Journal of Korea Multimedia Society Vol. 19, No. 2, February 2016(pp. 155-162)
http://dx.doi.org/10.9717/kmms.2016.19.2.155

An Analysis of Effective Throughput in Distributed Wireless Scheduling

Amr Radwan†

ABSTRACT

Several distributed scheduling policies have been proposed with the objective of attaining the maximum throughput region or a guaranteed fraction throughput region. These policies consider only the theoretical throughput and do not account the lost in throughput due to the time complexity of implementing an algorithm in practice. Therefore, we propose a novel concept called effective throughput to characterize the actual throughput by taking into account the time complexity. Effective throughput can be viewed as the actual transmitted data without including the control message overhead. Numerical results demonstrate that in practical scheduling, time complexity significantly affects throughput. The performance of throughput degrades when the time complexity is high.

Key words: Wireless Scheduling; Effective Throughput; Complexity; Random Pick-Compare

1. INTRODUCTION

Scheduling algorithms for wireless network have received a great deal of attention in recent years. Much research effort is being devoted to the design of scheduling policies that can achieve the maximum attainable throughput region. In [1], Tassiulas et all provided a scheduling policy that obtains the maximum throughput region in an arbitrary wireless network. However, these policies lead to centralize operation and have exponential complexity. Such scheduling policies are inappropriate for wireless network, which requires simple and distributed computation. Thus a number of distributed scheduling policies have been studied in literature. These distributed policies addressed the scheduling base on the following ideal: (i) Maximal algorithm [2], [3] (ii) Randomized Pick and Compare algorithms [4] (iii) Random Access approach [5], [6]. In these approaches, at every transmission period, each link needs to determine whether it would transmit based on its own state and the information it obtains about the states of other nodes.

In practice, distributed algorithm requires the communication overhead of message passing to (i) collect information of neighbor links: e.g., queue length and channel states information (ii) decide link which is transmitted based on its received information. We refer the total time spent in these two parts as the time complexity. Thus, the time complexity is a key performance metric for any dynamic distributed scheduling policy. Most of these distributed policies attempted to maximize the throughput region while ignoring the impact of the time complexity. For example, in a large network, the deciding link which is transmitted may require substantial overhead, message exchange, in the term of time. More specifically, an algorithm in [7] at least required control message exchange of an order of network size to compute the schedule, henceforth denote by O(n). In that algorithm,

※ Corresponding Author : Amr Radwan, Address : (621-749) Obangdong, Gimhae, Gyeongnam, Korea, TEL : +82-55-320-3847, FAX : +82-55-322-6275, E-mail : amr_or@yahoo.com

Receipt date : Dec. 15, 2015, Approval date : Dec. 23, 2015

† Dept. of Electronics and Telecommunication Engineering, Inje University, Korea

the actual attained throughput is at least a factor of O(n) away from the optimal.

In recent work, it has been shown that the Randomized Pick and Compare algorithms (RPC) can also guarantee a fraction of capacity region [7]. By using the RPC approach, the author in [7] developed the distributed scheduling algorithms working under one-hop interference model. This algorithm is parameterized by k, and guarantees to attain a fix fraction of the throughput region while requiring constant overhead that does not grow with the network size. This policy tradeoffs between the throughput guarantees and the computation time. However, this paper refers to the throughput performance, since it does not account the resource used in communication overheard. The loss in throughput caused by the time complexity of implement an algorithm in practice is not explicitly accounted.

In particular, when we use the large fraction of time to computing schedule, the performance of the algorithm is significantly influenced because the remained portion time used for data transmission reduces. In contrast with previous research, we define the notation of effective throughput to characterize the performance of the system by taking into account the time complexity in the scheduling. Effective throughput can be viewed as the actual transmitted data without including the control message overhead. It is achieved by using the data part in time slot to transmit packet after excluding portion time slot occupied by control message overhead. This causes the effective throughput to be a function of the time complexity. Thus, an important question is how we consider effective throughput of scheduling algorithms by deducting the impact of complexity. More specifically, we consider the effective throughput of a class distributed scheduling policies RPC. In this class policy, a set of link is picked at each time slot and the weight of the chosen set of link is compared with the weight of the set of link at previous time slot. And the one with maximum weight is chosen to transmit in this time slot [4].

