is throughput-optimal.

Based on the actual data queues of persistent flows and virtual data queues of dynamic flows, we define the quadratic

Lyapunov function as

$$L(\mathbf{\Theta}(t)) = \frac{1}{2}\Big\{\sum_{i=1}^{M}\sum_{j=1}^{N_i^{(p)}} Q_{ij}^{(p)}(t)^2 + \sum_{k=1}^{K}\tilde{Q}_k^{(d)}(t)^2\Big\}, \quad (19)$$

where $\mathbf{\Theta}(t) = [Q_{ij}^{(p)}(t), \tilde{Q}_k^{(d)}(t)]$ denotes the concatenated vector of all queues at the beginning of time slot t. The value of $L(\mathbf{\Theta}(t))$ measures the current data queue length of all flows, which is defined to grow larger with the queue system towards undesired states. $L(\mathbf{\Theta}(t))$ grows larger as data queues increase. The Lyapunov drift is defined to evaluate the expected change in the Lyapunov function between two consecutive time slots, which is given by

$$\Delta(\mathbf{\Theta}(t)) = \mathbb{E}\{L(\mathbf{\Theta}(t+1)) - L(\mathbf{\Theta}(t))|\mathbf{\Theta}(t)\}, \quad (20)$$

where expectation is taken over the randomness of channel rate and the scheduling control decision given the current queue vector $\mathbf{\Theta}(t)$ at time slot t. To ensure that the stability of $Q_{ij}^{(p)}(t)$ and $\tilde{Q}_i^{(d)}(t)$ can be achieved, we should try to minimize the above Lyapunov drift.

Lemma 2: The Lyapunov drift can be upper bounded by

$$\begin{aligned}&\Delta(\mathbf{\Theta}(t))\\ &\le B + \sum_{i=1}^{M}\sum_{j=1}^{N_i^{(p)}} Q_{ij}^{(p)}(t)\mathbb{E}\Big\{A_{ij}^{(p)}(t) - d_{ij}^{(p)}(t)R_{ij}^{(p)}(t)\Big|\mathbf{\Theta}(t)\Big\}\\ &\quad + \sum_{k=1}^{K} Q_k^{(d)}(t)\mathbb{E}\Bigg\{\sum_{l=1}^{A_k^{(d)}(t)} B_{kl}^{(d)}(t) - \sum_{l\in\mathcal{N}_k^{(d)}(t)} d_{kl}^{(d)}(t)R_{kl}^{(d)}(t)\Big|\mathbf{\Theta}(t)\Bigg\},\end{aligned} \quad (21)$$

where $B = \frac{1}{2}\Big(\sum_{i=1}^{M}\big(R_i^{(p)\max 2} + N_i^{(p)}A_i^{(p)\max 2}\big) + \sum_{k=1}^{K}\big(R_k^{(d)\max 2} + (A_k^{(d)\max}B_k^{(d)\max})^2\big)\Big)$, $A_i^{(p)\max}$ *is the maximal arrival rate of class-*i *persistent flow,* $A_k^{(d)\max}$ *is the maximal number of class-*k *dynamic flows arriving during each slot, and* $B_k^{(d)\max}$ *is the maximum initial flow size for class-*k *dynamic flow.*

Proof: Based on the queue evolution of persistent flows, we have

$$\begin{aligned}&Q_{ij}^{(p)}(t+1)^2\\ &= \Bigg(\Big(Q_{ij}^{(p)}(t) - d_{ij}^{(p)}(t)R_{ij}^{(p)}(t)\Big)^+ + A_{ij}^{(p)}(t)\Bigg)^2\\ &= Q_{ij}^{(p)}(t)^2 + \Big(d_{ij}^{(p)}(t)R_{ij}^{(p)}(t)\Big)^2 - 2Q_{ij}^{(p)}(t)d_{ij}^{(p)}(t)R_{ij}^{(p)}(t)\\ &\quad + A_{ij}^{(p)}(t)^2 + 2A_{ij}^{(p)}(t)\Big(Q_{ij}^{(p)}(t) - d_{ij}^{(p)}(t)R_{ij}^{(p)}(t)\Big)^+\\ &\overset{(a)}{\le} Q_{ij}^{(p)}(t)^2 + d_{ij}^{(p)}(t)R_i^{(p)\max 2} + A_i^{(p)\max 2}\\ &\quad + 2Q_{ij}^{(p)}(t)\Big(A_{ij}^{(p)}(t) - d_{ij}^{(p)}(t)R_{ij}^{(p)}\Big),\end{aligned} \quad (22)$$

where step (a) applies the fact that $d_{ij}^{(p)}(t)$ is a binary variable, i.e., $d_{ij}^{(p)}(t) = d_{ij}^{(p)}(t)^2$. Similarly, for the virtual queue evolution of dynamic flows, we have

$$\begin{aligned}&\tilde{Q}_i^{(d)}(t+1)^2\\ &\le \tilde{Q}_i^{(d)}(t)^2 + d_{ij}^{(p)}(t)R_k^{(d)\max 2} + \big(A_k^{(d)\max}B_k^{(d)\max}\big)^2\\ &\quad + 2\tilde{Q}_i^{(d)}(t)\Bigg(\sum_{j=1}^{A_i^{(d)}(t)} B_{ij}^{(d)}(t) - \sum_{j\in\mathcal{N}_i^{(d)}(t)} d_{ij}^{(d)}(t)R_{ij}^{(d)}(t)\Bigg).\end{aligned} \quad (23)$$

Since the BS can select at most one flow for transmission at each time slot, by substituting (22) and (23) into (20), we can conclude Lemma 2. ■

Lemma 2 provides the upper bound of the Lyapunov drift, and we solve the following optimization problem instead of directly minimizing the Lyapunov drift item. At each time slot, given the current data queue length $\mathbf{\Theta}(t)$, we make decisions on transmission scheduling policy by solving the following optimization problem:

$$\begin{aligned}\min_{\mathbf{d}(t)}:\ &-\sum_{i=1}^{M}\sum_{j=1}^{N_i^{(p)}} Q_{ij}^{(p)}(t)R_{ij}^{(p)}(t)d_{ij}^{(p)}(t)\\ &-\sum_{k=1}^{K}\sum_{l\in\mathcal{N}_l^{(d)}(t)} \tilde{Q}_k^{(d)}(t)R_{kl}^{(d)}(t)d_{kl}^{(d)}(t)\\ s.t.,\ &d_{ij}^{(p)}(t), d_{kl}^{(d)}(t) \in \{0,1\},\quad \forall i,j,k,l,t,\\ &\sum_{i=1}^{M}\sum_{j=1}^{N_i^{(p)}} d_{ij}^{(p)}(t) + \sum_{k=1}^{K}\sum_{l\in\mathcal{N}_i^{(d)}(t)} d_{kl}^{(d)}(t) \le 1\ \forall t.\end{aligned} \quad (24)$$

We may notice that the optimization problem $d_{ij}^{(\cdot)}(t)$ is a binary variable and at most one flow will be scheduled to transmit. Thus, it is easy to solve the above optimization problem and the scheduling rule of H-QMW is given in the following Algorithm.

