BACKPRESSURE-BASED ROUTING AND SCHEDULING PROTOCOLS FOR WIRELESS MULTIHOP NETWORKS: A SURVEY

ZHENZHEN JIAO, BAOXIAN ZHANG, CHENG LI, AND HUSSEIN T. MOUFTAH

ABSTRACT

Backpressure-based routing and scheduling has been considered as a promising strategy for efficient resource allocation in wireless multihop networks. It has many attractive features such as throughput optimality, achievable adaptive resource allocation, support for agile load-aware routing, and simplicity. In this article, we first categorize existing backpressure-based protocols based on different design criteria and then summarize several key issues faced in the design of efficient backpressure-based protocols. We present a comprehensive survey of state-of-the-art backpressure-based protocols for wireless multihop networks. We illustrate how each of the protocols works, and discuss their merits and deficiencies. Finally, we point out some future directions for supporting efficient backpressure-based routing and scheduling.

INTRODUCTION

Backpressure-based routing and scheduling is a promising strategy for supporting efficient packet transmission scheduling in a wireless multihop network [1]. It has many attractive features including throughput optimality, achievable adaptive resource allocation, support to agile load-aware routing, and simplicity. Backpressure-based protocols work to schedule the packet transmissions based on queue length information and can schedule packets to pass through a queuing network effectively. A lot of research work has shown that backpressure-based protocols can provide efficient network resource allocation. However, there are still some problems that prevent it from being widely deployed in the real world, including its centralized control mode, high computational complexity, poor delay performance, and high queuing complexity. Recently, much research work has been carried out to address these issues and provide effective solutions to support efficient backpressure-based scheduling and routing.

In this article, we first categorize existing backpressure protocols based on different design criteria and then summarize three key issues faced in the design of efficient backpressure-based routing and scheduling protocols for wireless multihop networks, including how to effectively reduce the end-to-end (E2E) delay, how to effectively reduce the queuing complexity at nodes, and how to achieve efficient cross-layer design. We discuss how these issues arise and present a comprehensive survey of state-of-the-art backpressure-based protocols, divided based on how they tackle the above issues. Specifically, among the existing protocols, [2–6] mainly focus on efficiently reducing the average E2E packet delay by addressing those problems causing large packet latency (e.g., routing loops and the last-packet problem); [7, 8] focus on effectively reducing the queuing complexity at nodes by using efficient queue structures instead of using per-flow queues as in traditional backpressure-based protocols; and [9,10] mainly focus on achieving efficient cross-layer design by taking into account the impact of other layers on backpressure routing and scheduling. We illustrate how different protocols work, and discuss their advantages and disadvantages. Finally, we conclude this article with some future directions for the design of efficient backpressure-based protocols. With a slight abuse of notations, in this article, we use the terms "algorithm" and "protocol" interchangeably unless otherwise stated.

The rest of this article is organized as follows. In the next section, we first introduce how the classical backpressure algorithm works. We then categorize existing protocols based on different design criteria and summarize several key issues affecting the performance of backpressure-based protocols. Following that, we present a comprehensive survey of state-of-the-art protocols. We illustrate how different protocols work, and discuss their advantages and disadvantages. We then conclude this article and discuss several future directions in this field.

PROTOCOL CLASSIFICATION AND FUNDAMENTAL ISSUES

In this section, we first introduce how the classical backpressure algorithm in [1] works. We then categorize existing protocols based on different design criteria. Finally, we present three key

Zhenzhen Jiao and Baoxian Zhang are with the University of the Chinese Academy of Sciences.

Cheng Li is with Memorial University.

Hussein T. Mouftah is with the University of Ottawa.

 1536-1284/16/$25.00

issues to be addressed when designing efficient backpressure-based protocols, and discuss how they arise and also how to address them.

Classical Backpressure Algorithm

The original backpressure algorithm was proposed in [1] and works as follows. Assume time is slotted. At each time slot, the backpressure algorithm works to activate a set of non-interference links in the network, which leads to the maximum sum of link-weight multiplying their respective link rates to transmit packets. Here, the weight associated with a link is defined as the largest flow-weight on the link, where the weight associated with a flow (flow-weight) is the differential of the flow's queue backlogs between the two endpoints of the link. More specifically, at the beginning of time slot t, for each link (a, b) in the network, its link-weight is assigned as the maximum backlog differential of all the flows passing through the link (i.e., the maximum flow-weight, ties broken arbitrarily):

$$W_{ab}(t) = \max_{f:(a,b)} \left(Q_a^f(t) - Q_b^f(t) \right), \tag{1}$$

where $Q_a^f(t)$ represents the queue backlog of flow f on node a at time t. Thus, packets belonging to flow f will be transmitted over link (a, b) if (a, b) is to be activated under a schedule $\pi(t)$ which is derived from the following optimization problem:

$$\pi(t) = \arg\max_{\pi \in \Gamma} \sum\nolimits_{(a,b)} W_{ab}(t) r_{ab}(t), \tag{2}$$

where Γ represents the set of all feasible schedules according to a given link interference model, and $r_{ab}(t)$ represents the link rate of (a, b).

Based on Eqs. 1 and 2, it can be seen that under backpressure-based scheduling, packet transmissions are driven by the congestion status in the network. Moreover, in [1], it has been proved that the schedule returned by the backpressure algorithm can support any arrival rates that are supportable by any other scheduling policies.

Protocol Classification

In this subsection, we categorize existing backpressure-based protocols based on different design criteria.

