Combining Machine Learning With Back-Pressure-Based Routing

Pauline Laßmann, Christoph Schwarzenberg*, Florian Wiedner*
*Chair of Network Architectures and Services, TUM School of Computation, Information, and Technology
Technical University of Munich, Germany
Email: ga27vit@mytum.de, schwarzenberg@net.in.tum.de, wiedner@net.in.tum.de

***Abstract*—The back-pressure routing algorithm guarantees optimal throughput but has poor delay performance. A variety of approaches have been proposed to solve the delay and also memory consumption problems. One way is to use machine learning. The goal of this paper is to find different back-pressure routing policies that are supported by machine learning. Two methods are presented, one using Q-learning and the other using predictive scheduling.**

***Index Terms*—back-pressure routing, machine leaning, Q-learning, predictive scheduling**

1. Introduction

Nowadays, applications in areas such as sensor networks, wired flow-based networks and traffic systems require a reliable method to distribute heavy traffic loads across the entire system or network. The back-pressure routing (BP) algorithm offers great potential for such a task. The algorithm examines all possible routes to balance traffic loads across an entire queuing network, thus guaranteeing network-wide throughput optimality [1].

When traffic loads are high, this algorithm works, and available network resources can be used in a highly dynamic manner. However, excessive route searching at low and medium traffic loads can lead to unnecessarily long routes or even routing loops. This leads to poor delay performance [2] [3].

Improvement approaches on various fronts have been made over the years one of them being machine learning aided back-pressure routing. Using prediction its implementations see an overall improvement in delay performance while still being able to efficiently forward packets with near-optimal throughput, having low computational complexity, a distributed implementation and not requiring statistical information about the system dynamics [2] [4].

In the next section of this paper, an overview of different BP algorithms is given. The third section briefly introduces the original BP concept and then presents various framework parameters under which it can be realized. The fourth section deals specifically with BP routing policies supported by machine learning.

2. Related Work

The back-pressure routing algorithm was first introduced 1990 by Tassiulas and Ephremides [1] and initially proposed for wireless multi-hop radio networks. One of its main shortcomings is its poor delay performance.

Over the years there have been a variety of different approaches trying to solve this problem. Each one builds its improvements on a different aspect such as:

- Using shadow queues
- Separating intra-cluster routing from inter-cluster routing
- Using the shortest path algorithm
- Using the last-in-first-out algorithm
- Considering local queue length information
- Eliminating loops in the network
- Introducing a delay parameter

Bui et al. [5] and Athanasopoulou et al. [6] improve the original algorithm with the help of shadow queues. Bui et al. [5] propose shadow queuing as a way of improving delay performance of the original back-pressure algorithm. Athanasopoulou et al. [6] combine the original algorithm with probabilistic routing tables and shadow queues. This way routing and scheduling is decoupled in the network.

In [7] Ryu et al. separate intra-cluster routing from inter-cluster routing. This is done using a two-phase routing method by combining back-pressure routing with source routing. This results in only a subset of nodes having large queues, thus improving delay performance.

The improved algorithms introduced in [8], [9] and [10] make use of the shortest path algorithm to archive better delay performance. Neely et al. [8] introduce BPbias. It combines the information of queue and shortest path length to shorten packet routes. The algorithm of Ying et al. [9], when making each scheduling decision based on the current network load, has a choice between shortest path routing and adaptive routing. The route searching process for the algorithm introduced by Yin et al. [10] dynamically switches between shortest path mode and traditional back-pressure routing mode based on a threshold.

The last-in-first-out algorithm (LIFO) is used in the works of [11] and [12] to improve the original back-pressure algorithm. Moeller et al. [11] combine it with LIFO queuing. Huang et al. [12] prove that near-optimal utility-delay trade-off is achievable with the help of LIFO.

Cui et al. [3] proposed a back-pressure routing algorithm considering local queue length information of up to two-hop nodes and another one considering global queue length information of all nodes, called BPmin.

To eliminate loops in the network Rai et al. [13] propose to use directed acyclic graphs, which in turn improves the delay performance.

doi: 10.2313/NET-2023-06-1_10

The works in [14] and [15] introduce a delay parameter for delay improvement. Ji et al. [15] introduce a back-pressure routing algorithm using a new delay metric to reduce packet delay for light traffic loads. Ji et al. [14] use a new queue management policy with a delay parameter that makes the algorithm select favorable routes by considering both delay requirements and network throughput.

