Survey on Back-Pressure Based Routing

Ke Wang, Christoph Schwarzenberg*, Florian Wiedner*,
*Chair of Network Architectures and Services, Department of Informatics
Technical University of Munich, Germany
Email: 19.ke.wang@tum.de, schwarzenberg@net.in.tum.de, wiedner@net.in.tum.de

Abstract—**Back-pressure based routing-algorithms have been studied extensively. They guarantee throughput optimality, but have poor delay performance. In this paper, we focus on wired Ethernet networks and compile a survey of back-pressure based routing-algorithms. Four variants are presented in this paper. We address their advantages, limitations, and possible combinations of different variants.**

Index Terms—**back-pressure routing, shortest-path, delay metric, cluster, machine learning**

1. Introduction

The back-pressure routing algorithm (BP algorithm) first introduced by L. Tassiulas and A. Ephremides in [1] draws many researchers' attention and has been studied extensively since then. Although it was initially proposed for wireless multi-hop radio networks, it can be applied to wire-line networks. The algorithm directs packets in multi-hop queuing networks based on congestion gradients. By exploring all possible routes to balance the loads, it guarantees network-wide throughput optimality [1].

However, there are still plenty of problems that need to be solved. As stated in [2], the BP algorithm requires routers to maintain a separate queue for each destination, which results in high memory complexity. Another problem that hinders the deployment of the algorithm is the poor delay performance. The algorithm works well only with heavy traffic loads. When the traffic loads are light or moderate, BP may lead to packets being directed to unnecessarily long routes or even loops [3].

Since BP guarantees throughput optimality, it has great potential to be applied in the industry. Therefore, many variants have been proposed to solve the delay and memory consumption issues.

2. Related work

In [4], Ying et al. use the shortest path method to avoid the extensive exploration of paths. In [5], Anurag Rai et al. propose to use directed acyclic graphs to eliminate loops in the network, which in turn improves the delay performance.

In [2], [6], [7], queue structure and management are improved to reduce the packet delay. In [6], Alresaini et al. introduce an adaptive redundancy technique that yields the benefits of replication, while at the same time preserving the benefits of traditional BP routing algorithm under high traffic loads. In [7], Ji et al. design a new queue management policy with a delay parameter. The algorithm will then select favorable routes by considering both delay requirements and network throughput. In [8] Bui et al. design a shadow queuing architecture that improves the delay performance for the original BP algorithm. In [9] Athanasopoulou et al. combine BP algorithm with probabilistic routing tables and shadow queues to decouple routing and scheduling in the network. In [10] Moeller et al. combine the BP algorithm with the LIFO queuing discipline. In [11] Huang et al. prove that the algorithm achieves near-optimal utility-delay trade-off. In [12] Gao et al. provide a general framework by combining several parameters addressed above to reduce the delay.

Another approach for improving the algorithm are delay-based BP algorithms [13]–[16]. In [13], Hai et al. propose a novel delay metric called sojourn time backlog and improve the BP algorithm by using this metric instead of backlog. In [16], Mekkittikul et al. propose a delay-based approach that uses head-of-line delays instead of queue lengths. Michael J. Neely analyses the approach with Lyapunov optimization for one-hop wireless networks in [14]. Ji et al. extend the approach to multi-hop wireless networks [15].

With the great progress in the field of data science, machine learning provides another solution for improving the original BP algorithm. In [17], Huang et al. investigate the benefit of predictive scheduling and establish a novel queue-equivalence result based on a look-ahead prediction window model. In [18], Gao et al. combine the traditional BP algorithm with multi-agent Q-Learning, and have shown that it reduces the average packet delay by 95% for light traffic loads.

3. Variants of the BP Algorithm

To improve the delay performance and reduce the memory complexity of the original BP algorithm, we compare four variants in this section, and present their method of operation, advantages, and disadvantages.

3.1. Reduce the Path Length

In [4], Ying et al. aim to minimize the average number of hops per packet delivery, or the average path lengths between sources and destinations. It has two interpretations. First, the number of hops can be thought of as the number of transmissions needed to support traffic. Minimizing it can be regarded as minimizing the network resource. Second, the number of hops is related to end-

doi: 10.2313/NET-2022-01-1_14

to-end delay. Decreasing the number of hops results in a lower delay.

