Distributed Scheduling Scheme for Optimal Performance in Wireless Networks

Nguyen H. Tran and Choong Seon Hong
Department of Computer Engineering, Kyung Hee University, 446-701, Republic of Korea
Email: {nguyenth, cshong}@khu.ac.kr

Abstract-We propose a randomized distributed scheduling algorithm which can achieve the optimal throughput under the general interference model. The proposed algorithm is analyzed to show an attractive performance in that it can return a maximal schedule with high probability and has a low time-complexity. We also provide the simulation results to validate performance analysis of our algorithm.

I. Introduction

In this paper, we consider the problem of how to achieve the maximal throughput in wireless networks with distributed scheduling algorithms under the general K-hop interference constraint. We choose the Pick and Compare approach [4] for our proposed algorithm. For Pick algorithm, we design a randomized distributed scheduling algorithm that can return a maximal matching with high probability and low time-complexity. Moreover, while some of the previous work require to exchange the queue length information to build transmission schedules [2], [5], ours does not.

II. System Models and Definition

The wireless network is represented by a graph $\boldsymbol{G}(\mathcal{V}, \mathcal{L})$, where $\mathcal{V}$ and $\mathcal{L}$ are the sets of V nodes and L links, respectively. We denote $A_l(t)$ the number of arrivals over link l and assume that the arrival process is i.i.d with mean $\lambda_l = \mathbb{E}[A_l(t)]$. We also define a mean arrival rate vector $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_L)$ and the *capacity region* is represented as a set $\{\boldsymbol{\lambda}\}$.

Definition 1 (Throughput-Optimal Scheduling Algorithm): A scheduling algorithm is defined to *achieve the capacity region* if it supports any $\boldsymbol{\lambda}$ in the capacity region while keeping the queues of all links finite. A *throughput-optimal* scheduling algorithm is defined as the scheduling policy that achieves the largest capacity region.

We denote network resources as a finite set $\mathcal{S}$ of the feasible schedule vectors. Here a feasible schedule means a set of links in which no two links interfered with each other. A resource allocation scheme then aims at choosing in each time-slot a schedule vector $\mathbf{S}(t) = (S_1(t), S_2(t), \ldots, S_L(t)) \in \mathcal{S}$, where $S_l(t)$ is the rate (in packets/slot) of link l.

Definition 2 (Pick-and-Compare): The algorithm first picks a random schedule $\mathbf{S}'(t)$ satisfying $\mathbf{P1}$ (i.e., Pick property) and then the actual schedule $\mathbf{S}(t)$ is updated as in $\mathbf{P2}$ (i.e., Compare property).

- $\mathbf{P1} : \mathbb{P}[\mathbf{S}'(t) = \mathbf{S}^{\star}(t)] \geq \delta$, for some $0 < \delta < 1$.
- $\mathbf{P2} : \mathbf{S}(t) = \arg\max_{\mathbf{S}=\{\mathbf{S}(t-1), \mathbf{S}'(t)\}} W(\mathbf{S})$

where $Q_l(t)$ is the queue length of link l at time t, $W(\mathbf{S}) = \sum_{l \in \mathcal{L}} S_l(t) Q_l(t)$ is the weight of the schedule $\mathbf{S}(t)$ and $\mathbf{S}^{\star}(t) = \arg\max_{\mathbf{S}(t) \in \mathcal{S}} W(\mathbf{S})$ is the *optimal schedule.* The algorithm only updates its schedule if the weight of new schedule exceeds that of the old schedule. And the essence of this algorithm, the throughput-optimal characteristic, is captured as below:

Theorem 1: (Eryilmaz) Any scheduling scheme satisfying $\mathbf{P1}$ and $\mathbf{P2}$ would achieve the largest capacity region.

