On The Throughput-Optimal Distributed Scheduling Schemes with Delay Analysis in Multi-hop Wireless Networks

Nguyen H. Tran, Choong Seon Hong
Department of Computer Engineering, Kyung Hee University, 449-701, Republic of Korea
Email: {nguyenth, cshong}@khu.ac.kr

***Abstract*—With K-hop interference model, especially when $K \geq 2$, the throughput-optimal centralized scheduler needs to solve a NP-Hard problem. It leads to the desire of a distributed, low-complexity but throughput-optimal scheduling algorithm. We generalize a randomized scheduling framework for a K-hop interference model and develop two randomized distributed scheduling algorithms which can be integrated into this framework. The delay performance of our scheme is also characterized.**

***Index Terms*—Wireless Networks, Scheduling Algorithms, Optimal Throughput, Resource Allocation.**

I. Introduction

The seminal paper of Tassiulas and Ephemides [1] on Max Weight scheduling specified that their throughput-optimal scheduling algorithm can stabilize the networks provided that the set of user arrival rates fall within the so-called capacity region. However, in general K-hop interference model, the authors in [3] showed that the centralized optimal scheduling is polynomial time solvable under 1-hop interference model, but it is NP-Hard and Non-Approximable in case of $K \geq 2$. This attracts researchers' attention to the distributed scheduling policies with low-complexity computation.

Generally, distributed scheduling schemes can be classified into three categories. One of them is Greedy Maximal scheduling algorithms [3], which require only local message exchange. Another distributed approach is the Constant-Time algorithms [5] in which the number of computation and local message exchange rounds at every slot is a constant. While both of these approaches guarantee only a fraction of the capacity region, most of Pick and Compare algorithms [6] can provide the maximal throughput. But all of above-mentioned algorithms only work with restrictive cases of 1-hop and 2-hop interference model.

In this paper, we consider the problem of how to achieve the maximum throughput in wireless networks with distributed scheduling algorithms under the K-hop interference constraint. We choose the Pick-and-Compare approach as the randomized scheduling framework for our proposed algorithms. In Section III, we generalize the throughput-optimal distributed scheduling schemes into a randomized scheduling framework. In Section IV, for Pick operation, we propose two randomized distributed scheduling algorithms that can be integrated into the randomized scheduling framework. Both of them can work for the K-hop interference model. We show that the probabilities of picking them are guaranteed. We also investigate the delay characteristic of the throughput-optimal randomized scheduling framework corresponding to two proposed distributed scheduling algorithms.

This work was supported by the Korea Science and Engineering Foundation(KOSEF) grant funded by the Korea government(MEST) (No. R01-2008-000-20801-0). Dr. CS Hong is the corresponding author.

II. System Model

The multi-hop wireless network is represented by a graph $\boldsymbol{G}(\mathcal{N}, \mathcal{L})$, which has the node set $\mathcal{N}$ and the link set $\mathcal{L}$. Denote the number of links and nodes by L and N. There is a set of sessions $\mathcal{S}$ in the network, which represents for the network traffic. Sessions (i.e., flows) are identified by a set of source and destination nodes. Denote S as the number of sessions. We assume that each node $n \in \mathcal{N}$ maintains per-session queues with finite buffer. We assume that the system is time-slotted and denote $A_n^s(t)$ as the exogenous traffic (in packets/slot) generated by session s during the time-slot t at node n. The arrival process $A_n^s(t)$ is assumed i.i.d with mean rate $\lambda_n^s = \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} A_n^s(t)$. Maximum arrival rate is assumed to be bounded by a constant A_{max} and we denote the arrival vector by $\boldsymbol{\lambda} = (\lambda_n^s)_{n\in\mathcal{N}, s\in\mathcal{S}}$.