They propose a joint traffic-control and shortest-path-aided BP algorithm. When the traffic is light, the algorithm chooses the shortest paths; when the traffic increases, more paths are exploited to support the traffic. To control the trade-off between shortest path selection and back pressure policy, a tuning parameter K is used. When K becomes quite large, the algorithm only uses the optimal path. The optimal value of K depends on the networks. The strategy does not only guarantee network stability (throughput-optimal), but also adaptively selects the optimal path according to the traffic demand [4].

To study the performance of the algorithm, the authors implement the simulation using OMNeT++. The setup is shown in [4]. All intercluster flows have the same arrival rate denoted by λ (packets/time slot). λ is used to observe the performance of the algorithm under different traffic loads. The performance of shortest-path based BP with different K is shown in Figure 1. A small K results in a small penalty on long paths. For $K = 100$, the penalty on long paths is too large, therefore the algorithm will only prefer the shorter path without considering the backlog length.

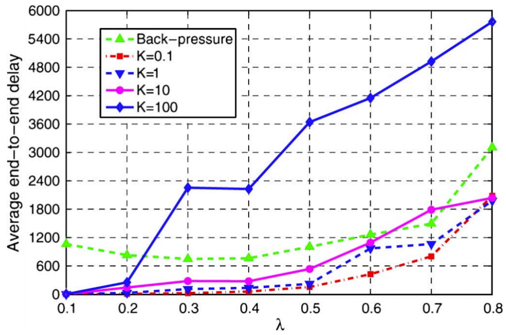

Figure 1: Simulation Result From [4]

The algorithm is simple and the performance is improved compared to traditional BP. It requires the calculation of the multiple source multiple destination shortest path, which can be calculated by Floyd-Warshall in $\mathcal{O}(N^3)$. For each node in the network, they need to store the path information besides the backlog. A pretty similar approach is also introduced in [9], [19].

Another aspect to reduce the path length is to remove loops before applying the BP algorithm as shown in [5]. The authors propose to assign directions to the links so that the network becomes a directed acyclic graph (DAG). Initially, an arbitrary DAG is generated and then the BP algorithm is used. When certain links are overloaded, a new DAG is created by reversing the direction of the links that point from non-overloaded to overloaded nodes. This approach avoids loops, thus the end-to-end delay is decreased. By iteratively creating new DAGs and performing BP, the throughput optimality is guaranteed.

3.2. Cluster the Nodes

In [2], Ying et al. show that the end-to-end delay is decreased without loss of throughput by properly clustering the nodes. The criteria for clustering nodes, e.g. geometrically by the frequency of information exchange, does heavily depend on the network. After clustering, routers need to maintain one queue for each cluster, therefore the variant also reduces the memory complexity significantly compared to the original BP algorithm. The principle is similar to the routing algorithm between different autonomous systems (AS). Packets are sent along the gateway between different clusters.

The cluster-based BP algorithm consists of three components, i.e. **traffic controller, regulator, and back-pressure scheduler**. The traffic controller decides the least congested gateway and how many packets can go through the gateway. The regulator is for limiting how many packets can be transferred through a certain gateway. The back-pressure scheduler decides the best route for transferring the packets.

The authors simulate the algorithm using OMNeT++. The setup is shown in [2]. The simulation result is shown in Figure 2. *Cluster-bp-w/o* and *cluster-bp-w* are two variants of cluster based BP. Both of them are better than the traditional BP according to the figure. Another interesting fact is the dramatic increase of delay of the shortest path algorithm when λ is equal to 0.5. This is because the algorithm only chooses shortest paths. When all paths are fully loaded and the traffic still increases, the delay grows significantly. The authors also propose further variants of cluster-based BP algorithm, such as multilevel clustering and combining it with policy-based routing.

