Minimum Delay in Class of Throughput-Optimal Control Policies on Wireless Networks

Reza Banirazi, Edmond Jonckheere, Bhaskar Krishnamachari
Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089
E-mail: {banirazi, jonckhee, bkrishna}@usc.edu

Abstract—**This paper considers the problem of average network delay minimization on multiclass, multihop, stochastic wireless networks subject to inter-channel interference and time-varying topology. We present a network control policy that solves this problem in the class of all policies whose control decision is a function only of current queue congestion and current channel states, including policies that have perfect knowledge of probabilities associated with future random events. As important features of our proposed control policy, it is throughput-optimal in the sense that it can stabilize queues for any stabilizable arrival rate, it is robust to varying network topology and arrival rates, and it is implemented without requiring any knowledge of statistics and probabilities in the system. The proposed control policy is analyzed via the theory of stochastic discrete-time Lyapunov drift with a significant difference that unlike prior works that merely push down an upper-bound on the drift, our design genuinely minimizes the drift itself.**

I. Introduction

Consider a time-slotted stochastic wireless network, where the channel conditions are time-varying according to some (unknown) probability laws, and where simultaneous transmission over two channels fail if they have interference. Packets of the same size randomly arrive to any node, while destining for different destinations and perhaps requiring multihop routing paths. At each timeslot, a network controller observes queue congestion and channel conditions to make a control action that determines which set of channels should be activated and how many packets should be transmitted over them. In this paper, the goal of the controller is to minimize time average total queue congestion in the network, which is proportional to average network delay by Little's Theorem [1].

Achieving this goal for a general case requires the Markov structure of topology process, plus arrival and channel state probabilities. Then in theory, the solution is obtained through dynamic programming for each possible topology along with solving a Markov decision problem. By even having all of these required information, the number of queue buffers and channel states increase exponentially with the size of network, making dynamic programming and Markov decision theory impractical. In fact, even for the case of a single channel, it is difficult to implement the resulting stochastic algorithms [2]. While having a practical solution for a general case seems dubious, this paper solves this problem within an important class of network controllers, without requiring any of the above-mentioned information, and without dealing with dynamic programing or Markov decision process.

We consider the class of all network controllers that make routing decision as a pure function of current queue congestion and current channel states, including the ones with perfect probability knowledge of arrivals and channel states. As an important quality, this class allows us to design throughput-optimal controllers that can stabilize all arrival rates within the network capacity region. Two specific families of control policies that are categorized in this class are as follows.

1) All stationary randomized algorithms that make independent, stationary, and randomized transmission decisions at each timeslot based only on current channel capacities, and so independent of both queue congestion and channel quality factors [3], [4]. While such controllers exist in theory, they are intractable in practice as they typically require a full knowledge of arrival statistics and channel state probabilities. Besides this, the network controller would still need to solve a dynamic programming problem for each topology state, where the number of states grows exponentially with the number of channels. Although these controllers are not practically attractive, the fact that they exist plays a key role in the theoretical analysis of wireless network problems [3], [4].

2) All opportunistic max-weight algorithms that do not incorporate the Markov structure of network topology process into their decisions. These controllers make a transmission decision at every timeslot via locally weighing each link and then globally scheduling a set of links with maximum sum weight. Our proposed control policy in this paper, also the well-known *Back-Pressure (BP)* algorithm [5] and almost all of its follow-up derivations, belong to this group.

The seminal paper on congestion-aware routing control by Tassiulas and Ephremides [5] showed that the queue-differential, link-capacity-based BP is throughput-optimal under very general conditions. We analytically prove that our proposed control is also throughput-optimal under the same general conditions and with the same complexity as BP, while on top of that minimizing the time average total queue congestion in the above-mentioned class of network controllers. We refer to our proposed policy as *Heat-Diffusion (HD),* emphasizing that in a simplistic metaphorical view of the network, where queue occupancy is heat quantity and packets are calories, the limit flow of packets towards the destination takes the form of the flow of heat towards the sink [6], [7].

The stochastic network optimization framework for the design and analysis of BP has proven to be a very popular research domain, with many new theoretical results in recent years to further enhance the original framework, where many of them have focused on enhancing the delay performance.

This work was supported by NSF grants CNS-1017881 and CNS-1049541.

Shadow queues has enabled BP to handle multicast sessions with reduced number of actual queues that need to be maintained [8], [9]. A delay-based BP formulation has obtained a significantly lower delay by using the cumulative time packet age queue [10]. Incorporating last-input-first-output (LIFO) service into BP has shown a better delay quality [11]. Adaptive redundancy has been designed to improve the low-rate delay performance of BP in intermittently connected mobile networks [12]. Using graph embedding, [13] combined BP with greedy routing in hyperbolic coordinates to obtain a throughput-delay tradeoff. The framework has also been extended to handle finite buffer sizes [14]. Other researchers have focused on making BP scheduling more distributed so that it can be implemented more easily [15]–[17]. More recently, there have been several reductions of BP theory to practice, in the form of practically implemented and experimentally evaluated distributed protocols [18]–[20].

An infant idea of HD protocol was introduced in [6] and then vastly revised and developed in [7] with the aim of providing the best tradeoff between average queue congestion and average routing cost on a *uniclass* network. This paper extends the result of [7] on minimizing average queue congestion to multiclass networks through a concise, in depth analysis.

Note: The page limit does not let us include the proofs, but all the proofs are available in the long-version paper [21].

II. Preliminaries

The network is described by a *simple, directed* connectivity graph with set of nodes $\mathcal{V}$ and directed edges $\mathcal{E}$. New packets with different destinations in a set $\mathcal{K}\subseteq\mathcal{V}$ randomly arrive into different nodes. Packets of the same destination form a *class.* Each node $i\in\mathcal{V}$ holds a *separate* queue $q_i^{(d)}$ for each *d-class* to transmit over its outgoing links. We assume that packets are not sent to trapping nodes in the network, i.e. when a node accepts d-class packets it means that there exists at least one possible route from that node to the destination d. While this assumption is not required for any of our analytical results, having it ensures that a dynamic control policy, with no routing path constraint, will not mistakenly send a packet to a trapping node that prevents it from ever reaching its destination.

Contrary to wireline networks where links are independent resources, in a wireless network two links cannot simultaneously transmit if they have interference. Define a schedule as a set of links in which no two links interfere with each other, and call it *maximal* if no more links can be added to that without violating the interference constraints. Each maximal schedule is represented as a vector, referred to as *scheduling vector,* in which each entry takes the value 1 if the corresponding link is included in the maximal schedule, and 0 otherwise. For a given connectivity graph $(\mathcal{V},\mathcal{E})$, we assume that each maximal scheduling vector $\boldsymbol{\pi}$ takes values in a finite *scheduling set* Π, which is the collection of all available maximal schedules.

Observe that the scheduling set varies according to interference model. The results of this paper is valid for the category of all interference models in which a node cannot transmit packets to more than one neighbor at each timeslot, i.e. a node may receive packets from several of its incoming links and at the same time may transmit packets over one of its outgoing links. To the best of our knowledge, interference constraints in all current network layer protocols, including general K-hop interference models, fall in this category.

A discrete-time stochastic process $x(n)$ is called stable if

$$\overline{x} := \limsup_{\tau\to\infty} 1/\tau \sum\nolimits_{n=0}^{\tau-1} \mathbb{E}\{x(n)\} < \infty \qquad (1)$$

where $\mathbb{E}$ denotes expectation. A queuing network is *stable* if all its queues are stable. An arrival rate matrix is *stabilizable* if there exists a control policy to stably support it. For a control policy, *stability region* is the set of all arrival rate matrices that it can stably support. Network layer *capacity region* $\mathcal{C}$ is the union of the stability regions achieved by all control policies (possibly unfeasible). A control policy is *throughput-optimal* if it stabilizes the entire capacity region.

