

Enhanced Backpressure Routing Using Wireless Link Features

Zhongyuan Zhao*, Gunjan Verma[‡], Ananthram Swami[‡], and Santiago Segarra*
*Rice University, USA [‡]US Army's DEVCOM Army Research Laboratory, USA

Abstract—Backpressure (BP) routing is a well-established framework for distributed routing and scheduling in wireless multi-hop networks. However, the basic BP scheme suffers from poor end-to-end delay due to the drawbacks of slow startup, random walk, and the last packet problem. Biased BP with shortest path awareness can address the first two drawbacks, and sojourn time-based backlog metrics were proposed for the last packet problem. Furthermore, these BP variations require no additional signaling overhead in each time step compared to the basic BP. In this work, we further address three long-standing challenges associated with the aforementioned low-cost BP variations, including optimal scaling of the biases, bias maintenance under mobility, and incorporating sojourn time awareness into biased BP. Our analysis and experimental results show that proper scaling of biases can be achieved with the help of common link features, which can effectively reduce end-to-end delay of BP by mitigating the random walk of packets under low-to-medium traffic, including the last packet scenario. In addition, our low-overhead bias maintenance scheme is shown to be effective under mobility, and our bio-inspired sojourn time-aware backlog metric is demonstrated to be more efficient and effective for the last packet problem than existing approaches when incorporated into biased BP.

Index Terms—Backpressure routing, resource allocation, shortest path distance, queue length, sojourn time, delay-aware routing.

I. INTRODUCTION

Backpressure (BP) routing [1] is a well-established algorithm for distributed routing and scheduling in wireless multi-hop networks. These networks have been widely used in military communications, disaster relief, and wireless sensor networks, and are envisioned to support emerging applications such as connected vehicles, robotic swarms, xG (device-to-device, wireless backhaul, and non-terrestrial coverage), Internet of Things, and machine-to-machine communications [2]–[7]. The distributed nature of the BP algorithms [1], [8]–[22] empowers wireless multi-hop networks to be self-organized without relying on infrastructure, and promotes scalability and robustness that are critical to many applications. In BP, each node maintains a separate queue for packets to each destination (also denominated as commodity), routing decisions are made by selecting the commodity that maximizes the differential backlog between the two ends of each link, and data transmissions are activated on a set of non-interfering links via MaxWeight scheduling [1], [23], [24]. This mechanism uses congestion gradients to drive data packets towards their destinations through all possible routes, avoiding congestion and stabilizing the queues in the network for any flow rate within the network capacity region [1], [8], [9] (this property is known as throughput optimality).

However, classical BP routing is known to suffer from poor delay performance, especially under low-to-medium traffic loads [8]–[12], exhibiting three drawbacks: 1) *slow startup*: when a flow starts, many packets have to be first backlogged to form stable queue backlog-based gradients, causing large initial end-to-end delay; 2) *random*

walk: during BP scheduling, the fluctuations in queue backlogs drive packets towards random directions, causing unnecessarily long routes or loops; and 3) the *last packet problem* [13], [14], which refers to the phenomena that packets of a short-lived flow could remain enqueued in the network for a long time due to the absence of pressure.

To improve the latency, four types of BP variations have been developed: 1) Biased BP adds pre-defined queue-agnostic biases, e.g., (functions of) shortest path distance [8]–[10], [22] to the backlog metric. Shortest path-biased BP (SP-BP) can mitigate the drawbacks of slow startup and random walk, while maintaining the throughput optimality [8]–[10] at a low cost, i.e., a one-time communication overhead for bias computation. 2) Delay-based BP replaces queue-length with delay metrics [11], [14], [16] for the backlog. This approach addresses the last packet problem and maintains throughput optimality. 3) Impose restrictions on the routes [17], [18] or hop counts [19] to prevent or reduce loops. However, approaches (2) and (3) above may shrink the network capacity region. 4) Use queue-dependent biases that aggregate the queue state information (QSI) of the local neighborhood (or global QSI) to improve the myopic BP decisions [11], [12] or use shadow queues [13], [15] to dynamically increase the backpressure. However, collecting neighborhood or global QSI at every time step increases communications overhead. Besides latency, other practical enhancements include extending BP to intermittently connected networks [13], [21] and uncertain network states [20]. In addition, most of the aforementioned approaches require careful parameter tuning, typically done via trial-and-error.

In this work, we aim to improve the SP-BP routing [8], [9], [22] in the presence of the last packet problem and network mobility, while retaining its low computational and communication overhead and throughput optimality. Specifically, we seek to address three longstanding challenges associated with SP-BP: 1) how to optimize the per-hop distance, in other words, the scaling of the shortest path bias; 2) how to efficiently update the shortest path bias under node mobility, e.g., nodes moving, joining or leaving the network; and 3) how to effectively incorporate sojourn time awareness into SP-BP in response to the last packet problem.

