



Adaptive Routing in Communication Network Based on PS Table

S. ASHOK KUMAR¹, A. DAMODHAR²

¹PG Scholar, Dept of CSE, Sri Venkateswara College of Engineering, Tirupati, Andhrapradesh, India.

²Asst Prof, Dept of CSE, Sri Venkateswara College of Engineering, Tirupati, Andhrapradesh, India.

Abstract: In the literature we have studied each packet is routed along a possibly different path by using Back Pressure based adaptive routing algorithm. So there is poor delay performance and involve high implementation complexity. After studied Back Pressure algorithm with clearly, we have developed a new adaptive routing algorithm. Here we have designed probabilistic routing table that is used to route packets to per destination queue to decouple the routing and scheduling components of the algorithm. In the case of wireless networks the scheduling decisions are made using counters called shadow queues. The results are also extended to the case of networks that employ simple forms of network coding. In that case, our algorithm provides a low-complexity solution to optimally exploit the routing-coding tradeoff.

Keywords: Network Coding, Routing, Scheduling.

I. INTRODUCTION

The traditional back-pressure algorithm, each node n has to maintain a queue q_{nd} for each destination d : Let $|D|$ and $|D|$ denote the number of nodes and the number of destinations in the network, respectively. Each node maintains queues. Generally, each pair of nodes can communicate along a path connecting them. Thus, the number of queues maintained at each node can be as high as one less than the number of nodes in the network, i.e., $|D| = |D| - 1$: In proposed system, the main purpose of this paper is to study the case of scheduling and routing the shadow queue extends, which brings new invention that the number of hops is minimized. In the antagonism the objectives of the invention is same, the solution involves per hop queue as compared to backpressure algorithm. In this paper, we have used different types of solution. Small number of real queues used as per neighbor, but the number of shadow queues is same as back pressure algorithm. The shadow queue size always upper bounds the real queue size, it follows that the real queue is also assured to be stable.

The advantage of this approach is that buildup of the shadow queues can take place to provide a routing "gradient" for the backpressure algorithm without corresponding buildup (and so packet delay) of the real queues, but at the cost of compact network capacity. So we brought a new idea which allows the reduction in the number of real queues by routing via probabilistic splitting. One more important observation in this paper to reduce delays in routing case because of partial decoupling of shadow back-pressure and real packet transmit allows us to activate more links as compare to regular back-pressure algorithm. By the modification of our routing algorithm automatically it balances with good performance. This is very good advantage for our proposed system instead

of keeping a queue for every destination, each node n maintains a queue n_j for every neighbor j ; which is called a real queue. Notice that real queues are per-neighbor queues. Let J_n denote the number of neighbors of node n ; and let $J_{\max} = \max_n J_n$: The number of queues at each node is no greater than J_{\max} . Generally, J_{\max} is much smaller than $|D|$: Thus, the number of queues at each node is much smaller compared with the case using the traditional back-pressure algorithm. In addition to real queues, each node n also maintains a counter, which is called shadow queue, p_{nd} for each destination d : Unlike the real queues, counters are much easier to maintain even if the number of counters at each node grows linearly with the size of the network. A backpressure algorithm run on the shadow queues is used to decide which links to activate. The statistics of the link activation are further used to route packets to the per-next-hop neighbor queues mentioned earlier.

A. Shadow Queue Algorithm

Traditional Back Pressure Algorithm is same as the Shadow algorithm but, the shadow algorithm works on the bases of shadow queuing. Here every node upholds a fictitious queue called shadow queue. These shadow queues are work as counter for every flow. By the movement of fictitious entities called shadow packets the shadow queues are updated. These packets are used for the purpose of scheduling and routing as an exchange of control messages. The shadow queue as counter it is incremented by 1 when packets are arrival, and decremented by 1 when these packets are departure. The packet arrival rate is slightly larger than the real external arrival rate of packets. . Just like real packets, shadow packets arrive from outside the network and eventually exit the network. The evolution of the shadow queue $p_{nd}[t]$ is

$$P_{nd}[t+1] = P_{nd}[t] - \sum_{k=j;(nj)} I_{\{d_{nj}^*[t]=d\}} \hat{\mu}_{nj}[t] + \sum_{k:(ln)\in\tau} I_{\{d_{nj}^*[t]=d\}} \hat{\mu}_{nj}[t] + \sum_{f\in\tau} I_{\{b(f)=n..a(f)=d\}} \hat{a}_f(t) \quad (1)$$