A. State Space Representation of the System

Consider a multiclass queuing network $(\mathcal{V},\mathcal{E},\mathcal{K})$. For each $i\in\mathcal{V}$ and $d\in\mathcal{K}$, let $q_i^{(d)}(n)$ be the integer number of d-classes in the node i at the slot n. For each link $ij\in\mathcal{E}$, let $\mu_{ij}(n)$ be the link *capacity* at the slot n. Link capacity, which is frequently called link *transmission rate* in literature, counts the maximum number of packets the link can transmit at one timeslot. We also define the link *actual-transmission* $f_{ij}^{(d)}(n)$ that counts the number of d-classes *genuinely* sent over the link ij at the slot n. It is important to discriminate between link actual-transmission and link capacity. In particular, while link capacities vary by channel states, link actual-transmissions are assigned by a network controller constrained to

$$0 \leqslant f_{ij}^{(d)}(n) \leqslant \min\{q_i^{(d)}(n),\, \mu_{ij}(n)\}.$$

It is assumed that each packet leaves the network as soon as reaching its destination, thus the backlog of d-classes at the destination node d is zero for all $d\in\mathcal{K}$. Then the state variables of the system can be represented by the hyper-vector

$$\boldsymbol{q}_\circ(n) := \big[\, \boldsymbol{q}_\circ^{(1)}(n),\ldots,\boldsymbol{q}_\circ^{(|\mathcal{K}|)}(n) \,\big]^\top \in \mathbb{R}^{(|\mathcal{V}|-1)|\mathcal{K}|}$$
$$\boldsymbol{q}_\circ^{(d)}(n) := \big[\, q_1^{(d)}(n),\ldots,q_{d-1}^{(d)}(n), q_{d+1}^{(d)}(n),\ldots,q_{|\mathcal{V}|}^{(d)}(n) \,\big]$$

where $q_d^{(d)}(n)\equiv 0$ is dropped from the set of state variables.

Notation 1: We use a subscript $\circ$ to denote a reduced vector or matrix obtained by discarding the entries corresponding to the destination node d. A superscript $\top$ denotes the transpose operation. The set of real numbers is denoted with $\mathbb{R}$. The cardinality of the set $\mathcal{V}$ is denoted with $|\mathcal{V}|$.

Let a stochastic process $a_i^{(d)}(n)$ represent the integer number of exogenous d-classes arriving into the node i at the slot n. Discarding $a_d^{(d)}(n)\equiv 0$, the hyper-vector of node arrivals

$$\boldsymbol{a}_\circ(n) := \big[\, \boldsymbol{a}_\circ^{(1)}(n),\ldots,\boldsymbol{a}_\circ^{(|\mathcal{K}|)}(n) \,\big]^\top \in \mathbb{R}^{(|\mathcal{V}|-1)|\mathcal{K}|}$$
$$\boldsymbol{a}_\circ^{(d)}(n) := \big[\, a_1^{(d)}(n),\ldots,a_{d-1}^{(d)}(n), a_{d+1}^{(d)}(n),\ldots,a_{|\mathcal{V}|}^{(d)}(n) \,\big].$$

Likewise, the hyper-vector of link actual-transmissions

$$\boldsymbol{f\!\!f}(n) := \big[\, \boldsymbol{f}^{(1)}(n),\ldots,\boldsymbol{f}^{(|\mathcal{K}|)}(n) \,\big]^\top \in \mathbb{R}^{|\mathcal{E}||\mathcal{K}|}$$
$$\boldsymbol{f}^{(d)}(n) := \big[\, f_1^{(d)}(n),\ldots,f_{|\mathcal{E}|}^{(d)}(n) \,\big]$$

where as before, $f_{ij}^{(d)}(n)$ is the integer number of d-classes actually sent over the link ij at the slot n.

Notation 2: For a value x of a directed link ℓ from node i to node j, we use the notation x_ℓ and x_{ij} interchangeably.

Given a directed graph $(\mathcal{V}, \mathcal{E})$, let $\boldsymbol{B}$ denote the *node-edge* incidence matrix in which $B_{i\ell}$ is 1 if node i is the tail of directed edge ℓ, is -1 if i is the head, and is 0 otherwise. For a class d, let $\boldsymbol{B}_{\circ}^{(d)}$ denote a reduction of $\boldsymbol{B}$ through discarding the row corresponding to the destination node d. We refer to $\boldsymbol{B}_{\circ}^{(d)}$ as the *basis* incidence matrix with respect to the node d. Extending this structure to the multiclass framework, the *generalized* basis incidence matrix is built as

$$\mathbb{B}_{\circ} := \operatorname{diag}\big(\big[\boldsymbol{B}_{\circ}^{(1)}, \ldots, \boldsymbol{B}_{\circ}^{(|\mathcal{K}|)} \big] \big) \in \mathbb{R}^{(|\mathcal{V}|-1)|\mathcal{K}| \times |\mathcal{E}||\mathcal{K}|}$$

where $\operatorname{diag}(\boldsymbol{v})$ is the block diagonal matrix expansion of $\boldsymbol{v}$. One can then verify that $\mathbb{B}_{\circ}\boldsymbol{f}(n)$ is a hyper-vector in which the entry corresponding to node i and class d becomes

$$(\mathbb{B}_{\circ}\mathbb{f})_i^{(d)}(n) = \sum\nolimits_{b \in \mathrm{out}(i)} f_{ib}^{(d)}(n) - \sum\nolimits_{a \in \mathrm{in}(i)} f_{ai}^{(d)}(n)$$

where $\mathrm{in}(i)$ and $\mathrm{out}(i)$ respectively denote the set of incoming and outgoing neighbors of node i.

Using these ingredients, the $\mathbb{f}$-controlled, stochastic state dynamics of a multiclass queuing network is captured by

$$\boldsymbol{q}_{\circ}(n+1) = \boldsymbol{q}_{\circ}(n) + \boldsymbol{a}_{\circ}(n) - \mathbb{B}_{\circ}\mathbb{f}(n). \tag{2}$$

Considering the difference between link capacity and link actual-transmission explains why despite traditional notation in literature, we do not need any $(\cdot)^{+}$ operation in (2).

Notation 3: Given x as a real number, $x^{+} := \max\{0, x\}$. The definition is extended entrywise to vectors and matrices.

Under the assumption of no transmission to trapping nodes, if node i is a trapping node for class d then one may discard $q_i^{(d)}(n)$ from the set of state variables $\boldsymbol{q}(n)$, $a_i^{(d)}(n)$ from the set of node arrivals $\boldsymbol{a}(n)$, and the row corresponding to the node i from the basis incidence matrix $\boldsymbol{B}_{\circ}^{(d)}$. Note that in this case, the node i supposedly receives no exogenous d-classes. Further, it does not accept d-classes from its incoming links, and so neither sends any d-classes on its outgoing links.

B. Characteristic of Network Capacity Region

In wireless systems, channel conditions are uncontrollable parameters that vary in time due to environmental change and user mobility. We assume that the sets $\mathcal{V}$ and $\mathcal{E}$ change much slower than channel states, so that we can fix them during the time of interest. We also assume that channel states remain fixed during a timeslot, while may change across slots.

