ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 218 - 222 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

A Joint Congestion Control and Scheduling Algorithm for Decreasing the Delay in Multihop Wireless Networks

Poluri Koteswararao[1]K Subbarao[2]

[1] M.Tech Student, Dept of CSE, St. Ann's College of Engineering Technology, Chirala, Prakasam Dist, A.P, India

[2]Associate Professor, Dept of CSE, St. Ann's College of Engineering Technology, Chirala, Prakasam Dist, A.P, India

ABSTRACT: ***In this paper a novel method is prepared that quantifies the end-to-delay performence in multihop wireless network. A general interference model with fixed route flows can be controlled to reduce the congestion with interference degree K.(which is near a minimum of the system capability region), This method quantifies a provable throughput and also leads to explicit maximum end-to-end delay for each flow. The trade off between throughput and delay is explicitly quantified and also per-flow end-to-end delay increases linearly as the number of hops increased. In this the window based and rate based algorithms are combined.A key contribution of our work is to use a completely unique random dominance approach to certain the corresponding per-flow throughput and delay, that otherwise typically wild in these forms of systems. This is a fully distributed and requires a low per-node complexity that doesn't increase the network size. Hence, it are often simply enforced in observe.***

Index words: multihop, region, fixed route, end-to-end delay.

INTRODUCTION: THE JOINT congestion management and programming downside in multihop wireless networks have been extensively studied within the literature [1], [2]. Often, every user is associated with a non decreasing and cotyloidal utility operate of its rate, and a cross-layer utility maximization downside is developed to maximize the entire system utility subject to the constraint that the speed vector is supported by some programming algorithm. One best resolution to the current downside is understood to be the max-weight back-pressure programming formula combined with a congestion management element at the supply [1], [2]. Furthermore, vital progress has been created in coming up with distributed programming algorithms with demonstrable turnout and lower complexness than the back-pressure formula [3]–[8]. However, most of the present works on joint congestion management and programming have solely thought-about the turnout performance metric and haven't accounted for delay performance issues. Though for flows with congestion management (e.g., file transfer) the turnout is commonly the foremost essential performance metric, packet delay is very important furthermore as a result of sensible congestion management protocols have to be compelled to set retransmission timeout values supported the packet delay, and such parameters may significantly impact the speed of recovery once packet loss occurs. Packet delay is additionally vital for transmission traffic, some of that are carried on congestion-controlled sessions.In this paper, we are going to offer a brand new category of joint congestion control and programing algorithms1 that may win each provable outturn and obvious per-flow delay. Then consider flows in a very multihop network in operation below a general interference model with the interference degree (the notion of the interference degree are going to be given in Section II), then each flow is given a set route with hops. Our formula consists of 3 main components: window-based flow management, virtual-rate computation, and programing. The most concepts of our formula to enhance the end-to-end delay area unit as follows.First, by victimisation window-based flow management, we will tightly control the quantity of packets within the network. Second, by using a rate-based programing formula with the computed virtual rate as input to schedule packets, we tend to don't have to be compelled to wait for the packets to accumulate before creating programing choices. However, the key issue in analyzing the end-to-end throughput and delay below this formula is that the services at completely different links area unit related. Hence, a Markoff chain analysis will now not offer a closed-form resolution. We employ a novel random dominance technique to avoid this issue and derive closed-form bounds on the per-flow outturn and delay. Recently, there are variety of papers that quantify the delay performance of the wireless networks with or while not congestion control [9], [11]–[20]. In [9] and [11], the authors propose methods to cut back the delay of the back-pressure algorithmic rule. The algorithm planned in [9] could be a shadow back-pressure algorithmic rule, which maintains one first-in–first-out (FIFO) queue connected delay certain may be shown for the programing algorithmic rule while not congestion management [10]. However, the delay analysis of joint congestion management and programing during this paper is harder because of the closed-loop feedback. Link and uses the multiple shadow queues to schedule the transmissions. This methodology decouples the management info from the real queues and thence reduces the delay. In our simulation, this algorithm looks to attain linear delay once the algorithmic rule converges. However, at the transient amount, the important queues can still follow the shadow queues,that ends up in an outsized queue backlog. In [11], the authors propose another mechanism, FQLA, to decouple the management signal from the important queues by injecting place holder bits into the queues. However, the FQLA algorithmic rule needs associate degree initial period to see the place holder bits for every queue. In this initial amount, traditional BP algorithmic rule is employed. Hence, the delay performance throughout this transient part would be comparable to that of the BP algorithmic rule at the best. At the tip of this first period, a major fraction of packets (that correspond to the placeholder bits) should be born. In distinction, our planned algorithmic rule does not drop packets that are admitted, and therefore the delay performance within the transient part doesn't deviate considerably from the worth once convergence. Finally, neither [9] nor [11] provides closed-form bounds on the per-flow end-to-end delay. Our result's additionally completely different from different works in providing a per-flow end-to-end delay certain. First, [12] and [13] solely prove delay bounds for single-hop flows instead of multihop flows. Second, [14]–[16] take into account the delay among all the flows rather than the per-flow delay. Similarly, the ends up in [17] may be used to construct a certain on the delay averaged over all flows. However, it's still not a per-flow delay certain. In distinction, our per-flow delay certain scales with the quantity of hops of the flow itself; thence, it's typically abundant tighter. A per-flow delay certain is provided in [18], however the certain scales with the scale of the network. Finally, a single-flow end-to-end delay analysis is given in [19] supported associate degree approximation of the departure method for each hop. However, it's unclear a way to extend the analysis to multiple flows.

