Heavy-Ball: A New Approach to Tame Delay and Convergence in Wireless Network Optimization

Jia Liu[†] Atilla Eryilmaz[†] Ness B. Shroff[†] Elizabeth S. Bentley[*]
[†]Department of Electrical and Computer Engineering, The Ohio State University
[*]Air Force Research Laboratory, Information Directorate

***Abstract*—The last decade has seen significant advances in optimization-based resource allocation and control approaches for wireless networks. However, the existing work suffer from poor performance in one or more of the metrics of optimality, delay, and convergence speed. To overcome these limitations, in this paper, we introduce a largely overlooked but highly effective *heavy-ball* optimization method. Based on this heavy-ball technique, we develop a cross-layer optimization framework that offers utility-optimality, fast-convergence, and significant delay reduction. Our contributions are three-fold: i) we propose a heavy-ball joint congestion control and routing/scheduling framework for both single-hop and multi-hop wireless networks; ii) we show that the proposed heavy-ball method offers an elegant *three-way trade-off* in utility, delay, and convergence, which is achieved under a near index-type simple policy; and more importantly, iii) our work opens the door to an unexplored network control and optimization paradigm that leverages advanced optimization techniques based on "memory/momentum" information.**

I. Introduction

Due to the rapidly increasing mobile data demands, recent years have witnessed a large body of works on resource allocation that aim to maximize the network utility in wireless networks (see, e.g., [1]–[4], and [5] for a survey). This has led to an elegant mathematical decomposition framework, from which "loosely-coupled" congestion control, scheduling, and routing algorithms naturally emerge. These algorithms do not require statistical knowledge of either the arrivals or channel states. Instead, they only rely on queue-lengths and channel state information to make control decisions. These algorithms are also inherently connected to the Lagrangian dual decomposition framework plus the subgradient method in nonlinear optimization theory [1], [2], where (scaled) queue-lengths can be interpreted as Lagrangian dual variables and the queue-length updates play the role of subgradient directions. Moreover, variants of the "MaxWeight" scheduling component in this framework have already been adopted and implemented in practice (e.g., Qualcomm's Flashlinq peer-to-peer wireless networks [6] and data center bridging by Cisco [7], etc.).

Despite the aforementioned attractive features, these queue-length-based algorithms (QLA) suffer from several key limitations. First, in the existing QLA framework, it has been shown that a utility-optimality gap $O(1/K)$ can be achieved with an $O(K)$ penalty in queueing delay, where $K > 0$ is a system parameter. Hence, a small utility-optimality gap necessitates a large K and results in large queueing delay. To address this limitation, there have been significant recent efforts (e.g., [4], [8]–[10], etc.) focusing on reducing the queueing delay (see Section II for more in-depth discussions). Also, in the existing QLA framework, the queue-length-based weight adjustment ignores the curvature of the objective function contour and uses a small step-size in each iteration [1]–[4], which leads to unsatisfactory convergence speed. To address this problem, some second-order congestion control and routing/scheduling algorithms have been proposed recently to accelerate the convergence speed (see, e.g., [11], [12]). However, due to their complex algorithmic structures, these second-order approaches require a much larger information exchange overhead and do not scale well with the network size. These limitations of the existing approaches motivate us to pursue a new design that leverages the *heavy-ball*-based optimization framework.

Historically, the heavy-ball method was first proposed by Polyak in 1964 [13] for solving unconstrained convex optimization problems, with the original goal of accelerating the convergence of the gradient descent method. The basic idea of the heavy-ball method is that, rather than using only the (sub)gradient information at the current iterate and being memoryless of the past iterates' trajectory, one computes the search direction using a linear combination of the gradient (analogous to "potential") and the update direction in the previous step (analogous to "momentum"). The method can be viewed as a discrete version of the second-order ordinary differential equation (ODE) that describes a heavy body's motion in a potential field, hence the name "heavy-ball." It has been shown that, by appropriately weighing the "potential" and "momentum," the algorithm is insensitive to the objective contour and leads to a much faster convergence [14]. Indeed, the convergence speed advantage is our initial motivation behind adopting the heavy-ball approach for wireless network optimization. Yet surprisingly, as we show later in this paper, the benefits of adopting the heavy-ball idea go *far beyond* convergence acceleration and entail dramatic *delay reduction*.

We note, however, that due to a number of technical challenges, developing a heavy-ball-based solution for wireless network utility optimization is not straightforward. First, since the heavy-ball method was originally designed for unconstrained static optimization, we need to modify the heavy-ball

This work has been supported by NSF grants CNS-1527078, 1514260, 1446582, 1409336, 1012700, WiFiUS-1456806, ECCS-1444026, 1232118; ONR grant N00014-15-1-2166; ARO grant W911NF-14-1-0368; DTRA grants HDTRA 1-14-1-0058, 1-15-1-0003, AFRL VFRP'15 award; DARPA grant HROOll-15-C-0097and QNRF grant NPRP 7-923-2-344. DISTRIBUTION STATEMENT A: Approved for Public Release; distribution unlimited 88ABW-2015-5989 on Dec. 14, 2015.

method for wireless network utility maximization, which is a constrained stochastic optimization problem with a much more complex structure. Second, unlike the obvious connection between queue-lengths and dual variables in the QLA design, the relationship between the heavy-ball method and the observable network state information (e.g., queue-lengths, channel states, etc.) is unknown. Hence, a key challenge that we will answer in this paper is to characterize the trade-off between delay and achieved network utility under the heavy-ball approach. Third, due to the inclusion of past iterations, the algorithmic structure of a heavy-ball method is different from that of the QLA design. As a result, new analytical techniques are required to analyze the performance of the heavy-ball approach.

The key contribution of this paper is that, by addressing the above challenges, we reveal the potential of many memory/momentum-based optimization techniques that could be leveraged to produce surprising network performance gains in delay, throughput, and convergence. The main results and technical contributions of this paper are as follows:

- Motivated by the heavy-ball idea, we propose a new weight adjustment scheme for joint congestion control and routing/scheduling in wireless networks. Our work not only provides a synergy between the heavy-ball algorithm and observable network state information (queue-lengths and channel states) to allow simple implementation in practice, it also extends and generalizes the classical heavy-ball method from unconstrained static optimization to the constrained stochastic network utility optimization paradigm, thus advancing the state-of-the-art of the heavy-ball method in mathematical optimization theory.
- Under our heavy-ball-based joint congestion control and scheduling scheme with a β-parameterized momentum ($\beta \in [0, 1)$ is a system parameter typically chosen close to 1), we show that the delay is $(1-\beta)$–*fraction* of that of the QLA approach. More specifically, our theoretical analysis unveils that a utility-optimality gap $O(1/K)$ can be achieved with an $O((1-\beta)K) + O((1+\beta)\sqrt{K})$ cost in queueing-delay, where the parameter K is inversely proportional to the step-size in the heavy-ball method. Further, in the asymptotic regime of K where β is chosen as $\beta = 1 - O(1/\sqrt{K})$, our heavy-ball algorithm achieves an $[O(1/K), O(\sqrt{K})]$ utility-delay trade-off, which is significantly better than the well-known $[O(1/K), O(K)]$ trade-off of the QLA methods.
- Given K and β, we show that the convergence factor of our heavy-ball algorithm scales as $\max\{\sqrt{\beta}, |1+\beta-\phi/K|-\sqrt{\beta}, |1+\beta-\Phi/K|-\sqrt{\beta}\}$, where Φ and ϕ are the upper and lower bounds of the Hessian eigen-spectrum. Combined with the results in the previous bullet, our heavy-ball algorithm offers an elegant *three-way performance trade-off* governed by *two control degrees of freedom* in K and β. Most notably, *simultaneous* utility-optimality and low-delay is made possible by trading off convergence speed. We note that this important three-way trade-off relationship has *not* been discovered so far in the literature.

We hope that, collectively, our results in this paper could pave the way for many new networking research directions that explore advanced memory/momentum-based optimization methods to improve key network performance metrics. The remainder of this paper is organized as follows. In Section II, we review related works. Section III introduces the network model and problem formulation. Section IV presents our heavy-ball algorithm and the performance analysis of the algorithm. In Section V, we extend the proposed algorithm to multi-hop networks. Section VI presents numerical results and Section VII concludes this paper.