Let a stochastic process $\boldsymbol{S}(n) = \big(S_1(n), \cdots, S_{|\mathcal{E}|}(n)\big)$ represent channel states at the slot n, describing all uncontrollable conditions that affect channel capacities. We use zero capacity to mark a temporarily unavailable channel—due to, for example, obstacle effect. Suppose that $\boldsymbol{S}(n)$ evolves according to an ergodic stationary process and takes values in a finite (but arbitrarily large) set $\mathcal{S}$. Then by Birkhoff's ergodic theorem, each state $\boldsymbol{S} \in \mathcal{S}$ is of probability

$$s := \mathbb{P}\big\{\boldsymbol{S}(n) = \boldsymbol{S}\big\} = \limsup_{\tau \to \infty} 1/\tau \sum\nolimits_{n=0}^{\tau-1} \mathbb{I}_{\boldsymbol{S}(n)=\boldsymbol{S}}$$

where $\sum_{\boldsymbol{S} \in \mathcal{S}} s = 1$, and $\mathbb{I}_X$ is an indicator function that takes the value 1 if the statement X is true, and 0 otherwise. We insist that our proposed control policy does not require the state probabilities s. However, the existence of s is important to establish the network capacity region, and also to characterize the stationary randomized control policies.

Now, consider a connectivity graph $(\mathcal{V}, \mathcal{E})$ together with a channel state process $\boldsymbol{S}(n)$. For an arrival *rate* vector $\overline{\boldsymbol{a}}$ to be in the network capacity region $\mathcal{C}$, the necessary and sufficient condition is the existence of a set of link actual-transmissions such that their expected time averages jointly satisfy node flow conservation and link capacity constraints, viz.

$$\overline{a_i^{(d)}} = \sum\nolimits_{b \in \mathrm{out}(i)} \overline{f_{ib}^{(d)}} - \sum\nolimits_{a \in \mathrm{in}(i)} \overline{f_{ai}^{(d)}} \tag{3}$$

$$\sum\nolimits_{i \in \mathcal{V}} \overline{a_i^{(d)}} = \sum\nolimits_{a \in \mathrm{in}(d)} \overline{f_{ad}^{(d)}} \tag{4}$$

$$\sum\nolimits_{d \in \mathcal{K}} \overline{f_{ij}^{(d)}} \leqslant \sum\nolimits_{\boldsymbol{S} \in \mathcal{S}} s\, \mathbb{E}\big\{\mu_{ij}(n) \big| \boldsymbol{S}(n) = \boldsymbol{S}\big\} \tag{5}$$

where the overbar notation is as defined in (1). Equality (3) secures flow conservation at intermediate nodes. Specifically, the matrix form of (3) becomes $\overline{\boldsymbol{a}_{\circ}} = \boldsymbol{B}_{\circ}\overline{\boldsymbol{f}}$ showing the expected time average of (2) subject to queue stability, where $\lim_{n \to \infty} \boldsymbol{q}_{\circ}(n+1) = \boldsymbol{q}_{\circ}(n) < \infty$. Equality (4) guarantees that there is no trapping node and so all d-classes arrived into the network are ultimately collected by the destination d. The right hand side of (5) reads $\overline{\mu_{ij}}$ and so the inequality guarantees the link capacity constraint. The constraints (3)–(5) imply that the capacity region is convex, closed, and bounded [3].

Observe that the link actual-transmissions are not fixed, but depend on the control policy. Also observe that there potentially exist infinite number of control policies that can meet the constraints (3)–(5). Among them are the ones that use the simple probability concept of randomly distributing packets such that holding the desired time averages (3)–(5). As mentioned in Sec. I, these stationary randomized policies typically require expensive computation along with perfect knowledge of arrival statistics and channel state probabilities that are prohibitive in practice. Nevertheless, the existence of these *queue-independent* routing policies is important in the analysis of our proposes control policy.

C. Back-Pressure (BP) Algorithm

At every timeslot n, BP policy [5] at network layer observes queue backlogs $q_i^{(d)}(n)$ and estimates channel capacities $\mu_{ij}(n)$ to make a transmission decision as follows.

BP1) *Weighing:* On every directed link ij and for each class d find $q_{ij}^{(d)}(n) := q_i^{(d)}(n) - q_j^{(d)}(n)$ and select the optimal class

$$d_{ij}^{*}(n) := \arg\max\nolimits_{d \in \mathcal{K}} q_{ij}^{(d)}(n). \tag{6}$$

Then give a weight to the link using its estimated capacity as

$$w_{ij}(n) := \mu_{ij}(n)\, q_{ij}^{(d^*)}(n)^{+}. \tag{7}$$

BP2) *Scheduling:* Find the scheduling vector such that

$$\boldsymbol{\pi}(n) = \arg\max_{\boldsymbol{\pi} \in \Pi} \sum\nolimits_{ij \in \mathcal{E}} \pi_{ij} w_{ij}(n) \tag{8}$$

where ties are broken randomly.

BP3) *Forwarding:* On each activated link ij with $w_{ij}(n) > 0$ transmit from the class $d_{ij}^{*}(n)$ at full capacity $\mu_{ij}(n)$. If there is no enough d^{*}-classes at node i, transmit null packets.

III. Heat-Diffusion Control Policy

To prepare a convenient way of unifying our proposed control policy with the previous works on BP schemes, we design HD with the same algorithmic structure, complexity, and overhead as BP. This provides an easy way to leverage all advanced improvements to BP (using e.g. LIFO service, packet ages, adaptive redundancy, queue prioritization, etc.) to further enhance HD quality. It also simplifies the approach to practice via a smooth software transition from BP to HD.

We first introduce HD for the uniclass networks, and then extend it to multiclass problems in two versions: (i) when an activated link can transmit packets from only one class at each timeslot, and (ii) when an activated link is allowed to transmit packets from different classes at each timeslot.

A. Heat-Diffusion (HD) Algorithm on Uniclass Networks

At every timeslot n, HD routing policy at the network layer observes queue backlogs $q_i(n)$ and estimates channel capacities $\mu_{ij}(n)$ to make a transmission decision as follows.

HDa.1) *Weighing:* On every directed link ij find $q_{ij}(n) := q_i(n) - q_j(n)$ and first calculate the number of packets the link would transmit if it were activated as

$$\widehat{f_{ij}}(n) := \min\{q_{ij}(n)^+,\ \mu_{ij}(n)\} \tag{9}$$

where the hat notation denotes a predicted value which would not necessarily be realized. Then give a weight to the link using its predicted actual-transmission as

$$w_{ij}(n) := 2\, q_{ij}(n)\, \widehat{f_{ij}}(n) - \big(\widehat{f_{ij}}(n)\big)^2. \tag{10}$$

HDa.2) *Scheduling:* Find the scheduling vector, in the same way as BP, using the max-weight scheduling in (8).

HDa.3) *Forwarding:* Transmit $\widehat{f_{ij}}(n)$ number of packets over each activated link ij, leading to

$$f_{ij}(n) = \begin{cases} \widehat{f_{ij}}(n) & \text{if } \pi_{ij}(n) = 1 \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

It is critical to discriminate among link actual-transmission $f_{ij}(n)$, link transmission prediction $\widehat{f_{ij}}(n)$, and link capacity $\mu_{ij}(n)$. Table 1 compares HD and BP algorithms, emphasizing the same structure, complexity, and overhead.

Some general notes on the HD algorithm: (i) If $q_{ij}(n) \leqslant 0$, we get $\widehat{f_{ij}}(n) = 0$ due to (9), and $w_{ij}(n) = 0$ from (10). Thus even if the link were scheduled, still no packet would be transmitted over it. (ii) If $q_{ij} > 0$, then $q_{ij}(n)^+ = q_{ij}(n)$. Also $\widehat{f_{ij}}(n) \leqslant q_{ij}(n)$ due to (9). Thus the link weight (10) is always nonnegative. (iii) Since $q_{ij}(n)^+ \leqslant q_i(n)$, $\widehat{f_{ij}}(n)$ never exceeds the number of packets in the transmitting node.