Centralized Protocols vs. Distributed Protocols: Existing backpressure-based protocols can be divided into centralized protocols [3, 5] and distributed protocols [2, 4, 6–10] based on where the routing and scheduling decisions are performed. In centralized protocols, routing and scheduling decisions are made at a coordinator or a central server. The coordinator knows the global network state information (e.g., global link state and accurate queue backlog information) in the network and makes scheduling decisions that can achieve global optimal. Centralized protocols can in general achieve high performance in routing and scheduling decision making. However, centralized protocols often have high computational complexity and thus have the scalability issue. Finding the optimal schedule has a complexity of $O(|V|^3)$ under a one-hop interference model and is in general NP-hard under K-hop interference models ($K \geq 2$), where $|V|$ represents the number of nodes in the network. In addition, centralized protocols also suffer from the single point of failure issue. In [11], a greedy method named Greedy Maximal Scheduling (GMS) was proposed to reduce the computational complexity in centralized link schedule calculation. In GMS, at each time slot, a link (m, n) leading to the global maximum link-weight being added to the link activation set, the initial value of which is null; furthermore, remove all those links that interfere with (m, n) and repeat the above greedy selection until no link is left. It has been shown that the performance by GMS can achieve at least 1/2 of the optimal one for many network topologies and interference models, that is, the capacity region supportable by GMS will be at least half of the optimal one, and furthermore, the performance by GMS is much better than the lower bound in most networks. In contrast, in a distributed protocol, a network node can make routing and scheduling decisions by using the network state information it keeps (either local or global). Distributed protocols are usually more scalable and attractive for dynamic wireless multihop networks. However, the design of efficient distributed backpressure protocols also faces some issues. One major issue is the difficulty in keeping the consistency and accuracy of queue backlog information kept at different network nodes. Use of outdated queue backlog information may lead to inefficient scheduling and routing decisions and thus affect the network performance. Another issue is the high communication overhead required for dissemination of such information if global state information is required.

Adaptive Backpressure Routing vs. Fixed Backpressure Routing: According to the routing strategy used, existing protocols can be divided into adaptive backpressure routing protocols [2, 6, 8] and fixed backpressure routing protocols [5, 10]. In adaptive backpressure routing, each packet's next hop is mainly determined by the queue-length-based backpressure scheduling decisions and also the link interference model. In the design of efficient adaptive backpressure routing protocols, how to ensure backpressure's throughput optimality, and reduced E2E delay and protocol overhead is the key challenge. To address these issues, [2, 8] choose to select more efficient routes in backpressure-based forwarding, which is aimed at achieving desirable delay performance and suppressing unnecessary waste of network resources caused by unnecessary transmissions. Similarly, Backpressure with Adaptive Redundancy (BWAR) [6] uses replication-based routing for delay reduction in disruption-tolerant networks (DTNs). In the category of fixed backpressure routing, the route for each flow is predetermined before packet delivery, and backpressure-based transmission scheduling is used to make decisions on the packet forwarding on different links on such routes. In this category, delay-based backpressure (D-BP) [5] uses packet delay as a metric for transmission scheduling instead of using queue backlog. The use of fixed routing, although simple, can lead to a certain loss in network capacity. Besides the above two

Use of outdated queue backlog information may lead to inefficient scheduling and routing decisions, and thus affect the network performance. Another issue is the high communication overhead required for dissemination of such information if global state information is required.

There are several key issues that largely affect the performance of backpressure based routing and scheduling. Among them, how to achieve desirable E2E delay performance, how to reduce the queuing complexity at nodes, and how to achieve efficient cross-layer design are three such issues that have received the most attention recently.

types of protocols, there has also been some other work trying to obtain a good trade-off between adaptive routing and fixed routing. For example, the shortest-path-aided backpressure algorithm (SBA) [3] and enhanced dynamic backpressure routing (EDR) [4, Ch. 4] try to make a good choice between shortest path routing and adaptive routing when making each scheduling decision based on the current network load.

Key Issues and Solutions

There are several key issues that largely affect the performance of backpressure-based routing and scheduling. Among them, how to achieve desirable E2E delay performance, how to reduce the queuing complexity at nodes, and how to achieve efficient cross-layer design are three such issues that have received the most attention recently. Next, we discuss how these three issues arise and also how existing solutions have addressed them, respectively.

How to Achieve Desirable Delay Performance: Backpressure-based protocols often suffer from poor delay performance. The main reason for this phenomenon is that in backpressure-based networks, smooth packet forwarding relies on consistent queue backlog gradient in the network (per-flow-based or per-destination-based). However, such a gradient may not exist in a network in some cases. More specifically, there are two major causes of large E2E delay in backpressure-based protocols.

One cause is due to the use of queue-length-based adaptive routing at the network layer. In this routing paradigm, packets are adaptively routed in the network according to congestion information only. In this case, packets may take unnecessary long paths and/or even loops, which results in large E2E packet delivery delay. Much work has been carried out to shorten data path lengths and thus improve the E2E delay performance of backpressure-based routing and scheduling. Among the existing work, backpressure-based collection protocol (BCP) [2] aims at avoiding routing loops by using estimated link rate and link usage penalty; SBA [3] and EDR [4] work to encourage packets to travel along shortest paths, when possible, by considering the current network load situation.

Another reason for large E2E delay is due to the so-called last-packet problem. To illustrate how the last-packet problem occurs, consider a queue that stores the last packet of a flow; no subsequent packet belonging to the flow will arrive. This packet may not be served for a long time since the backpressure algorithm gives higher priority to other flows with larger queue length differentials. Moreover, if it does get a chance to be served, it may take a random walk in the network due to the absence of consistent backpressure to push it to move toward its destination. This situation is quite common in networks with dynamic sporadic traffic, and it may cause large E2E delay. In the literature, D-BP [5] and BWAR [6] aim at addressing the last packet problem. Specifically, D-BP uses the delay experienced by each packet as the key metric for scheduling decision making, and BWAR enables packets to replicate themselves when network load is light, hence serving as backpressure to push packets to be forwarded.