Improvement method	Improvement
Using shadow queues	- Improved delay performance - Decoupling routing and scheduling in the network
Separating intra-cluster routing from inter-cluster routing	- Only a subset of nodes have large queues - Improved delay performance
Using the shortest path algorithm	- Shorter packet routes - Improved delay performance - Dynamically switching between shortest path mode and traditional back-pressure routing
Using the last-in-first-out algorithm	- Combining BP routing and LIFO queuing - Near-optimal utility-delay trade-off - Improved delay performance
Considering local/global queue length information	- Improved delay performance
Eliminating loops in the network	- Improved delay performance
Introduction of a delay parameter	- Reduce packet delay for light traffic loads - Electing favorable routes by considering delay requirements and network throughput - Improved delay performance

TABLE 1: Different back-pressure routing improvement methods and their improvements

3. Back-Pressure Routing Framework Parameters

BP uses time slots to operate. To balance the traffic load in the network, it tries to forward data in each time slot in a way that optimizes the differential backlog between neighboring nodes. This is done by considering all potential routes. In each timeslot, nodes can transmit data that they store in different queues for each destination to a neighboring node. The algorithm forwards packets based on congestion gradients, so it checks which of its neighbours queues for that destination is the smallest and routes the data that way. Data transmitted from one node to another is removed from the first node's queue of the destination and added to the second node's queue of the destination [1]. An example can be seen in figure 1.

As described in section 2, several versions of this algorithm exist. These can have various framework parameters under which they can be realized as seen in Figure 2.

First, back-pressure protocols can be divided into centralized protocols [9], [15] and distributed protocols [4], [2]. It differentiates on where routing and scheduling decisions are made. A coordinator or central server is responsible for routing and scheduling decision making in centralized protocols [9], [15]. High performance can be achieved with routing and scheduling decisions, but on occasion scalability issues due to the high computational complexity can be observed. Distributed protocols [4], [2] are generally more scalable. Network

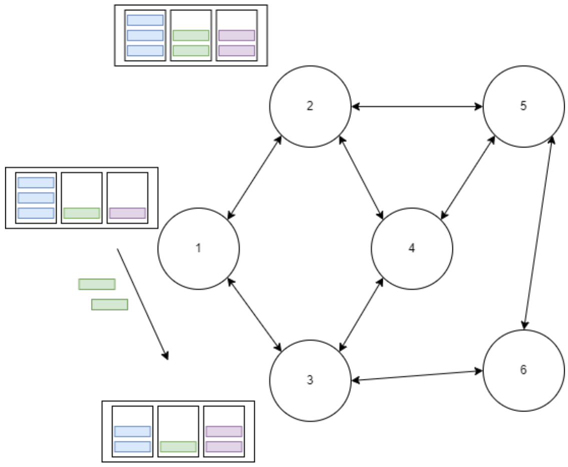


Figure 1: Workings of Back-pressure routing

nodes can use the network state information they maintain to make routing and scheduling decisions. Maintaining the consistency and accuracy of the queue backlog information stored in different network nodes however can be a difficulty. Network performance can be affected by inefficient scheduling and routing decisions when outdated queue backlog information are used [16].

Existing protocols can be classified as adaptive back-pressure routing protocol [11] or fixed back-pressure routing protocol [15]. In adaptive back-pressure routing protocols, the back-pressure scheduling decision based on the queue length primarily determines the next hop of each packet. Fixed back-pressure routing protocols predetermine the route for each flow before the packets are delivered. Back-pressure-based transmission scheduling is used to decide on packet forwarding. However, it has the disadvantage of leading to a minor loss of network capacity [16].

Over the course of time, the original algorithm has been modified again and again in various ways to improve it. Different information of queuing networks such as queue length, path length, clusters and packet delay can be incorporated into the algorithm. Additionally, back-pressure routing in combination with machine learning also gained popularity over the last years [2].