Remark 1: While BP is driven by link capacities $\mu_{ij}(n)$, HD emphasizes actual number of transmittable packets $\widehat{f_{ij}}(n)$, though through (9), it indirectly takes into account the link capacities too. Thus HD allocates resources based only on genuinely transmittable packets, without counting on null packets as is practiced in BP schemes.

Remark 2: The link weight (10), which itself directly controls the scheduling optimization problem, is taken quadratic in the queue-differential $q_{ij}(n)$, where for $q_{ij}(n) \leqslant \mu_{ij}(n)$ is simplified into $w_{ij}(n) = q_{ij}(n)^2$. This contrasts with BP weighing $w_{ij}(n) = \mu_{ij}(n) q_{ij}(n)$ which is linear in $q_{ij}(n)$. The quadratic weight is central to the HD key property (Th. 1) which is fundamental to other HD qualities.

TABLE I
COMPARING HD AND BP ROUTING POLICIES IN A UNICLASS NETWORK.

Weighing	$\widehat{f_{ij}}(n)$	BP	$\min\{\mu_{ij}(n), q_i(n)\}$
		HD	$\min\{q_{ij}(n)^+,\ \mu_{ij}(n)\}$
	$w_{ij}(n)$	BP	$\mu_{ij}(n)\, q_{ij}(n)^+$
		HD	$2\, q_{ij}(n)\, \widehat{f_{ij}}(n) - \widehat{f_{ij}}(n)^2$
Scheduling	$\boldsymbol{\pi}(n) = \arg\max_{\boldsymbol{\pi}\in\Pi} \sum_{ij\in\mathcal{E}} \pi_{ij} w_{ij}(n)$		
Forwarding	$f_{ij}(n) = \begin{cases} \widehat{f_{ij}}(n) & \text{if } \pi_{ij}(n) = 1 \\ 0 & \text{otherwise} \end{cases}$		

Remark 3: Unlike BP that forwards the highest possible number of packets over activated links, HD controls the packet forwarding by restricting it to the link queue differential. This reduces *queue oscillations* by the decrease of unnecessary packet forwardings across the links, which in turn can reduce transmission cost for supporting a given traffic.

Remark 4: Like BP, also HD is based on a centralized scheduling whose complexity is prohibitive in practice. However, much progress has recently been made to ease this difficulty by deriving decentralized schedulers with the performance of arbitrarily close to the centralized version [15]–[17].

B. Multiclass HD with Single Class Transmission

Now consider a multiclass network where each node may have packets for different destinations. In the same way that BP performs, here we assume that each link may transmit at most one class of packets per timeslot. For this case, HD makes a timeslot transmission decision as follows.

HDb.1) *Weighing:* On every directed link ij select the optimal class, in the same way as BP, as defined in (6) and calculate the link actual-transmission prediction as

$$\widehat{f_{ij}^{(d^*)}}(n) := \min\{q_{ij}^{(d^*)}(n)^+,\ \mu_{ij}(n)\}. \tag{12}$$

Then endow each link with a weight as

$$w_{ij}(n) := 2\, q_{ij}^{(d^*)}(n)\, \widehat{f_{ij}^{(d^*)}}(n) - \Big(\widehat{f_{ij}^{(d^*)}}(n)\Big)^2. \tag{13}$$

HDb.2) *Scheduling:* Find the scheduling vector using the max-weight scheduling in (8).

HDb.3) *Forwarding:* Transmit $\widehat{f_{ij}^{(d^*)}}(n)$ number of packets from the class $d_{ij}^*(n)$ over each activated link ij.

Remark 5: In packet switches, the work of [22] extends queue-based scheduling to α-weighted schedulers which use α-exponent of queue lengths. There has been a non-proven conjecture that in heavy traffic condition, average delay is minimized when $\alpha \to 0$, with a discussion of it given in [23] along with some counterexamples. In a very special case when all link capacities are the same, i.e. $\mu_{ij}(n) = \mu(n)$, and all link queue-differentials are always less than it, i.e. $q_{ij}(n) < \mu(n)$, HD control policy with single class transmission assumption and α-weighted policy with $\alpha = 2$ become equivalent.

C. Multiclass HD with Multiple Class Transmission

It seems obvious that network resources are squandered by restricting the control policy to transmit only one class of packets per link per timeslot. In other words, the larger capacity of network would be utilized, and so the average network delay would decrease, if each activated link were properly filled up to its full capacity. At the same time, we will show in Th. 3, and also by simulation, that blindly filling up the links by simply sending the maximum number of packets from only one selected class, which is basically practiced in BP schemes, only depletes the network resources with even negative impact on delay performance. Thus the important question is how a dynamic control policy with no routing path constraint can minimize the average network delay by utilizing the maximum timeslot resources. We answer this question by introducing an enhanced HD algorithm that chooses packets from different classes to transmit over an activated link, where the timeslot transmission decision is made as follows.

HDc.1) *Weighing:* On every directed link ij and for each class d find $q_{ij}^{(d)}(n) := q_i^{(d)}(n) - q_j^{(d)}(n)$ and create a set

$$\mathcal{K}_{ij}(n) \subseteq \mathcal{K} \text{ such that } q_{ij}^{(d)}(n) > 0\,,\, \forall d \in \mathcal{K}_{ij}(n).$$

Fix $\widehat{f_{ij}^{(d)}}(n) = 0$ for each $d \notin \mathcal{K}_{ij}(n)$, and find $\widehat{f_{ij}^{(d)}}(n)$ for every $d \in \mathcal{K}_{ij}(n)$ by solving the optimization problem

$$\begin{aligned} &\text{Minimize:} \quad \textstyle\sum_{d\in\mathcal{K}_{ij}(n)} \Big(q_{ij}^{(d)}(n) - \widehat{f_{ij}^{(d)}}(n)\Big)^2 \\ &\text{Subject to:} \begin{cases} \textstyle\sum_{d\in\mathcal{K}_{ij}(n)} \widehat{f_{ij}^{(d)}}(n) \leqslant \mu_{ij}(n) \\ 0 \leqslant \widehat{f_{ij}^{(d)}}(n) \leqslant q_{ij}^{(d)}(n),\ \forall d \in \mathcal{K}_{ij}(n). \end{cases} \end{aligned} \tag{14}$$

Then give a weight to each class $d \in \mathcal{K}_{ij}(n)$ as

$$w_{ij}^{(d)}(n) := 2\, q_{ij}^{(d)}(n) \widehat{f_{ij}^{(d)}}(n) - \Big(\widehat{f_{ij}^{(d)}}(n)\Big)^2 \tag{15}$$

and aggregate them to determine the final link weight as

$$w_{ij}(n) := \textstyle\sum_{d\in\mathcal{K}_{ij}(n)} w_{ij}^{(d)}(n). \tag{16}$$

HDc.2) *Scheduling:* Find the scheduling vector using the max-weight scheduling (8).
HDc.3) *Forwarding:* Transmit $\widehat{f_{ij}^{(d)}}(n)$ number of packets from the class d over each activated link ij.

Problem (14) is a standard least-norm optimization with variable bounds that can be solved in fast polynomial time at each node, i.e. in a fully decentralized manner. A related algorithm is developed in the following.