II. Related Work

In this section, we first review the state-of-the-art of the QLA literature that is closely related to this paper. As mentioned earlier, there have been significant efforts on reducing the delay of the QLA approaches. For example, in [4], a virtual queue technique similar to those in [15]–[17] was adopted, where the virtual queue-lengths evolve based on service rates that are a fraction of the actual service rates. In [9], a virtual backlog mechanism with place-holder bits instead of real data was proposed. It was shown that, by accepting some non-zero packet dropping probability, this approach achieves an $[O(1/K), O(\log^2(K))]$ utility-delay trade-off. An exponential Lyapunov virtual backlog method combined with a threshold-based packet-dropping scheme was also proposed in [8] to achieve an $O(\log(K))$ delay. Although having a log-type delay growth, a major limitation of [8], [9] is that choosing the size of place-holder bits [9] and the threshold value [8] require *non-causal* global arrival and channel statistics (cf. [8, Eq. (17)], [9, Eq. (45)]), which is usually infeasible. Also, if the parameters are set inappropriately, these schemes may suffer non-negligible packet dropping probability. To address this problem, a per-iteration learning was proposed in [10] to learn the optimal size of place-holder bits in an online fashion. However, the per-iteration learning mechanism significantly increases the complexity. In some sense, all these delay reduction schemes can be viewed as sacrificing throughput-optimality (reflected in reduced service rates or packet dropping) for delay reduction. In contrast, *without sacrificing any throughput-optimality and without requiring any non-causal statistical knowledge*, our heavy-ball scheme achieves an $[O(1/K), O(\sqrt{K})]$ utility-delay trade-off by setting $\beta = 1 - O(\frac{1}{\sqrt{K}})$. Moreover, our heavy-ball algorithm enables an elegant three-way trade-off that *cannot* be offered by the existing works in [4], [8]–[10].

Next, we provide further background of the heavy-ball method and then review the related works in the heavy-ball domain. In the optimization literature, the heavy-ball method is also referred to as the multi-step or momentum method. Since its inception [13], the heavy-ball method has found applications in signal processing and machine learning (see, e.g., [18] and references therein). However, the heavy-ball method remains largely unexplored in networking research so far. To our knowledge, the only application of the heavy-ball method in networking areas can be found in [19], where the authors developed a heavy-ball-based Internet congestion control scheme. We note that our work differs from [19]

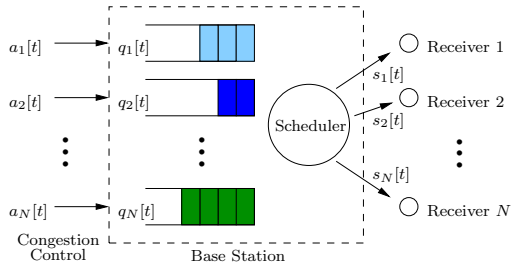


Fig. 1. An illustration of the single-hop cellular downlink.

in the following key aspects: First, our proposed heavy-ball algorithm is a dynamic scheme that works with stochastic wireless channels, while the algorithm proposed in [19] solves a static congestion control rate optimization problem for wireline networks. Second, the algorithm in [19] requires some assumptions (c.f. [19, Sec. VII-C]) to turn the problem into an *unconstrained* formulation, so that the classical heavy-ball method can be applied. However, as indicated in [19], these assumptions restricted the use of the heavy-ball method to problems with certain routing structures. In contrast, our proposed method can handle all network constraints and works with all utility optimization problems. Third, we derive explicit utility-delay-convergence trade-off scaling laws in this paper, while no such results were provided in [19].

III. Network Model and Problem Formulation

From this section until Section IV, we will consider a single-hop wireless network with N links, which can be used to represent a cellular base station (or access point) downlink/uplink channel with N users or a set of distributed communication pairs in an ad hoc network. We will later (in Section V) discuss how to extend the results to multi-hop wireless networks. The rationale behind this presentation approach is that the single-hop network model will allow us to present the *core idea* behind the heavy-ball-based design with less notational clutter, before we integrate further system dynamics in multi-hop wireless networks. Also, as mentioned above, since the single-hop model encompasses a large number of networks in practice, it is important in its own right.

Notation: We use boldface to denote matrices/vectors. We let $\mathbf{A}^\top$ denote the transpose of $\mathbf{A}$. We let $\mathbf{I}_N$ and $\mathbf{O}_N$ denote the $N \times N$ identity and all-zero matrices, respectively. Also, we let $\mathbf{1}_N$ and $\mathbf{0}_N$ denote the N-dimensional all-one and all-zero vectors, respectively. We will often omit "N" for brevity if the dimension is clear from the context. We use $\|\cdot\|$ and $\|\cdot\|_1$ to denote L^2- and L^1-norms, respectively.

Network model: In the single-hop case, we will base our discussions on the cellular downlink system, as shown in Fig. 1. We assume that time is slotted and indexed by $t \in \{0, 1, 2, \ldots\}$. The channel between the base station and the receivers can be characterized by a total of M states and denoted by a matrix $\mathbf{\Pi} = [\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_M] \in \mathbb{R}^{N \times M}$, where each column vector $\boldsymbol{\pi}_m \in \mathbb{R}^N$ corresponds to the N links' channel qualities under state m. For each $\boldsymbol{\pi}_m$, we let $\mathcal{C}_{\boldsymbol{\pi}_m}$ denote the achievable rate region, which is defined as the convex hull of the feasible scheduling rate vectors, i.e., $\mathcal{C}_{\boldsymbol{\pi}_m} \triangleq Conv\{x_1^{(m)}, \ldots, x_N^{(m)}\}$, where $Conv\{\cdot\}$ represents the convex hull operation and $x_n^{(m)}$ denotes a feasible rate of link n that can be scheduled under channel state m. We assume that, for each link n and channel state m, the feasible rates satisfy $x_n^{(m)} \leq s^{\max} < \infty$. We use a vector $\mathbf{x}^{(m)} = [x_1^{(m)}, \ldots, x_N^{(m)}]^\top \in \mathbb{R}^N$ to denote the feasible rates under channel state m. We assume that the channel state process is independent and identically distributed in each time slot[1]. We let $\boldsymbol{\pi}[t]$ denote the channel state vector in time-slot t and let $p_m \triangleq \Pr\{\boldsymbol{\pi}[t] = \boldsymbol{\pi}_m\}$ be the stationary distribution of the channel state process being in state m. We let $\bar{\mathcal{C}}$ represent the mean achievable rate region, which can be computed as:

$$\bar{\mathcal{C}} \triangleq \left\{ \mathbf{x} \middle| \mathbf{x} = \sum_{m=1}^{M} p_m \mathbf{x}^{(m)},\ \forall \mathbf{x}^{(m)} \in \mathcal{C}_{\boldsymbol{\pi}_m} \right\}.$$

Note that, in this paper, neither the channel state statistics nor $\bar{\mathcal{C}}$ is assumed to be known at the base station.

Queue-stability: In each time-slot t, the controller observes the current channel state $\boldsymbol{\pi}[t] \in \mathbf{\Pi}$ and then chooses a service rate vector $\mathbf{s}[t] \triangleq [s_1[t], \ldots, s_N[t]]^\top \in \mathcal{C}_{\boldsymbol{\pi}[t]}$ and a congestion controlled rate vector $\mathbf{a}[t] \triangleq [a_1[t], \ldots, a_N[t]]^\top \in \mathbb{R}_+^N$. We assume that each link n is associated with a queue, whose queue-length in time-slot t is denoted as $q_n[t]$. Then, the queue-lengths evolve as:

$$q_n[t+1] = (q_n[t] - s_n[t] + a_n[t])^+, \quad \forall n, \tag{1}$$

where $(\cdot)^+ \triangleq \max\{0, \cdot\}$. Let $\mathbf{q}[t] \triangleq [q_1[t], \ldots, q_N[t]]^\top$ be the queue-length vector in time-slot t. In this paper, we adopt the following notion of queue-stability (same as in [2], [3]): a network is said to be *stable* if the steady-state total queue-length is finite, i.e.,

$$\limsup_{t \to \infty} \mathbb{E}\{\|\mathbf{q}[t]\|_1\} < \infty \tag{2}$$

Problem formulation: Let $\bar{a}_n \triangleq \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} a_n[t]$ denote the long-term average controlled arrival rate of link n. Each link n is associated with a utility function $U_n(\bar{a}_n)$, representing the utility gained by link n when data is injected at rate $\bar{a}_n$. We assume that $U_n(\cdot)$, $\forall n$, is strictly concave, increasing, and twice continuously differentiable. We further assume that $U_n(\cdot)$ satisfies the following strong concavity condition: there exist constants $0 < \phi \leq \Phi < \infty$ such that $\phi \leq -U_n''(a_n) \leq \Phi$, $\forall a_n \in [0, a^{\max}]$, where $a^{\max}$ is the maximum arrival rates. For example, the function $\log(\epsilon + a_n)$ with some constant $\epsilon > 0$ is strongly concave. In this paper, our goal is to maximize $\sum_{n=1}^N U_n(\bar{a}_n)$, subject to achievable rate region $\mathcal{C}_{\boldsymbol{\pi}[t]}$ in each time-slot and the queue-stability constraint. Putting together the models presented above yields the following joint congestion control and scheduling (JCCS) optimization problem:

$$\textbf{JCCS: Max} \quad \sum_{n=1}^{N} U_n(\bar{a}_n)$$

$$\text{s.t. Queue-stabiltity in (2)}, s_n[t] \in \mathcal{C}_{\boldsymbol{\pi}[t]}, a_n[t] \in [0, a^{\max}], \forall n, t.$$

[1]Following the same arguments such as those in [9], our results can be readily generalized to Markov channel state processes.