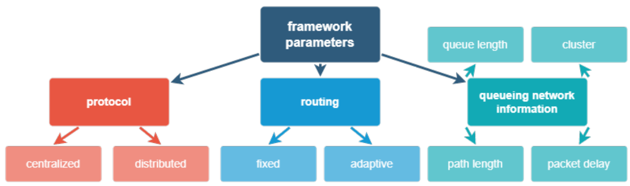


Figure 2: Back-pressure framework parameters overview

4. Back-Pressure Routing in Combination With Data Science

In this section, back-pressure routing algorithms in combination with Data Science are examined in more detail. Q-learning in combination with back-pressure routing is discussed as well as the back-pressure algorithm using predictive scheduling.

doi: 10.2313/NET-2023-06-1_10

4.1. Q-Learning Aided Back-Pressure Routing

Before Q-learning aided back-pressure routing is discussed the concept of Q-learning is conveyed using the Q-routing algorithm. Then the multi-agent Q-learning-based back-pressure routing (QL-BP) algorithm and the adaptive traffic control algorithm are presented.

4.1.1. Q-Routing as a Reinforcement Learning Approach for Packet Routing. Boyan et al. [17] present Q-routing as an algorithm that learns a routing policy which attempts to strike a balance between minimizing the number of hops for packet delivery and the possibility of congestion on popular routes. They refer to their algorithm as a version of the Bellman-Ford shortest path algorithm. For Q-routing to work, a reinforcement learning module is embedded in each node of a network. To keep accurate statistics on which routing decisions result in minimal delivery times, only local communications are used. Furthermore, the Q-routing algorithm is able to route efficiently even when critical aspects of the simulation, such as network utilization, are allowed to vary dynamically.

Based on experiments with different routing policies, the algorithm selects the one to use. Reinforcement learning can be used to update the selected routing policy faster. The performance of a policy is measured by the total time it takes to deliver a packet. To calculate this, Q-learning uses a "learning rate" parameter, as well as an old time estimate and a revised time estimate for packet delivery, to obtain a solution [17].

Q-learning has the disadvantage of being greedy and therefore cannot fine-tune a shortcut discovery strategy. One solution presented in the paper is for the algorithm to select routing directions with a degree of randomness in the initial learning phase. Since this would have an extremely negative impact on congestion, a node uses what is called a "full echo" modification instead of sending actual packets in a random direction. Using this, a node sends information requests to its immediate neighbors each time it needs to make a decision. Each neighbor sends back an estimate of the total time to reach the destination. If shortcuts appear or the policy is inefficient, this information quickly propagates through the network and the strategy is adjusted accordingly. This revised Q-routing is referred to as "full-echo" Q-learning [17].

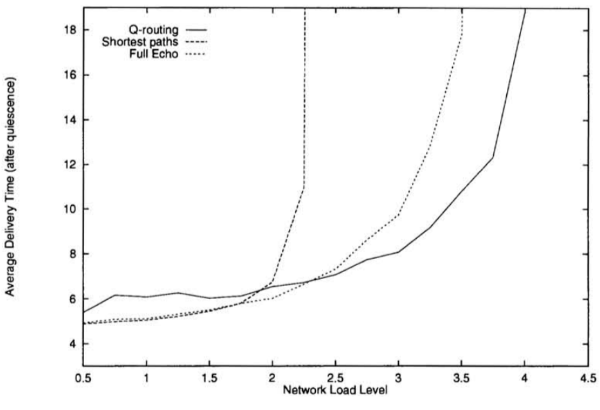


Figure 3: Delivery time for Q-routing, "full echo" Q-routing and shortest path routing [17]

As seen in Figure 3 Q-learning exhibits initial inefficiency when traffic load is low compared to the shortest-path routing strategy, because it first learns the network topology. Once the learning phase is overcome, it performs equivalently to the shortest path. Q-routing with "full echo" is indistinguishable from the shortest path strategy. As the network load increases, the shortest path routing strategy is outperformed by Q-routing with "full echo". Q-routing performs best because it learns an efficient routing strategy and continues to route that way. Q-routing with "full echo", on the other hand, constantly changes its strategy under high load. Not until a further significant increase in traffic load does the Q-routing algorithm also succumbs to overload [17].