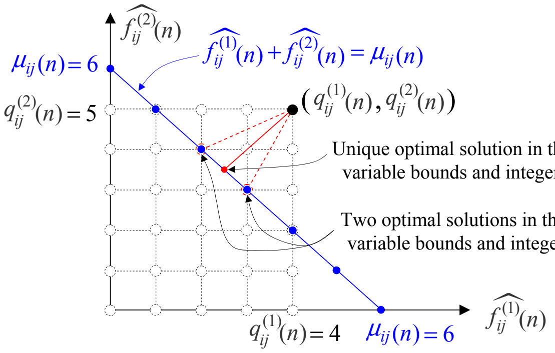
To simplify the notation, let us drop the overhat symbol and the time variable (n). First observe that in the problem (14),

$$\text{if } \textstyle\sum_{d\in\mathcal{K}_{ij}} q_{ij}^{(d)} \leqslant \mu_{ij} \text{ then } f_{ij}^{(d)} = q_{ij}^{(d)}, \forall\, d \in \mathcal{K}_{ij}.$$

Thus assume $\sum_{d\in\mathcal{K}_{ij}} q_{ij}^{(d)} > \mu_{ij}$. This converts the first constraint from inequality into equality, viz. $\sum_{d\in\mathcal{K}_{ij}} f_{ij}^{(d)} = \mu_{ij}$. Then, in the absence of lower variable bounds and integer constraints, the problem has a unique solution as

$$f_{ij}^{(d)} = q_{ij}^{(d)} + \big(\mu_{ij} - \textstyle\sum_{d\in\mathcal{K}_{ij}} q_{ij}^{(d)}\big)/|\mathcal{K}_{ij}|\,,\, \forall\, d \in \mathcal{K}_{ij}$$

which can be confirmed using a basic Lagrange argument.



$\widehat{f_{ij}^{(2)}}(n)$
$\widehat{f_{ij}^{(1)}}(n)+\widehat{f_{ij}^{(2)}}(n)=\mu_{ij}(n)$
$\mu_{ij}(n)=6$
$q_{ij}^{(2)}(n)=5$
$(q_{ij}^{(1)}(n), q_{ij}^{(2)}(n))$
Unique optimal solution in the absence of variable bounds and integer constraints
Two optimal solutions in the presence of variable bounds and integer constraints
$q_{ij}^{(1)}(n)=4$
$\mu_{ij}(n)=6$
$\widehat{f_{ij}^{(1)}}(n)$

Fig. 1. Solving (14) for a two-class case when $q_{ij}^{(1)}(n) + q_{ij}^{(2)}(n) > \mu_{ij}(n)$.

From a geometrical standpoint, the latter represents projection of the point $(q_{ij}^{(1)}, \cdots, q_{ij}^{(|\mathcal{K}_{ij}|)})$ onto the hyperplane $\sum_{d\in\mathcal{K}_{ij}} f_{ij}^{(d)} = \mu_{ij}$. Under integer constraints, $\sum_{d\in\mathcal{K}_{ij}} f_{ij}^{(d)} = \mu_{ij}$ represents an integer hypergrid where the optimal solution(s) will be the vertex(es) of this hypergrid with shortest Euclidean distance to the point $(q_{ij}^{(1)}, \cdots, q_{ij}^{(|\mathcal{K}_{ij}|)})$. Note that the solution to the integer problem is not necessarily unique. Subjecting the solution to the lower variable bounds, it must also meet $f_{ij}^{(d)} \geqslant 0,\ \forall\, d \in \mathcal{K}_{ij}$. This procedure is displayed in Fig. 1 for a two-class case.

The following algorithm implements the above-explained process of solving (14) when $\sum_{d\in\mathcal{K}_{ij}(n)} q_{ij}^{(d)}(n) > \mu_{ij}(n)$.

S1: Let $h = \big(\sum_{d\in\mathcal{K}_{ij}(n)} q_{ij}^{(d)}(n) - \mu_{ij}(n)\big)/|\mathcal{K}_{ij}(n)|$ and for every $d \in \mathcal{K}_{ij}(n)$ take $\widehat{f_{ij}^{(d)}}(n) = q_{ij}^{(d)}(n) - h$.

S2: Find $d_1 = \arg\min_{d\in\mathcal{K}_{ij}(n)} \widehat{f_{ij}^{(d)}}(n)$ and if $\widehat{f_{ij}^{(d_1)}}(n) < 0$, then remove d_1 from $\mathcal{K}_{ij}(n)$ and go back to S1.

S3: Let $r = \mu_{ij}(n) - \sum_{d\in\mathcal{K}_{ij}(n)} \big\lfloor \widehat{f_{ij}^{(d)}}(n) \big\rfloor$. For r randomly chosen classes in $\mathcal{K}_{ij}(n)$ assign $\widehat{f_{ij}^{(d)}}(n) = \big\lceil \widehat{f_{ij}^{(d)}}(n) \big\rceil$ and for other classes in $\mathcal{K}_{ij}(n)$ assign $\widehat{f_{ij}^{(d)}}(n) = \big\lfloor \widehat{f_{ij}^{(d)}}(n) \big\rfloor$.

Notation 4: Given x as a real number, the floor function $\lfloor x \rfloor$ maps x to the largest preceding integer, and the ceiling function $\lceil x \rceil$ maps x to the smallest following integer.

Note in S2 that in case of discarding d_1 from $\mathcal{K}_{ij}(n)$, still $\sum_{d\in\mathcal{K}'_{ij}(n)} q_{ij}^{(d)}(n) > \mu_{ij}(n)$ where $\mathcal{K}'_{ij}(n) := \mathcal{K}_{ij}(n) - \{d_1\}$. Strictly speaking, having $\widehat{f_{ij}^{(d_1)}}(n) < 0$ implies

$$|\mathcal{K}'_{ij}(n)|\, q_{ij}^{(d_1)}(n) < \textstyle\sum_{d\in\mathcal{K}'_{ij}(n)} q_{ij}^{(d)}(n) - \mu_{ij}(n).$$

Then as $q_{ij}^{(d_1)}(n) > 0$, the left hand side of the above inequality is positive that leads to $\sum_{d\in\mathcal{K}'_{ij}(n)} q_{ij}^{(d)}(n) > \mu_{ij}(n)$.

Observe that S1 finds the optimal solution in the absence of variable bounds and integer constraints, S2 ensures that the solution meets the variable lower bounds, and S3 determines an integer solution by finding a vertex on the integer hypergrid—the first constraint of (14) with equality—with the shortest distance to the initial solution obtained by S1-S2. The term "r randomly chosen classes" in S3 comes due to the fact that the integer problem may have more than one solution. When the initial solution of S1-S2 is integer, it will be the solution to the integer problem too, and so unique. Otherwise, there potentially exist several vertexes on the integer hypergrid with equal distance from the non-integer initial solution and shorter than the distance of other vertexes (see Fig. 1).

IV. The Key Property of Heat-Diffusion Policy

This section formalizes the key property of HD control policy which is central to the proof of Th. 2 on HD Throughput-Optimality and Th. 3 on HD delay minimization.

Theorem 1: At every timeslot n and subject to network constraints, the HD control policy maximizes the functional

$$J(\mathbb{f},n) := 2\,\mathbb{f}(n)^{\top}\mathbb{B}_{\circ}^{\top}\boldsymbol{q}_{\circ}(n) - \mathbb{f}(n)^{\top}\mathbb{B}_{\circ}^{\top}\mathbb{B}_{\circ}\mathbb{f}(n). \quad (17)$$

Specifically, assuming that a node cannot transmit to more than one neighbor at a timeslot, the following is true:

- On a uniclass network, the HD algorithm of Sec. III-A maximizes $J(\boldsymbol{f},n) = 2\,\boldsymbol{f}(n)^{\top}\boldsymbol{B}_{\circ}^{\top}\boldsymbol{q}_{\circ}(n) - \boldsymbol{f}(n)^{\top}\boldsymbol{B}_{\circ}^{\top}\boldsymbol{B}_{\circ}\boldsymbol{f}(n)$.
- On a multiclass network subject to transmitting from at most one class per link per timeslot, the HD algorithm of Sec. III-B maximizes the $J(\mathbb{f},n)$ functional of (17).
- On a multiclass network that allows transmitting from multiple classes per link per timeslot, the HD algorithm of Sec. III-C maximizes the $J(\mathbb{f},n)$ functional of (17).