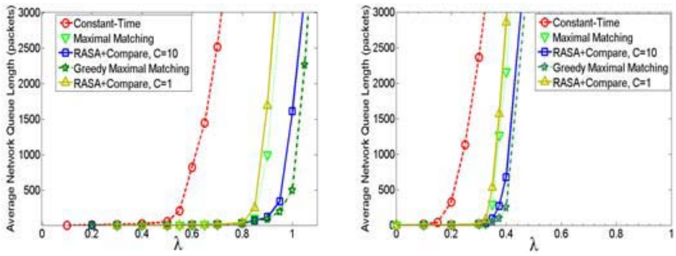
III. Pick Algorithm Description and Analysis

Algorithm 1 Randomized Scheduling Algorithm (RASA)

At each time-slot, each node $v \in \mathcal{V}$ does
if v senses ACK(v', u') **then**
    $disabled(v) = 1$
if $disabled(v) \neq 1$ **then**
    for $s = 1$ to $(\log V)$ **do**
        $\mathbf{S}_s(t) := \mathbf{S}_{s-1}(t)$
        for $i = 1$ to $(C\, e^{4{d_n}^{K-1}} \log V)$ **do**
            v chooses an arbitrary $u \in N(v)$ and sends
            REQ with probability p
            if v senses other REQ's or senses COL **then**
                v sends COL
            if u senses REQ of v and senses no COL **then**
                u sends RPL
            if u senses other's REQ's **then**
                u sends COL
            if v senses RPL of u and senses no COL **then**
                v sends ACK(v, u)
            if u senses ACK(v, u) successfully **then**
                $\mathbf{S}_s(t) := \mathbf{S}_s(t) \cup (v, u)$

Here, we denote the 1-hop neighbors of node v by $N(v)$ and we say that a node *senses* a control packet if it can decode a control packet or receive a non-decodable collision packet. When a node want to request a matching, it sends a *request* (REQ) packet to a chosen neighbor node with a probability p. If it senses another ongoing transmission or a *collision* (COL) packet from that transmission, it sends a COL in the next round. If the receiver can decode its REQ successfully and sense no COL packet, it means that no other transmissions are within K hops of the transmitter side at that time. Subsequently, the receiver responds with a *reply* (RPL) packet. While sending its RPL, if the receiver detects

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by MEST (No. 2010-0027645). Dr. CS Hong is corresponding author.



(a) $K = 1$ (b) $K = 2$

Fig. 1: Capacity region of scheduling algorithms

an ongoing transmission, it will send a COL packet in the next round. Thus, no COL packets being sent from receiver guarantees that no other transmissions are within K hops of the receiver side. Next, if no COL packets is sensed from both side, the transmitter broadcasts the *acknowledge* packets ACK(v, u) to announce that their matching is successful. Every node within K hops neighborhood of transmitter v and receiver u after realizing the ID of this link through sensing this ACK(v, u) packet will be *disabled*, preventing them from requesting their matchings in the subsequent rounds. And after receiver senses ACK(v, u), this link is ready to be active in the data transmission phase (i.e., knowing that it belongs to $\mathbf{S}(t)$).

Theorem 2: If $C > 6$, then RASA has $\Theta\left(e^{4{d_n}^{K-1}} \log^2 V\right)$ time-complexity and can return a maximal schedule with high probability no less than $\delta = 1 - \frac{1}{V^{C'}}$, with $C' > 1$.

Proof: Link (v, u) is matched successfully if during the whole operation of this algorithm, provided that nodes v and u are exchanging control packets, all nodes in $(u_1|u_1 \in N(v)) \cup (u_2|u_2 \in N(u_1)) \ldots \cup (u_K|u_K \in N(u_{K-1}))$ and all nodes in $(v_1|v_1 \in N(u)\backslash v) \cup (v_2|v_2 \in N(v_1)) \ldots \cup (v_K|v_K \in N(v_{K-1}))$ keep silent. We first consider the transmitter side, where node v sends all of its control packets successfully (i.e., without sending COL packets) with a probability $\mathbb{P}_{v\to u}$ satisfying:

$$\mathbb{P}_{v\to u} = \frac{e^{-1} \cdot e^{-|N(v)|} \ldots e^{-|N(v)||N(u_1)|\cdots|N(u_{K-2})|}}{|N(v)| + 1} \quad (1)$$

$$\geq \frac{e^{-2{d_n}^{K-1}}}{|N(v)| + 1} \text{ when } d_n > 1. \quad (2)$$

The probability that node v is matched is

$$\mathbb{P}[v\, is\, matched] = \sum_{u \in N(v)} \mathbb{P}_{(v,u)} \geq \frac{e^{-4{d_n}^{K-1}}}{2}. \quad (3)$$