How to Reduce Queuing Complexity at Nodes: High queuing complexity is another problem that obstructs a backpressure algorithm's wide deployment. Many backpressure algorithms require each node to maintain one queue for each flow traversing it. This requirement could cause prohibitively high storage overhead at nodes in large networks and may lead to "state-space explosion" in number of queues as the network size increases. Furthermore, a large number of such per-flow queues at nodes may take away the statistical multiplexing advantage [8]: Since only one queue can be scheduled to be activated at a time slot, if the scheduled queue has few packets and cannot fill up the scheduled link, a waste of transmission capacity can be seen, which also contributes to the increase of E2E delay. In the literature, there have been some queue-complexity-reduction-based protocols. Among them, [7] uses two-level cluster-based queue structures for delay-tolerant networks, and packet-by-packet adaptive routing and scheduling (PARN) [8] uses per-neighbor queue-based backpressure routing and scheduling, both of which can largely reduce the number of queues required to be maintained at nodes.

How to Achieve Efficient Cross-Layer Design: Cross-layer design is an important strategy for supporting efficient backpressure-based scheduling and routing in wireless networks. Existing protocols in this aspect have mainly focused on how to handle the interaction between backpressure-based scheduling and TCP, and the interaction between backpressure-based scheduling and inter-flow network coding. In [9], Seferoglu *et al.* first showed that TCP cannot work well with the classical backpressure protocol. To explain this, recall that TCP determines whether and how many packets of a flow should be injected into the network layer via tuning its congestion window size, the adjustment of which relies on the receipt of end-to-end acknowledgment (ACK) packets. Thus, when TCP is used, once a flow with a larger queue backlog difference than others is scheduled to transmit at a time slot, this flow's congestion window at the TCP layer will increase when more ACKs of this flow are transmitted back. As a result, TCP will transmit more packets of this flow to the network layer. In contrast, other flows will have fewer and fewer chances to be activated by backpressure routing in the subsequent time slots. This leads to the unfairness issue among flows. To address this incompatibility issue, [9] proposes a new K-based backpressure algorithm, in which K is an important factor to control the link weight calculation and therefore give chances for flow(s) with low queue backlog(s) to transmit. In inter-flow network-coding-enabled networks, one big issue is how to efficiently incorporate the network coding gain into the link weight calculation for supporting efficient backpressure scheduling and thus improved network performance. Reference [10] proposed such a network-coding-based backpressure protocol that jointly considers adaptive inter-flow network coding and backpressure-based scheduling.

Survey of State-of-the-Art Backpressure-Based Protocols

In this section, we present a comprehensive survey of state-of-the-art backpressure-based routing and scheduling protocols for wireless multihop networks. Based on how the key issues in the last section were addressed, we categorize the existing work into three categories: delay-based protocols, queue-complexity-based protocols, and cross-layer-design-based protocols, which aim to reduce the E2E delay, queuing complexity at nodes, and efficient cross-layer design, respectively. Next, we shall illustrate how different protocols work, and discuss their advantages and disadvantages.

Delay-Based Protocols

Some existing work [2–6] focuses on reducing the E2E delay of backpressure-based routing and scheduling. Among these existing protocols, the former three protocols (i.e., [2–4]) aim at selecting more efficient routes for delay reduction, and [5, 6] focus on resolving the last packet problem.

Backpressure-Based Collection Protocol: Reference [2] is a distributed, highly agile backpressure-based data collection protocol for a wireless sensor network (WSN). In Backpressure-Based Collection Protocol (BCP), routing and forwarding decisions are made on a per-packet basis. To maximally avoid the occurrence of routing loops, in BCP, when a node needs to select a link for next-hop forwarding, it takes into account not only the queue length differential on this link, but also the estimated link rate and link usage penalty. Here, the usage penalty of a link is deduced via certain utility and penalty functions, in which any routing loop may result in an increase of the penalty and thus decrease the possibility of that link being chosen. To further decrease the average E2E delay of packets, BCP uses a last-in first-out (LIFO) queue structure instead of a first-in first-out (FIFO) queue. As a result, newly arrived packets can be sent to their destinations more rapidly and thus reduce the average E2E delay. However, under the LIFO queue structure, some early arriving packets may never be served and thus be trapped at some intermediate nodes. To address this issue, a mechanism called a floating queue was introduced, which discards those packets that are not served for a long time and adds them to a virtual queue which is actually just a counter. Furthermore, the length of the virtual queue and also the length of the real queue will be added together when calculating the backpressure-based weight for each link. BCP is the first backpressure-based routing protocol designed for WSNs, and it exhibits good packet delivery ratio performance. However, BCP's LIFO queue structure and its packet dropping discipline may not be suitable for some applications with high QoS requirements.