H-QMW scheduling algorithm. *Given the current data queue length* $Q_{ij}^{(\cdot)}(t)$ *and transmission rate* $R_{ij}^{(\cdot)}(t)$*, the Hybrid Queue-length-based MaxWeight scheduling algorithm (H-QMW) is given by*

$$d_{ij}^{(p)}(t) = \begin{cases}1, & \text{if } W^{(p)} \ge W^{(d)} \wedge Q_{ij}^{(p)}(t)R_{ij}^{(p)}(t) = W^{(p)},\\ 0, & \text{otherwise,}\end{cases}$$

$$d_{ij}^{(d)}(t) = \begin{cases}1, & \text{if } W^{(p)} \le W^{(d)} \wedge \tilde{Q}_i^{(d)}(t)R_{ij}^{(d)}(t) = W^{(d)},\\ 0, & \text{otherwise,}\end{cases}$$

where $W^{(p)} = \max_{i,j}\{Q_{ij}^{(p)}(t)R_{ij}^{(p)}(t)\}$ *and* $W^{(d)} = \max_{i,j}\{\tilde{Q}_i^{(d)}(t)R_{ij}^{(d)}(t)\}$*. The scheduler applies uniform tie-breaking, if there are more than one flow satisfying the condition. The scheduling decision is made at the beginning of each time slot independently.*

The "*hybrid queue*" represents the actual queue of persistent flows and the virtual queue of dynamic flows. Based on the scheduling rule of H-QMW algorithm, we notice that, similar to the traditional QMW scheme, H-QMW prioritizes a flow with the maximum product of the backlog and the current transmission rate. However, the difference is that for dynamic

flows of class-i, they always share the same data backlog. In other words, the system determines whether $Q_{ij}^{(d)}(t)$ is scheduled depending on the total amount of backlog of class-i dynamic flow, rather than the queue length of each individual dynamic flow $Q_{ij}^{(d)}(t)$. If $\tilde{Q}_i^{(d)}(t)$ tends to be unstable, the system prefers to selecting one of class-i dynamic flows for transmission. Moreover, we observe that if $Q_{ij}^{(d)}(t)$ is scheduled, then it must have the maximum transmission rate among the class-i dynamic flows. In particular, we can find that the H-QMW scheduling algorithm will degenerate to the traditional QMW scheme when the system only has persistent flows, and it acts similar to the Maximum Rate scheduler if the system only has dynamic flows. In addition, to ensure the fairness of each dynamic flow in the same class, we can classify those dynamic flows that have the same channel state probability distribution into the same class. Furthermore, our proposed scheduling algorithm can be applied to general scenarios for orthogonal resource allocation, where the orthogonality can be in the time, frequency, or code domains. We do not need to restrict the scheduler to schedule one flow at a time. Instead, the scheduler can allocate each orthogonal resource block to each flow within the scheduling period. This is inline with the current cellular system. For instance, if the system has N orthogonal resource blocks in the current time slot, the scheduler only needs to perform N times of the proposed H-QMW scheduling algorithm. When the scheduler decides to allocate the n-th resource block to a flow, the corresponding data queue needs to be updated. Then, this flow will continue to participate in the $(n+1)$-th resource block allocation with the new data queue size. Thus, multiple resource blocks can be allocated to the same flow. After the system executes the H-QMW scheduling algorithm N times, the system starts to perform the actual transmission according to the obtained scheduling strategy.

C. Stability Analysis

In this subsection, we analyze the performance of the proposed H-QMW policy and prove that the proposed scheduling algorithm is throughput-optimal.

Lemma 3: For any traffic intensity is strictly within the capacity region, i.e., $\rho < 1$, the proposed H-QMW scheduling algorithm can ensure the stability of all persistent flows and dynamic flows, namely H-QMW scheduling algorithm is throughput-optimal.

Proof: For $\rho < 1$, there exists a stationary scheduling policy $\boldsymbol{d^*(t)}$ and $\epsilon > 0$ with the following features [33], [35]

$$\mathbb{E}\left\{ A_{ij}^{(p)}(t) - d_{ij}^{(p)^*}(t) R_{ij}^{(p)}(t) \middle| \boldsymbol{\Theta(t)} \right\} = \mathbb{E}\left\{ \lambda_i^{(p)} - d_{ij}^{(p)^*}(t) R_{ij}^{(p)}(t) \right\} \leq -\epsilon, \quad \forall i, j, \tag{25}$$

$$\mathbb{E}\left\{ \sum_{l=1}^{A_k^{(d)}(t)} B_{kl}^{(d)}(t) - \sum_{l \in \mathcal{N}_k^{(d)}(t)} d_{kl}^{(d)^*}(t) R_{kl}^{(d)}(t) \middle| \boldsymbol{\Theta(t)} \right\} = \mathbb{E}\left\{ \lambda_k^{(d)} - \sum_{l \in \mathcal{N}_k^{(d)}(t)} d_{kl}^{(d)^*}(t) R_{kl}^{(d)}(t) \right\} \leq -\epsilon, \quad \forall k. \tag{26}$$

Since our scheduling policy minimizes the right hand side of (21). Therefore, we can upper bound the right hand side of (21) by the stationary scheduling policy $\boldsymbol{d^*(t)}$. By substituting (25) and (26) into (21), we have

$$\Delta(\boldsymbol{\Theta}(t)) \leq B - \sum_{i=1}^{M} \sum_{j=1}^{N_i^{(p)}} Q_{ij}^{(p)}(t)\epsilon - \sum_{k=1}^{K} \tilde{Q}_k^{(d)}(t)\epsilon. \tag{27}$$

According to the law of iterated expectation, taking expectations of both side for (27), we can obtain

$$\mathbb{E}\{L(\boldsymbol{\Theta}(t+1))\} - \mathbb{E}\{L(\boldsymbol{\Theta}(t))\} \leq B - \sum_{i=1}^{M} \sum_{j=1}^{N_i^{(p)}} \mathbb{E}\left\{ Q_{ij}^{(p)}(t) \right\}\epsilon - \sum_{k=1}^{K} \mathbb{E}\left\{ \tilde{Q}_k^{(d)}(t) \right\}\epsilon. \tag{28}$$

Summing (28) over T time slots, dividing it by T, we have

$$\frac{\mathbb{E}\{L(\boldsymbol{\Theta}(T))\} - \mathbb{E}\{L(\boldsymbol{\Theta}(0))\}}{T} \leq B - \frac{1}{T} \sum_{t=0}^{T-1} \left\{ \sum_{i=1}^{M} \sum_{j=1}^{N_i^{(p)}} \mathbb{E}\left\{ Q_{ij}^{(p)}(t) \right\} + \sum_{k=1}^{K} \mathbb{E}\left\{ \tilde{Q}_k^{(d)}(t) \right\} \right\}\epsilon. \tag{29}$$

Note that $L(\boldsymbol{\Theta}(t)) \geq 0$ and $L(\boldsymbol{\Theta}(0)) = 0$. Taking $\lim_{T\to\infty}$ for (29), we have

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \left\{ \sum_{i=1}^{M} \sum_{j=1}^{N_i^{(p)}} \mathbb{E}\{ Q_{ij}^{(p)}(t) \} + \sum_{k=1}^{K} \mathbb{E}\{ \tilde{Q}_k^{(d)}(t) \} \right\} \leq \frac{B}{\epsilon}, \tag{30}$$

which indicates that the persistent flows and dynamic flows can be stabilized by H-QMW. ■

Eq. (30) indicates that H-QMW can ensure all the queues are strongly stable, which means that all the persistent queues and virtual queues of dynamic flows are also rate stable. Eq. (30) combined with Lemma 1 indicates that the proposed H-QMW scheduling algorithm is throughput-optimal for hybrid systems with the coexistence of persistent and dynamic flows, which can effectively ensure the network queue stability with any traffic arrival rate in the capacity region.