V. Heat-Diffusion Throughput Optimality

To analyze the HD stability, we exploit the theory of Lyapunov drift for stochastic discrete-time systems. Consider the basic quadratic Lyapunov candidate

$$W(n) := \boldsymbol{q}_{\circ}(n)^{\top}\boldsymbol{q}_{\circ}(n) = \sum\nolimits_{i\in\mathcal{V}}\sum\nolimits_{d\in\mathcal{K}} q_i^{(d)}(n)^2.$$

Letting the Lyapunov drift $\Delta W(n) := W(n+1) - W(n)$ and substituting for $\boldsymbol{q}_{\circ}(n+1)$ from (2) lead to

$$\begin{aligned}\Delta W(n) = 2\big(\boldsymbol{a}_{\circ}(n) - \mathbb{B}_{\circ}\mathbb{f}(n)\big)^{\top}\boldsymbol{q}_{\circ}(n) + \boldsymbol{a}_{\circ}(n)^{\top}\boldsymbol{a}_{\circ}(n)\\ + \mathbb{f}(n)^{\top}\mathbb{B}_{\circ}^{\top}\mathbb{B}_{\circ}\mathbb{f}(n) - 2\,\mathbb{f}(n)^{\top}\mathbb{B}_{\circ}^{\top}\boldsymbol{a}_{\circ}(n).\end{aligned} \quad (18)$$

Using the $J(\mathbb{f},n)$ expression of (17) yields

$$\begin{aligned}\Delta W(n) = \;& 2\,\boldsymbol{a}_{\circ}(n)^{\top}\boldsymbol{q}_{\circ}(n) - J(\mathbb{f},n)\\ & + \boldsymbol{a}_{\circ}(n)^{\top}\boldsymbol{a}_{\circ}(n) - 2\,\mathbb{f}(n)^{\top}\mathbb{B}_{\circ}^{\top}\boldsymbol{a}_{\circ}(n).\end{aligned} \quad (19)$$

Now consider an arrival rate $\overline{\boldsymbol{a}_{\circ}}$ being interior to the capacity region $\mathcal{C}$, i.e. there exists a vector $\boldsymbol{\epsilon}$ with positive entries such that $\overline{\boldsymbol{a}_{\circ}} + \boldsymbol{\epsilon} \in \mathcal{C}$. Thus by condition (3), there exists a hyper-flow $\mathbb{f}'(n)$ such that $\overline{\mathbb{B}_{\circ}\mathbb{f}'} = \overline{\boldsymbol{a}_{\circ}} + \boldsymbol{\epsilon}$. At the same time, Th. 1 guarantees that $J(\mathbb{f}^{\star},n) \geqslant J(\mathbb{f}',n)$ at each slot n, where $\mathbb{f}^{\star}(n)$ is the link actual-transmissions yielded by HD at the slot n. Then the next theorem is proven by showing that the expected value of Lyapunov drift (18) is bounded.

To simplify the proofs, throughout this paper we assume both arrival and channel state processes are independently and identically distributed (i.i.d.) over timeslots. However, all the results can easily be extended to non-i.i.d. systems with stationary ergodic processes of finite mean and variance.

Theorem 2: Suppose that arrivals and channel states are i.i.d. over timeslots and with respect to each other. The HD control policy is throughput-optimal in the sense that it secures network stability for any arrival rate interior to the network capacity region characterized by the constraints (3)–(5).

Remark 6: A common theme to all of the works on BP, going back to the original paper [5], is that the algorithm is derived by the greedy minimization of a bound on the Lyapunov drift. As a result, the BP scheduling is formulated based on link capacities, and the BP forwarding spreads the maximum number of packets along the activated links. To the best of our knowledge, this is the first time a network controller genuinely minimizes the Lyapunov drift (via maximizing the $J(\mathbb{f})$ functional), rather than merely pushing down an *upper-bound* on the drift. As a result, the HD scheduling is formulated based on link actual-transmissions, and the HD forwarding is controlled by link queue differentials.

VI. Congestion Minimizing Control Policy

As the major result of this paper, this section shows that HD control minimizes the average network delay in the class of all control algorithms that act based only on current queue congestion and current channel states. More precisely, it is shown that in the aforementioned class of control algorithms, HD solves the following optimization problem:

$$\begin{aligned}\text{Minimize:}\quad & \overline{Q} := \sum\nolimits_{i\in\mathcal{V}}\sum\nolimits_{d\in\mathcal{K}} \overline{q_i^{(d)}}\\ \text{Subject to:}\quad & \text{Network constraints}\end{aligned} \quad (20)$$

without requiring the knowledge of topology structure, arrival statistics, or channel state probabilities.

Remark 7: By Little's Theorem [1], for a given arrival rate, expected time average total queue congestion $\overline{Q}$ is proportional to long-term average end-to-end network delay. Hence, minimizing $\overline{Q}$ indeed ensures minimizing average network delay.

Prior to formulating the main result in Th. 3, we propose the following lemma which is used in the proof. The lemma implies that any stabilizing network control that results in a higher average total variance of link forwardings will necessarily lead to a higher average total covariance between link forwardings and link queue differentials as well.

Lemma 1: Suppose that a general control policy stabilizes an arrival rate vector $\overline{\boldsymbol{a}_{\circ}}$ resulting in timeslot queue occupancies $\boldsymbol{q}_{\circ}(n)$ and link actual-transmissions $\mathbb{f}(n)$. Then

$$2\,\overline{\mathbb{C}\mathrm{ov}\{\mathbb{B}_{\circ}^{\top}\boldsymbol{q}_{\circ},\mathbb{f}\}} - \overline{\mathbb{V}\mathrm{ar}\{\mathbb{B}_{\circ}\mathbb{f}\}} = \overline{\mathbb{V}\mathrm{ar}\{\boldsymbol{a}_{\circ}\}} \quad (21)$$

where for two vector random variables $\boldsymbol{X}$ and $\boldsymbol{Y}$ we define $\mathbb{C}\mathrm{ov}\{\boldsymbol{X},\boldsymbol{Y}\} := \mathbb{E}\{\boldsymbol{X}^{\top}\boldsymbol{Y}\} - \mathbb{E}\{\boldsymbol{X}\}^{\top}\mathbb{E}\{\boldsymbol{Y}\}$ and $\mathbb{V}\mathrm{ar}\{\boldsymbol{X}\} := \mathbb{C}\mathrm{ov}\{\boldsymbol{X},\boldsymbol{X}\}$, and where the overbar notation denotes the $\limsup$ expected time average as defined in (1).

Theorem 3: Suppose that arrivals and channel states are i.i.d. over timeslots and with respect to each other, and that a node cannot transmit to more than one neighbor at a timeslot. Consider a class of network controllers that act based only on current queue congestion and current channel states. Within this class, the HD control policy solves the average network delay minimization problem (20). Specifically, expected time average number of waited packets is minimized by the HD algorithm of Sec. III-A on a uniclass network, by the HD algorithm of Sec. III-B on a multiclass network subject to transmitting from at most one class per link per timeslot, and by the HD algorithm of Sec. III-C on a multiclass network that allows transmitting from multiple classes per link per timeslot.