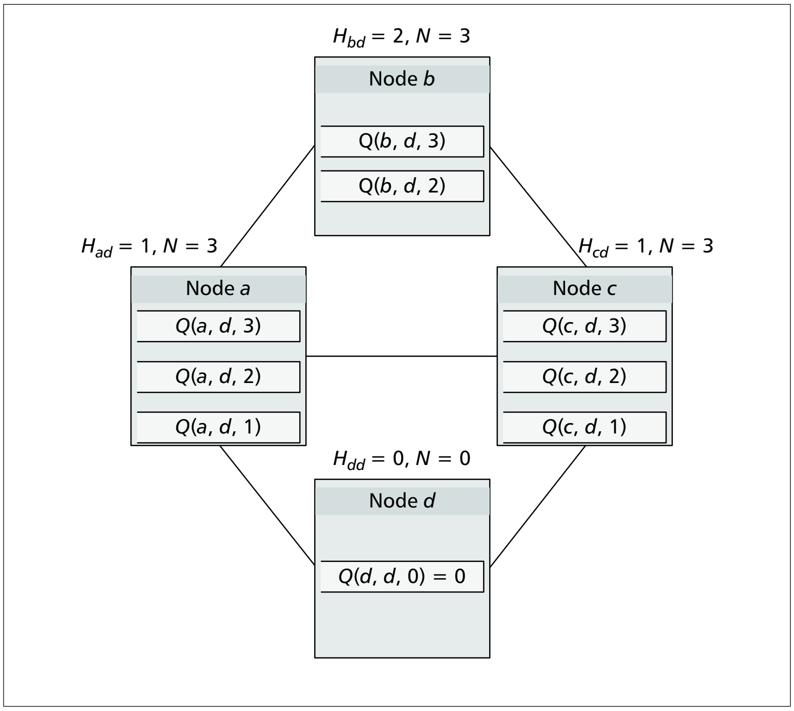

Figure 1. An example illustrating the per-hop queues in SBA.

Shortest-Path-Aided Backpressure Algorithm: Reference [3] adaptively selects routing paths according to the current network load. Specifically, when the network load is light, the shortest-path-aided backpressure algorithm (SBA) requires each packet to be transmitted along its shortest path toward the destination. As network load increases, more paths are exploited to support such traffic. SBA uses a newly introduced queue structure called per-hop queues. Specifically, let H_{ab} represent the minimum hop distance from node a to node b. Let N represent the hop distance of the longest possible simple path in the network. In [3], each node needs to maintain $N - H_{ab} + 1$ per-hop queues for the destination of each packet that it locally stores. Let $Q(a, b, h)$ represent the queue for packets required to be delivered from a to b within h hops (here, h is called hop-constraint). Accordingly, the per-hop queues maintained at node a for destination b are as follows: $Q(a, b, H_{ab})$, $Q(a, b, H_{ab}+1)$, …, $Q(a, b, N)$. Figure 1 shows an example wherein a simple network consisting of four nodes is considered. According to SBA, packets from a queue with hop-constraint h can only be transmitted into queues with a hop-constraint smaller than h. For example, in Fig. 1, packets transmitted from $Q(a, d, 3)$ to $Q(b, d, 2)$, $Q(c, d, 2)$, $Q(c, d, 1)$, or $Q(d, d, 0)$ is allowable, but those transmitted from $Q(a, d, 3)$ to $Q(b, d, 3)$ or $Q(c, d, 3)$ are prohibited. Under such a per-hop queue structure, new backpressure weight is calculated based on the queue length differential between the per-hop queues kept at neighboring nodes. Furthermore, to minimize the average E2E path lengths taken by packets, in SBA, the hop-constraint for each flow is adaptively determined according to a network utility optimization model. Not only can SBA guarantee the classical backpressure algorithm's throughput optimality, it can also achieve much lower E2E delay than the classical algorithm via the assistance of shortest-path routing. However, SBA largely increases the number of queues kept at each network node, which can affect its scalability.

By using such an adaptive redundancy technique, BWAR achieves significant E2E delay performance improvement. However, BWAR needs each node to maintain many duplicate queues, which largely increases the number of queues kept at nodes.

Enhanced Dynamic Backpressure Routing: Reference [4] is another shortest-path assisted backpressure algorithm. Consider the fact that only allowing packets to be transmitted along restricted routes may reduce the network capacity; enhanced dynamic backpressure routing (EDR) does not prohibit the use of any potential route in a network but imposes two biases (i.e., shortest path bias and, optionally, flow priority bias) on the backpressure-based link-weight calculation. Specifically, in EDR, link-weight is calculated as follows instead of as done in Eq. 1:

$$W_{ab}(t) = \max_{f:(a,b)} \left(\theta_a^f \left(Q_a^f(t) + V_a^f \right) - \theta_b^f \left(Q_b^f(t) + V_b^f \right) \right). \quad (3)$$

In Eq. 3, $Q_a^f(t)$ and $Q_b^f(t)$ represent the queue lengths of flow f at nodes a and b (as in Eq. 1), respectively, and V_a^f and V_b^f represent the distances (e.g., hop counts) from nodes a and b to the destination of flow f, respectively, which is the shortest path bias. With the shortest path bias, packets are inclined to move along the shortest paths and thus reduce the packet delivery delay. In Eq. 3, θ_b^f and θ_b^f represent the flow priority weight of flow f at nodes a and b, respectively. A higher θ_x^f value means a higher priority of flow f at node x. However, determining appropriate values for these parameters in time varying wireless networks is nontrivial in practice.