IV. Heavy-Ball-Based Network Utility Optimization

In this section, we first present our heavy-ball-based network utility optimization algorithm and the main theoretical results in Section IV-A and Section IV-B, respectively. Then, in Section IV-C, we will discuss some key insights and intuition of the theoretical results. Section IV-D focuses on performance analysis and provides the proofs for the main theorems.

A. The Algorithm

Our heavy-ball-based network utility optimization algorithm is described in Algorithm 1:

Algorithm 1: The Heavy-Ball-Based Wireless Network Utility Optimization Algorithm.

Initialization:

1. Choose parameters $K > 0$ and $\beta \in [0, 1)$. Set $t = 0$.
2. Let all queues be empty at the initial state: $q_n[0] = 0$, $\forall n$.
3. Under a given K, associate each link n with a non-negative weight $w_{(K),n}$ and set $w_{(K),n}[0] = w_{(K),n}[-1] = 0$, $\forall n$.

Main Loop:

4. *MaxWeight Scheduler:* In time-slot $t \geq 0$, given the current weight vector $\mathbf{w}_{(K)}[t] \triangleq [w_{(K),1}[t], \ldots, w_{(K),N}[t]]^\top$ and the current channel state $\boldsymbol{\pi}[t]$, the scheduler chooses a service rate vector $\mathbf{s}[t]$ as follows (breaking ties arbitrarily):

$$\mathbf{s}[t] = \arg \max_{\mathbf{x} \in \mathcal{C}_{\boldsymbol{\pi}[t]}} (\mathbf{w}_{(K)}[t])^\top \mathbf{x}. \quad (3)$$

5. *Congestion Controller:* For each link n, given its current weight $w_n[t]$, the data injection rate $a_n[t]$ is an integer-valued random variable that satisfies:

$$\mathbb{E}\{a_n[t]|w_{(K),n}[t]\} = \min\left\{U_n'^{-1}\left(\frac{w_{(K),n}[t]}{K}\right), a^{\max}\right\}, \quad (4)$$

$$\mathbb{E}\{a_n^2[t]|w_{(K),n}[t]\} \leq A < \infty, \quad \forall w_{(K),n}[t], \quad (5)$$

where $U_n'^{-1}(\cdot)$ represents the inverse function of the first-order derivative of $U_n(\cdot)$. In (4) and (5), $a^{\max}$ and A are some predefined sufficiently large positive constants.

6. *Queue-Length and Heavy-Ball Weight Updates:* Update the queue-lengths following (1). Let $\Delta q_n[t] \triangleq q_n[t+1] - q_n[t]$ be the resultant queue-length change, $\forall n$. Next, update the weights in the following (projected) ***heavy-ball*** fashion:

$$w_{(K),n}[t+1] = \big[w_{(K),n}[t] + \Delta q_n[t] + \beta(w_{(K),n}[t] - w_{(K),n}[t-1])\big]^+, \ \forall n. \quad (6)$$

Let $t = t + 1$. Go to Step 4 and repeat the scheduling and congestion control processes.

In Algorithm 1, we can see that the congestion control and scheduling components are similar to those in the QLA schemes (see, e.g., [2], [3]), but with the following *key* differences: First, in both components, the weights in (3) and (4) are *not* directly using current queue-lengths (or some direct functions of current queue-lengths). It is this *separation* of weights and queue-lengths that leads to significant delay reductions. Also, we note that the weight update in (6) is motivated by the *heavy-ball* idea: It includes a β-parameterized *first-order memory* (or called "momentum") of the *weight change* in the previous time-slot. In contrast, the weight updates in traditional QLA algorithms are of *zero-order memory* in the sense that queue-lengths only inherit the absolute weight values in the previous time-slot. We note that this algorithmic structural difference necessitates new proof techniques in establishing the theoretical results. Also, the choices of K and β will be discussed in detail in Section IV-C.

B. Main Results

The first key result in this paper is on the delay reduction performance of our proposed heavy-ball algorithm:

Theorem 1 (Delay reduction and queue-stability)**.** *Under the β-parameterized heavy-ball algorithm, the scaling of the steady-state total queue-length with respect to K satisfies:*

$$\limsup_{t\to\infty} \mathbb{E}\{\|\mathbf{q}[t]\|_1\} = O\big((1-\beta)K\big) + O\big((1+\beta)\sqrt{K}\big). \quad (7)$$

Further, if β approaches 1 in such a way that $\beta = 1 - O(\frac{1}{\sqrt{K}})$, then Eq. (7) implies that $\limsup_{t\to\infty} \mathbb{E}\{\|\mathbf{q}[t]\|_1\} = O(\sqrt{K})$.

Three remarks on Theorem 1 are in order: i) If β is fixed and $K \to \infty$, the first term on the right-hand-side of (7) dominates the second term and thus $\limsup_{t\to\infty} \mathbb{E}\{\|\mathbf{q}[t]\|_1\} \approx O\big((1-\beta)K\big)$. Recall that in a K-parameterized QLA algorithm (see, e.g., [3], [4]), the total queue-length scales as $O(K)+O(\sqrt{K})$. This means that a β-parameterized heavy-ball scheme leads to a delay that is approximately $(1-\beta)$*–fraction* of that of the traditional QLA methods; ii) If β is varying in relation to K, then Theorem 1 states that if $\beta \uparrow 1$ fast enough as $K \to \infty$, the total queue-length scales as $O(\sqrt{K})$, which significantly outperforms the $O(K)$ delay of the QLA algorithms. We note that this $O(\sqrt{K})$ delay is achieved *without* sacrificing any throughput and *without* requiring non-causal global statistics as in [8], [9]; iii) In some sense, including the weight changes in (6) can be viewed as a simple way of "learning" how the queues had evolved in history. Interestingly, Theorem 1 shows that even simply paying attention to "*yesterday's memory*" makes a big difference in delay performance.

Now, let $U(\mathbf{a}) = \sum_{n=1}^N U_n(a_n)$ be the total utility function of Problem JCCS and let $\mathbf{a}^*$ be the optimal solution. Also, let $a_{(K),n}^\infty \triangleq \mathbb{E}\{\min\{U_n'^{-1}(w_{(K),n}^\infty/K), a^{\max}\}\}$, $\forall n$, be the mean steady-state congestion control rates offered by our heavy-ball algorithm (the existence of steady-state will be proved in Section IV-D). Further, we let $\mathbf{a}_{(K)}^\infty \triangleq [a_{(K),1}^\infty, \ldots, a_{(K),N}^\infty]^\top$. Then, the next result states that our proposed heavy-ball algorithm is *utility-optimal*:

Theorem 2 (Utility-optimality)**.** *Under Algorithm 1 and for some given K, the mean of the stationary rate vector $\mathbf{a}_{(K)}^\infty$ satisfies $\|\mathbf{a}_{(K)}^\infty - \mathbf{a}^*\| = O(1/\sqrt{K})$. Also, the optimal utility objective value can be bounded as $U\big(\mathbf{a}_{(K)}^\infty\big) \geq U(\mathbf{a}^*) - O(1/K)$. Hence, $\mathbf{a}_{(K)}^\infty$ converges to $\mathbf{a}^*$ asymptotically as K increases.*

We note that the utility-optimality results stated in Theorem 2 are *independent* of β, and the optimality gap scaling

results are same as those of the QLA schemes (see, e.g., [3], [4]). This shows a salient feature of our proposed heavy-ball approach: Although we have introduced the heavy-ball-based weight updates in (6), such an algorithmic change does *not* affect the utility-optimality of the original QLA framework.

Our third result is on the convergence speed performance. In this paper, the notion of convergence speed is defined in terms of the fewest number of time-slots that the sequence $\{\mathbf{w}_{(K)}[t]\}$ takes so that the resultant sequence $\{\mathbb{E}\{\mathbf{a}_{(K)}[t]|\mathbf{w}_{(K)}[t]\}\}$ reaches the $O(\frac{1}{\sqrt{K}})$-neighborhood of $\mathbf{a}^*$ stated in Theorem 2.