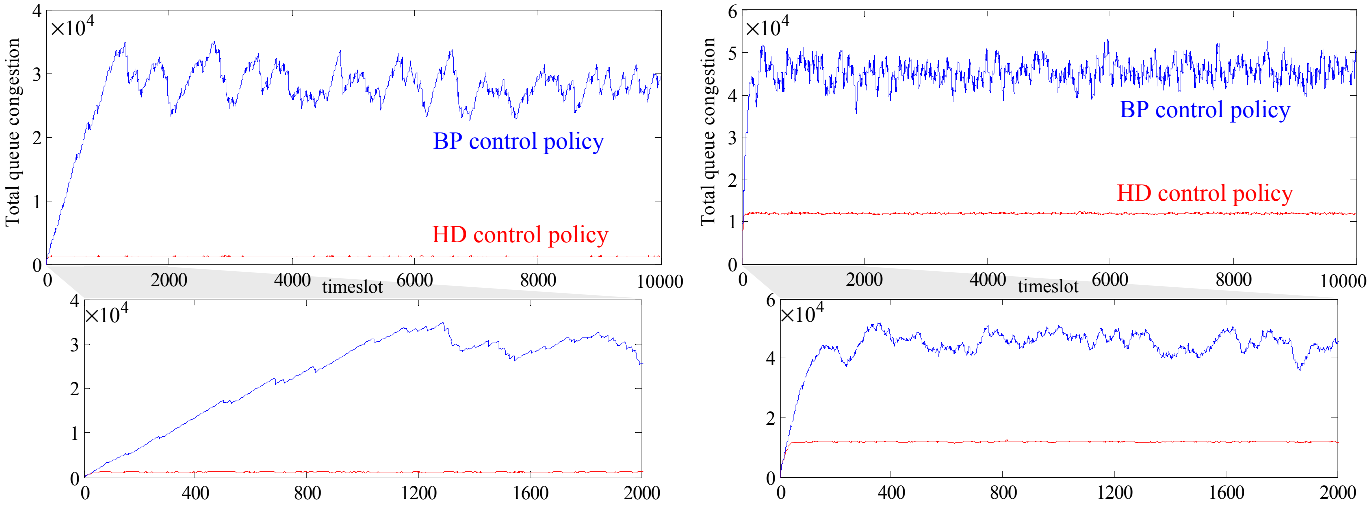


Fig. 2. Timeslot evolution of total number of packets in the sensor network controlled by HD compared with that by BP for $\lambda = 1$ (left) and $\lambda = 10$ (right). The bottom panels are zoomed in the 0-2000 timeslot interval, emphasizing the transient performance of the two control policies.

VII. Simulation Results

We consider a wireless network with 50 nodes randomly distributed on a surface. Links are placed between every two nodes whose proximity distance is less than a threshold, and extra links are added to make the network connected. Links are considered as two-way wireless channels, i.e. for any directed link $ij \in \mathcal{E}$ there exists $ji \in \mathcal{E}$ with the same capacity. The network runs under 1-hop interference model, i.e. links with common node cannot transmit at the same time.

Every timeslot, the capacity of each link ij follows a Gaussian distribution with the mean m_{ij} and the variance equal to 150. To assign m_{ij} to different links, we adopt Shannon capacity with power transmission P_{ij}, noise intensity N_{ij}, and a bandwidth of 1500, viz. $m_{ij} = 1500 \log_2(1 + P_{ij}/N_{ij})$. We randomly assign a noise intensity $Nij \in [1, 5]$ to each link at first and keep it fixed during the simulation.

A. Simulation Results for Uniclass Network

In this simulation we assume that each node is a sensor that sends packets to a unique destination. For this low power sensor network, we take $P_{ij} = 2$ for all links. Exogenous arrivals follow Poisson distribution with parameter λ, which are i.i.d. over timeslots and with respect to each other.

Figure 2 displays timeslot evolution of total number of packets for two Poisson parameters $\lambda = 1$ and $\lambda = 10$. Notice the small steady-state oscillations in HD contrary to large variations in BP that approves the efficiency of (i) taking link weights as quadratic in queue differential, (ii) scheduling links based on actual transmittable packets rather than link capacities, and (iii) restricting packet forwarding to link queue differential rather than spreading the most possible number of packets along the activated links. Also notice the fast transient time in HD with about 50 timeslots for both $\lambda = 1$ and $\lambda = 10$, compared with the slow transient time in BP with about 1200 timeslots for $\lambda = 1$ and 250 timeslots for $\lambda = 10$.

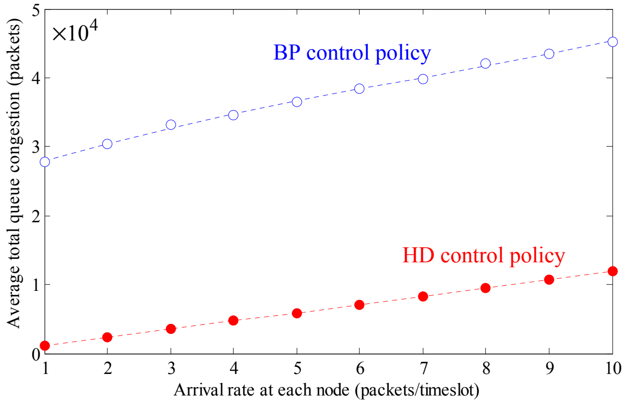


Fig. 3. Expected time average total number of packets in the sensor network against the exogenous arrival rates changing from $\lambda = 1$ to $\lambda = 10$.

Figure 3 displays the average total queue congestion as a function of arrival rate (Poisson parameter), comparing the performance of HD control with that of BP control. Each circle represents the average total number of packets in the sensor network while λ increases from 1 to 10 in unit steps. The average is taken on the last 40000 slots, when the system runs for 50000 slots starting from zero initial condition. Dashed lines display third degree polynomial interpolation.

B. Simulation Results for Multiclass Network

On the same simulation testbench, assume that every node sends packets to every other node, forming a multiclass, multihop wireless network. Different classes are generated at

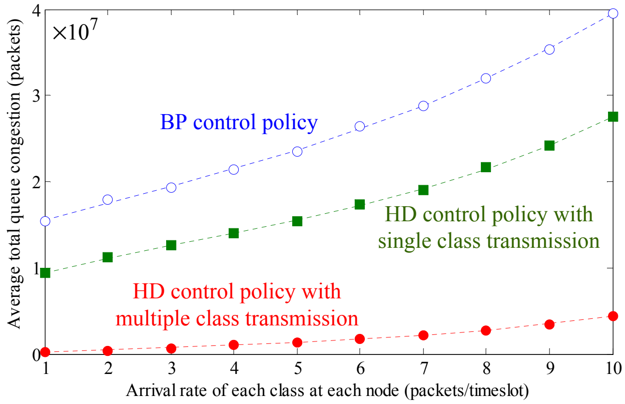


Fig. 4. Expected time average total number of packets in the multiclass network against the exogenous arrival rates changing from $\lambda = 1$ to $\lambda = 10$.