Delay Based Backpressure Algorithm: Reference [5] aims at addressing the last packet problem in backpressure-based scheduling. In [5], each flow is assumed to take a fixed, single, and loop-free route. Let $W_{f,k}(t)$ denote the sojourn time of the head-of-line packet of queue $Q_{f,k}(t)$, where $Q_{f,k}(t)$ represents the queue at the node that is the kth hop away from the source on the route of flow f. A new delay metric, $V_{f,k}(t)$, is designed with a value equal to the differential between $W_{f,k}(t)$ and $W_{f,k-1}(t)$, that is, $V_{f,k}(t) = W_{f,k}(t) - W_{f,k-1}(t)$. Then the delay-based backpressure algorithm (D-BP) uses a new delay-metric-based backpressure link-weight calculation method instead of that in Eq. 1, that is, the link-weight of link (k, $k+1$) equals the differential of the delay metrics $V_{f,k}(t)$ and $V_{f,k+1}(t)$. This new delay-based link weight is actually in some sense "equivalent" to the queue-length-based link weight in the classical backpressure-based algorithm: Recall that a larger $V_{f,k}(t)$ implies a larger queue length of $Q_{f,k}(t)$, and a larger differential between $V_{f,k}(t)$ and $V_{f,k+1}(t)$ implies a larger queue length differential. However, D-BP may suffer from the scalability issue due to its centralized nature and high computational complexity.

Backpressure with Adaptive Redundancy: Reference [6] is a backpressure routing protocol for DTNs. To address the last packet problem under light traffic in such networks, backpressure with adaptive redundancy (BWAR) introduces a replication-based forwarding mechanism. Replication-based forwarding is a commonly used routing strategy in DTNs, and it allows a data packet to have multiple replicas transmitted in a network simultaneously for improving the packet delivery ratio performance. Consider the fact that excessive replicas may cause congestion; in BWAR, an adaptive redundancy technique for backpressure routing is designed, in which replicas are only generated to reduce the E2E delay under low load conditions, while traditional backpressure routing is used under high traffic load conditions. More specifically, BWAR works as follows. Each node maintains a set of duplicate queues to buffer replicas of data packets and uses an original queue to buffer original data packets. Backpressure-based scheduling decisions are then made based on the sum of the actual lengths of original queues and weighted lengths of duplicate queues. When original packets have been sent out and removed from the original queue, if the queue size is lower than a certain threshold, those already sent-out-but-not-acknowledged-yet packets are duplicated and then stored in the local duplicate queue. Such duplicate replicas will not be removed unless they are reported to have been received by their destinations or timed out. When a link is scheduled to transmit, replicas can only be transmitted when the sending node has no original packets in its buffer. By using such an adaptive redundancy technique, BWAR achieves significant E2E delay performance improvement. However, BWAR needs each node to maintain many duplicate queues, which greatly increases the number of queues kept at nodes.

Queuing-Complexity-Based Protocols

Some existing backpressure-based algorithms were designed to reduce the queuing complexity at nodes. Typical algorithms in this aspect have used two-level cluster-based queues [7] and per-neighbor queues [8]. Next, we introduce how the algorithms using these queues work.

Two-Level Backpressure with Source-Routing (BP+SR): Reference [7] was proposed for DTNs, which consist of geographically separated clusters of nodes and are intermittently connected via mobile relay nodes (i.e., mobile carriers). In such networks, if traditional backpressure algorithm was directly used, the low moving speed of mobile carriers can cause very large queue backlogs at nodes on the path between sources and destinations, and thus large E2E delay. To address this issue, in [7], the authors proposed an algorithm called two-level cluster-based backpressure with source routing (BP+SR). In the BP+SR algorithm, inter-cluster communications can only be realized via gateway nodes of clusters and mobile carriers. The gateway nodes form an *overlay* network of the DTN, which are connected via mobile carriers. For intra-cluster communications, traditional backpressure scheduling is used. For inter-cluster communications, backpressure routing and scheduling is also used but on top of the *overlay* network. This is so-called two-level backpressure. Accordingly, two types of queues (i.e., intra-cluster and inter-cluster) must be kept at nodes to support two-level backpressure routing. In BP+SR, if the target destination is in a remote cluster, the source node first uses source routing to choose a source-cluster-gateway and a destination-cluster-gateway for the flow, and then uses two-level backpressure to schedule the packet transmissions. BP+SR can greatly reduce the number of queues required to be maintained at nodes and also the queue lengths at gateway nodes. One big issue in the implementation of the BP+SR algorithm is how to keep the consistency and also accuracy of the queue backlog information kept at gateway nodes of different clusters in a timely fashion, which are intermittently connected via mobile carriers.

Packet-by-Packet Adaptive Routing and Scheduling Algorithm: Reference [8] is a backpressure-based adaptive routing algorithm using per-neighbor queue. Specifically, the packet-by-packet adaptive routing and scheduling algorithm (PARN) introduces an M-backpressure mechanism to perform transmission scheduling such that a link is eligible to be activated only when the queue backlog differential between the two ends of the link is larger than M ($M \geq 1$). The introduction of M-backpressure in PARN can force packets to go through shorter routes. However, its introduction may also cause large delay in networks with moderate or light load wherein nodal queues may need a long time to backlog more than M packets. To address this issue and also reduce the queuing complexity, PARN chooses to use per-neighbor queues. In this way, PARN enables each node to only keep one queue for each neighbor instead of one for each traversing flow, which largely reduces the queuing complexity. To make per-neighbor queues possible, PARN uses a probabilistic forwarding mechanism, which determines each packet's next hop upon its arrival at the current node. This is quite different from most existing backpressure algorithms, in which each packet's next hop is determined upon its departure and based on how its associated backpressure-based transmission scheduling decision is made. Furthermore, PARN introduces the shadow queue, which is per-flow based, and each shadow queue is actually just a counter that denotes the number of shadow packets belonging to a specific flow at a node. The expected number of shadow packets equals the number of actual packets at the current node multiplying $1 + \varepsilon(\varepsilon > 0)$, where ε is a system parameter. PARN performs M-backpressure-based scheduling on shadow queues instead of real queues. Moreover, in [8], PARN was also slightly modified to co-work with inter-flow network coding. However, its shadow-queue-based scheduling decision making process may overestimate the coding gain brought by a coding chance and thus affect the achievable network throughput. Furthermore, the appearance of coding chances may be affected due to its probabilistic routing strategy, which can largely affect its ability in fully utilizing the network coding's capability to improve the performance of the network.