B. Adaptive Routing Algorithms

Now we discuss about packets how it routes once when it arrives at a node. Let us define a variable called $\sigma_{nj}^d[t]$ has number of shadow packets, which are transferred from node say n to node j for destination d during time slot t by the shadow queue algorithm as shown in Fig.1. When shadow queuing process is in a stationary command, the value of $\sigma_{nj}^d[t]$ is denoted by $\sigma_{nj}^d[t]$ and it estimated at. time t. A packet arriving at node n for destination d is inserted in the real queue qnj for next-hop neighbor j with probability

$$\sigma_{nj}^d[t] = \frac{\sigma_{nj}^d[t]}{\sum_{K:(nK)\in\tau} \sigma_{nk}^d[t]} \quad (2)$$

Also notice that $\sigma_{lnj}^d[t]$ is contributed by shadow traffic point-to-point transmission as well as shadow traffic broadcast transmission,

TABLE I: Probability Routing Table

Destination	(Next-hop, probability) Pairs
1	(1,pid(1)),.....,(j,pid(j)),....(N,pid(n))
:	:
D	(1,pid(1)),.....,(j,pid(j)),....(N,pid(n))

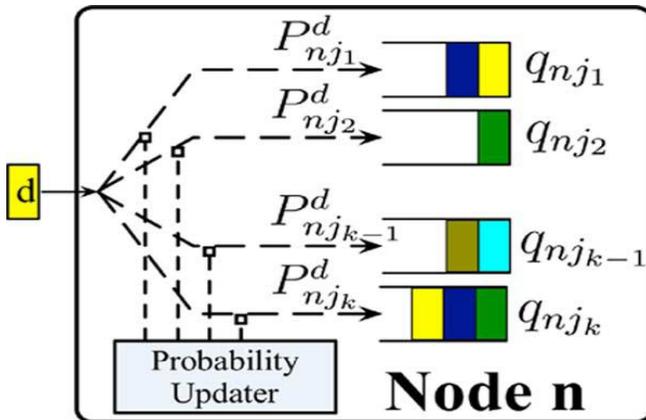


Fig.1. Probabilistic splitting algorithm at node n.

II. TECHNICAL OVERVIEW

We have studied in the literature about back pressure algorithm which is introduced in[2]. While the ideas behind scheduling using the weights suggested in that paper have been successful in practice in base stations and routers, the adaptive routing algorithm is rarely used. The previous work carried out in [3] has accepted the significance of under taking shortest path routing to improve performance of delay and the algorithm of back pressure has modified to bias it towards taking shortest hop routs. A part of our algorithm has related inspiring idea. In the network the throughput optimal routing minimizes the number of hops, which are taken by packets. We use probabilistic routing tables also called as

shadow queue used for scheduling in the network. In conference paper [4] the idea of min hop routing was studied first.

In [5] and [10] the shadow queues were introduced, but in this paper the main step of incomplete decoupling the routing and scheduling, where indicate to both substantial delay reduction and the use of per-next-hop queuing is original here. To solve a fixed routing problem the, the authors introduced shadow queues in [5]. In [11] we studied the min hop routing idea, so we require more queues than the original back pressure algorithm. In this paper we compare with [5] we study the shadow queues methodology covers the case of scheduling and routing. We consider some network where the packets are XORs and broadcast them to decrease the transmission between two nodes by

III. IMPLEMENTATION

1. Exponential Averaging: In this module, using the concept of shadow queues, we partially decouple routing and scheduling. A shadow network is used to up-date a probabilistic routing table that packets use upon arrival at a node. The same shadow network, with back-pres-sure algorithm, is used to activate transmissions between nodes. However, first, actual transmissions send packets from first-in-first-out (FIFO) per-link queues, and second, potentially more links are activated, in addition to those activated by the shadow algorithm [9] we use the following iterative Exponential averaging algorithm

$$\sigma_{nj}^d[t] = (1-\beta) \sigma_{nj}^d[t-1] + \beta \sigma_{nj}^d[t] \quad (3)$$