Theorem 3 (Linear convergence rate)**.** *Let K and β be chosen as $K \in (\frac{\Phi}{4}, \infty]$ and $\beta \in \big[\max\big\{0, \frac{\Phi}{2K} - 1\big\}, 1\big)$. Then, $\{\mathbb{E}\{\mathbf{a}_{(K)}[t]|\mathbf{w}_{(K)}[t]\}\}$ converges linearly[2] with a factor $R_{(K,\beta)} \leq \max\big\{\sqrt{\beta}, |1+\beta-\phi/K| - \sqrt{\beta}, |1+\beta-\Phi/K| - \sqrt{\beta}\big\}$. Moreover, minimizing the upper-bound of $R_{(K,\beta)}$ yields: $R^* = (\sqrt{\Phi}-\sqrt{\phi})/(\sqrt{\Phi}+\sqrt{\phi})$, which is obtained by $K^* = (\sqrt{\Phi}+\sqrt{\phi})^2/4$ and $\beta^* = (\sqrt{\Phi}-\sqrt{\phi})^2/(\sqrt{\Phi}+\sqrt{\phi})^2$.*

Theorem 3 says that we can optimize K and β to achieve $R^* = (\sqrt{\kappa}-1)/(\sqrt{\kappa}+1)$, where $\kappa \triangleq \Phi/\phi$ is the condition number [14]. The optimized R^* is always *smaller* compared to that of the QLA approaches, where $R_{\rm QLA} = (\kappa-1)/(\kappa+1)$ (cf. e.g., [2]), thus implying a faster convergence. Moreover, this convergence speedup phenomenon is even more pronounced when κ is large (i.e., the problem is ill-conditioned).

The proofs of Theorems 1–3 will be provided in Section IV-D. In what follows, we will further discuss an important *three-way performance trade-off* implied by the theoretical results in Theorems 1–3.

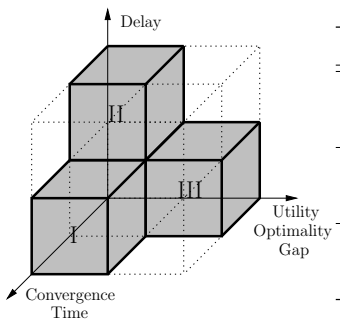
C. A Three-Way Performance Trade-off

Collectively, Theorems 1–3 suggest a *new* three-way trade-off relationship where, by appropriately selecting K and β, one can *simultaneously improve two out of the three performance metrics by trading-off the third*. To facilitate better understanding, we illustrate this three-way trade-off relationship in Fig. 2. In Fig. 2, the arrow of each axis is pointing toward worse performance in utility, delay, and convergence, respectively. The regions I, II, and III represent three types of trade-off relationships achieved under our heavy-ball algorithm, and the table in Fig. 2 illustrates how each region corresponds to the settings of the two control knobs K and β.

First, Region I in Fig. 2 represents "*achieving both utility-optimality and low-delay by setting a large K and choosing β close to 1, at the cost of slower convergence*." To see this, we first note from Theorem 2 that a large K implies small utility-optimality gap $O(1/K)$. Also, by choosing β close to 1, Theorem 1 implies that the $(1-\beta)$–fraction delay reduction is significant. However, when $K \to \infty$ and $\beta \to 1$, it is not difficult to verify from Theorem 3 that:

$$\lim_{\substack{\beta\to 1\\ K\to\infty}} R_{(K,\beta)} \leq \lim_{\substack{\beta\to 1\\ K\to\infty}} \Big\{ \max\big\{\sqrt{\beta}, |1+\beta-\phi/K| - \sqrt{\beta}, |1+\beta-\Phi/K| - \sqrt{\beta}\big\}\Big\} \to 1.$$

[2]We say that a sequence $\{x_k\}_{k=1}^{\infty}$ converges linearly to x^* if there exists a factor $R \in (0,1)$ such that $\|x_{k+1} - x^*\| \leq R\|x_k - x^*\|$ for all k.



	K	β
I	Large	Close to 1
II	Large	Optimized for Rate of Convergence
III	Optimized for Rate of Convergence	Close to 1

Fig. 2. An illustration of the three-way trade-off relationships.

That is, as $K \to \infty$ and $\beta \to 1$, the worst case convergence rate factor $R_{(K,\beta)}$ asymptotically approaches 1, which implies an increasingly slower convergence speed.

Second, Region II represents "*achieving utility-optimality and fast-convergence by setting a large K and optimizing β, at the cost of less delay performance gain.*" To see this, we again note from Theorem 2 that a large K implies small utility-optimality gap $O(1/K)$. Also, by Theorem 3, we can optimize β under a given K to minimize the convergence factor $R_{(K,\beta)}$ to increase the convergence speed. However, the obtained β is not necessarily close to 1 and thus the delay performance gain may not be dramatic. We note that, in Region II, even though the optimized β may not entail dramatic delay reduction, one still enjoys the benefit of $(1-\beta)$–fraction delay compared to the QLA approaches, according to Theorem 1.

Lastly, Region III represents "*achieving low-delay and fast convergence by setting β close to 1 and optimizing K, at the cost of larger utility-optimality gap.*" To see this, we note from Theorem 1 that we can first push β close to 1 to achieve low delay. With the given β, by Theorem 3, we can optimize K to minimize the convergence factor $R_{(K,\beta)}$ to increase the convergence speed. However, the obtained K is not necessarily large and thus the utility-optimality gap may not be small.

D. Proofs of the Main Theorems

In this subsection, due to space limitation, we provide sketched proofs for the theorems in Section IV-B and relegate the remaining proof details to our online technical report [20].

Sketch of the proof of Theorem 1. The key steps for proving Theorem 1 are as follows. First, we consider a K-parameterized deterministic version of Problem JCCS (see Problem K-DJCCS in [20]), where the channel state process is not random but fixed at its mean (i.e., the achievable rate region is $\bar{\mathcal{C}}$), and the objective function is changed to $K\sum_{n=1}^{N} U_n(a_n)$. Then, it is easy to show that its optimal dual solution $\mathbf{w}^*_{(K)}$ scales as $O(K)$ (cf. [20, Lemma 1]). Our next key step toward proving Theorem 1 is to establish the following mean weight deviation bound [20, Theorem 2]:

Proposition 4 (Mean weight deviation bound [20])**.** *Under Algorithm 1 and a given K, there exists a constant C that depends on Φ, $s^{\max}$, and $a^{\max}$, such that $\mathbb{E}\{\|\mathbf{w}^{\infty}_{(K)} - \mathbf{w}^*_{(K)}\|\} \leq C\sqrt{K}$, where $\mathbf{w}^{\infty}_{(K)}$ denotes the weights $\mathbf{w}_K[t]$ under parameter K in steady-state.*

To show Proposition 4, we note that the heavy-ball update in (6) can be rewritten as (see [20, Eq.(17)–(18)] for details):

$$\mathbf{w}_{(K)}[t+1]-\mathbf{w}^*_{(K)}=\mathbf{w}_{(K)}[t]-\mathbf{w}^*_{(K)} + \big(\mathbf{a}[t]-\mathbf{s}[t]+\mathbf{u}[t]\big)+\beta\big(\mathbf{w}_{(K)}[t]-\mathbf{w}_{(K)}[t-1]\big), \quad (8)$$

where $\mathbf{u}[t] \geq \mathbf{0}$ is some projection term. Note that since the momentum term in (8) depends on two consecutive time-slots of memory $\mathbf{w}_{(K)}[t]$ and $\mathbf{w}_{(K)}[t-1]$, traditional techniques used in establishing similar mean dual distance bounds (see, e.g., [4], [9]) cannot be directly applied. To overcome this challenge, we define a $2N$-dimensional vector $\mathbf{z}[t] \triangleq [(\mathbf{w}_{(K)}[t]-\mathbf{w}^*_{(K)})^\top, (\mathbf{w}_{(K)}[t-1]-\mathbf{w}^*_{(K)})^\top]^\top$ and a special block-structured matrix $\mathbf{\Gamma} \in \mathbb{R}^{2N\times 2N}$ as follows:

$$\mathbf{\Gamma} \triangleq \begin{bmatrix} (1+\beta)\mathbf{I}_N & -\beta\mathbf{I}_N \\ \mathbf{I}_N & \mathbf{O}_N \end{bmatrix}.$$