Cross-Layer-Based Protocols

Some existing backpressure-based algorithms were designed with efficient cross-layer design. Typical existing algorithms in this aspect have considered the interaction between backpressure-based scheduling and TCP [9], and the interaction between backpressure-based scheduling and inter-flow network coding [10]. Next, we introduce how these two algorithms work.

TCP-Aware Backpressure Routing and Scheduling: Reference [9] was proposed to address the incompatibility between TCP and backpressure-based scheduling. For this purpose, it introduces a new backpressure-based flow-weight calculation method. Recall that in the classical backpressure algorithm, a flow f's flow-weight on a link (a, b) equals $Q_a^f(t) - Q_b^f(t)$, where $Q_a^f(t)$ and $Q_b^f(t)$ are the queue length of flow f at nodes a and b, respectively. In contrast, in [9], flow-weight is calculated as $max\{K, Q_a^f(t)\} - Q_b^f(t)$. In practice, the value of K is crucial for enabling TCP and backpressure to be very compatible. On one hand, if K is selected too small, more packets will be trapped in some nodes' buffers, that is, the number of packets that cannot get transmission opportunity increases, which reduces TCP throughput. On the other hand, if K is too large, the TCP-aware backpressure may not exploit the throughput improvement benefit of backpressure routing and scheduling. In [9], K is suggested to be set to $B_a/|F_a|$, where B_a represents the buffer size of node a and $|F_a|$ denotes the number of flows traversing a. This method of flow-weight calculation can encourage flows with short queues to transmit first. However, when TCP-aware backpressure routing works in networks with light traffic, if the backlog of each flow's queue is less than K, the scheduling decisions might seem to be made randomly and disobey the backpressure-based scheduling policy.

Network-Coding-Based Backpressure Algorithm: Reference [10] considers the integration of backpressure-based scheduling and inter-flow network coding. The design objective of the network-coding-based backpressure algorithm (NBP) is to maximally exploit the network coding gain for enhancing the achievable network transmission capacity under backpressure scheduling and thus improve the network performance. Here, we use an example to explain the main idea behind NBP. Consider the network topology in Fig. 2a. In Fig. 2a, there are three flows, f_1, f_2, and f_3, among which both f_1 and f_2 go from a to c while f_3 goes from c to a. At the beginning of a time slot t, the queue backlogs of the three nodes for different flows are shown in Fig. 2a. Under classical backpressure scheduling, we can see that the maximum achievable link-weight in b's one-hop scope is $W_{bc}(t) = 3$, that is, three packets belonging to flow f_2 can be scheduled to transmit on link (b, c). This means that in Fig. 2a, at most three packets can be transmitted in b's one-hop scope under classical backpressure scheduling. However, by using inter-flow network coding, a larger one-hop-scope "contribution" by b might be possible if there are network coding opportunities at node b, which can be used for b to forward packets to a and c simultaneously via a single transmission (by XOR decodable packets together as shown in Fig. 2b), that is, enable links (b, a) and (b, c) to be activated simultaneously. NBP encourages nodes to detect and make the best use of network coding opportunities under backpressure-based scheduling and thus utilize the gain brought by network coding for network performance enhancement. However, the introduction of network coding also causes increased computational complexity and queuing complex-

One big issue in the implementation of the BP+SR algorithm is how to keep the consistency and also accuracy of the queue backlog information kept at gateway nodes of different clusters in a timely fashion, which are intermittently connected via mobile carriers.

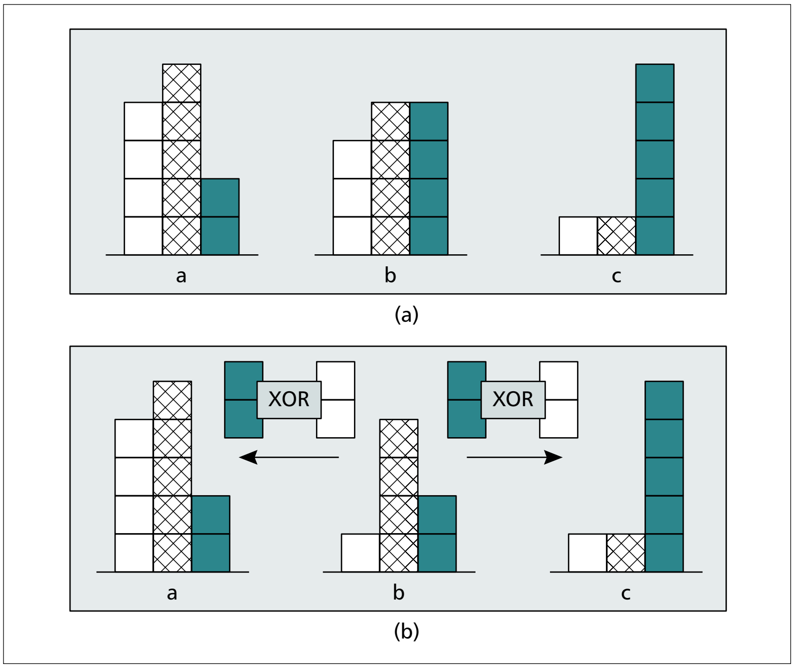

Figure 2. An example illustrating the idea behind NBP. In this figure, white, lattice-filled, and green grids represent packets of flows f_1, f_2, and f_3, respectively.

ity at nodes, where extra queues are needed for buffering already sent packets for decoding.