Contribution. The contributions of this paper are as follows:

- 1) We develop a principled approach to optimize the per-hop distance metric (bias scaling) based on features of wireless links, to mitigate the random walk behavior and the last packet problem in SP-BP.
- 2) We develop an efficient bias maintenance rule for node mobility based on local information exchange between 1-hop neighbors.
- 3) We further propose a new delay metric, $expQ$, to prioritize old packets without having to track the sojourn time of individual packets, which improves the bandwidth efficiency of the network.
- 4) We demonstrate via numerical experiments that shortest path biases can address all three drawbacks of basic BP and are more effective than delay metrics [14], [16] in reducing end-to-end delay.

II. (BIASED) BACKPRESSURE ROUTING

System Model: We model a wireless multi-hop network as an undirected graph $\mathcal{G}^n = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} is a set of nodes representing user devices in the network, and \mathcal{E} represents a set of links, where

arXiv:2310.04364v2 [cs.NI] 14 Oct 2023

Research was sponsored by the Army Research Office and was accomplished under Cooperative Agreement Number W911NF-19-2-0269. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

Emails: *{zhongyuan.zhao, segarra}@rice.edu, [‡]{gunjan.verma.civ, ananthram.swami.civ}@army.mil

$e = (i, j) \in \mathcal{E}$ for $i, j \in \mathcal{V}$ represents that node i and node j can directly communicate. \mathcal{G}^n is called a connectivity graph and assumed to be a connected graph, i.e., two arbitrary nodes in the network can always reach each other. Notice that routing involves directed links, so we use (\vec{i}, \vec{j}) to denote data packets being transmitted from node i to node j over link (i, j) . There is a set of flows \mathcal{F} in the network, in which a flow $f = (i, c) \in \mathcal{F}$, where $i \neq c$ and $i, c \in \mathcal{V}$, describes the stream of packets from a source node i to a destination node c , potentially through multiple links. The medium access control (MAC) of the wireless network is assumed to be time-slotted orthogonal multiple access. Each time slot t contains a stage of decision making for routing and scheduling, followed by a second stage of data transmission. Each node hosts multiple queues, one for each flow destined to node $c \in \mathcal{V}$ (or packets of commodity c). We use $\mathbf{Q}_i^{(c)}(t)$ to denote the queue of commodity c at node i at the beginning of time slot t , and $Q_i^{(c)}(t)$ for its queue length.

Matrix $\mathbf{R} \in \mathbb{Z}_+^{|\mathcal{E}| \times T}$ collects the (stochastic) real-time link rates, of which an element $\mathbf{R}_{e,t}$ represents the number of packets that can be delivered over link e in time slot t . The long term link rate of a link $e \in \mathcal{E}$ is denoted by $r_e = \mathbb{E}_{t \leq T} [\mathbf{R}_{e,t}]$, and $\bar{r} = \mathbb{E}_{e \in \mathcal{E}, t \leq T} [\mathbf{R}_{e,t}]$ is the network-wide average link rate.

Under orthogonal multiple access, the conflict relationships between wireless links are captured by a *conflict graph*, $\mathcal{G}^c = (\mathcal{E}, \mathcal{C})$, defined as follows: a vertex $e \in \mathcal{E}$ represents a link in the network, and the presence of an undirected edge $(e_1, e_2) \in \mathcal{C}$ means that simultaneous communications on links $e_1, e_2 \in \mathcal{E}$ cause interference. We consider the scenario where all the users transmit at identical power levels with an omnidirectional antenna, which can be captured by the unit-disk interference model, in which two links conflict with each other if they share the same node or if any of their nodes are within a pre-defined distance. For the rest of this paper, we assume the conflict graph \mathcal{G}^c to be known, e.g., by each link monitoring the wireless channel [25].

BP Algorithm: BP routing and scheduling consist of 4 steps. In step 1, the optimal commodity $c_{ij}^*(t)$ on each directed link (\vec{i}, \vec{j}) is selected as the one with the maximal backpressure,

$$c_{ij}^*(t) = \operatorname{argmax}_{c \in \mathcal{V}} \{U_i^{(c)}(t) - U_j^{(c)}(t)\}, \quad (1)$$

where $U_i^{(c)}(t)$ is the backlog metric, whose relationship with queue lengths is discussed later. The backpressure of commodity c on directed link (\vec{i}, \vec{j}) is defined as $U_{ij}^{(c)}(t) = U_i^{(c)}(t) - U_j^{(c)}(t)$. In step 2, the maximum backpressure of (\vec{i}, \vec{j}) is found as:

$$w_{ij}(t) = \max\{U_i^{(c_{ij}^*(t))}(t) - U_j^{(c_{ij}^*(t))}(t), 0\}. \quad (2)$$