4.1.2. Multi-agent Q-learning-based back-pressure routing (QL-BP) algorithm. Gao et al. [2] propose the multi-agent Q-learning-based back-pressure routing (QL-BP) algorithm. They take a general delay reduction framework based on information of the queuing network (bias) and build their QL-BP algorithm on it. The framework goes through three stages:

- Information collection: in this stage useful, local or global, information is collected including queue length, shortest path and packet delay
- Bias extraction: in this stage useful features (such as route congestion estimation) are extracted either in a heuristic manner or with the aid of machine learning based methods like Q-learning
- Back-pressure routing: the extracted bias are programmed into the back-pressure routing algorithm after which the algorithm is capable of adaptively changing packet routes

Each node maintains multiple Q-learning agents that are responsible for generating route congestion estimates from the collected information in the bias extraction phase. Each agent updates the route congestion estimate using the queue length information and the route congestion estimates of the neighboring nodes. Since route congestion is estimated using only local information from neighboring nodes a distributed implementation is possible. Based on the estimated route congestion, each node routes packets to their destinations along the least congested routes [2].

The QL-BP algorithm can be further improved by considering information about the shortest path (QLSP-BP). In this case, the QL-BP algorithm remains the same, except that the shortest path between a source and a destination node is considered in the bias extraction [2].

The QL-BP algorithm is able to maintain a distributed implementation, low computational complexity, and an optimal throughput rate. It reduces the average packet delay by 71% compared to the original BP algorithm at low traffic load. At moderate traffic load, it is 82% higher. The QL-BP algorithm effectively learns the congestion of the routes and adaptively reroutes the packets to better routes. For this, a slight amount of packet delay is accepted in favor of distributed algorithm implementation and low computational complexity. As mentioned earlier, the QL-BP algorithm can be significantly improved by considering shortest path information. The QLSP-BP algorithm outperforms all variants of back-pressure routing algorithms. It reduces the average packet delay by 95%

doi: 10.2313/NET-2023-06-1_10

under light traffic load and 41% under medium traffic load and is the best variant of the improved back-pressure routing algorithms [2]. Figure 4 shows all this graphically.

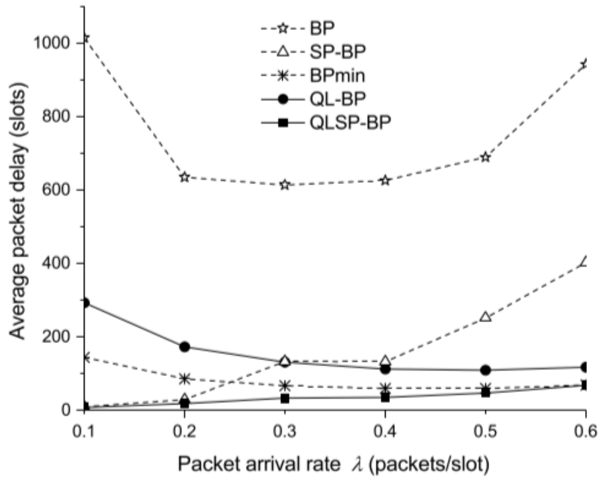


Figure 4: Packet delay for different back-pressure algorithms [2]

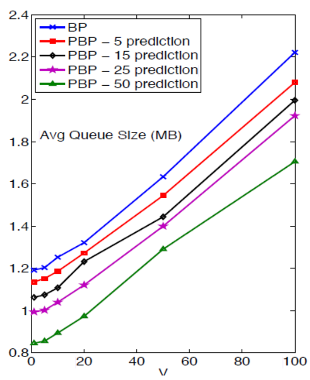
4.1.3. Adaptive traffic control algorithm. Maipradit et al. [18] also use the Q-learning-based back-pressure algorithm. They use it as an adaptive traffic control algorithm. They manage to significantly decrease the average vehicle travel time from 16% to 36% compared to other algorithms. Although this algorithm is applied for traffic control, it should be easily transferable to routing networks. Each intersection has a control agent. This agent collects vehicle speed and vehicle position information in each time window. Congestion information is also exchanged between neighboring agents. Based on the exchanged congestion information, the agent updates its own congestion estimate based on Q-learning. Eventually, all agents receive global congestion information. These are helpful in tasks two and three of the three tasks that each agent performs in every time slot: Learning global congestion information, selecting the optimal traffic phase based on the back-pressure algorithm and vehicle steering, where after a vehicle passes the intersection and enters the next road under the traffic phase selected in task two, the agent determines which lane of this road the vehicle shall use [18].