In this section, we have reviewed nine typical backpressure-based routing and scheduling protocols for wireless multihop networks. Table 1 summarizes and compares the properties of these protocols.

Conclusion and Future Directions

Backpressure-based routing and scheduling is a promising strategy for efficient resource allocation in wireless multihop networks. In particular, its robustness to time-varying network conditions makes it suitable for wireless multihop networks with mobile elements. Furthermore, since backpressure-based routing makes packet transmission decisions mainly based on congestion status, it is thus also suitable for wireless multihop networks with burst traffic. In this article, we present a comprehensive survey on existing backpressure-based routing and scheduling protocols for wireless multihop networks. Existing work has made significant progress in enabling efficient backpressure-based routing and scheduling for such networks. However, there are still some open issues in this research area, which are far from being well studied. Next, we list two such issues.

Localized Backpressure Scheduling

Most existing backpressure algorithms (either centralized or distributed) have been focused on pursuing throughput optimality using global state information, which may cause a scalability issue in practice. Backpressure scheduling based on local state information (including local network state and local traffic state) is attractive in particular for large-scale resource-limited wireless multihop networks. In such scenarios, each node needs to make a localized backpressure-based scheduling decision on whether it should transmit and also to which neighbor based on its local state information and also its ranking in the neighborhood (e.g., two- or k-hop neighborhood). In [12], a lightweight anycast-based localized implementation of a backpressure scheduling algorithm was proposed. In [12], backpressure scheduling is implemented on the IEEE 802.11 medium access control protocol in an anycast manner for wireless sensor networks. That is, each node sends a request-to-send packet that carries its current queue length information before transmitting a data packet; then its neighbors closer to the target sink node compete to reply with a clear-to-send packet if their queue lengths are shorter than that of the sender and the neighbor, leading to the largest queue length differential having the highest priority to respond. Some interesting topics for enabling efficient localized backpressure scheduling in wireless multihop networks include:

- How to promptly exchange local state information among neighbor nodes with low overhead, with freshness and accuracy that can largely affect the scheduling performance
- How to effectively coordinate neighbor nodes so as to make high ranking nodes transmit first in a localized manner
- How to design efficient strategies for joint routing selection and localized backpressure scheduling while achieving high network performance

Whether it is possible to incorporate residual energy information of nodes into the localized backpressure scheduling decision making while maintaining high network performance (e.g., in terms of network throughput and lifetime) is also an interesting topic to explore.

Predictive Backpressure Scheduling

In [13], a predictive scheduling technique is introduced for guiding predictive backpressure scheduling decision making. In [13], a predictive backpressure algorithm (PBP) was proposed. In PBP, virtual prediction queues are established at each node, which record the number of packets that are expected to arrive in subsequent time slots (a node can know this via a predictive look-ahead window model). PBP then makes a backpressure-based scheduling decision based on the sum of the lengths of these prediction queues and the real queue. In this way, PBP largely reduces the E2E packet delay. For predictive backpressure scheduling, the accuracy of queue length prediction at network nodes greatly depends on how the queues are managed. Therefore, how to efficiently incorporate predictive backpressure scheduling with proper queue structures (e.g., per-neighbor queue, per-destination queue, shadow queue) as well as appropriate link weight calculation for achieving further improved network performance still needs in-depth study.

In summary, backpressure-based routing and scheduling is a promising strategy for efficient network resource allocation. However, there is still much work to be done on enabling it to achieve high network performance with low protocol overhead in wireless multihop networks.

	Properties				
Protocols	Centralized/ distributed	Queue structures	Routing strategies	Support network coding	Support for mobility
BCP [2]	Distributed	Per flow and LIFO	Adaptive routing	N	Y
SBA [3]	Centralized	Per-hop queues for each flow	Adaptive routing (online) + shortest path routing (offline)	N	—
EDR [4]	Distributed	Per flow	Adaptive routing with shortest path bias	N	Y
D-BP [5]	Centralized	Per flow	Fixed routing	N	—
BWAR [6]	Distributed	Per-flow queues + duplicate queues	Adaptive replication routing	N	Y
BP+SR [7]	Distributed	Cluster-based two-level queue	Two-level backpressure routing + source routing	N	Y
PARN [8]	Distributed	Per-neighbor queue + shadow queue	Adaptive probabilistic routing	Y	—-
TBP [9]	Distributed	Per flow	Adaptive routing	N	—
NBP [10]	Distributed	Per flow	Adaptive routing	Y	—

Table 1. Comparison of various backpressure-based protocols for wireless multihop networks.

Backpressure based routing and scheduling is a promising strategy for efficient network resource allocation. However, there still exists much work to be done for enabling it to achieve high network performance with low protocol overhead in wireless multihop networks.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 61501125, 61531006 and 61471339, and the funds from the Natural Sciences and Engineering Research Council (NSERC) of Canada (Discovery Grant 293264-07 and Strategic Project Grant STPGP 397491-10). Z. Jiao is also with China Academy of Information and Communications Technology, Beijing, China. B. Zhang is the corresponding author.

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. Auto. Control*, vol. 37, no. 12, Dec. 1992, pp. 1936–48.

[2] S. Moeller *et al.*, "Routing without Routes: The Backpressure Collection Protocol," *Proc. IEEE/ACM IPSN '10*, Apr. 2010, pp. 279–90.

[3] L. Ying *et al.*, "On Combining Shortest-Path and Back-Pressure Routing over Multihop Wireless Networks," *IEEE/ACM Trans. Networking*, vol.19, no. 3, June 2011, pp. 841–54.