Then, it can be readily verified that (8) can be rewritten in terms of $\mathbf{z}[t]$ as follows: $\mathbf{z}[t+1] = \mathbf{\Gamma}\mathbf{z}[t] + \Delta\tilde{\mathbf{q}}[t]$, where $\Delta\tilde{\mathbf{q}}[t] \triangleq [(\mathbf{a}[t]-\mathbf{s}[t]+\mathbf{u}[t])^\top, \mathbf{0}_N^\top]^\top$. Now, consider the following quadratic Lyapunov function $V(\mathbf{z}[t]) \triangleq \frac{1}{2}\|\mathbf{z}[t]\|^2$ and evaluate the one-slot conditional expected Lyapunov drift of $V(\mathbf{z}[t])$: $\mathbb{E}\big\{\Delta V(\mathbf{z}[t])\big|\mathbf{z}[t]\big\} \triangleq \frac{1}{2}\mathbb{E}\big\{\|\mathbf{z}[t+1]\|^2-\|\mathbf{z}[t]\|^2\big|\mathbf{z}[t]\big\}$. Let $\mathbb{1}_{\mathcal{A}}(\mathbf{x})$ be the indicator function that takes value 1 if $\mathbf{x} \in \mathcal{A}$ and 0 otherwise. After showing that $\mathbf{\Gamma}$ is a *non-expansive* linear transformation (cf. [20, Lemma 2]) and some algebraic derivations, we arrive at the following result (see [20, Appendix A] for proof details):

Proposition 5. *Let* $\mathbf{w}$ *be the first* N *entries in* $\mathbf{z}[t]=\mathbf{z}$. *Let* $B \triangleq \frac{N}{2}[A+(s^{\max})^2]$. *There exist constants* $\delta,\eta>0$ *such that*

$$\mathbb{E}\{\Delta V(\mathbf{z}[t])|\mathbf{z}[t]=\mathbf{z}\} \leq -\frac{\delta}{\sqrt{K}}\big\|\mathbf{w}-\mathbf{w}^*_{(K)}\big\|\mathbb{1}_{\mathcal{B}^c}(\mathbf{w})+\eta\mathbb{1}_{\mathcal{B}}(\mathbf{w}),$$

where $\mathcal{B} \triangleq \{\mathbf{w}: \big\|\mathbf{w}-\mathbf{w}^*_{(K)}\big\| \leq \sqrt{B\Phi K}\}$, *and* $\mathcal{B}^c$ *denotes the complement of* $\mathcal{B}$.

Note that $\{\mathbf{z}[t]\}$ is a continuous state Markov chain in $\mathbb{R}^{2N}$ and Proposition 5 assures the Foster-Lyapunov criterion for positive Harris-recurrence. Hence, a steady-state exists [21]. Next, we define a set $\Omega \triangleq \{\mathbf{z}\in\mathbb{R}^{2N}: (\mathbf{z})_{1:N} \in \mathcal{B}\}$, where $(\mathbf{z})_{1:N}$ denotes the first N entries in $\mathbf{z}$. Then, telescoping the inequality in Proposition 5 and after some derivations (see [20, Eq. (24)]), we can show that $0 \leq -\frac{\delta}{\sqrt{K}}\int_{\Omega^c} p^\infty_{\mathbf{z}}\|\mathbf{w}-\mathbf{w}^*_{(K)}\|d\mathbf{z} + \eta\int_\Omega p^\infty_{\mathbf{z}} d\mathbf{z}$, where $p^\infty_{\mathbf{z}}$ denotes the stationary distribution of the continuous state Markov chain $\{\mathbf{z}[t]\}$. Lastly, rearranging terms, adding $\frac{\delta}{\sqrt{K}}\int_\Omega p^\infty_{\mathbf{z}}\|\mathbf{w}-\mathbf{w}^*_{(K)}\|$, and multiplying both sides by $\frac{\sqrt{K}}{\delta}$ yields $\mathbb{E}\{\|\mathbf{w}^\infty_{(K)}-\mathbf{w}^*_{(K)}\|\} \leq (\frac{\eta}{\delta}+\sqrt{B\Phi})\sqrt{K} = O(\sqrt{K})$, i.e., the result in Proposition 4.

To finish the proof of Theorem 1, we re-write the heavy-ball weight update in the following form: $\mathbf{w}_{(K)}[t+1]=\mathbf{w}_{(K)}[t]+\Delta\mathbf{q}[t]+\beta(\mathbf{w}_{(K)}[t]-\mathbf{w}_{(K)}[t-1])+\mathbf{u}^{(2)}[t]$, where $\mathbf{u}^{(2)}[t] \geq \mathbf{0}$ is a projection term (see [20, Eq. (17)]). Rearranging terms yields:

$$\Delta\mathbf{q}[t] \leq \big(\mathbf{w}_{(K)}[t+1]-\mathbf{w}_{(K)}[t]\big)-\beta\big(\mathbf{w}_{(K)}[t]-\mathbf{w}_{(K)}[t-1]\big). \quad (9)$$

Telescoping the inequality in (9) from $t=0$ to $T-1$ yields: $\sum_{t=0}^{T-1}\Delta\mathbf{q}[t] \leq \big(\mathbf{w}_{(K)}[T]-\mathbf{w}_{(K)}[0]\big)-\beta\big(\mathbf{w}_{(K)}[T-1]-\mathbf{w}_{(K)}[-1]\big) = \mathbf{w}_{(K)}[T]-\beta\mathbf{w}_{(K)}[T-1]$, where the last equality follows from the fact that $\mathbf{w}_{(K)}[0]=\mathbf{w}_{(K)}[-1]=\mathbf{0}$. Also, since $\mathbf{q}[0]=\mathbf{0}$, we have $\|\mathbf{q}[T]\|_1 = \|\mathbf{q}[0]+\sum_{t=0}^{T-1}\Delta\mathbf{q}[t]\|_1 \leq \|\mathbf{w}_{(K)}[T]-\beta\mathbf{w}_{(K)}[T-1]\|_1$. Taking expectation on both sides, letting $T\to\infty$, and taking limits yields: $\limsup_{T\to\infty}\mathbb{E}\{\|\mathbf{q}[T]\|_1\} \leq \mathbb{E}\{\mathbf{w}^\infty_{(K)}-\beta\mathbf{w}^\infty_{(K)}\} \leq \mathbf{w}^*_{(K)} + O(\sqrt{K})-\beta(\mathbf{w}^*_{(K)}-O(\sqrt{K}))=O((1-\beta)K)+O((1+\beta)\sqrt{K})$, where the first inequality follows from Proposition 4 and $\|\cdot\|_1 \leq \sqrt{N}\|\cdot\|$; the second equality follows from $\mathbf{w}^*_{(K)} = O(K)$ (cf. [20, Lemma 1]). Moreover, when $\beta = 1-O(\frac{1}{\sqrt{K}})$, it follows from (7) that $\limsup_{T\to\infty}\mathbb{E}\big\{\|\mathbf{q}[t]\|_1\big\} \approx O(\sqrt{K})$, i.e., the delay grows as $O(\sqrt{K})$. This completes the proof. □

Sketch of the proof of Theorem 2. We first prove the optimality gap result for $a^\infty_{(K),n} \triangleq \mathbb{E}\{\min\{U_n^{'-1}(\frac{w^\infty_{(K),n}}{K}), a^{\max}\}\}$. Note that $a^*_n = U_n^{'-1}(\frac{w^*_{(K),n}}{K})$, $\forall n$. Thus, plugging in these definitions in $\big\|\mathbf{a}^\infty_{(K)}-\mathbf{a}^*\big\|^2$ and then upper-bounding by using Jensen's inequality, mean value theorem, and inverse function lemma, we obtain (see [20, Eq. (31)] for detailed derivations): $\|\mathbf{a}^\infty_{(K)}-\mathbf{a}^*\|^2 \leq \frac{1}{\phi^2K^2}\mathbb{E}\{\|\mathbf{w}^\infty_{(K)}-\mathbf{w}^*_{(K)}\|^2\}$. Now, consider the term $\mathbb{E}\{\|\mathbf{w}^\infty_{(K)}-\mathbf{w}^*_{(K)}\|^2\}$. From the proof of Proposition 5, we have the following one-slot mean Lyapunov drift bound (cf. [20, Appendix A, Eq. (51)]:

$$\mathbb{E}\{\Delta V(\mathbf{z}[t])|\mathbf{z}[t]\} \leq -\frac{1}{\Phi K}\big\|\mathbf{w}_{(K)}[t]-\mathbf{w}^*_{(K)}\big\|^2 + B. \quad (10)$$

Following the same steps in the proof of Proposition 4, we telescope (10) from $t=0$ to $T-1$ to obtain: $\mathbb{E}\{V(\mathbf{z}[T])|\mathbf{z}[0]\} - V(\mathbf{z}[0]) \leq -\frac{1}{\Phi K}\sum_{t=0}^{T-1}\int_{\mathbb{R}^{2N}} p_{\mathbf{z}[t]|\mathbf{z}[0]}(\mathbf{z})\big\|\mathbf{w}-\mathbf{w}^*_{(K)}\big\|^2 d\mathbf{z} + TB$. Dividing both sides by $\frac{T}{\Phi K}$, rearranging terms, and letting $T\to\infty$, we have $\limsup_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\int_{\mathbb{R}^{2N}} p_{\mathbf{z}[t]|\mathbf{z}[0]}(\mathbf{z})\big\|\mathbf{w}-\mathbf{w}^*_{(K)}\big\|^2 d\mathbf{z} \leq B\Phi K$. Note here that the left-hand-side is precisely $\mathbb{E}\big\{\|\mathbf{w}^\infty_{(K)}-\mathbf{w}^*_{(K)}\|^2\big\}$. Hence, we have

$$\|\mathbf{a}^\infty_{(K)}-\mathbf{a}^*\|^2 \leq \frac{1}{\phi^2K^2}\mathbb{E}\big\{\|\mathbf{w}^\infty_{(K)}-\mathbf{w}^*_{(K)}\|^2\big\} \leq \frac{B\Phi}{\phi^2}\frac{1}{K}. \quad (11)$$

Taking square root on both sides of (11) yields $\|\mathbf{a}^\infty_{(K)}-\mathbf{a}^*\| = O(\frac{1}{\sqrt{K}})$ and the proof of the first half is complete.