D. Adaptive H-QMW Scheduling Design

In realistic application scenarios, some flow identification and classification techniques can be used to identify persistent and dynamic flows, such as Naive Bayes estimator [36] and machine learning [37]. For the persistent flows which are long-lived and have continuous data injection, they typically have fixed ports and IP addresses, and thus we can use pattern recognition to extract features and construct an appropriate model to identify persistent flows. However, dynamic flows have a finite amount of services and leave the system once the demanded services are fulfilled, which make it difficult to identify the category of a dynamic flow once it is added in the system. Thus, in this subsection, we propose a more realistic adaptive H-QMW (A-H-QMW), in which the BS does not need to know the classification of each dynamic flow. In the proposed A-H-QMW scheduling, we regard all the dynamic

flows as a virtual dynamic class. Let $\tilde{Q}^{(d)}$ denote the queue backlog of the virtual dynamic class, which is given by

$$\tilde{Q}^{(d)}(t) = \sum_{i=1}^{K} \sum_{j \in \mathcal{N}_i^{(d)}(t)} Q_{ij}^{(d)}(t), \ \forall t, \tag{31}$$

and the corresponding queue evolution of the virtual dynamic class is given by

$$\begin{aligned} &\tilde{Q}^{(d)}(t+1) \\ &\quad = \Big(\tilde{Q}^{(d)}(t+1) - \sum_{i=1}^{K} \sum_{j \in \mathcal{N}_i^{(d)}(t)} d_{ij}^{(d)}(t) R_{ij}^{(d)}(t)\Big)^{+} \\ &\qquad + \sum_{i=1}^{K} \sum_{j=1}^{A_i^{(d)}(t)} B_{ij}^{(d)}(t), \quad \forall t. \end{aligned} \tag{32}$$

Intuitively, if a dynamic flow is to be scheduled for transmission, it should have the maximum transmission rate in that type of dynamic flows. However, unlike H-QMW scheduling, all the dynamic flows are considered as a virtual class in the proposed A-H-QMW scheduling. If using H-QMW scheduling directly, dynamic flows of certain classes with the lower transmission rate are always not scheduled. Thus, to address this issue, we define the following "*virtual transmission rate*" of the j-th of class-i dynamic flow

$$\tilde{R}_{ij}^{(d)}(t) = \theta_{ij}(t) R_{ij}^{(d)}(t) \ \forall t, \tag{33}$$

where

$$\theta_{ij}(t) = \frac{\psi}{\max\{R_{ij}^{(d)}(t_{ij}^{(d)}), R_{ij}^{(d)}(t_{ij}^{(d)}+1), \ldots, R_{ij}^{(d)}(t)\}}, \tag{34}$$

and $t_{ij}^{(d)}$ is the initial time that the j-th of class-i dynamic flow is added to the system, and ψ is a given non-negative constant. It is worth noting that $\max\{R_{ij}^{(d)}(t_{ij}^{(d)}), R_{ij}^{(d)}(t_{ij}^{(d)}+1), \ldots, R_{ij}^{(d)}(t)\}$ is used to perceive the maximum possible transmission rate of the j-th of class-i dynamic flow. Moreover, for any dynamic flow, if it achieves the maximum possible transmission rate at time slot t, its corresponding virtual transmission rates is equal to ψ, otherwise it is less than ψ. Similar to the H-QMW scheduling, A-H-QMW prioritizes a flow with the maximum product of the data backlog and the current transmission rate. However, the difference is that all the dynamic flows share the same data backlog in the A-H-QMW scheduling such that the system does not need to know the category of each dynamic flow, and $\tilde{R}_{ij}^{(d)}(t)$ replaces the actual transmission rate of the j-th of class-i dynamic flow. Based on the idea of H-QMW scheduling, the detailed process.

From Algorithm 1, we can observe that if a dynamic flow is scheduled, then it must have the maximum virtual transmission rate, that is, its actual transmission rate may achieve the maximum in the dynamic flows of the corresponding class. Moreover, all the dynamic flows have the same maximum virtual transmission rate, which results in each dynamic flow can be fairly scheduled when its transmission rate is at its maximum. Furthermore, similar to the H-QMW scheduling, the system can adaptively decide whether to schedule a persistent flow or a dynamic flow based on the data backlog of all dynamic flows and each persistent flow. On the other hand, we can find that the parameter ψ can be regarded as a weighting coefficient of $\tilde{Q}^{(d)}(t)$, which can be used to achieve the tradeoff between dynamic and persistent flow backlog. If the value of ψ is large enough, the system prioritizes the allocation of resources to dynamic flows, and vice versa.

Algorithm 1 A-H-QMW Scheduling Algorithm

Require: ψ;
Ensure: The scheduling policy $d_{ij}^{(\cdot)}(t)$;
1: **for** $t = 1, 2, \ldots, \infty$ **do**
2: Update $\theta_{ij}(t) = \frac{\psi}{\max\{R_{ij}^{(d)}(t_{ij}^{(d)}), R_{ij}^{(d)}(t_{ij}^{(d)}+1), \ldots, R_{ij}^{(d)}(t)\}}$ for each dynamic flow in the system;
3: Calculate the virtual transmission rate $\tilde{R}_{ij}^{(d)}(t) = \theta_{ij}(t) R_{ij}^{(d)}(t)$;
4: Calculate $W^{(p)} = \max_{i,j}\{Q_{ij}^{(p)} R_{ij}^{(p)}(t)\}$ and $W^{(d)} = \max_{i,j}\{\tilde{Q}^{(d)} \tilde{R}_{ij}^{(d)}(t)\}$;
5: **if** $W^{(p)} > W^{(d)}$ **then**
6: Select a persistent flow $\{i^*, j^*\} \in \arg\max_{i,j} Q_{ij}^{(p)}(t) R_{ij}^{(p)}(t)$ for transmission;
7: **else**
8: Randomly select a dynamic flow $\{i^*, j^*\}$ that satisfies $\{i^*, j^*\} \in \arg\max_{i,j}\{\tilde{Q}^{(d)} \tilde{R}_{ij}^{(d)}(t)\}$ for transmission;
9: **end if**
10: Update $Q_{ij}^{(p)}(t)$, $Q_{ij}^{(d)}(t)$, and $\tilde{Q}^{(d)}$.
11: **end for**

TABLE I
MCS TABLE

Order	MCS	SNR range (linear scale)
0	transmission failed	$(0, 0.6]$
1	QPSK, Rate 1/3	$(0.6, 2.135]$
2	QPSK, Rate 2/3	$(2.135, 4.565]$
3	16QAM, Rate 1/2	$(4.565, 8.584]$
4	16QAM, Rate 2/3	$(8.584, 13.583]$
5	16QAM, Rate 4/5	$(13.583, 19.498]$
6	64QAM, Rate 2/3	$(19.498, \infty]$

V. NUMERICAL ANALYSIS

In this section, we evaluate the performance of H-QMW with the existing representative throughput-optimal scheduling algorithms, including the QMW, MR, and F-D-MW scheduling algorithms. In addition, the algorithm of separating persistent and dynamic flows and scheduling them independently is also presented in our simulations. The throughput-optimal of H-QMW has been validated by simulations in Matlab.