Their adaptive traffic control algorithm based on back-pressure and Q-learning (ARD-BP-Q) is decentralized and the agent at each intersection executes the algorithm independently. An additional feature is that vehicles with longer travel times pass through an intersection first [18].

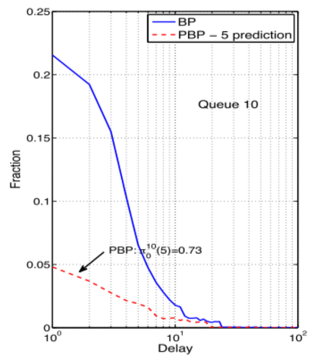
4.2. Predictive Scheduling Aided Back-Pressure Routing

Huang et al. [4], discuss predictive scheduling. Using a look-ahead window model for pre-allocating rates the delay performance of the original back-pressure algorithm is improved. They draw inspiration from pre-fetching techniques used in memory management, branch prediction in computer architecture, and current advances in data mining for learning user behavior patterns. The model is implemented using prediction queues created by the server based on the previous packets.

The authors propose the predictive back-pressure (PBP) algorithm, which performs the BP algorithm based on the prediction queues. PBP achieves a cost performance that is arbitrarily close to optimality. At the same time, it guarantees that the average system delay vanishes as the size of the prediction window increases. Moreover, PBP retains all the desired properties of the original back-pressure algorithm. It remains greedy and does not require statistical information about the system dynamics. In addition, the look-ahead window helps the server use connections more efficiently. The queuing policy chosen for the look-ahead queue leads to different improvements. With first-in-first-out (FIFO) queuing, PBP achieves an average reduction in packet delay that is linear with the size of the prediction window as seen in Figure 5. With last-in-first-out (LIFO) queuing, the average packet delay decreases exponentially with the window size as seen in Figure 5. Thus, the average delay under PBP is strictly better than under the original back-pressure algorithm and totally vanishes as the prediction window size increases [4]. The authors of [4] prove that the algorithm achieves a cost performance arbitrarily close to optimality and that the prediction is more accurate with a larger window size.



Average queue seize of different prediction windows using PBP with first-in-first-out queuing policy (V being a control parameter used to tradeoff utility performance and system delay) [4]



Packet delay distribution using PBP with last-in-first-out queuing policy [4]

Figure 5: PBP performance results

5. Summary and Conclusion

In this paper, the original back-pressure routing algorithms, ones using machine learning and one making use of other improvement methods were presented.

The goal of this work was to find several back-pressure routing policies supported by machine learning, about which Table 2 gives an overview. .

The multi-agent Q-learning aided back-pressure routing algorithm [2] is able to significantly improve delay performance and maintain the following attractive features: distributed implementation, low computational

Variant	Improvement
Multi-agent Q-learning aided back-pressure routing algorithm	- Improved delay performance - Distributed implementation - Low computational complexity - Throughput optimality
Adaptive Traffic Control Algorithm	- Reduce average travel time - Decentralized - Longest travel times passes first
Predictive Back-Pressure algorithm	- Better cost performance - System delay vanishes with increasing prediction window size - No statistical information about system dynamics required - Greedy

TABLE 2: Comparison of machine learning aided back-pressure routing algorithms

complexity, and throughput optimality. A similar system is also used for the adaptive traffic control [18] algorithm, where the traffic delay is also significantly reduced.

The PBP (predictive back-pressure) [4] algorithm, based on a lookahead prediction window model, achieves cost performance arbitrarily close to the optimum. At the same time, it guarantees that the average system delay vanishes as the size of the prediction window increases.

We found that at this stage, only two back-pressure algorithms supported by machine learning could be found. Q-routing and predictive scheduling. Since both work in their specific theoretical models introduced in the respective paper it is however difficult to compare them in effectiveness and suitability to other network models. They are nevertheless both able to significantly reduce delay power and efficiently forward packets with near-optimal throughput, but face other challenges. More computation time is required, and nodes must constantly record data and update the parameters stored there. Since both discussed algorithms based on machine learning have advantages over other state-of-the-art back-pressure algorithms, especially in throughput optimality and cost performance, we believe that both Q-learning and predictive scheduling are attractive optimizations of the original algorithm.