To prove that $U\big(\mathbf{a}^\infty_{(K)}\big) \geq U(\mathbf{a}^*) - O(1/K)$, similar to the proof of Proposition 4, we define an augmented vector $\mathbf{y}[t] \triangleq [\mathbf{w}^\top_{(K)}[t], \mathbf{w}^\top_{(K)}[t-1]]^\top$ and a quadratic Lyapunov function $L(\mathbf{y}[t]) = \frac{1}{2}\|\mathbf{y}[t]\|^2$. Following the same steps as in the proof of Proposition 4, one can verify that $\mathbf{y}[t+1] = \mathbf{\Gamma}\mathbf{y}[t] + \Delta\tilde{\mathbf{q}}[t]$. Then, following the same argument as in the proof of Proposition 5, we can show that the one-slot conditional expected Lyapunov drift can be upper-bounded as $\mathbb{E}\big\{\Delta L(\mathbf{y}[t])\big|\mathbf{y}[t]\big\} \leq -\mathbf{w}^\top_{(K)}[t]\mathbb{E}\{\mathbf{a}[t]-\mathbf{s}[t]\big|\mathbf{y}[t]\big\}+B$. Note that the right-hand-side is in the same form as in [3, Eq. (24)]. Thus, the rest of the proof follows from the same arguments in [3] and the proof is complete. □

Sketch of the proof of Theorem 3. Because of the one-to-one mapping between $\mathbb{E}\{\mathbf{a}_{(K)}[t]|\mathbf{w}_{(K)}[t]\}$ and $\mathbf{w}_{(K)}[t]$, the convergence of $\{\mathbb{E}\{\mathbf{a}_{(K)}[t]|\mathbf{w}_{(K)}[t]\}\}$ can be equivalently analyzed by examining $\{\mathbf{w}_{(K)}[t]\}$. Note that (6) can be written as:

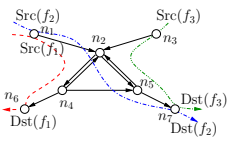


Fig. 3. A multi-hop wireless network.

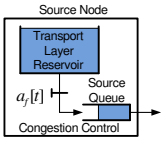


Fig. 4. Congestion control at source node.

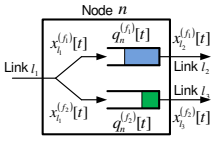


Fig. 5. Routing at intermediate node.

$\mathbf{w}_{(K)}[t+1] \leq \mathbf{w}_{(K)}[t]+(\mathbf{a}[t]-\mathbf{s}[t])+\beta(\mathbf{w}_{(K)}[t]-\mathbf{w}_{(K)}[t-1])$. Dividing both sides by K (scaling does not affect convergence), we have: $\mathbf{w}_{(1)}[t+1] \leq \mathbf{w}_{(1)}[t] + \frac{1}{K}(\mathbf{a}[t] - \mathbf{s}[t]) + \beta(\mathbf{w}_{(1)}[t] - \mathbf{w}_{(1)}[t-1])$. Note that the right-hand-side is the same as the standard unconstrained heavy-ball method (cf. [14]) with step-size $\frac{1}{K}$. Hence, from [14, Chap. 3.2, Theorem 1], we have the sufficient condition for convergence as: $\frac{1}{K} \in (0, \frac{(1+\beta)}{\Phi}]$, and $\beta \in [0,1)$. After some manipulations and noting that $\beta > 0$, we arrive at $K \in (\frac{\Phi}{4}, \infty]$ and $\beta \in [\max\{0, \frac{\Phi}{2K}-1\}, 1)$, i.e., the result stated in Theorem 3. Also, the convergence factor upper-bound in Theorem 3 follows directly from [19, Theorem 1]. This completes the proof. □

V. Extension to Multi-Hop Networks

In this section, we will generalize our heavy-ball algorithmic framework to multi-hop wireless networks. In the multi-hop setting, the utility optimization problem becomes the joint congestion control and routing optimization as in [1]–[3]. Here, we first state the network model and problem formulation.

Network model and problem formulation: *1) Congestion control:* Consider an N-node L-link multi-hop wireless network system as illustrated in Fig. 3. There are F end-to-end flows in the network. The source and destination nodes of each flow f are denoted by $\mathrm{Src}(f)$ and $\mathrm{Dst}(f)$, respectively. As in [1]–[3], node $\mathrm{Src}(f)$ has a continuously-backlogged transport layer reservoir that contains flow f's data, as shown in Fig. 4. In each time-slot t, the congestion controller determines the amount of data $a_f[t] \in [0, a_f^{\max}]$ to be released from the reservoir into a network layer source queue, where the data awaits to be sent to $\mathrm{Dst}(f)$. We let $\bar{a}_f = \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} a_f[t]$ be the time-average rate at which flow f is injected at $\mathrm{Src}(f)$. Similar to the single-hop case, each flow is associated with a strongly concave, increasing, and twice continuously differentiable utility function $U_f(\bar{a}_f)$.

2) Multi-hop routing: We let $x_l^{(f)}[t] \geq 0$ denote the rate offered to route flow f's data in time-slot t at link l, as shown in Fig. 5. We let $\bar{x}_l^{(f)} \triangleq \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} x_l^{(f)}[t]$ represent the time-average service rate of flow f at link l. The channel state process model remains the same as in the single-hop case so that we have $x_l^{(f)}[t] \in \mathcal{C}_{\boldsymbol{\pi}[t]}, \forall l, f, t$. As in [1]–[3], the following routing constraints need to be satisfied:

$$\sum_{l\in\mathcal{O}(n)} \bar{x}_l^{(f)} \geq \sum_{l\in\mathcal{I}(n)} \bar{x}_l^{(f)} + \bar{a}_f \mathbb{1}_f(n), \ \forall f, \forall n \neq \mathrm{Dst}(f), \quad (12)$$

where $\mathcal{O}(n)$ and $\mathcal{I}(n)$ represent the sets of outgoing and incoming links at node n, respectively; $\mathbb{1}_f(n)$ is an indicator function that takes the value 1 if $n = \mathrm{Src}(f)$ and 0 otherwise.

3) Queue-stability: We assume that each node maintains a separate queue for each flow f, as shown in Fig. 5. We let $q_n^{(f)}[t] \geq 0$ denote the queue-length of flow f at node n at time t. Then, the queue-length evolution can be written as:

$$q_n^{(f)}[t+1] = \left(q_n^{(f)}[t] - \sum_{l\in\mathcal{O}(n)} x_l^{(f)}[t]\right)^+ + \sum_{l\in\mathcal{I}(n)} \widehat{x}_l^{(f)}[t] + a_f[t]\mathbb{1}_f(n), \quad (13)$$

where $\widehat{x}_l^{(f)}[t]$ is the *actual* routing rate. Note that $\widehat{x}_l^{(f)}[t] \leq x_l^{(f)}[t]$ since the transmitter node of link $l \in \mathcal{I}(n)$ may have fewer than $x_l^{(f)}[t]$ amount of packets left. Let $\mathbf{q}[t] \triangleq [q_n^{(f)}[t], \forall n, f]$. Similar to the single-hop case, we say that the network is *stable* if the steady-state total queue-length remains finite, i.e.,

$$\limsup_{t\to\infty} \mathbb{E}\{\|\mathbf{q}[t]\|_1\} < \infty. \quad (14)$$

4) Problem formulation: In the multi-hop wireless network case, our goal is to develop an optimal joint congestion control and routing scheme to maximize the total utility $\sum_{f=1}^{F} U_f(\bar{a}_f)$, subject to the network capacity region and network stability constraints. Putting together the models presented earlier yields the following joint congestion control and routing (JCCR) optimization problem:

JCCR: Max $\sum_{f=1}^{F} U_f(\bar{a}_f)$

s.t. Routing constr. (12); Queues stability constraint in (14),

$$x_l^{(f)}[t] \in \mathcal{C}_{\boldsymbol{\pi}[t]}, \ \forall l, t, f, \quad a_f[t] \geq 0, \ \forall f, t.$$

The Algorithm: Similar to the generalization of the QLA schemes to the multi-hop case, in our multi-hop heavy-ball algorithm, the weights are replaced by *weight differentials* to perform dynamic routing. To this end, we let $\mathcal{E}(l)$ denote the two end nodes of link l. The heavy-ball-based joint congestion control and routing algorithm is stated as follows:

Algorithm 2: The Heavy-Ball-Based Joint Congestion Control and Routing Algorithm for Multi-Hop Wireless Networks.