[4] L. Georgiadis, M. J. Neely, and L. Tassiulas, *Resource Allocation and Cross-Layer Control in Wireless Networks*, Now Publishers Inc., 2006.

[5] B. Ji, C. Joo, and N. B. Shroff, "Delay-Based Back-Pressure Scheduling in Multi-Hop Wireless Networks," *Proc. IEEE INFOCOM '11*, Apr. 2011, pp. 2579–87.

[6] M. Alresaini *et al.*, "Backpressure with Adaptive Redundancy (BWAR)," *Proc. IEEE INFOCOM '12*, Mar. 2012, pp. 2300–08.

[7] J. Ryu, L. Ying, and S. Shakkottai, "Back-Pressure Routing for Intermittently Connected Networks," *Proc. IEEE INFOCOM '10*, Mar. 2010, pp. 1–5.

[8] E. Athanasopoulou *et al.*, "Back-Pressure-Based Packet-by-Packet Adaptive Routing in Communication Networks," *IEEE/ACM Trans. Networking*, vol. 21, no. 1, Feb. 2013, pp. 244–57.

[9] H. Seferoglu and E. Modiano, "TCP-Aware Backpressure Routing and Scheduling," *Proc. IEEE ITA '14*, Feb. 2014, pp. 1–9.

[10] Z. Jiao et al., "NBP: An Efficient Network-Coding Based Backpressure Algorithm," *Proc. IEEE ICC '13*, June 2013, pp. 1625–29.

[11] X. Lin and N. B. Shroff, "The Impact of Imperfect Scheduling on Cross-Layer Congestion Control in Wireless Networks," *IEEE/ACM Trans. Networking*, vol. 14, no. 2, Apr. 2006, pp. 302–15.

[12] Z. Jiao *et al.*, "A Distributed Gradient-Assisted Anycast-Based Back-Pressure Framework for Wireless Sensor Networks," *Proc. IEEE ICC '14*, June 2014, pp. 2815–20.

[13] L. Huang *et al.*, "When Backpressure Meets Predictive Scheduling," *Proc. ACM MOBIHOC '14*, Aug. 2014, pp. 33–42.

Biographies

Zhenzhen Jiao received his Ph.D. degree from the Research Center of Ubiquitous Sensor Networks, University of Chinese Academy of Sciences (UCAS), Beijing, China, in 2015. He is currently a research engineer at the China Academy of Information and Communications Technology, Beijing, China. He has served on Technical Program Committees for several international conferences and symposia. He has published over 10 papers in archival journals and conference proceedings. His research interests include wireless multihop network protocol and algorithm design, and the Internet of Things.

Baoxian Zhang [SM'12]received his Ph.D. degree in electrical engineering from Northern Jiaotong University, China, in 2000. He is currently a full professor with the Research Center of Ubiquitous Sensor Networks at UCAS. Prior joining UCAS, he was a research scientist with the School of Information Technology and Engineering, University of Ottawa, Canada, from 2002 to 2005. From 2001 to 2002, he was a postdoctoral fellow with the Department of Electrical and Computer Engineering, Queen's University, Kingston, Canada. He is currently an Associate Editor of the *IEEE Systems Journal* and has served as a Guest Editor of several Special Issues of publications including the *IEEE Journal on Selected Areas in Communications* and *Elsevier Ad Hoc Networks Journal*. He has served on Technical Program Committees for many international conferences and symposia. He has published over 150 refereed technical papers in archival journals and conference proceedings. His research interests cover network protocol and algorithm design, wireless ad hoc and sensor networks, the Internet of Things, and IP networks.

CHENG LI [SM] received his B. Eng. and M. Eng. degrees from Harbin Institute of Technology, P. R. China, in 1992 and 1995, respectively, and his Ph.D. degree in electrical and computer engineering from Memorial University, St. John's, Canada, in 2004. He is currently a full professor at the Faculty of Engineering and Applied Science of Memorial University. His research interests include mobile ad hoc and wireless sensor networks, wireless communications and mobile computing, switching and routing, and broadband communication networks. He is an editorial board member of *Wiley Wireless Communications and Mobile Computing* and the *Journal of Networks*, and an Associate Editor of *Wiley Security and Communication Networks*. He has served as a Co-Chair for various technical symposia of many international conferences, including IEEE GLOBECOM and ICC. He has served as a TPC member for many international conferences, including IEEE ICC, GLOBECOM, and WCNC. He is a registered Professional Engineer (P. Eng.) in Canada and a member of the IEEE Communication, Computer, Vehicular Technology, and Ocean Engineering Societies.

HUSSEIN T. MOUFTAH [F'90] joined the School of Electrical Engineering and Computer Science of the University of Ottawa in September 2002 as a Tier 1 Canada Research Chair, where he became a University Distinguished Professor in 2006. He was previously with the Department of Electrical and Computer Engineering at Queen's University (1979–2002), where prior to his departure he was a full professor and the Department Associate Head. He served as Editor-in-Chief of *IEEE Communications Magazine* (1995–1997) and IEEE Communications Society Director of Magazines (1998–1999), Chair of the Awards Committee (2002–2003), member of the Board of Governors (1997–2000 and 2006–), and Director of Education (2006–). He is the author or coauthor of 1000 technical papers, 7 books, and 48 book chapters. He was the recipient of the 1989 Engineering Medal for Research and Development from the Association of Professional Engineers of Ontario, and the 2002 Ontario Distinguished Researcher Award of the Ontario Innovation Trust. He has received 12 Outstanding/Best Paper Awards. He is a Fellow of the Canadian Academy of Engineering (2003) and the Engineering Institute of Canada (2005).