A resource allocation scheme then aims at choosing in each time-slot a rate schedule $\mathbf{R}(t) = (R_1(t), ..., R_L(t)) \in \mathcal{R}$, where $R_l(t)$ is the rate (in packets/slot) of link l. The rate allocated to session s over link l at time t is introduced by the notation $R_l^s(t)$, and in a given slot, $R_l^s(t) \in \{0, 1\}$ is 1 if link l serves a packet of session s, and 0 otherwise. We denote a constant $\Phi = A_{max} + 2\Delta$, where Δ is the maximum node degree of the wireless network. We denote $Q_n^s(t)$ the queue length of session s at node n at time t. We introduce $\mathcal{I}(n)$ and $\mathcal{O}(n)$ as the notation for the set of incoming and outgoing links of node n, respectively. Then, the queueing dynamics are described by the following recursion:

$$Q_n^s(t+1) = Q_n^s(t) - \sum_{l\in\mathcal{O}(n)} R_l^s(t) + \sum_{l\in\mathcal{I}(n)} R_l^s(t) + A_n^s(t). \quad (1)$$

Next, we introduce the notion of network stability and capacity region, which have already been well-characterized in the literature [1].

Definition 1 (Stability): The queueing constrained network system is stable, if

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[\sum_{s\in\mathcal{S},n\in\mathcal{N}} Q_n^s(t)\right] < \infty. \quad (2)$$

Definition 2 (Capacity Region): The capacity region $\mathbf{\Lambda}$ is the closure of the set of all arrival rate vectors $(\lambda_n^s)_{n\in\mathcal{N},s\in\mathcal{S}}$ for which there exists a resource allocation scheme that can stabilize the network.

III. Throughput and Delay Performance of Randomized Scheduling Framework

A. Randomized Scheduling Framework

In this section, we describe the general randomized scheduling framework. This randomized scheduling strategy was presented first in the seminal work [2]. We will extend this framework to K-hop interference model and apply it to multi-hop wireless networks.

Definition 3 (Randomized Scheduling Framework): Any algorithm of the randomized scheduling framework aiming at choosing a feasible rate schedule $\mathbf{R}(t)$ must meet two following constraints in its operations. First, it generates a new random schedule $\mathbf{N}(t) \in \mathcal{R}$ satisfying the first constraint:

- $\mathbf{C1}$: $\mathbb{P}[\mathbf{N}(t) = \mathbf{R}^\star(t)] \geq \delta$, for some $0 < \delta < 1$ and for all t.

And then, the real schedule at time-slot t, $\mathbf{R}(t)$, is decided by the second constraint:

- $\mathbf{C2}$: $\mathbb{P}\left[W_{\mathbf{R}(t)}(t) \geq \max\{W_{\mathbf{R}(t-1)}(t), (1-\gamma)W_{\mathbf{N}(t)}(t)\}\right] \geq 1-\phi.$

In what follows, for an arbitrary resource allocation scheme, the weight of the rate schedule $\mathbf{R}(t)$ at time-slot t is denoted by $W_{\mathbf{R}(t)}(t)$. Let $tx(l)$ and $rx(l)$ be the transmitting and receiving nodes respectively of link l, we have:

$$W_{\mathbf{R}(t)}(t) = \sum_{l\in\mathcal{L}} R_l(t) \max_{s\in l} |Q_{tx(l)}^s(t) - Q_{rx(l)}^s(t)|.$$

And the *optimal rate schedule vector* at time-slot t is defined as:

$$\mathbf{R}^\star(t) = \arg\max\nolimits_{\mathbf{R}(t)\in\mathcal{R}} W_{\mathbf{R}(t)}(t). \quad (3)$$

The parameter δ in $\mathbf{C1}$ allows us to design an algorithm in a low-complexity manner, which simplifies substantially the heavy computation of the optimal schedule at every time-slot. In $\mathbf{C2}$, while γ represents the error probability in weight computation of the new randomly generated schedule $\mathbf{N}(t)$, ϕ draws inaccurate estimates of the difference between the weights of previous schedule $\mathbf{R}(t-1)$ and new schedule $\mathbf{N}(t)$. Before presenting the throughput and delay performance of the framework, we introduce an important concept which is used frequently in later sections.

Definition 4 (Network Load Parameter):

$$\theta_{max} = \sup\{\theta \,|\, \boldsymbol{\lambda} \in (1-\theta)\mathbf{\Lambda}\}.$$

Here θ_{max} reflects how heavily the system is loaded (e.g., when $\theta_{max} = 0$, the system is fully loaded). In the following theorem, we investigate the trade-off between parameters δ, γ, ϕ and the obtained throughput. We assume that the mean arrival rate vector $\boldsymbol{\lambda}$ is fixed over time and lies in the interior of the $(1-\theta)\mathbf{\Lambda}$.