The multi-agent Q-learning based back-pressure routing algorithm has already been improved using the shortest path algorithm [2]. Even though the presented machine learning based back-pressure routing algorithms are already a major improvement over the original, this proves there is still room for further optimization to be found in the future.

References

[1] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," in *29th IEEE Conference on Decision and Control*, 1990, pp. 2130–2132 vol.4.

[2] J. Gao, Y. Shen, M. Ito, and N. Shiratori, "Multi-agent q-learning aided backpressure routing algorithm for delay reduction," 2017. [Online]. Available: https://arxiv.org/abs/1708.06926

[3] Y. Cui, E. M. Yeh, and R. Liu, "Enhancing the delay performance of dynamic backpressure algorithms," *IEEE/ACM Transactions on Networking*, vol. 24, no. 2, pp. 954–967, 2016.

[4] L. Huang, S. Zhang, M. Chen, and X. Liu, "When backpressure meets predictive scheduling," *IEEE/ACM Transactions on Networking*, vol. 24, no. 4, pp. 2237–2250, 2016.

[5] L. Bui, R. Srikant, and A. Stolyar, "Novel architectures and algorithms for delay reduction in back-pressure scheduling and routing," in *IEEE INFOCOM 2009*, 2009, pp. 2936–2940.

[6] 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 Transactions on Networking*, vol. 21, no. 1, pp. 244–257, 2013.

[7] J. Ryu, L. Ying, and S. Shakkottai, "Back-pressure routing for intermittently connected networks," in *2010 Proceedings IEEE INFOCOM*, 2010, pp. 1–5.

[8] M. Neely, E. Modiano, and C. Rohrs, "Dynamic power allocation and routing for time varying wireless networks," in *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428)*, vol. 1, 2003, pp. 745–755 vol.1.

[9] L. Ying, S. Shakkottai, A. Reddy, and S. Liu, "On combining shortest-path and back-pressure routing over multihop wireless networks," *IEEE/ACM Transactions on Networking*, vol. 19, no. 3, pp. 841–854, 2011.

[10] P. Yin, S. Yang, J. Xu, J. Dai, and B. Lin, "Improving backpressure-based adaptive routing via incremental expansion of routing choices," in *2017 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS)*, 2017, pp. 1–12.

[11] S. Moeller, A. Sridharan, B. Krishnamachari, and O. Gnawali, "Routing without routes: The backpressure collection protocol," in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, ser. IPSN '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 279–290. [Online]. Available: https://doi.org/10.1145/1791212.1791246

[12] L. Huang, S. Moeller, M. J. Neely, and B. Krishnamachari, "Lifo-backpressure achieves near-optimal utility-delay tradeoff," *IEEE/ACM Transactions on Networking*, vol. 21, no. 3, pp. 831–844, 2013.

[13] A. Rai, C.-p. Li, G. Paschos, and E. Modiano, "Loop-free backpressure routing using link-reversal algorithms," in *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 87–96. [Online]. Available: https://doi.org/10.1145/2746285.2746304

[14] Z. Ji, Y. Wang, and J. Lu, "Distributed delay-aware resource control and scheduling in multihop wireless networks," in *2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*, 2015, pp. 1–5.

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

[16] Z. Jiao, B. Zhang, C. Li, and H. T. Mouftah, "Backpressure-based routing and scheduling protocols for wireless multihop networks: A survey," *IEEE Wireless Communications*, vol. 23, no. 1, pp. 102–110, 2016.

[17] J. Boyan and M. Littman, "Packet routing in dynamically changing networks: A reinforcement learning approach," in *Advances in Neural Information Processing Systems*, J. Cowan, G. Tesauro, and J. Alspector, Eds., vol. 6. Morgan-Kaufmann, 1993. [Online]. Available: https://proceedings.neurips.cc/paper/1993/file/4ea06fbc83cdd0a06020c35d50e1e89a-Paper.pdf

[18] A. Maipradit, J. Gao, T. Kawakami, and M. Ito, "Adaptive traffic control algorithm based on back-pressure and q-learning," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 1995–1999.