Where $0 < \beta < 1$

2. Token Bucket Algorithm: In this unit, the traditional method which compute the average shadow rate and producing arbitrary numbers for routing packets may impose a computational overhead of routers, which should be avoided if possible. Thus, as substitute, we suggest the following simple algorithm. At each node n, for each next-hop neighbor j and each destination d, maintain a token bucket. Consider the shadow traffic as a guidance of the real traffic, with tokens removed as shadow packets traverse the link. In detail, the token bucket is decremented by each time slot, but cannot below the lower bound 0;

$$\gamma_{nj}^d[t] = \text{mak}\{\gamma_{nj}^d[t-1] - \sigma_{nj}^d[t] - \sigma_{nj}^d[t], 0\} \quad (4)$$

3. Extra Link Activation: Links with backpressure can be activated greater than or equal to parameter only under the shadow back pressure algorithm. This can adequate to condense the real queues. But the delay recital can still be deplorable. Use of unnecessarily long path can be disappointed. So to avoid this we introduce the parameter. The shadow back pressure at a link may be habitually less than this parameter, when light and moderate traffic loads. Because of this the packets are processed after waiting a long time at this links. To cure these circumstances we can establish additional links. With the extra activation, a certain degree of decoupling between routing and scheduling is achieved.

Adaptive Routing in Communication Network Based on PS Table

4. Extension to the Network Coding Case: In this fragment, we spread out tactic to reflect networks, where network coding is used to progress throughput. We use network coding which reduces the transmission between two nodes as shown in Fig.2. Suppose if a node i wants to send some packets to node j , for this as per traditional back pressure it has transmit i to n and n to j again j to n and n to i . so it requires more transmission. To avoid such kind of transmission we use intermediate relay say n . Here the two of the packets are gets XORed and simultaneously it broadcast two of them to i and j . From this we can reduces the number of transmission. We need to design to build an algorithm to find right adjustment by via possible long routes to arrange for network coding prospects and delay incurred by using long routes.

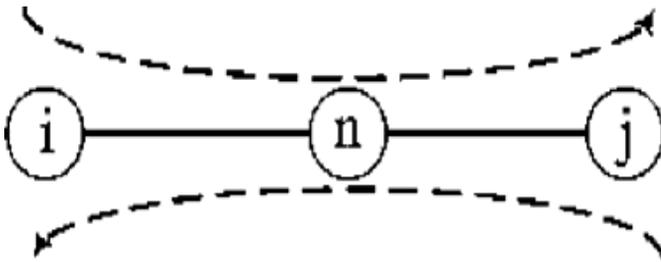


Fig.2. Network Coding Oppurtunity.

IV. SIMULATION

Wire line and wireless are the two networks. We consider these two networks in our simulation. Her we see the topology of these two and also simulation parameter which is used in our simulation

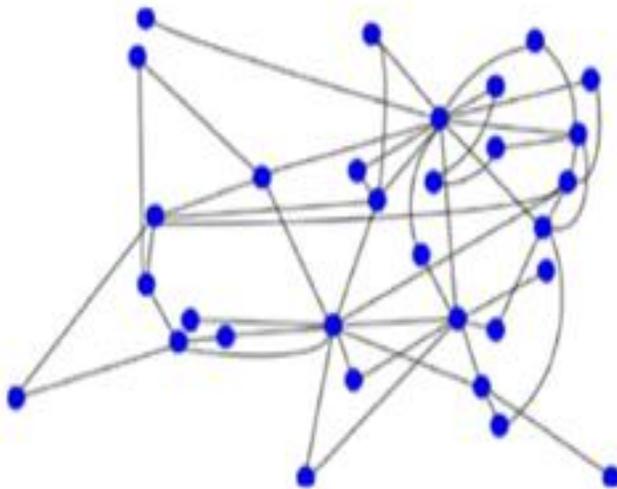


Fig.3. GMPLS Network topology with 31 nodes.