From all discussion above, in this paper, we derive an analytical model capturing the effects of time complexity to the effective throughput. We describe the throughput as a function of time complexity and analyze the tradeoff between the throughput guarantees and the time complexity for RPC policy. We show that in practical scheduling, time complexity does not always lead to increment of effective throughput. Simulation is conducted to demonstrate the influence of the time complexity on throughput. We show that, in practical distributed scheduling, a portion of throughput is lost due to the required overhead or time complexity.

This paper is structured as follows. In section II, we provide the notation, network model and some definitions of related problem. We analyze the effective throughput in term of time complexity in section III. The numerical analysis is provided in section IV. We conclude our work in section V.

2. SYSTEM DESCRIPTION

2.1 System Model

In this paper, we consider scheduling problem at the MAC layer of the wireless network. The wireless network consists of N nodes and L links. Time is divided into equal slot $t=\{1,2,3...\}$. Packets arriving at each link are queued. We use $Q(t)=\{Q_1(t),Q_2(t),Q_3(t)...Q_L(t)\}$ to denote the queue length of the queue at time slot t. A link can be activated only if its queue length is not empty. A scheduling algorithm is to decide the set of active links that can be transmitted at during each time slot. Define $I_l(t)\in\{0,1\}$as the indicator function indicating whether or not link l is activated at time slot t. That is, $I_l(t)=1$ if link l is activated (scheduled) at time slot t and $I_l(t)=0$ in other case. Next, we present some important definition.

Definition.1 (Network stability): Queue q is stable if $\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T}\sum_{i=1}^{L}Q_i(t)<\infty$.

Network is stable if all queues are stable.

Definition 2 (Stability Region): The Stability region of scheduling policy is the set of arrival rates $\{\lambda_i\}_{i=1,2...L}$ that stabilize the system under the policy. The union of stability region of all scheduling policies is the stability region of the system.

Definition 3 (Capacity Region): The capacity region of network, denoted by Λ, is a set of arrival rate vectors that can be stably supported by network.

Definition 4 (Throughput optimal): A scheduling policy is throughput-optimal if it can stabilize all arrival rate vectors that lie strictly inside the capacity region Λ.

Definition 5 (γ-throughput optimal): A scheduling policy is γ-throughput-optimal if it can stabilize all arrival rate vectors that lie strictly inside the capacity region Λ.

2.2 Randomized Pick and Compare Scheduling

The RPC approach was first developed for the scheduling in input queue network [8]. The key feature of the randomization approach is that it does not seek to find an optimal schedule in every slot, and hence, it can significantly reduce the computation time. In every time slot, the scheduling(γ; σ)-RPC determines the activate set I(t) in two steps:

- Step 1(**Pick**): At each time slot, it generates a new schedule $\overline{I(t)}$ such that: $\Pr\left(Q(t)\overline{I(t)}\geq\gamma Q(t)I^{*}(t)\right)>\sigma$ for constant σ>0 and γ>0 where $I^{*}(t)=\arg max(I(t)Q(t))$.
- Step 2(**Compare**): Determine the schedule at time slot t by comparing the previous scheduler I(t-1) and the new scheduler $\overline{I(t)}$: $I(t)=\arg max\left(I(t-1)Q(t),\overline{I(t)}Q(t)\right)$.

In this scheduler, a set of link is picked at each time slot with probabilistic guarantee of -optimal policy and the weight of the chosen set of link is compared with the weight of the set of link at previous time slot. The one with maximum weight is selected to transmit in this time slot. The guaranteed throughput of the (γ;σ)-RPC policy is derived by the following proposition.

Proposition 1:[7] Given any $\gamma\in(0,1)$, suppose that an algorithm has a probability at least $\sigma>0$ of generating a independent set $\overline{I(t)}$of links with weight at least γ times the weight of the optimal. Then, capacity $\gamma\Lambda$ can be achieved by switching links to the new independent set when its weight is larger than the previous one (otherwise, previous set of links will be kept for scheduling).