Initialization:

1. Choose parameters $K > 0$ and $\beta \in [0,1)$. Set $t = 0$.
2. Let all queues be empty at the initial state: $q_n^{(f)}[0] = 0, \forall n$.
3. Under a given K, associate each link n with a weight $w_{(K),n}^{(f)} \geq 0$ and set $w_{(K),n}^{(f)}[0] = w_{(K),n}^{(f)}[-1] = 0, \forall n, f$.

Main Loop:

4. *Weight Differentials:* In each time-slot $t \geq 0$, we let $\Delta w_{(K),l}^{(f)}[t] = \max\left\{w_{(K),n}^{(f)}[t] - w_{(K),\mathcal{E}(l)\backslash n}^{(f)}[t], 0\right\}$ denote the *weight differential* of flow f, $\forall n$, $\forall f$, $\forall l \in \mathcal{O}(n)$. Let $\Delta w_{(K),l}^{*}[t] = \max_f \Delta w_{(K),l}^{(f)}[t]$ and let $f_l^*[t] = \arg\max_f \Delta w_{(K),l}^{(f)}[t]$ (breaking ties arbitrarily). Let $\Delta \mathbf{w}_{(K)}^{*}[t] \triangleq [w_{(K),1}^{*}[t], \ldots, w_{(K),L}^{*}[t]]^\top$ be the maximum weight differentials vector over all links.
5. *Routing and MaxWeight Scheduling:* Given $\Delta \mathbf{w}_{(K)}^{*}[t]$ and the channel state $\boldsymbol{\pi}[t]$, the controller schedules a service rate vector $\mathbf{x}[t] \in \mathbb{R}^L$ to route only flow $f_l^*[t]$ at link l, $\forall l$:

$$\mathbf{x}[t] = \arg\max_{\mathbf{x}\in\mathcal{C}_{\boldsymbol{\pi}[t]}} (\Delta \mathbf{w}_{(K)}^{*}[t])^\top \mathbf{x}. \quad (15)$$

6. *Congestion Controller:* For each flow f and in each time-slot t, let w be the value of $w_{(K),\mathrm{Src}(f)}[t]$ that the source

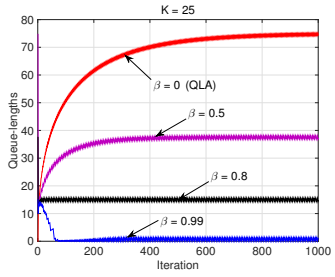


Fig. 6. The impact of β on queueing delay.

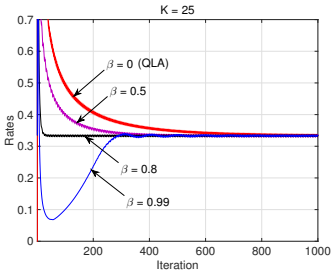


Fig. 7. The impact of β on convergence speed.

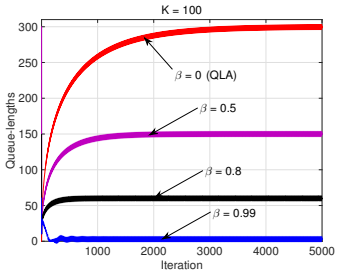


Fig. 8. The impact of K on queueing delay.

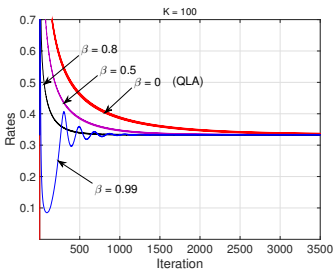


Fig. 9. The impact of K on convergence speed.

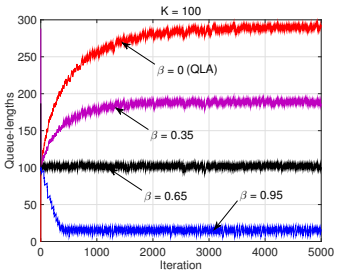


Fig. 10. The impact of β on queueing delay for a 15-user cellular downlink with fading ($K = 100$).

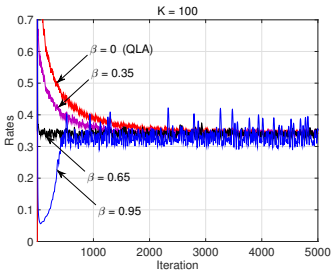


Fig. 11. The impact of β on convergence speed for a 15-user cellular downlink with fading ($K = 100$).

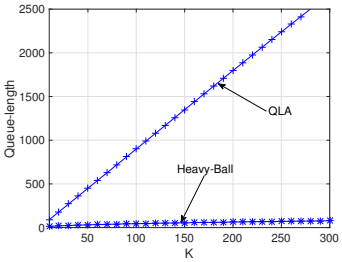


Fig. 12. Steady-state queue-lengths comparisons between QLA and the heavy-ball approach.

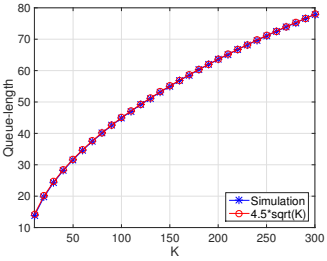


Fig. 13. Zoom-in view of the heavy-ball approach in Fig. 12, which shows the $O(\sqrt{K})$ scaling.

node $\mathrm{Src}(f)$ observes. Then, $\mathrm{Src}(f)$ sets $a_f[t]$ to be an integer-valued random variable that satisfies:

$$\mathbb{E}\{a_f[t]\} = \min\left\{U_f'^{-1}\left(\frac{w}{K}\right), a^{\max}\right\}, \quad (16)$$

$$\mathbb{E}\{a_n^2[t]\} \leq A < \infty, \quad (17)$$

where $U_f'^{-1}(\cdot)$ represents the inverse function of first-order derivative of $U_f(\cdot)$. In (16) and (17), $a^{\max}$ and A are some predefined sufficiently large positive constants.

7. *Queue-Length and Heavy-Ball Weight Updates:* Update the queue-lengths following (13). Let $\Delta q_n^{(f)}[t] \triangleq q_n^{(f)}[t+1] - q_n^{(f)}[t]$ be the resultant queue-length change of flow f at node n, $\forall n, f$. Next, update the weights in the following (projected) ***heavy-ball*** fashion:

$$w_{(K),n}^{(f)}[t+1] = \big[w_{(K),n}^{(f)}[t] + \Delta q_n^{(f)}[t] + \beta\big(w_{(K),n}^{(f)}[t] - w_{(K),n}^{(f)}[t-1]\big)\big]^+, \forall n, f. \quad (18)$$

Let $t = t+1$. Go to Step 4 and repeat the whole dynamic routing, scheduling and congestion control processes.

Distributed Implementation: As in the QLA algorithms, Algorithm 2 only requires weight information locally and from one-hop neighbors. Thus, the congestion control and dynamic routing can be implemented in a *distributed* fashion exactly the same as that in the QLA algorithms and do *not* incur any additional complexity in terms of messaging passing. Also same as in the QLA algorithms, the scheduling component in (15) is challenging for distributed implementations since it requires global weight information. Fortunately, thanks to the same messaging passing requirement, it can be readily verified that all distributed algorithms developed for the QLA framework can be directly applied in the scheduling component in our heavy-ball algorithm, e.g., adaptive CSMA [22], [23], etc.

Performance Analysis: The same utility, delay, and convergence results in Theorems 1–3 continue to hold in the multi-hop case, and their proofs follow similar steps and arguments, but with more complicated notation. Due to space limitation, we omit these results and their proofs for brevity. We refer readers to [20, Section V] for further information.