In step 3, MaxWeight scheduling [1] finds the schedule $\mathbf{s}(t) \in \{0, 1\}^{|\mathcal{E}|}$ to activate a set of *non-conflicting links* achieving the maximum total utility, where the per-link utility is $u_{ij}(t) = \mathbf{R}_{ij,t} \tilde{w}_{ij}(t)$,

$$\mathbf{s}(t) = \operatorname{argmax}_{\tilde{\mathbf{s}}(t) \in \mathcal{S}} \tilde{\mathbf{s}}(t)^\top [\mathbf{R}_{*,t} \odot \tilde{\mathbf{w}}(t)], \quad (3)$$

in which \mathcal{S} denotes the set of all non-conflicting configurations, vector $\mathbf{R}_{*,t}$ collects the real-time link rate of all links, vector $\tilde{\mathbf{w}}(t) = [\tilde{w}_{ij}(t)]_{(i,j) \in \mathcal{E}}$, where $\tilde{w}_{ij} = \max\{w_{ij}(t), w_{ji}(t)\}$, and the direction of the link selected by the max function will be recorded for step 4. MaxWeight scheduling involves solving an NP-hard maximum weighted independent set (MWIS) problem [26] on the conflict graph to find a set of non-conflicting links. In practice, (3) can be solved approximately by distributed heuristics, such as local greedy scheduler (LGS) [23] and its GCN-based enhancement [25]. In step 4, all of the real-time link rate $\mathbf{R}_{ij,t}$ of a scheduled link is

allocated to its optimal commodity $c_{ij}^*(t)$. The final transmission and routing variables of commodity $c \in \mathcal{V}$ on link (\vec{i}, \vec{j}) is

$$\mu_{ij}^{(c)}(t) = \begin{cases} \mathbf{R}_{ij,t}, & \text{if } c = c_{ij}^*(t), w_{ij}(t) > 0, s_{ij}(t) = 1, \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

(Biased) backlog metrics: The general form of the backlog metric, i.e., $U_i^{(c)}(t)$ in (1), in low-cost BP schemes can be expressed as

$$U_i^{(c)}(t) = g(Q_i^{(c)}(t)) + B_i^{(c)}, \quad (5)$$

where $g(\cdot)$ is a function of QSI, and $B_i^{(c)}$ is a queue-agnostic bias. In queue length-based BP [1], [8]–[10], [22], $g(Q_i^{(c)}(t)) = Q_i^{(c)}(t)$, whereas in delay-based BP, $g(Q_i^{(c)}(t))$ can be the sojourn time of the head-of-line (HOL) packet [14] or the entire queue, i.e., sojourn time backlog (SJB) [16]. In unbiased BP, such as the basic and delay-based BP, $B_i^{(c)} = 0$. In biased BP [8]–[10], [22], $B_i^{(c)} \geq 0$ is defined based on the shortest path distance between node i and destination c . The set of biases is denoted as $\mathcal{B} = \{B_i^{(c)} | i, c \in \mathcal{V}\}$.

The shortest path distances can be computed by distributed algorithms for single source shortest path (SSSP) or all pairs shortest path (APSP) on the initialization of a node or network, and updated periodically to capture the evolving network topology. On weighted graphs, the distributed SSSP with the Bellman-Ford algorithm [27], [28] and state-of-the-art APSP [29] both take $\mathcal{O}(|\mathcal{V}|)$ rounds. On unweighted graphs, the distributed SSSP and APSP take $\mathcal{O}(D)$ and $\mathcal{O}(|\mathcal{V}|)$ rounds, respectively, where D is the diameter of the unweighted \mathcal{G}^n .

III. IMPROVEMENTS TO BIASED BACKPRESSURE

We focus on SP-BP routing [8], [22], where the length of every edge $e \in \mathcal{E}$ in the shortest path computation, namely the *per-hop distance*, denoted as δ_e , is scaled by a constant. For example, $\delta_e = K$ for all $e \in \mathcal{E}$ in enhanced dynamic routing (EDR) [8] and $\delta_e = R/(x_e r_e)$ in [22], where x_e is the link duty cycle estimated by a graph convolutional network (GCN), and R is the scaling constant. Our methods are to i) optimize the scaling constant in the per-hop distance (K in [8] and R in [22]), ii) efficiently update the biases under node mobility, and iii) prioritize older packets over newer ones.

Optimal scaling of per-hop distance. To the best of our knowledge, the parameter(s) for the scaling of the per-hop distance in biased BP are often selected via trial-and-error, whereas our goal is to provide a formal method of optimization for such scaling parameter(s).