So, probability that node v is not matched during an iteration i is $\mathbb{P}_i[v\, is\, not\, matched] \leq (1 - \frac{e^{-4{d_n}^{K-1}}}{2})$. Thus, for one step, we have:

$$\mathbb{P}_s[v\, is\, not\, matched] \leq \left(1 - \frac{e^{-4{d_n}^{K-1}}}{2}\right)^{C\, e^{4{d_n}^{K-1}} \log V} \quad (4)$$

$$\leq e^{-\frac{e^{-4{d_n}^{K-1}}}{2} \times C\, e^{4{d_n}^{K-1}} \log V} \leq e^{-\frac{C \log V}{2}} = \frac{1}{V^{\frac{C}{2}}}. \quad (5)$$

Clearly, we have exactly $|M|$ nodes of any maximal matching $M \in \mathcal{S}$ that are matched at the end of step s with a probability:

$$\mathbb{P}_s[M] \geq (1 - \frac{1}{V^{\frac{C}{2}}})^{|M|} \geq 1 - \frac{|M|}{V^{\frac{C}{2}}} \geq 1 - \frac{1}{V^{\frac{C}{2}-1}}. \quad (6)$$

Since $\log V$ steps are satisfied for the requirement of a maximal matching running time, this algorithm can return a maximal schedule M with probability at least:

$$\mathbb{P}[M] \geq (1 - \frac{1}{V^{\frac{C}{2}-1}})^{\log V} \geq 1 - \frac{\log V}{V^{\frac{C}{2}-1}} \geq 1 - \frac{1}{V^{\frac{C}{2}-2}}. \quad (7)$$

Letting $C' = C/2 - 2$, we have the result.

For time-complexity, we can see that the algorithm has at most $\Theta(\log V)$ steps. Each step involves $\Theta(e^{4{d_n}^{K-1}} \log V)$ rounds. So RASA takes $\Theta(e^{4{d_n}^{K-1}} \log^2 V)$ rounds of local message exchange and computation. We complete our proof here.

IV. Simulation Results

In this section, first we evaluate the throughput performance between Constant-Time (CT) [5], Maximal Matching (MM), Greedy Maximal Matching (GMM) [5] algorithms and RASA with Compare operation.

We use a 5×5 grid topology to evaluate the throughput performance. Every link has a capacity 1 packet/slot. Fig. 1a shows that with $C = 10$, RASA has almost the same performance as GMM, which is close to the optimal and dominates other algorithms. This means that in this case RASA can work at load very close to the threshold $\lambda = 1$ while in case of $\lambda \geq 1$, the queues become unstable. However, when $C = 1$, RASA and MM are almost the same performance which is more than 80% capacity region. Therefore, the value of C should be chosen carefully to achieve the optimal performance, which supports our analysis result. With more than 80% and 60% capacity empirically obtaining, the performance of MM and CT coincide with those ones in the literature [5]. In case of 2-hop interference model ($K = 2$), the behaviors of all algorithms are similar to the first case except the capacity region approximately corresponds to $\lambda < 0.4$.

V. Conclusion

In this work, we propose a randomized distributed scheduling algorithms which is proved that it can be integrated into the Pick-and-Compare algorithm for throughput-optimal achieving. Especially, this algorithm is designed as a maximal schedule to work for the general K-hop interference model.

References

[1] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Transaction on Automatic Control*, 37(12):1936-1948, December 1992.

[2] G. Sharma, C. Joo, and N. B. Shroff. Distributed Scheduling Schemes for Throughput Guarantees in Wireless Networks. *44rd Annual Allerton Conf. on Communications, Control, and Computing*, 2006.

[3] G. Sharma, R. R. Mazumdar, and N. B. Shroff, "On the complexity of scheduling in multihop wireless systems," *MOBICOM 2006*, 2006.

[4] A. Eryilmaz, A. Ozdaglar, and E. Modiano, "Polynomial Complexity Algorithms for Full Utilization of Multi-hop Wireless Networks," *IEEE INFOCOM*, 2007.

[5] X. Lin and S. Rasool, "Constant-time distributed scheduling policies for ad hoc wireless networks," *IEEE CDC*, 2006.