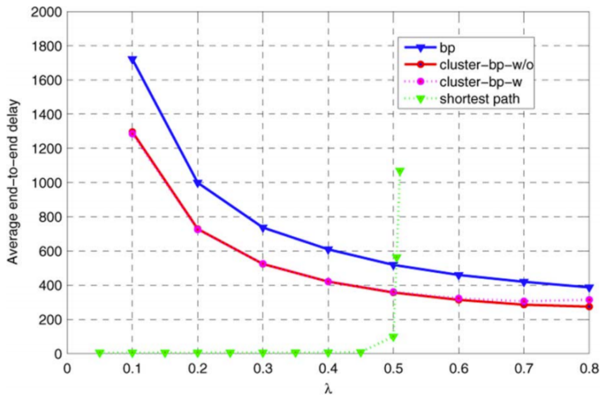

Figure 2: Simulation Result From [2]

In [20], the authors introduce a new metric called greedy back-pressure metric value (GBM). GBM values are evaluated to route the packets toward gateways in the direction of the steepest gradient. It uses a combination of traffic load and the mesh node's hop count to the nearest gateway. The authors show that the GBM based back-pressure algorithm outperforms the traditional BP algorithm. Cluster-based BP can be improved by combining the GBM method, since the GBM method utilizes the gateways more efficiently.

Even though the simulation of [2] gives a decent result, one important problem is how to cluster the nodes properly. The authors give some suggestions such as clustering nodes according to the network topology or physical location. But it is still an open question that needs to be solved.

3.3. Delay Based Algorithm

In [13], the authors introduce a new delay metric called the sojourn time backlog (STB). The STB considers the queue length and accumulated packet delays comprehensively. The authors propose an STB-based back-pressure algorithm called STBP. It applies the BP algorithm based on the STB. STBP routes the packets to a shorter or faster path compared to the traditional BP algorithm. The authors prove that STBP is stable and throughput optimal.

One challenge of implementing STB is to realize time synchronization (a well-known challenge [21]). For packets moving between different nodes in the network, the clocks of these nodes need to be synchronized so that the sojourn time is correctly recorded. The accuracy of the synchronization affects the performance significantly. To overcome the problem, the authors propose to use hop-count instead of exact sojourn time.

To illustrate the performance, the authors simulate the algorithm using the NS-2 network simulator. They compared STBP and STBP-hop based algorithm with the traditional BP algorithm. The setup is described in [13]. The performance is shown in Figure 3. Before the saturated point, all algorithms have similar performance except STBP. After 100 kb/s, STBP and STBP-hop have smaller delays compared to traditional BP. Another fact is that the hop-count (STBP-hop) method performs worse than the STBP method based on the figure.

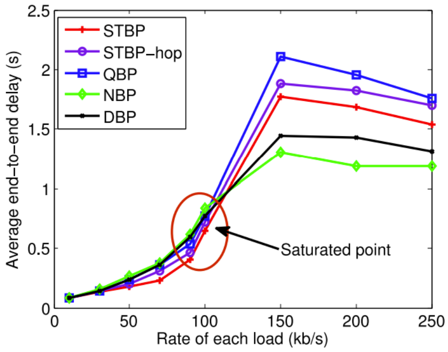

Figure 3: Result From [13], traditional BP is denoted as QBP

In [10], the authors combine the traditional BP algorithm with the LIFO policy. The algorithm transfers the new packets to their destination with less waiting time. In [11] the authors improve the algorithm further by combining the LIFO and the FIFO policy. At every time slot, the algorithm randomly decides to serve packets from either the back of the queue or the front of the queue. It avoids some packets staying in the queue for long time. The authors prove that the algorithm achieves close-to-optimal performance and decreases the delay. This kind of strategy is close to the STBP, which aims to reduce the waiting time of the packets. It is easier to implement and does not require synchronization between different nodes. But it does not always produce the optimal result.

3.4. Combine with Data Science

The fast development of data science draws the attention of researchers in many areas. In [17], [18], the authors combine machine learning with the traditional BP algorithm.

In [17], the authors propose a *lookahead window model* to pre-allocate rates. The lookahead window, also called prediction queue, is constructed by the server according to the previous packets. The lookahead window helps the server to use links more efficiently. They perform the BP algorithm based on the prediction queue. The authors prove that the algorithm achieves a cost performance that is arbitrarily close to the optimality, while guaranteeing that the average system delay vanishes as the prediction window size increases. The reason is that with larger window size, the prediction is more accurate. They also simulate the algorithm in a 10-user single server system. The result shows that when the prediction window size increases, the network delay decreases. However, the algorithm requires more computational power and more time to work properly.