In the simulation, the BS adopts the adaptive Modulation and Coding Scheme (MCS). The detailed MCSs used in the simulation are shown in Table I. The order of each MCS indicates that the number of packets that this MCS can transmit in one physical resource block. We do not specify the data rate for generality because different systems may have different

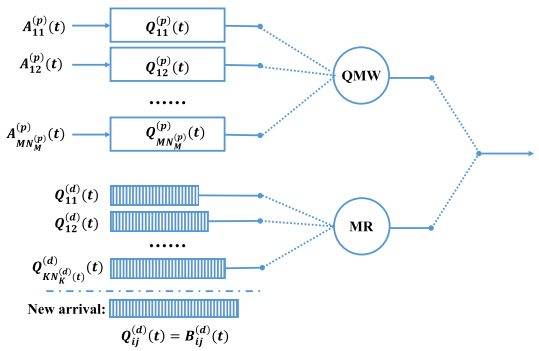


Fig. 4. Separating persistent and dynamic flows scheduling algorithm.

frame structures. In numerical simulations, we assume a basic data rate of 88 kb/s for the order-1 MCS as an example, which is calculated based on the current LTE system where each scheduling resource unit uses one physical resource block [38]. It is assumed that all channels are i.i.d Rayleigh fading channels, and thus the received signal-to-noise ratio (SNR) of each flow follows the exponential distribution. The average received SNR is set to 10 dB. The arrival process of dynamic flows follows a Poisson process, and the average initial flow size of dynamic flows is set to 10 Mbits. The arrival traffic rate of the system is calculated according to the traffic intensity in each simulation. All the presented simulation results are obtained for $T = 5 \times 10^5$ seconds which is long enough to examine if a scheduler is throughput-optimal or not. Unless otherwise stated, the traffic intensities for all the simulations are 0.99, which is very close to 1 and can easily result in instability if the adopted scheduler is not throughput-optimal.

A. Separating Persistent and Dynamic Flows Scheduling Algorithm

In order to explore the multiplexing gain and efficiency of the proposed H-QMW algorithm, we first introduce the separating persistent and dynamic flows scheduling algorithm (S-PDS). As shown in Fig. 4, for the S-PDS scheme, we assume that the scheduler can determine whether to select a persistent flow or a dynamic flow for transmission based on the pre-known traffic densities $\rho^{(p)}$ and $\rho^{(d)}$. Since QMW has been proved to achieve throughput-optimality policy for networks with persistent flows only, and MR is a throughput-optimal policy for the system with only dynamic flows, so when S-PDS scheduler decides to allocate this time slot to persistent flows, a persistent flow will be scheduled by adopting the QMW strategy; otherwise the MR algorithm will be used to schedule a dynamic flow for transmission. The detailed process of the S-PDS algorithm is outlined in Algorithm 2.

It is worth noting that, different from H-QMW, S-PDS determines which type of flows to transmit according to pre-known traffic intensities, which is independent of the current system backlog, resulting in a lower multiplexing gain among different types of flows. However, H-QMW can adaptively select a dynamic flow or persistent flow for transmission based on the current queue length and channel rate.

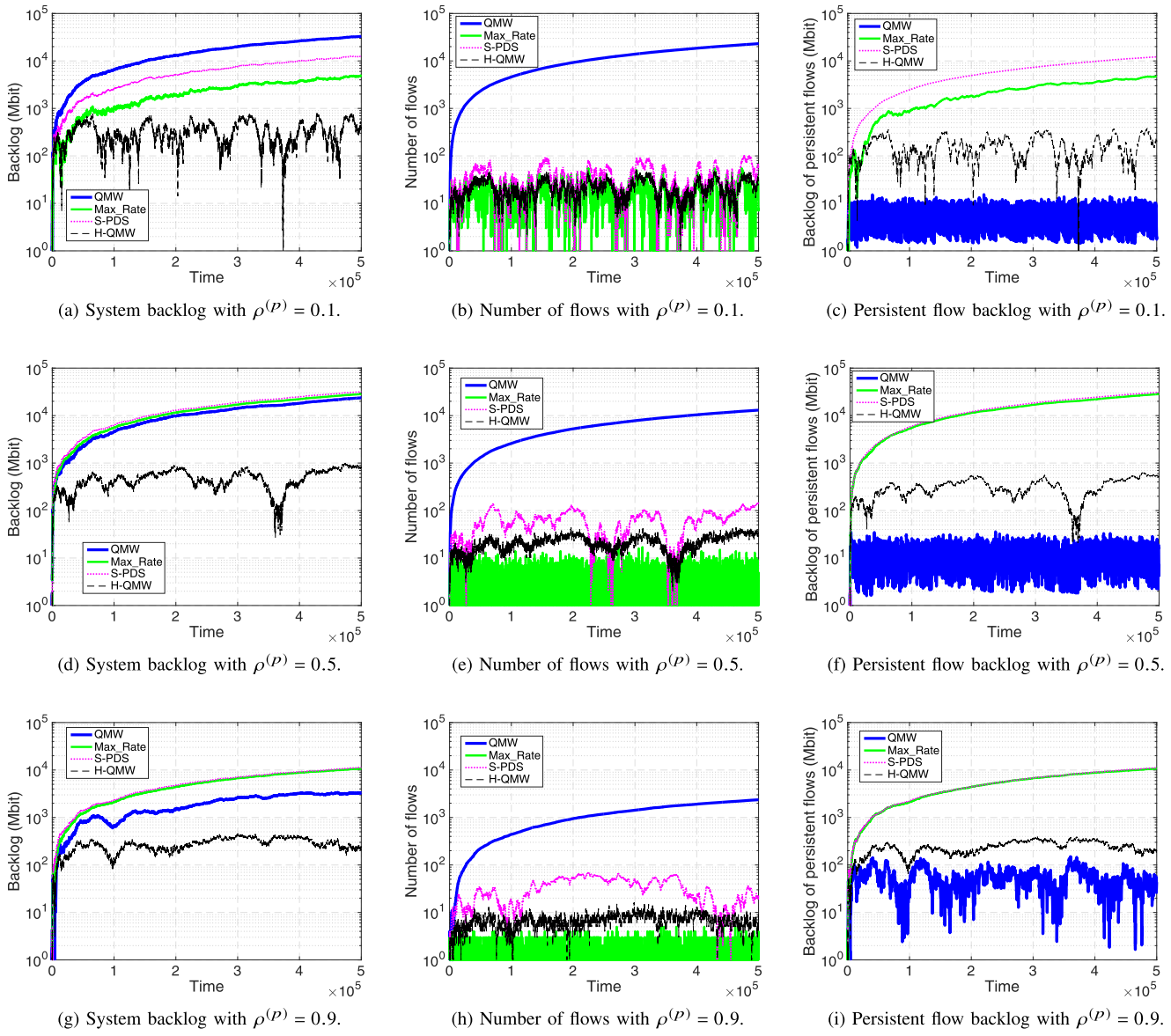
Algorithm 2 S-PDS Algorithm

Given $\rho^{(p)}$ and $\rho^{(d)}$
for $t = 1, 2, \ldots, \infty$ **do**
  Generate a random number $rand()$ follows the uniform distribution $U(0, 1)$;
  if $rand() \leq \frac{\rho^{(p)}}{\rho^{(p)}+\rho^{(d)}}$ **then**
    Select a persistent flow $\{i^*, j^*\} \in \arg\max_{i,j} Q_{ij}^{(p)}(t) R_{ij}^{(p)}(t)$ for transmission;
  else
    Randomly select a dynamic flow $\{i^*, j^*\}$ that satisfies $\{i^*, j^*\} \in \arg\max_{i,j} \frac{R_{ij}^{(d)}(t)}{R_i^{(d)\max}}$ for transmission;
  end if
end for