A key observation is that the scaling of the per-hop distance in biased BP influences the degree of random walk of packets and subsequently, the end-to-end delay, under low to medium traffic loads, including the last packets situation. Consider an exemplary situation of last packets in BP routing with only one commodity (flow), for which the queue states of four nodes in two consecutive time steps are illustrated in Fig. 1. The hop distances of nodes A, B, C, D to the destination are $b + 2, b + 1, b + 1, b$, respectively. Every link has a constant link rate of \bar{r} . At time t , $Q_A^{(c)}(t) = \bar{r}$ and $Q_B^{(c)}(t) = Q_C^{(c)}(t) = 0$, the link pressures point from A towards B and C . Assume that at time $t + 1$, all the packets on A moved to B . Different choices of δ_e would lead to different routing decisions at $t + 1$.

For unbiased BP, i.e., $\delta_e = 0$, the backpressures (orange arrows) are pointed from B towards A, C , and D , and $U_{BA}^{(c)}(t+1) = U_{BC}^{(c)}(t+1) = \bar{r} > U_{BD}^{(c)}(t+1)$. At the end of time slot $t + 1$, the packets on B will move to either A or C , packets on D may also move back to C , causing an unwanted meandering of the packets. For biased

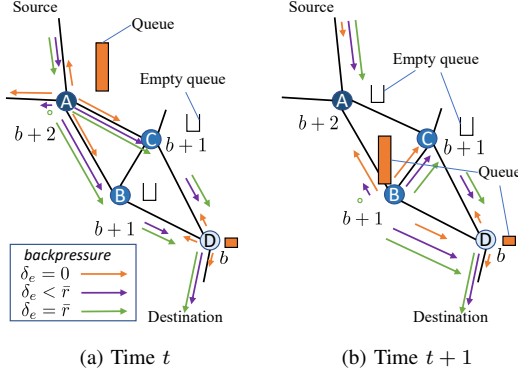


Fig. 1: The queue states at four nodes (with hop distance to destination marked) in an exemplary case of backpressure routing with a single commodity, at time (a) t and (b) $t+1$. An arrow indicates the magnitude and direction of pressure on a link, its color encodes the choice of per-hop distance. All links have an identical rate of \bar{r} .

BP, i.e., $\delta_e > 0$, the backpressures (magenta and green arrows) at time $t+1$ originating from B are skewed towards the destination, $U_{BD}^{(c)}(t+1) > U_{BC}^{(c)}(t+1) = \bar{r} > U_{BA}^{(c)}(t+1)$. At the end of time slot $t+1$, packets in B will move forward to D . In particular, when $\delta_e = \bar{r}$, the backpressure (i.e., green arrows) $U_{BA}^{(c)}(t+1) = 0$.

The previous example can be formally described as follows: in a wireless multi-hop network with homogeneous link rate, i.e., all links have a link rate of \bar{r} , there are two links (i, j) and (j, k) , where nodes j, k are on the shortest path from node i to node c ; consider a congestion-free last packets (CFLP) scenario: at time t , the last packets of commodity c reside on node i , and link (i, j) is congestion-free, i.e., $0 < Q_i^{(c)}(t) = q \leq \bar{r}$, $Q_j^{(c)}(t) = 0$ and no external packets of commodity c will arrive at nodes i, j, k from the rest of the network or users. If link (i, j) is scheduled at time t , such that $Q_i^{(c)}(t+1) = 0$, $Q_j^{(c)}(t+1) = q \leq \bar{r}$, we have the following.

Lemma 1. *In SP-BP routing under the CFLP scenario, the per-hop distance δ_e should be greater than or equal to the homogeneous link rate, $\delta_e \geq \bar{r}$, to avoid the immediate reversal of the direction of backpressure for commodity c on the scheduled link $(\overline{i, j})$ after the transmission.*

Proof. In queue length-based SP-BP, the backpressure on link (i, j) is $U_{ij}^{(c)}(t) = Q_{ij}^{(c)}(t) + (B_i^{(c)} - B_j^{(c)}) = q + \delta_e$. Since link (i, j) is scheduled at time t , we have that $U_{ij}^{(c)}(t+1) = -q + \delta_e$. To avoid the direction of the backpressure on link (i, j) being reversed at time $t+1$, i.e., $U_{ij}^{(c)}(t+1) < 0$, we need to set $\delta_e \geq \bar{r}$. \square

In the described CFLP scenario, Lemma 1 ensures that the backpressure algorithm does not oscillate as the last packets travel through the shortest path when $\delta_e \geq \bar{r}$. On the other hand, a smaller value of δ_e is preferable for path finding and congestion prevention. Indeed, an extremely large SP bias scaling δ_e would force every packet through the shortest path, thus hindering any information from the queue lengths and defeating the original purpose of BP routing. Combining these two observations, we advocate for setting the minimal per-hop distance as the average link rate, i.e., $\min_{e \in \mathcal{E}} \delta_e := \bar{r}$. In particular, for EDR in [8], this implies setting $\delta_e = K := \bar{r}$.