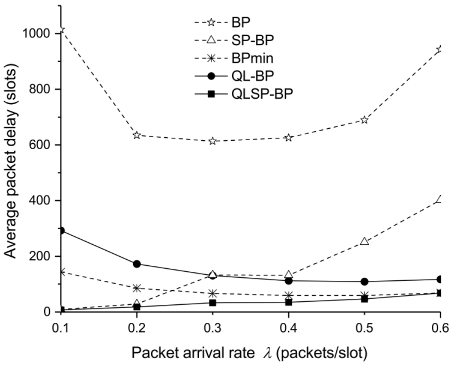

Figure 4: Simulation Result from [18]

In [18], the authors use multi-agent Q-learning to extract biases and based on these biases to perform the BP algorithm. Q-learning is a variant of reinforcement learning, it keeps learning based on the environment according to the reward [22]. Each node in the network estimates route congestion using local information of the neighboring nodes. And every node has multiple Q-learning agents that continuously update its route congestion estimate. Based on the estimate, every node then directs packets via the least congested routes to their destinations.

The algorithm is based on the *Bias Based General Framework* [12]. The framework consists of three parts, i.e. *information collection, bias extraction, and back-pressure routing*. At the information collection stage, the framework collects useful information (local or global) such as queue length, shortest path, and packet delay, for delay reduction. As for bias extraction, the framework extracts useful biases. Finally, the BP algorithm directs packets based on the features from the second stage.

The authors prove that the Q-learning based BP algorithm is throughput optimal. They also simulate the algorithm to test its performance, the result is shown in Figure 4. The traditional BP algorithm is denoted as BP, and QL-BP is the abbreviation for the Q-learning aided back-pressure algorithm. QL-BP's delay performance is much better than original BP.

Since each node extracts the bias based on its local knowledge, this enables distributed implementation. The

TABLE 1: Overview of all variants

Variant	Improvement	Challenge
Reduce path length	Shortest path, Remove loop, LIFO	Balance delay and throughput
Cluster nodes	Cluster nodes using gateway	How to cluster
Delay based algorithm	Delay metric	Time synchronization
Data science	Prediction	Computation, Distribution

computation complexity is low compared to algorithms maximizing the weighted sum globally. Even though it requires computation for extracting biases, but with the growth of the computational power of the electronic components, it provides a good way for improving the backpressure algorithm.

4. Summary and Conclusion

We have introduced four variants of BP based algorithms. Table 1 shows the overview of all variants. All methods reduce the delay compared to the traditional BP algorithm. The Cluster-based algorithm reduces the memory complexity. With the rapid development of data science, machine learning has potential to decrease the delay further.

There are still challenges. The balance between delay and throughput is an important problem for path-related algorithms. As for clustering, how to cluster nodes in the network properly is still an open question, and it is also a fundamental requirement for applying the cluster-based algorithm. To use the delay-based algorithm, it is crucial to synchronize the time. Otherwise, the nodes are not able to record the delay of the packets properly and thus the algorithm cannot work. For machine learning methods, it requires more computation time, and nodes need to constantly record the data and update the parameters.

In this paper, we discussed the advantages and disadvantages of every variant. Reducing the path length can be achieved by the shortest path algorithm, removing loops in networks, using DAGs or LIFO. This variant reduces the delay when the traffic load is light. Clustering the nodes properly can reduce the memory complexity and end-to-end delay. The delay-based algorithm reduces the delay while guaranteeing throughput optimality. The last variant is machine learning. Machine learning helps to route the packets efficiently with near throughput optimality. We also proposed potential combinations of different variants, such as combining clustering method with GBM 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 Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936–1949, 1952.

[2] Ying, Lei and Srikant, R. and Towsley, Don and Liu, Shihuan, “Cluster-Based Back-Pressure Routing Algorithm,” *IEEE/ACM Transactions on Networking*, vol. 19, no. 6, pp. 1773–1786, 2011.

[3] Gao, Juntao and Shen, Yulong and Ito, Minoru and Shiratori, Norio, “Bias Based General Framework for Delay Reduction in Backpressure Routing Algorithm,” *2018 International Conference on Computing, Networking and Communications (ICNC)*, pp. 215–219, 2018.

[4] Ying, L. and Shakkottai, S. and Reddy, A., “On Combining Shortest-Path and Back-Pressure Routing Over Multihop Wireless Networks,” *IEEE INFOCOM 2009*, pp. 1674–1682, 2009.

[5] A. Rai and C. Li and G. Paschos and E. Modiano, “Loop-Free Backpressure Routing Using Link-Reversal Algorithms,” *IEEE/ACM Transactions on Networking*, vol. 25, no. 05, pp. 2988–3002, 2017.