B. Performance Assessment

We first set the number of persistent flow to be one. The system backlog, the total number of flows, and the backlog of persistent flows with $\rho^{(p)} = \{0.1, 0.5, 0.9\}$ are shown in Fig. 5, respectively. With $\rho^{(p)} = 0.1$, H-QMW can suppress the growth of the system backlog for all time. For H-QMW, the number of dynamic flows and the backlog of persistent flows are always bounded over time. However, MR, QMW, and S-PDS fail to achieve queue stability. One may observe that QMW can ensure the backlog of persistent flows bounded, but the number of dynamic flows has the trend of growing into infinity over time. This can be interpreted as a dynamic flow has a finite amount of service requests when it arrives at the system, and once the backlog of some dynamic flows become smaller, these dynamic flows will be scheduled with a very low probability according to QMW, resulting in the number of dynamic flows is constantly increasing. The above analysis for the instability of QMW can be verified through Figs. 5(b)(e)(h). Also, we can find that S-PDS can guarantee the number of dynamic flows bounded, but the backlog of persistent flows shows an infinite growth trend over time. The reason is that the S-PDS scheduler only determines whether to schedule a persistent or dynamic flow based on the pre-known traffic intensities without considering the data backlog and channel rate state among different types of flows. This leads to the multiplexing gain among different types of flows is not fully utilized.

Increasing $\rho^{(p)}$ to 0.5, as expected, H-QMW always keeps the total amount of system backlog bounded over time demonstrating throughput-optimality. The number of dynamic flows is also stabilized by H-QMW. In contrast, the total backlog with QMW, MR and S-PDS cannot be stabilized. One interesting observation is that although MR can ensure the number of dynamic flows is bounded, the backlog of persistent flows tends to be unstable. The reason is that as the number of dynamic flows increases, the dynamic flows will be scheduled with a higher probability. This leads to the dynamic flows in MR share too much of the channel time, and thus the persistent flow has an insufficient share of channel time to transmit the



(a) System backlog with $\rho^{(p)} = 0.1$. (b) Number of flows with $\rho^{(p)} = 0.1$. (c) Persistent flow backlog with $\rho^{(p)} = 0.1$.

(d) System backlog with $\rho^{(p)} = 0.5$. (e) Number of flows with $\rho^{(p)} = 0.5$. (f) Persistent flow backlog with $\rho^{(p)} = 0.5$.

(g) System backlog with $\rho^{(p)} = 0.9$. (h) Number of flows with $\rho^{(p)} = 0.9$. (i) Persistent flow backlog with $\rho^{(p)} = 0.9$.

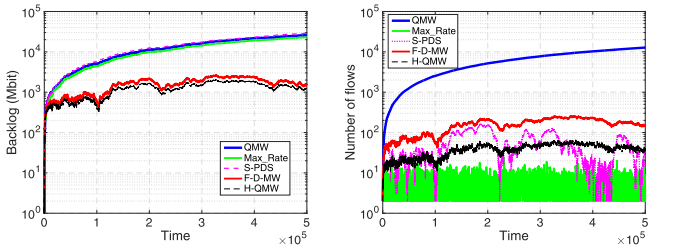
Fig. 5. The system backlog, the total number of flows, and the backlog of persistent flows with $\rho^{(p)} = \{0.1, 0.5, 0.9\}$.

arrival traffic, i.e., for MR, the time allocation of the persistent is less than $\rho^{(p)}$. Meanwhile, the higher share of resources for $Q^{(d)}$ keeps the number of dynamic flows very low, and thus the total number of flows, including both the persistent and dynamic flows, is also limited to very low. As a result, the probability of always scheduling a flow in the maximum possible transmission rate reduces dramatically, and hence the channel is not efficiently utilized without fully exploring the multiuser diversity gain. The above analysis for the instability of MR can be verified through Figs. 5(d)(e)(f).

In Fig. 5(b) with $\rho^{(p)} = 0.1$, MR has the least $N(t)$, H-QMW and S-PDS have a much higher $N(t)$ than MR, but it is well bounded. $N(t)$ of QMW cannot be stabilized. In Fig. 5(e) with $\rho^{(p)} = 0.5$, since the workload from the dynamic flows is reduced, $N(t)$ of MR is further reduced. However, a fewer number of flows means less chance to transmit in $R^{\max}$, which leads to the scheduler will take more time resources to transmit dynamic flows and the backlog of persistent flows become larger. As a result, MR shows instability in Figs. 5(d)(f). $N(t)$ of H-QMW is maintained on the same level as that in Fig. 5(b), which gives the scheduler plenty of possibilities to choose the flows in $R^{\max}$ channel state.

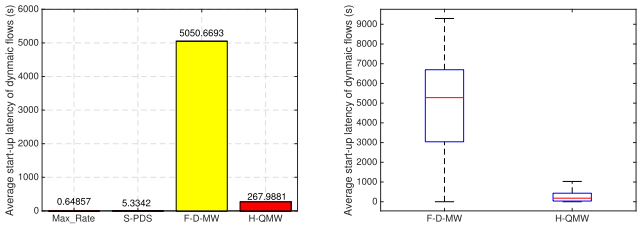
Increasing $\rho^{(p)}$ to 0.9, the system backlog is shown in Fig. 5(g). We can observe the same trend comparing with that in Fig. 5(d), i.e., only H-QMW is able to stabilize the system backlog, while QMW, MR, and S-PDS result in an unstable system. The number of flows with $\rho^{(p)} = 0.9$ is shown in Fig. 5(h). We can observe again that $N(t)$ of MR is too small to fully exploit the multiuser diversity gain, and $N(t)$ of QMW cannot be stabilized. Although S-PDS can ensure that $N(t)$ is bounded, it fails to achieve stability of persistent flows. Only $N(t)$ of H-QMW is stabilized at a proper level so that the multiuser diversity gain is fully used, and the throughput-optimality is achieved.

Increasing the number of persistent flows in the system to two and setting $\rho^{(p)} = 0.5$, the system backlog and the number



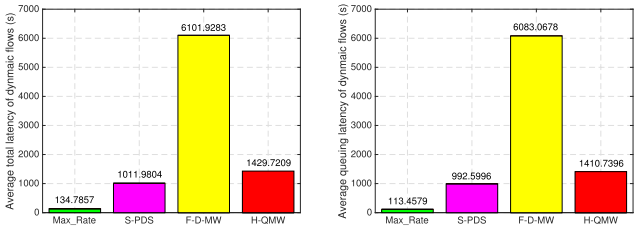
(a) System backlog with $\rho^{(p)} = 0.5$. (b) Number of flows with $\rho^{(p)} = 0.5$.