Although the above discussion is based on the CFLP setting with all link rates equal to the average \bar{r} , in Section IV we demonstrate

the experimental optimality of the above choice under heterogeneous link rates and general traffic settings.

Bias maintenance. Due to the mobility and distributed nature of networks, bias maintenance requires frequent or periodical SSSP and/or APSP computation. Although the biases \mathcal{B} can be re-used for many time slots to match the slowly changing topology, the overhead of bias maintenance is still high for large networks.

To address this issue, we propose a neighborhood update rule for bias maintenance at node $i \in \mathcal{V}$, whenever one or more of its incidental links are established or destroyed

$$B_i^{(c)}(t+1) = \begin{cases} \min_{j \in \mathcal{N}^{(i)}} [B_j^{(c)}(t) + \delta_{ij}(t)], & i \neq c \\ 0, & i = c \end{cases} \quad (6)$$

where $\delta_{ij}(t)$ is the per-hop distance between neighboring nodes i and j . This rule allows the shortest path bias to be updated within $\mathcal{O}(D)$ steps of local message exchange.

Sojourn time-aware backlog (expQ). To prioritize old packets, we further introduce a delay-aware QSI function $g(\cdot)$ in (5):

$$g(Q_i^{(c)}(t+1)) = (1 + \epsilon)g(Q_i^{(c)}(t)) \left[1 - \frac{\Delta Q_{i, Tx}^{(c)}(t+1)}{Q_i^{(c)}(t)} \right] + \Delta Q_{i, Rx}^{(c)}(t+1) \quad (7)$$

where $\epsilon \geq 0$ is a small constant parameter, $\Delta Q_{i, Tx}^{(c)}(t)$ and $\Delta Q_{i, Rx}^{(c)}(t)$ are the numbers of packets that left and arrived at the queue $Q_i^{(c)}$ at node i at time t . For $\epsilon = 0$, $g(Q_i^{(c)}(t)) = Q_i^{(c)}(t)$.

The $g(\cdot)$ in (7) is named as expQ, as it will increase exponentially over time if commodity c at node i is not scheduled. Compared to the backpressure metric based on sojourn time backlog [16] and HOL sojourn time [14], expQ can increase the bandwidth utilization efficiency, since it does not require tracking the sojourn time of individual packets.

Throughput optimality. Queue length-based biased BP schemes with queue-agnostic biases automatically inherit the throughput optimality of basic BP as long as the biases are non-negative and queue-agnostic, according to the proof in [8].

IV. NUMERICAL EXPERIMENTS

We evaluate various low-cost BP routing algorithms on simulated wireless multi-hop networks, generated by a 2D point process. The simulations are configured similar to those in [22], with only three differences: a) Each test instance contains a number (uniformly chosen between $[0.30|\mathcal{V}|]$ and $[0.50|\mathcal{V}|]$) of random flows. b) Only a unit-disk interference model is used. c) Two types of traffic are employed: Under streaming traffic, the arrival rate of a flow $\lambda(f) \sim \mathbb{U}(0.2, 1.0)$ for all time slots. Under bursty traffic, the flow arrival rate $\lambda(f) \sim \mathbb{U}(2.0, 10.0)$ for $t < 30$, and no packet arrivals for $t \geq 30$. The end-to-end delay of a test BP scheme is collected by tracking the time each packet arrived at the source node t_0 and the destination node (departure time) t_1 . To be conservative, we treat the delay of an undelivered packet as $T - t_0$.

The tested BP algorithms (acronyms in parentheses identify legends in Fig. 2) include 1) unbiased BP: vanilla BP (BP), delay-based unbiased BP with SJB [16] (BP-SJB) and HOL [14] (BP-HOL); 2) queue-length based biased BP: EDR [8] with per-hop distance $\delta_e = \bar{r}$ (EDR- \bar{r}), and GCN-based delay-aware shortest path bias [22] (SP- $\bar{r}/(xr)$) with per-hop distance $\delta_e = \bar{r}/(x_e r_e)$, and its scaled version (SP- $\bar{r}/(xr)$ -min) with per-hop distance $\tilde{\delta}_e = \delta_e \bar{r} / (\min_{e \in \mathcal{E}} \delta_e)$; and