Theorem 1: ([6]) Any algorithm of randomized scheduling framework can stabilize the network system if the mean arrival rate vector $\boldsymbol{\lambda} \in (1-\theta)\mathbf{\Lambda}$ falls within the region $(1-\gamma-2\sqrt{\frac{\phi}{\delta}})\mathbf{\Lambda}$.

Next, we discuss the delay performance of the framework.

Theorem 2: If any algorithms of randomized scheduling framework can achieve throughput-optimal performance with $\delta > 0,\ \gamma = \phi = 0$, and $\boldsymbol{\lambda} \in (1-\theta)\mathbf{\Lambda}$, we have:

$$\limsup_{\tau\to\infty} \frac{1}{\tau}\sum_{t=0}^{\tau-1}\sum_{s,n} Q_n^s(t) \leq \frac{SN^2\Phi}{\theta_{max}}\left(\frac{2}{\delta}+\frac{\Phi}{2}\right). \quad (4)$$

Proof: The proof is provided in full version if demanded. ■

IV. Algorithm Description and Analysis

A. Distributed Computation Model

Feasible allocations are supposed to happen in consecutive scheduling time units. We define the *time-complexity* of an algorithm as the number of *rounds* required by the algorithm for computing a schedule.

B. Randomized Feasible Allocation Algorithm

Algorithm 1 Randomized Feasible Allocation Algorithm

```
At each time-slot, each node n ∈ 𝒩 does
if n ∈ N_K(n') ∪ N_K(m') senses ACK(n', m'), ∀n', m' ∈ 𝒩 then
    disabled(n) = 1
if disabled(n) ≠ 1 then
    n chooses an arbitrary m ∈ N(n) and sends
    RTS with probability p
if n senses other RTS's or senses COL then
    n sends COL
if m senses n's RTS and senses no COL then
    m sends CTS
if m senses other's RTS's when sending its CTS then
    m sends COL
if n senses RTS from m and senses no COL then
    n sends ACK(n, m)
if m senses ACK(n, m) successfully then
    R(t) := R(t) ∪ (n, m)
```

We first propose an algorithm named Randomized Feasible Allocation Algorithm aiming at generating a feasible allocation at the end of Pick operation. Here, we denote the 1-hop and K-hop neighbors of node n by $N(n)$ and $N_K(n)$ respectively. We say that a node *senses* a control packet if it can decode a control packet or receive a non-decodable collision packet . At the beginning, a node that wishes to request a matching, sends a *ready-to-send* (RTS) packet to a chosen neighbor node. 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 RTS successful 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 *clear-to-send* (CTS) packet. While sending its CTS, if the receiver detects an ongoing transmission, it will send a COL packet in the next round. Thus, no COL packets being sent by receiver guarantees that no other transmissions are within K hops of the receiver side. Next, if no COL packet is sensed from both sides, the transmitter broadcasts the *acknowledge* packet (i.e., ACK(n, m)) to announce that their matching is successful. Every node within K hops neighborhood of transmitter n and receiver m after realizing the ID of this link through sensing this ACK(n, m) packet will be *disabled*, preventing them from requesting their matchings in the subsequent rounds. And after receiver senses ACK(n, m), this link is ready to be active in the data transmission phase.

Proposition 1: If $p = \frac{1}{|N(n)|+1}, \forall n \in \mathcal{N}$, the above algorithm has $O(1)$ time-complexity and returns a feasible schedule with a positive probability at least δ satisfying **C1** of the randomized scheduling framework as below:

$$\mathbb{P}[\mathbf{N}(t) = \mathbf{R}^{\star}(t)] \geq \delta = \left(\frac{1}{2e^{4\Delta^K}}\right)^N. \quad (5)$$