Thus, as shown as in above proposition, the (γ; σ)-RPC policy is the γ-throughput optimal. We say that the throughput of the system under (γ; σ)-RPC policy is γ. Note that, the throughput given by proposition 1 does not consider the time complexity. They assumed that users use the whole time slot to transmit data.

Implementation of (γ;σ)-RPC policy uses time slot structure in Fig. 1. As shown in Fig. 1, a time slot consists of two parts including a control part and a data part. The control part consists of multiple mini slots in which each node collects information of its neighbour and decides link which is transmitted based on information acquired. The data is finally transmitted base on the above decision.

The partitioning of the time slot explicitly captures the time complexity of the algorithms. Let $\Theta(t)$ represent the number of mini time slots in control part at time slot t. Then the time complexity of the scheduling algorithms can be measured by the number of mini time slots which is used

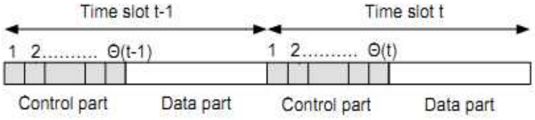

Fig. 1. Time slot structure.

to compute the schedule: $\overline{\Theta} = E[\Theta(t)]$.

3. EFFECTIVE THROUGHPUT ANALYSIS

In this section, we derive a mathematical framework for effective throughput of the system by considering the time complexity. First, we present the concept of the effective throughput. Effective throughput is achieved by using the data part in time slot to transmit packet. As show above, in the case we use completely time slot to transmit data, the throughput guaranteed by the $(\gamma;\sigma)$-RPC policy is γ. Otherwise, if we just use the data part which was described in figure 1 to transmit data, then the effective throughput can be computed by multiplying the throughput above by the function of the time complexity $\overline{\Theta}$. Hence, the effective throughput under $(\gamma;\sigma)$-RPC policy is defined as follows

$$\hat{\gamma} = \frac{T-\overline{\Theta}\Delta t}{T}\gamma \tag{1}$$

where T;Δt is the length of time slot and mini slot. The definition of effective throughput considers the lost throughput due to the influence of the time complexity. Thus, to obtain the effective throughput, we must determine the number of mini time slot or the time complexity $\overline{\Theta}$.

Second, in the remains of this section, we evaluate the complexity in the term of the parameters of $(\gamma;\sigma)$-RPC policy. In fact, the complexity algorithms related with the probability of generating a set $\overline{I(t)}$ that satisfies Step 1(Pick) in Lemma 1.

Lemma 1: The probability of generating an independent set $\overline{I(t)}$ of links with weight at least γ times the weight of the optimal is increased, it leads to increase the time complexity of the $(\gamma;\sigma)$-RPC scheduling.

Proof: First we introduce the definition of m-stretching

Definition (m-stretching): Consider a sequence of random time slot $t_0,t_1,t_2,\ldots$ such that $E[t_i - t_{i-1}]$ original scheduling algorithms Π. The m-stretching Π(m) can be obtained from Π by using the same algorithms on every m slots. And between the m slots, the schedule remains the same.

Because the m-stretching schedule computes the scheduling on every m time slots, the time complexity of this algorithm can be reduced by a fraction $\frac{1}{m}$. As shown in [7], the $(\gamma;\sigma)$-RPC algo rithms is the version of the m-stretching, where m = $\frac{1}{\sigma}$. Therefore, the time complexity of the RPC schedule naturally becomes Θ_{MW} where Θ_{MW} is the number of mini time slots of the throughput optimal scheduling.

Then we show that the increasing of the probability of generating an independent set $\bar{I}$(t) of links with weight at least γ times the weight of the optimal leads to the increasing of the σ value. Denote by $\{t_i\}_{i=0,1\ldots}$ be the sequence of time slots where $Q(t)\overline{I(t)} \geq \gamma Q(t) I^*(t)$. Let $\Delta t_i = t_{i+1} - t_i$. Notice that when the probability of generating an independent set $\bar{I}$(t) is decreased, the value of Δt_i is increased. Thus, the result follow, from the fact that E $[\Delta t_i]$ = $\frac{1}{\sigma}$ [7]. △

From Lemma 1, the time complexity can be formulated as the function of the probability of generating a set $\overline{I(t)}$. Thus, the complexity of the$(\gamma;\sigma)$-RPC can be expressed as:

$$\overline{\Theta_{RPC}} = g(\gamma)\Theta_{MW}.$$

where $g(\gamma) = \Pr\left(Q(t)\overline{I(t)} \geq \gamma Q(t) I^*(t)\right)$.