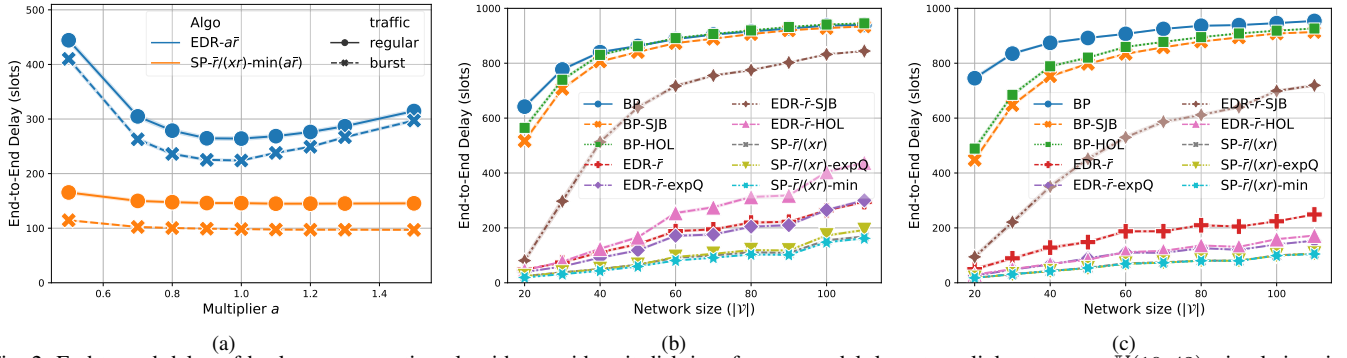


Fig. 2: End-to-end delay of backpressure routing algorithms, with unit-disk interference model, long-term link rates $r_e \sim \mathbb{U}(10, 42)$, simulation time steps $T = 1000$, and 100 test instances per point (10 random networks \times 10 realizations of random source-destination pairs and random link rates). (a) Delay vs. varying minimal per-hop distance multiplier $\min(\delta_e) = a\bar{r}$ for SP-BP on networks of 100 nodes. (b) Delay under streaming traffic, and (c) Delay under bursty traffic, v.s. network size (of 20-110 nodes). Zoom in to view the error band (95% confidence interval) in light colors.

3) delay-based biased BP: combination of BP-SJB and EDR- \bar{r} (EDR- \bar{r} -SJB), combination of BP-HOL and EDR- \bar{r} (EDR- \bar{r} -HOL), and the two biased BP schemes based on expQ (EDR- \bar{r} -expQ and SP- $\bar{r}/(xr)$ -expQ) where $\epsilon = 0.01$ in (7). The end-to-end delay of the tested BP schemes are illustrated in Figs. 2, where each point in the plot is obtained by averaging 100 test instances, specifically, on 10 random networks of a particular size, each with 10 realizations of random source-destination pairs and random link rates.

Optimal per-hop distance adjustment. We test EDR- δ and SP- $\bar{r}/(xr)$ -min with per-hop distance adjustment $\min_{e \in \mathcal{E}} \delta_e := a\bar{r}$, for $a \in [0.5, 1.5]$, on random networks of 100 nodes. The results in Fig. 2(a) show that $a = 1.0$ is indeed the optimal setting for EDR- δ and near-optimal setting for SP- $\bar{r}/(xr)$ -min, under streaming and bursty traffics. This validates the optimality of adjustment rule $\min_{e \in \mathcal{E}} \delta_e := \bar{r}$ advocated in Section III. Notice that the settings tested in Fig. 2(a) go beyond the simplified setting of CFLP with constant rates treated in Section III. However, the proposed scaling is still empirically optimal in these broader scenarios. Since SP- $\bar{r}/(xr)$ -min can tolerate a wide range of a , to keep its distributed execution, it can be implemented based on statistical information of per-hop distances rather than their global minimal. The different sensitivities of the end-to-end delay to the scaling of per-hop distance under the two tested SP-BP schemes, as shown in Fig. 2(a), reveal that the shortest path based on link features [22] not only achieves better performance but is also more robust to the scaling choice.

Streaming traffic. The average end-to-end delays (in time slots) of the 10 routing algorithms under streaming traffic, as a function of the network size are presented in Fig. 2(b). Most BP variations outperform the vanilla BP. However, the benefit of delay-based unbiased BP (BP-SJB and BP-HOL) diminishes on larger networks, as unbiased BP schemes still suffer from the drawbacks of slow startup and random walk, leading to very low delivery rates.