Fig. 6. The system backlog and the backlog of persistent flows with two persistent flows and $\rho^{(p)} = 0.5$.



(a) Average start-up latency of dynamic flows (b) Start-up latency of dynamic flows

Fig. 7. Start-up latency of dynamic flows with two persistent flows and $\rho^{(p)} = 0.5$.



(a) Average total latency of dynamic flows (b) Average queuing delay of dynamic flows

Fig. 8. Average total and queuing latency of each dynamic flow with two persistent flows and $\rho^{(p)} = 0.5$.

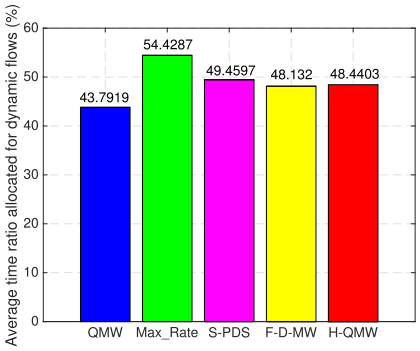
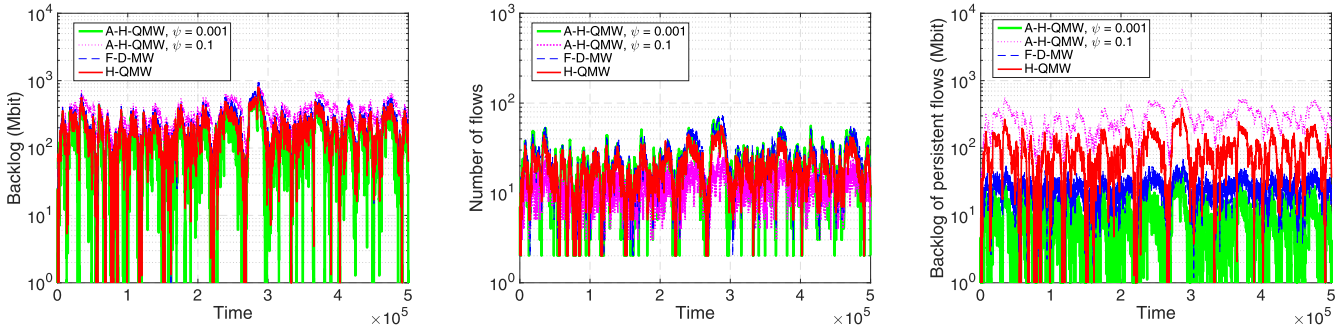


Fig. 9. Average time ratio allocated for dynamic flows with two persistent flows and $\rho^{(p)} = 0.5$.

of flows are shown in Fig. 6. The simulation results further validate our conclusion that H-QMW is able to stabilize the hybrid system with both persistent and dynamic flows. In order to further verify this, we present the performance of F-D-MW, which has been proven to be throughput-optimal for hybrid systems in [33]. From Fig. 6(a), we may observe that there is only a marginal gap between H-QMW and F-D-MW in terms of system backlog, and their trajectory is also identical, which validates our results. However, from Fig. 6(b), we can find that F-D-MW has a much higher $N(t)$ than H-QMW. This because by adopting F-D-MW, new arrival dynamic flows in the system may suffer a long start-up latency, resulting in a certain amount of dynamic flows that are backlogged in the system.

Fig. 7(a) presents the average start-up latency of dynamic flows with different scheduling algorithms. Since QMW cannot ensure the stability of dynamic flows, resulting in the average start-up latency has a trend of growing into infinity over time, so the average start-up latency of QMW is not shown in this figure. From Fig. 7(a), we can find that F-D-MW has the largest start-up latency and MR has the smallest start-up latency. This is because F-D-MW always gives the priority to the flows with the largest product of the delay and the current transmission rate, and so new arrival dynamic flows have to wait long enough before being scheduled for the first time. On the other hand, as the number of dynamic flows increases, MR will schedule the dynamic flows for transmission with a higher probability, which leads to smaller start-up latency. Besides, we can observe that the average start-up latency of H-QMW is much smaller than that of F-D-MW. The reason is that for H-QMW, if a dynamic flow is scheduled, then it must have the maximum transmission rate, resulting in new arriving dynamic flows have the same opportunity to be scheduled as the previously existing dynamic flows (the wireless channel is i.i.d.). Fig. 7(b) shows the box-plot of the average start-up latency delay for H-QMW and F-D-MW scheduler. We can see that the H-QMW has a smaller deviation of the start-up latency than the F-D-MW scheduling, which indicates that the H-QMW has better fairness in scheduling dynamic flows than the F-D-MW.

Fig. 8 shows the average total latency and queuing delay of each dynamic flow for different scheduling policies, in which the average total latency consists of the average queuing delay and transmission delay. The average transmission delay can be calculated by subtracting the average queuing delay from the total latency, and thus the average transmission delay of MR, S-PDS, F-D-MW, and H-QMW are $\{21.33, 19.38, 18.86, 18.98\}$ seconds, respectively. We can see that MR has the smallest total latency and queuing delay, followed by S-PDS. Also, the average total latency and queuing delay of H-QMW are much smaller than those using F-D-MW. Furthermore, the average transmission delay of MR is larger than that of H-QMW and F-D-MW. This can be explained as follows: i) For the MR scheme, since the number of dynamic flows is much larger than that of persistent flows, the MR scheduler prefers to selecting a dynamic flow for transmission. From Fig. 6(b), we can find that, with MR, the number of dynamics flows in the system is smaller than those of other schemes, result in a lower multiuser diversity gain. Therefore, MR may not always schedule a dynamic flow at its maximum transmission rate, leading to a longer average transmission delay for dynamic flows. ii) F-D-MW scheduler gives a higher priority to the flows with the longer delay. The persistent flow delay is defined as the sum of all the files (packets) delay in the corresponding persistent flow queue, so F-D-MW prefers to schedule a persistent flow for transmission.



(a) System backlog. (b) Number of flows. (c) Backlog of persistent flows.

Fig. 10. System backlog, the total number of flows, and the backlog of persistent flows.

TABLE II
AVERAGE RECEIVED SNR (LINEAR SCALE)

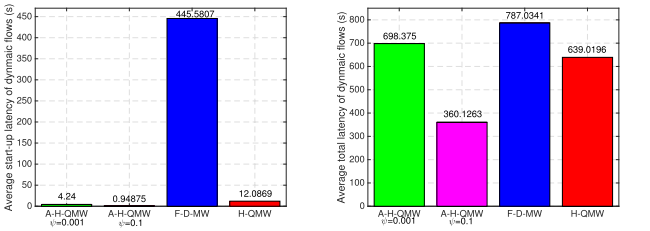
Average SNR / Class / Flow type	1	2	3
Persistent flows	5	10	
Dynamic flows	5	10	15

Fig. 9 presents the average time ratio allocated for dynamic flows with different scheduling policies. As expected, MR and QMW allocate the most and the least time resources for dynamic flows, respectively. In addition, we can observe that the time ratio of H-QMW assigned for dynamic flows is slightly larger than F-D-MW, combined with Fig. 8, which explains why the average total latency of each dynamic flow for H-QMW is smaller than F-D-MW.