We will introduce the method to estimate the complexity function $g(\gamma)$. Now, some useful probabilistic lemmas:

Lemma 2: Let $X_1,X_2,\ldots X_k$ be the independent random variables with exponential distribution and parameters $e_1,e_2\ldots.e_k$. Then $X_{\min} = \min X_i$ has an exponential distribution with parameter $\sum_{k=1}^{K} e_k$.

Lemma 3: Consider a sequence of independent exponential random variables $Z_1, Z_2, \ldots Z_N$ with parameter e. Now let $M_N = \frac{1}{N}\sum_{i=1}^{N} Z_i$. Then for $z > 1$, larger deviation theory [10] states that the following limit exits:

$$P(M_N < z) \approx \exp(-NI(z)).$$

The rate function I(z) can be given by Legendre transform:

$$I(z) = (\theta z - \lambda(\theta)) = ez - 1 - \log(ez)$$

where $\lambda(\theta) = \log E(\exp(\theta z))$.

The Lemma 2 is a well-known lemma about the exponential distribution. Lemma 3 follows from the Central Limit Theorem about the large deviation theory for exponential distribution [10]. Given the independent set I(t), and vector queue lengths Q(t).At each link l∈ I(t), we make an independent exponential distribution with parameter Q_l, then we compute the minimum of these value. Replace it for B times to get the minimum $Z_i^*, i = 1 \ldots B$. Thus Z_i^* has an exponential distribution with parameter $w = \sum Q_l(t)$.

Now set $S_B = \frac{1}{B}\sum_{i=1}^{B} Z_i^*$ and declare $M_B = \frac{1}{S_B}$.

Clearly, for B large enough: $M_B \xrightarrow{B \to \infty} \sum Q_l(t) I_l(t)$.

From Lemma 2, we obtain estimation of complexity function

$$\begin{aligned} g(\gamma) &= \Pr\left(Q(t)\bar{I}(t) \geq \gamma Q(t) I^*(t)\right) = \Pr\left(M_B > \gamma W^*\right) \\ &= \Pr\left(S_B W^* < \frac{1}{\gamma}\right) = \exp\left(-B\left(\frac{w}{W^*}\frac{1}{\gamma} - 1 - \log\left(\frac{w}{W^*}\frac{1}{\gamma}\right)\right)\right) \end{aligned}$$

Finally, we conclude that

$$\overline{\Theta_{RPC}} = g\left(\frac{1}{\gamma}\right)\Theta_{MW} \tag{2}$$

where $g(x) = \exp\left(-B\left(\frac{w}{W^*}x - 1 - \log\left(\frac{w}{W^*}x\right)\right)\right)$.

The equation above expresses complexity of the $(\gamma;\sigma)$-RPC policy or the number of mini time slots in the control signaling part. Then from (1), (2), the effective throughput can be formulated as follow:

$$\hat{\gamma} = \frac{T - g\left(\frac{1}{\gamma}\right)\Theta_{MW}\Delta t}{T}\gamma = \left[1 - g\left(\frac{1}{\gamma}\right)\Theta_{MW}\tau\right]\gamma$$

where $\tau = \frac{\Delta t}{T}$.

4. PERFORMANCE EVALUATION

The primary purpose of this analysis is to observe the relationship between effective throughput and the complexity of this algorithm.

We generate a random topology with 36 nodes in an 1 × 1 rectangular space and connect a link between two nodes if their distance is less than 0.3, which result in 128 directed links. We use the 1-hop interference model and assume that node i has the information of its neighborhood, that is, node i is supposed to know the queue length Q_i for all $i \in N(j)$, where N(j) is the set of nodes connected to node k by single hop. Since RPC scheduling requires this local information, the comparison between them remains valid.