Biased BP based on queue length (EDR- \bar{r} , SP- $\bar{r}/(xr)$, and SP- $\bar{r}/(xr)$ -min) can significantly reduce the end-to-end delay, as the pre-defined distance gradients lead to fast startup and mitigated random walk. SP- $\bar{r}/(xr)$ can reduce the delay of EDR- \bar{r} by nearly half, which is consistent with the results in [22]. Notice that SP- $\bar{r}/(xr)$ -min only slightly improves SP- $\bar{r}/(xr)$. Delay metrics such as SJB and HOL do not directly benefit biased BP with our per-hop distance adjustment, which is based on queue length. As shown in Fig. 2(b), EDR- \bar{r} -SJB and EDR- \bar{r} -HOL both significantly underperform EDR- \bar{r} , whereas expQ only slightly improves EDR- \bar{r} for smaller networks ($|\mathcal{V}| \leq 90$), while slightly degrading SP- $\bar{r}/(xr)$.

Bursty traffic. The end-to-end delays of the 10 BP algorithms under bursty traffic are presented in Fig. 2(c), which shows their effectiveness in the last packet problem. Compared to streaming traffic, every BP scheme performs better under bursty traffic, except vanilla BP. EDR- \bar{r} is substantially improved by both HOL and expQ, where the latter slightly outperforms the former. For SP- $\bar{r}/(xr)$, the benefits of per-hop distance adjustment and expQ are negligible since SP- $\bar{r}/(xr)$ already reaches 100% delivery rate. This shows that the last packet problem can be significantly mitigated by our proposed per-hop distance scaling, as EDR- \bar{r} can achieve a high delivery rate of 93.3%, and further improved by combining with HOL and expQ with delivery rate 99.7% and 99.9%, respectively. Compared to HOL, expQ is not degraded under streaming traffic and does not require tracking the sojourn time of individual packets.

TABLE I: End-to-end delay on random networks of 100 nodes, $T = 1000$ under unit-disk interference model and node mobility.

	EDR- \bar{r}		SP- $\bar{r}/(xr)$	
	Delay (std.)	Delivery (std.)	Delay (std.)	Delivery (std.)
Ideal	278.3 (106.1)	81.2% (8.7%)	155.2 (80.9)	90.8% (6.1%)
Neighbor	331.8 (103.8)	75.7% (9.1%)	282.5 (87.5)	78.5% (7.8%)

Node mobility. Lastly, we run 100 test instances on random networks with 100 nodes and node mobility, in which for every 100 time steps, 10 random nodes take a random step ($\Delta \mathbf{x} \sim \mathcal{N}^2(0, 0.1) \in \mathbb{R}^2$) on the 2D plane, while keeping the network connected. The long-term link rate of a newly created link e is set as $r_e \sim \mathbb{U}(10, 42)$. We compare the delay and delivery rate of our neighborhood bias update rule in (6) with those of ideal (instantaneous) SSSP in Table I. Under the practical neighborhood update rule, SP- $\bar{r}/(xr)$ suffers a larger loss from the ideal (but unpractical) SSSP than EDR- \bar{r} , i.e., the delivery rate of SP- $\bar{r}/(xr)$ dropped by 12% compared to a drop of 5.6% with EDR- \bar{r} . Despite larger delays and lower delivery rates, EDR- \bar{r} suffers less under node mobility likely due to its simplicity.

V. CONCLUSIONS

In this paper, we improve the latency and practicality of shortest path-based Backpressure routing by introducing three components: optimal bias scaling, low-overhead bias maintenance, and a delay-aware backlog metric. These enhancements are demonstrated to be effective in wireless multi-hop networks with relatively low mobility. In the future, it is worthwhile to investigate and further improve the performance of our approach in scenarios with varying network mobility, mixed streaming and bursty traffics, as well as uncertainties and shifts in link features and their global minimal values.