VI. Numerical Results

In this section, we conduct numerical studies to verify the theoretical results presented in Section IV. To clearly visualize the key insights of our theoretical results and not being blurred by random noises, we first use a three-link non-fading cellular network as an example. We assume that each link has one unit capacity and only one link can be activated in each time-slot. We use $\log(0.001 + a)$ as the utility function for each link, i.e., the proportional fairness metric [5]. Due to the symmetry of the setting, the optimal congestion control rates are $\bar{a}_1^* = \bar{a}_2^* = \bar{a}_3^* = \frac{1}{3}$. To see the impact of β on delay and convergence speed, we fix $K = 25$ and increase β from 0 to 0.99 (note that $\beta = 0$ corresponds to the QLA approach). Because of the symmetry of the setting, we only plot the results of link 1. As shown in Fig. 6, as β increases, the average queue-lengths are 74.6, 37.4, 14.8, and 1.1, respectively, which corroborates the $(1-\beta)$–fraction reduction result in Theorem 1. We can see from Fig. 7 that, for all choices of β, the congestion control rates all converge to the optimal solution, confirming Theorem 2 that utility-optimality is independent of β. However, changing β has a significant impact on the convergence speed. In Fig. 7, as β increases from 0 to 0.99, the convergence speed initially

increases, peaks at $\beta = 0.8$, and then decreases. Interestingly, we note from Fig. 6 and Fig. 7 that, by setting $\beta = 0.99$, both utility-optimality and low-delay can be achieved at the cost of slower convergence speed, hence confirming Theorem 3. Next, we increase K from 25 to 100 and conduct another set of experiments on the same network. The results are shown in Fig. 8 and Fig. 9, respectively. With a larger K, the congestion control rates again converge to the same optimal solution with a smaller variance, but at the cost of larger delay and slower convergence. This again confirms the results in Theorems 1–3.

Next, we test our heavy-ball algorithm in a larger 15-link cellular downlink with a quasi-static block fading (channel states are constant in each time slot and vary from one time-slot to the next). We again assume that only one user can be activated in each time-slot. We fix $K = 100$ and vary β. For clearer visualization, we only plot the results of link 1 in Fig. 10 and Fig. 11. In Fig. 10, as β increases, the queue-lengths also monotonically decrease and follow the $(1-\beta)$-fraction reduction law stated in Theorem 1. In Fig. 11, we can see that the congestion control rates under different choices of β all converge to the same optimal solution. Also, the convergence speed initially increases but eventually decreases as β increases. This again verifies the same three-way trade-off effect in this larger network example with fading.

Lastly, we compare the delay scaling with respect to K under QLA and our heavy-ball algorithm, respectively. Here, as K increases, we let $\beta \uparrow 1$ as $\beta = 1 - \frac{1}{2\sqrt{K}}$. As expected, in Fig. 12, the total queue-length of QLA exhibits the well-known $O(K)$ linear scaling law and is significantly larger than that of our heavy-ball algorithm. Further, from the "zoom-in" view of the heavy-ball results in Fig. 13, we can see that the total queue-length increases as $4.5\sqrt{K}$, which *perfectly* matches the $O(\sqrt{K})$-delay theoretical result stated in Theorem 1.

VII. Conclusion

In this paper, we have developed a new heavy-ball algorithmic framework for network utility optimization in wireless networks. Compared to the traditional queue-length-based algorithms, our proposed heavy-ball algorithmic framework offers not only utility-optimality and queue-stability, but also fast-convergence and low-delay. Our main contributions in this paper are three-fold: i) We have proposed a heavy-ball joint congestion control and scheduling/routing framework that is well-suited for implementation in practice; ii) we have rigorously shown the utility-optimality of the proposed heavy-ball algorithmic framework and characterized the delay reduction and convergence speed performances; and iii) we offered design rules for optimal selection of systems parameters, as well as insights on an elegant three-way trade-off between utility, delay, and convergence speed. Collectively, these results serve as an exciting first step toward a cross-layer network control and optimization theory that leverages "momentum/memory" information. Memory/momentum-based cross-layer network optimization is an important and yet under-explored area. Future research topics may include, e.g., heavy-traffic delay performance analysis for memory/momentum-based scheduling algorithms, time-varying adaptive memory weight adjustments, and investigating the impact of higher order memory on network utility, delay, and convergence performances.

References

[1] X. Lin and N. B. Shroff, "The impact of imperfect scheduling on cross-layer congestion control in wireless networks," *IEEE/ACM Trans. Netw.*, vol. 14, no. 2, pp. 302–315, Apr. 2006.

[2] A. Eryilmaz and R. Srikant, "Joint congestion control, routing, and MAC for stability and fairness in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1514–1524, Aug. 2006.

[3] M. J. Neely, E. Modiano, and C.-P. Li, "Faireness and optimal stochastic control for heterogeneous networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 2, pp. 396–409, Apr. 2008.

[4] A. Eryilmaz and R. Srikant, "Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control," *IEEE/ACM Trans. Netw.*, vol. 15, no. 6, pp. 1333–1344, Dec. 2007.

[5] X. Lin, N. B. Shroff, and R. Srikant, "A tutorial on cross-layer optimization in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1452–1463, Aug. 2006.

[6] "Qualcomm aims at peer-to-peer with flashlinq," Feb. 2011. [Online]. Available: http://www.pcworld.com/article/219048/article.html

[7] "Data center bridging." [Online]. Available: http://www.cisco.com/c/dam/en/us/solutions/collateral/data-center-virtualization/ieee-802-1-data-center-bridging/at_a_glance_c45-460907.pdf

[8] M. J. Neely, "Super-fast delay tradeoffs for utility optimal fair scheduling in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1489–1501, Aug. 2006.

[9] L. Huang and M. J. Neely, "Delay reduction via lagrange multipliers in stochastic network optimization," *IEEE Trans. Autom. Control*, vol. 56, no. 4, pp. 842–857, Apr. 2011.

[10] L. Huang, X. Liu, and X. Hao, "The power of online learning in stochastic network optimization," in *Proc. ACM Sigmetrics*, Austin, TX, Jun.16-20, 2014, pp. 153–165.

[11] J. Liu, C. H. Xia, N. B. Shroff, and H. D. Sherali, "Distributed cross-layer optimization in wireless networks: A second-order approach," in *Proc. IEEE INFOCOM*, Turin, Italy, Apr. 14-19, 2013.

[12] J. Liu, N. B. Shroff, C. H. Xia, and H. D. Sherali, "Joint congestion control and routing optimization: An efficient second-order distributed approach," *IEEE/ACM Trans. Netw.*, 2015.

[13] B. T. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Computational Mathematics and Mathematical Physics*, vol. 4, no. 5, pp. 1–17, 1964.

[14] ——, *Introduction to Optimization*. New York, NY: Optimization Software, Inc., May 1987.

[15] R. J. Gibbens and F. P. Kelly, "Resource pricing and the evolution of congestion control," *Automatica*, vol. 35, pp. 1969–1985, 1999.

[16] S. Kunniyur and R. Srikant, "Analysis and design of an adaptive virtual queue algorithm for active queue management," in *Proc. ACM SIGCOMM*, San Diego, CA, Aug. 2001, pp. 123–134.

[17] A. Laksmikantha, C. Beck, and R. Srikant, "Robustness of real and virtual queue-based active queue management schemes," *IEEE/ACM Trans. Netw.*, vol. 13, no. 1, pp. 81–93, Feb. 2005.

[18] P. Ochs, T. Brox, and T. Pock, "iPiasco: Inertial proximal algorithm for strongly convex optimization," *Journal of Mathematical Imaging and Vision (JMIV)*, 2015.

[19] E. Ghadimi, I. Shames, and M. Johansson, "Multi-step gradient methods for networked optimization," *IEEE Trans. Signal Process.*, vol. 61, no. 21, pp. 5417–5429, Nov. 2013.

[20] "Heavy-Ball: A new approach to tame delay and convergence in wireless network optimization," *Technical Report*, Jul. 2015. [Online]. Available: https://www.dropbox.com/s/orszj3imzcg3p1y/HeavyBall_JCCR_TR.pdf?dl=0

[21] S. P. Meyn and R. L. Tweedie, *Markov Chains and Stochastic Stability*, 2nd ed. Cambridge, UK: Cambridge University Press, 2009.

[22] L. Jiang and J. Walrand, "A distributed CSMA algorithm for throughput and utility maximization in wireless networks," *IEEE/ACM Trans. Netw.*, vol. 18, no. 3, pp. 960–972, Jun. 2010.

[23] J. Ni, B. Tan, and R. Srikant, "Q-CSMA: Queue-length-based CSMA/CA algorithms for achieving maximum throughput and low delay in wireless networks," *IEEE/ACM Trans. Netw.*, vol. 20, no. 3, pp. 825–836, Jun. 2010.