Fig. 10 and Fig. 11 show the performance of the proposed A-H-QMW scheduling. In these figures, we set the number of classes for persistent and dynamic flows to be two and three, respectively. All the channels are assumed to follow Rayleigh distributions, and the corresponding average received signal-to-noise ratio of each class is given in Table II. The arrival process of all dynamic flows is assumed to follow a Poisson process. The initial flow size of the first class dynamic flow is set to a constant, which equals 5 Mbit. Each flow size of the second and third classes dynamic flows follows the exponential distribution with mean $\{10, 20\}$ Mbit, respectively. The traffic intensities of the persistent and dynamic flows are set to $\rho^{(p)} = \{0.2, 0.29\}$ and $\rho^{(d)} = \{0.1, 0.2, 0.2\}$, respectively.

Fig. 10 presents the system backlog, the number of flows, and the backlog of persistent flows. We can easily observe that the trajectory of the A-H-QMW is almost identical to other throughput-optimal scheduling algorithms. Moreover, A-H-QMW also keeps the number of flows and the backlog of persistent flow bounded over time. All of these verify that the proposed A-H-QMW can effectively ensure the stability of the system without the knowledge of the class of each dynamic flow.

Fig. 11 shows the average start-up and total latency of each dynamic flows for different scheduling policies. We can find that, compared to F-D-MW and H-QMW scheduling, A-H-QMW has a smaller start-up and total latency of each dynamic



(a) Average start-up latency of dynamic flows (b) Average total latency of dynamic flows

Fig. 11. Start-up and total latency of dynamic flows.

flow when $\psi = 0.1$, which implies that the system priorities the allocation of resources to dynamic flows. However, when $\psi = 0.001$, A-H-QMW has a larger total latency for dynamic flows than H-QMW. Combined with Fig. 10, we can find that A-H-QMW has the smaller backlog of persistent flows in the system than H-QMW when $\psi = 0.001$. All of these indicate that A-H-QMW can achieve the tradeoff between dynamic and persistent flow backlog. Thus, we can design an appropriate ψ to meet the different QoS requirements.

From all the performance evaluation, we make the following conclusions for the schedulers in hybrid systems. First, H-QMW and A-H-QMW are verified to be throughput-optimal. Second, although when the system has only dynamic flows, MR is throughput-optimal, when the system has both persistent and dynamic flows, MR is not throughput-optimal. Third, the proposed H-QMW and A-H-QMW have smaller start-up latency and total latency of each dynamic flow than that of F-D-MW. Finally, the approach of separating the two types of flows and scheduling them independently cannot ensure the stability of the hybrid system.

VI. CONCLUSION

In this work, we have investigated the scheduling algorithm design for the coexistence of persistent and dynamic flows. First, we present the definition of the capacity region for the hybrid system with channel rate variation. Second, we proposed an online H-QMW scheduling algorithm by introducing the virtual queue for dynamic flows and using the Lyapunov framework, which has been proven to achieve throughput-optimality for hybrid systems. Third, we propose a more realistic A-H-QMW scheduling algorithm for the scenario that the system does not need to know the classification of dynamic

flows. At last, we not only reveal that H-QMW and A-H-QMW can achieve the smaller start-up and total latency for dynamic flows than F-D-MW, but also show that the approach separating the two types of flows and scheduling them independently cannot ensure the stability of hybrid systems. In addition, it is shown that the MR scheduling algorithm, throughput-optimal for dynamic flows, is not throughput-optimal when the dynamic flows coexist with persistent flows.

In this paper, since our objective is to design a scheduling algorithm to achieve the capacity region, the delay is not incorporated in our design. However, the delay performance is a sensitive QoS metric in practice. In fact, if we consider an exact delay QoS demand, this may lead to some loss in the achievable rate region. Therefore, there exists a tradeoff between the stability of all queues and the average delay. In particular, for the proposed H-QMW scheduling algorithm, when there are no more new dynamic flows arriving at the system, this may result in that the dynamic flow in the existing system will not be scheduled. However, we can address this issue if the exact delay QoS demand is incorporated in our scheduling algorithm design. Thus, a further in-depth research is beckon to design an efficient scheduling algorithm to guarantee the exact delay QoS demand while satisfying the stability of all queues.

REFERENCES

[1] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Autom. Control*, vol. 37, no. 12, pp. 1936–1948, Dec. 1992.

[2] L. Tassiulas, "Scheduling and performance limits of networks with constantly changing topology," *IEEE Trans. Inf. Theory*, vol. 43, no. 3, pp. 1067–1073, May 1997.

[3] L. Tassiulas and A. Ephremides, "Dynamic server allocation to parallel queues with randomly varying connectivity," *IEEE Trans. Inf. Theory*, vol. 39, no. 2, pp. 466–478, Mar. 1993.

[4] P. van de Ven, S. Borst, and S. Shneer, "Instability of MaxWeight scheduling algorithms," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Apr. 2009, pp. 1701–1709.

[5] S. Liu, L. Ying, and R. Srikant, "Scheduling in multichannel wireless networks with flow-level dynamics," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 38, no. 1, pp. 191–202, 2010.

[6] B. Sadiq and G. de Veciana, "Throughput optimality of delay-driven MaxWeight scheduler for a wireless system with flow dynamics," in *Proc. Allerton Conf. Commun., Control Comput. (Allerton)*, Oct. 2009, pp. 1097–1102.

[7] L. B. Le, K. Jagannathan, and E. Modiano, "Delay analysis of maximum weight scheduling in wireless Ad Hoc networks," in *Proc. 43rd Annu. Conf. Inf. Sci. Syst.*, Mar. 2009, pp. 389–394.

[8] Y. Song, C. Zhang, and Y. Fang, "Minimum energy scheduling in multi-hop wireless networks with retransmissions," *IEEE Trans. Wireless Commun.*, vol. 9, no. 1, pp. 348–355, Jan. 2010.

[9] A. Eryilmaz and R. Srikant, "Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control," in *Proc. IEEE 24th Annu. Joint Conf. IEEE Comput. Commun. Soc. (INFOCOM)*, Mar. 2005, pp. 1794–1803.

[10] X. Wang, Y. Chen, L. Cai, and J. Pan, "Scheduling in a secure wireless network," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Apr./May 2014, pp. 2184–2192.

[11] L. Zheng and L. Cai, "A distributed demand response control strategy using Lyapunov optimization," *IEEE Trans. Smart Grid*, vol. 5, no. 4, pp. 2075–2083, Jul. 2014.

[12] J. Chen, W. Xu, S. He, Y. Sun, P. Thulasiraman, and X. Shen, "Utility-based asynchronous flow control algorithm for wireless sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 7, pp. 1116–1126, Sep. 2010.

[13] S. Shakkottai and A. L. Stolyar, "Scheduling for multiple flows sharing a time-varying channel: The exponential rule," *Transl. Amer. Math. Soc.*, vol. 207, pp. 185–202, Dec. 2002.