REFERENCES

- [1] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. on Automatic Control*, vol. 37, no. 12, pp. 1936–1948, 1992.
- [2] 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, 2006.
- [3] S. K. Sarkar, T. G. Basavaraju, and C. Puttamadappa, *Ad hoc Mobile Wireless Networks: Principles, Protocols and Applications*. CRC Press, 2013.
- [4] A. Kott, A. Swami, and B. J. West, "The internet of battle things," *Computer*, vol. 49, no. 12, pp. 70–75, 2016.
- [5] I. F. Akyildiz, A. Kak, and S. Nie, "6G and beyond: The future of wireless communications systems," *IEEE Access*, vol. 8, pp. 133995–134030, 2020.
- [6] "Cisco annual internet report (2018–2023)," white paper, Cisco Systems, Inc., Mar. 2020. <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>, accessed June 2023.
- [7] X. Chen, D. W. K. Ng, W. Yu, E. G. Larsson, N. Al-Dhahir, and R. Schober, "Massive access for 5G and beyond," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 3, pp. 615–637, 2021.
- [8] M. J. Neely, E. Modiano, and C. E. Rohrs, "Dynamic power allocation and routing for time-varying wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 1, pp. 89–103, 2005.
- [9] L. Georgiadis, M. J. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," *Foundations and Trends® in Networking*, vol. 1, no. 1, pp. 1–144, 2006.
- [10] Z. Jiao, B. Zhang, W. Gong, and H. Mouftah, "A virtual queue-based back-pressure scheduling algorithm for wireless sensor networks," *EURASIP J. on Wireless Commun. and Netw.*, vol. 2015, no. 1, pp. 1–9, 2015.
- [11] Y. Cui, E. M. Yeh, and R. Liu, "Enhancing the delay performance of dynamic backpressure algorithms," *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 954–967, 2016.
- [12] J. Gao, Y. Shen, M. Ito, and N. Shiratori, "Bias based general framework for delay reduction in backpressure routing algorithm," in *Intl. Conf. Computing, Networking and Communications (ICNC)*, pp. 215–219, IEEE, 2018.
- [13] M. Alresaini, K.-L. Wright, B. Krishnamachari, and M. J. Neely, "Back-pressure delay enhancement for encounter-based mobile networks while sustaining throughput optimality," *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 1196–1208, 2016.
- [14] B. Ji, C. Joo, and N. B. Shroff, "Delay-based back-pressure scheduling in multihop wireless networks," *IEEE/ACM Trans. Netw.*, vol. 21, no. 5, pp. 1539–1552, 2012.
- [15] E. Athanasopoulou, L. X. Bui, T. Ji, R. Srikant, and A. Stolyar, "Back-pressure-based packet-by-packet adaptive routing in communication networks," *IEEE/ACM Trans. Netw.*, vol. 21, no. 1, pp. 244–257, 2012.
- [16] L. Hai, Q. Gao, J. Wang, H. Zhuang, and P. Wang, "Delay-optimal back-pressure routing algorithm for multihop wireless networks," *IEEE Trans. Vehicular Tech.*, vol. 67, no. 3, pp. 2617–2630, 2018.
- [17] A. Rai, C.-p. Li, G. Paschos, and E. Modiano, "Loop-free backpressure routing using link-reversal algorithms," *IEEE/ACM Trans. Netw.*, vol. 25, no. 5, pp. 2988–3002, 2017.
- [18] P. Yin, S. Yang, J. Xu, J. Dai, and B. Lin, "Improving backpressure-based adaptive routing via incremental expansion of routing choices," in *ACM/IEEE Symp. Arch. for Networking and Communications Systems (ANCS)*, pp. 1–12, IEEE, 2017.
- [19] L. Ying, S. Shakkottai, A. Reddy, and S. Liu, "On combining shortest-path and back-pressure routing over multihop wireless networks," *IEEE/ACM Trans. Netw.*, vol. 19, no. 3, pp. 841–854, 2010.
- [20] L. Ying and S. Shakkottai, "Scheduling in mobile ad hoc networks with topology and channel-state uncertainty," *IEEE Trans. on Automatic Control*, vol. 57, no. 10, pp. 2504–2517, 2012.
- [21] J. Ryu, L. Ying, and S. Shakkottai, "Timescale decoupled routing and rate control in intermittently connected networks," *IEEE/ACM Trans. Netw.*, vol. 20, no. 4, pp. 1138–1151, 2012.
- [22] Z. Zhao, B. Radojicic, G. Verma, A. Swami, and S. Segarra, "Delay-aware backpressure routing using graph neural networks," in *IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*, pp. 4720–4724, 2023.
- [23] C. Joo and N. B. Shroff, "Local greedy approximation for scheduling in multihop wireless networks," *IEEE Trans. on Mobile Computing*, vol. 11, no. 3, pp. 414–426, 2012.
- [24] Z. Zhao, G. Verma, C. Rao, A. Swami, and S. Segarra, "Distributed scheduling using graph neural networks," in *IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*, pp. 4720–4724, 2021.
- [25] Z. Zhao, G. Verma, C. Rao, A. Swami, and S. Segarra, "Link scheduling using graph neural networks," *IEEE Transactions on Wireless Communications*, vol. 22, no. 6, pp. 3997–4012, 2023.
- [26] C. Joo, G. Sharma, N. B. Shroff, and R. R. Mazumdar, "On the complexity of scheduling in wireless networks," *EURASIP J. on Wireless Commun. and Netw.*, vol. 2010, no. 1, p. 418934, 2010.
- [27] R. Bellman, "On a routing problem," *Quarterly of Applied Mathematics*, vol. 16, no. 1, pp. 87–90, 1958.
- [28] L. R. Ford Jr, "Network flow theory," tech. rep., Rand Corp Santa Monica CA, 1956.
- [29] A. Bernstein and D. Nanongkai, "Distributed exact weighted all-pairs shortest paths in near-linear time," in *ACM SIGACT Symp. Theory of Comp.*, pp. 334–342, 2019.