A slot is divided into two periods; one for control message exchange and the other for actual data transmissions. The period for message exchange consists of smaller mini-slots. Two messages can be exchanged in a single mini-slot. In this simulation, we set the $\tau = 0.002$ compared to the actual data transmission time.

First, we consider the performance of the effective throughput of the random pick and compare scheme. More specifically, we analyze the performance of effective throughput of two specifics policies in class of the random pick and compare schemes. The two policies are call *distributed link scheduling with constant overheard* (DCO) [10] and *information (1) policy for approximating the maximum throughput region* (Inf(1)) [11]. Two classes of the algorithms are parameterized by k.

In the following, we will describe these two policies in details.

The first class, DCO, requires that the length of the control signalling part is 4k + 2 round-trip time, where one round-trip time is the amount of time required for a node to make a very basic two-way handshake with a neighbouring node. It is then guaranteed to achieve a fraction $\frac{k}{k+2}$ of the throughput for any network. Then, the effective throughput can be computed by multiplying the fraction of throughput above by the function of time complexity f(k). The function f(k) is the proportion of time for data transmission to time in one slot:

$$f(k) = \frac{T-\Delta t(4k+1)}{T}.$$

Hence, we have:

$$\widehat{\gamma_{DCO}} = \frac{k}{k+2}f(k) = \frac{k}{k+2}(1-(4k+2)\Delta t) \ .$$

Similarly, the remain class policy, inf(1) policy for approximating the maximum throughput region, need to compute the new scheduling and can guarantee a fraction $\frac{k-1}{k}$ of the throughput. Then we can obtain the effective throughput as follow:

$$\widehat{\gamma_{inf(1)}} = \frac{k-1}{k}(1-O(k)\Delta t).$$

We compare the actual throughput and the analytical throughput of two policies and observe that, time complexity will affect significantly to the throughput. But, when complexity increases greater than a threshold value, the actual throughput will decrease and the analytical throughput continues increase. This is because a lager value of the complexity requires a longer length of the signalling control part. Thus, the remaining portion that has been used for data transmission will be reduced and affect to the performance of the system. In the case of the portion of time used for scheduling is so lager, the actual throughput of the system will go to zero(in the case of the DCO policy in Fig. 2). This result is similar to our conclusion in general case.

The Fg. 3 shows the decrease of throughput in the system as the value of mini slot time is increased. Because the increase of time use for control signalling, it will lead to the decline of time use for transmission data. Thus, the throughput is decreased.

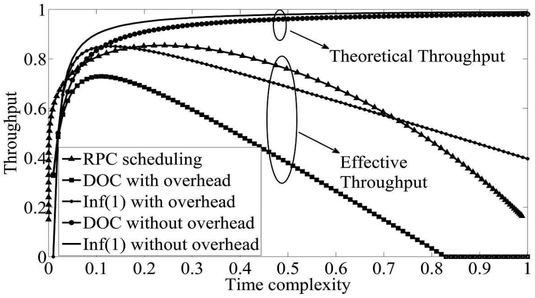

Fig. 2. The performance of effective throughpu.

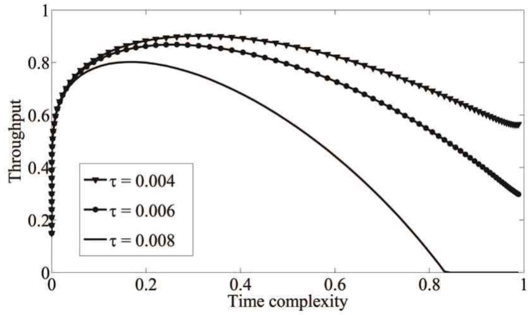

Fig. 3. The effective throughput under variance of mini time slot.

Next, we compare the performance of DCO and Inf(1) policies in terms of the throughput region. Because the throughput region of control policy is defined as the arrival rate while the sum of average queue length is finite, the policy with the smaller sum attains the better throughput region. Fig. 4 shows the sum of the average queue length over all node in network. We can see that in these policies, the actual attained throughput (effective throughput) is far away from the theoretical throughput. The gap between the effective throughput and the theoretical throughput presents the loss of the throughput while taking account the message overhead into the throughput.