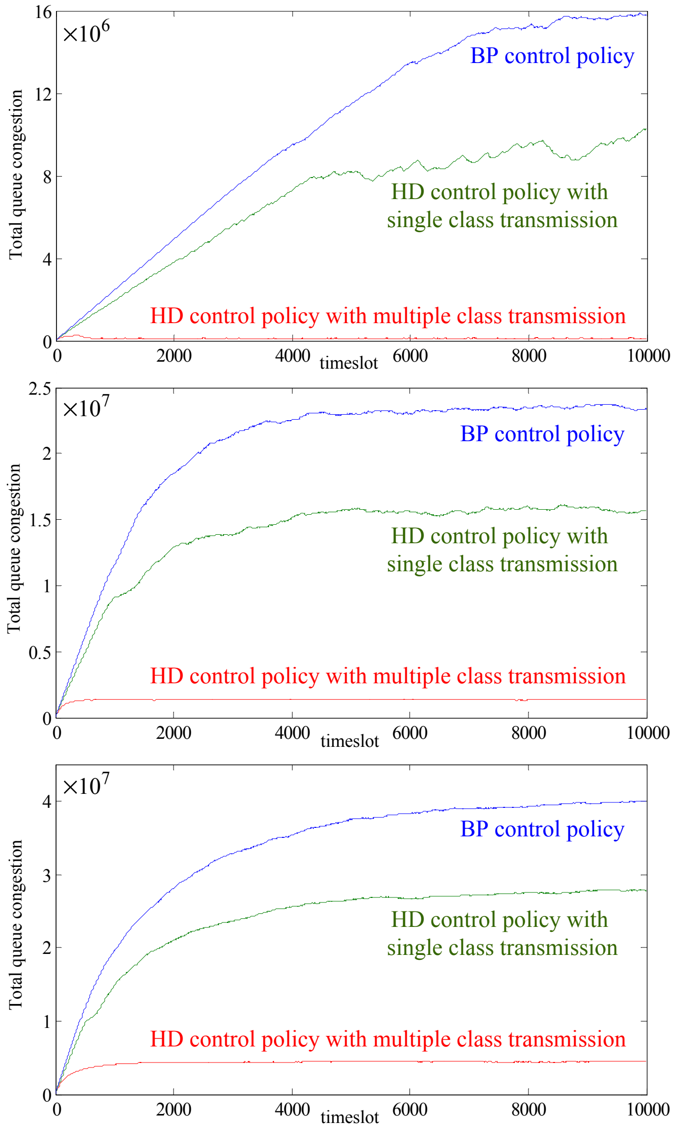


Fig. 5. Timeslot evolution of total number of packets in the multiclass network for $\lambda = 1$ (top), $\lambda = 5$ (middle) and $\lambda = 10$ (bottom).

each node following Poisson random variables with parameter λ, where all of them are i.i.d. over timeslots and with respect to each other. To support this traffic, we assume that each node can expend 30 units of transmission power per timeslot, which under 1-hop interference model leads to $P_{ij} = 30$.

Time average performances of the three control policies are compared in Fig. 4 for λ growing from 1 to 10 in unit steps. The average is taken on the last 40000 slots, when the system runs for 50000 slots starting from zero initial condition, and the dashed lines display third degree polynomial interpolation. For $\lambda = 1$, average total number of packets under the HD algorithm of Sec. III-C is only 220K packets, compared with 15400K packets under the BP algorithm, and 9380K packets under the HD algorithm of Sec. III-B. This difference in performance gets even larger by the growth of λ.

Figure 5 displays timeslot evolution of total number of packets for three arrival rates of $\lambda = 1$, $\lambda = 5$ and $\lambda = 10$ packets per timeslot. Like the uniclass case, the HD with multiple class transmission shows a remarkably better performance in terms of both steady-state oscillations and transient time.

VIII. Conclusion

For stochastic multiclass wireless networks with channel interference and time-varying topology, we developed a dynamic routing control that requires no knowledge of statistics and probabilities in the system. It is throughput-optimal and minimizes average network delay within the class of all routing algorithms that perform based only on current queue congestion and current channel states. This important class includes all opportunistic max-weight schedulers that do not incorporate the Markov structure of topology process into their decisions, among which is Back-Pressure and most of its derivations. It also includes all stationary randomized algorithms that make a routing decision as a pure (possibly randomized) function only of current channel states, typically requiring the perfect knowledge of arrival and channel probabilities.

References

[1] D. Bertsekas and R. Gallager, *Data Networks*. Prentice-Hall, 1992.

[2] R. Berry and R. Gallager, “Communication over fading channels with delay constraints,” *Tr. Info. Theo.*, 48:1135–1149, 2002.

[3] L. Georgiadis, M. Neely, and R. Tassiulas, “Resource allocation and cross-layer control in wireless networks,” *Found. Tren. Net.*, 2006.

[4] M. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan & Claypool, 2010.

[5] L. Tassiulas and A. Ephremides, “Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks,” *Tr. Autom. Cont.*, 37:1936–1949, 1992.

[6] R. Banirazi, E. Jonckheere, and B. Krishnamachari, “Heat diffusion algorithm for resource allocation and routing in multihop wireless networks,” in *Globecom*, 2012.

[7] R. Banirazi, E. Jonckheere, and B. Krishnamachari, “Heat-diffusion: Pareto optimal dynamic routing for time-varying wireless networks,” in *INFOCOM*, 2014.

[8] L. Bui, R. Srikant, and A. Stolyar, “Optimal resource allocation for multicast flows in multihop wireless networks,” *Perf. Eval. Rev.*, 2007.

[9] L. Bui, R. Srikant, and A. Stolyar, “A novel architecture for delay reduction in back-pressure scheduling algorithm,” *Tr. Net.*, 19:1597–1609, 2011.

[10] B. Ji, C. Joo, and N. Shroff, “Delay-based back-pressure scheduling in multi-hop wireless networks,” in *INFOCOM*, 2011.

[11] L. Huang, S. Moeller, M. Neely, and B. Krishnamachari, “Lifo-backpressure achieves near optimal utility-delay tradeoff,” *Tr. Net.*, 21:831–844, 2013.

[12] M. Alresaini, M. Sathiamoorthy, B. Krishnamachari, and M. Neely, “Backpressure with adaptive redundancy (bwar),” in *INFOCOM*, 2012.

[13] E. Stai, J. Baras, and S. Papavassiliou, “Throughput-delay tradeoff in wireless multi-hop networks via greedy hyperbolic embedding,” in *MTNS*, 2012.

[14] D. Xue, R. Murawski, and E. Ekici, “Distributed utility-optimal scheduling with finite buffers,” in *WiOpt*, 2012.

[15] A. Gupta, X. Lin, and R. Srikant, “Low-complexity distributed scheduling algorithms for wireless networks,” in *INFOCOM*, 2007.

[16] L. Bui, S. Sanghavi, and R. Srikant, “Distributed link scheduling with constant overhead,” *Tr. Net.*, 17:1467–1480, 2009.

[17] L. Jiang and J. Walrand, “Approaching throughput-optimality in distributed csma scheduling algorithms with collisions,” *Tr. Net.*, 19:816–829, 2011.

[18] J. Ryu, V. Bhargava, N. Paine, and S. Shakkottai, “Back-pressure routing and rate control for icns,” in *Mobicom*, 2010.

[19] J. Martinez and J. Bafalluy, “Design, implementation, tracing of dynamic backpressure routing for ns-3,” in *SIMUTools*, 2011.

[20] S. Moeller, A. Sridharan, and O. G. B. Krishnamachari, “Routing without routes: the backpressure collection protocol,” in *IPSN*, 2010.

[21] R. Banirazi, E. Jonckheere, and B. Krishnamachar, “Minimum delay in class of throughput-optimal control policies on wireless networks.” Available at http://eudoxus2.usc.edu, USC, 2014.

[22] D. Shah and D. Wischik, “Optimal scheduling algorithms for input-queued switches,” in *INFOCOM*, 2006.

[23] A. Gupta, L. Xiaojun, and R. Srikant, “Low-complexity distributed scheduling algorithms for wireless networks,” 17:1846–1859, 2009.