[6] Alresaini, Majed and Sathiamoorthy, Maheswaran and Krishnamachari, Bhaskar and Neely, Michael J., “Backpressure with Adaptive Redundancy (BWAR),” *2012 Proceedings IEEE INFOCOM*, pp. 2300–2308, 2012.

[7] Ji, Zhe and Wang, Youzheng and Lu, Jianhua, “Distributed Delay-Aware Resource Control and Scheduling in Multihop Wireless Networks,” *2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*, pp. 1–5, 2015.

[8] Bui, Loc X. and Srikant, R. and Stolyar, Alexander, “A Novel Architecture for Reduction of Delay and Queueing Structure Complexity in the Back-Pressure Algorithm,” *IEEE/ACM Transactions on Networking*, vol. 19, no. 6, pp. 1597–1609, 2011.

[9] Athanasopoulou, Eleftheria and Bui, Loc X. and Ji, Tianxiong and Srikant, R. and Stolyar, Alexander, “Back-Pressure-Based Packet-by-Packet Adaptive Routing in Communication Networks,” *IEEE/ACM Transactions on Networking*, vol. 21, no. 1, pp. 244–257, 2013.

[10] Moeller, Scott and Sridharan, Avinash and Krishnamachari, Bhaskar and Gnawali, Omprakash, “Routing without Routes: The Backpressure Collection Protocol,” *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, p. 279–290, 2010.

[11] Huang, Longbo and Moeller, Scott and Neely, Michael J. and Krishnamachari, Bhaskar, “LIFO-Backpressure Achieves Near-Optimal Utility-Delay Tradeoff,” *IEEE/ACM Transactions on Networking*, vol. 21, no. 3, pp. 831–844, 2013.

[12] Gao, Juntao and Shen, Yulong and Ito, Minoru and Shiratori, Norio, “Bias Based General Framework for Delay Reduction in Backpressure Routing Algorithm,” *2018 International Conference on Computing, Networking and Communications (ICNC)*, pp. 215–219, 2018.

[13] Hai, Long and Gao, Qinghua and Wang, Jie and Zhuang, He and Wang, Ping, “Delay-Optimal Back-Pressure Routing Algorithm for Multihop Wireless Networks,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2617–2630, 2018.

[14] Neely, Michael J., “Delay-Based Network Utility Maximization,” *2010 Proceedings IEEE INFOCOM*, pp. 1–9, 2010.

[15] Ji, Bo and Joo, Changhee and Shroff, Ness B., “Delay-Based Back-Pressure Scheduling in Multihop Wireless Networks,” *IEEE/ACM Transactions on Networking*, vol. 21, no. 5, pp. 1539–1552, 2013.

[16] McKeown, N. and Mekkittikul, A. and Anantharam, V. and Walrand, J., “Achieving 100% Throughput in An Input-queued Switch,” *IEEE Transactions on Communications*, vol. 47, no. 8, pp. 1260–1267, 1999.

[17] Huang, Longbo and Zhang, Shaoquan and Chen, Minghua and Liu, Xin, “When Backpressure Meets Predictive Scheduling,” *IEEE/ACM Transactions on Networking*, vol. 24, no. 4, pp. 2237–2250, 2016.

[18] Juntao Gao and Yulong Shen and Minoru Ito and Norio Shiratori, “Multi-Agent Q-Learning Aided Backpressure Routing Algorithm for Delay Reduction,” *CoRR*, vol. abs/1708.06926, 2017.

[19] Yin, Ping and Yang, Sen and Xu, Jun and Dai, Jim and Lin, Bill, “Improving Backpressure-based Adaptive Routing via Incremental Expansion of Routing Choices,” *2017 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS)*, pp. 1–12, 2017.

[20] Hu, Bin and Gharavi, Hamid, “Greedy Backpressure Routing for Smart Grid Sensor Networks,” *2014 IEEE 11th Consumer Communications and Networking Conference (CCNC)*, pp. 32–37, 2014.

[21] “Time Synchronization,” https://www.cs.usfca.edu/~srollins/courses/cs686-f08/web/notes/timesync.html.

[22] “An introduction to q-learning: Reinforcement learning,” https://blog.floydhub.com/an-introduction-to-q-learning-reinforcement-learning/.