5. CONCLUSION

We have analyzed an effective throughput for

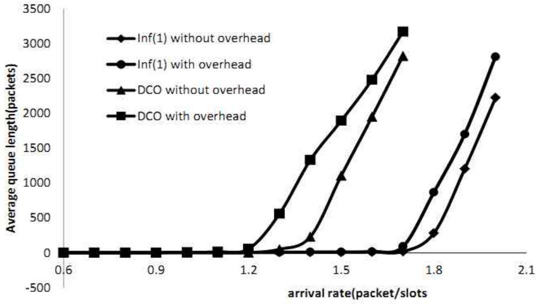

Fig. 4. Throughput region of policies.

a specific class of distributed scheduling algorithms. We take the overhead into account of effective throughput. By using the larger deviation technique, we develop a function of the effective throughput in the term of time complexity.

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. 1936-1948, 1992.

[2] K. P.Chaporkar, K. Sarkar and S. Sarkar, "Throughput Guarantees through Maximal Scheduling in Multihop Wireless Networks," *Proceedings of 43rd Annual Allerton Conference on Communication, Control and Computing*, pp. 28-30, 2005.

[3] C. Joo, X. Lin, and N. Shroff, "Understanding the Capacity Region of the Greedy Maximal Scheduling Algorithm in Multi-Hop Wireless Networks," *Proceedings of The 27th Conference on Computer Communications,* pp. 1103-1111, 2008.

[4] A. Eryilmaz, O. Asuman, and E. Modiano, "Polynomial Complexity Algorithms for Full Utilization of Multi-Hop Wireless Networks," *Proceedings of The 26th IEEE International Conference on Computer Communications*, pp. 499-507, 2007.

[5] C. Joo and N. Shroff, "Performance of Random Access Scheduling Schemes in Multi-Hop Wireless Networks," *IEEE/ACM Transactions on Networking,* Vol. 17, No. 5, pp. 1481-1493, 2009.

[6] A. Proutiere, Y. Yi, and M. Chiang, "Throughput of Random Access Without Message Passing," *Proceedings of The 2nd Annual Conference on Information Sciencesand Systems,* pp. 509-514, 2008.

[7] S. Sanghavi, L. Bui, and R. Srikant, "Distributed Link Schedul-ing with Constant Overhead," *Proceedings of the 2007 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pp. 313-324, 2007.

[8] L. Tassiulas, "Linear Complexity Algorithms for Maximum Throughput in Radio Networks and Input Queued Switches," *Proceedings of The Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies.* Vol. 2, pp. 533-539, 1998.

[9] Y. Yi, A. Prouti` ere, and M. Chiang, "Complexity in Wireless Scheduling: Impact and Tradeoffs," *Proceedings of the 9th ACM International Symposium on Mobile Ad hoc Networking and Computing,* pp. 33 - 42, 2008.

[10] O. Z. Amir Dembo, Large Deviations Techniques and Applications, *2nd ed.* Springer Verlag Berlin Heidelberg, 1998.(Berlin Heidelberg)

[11] S. Sarkar and S. Ray, "Arbitrary Throughput Versus Complexity Tradeoffs in Wireless Networks Using Graph Partitioning," *IEEE Transactions on Automatic Control,* Vol. 53, No. 10, pp. 2307-2323, 2008.

Amr Radwan

received his B.S. and M.S. degrees in Mathematics from Sohag University, Sohag, Egypt, in 1999 and 2005. He received the Ph.D. degree in Mathematics from Humboldt University zu Berlin, Germany in 2012. From Aug. 2012-Sep.2014, he worked as a lecturer at Mathematics Department, Faculty of Science, Sohag University, Egypt. Dr. Radwan was awarded an NRF Postdoctoral Research Fellowship (2014, Oct.-2015, Feb.), and joined Prof. Won-Joo Hwang's research group in the Department of Information and Communications Engineering at Inje University, where he worked problems related to wireless networks. Since Mar. 2015, he has been an assistant professor at Department of Information and Communications Engineering, Inje University, Gyeongnam, Korea. His research interests are in the area of nonlinear optimization, optimal control problem, automatic differentiation, evolutionary computing and wireless networks.