A. Wireline Setting

The fig.3 shows 31 nodes with GMPLS topology. Here every link assume to be transmit 1 packet to each slot we assume that the arrival process is a Poisson process with parameter; and we consider the arrivals come within a slot are considered for service at the beginning of the next slot. Once a packet arrives from an external flow at a node n , the destination is decided by probability mass 2. Wireless

Setting: We used the following procedure to generate the random network: 30 nodes are placed uniformly at random in a unit square; then starting with a zero transmission range, the transmission range was increased till the network was connected. We assume that each link can transmit one packet per time-slot. We assume a 2-hop interference model in our simulations. By a -hop interference. model, we mean a wireless network where a link activation silences all other links that are hops from the activated link. The packet arrival processes are generated using the same method as in the wireline case. We simulate two cases given the network topology: the no coding case and the network coding case. In both wireline and wireless simulations, we chose to be, and we use probabilistic splitting algorithm for simulations.

B. Simulation Results

1. Wireline Networks: First, we compare the performance of three algorithms: the traditional back-pressure algorithm, the basic shadow queue routing/scheduling algorithm without the extra link activation enhancement and PARN. Without extra link activation, to ensure that the real arrival rate at each link is less than the link capacity provided by the shadow algorithm. We also compare the delay performance of PARN with that of the shortest path routing in Fig.4 For each pair of source and destination, we find a hortest path between them by using Dijkstra's algorithm However, a wireline network does not capture the scheduling aspects inherent to wireless networks, which is studied next.

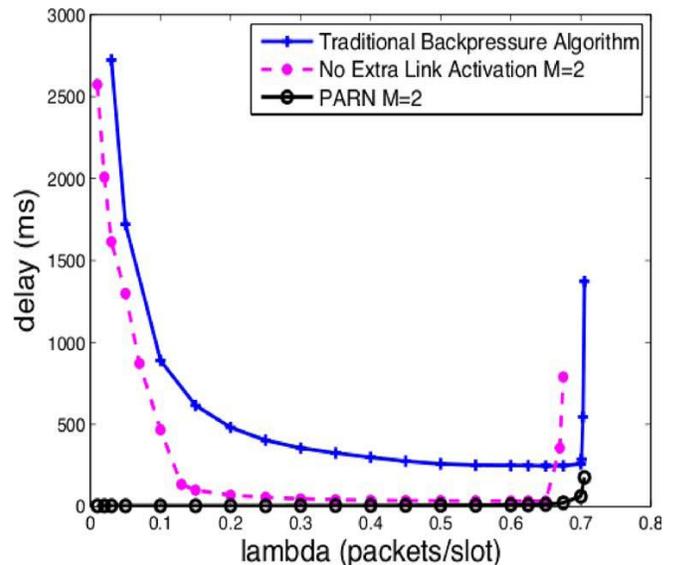


Fig.4. Impact of the parameter M and extra link activation in Sprint GMPLS network topology.

2. Wireless Networks: However, a wireline network does not capture the scheduling aspects inherent to wireless networks, we need we study wireless networks without network coding (Fig.9). Here the delay performance is relatively insensitive to the choice of as long as it is sufficiently greater than zero. However, does play an important role because it suppresses the search of long paths when the traffic load is not high.

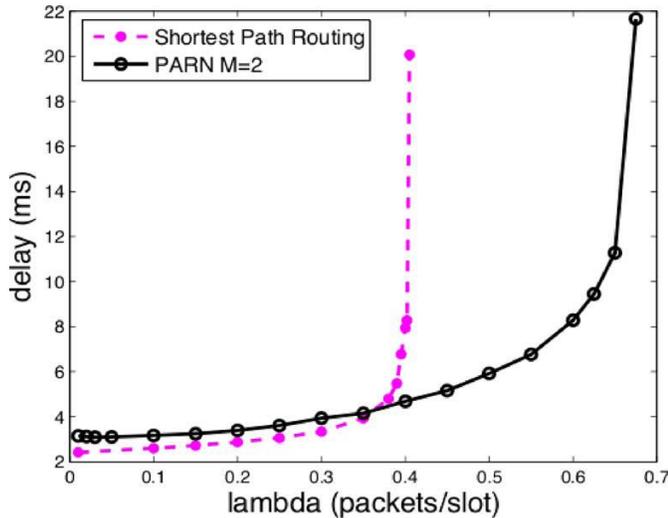


Fig.6. Delay performance of PARN and shortest path routing.