JOINT CONGESTION CONTROL AND SCHEDULINGALGORITHM:

As we tend to mentioned in Section I, there are several approaches available within the literature to resolve (1), and most of them do not think about delay performance. A typical best resolution can be obtained by this approach that results into the back-pressure algorithmic program and a congestion-control part at the supply node [1], [2]. What is most a substantial amount of effort has centered on developing low-complexity and distributed programming algorithms which will replace the centralized back-pressure algorithmic program and nonetheless still deliver the goods demonstrably good outturn performance [3]–[8]. Just like the back-pressure algorithm, these low-complexity programming algorithms are usually additionally queue-length-based. the downside of those approaches, however, is that the end-to-end delay of the ensuing queue-length-based programming algorithmic program is extremely troublesome to quantify, and as we tend to delineated in Section I, below sure cases the back-pressure algorithmic program will have poor delay performance [9], [24]. during this paper, we will use a window-based flow management algorithmic program and a rate-based programming algorithmic program that are terribly completely different from the back-pressure algorithmic program. Our solution strategy is to 1st more or less solve (1) and reason.

WINDOW-BASED FLOW CONTROL ALGORITHM

Now, we have a tendency to describe the congestion management part. Our approach is to use the window-based flow management. For every flow, we have a tendency to maintain a window at the supply node, and we only inject new packets to the queue at the supply node once the total range of packets for this flow within the network is smaller than the window size. this could be achieved by material possession the destination node send Associate in nursing acknowledgement (ACK) back to the supply node whenever it receives a packet. There square measure 2 blessings for this approach. First, for every flow, we are able to tightly control the utmost range of packets in every intermediate node on the route. This may forestall buffer overflows, which is a crucial issue as self-addressed in [17]. Second, as we will show in Section IV, every flow's exchange between output and delay will be severally controlled by the window size. Note that when we gift the analysis, we assume that there's a feedback channel from the destination node to the source node at every time-slot. Through this feedback channel, the destination node will send the ACK to the supply node, and the supply node will then decide if it's doable to inject another packet at subsequent time-slot. In reality, so as to achieve the supply node, every ACK also will bear the whole route in a hop-by-hop fashion within the reverse direction. We will discuss however this could be achieved by piggybacking the ACK once every packet transmission. As readers can see, this method will be analyzed with constant approach bestowed in and this additional ACK delay doesn't amendment the delay order of our result.

SIMULATION RESULTS

We begin from simulating our projected formula exploitation the linear topology in Fig. one with links beneath the one-hop interference constraint. The capacities of the left four links are 5, 2, 3, and 4, severally. Then, each four links repeat this pattern. We have a tendency to use the improved version of our programming algorithm as mentioned, and that we let the supply node (resp. every link) collect the add of the twin variables on the path asynchronously. we have a tendency to set the quantity of back off mini-slots. The window size of every short flow is if the short flow passes link. The window size of the long flow .The utility operate is for the long flow and for every short flow. Hence, once we increase the number of hops, the best rate assignment for the flows will be roughly a similar. This can facilitate U.S. to look at however the average delay changes because the variety of hops will increase whereas the throughput is comparatively fastened. We conjointly use BP- to represent the back-pressure formula with step size and SBP- to represent the shadow back-pressure formula with step size [9].The corresponding long-flow turnout is roughly an equivalent as the BP algorithmic rule, that is perfect. However, just like the BP algorithmic rule, the SBP algorithmic rule needs centralized computation to achieve most capability region. What is more, we observe that SBP needs roughly 7000 time-slots for the entire algorithmic rule to converge, and therefore the total queue length within the network will 1st rise to a really massive. In distinction, the management variables below our algorithmic rule can converge after around two hundred time-slots. Furthermore, thanks to window-based flow management, the total queue backlog of our algorithmic rule is systematically rock bottom in the slightest degree time, even throughout the transient amount. Finally, we have a tendency to plot the delay evolution of the outbound. Specifically, we have a tendency to plot the common delay over the previous 200 outbound packets right before every outbound packet. The simulation result shows that the common delay of SBP can be considerably larger within the transient part. In distinction, the common delay of our algorithmic rule doesn't deviate considerably from the theoretical worth even within the transient amount we have a tendency to demonstrate the per-flow controllability of our scheme by plotting the throughput-delay curve for the long flow. We 1st fix the window size of every short flow to be if the short flow passes link. We have a tendency to then vary the window size of the long flow. Because the window size will increase, the common turnout