[14] B. Sadiq and G. de Veciana, "Optimality and large deviations of queues under the pseudo-log rule opportunistic scheduling," in *Proc. Allerton Conf. Commun., Control Comput. (Allerton)*, Sep. 2008, pp. 776–783.

[15] B. Sadiq, S. J. Baek, and G. De Veciana, "Delay-optimal opportunistic scheduling and approximations: The log rule," *IEEE/ACM Trans. Netw.*, vol. 19, no. 2, pp. 405–418, Apr. 2011.

[16] K. Jagannathan, M. Markakis, E. Modiano, and J. N. Tsitsiklis, "Throughput optimal scheduling in the presence of heavy-tailed traffic," in *Proc. Allerton Conf. Commun., Control Comput. (Allerton)*, Sep./Oct. 2010, pp. 953–960.

[17] C. Joo, "On the performance of back-pressure scheduling schemes with logarithmic weight," *IEEE Trans. Wireless Commun.*, vol. 10, no. 11, pp. 3632–3637, Nov. 2011.

[18] P. van de Ven, S. Borst, and L. Ying, "Spatial inefficiency of MaxWeight scheduling," in *Proc. IEEE WiOpt*, May 2011, pp. 62–69.

[19] S. Liu, L. Ying, and R. Srikant, "Throughput-optimal opportunistic scheduling in the presence of flow-level dynamics," *IEEE/ACM Trans. Netw.*, vol. 19, no. 4, pp. 1057–1070, Aug. 2011.

[20] Y. Chen, X. Wang, and L. Cai, "HOL delay based scheduling in wireless networks with flow-level dynamics," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2014, pp. 4898–4903.

[21] B. Li and A. Eryilmaz, "Optimal distributed scheduling under time-varying conditions: A fast-CSMA algorithm with applications," *IEEE Trans. Wireless Commun.*, vol. 12, no. 7, pp. 3278–3288, Jul. 2013.

[22] Q. Li and R. Negi, "Distributed throughput-optimal scheduling in ad hoc wireless networks," in *Proc. IEEE ICC*, vol. 2011, pp. 1–5.

[23] S. Xia and P. Wang, "Distributed throughput optimal scheduling in the presence of heavy-tailed traffic," in *Proc. IEEE ICC*, Jun. 2015, pp. 1–5.

[24] N. Lu, B. Li, R. Srikant, and Y. Lei, "Optimal distributed scheduling of real-time traffic with hard deadlines," in *Proc. IEEE CDC*, Dec. 2016, pp. 4408–4413.

[25] M. Andrews *et al.*, "Scheduling in a queuing system with asynchronously varying service rates," *Probab. Eng. Inf. Sci.*, vol. 18, no. 2, pp. 191–217, 2004.

[26] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch," *IEEE Trans. Commun.*, vol. 47, no. 8, pp. 1260–1267, Aug. 1999.

[27] A. Eryilmaz, R. Srikant, and J. R. Perkins, "Stable scheduling policies for fading wireless channels," *IEEE/ACM Trans. Netw.*, vol. 13, no. 2, pp. 411–424, Apr. 2005.

[28] B. Li, R. Li, and A. Eryilmaz, "Throughput-optimal scheduling design with regular service guarantees in wireless networks," *IEEE/ACM Trans. Netw.*, vol. 23, no. 5, pp. 1542–1552, Oct. 2015.

[29] B. Li, R. Li, and A. Eryilmaz, "Wireless scheduling design for optimizing both service regularity and mean delay in heavy-traffic regimes," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1867–1880, Jun. 2016.

[30] N. Lu, B. Ji, and B. Li, "Age-based scheduling: Improving data freshness for wireless real-time traffic," in *Proc. ACM MobiHoc*, 2018, pp. 191–200.

[31] M. J. Neely, "Delay-based network utility maximization," *IEEE/ACM Trans. Netw.*, vol. 21, no. 1, pp. 41–54, Feb. 2013.

[32] B. Ji, C. Joo, and N. B. Shroff, "Delay-based back-pressure scheduling in multihop wireless networks," *IEEE/ACM Trans. Netw.*, vol. 21, no. 5, pp. 1539–1552, Oct. 2013.

[33] B. Li, A. Eryilmaz, and R. Srikant, "On the universality of age-based scheduling in wireless networks," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Apr./May 2015, pp. 1302–1310.

[34] Y. Chen, X. Wang, and L. Cai, "On achieving fair and throughput-optimal scheduling for TCP flows in wireless networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 12, pp. 7996–8008, Dec. 2016.

[35] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures Commun. Netw.*, vol. 3, no. 1, pp. 1–211, 2010.

[36] A. W. Moore and D. Zuev, "Internet traffic classification using Bayesian analysis techniques," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 33, no. 1, pp. 50–60, 2005.

[37] Z. Li, R. Yuan, and X. Guan, "Accurate classification of the Internet traffic based on the SVM method," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2007, pp. 1373–1378.

[38] Y. Li and L. Cai, "Cooperative device-to-device communication for uplink transmission in cellular system," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 3903–3917, Jun. 2018.

Xiaolong Lan received the B.S. degree in mathematics and applied mathematics from the Chengdu University of Technology, China, in 2012. He is currently pursuing the Ph.D. degree with the School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China. Since 2017, he has been a visiting Ph.D. student with the University of Victoria. His current research interests include buffer-aided communication, energy-harvesting wireless communication, mobile edge computing, and network scheduling.

Yi Chen received the B.Eng. and M.S. degrees from Northwest Polytechnical University, Xi'an, China, in 2011 and 2008, respectively, and the Ph.D. degree in electrical engineering from the University of Victoria, Canada, in 2016. His research interests include scheduling and resource allocation in wireless networks.

Lin Cai (S'00–M'06–SM'10) received the M.A.Sc. and Ph.D. degrees in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2002 and 2005, respectively.

Since 2005, she has been with the Department of Electrical and Computer Engineering, University of Victoria, where she is currently a Professor. Her research interests span several areas in communications and networking, with a focus on network protocol and architecture design supporting emerging multimedia traffic and the Internet of Things.

Dr. Cai was a recipient of Outstanding Achievement in Graduate Studies, the NSERC E. W. R. Steacie Memorial Fellowships in 2019, the NSERC Discovery Accelerator Supplement (DAS) Grants in 2010 and 2015, respectively, and the Best Paper awards of the IEEE ICC 2008 and the IEEE WCNC 2011. She has founded and chaired the IEEE Victoria Section Vehicular Technology and Communications Joint Societies Chapter. She has been elected to serve the IEEE Vehicular Technology Society Board of Governors, from 2019 to 2021. She has served as an Area Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, a member of the Steering Committee of the IEEE TRANSACTIONS ON BIG DATA (TBD) and the IEEE TRANSACTIONS ON CLOUD COMPUTING (TCC), an Associate Editor of the IEEE INTERNET OF THINGS JOURNAL, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the IEEE TRANSACTIONS ON COMMUNICATIONS, the *EURASIP Journal on Wireless Communications and Networking*, the *International Journal of Sensor Networks*, and the *Journal of Communications and Networks* (JCN), and the Distinguished Lecturer of the IEEE VTS Society. She has served as a TPC Symposium Co-Chair for the IEEE Globecom'10 and Globecom'13. She is a Registered Professional Engineer of British Columbia, Canada.