V. CONCLUSION

The back-pressure algorithm, while being throughput-optimal, is not useful in practice for adaptive routing since the delay performance can be really bad. In this paper, we have presented an algorithm that routes packets on shortest hops when possible and decouples routing and scheduling using a probabilistic splitting algorithm built on the concept of shadow queues. By maintaining a probabilistic routing table that changes slowly over time, real packets do not have to explore long paths to improve throughput; this functionality is performed by the shadow “packets.” Our algorithm also allows extra link activation to reduce delays. The algorithm has also been shown to reduce the queueing complexity at each node and can be extended to optimally tradeoff between routing and network coding.

VI. REFERENCES

- [1] Eleftheria Athanasopoulou, Member, IEEE, LocX.Bui, Associate Member, IEEE, Tianxiong Ji, Member, IEEE.
- [2] R. Srikant, Fellow, IEEE, and Alexander Stolyar.
- [3] L. Tassiulas and A. Ephremides, “Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks,” *IEEE Trans. Autom. Control*, vol. 37, no. 12, pp. 1936–1948, Dec. 1992.
- [4] M. J. Neely, E. Modiano, and C. E. Rohrs, “Dynamic power allocation and routing for time varying wireless networks,” *IEEE J. Sel. Areas Commun.*, vol. 23, no. 1, pp. 89–103, Jan. 2005.
- [5] L. Bui, R. Srikant, and A. L. Stolyar, “Novel architectures and algorithms for delay reduction in back-pressure scheduling and routing,” in *Proc. IEEE INFOCOM Mini-Conf.*, Apr. 2009, pp. 2936–2940.
- [6] L. Bui, R. Srikant, and A. L. Stolyar, “A novel architecture for delay reduction in the backpressure scheduling algorithm,” *IEEE/ACM Trans. Netw.*, vol. 19, no. 6, pp. 1597–1609, Dec. 2011.
- [7] L. Bui, R. Srikant, and A. L. Stolyar, “Optimal resource allocation for multicast flows in multihop wireless

networks,” *Phil. Trans. Roy. Soc., Ser. A*, vol. 366, pp. 2059–2074, 2008.

[8] Sprint IP network performance. [Online] Available: <https://www.sprint.net/performance/>.

[9] On Delay Performance of Throughput Optimal Back-pressure Routing: Testbed Results Venkataramana Badarla and Douglas J. Leith Hamilton Institute, NUI Maynooth, Ireland.

[10] The Delay Performance of Adaptive routing and scheduling in communication network by Tianxiong Ji.

[11] L. Ying, S. Shakkottai, and A. Reddy, “On combining shortest-path and back-pressure routing over multihop wireless networks,” in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 1674–1682.

[12] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, “XORs in the air: Practical wireless network coding,” *Comput. Commun. Rev.*, vol. 36, pp. 243–254, 2006.

Author’s Profile:



Ashok Kumar.S received the B.Tech Degree in the Stream of Information Technology From Sri Venkatesa Perumal College of Engineering & Technology, Puttur. University of JNTUA in 2012. He is currently working towards the Master’s Degree in the Stream of Computer Science & Engineering, in Sri Venkateswara College of Engineering, Tirupati. University of JNTUA. His interest lies in the areas of Computer Networks, Dbms, Uml.



A. Damodhar Received B.Tech in Computer Science and Engineering from AITS, Rajampet in 2009 & M.Tech in Computer Science and Engineering from SITAMS, Chittoor in 2011 respectively. Currently He is an Assistant Professor in the Department of Computer Science and Engineering at SV College of Engineering-Tirupati. He has published a paper titled “A Rectangle filling using Divide & Conquer Methods” in journals and referred conference proceedings. His Current Interest Include Computer Networks, Operating Systems, Information Security & Software Engineering.