CONCLUSION

In this paper, we tend to propose a low-complexity and distributed algorithm for joint congestion management and programing in multihop wireless networks below a general interference model. The main concepts of the projected algorithmic program is to manage the congestion with window-based flow management and to use each virtual-rate info and queue info (rather than simply queue information) to perform programing. Our programing algorithmic program is absolutely distributed and solely needs a continuing time (independent of network size) to work out a schedule [8]. We prove that our congestion management and programing algorithmic program will utilize nearly of the capability region and supply a per-flow delay sure that will increase linearly with the amount of hops. Our analysis uses a completely unique random dominance approach to derive the per-flow outturn and delay bounds. In our future work, we will study a way to extend this novel technique to the case with dynamic routing.

REFERENCES

[1] L. Georgiadis, M. J. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," *Found. Trends Netw.*, vol. 1, no. 1, pp. 1–144, 2006.

[2] X. Lin, N. B. Shroff, and R. Srikant, "A tutorial on cross-layer optimization in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1452–1463, Aug. 2006.508 IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 21, NO. 2, APRIL 2013.

[3] E. Modiano, D. Shah, and G. Zussman, "Maximizing throughput in wireless networks via gossiping," in *Proc. ACM SIGMETRICS*, 2006, pp. 27–38.

[4] S. Sanghavi, L. Bui, and R. Srikant, "Distributed link scheduling with constant overhead," in *Proc. ACM SIGMETRICS*, 2007, pp. 313–324.

[5] X. Lin and N. B. Shroff, "The impact of imperfect scheduling on cross-layer congestion control in wireless networks," *IEEE/ACMTrans. Netw.*, vol. 14, no. 2, pp. 302–315, Apr. 2006.

[6] P. Chaporkar, K. Kar, and S. Sarkar, "Throughput guarantees through maximal scheduling in wireless networks," *IEEE Trans. Inf. Theory*, vol. 54, no. 2, pp. 572–594, Feb. 2008.

[7] C. Joo, X. Lin, and N. B. Shroff, "Understanding the capacity region of the greedy maximal scheduling algorithm in multi-hop wireless networks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 4, pp. 1132–1145, Aug. 2009.

[8] A. Gupta, X. Lin, and R. Srikant, "Low-complexity distributed scheduling algorithms for wireless networks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 6, pp. 1846–1859, Dec. 2009.

[9] 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.*, 2009, pp. 2936–2940.

[10] P.-K. Huang and X. Lin, "The end-to-end delay performance of a class of wireless scheduling algorithms," in *Proc. Allerton Conf. Commun.,Control, Comput.*, 2010, pp. 951–952.

[11] L. Huang and M. J. Neely, "Delay reduction via Lagrange multipliers in stochastic network optimization," in *Proc. WiOpt*, 2009, pp. 1–10.

[12] M. J. Neely, "Delay-based network utility maximization," in *Proc.IEEE INFOCOM*, 2010, pp. 1–9.

[13] M. J. Neely, "Delay analysis for maximal scheduling in wireless networks with bursty traffic," in *Proc. IEEE INFOCOM*, 2008, pp. 6–10.

[14] L. B. Le, K. Jagannathan, and E. Modiano, "Delay analysis of maximum weight scheduling in wireless ad hoc networks," in *Proc. IEEECISS*, Baltimore, MD, Mar. 2009, pp. 389–394.

AUTHORS:

PoluriKoteswararao received the **B.Tech(CSE)** from Acharya Nagarjuna University GUNTUR, in 2011&his pursuing his M.Tech in Computer Science & Engineering from JNTU Kakinada.

Mr.K.Subbaraois presently working as aAssociate Professor, Dept of Computer Science and Engineering, in St.Ann's College of Engineering and Technology, Chirala.He is pursuing**Ph.D.** in Image Processing from JNTUH, He Guided Many UG and PG Students. He has **10 Years Teaching Experience.**