Proof: We clearly see that the maximum number of rounds for each node to terminate is 6, so the time complexity is $O(1)$. We realize that the link (n, m) will be matched successfully if during the whole operation of this algorithm, provided that nodes n and m are exchanging control packets, all nodes in K-hop neighbors of n and m must keep silent. We first consider the transmitter side, where node n sends all of its control packets successfully with a probability $\mathbb{P}_{n \to m}$:

$$\begin{aligned}
\mathbb{P}_{n \to m} &= \frac{1}{|N(n)|+1}\left(1 - \frac{1}{|N(n)|+1}\right)^{|N(n)|} \\
&\quad \times \underbrace{\left(1 - \frac{1}{|N(m_1)|+1}\right)^{|N(m_1)|} \cdots}_{|N(n)|\ times} \\
&\quad \times \ldots \times \underbrace{\left(1 - \frac{1}{|N(m_K)|+1}\right)^{|N(m_K)|} \cdots}_{|N(n)| \times |N(m_1)| \times \ldots \times |N(m_{K-1})|\ times} \\
&\geq \frac{1}{|N(n)|+1} e^{-1} \cdot e^{-|N(n)|} \cdot e^{-|N(n)||N(m_1)|} \cdots \\
&\qquad \cdot e^{-|N(n)||N(m_1)| \cdots |N(m_{K-1})|} \\
&\geq \frac{e^{-2\Delta^K}}{|N(n)|+1} \text{ when } \Delta > 1.
\end{aligned}$$

The final inequality follows from the fact that $\Delta = \max_{n \in \mathcal{N}} |N(n)|$ and $(1 + \Delta + \Delta^2 + \ldots + \Delta^K) \leq 2\Delta^K$ if $\Delta > 1$. By using the same analysis for the receiver side, when node m responds all of its control packets successfully, we have the probability $\mathbb{P}_{n \leftarrow m} \geq e^{-2\Delta^K}$. So the link (n, m) is matched with probability $\mathbb{P}_{(n,m)} \geq \frac{e^{-4\Delta^K}}{|N(n)|+1}$. Since the event node n is matched to node m or m' and the event $m \neq m'$ are disjoint, the probability that node n is matched is:

$$\begin{aligned}
\mathbb{P}[n\, is\, matched] = \sum_{m \in N(n)} \mathbb{P}_{(n,m)} &\geq \frac{|N(n)|\, e^{-4\Delta^K}}{|N(n)|+1} \\
&\geq \frac{e^{-4\Delta^K}}{2}. \quad (6)
\end{aligned}$$

Suppose that we have a feasible allocation $F \in \mathcal{R}$, which means $|F|$ nodes are matched, and the remaining nodes are inactive. The probability of picking an arbitrary feasible allocation F is:

$$\begin{aligned}
\mathbb{P}[F] &\geq \left(\frac{e^{-4\Delta^K}}{2}\right)^{|F|} \left(1 - \frac{e^{-4\Delta^K}}{2}\right)^{N-|F|} \\
&\geq \left(\min\left\{\frac{e^{-4\Delta^K}}{2},\ 1 - \frac{e^{-4\Delta^K}}{2}\right\}\right)^N \\
&\geq \left(\frac{1}{2e^{4\Delta^K}}\right)^N.
\end{aligned}$$

■

And $\mathbf{R}^{\star}(t)$ is also a feasible schedule so we have $\mathbb{P}[F = \mathbf{R}^{\star}(t)] \geq \delta = \left(\frac{1}{2e^{4\Delta^K}}\right)^N$.

C. Randomized Maximal Matching Algorithm

In this scheme, by running more iterations nested in a number of steps, we can improve the probability of a node matched.

Algorithm 2 Randomized Maximal Matching Algorithm

```
At each time-slot, each node n ∈ 𝒩 does
if n ∈ N_K(n') ∪ N_K(m') senses ACK(n', m'), ∀n', m' ∈ 𝒩 then
    disabled(n) = 1
if disabled(n) ≠ 1 then
    for s = 1 to (log N) do
        R_s(t) := R_{s-1}(t)
        for i = 1 to (C e^{4Δ^K} log N) do
            n chooses an arbitrary m ∈ N(n) and sends
            RTS with probability p
            if n senses other RTS's or senses COL then
                n sends COL
            if m senses n's RTS and senses no COL then
                m sends CTS
            if m senses other's RTS's then
                m sends COL
            if n senses m's CTS and senses no COL then
                n sends ACK(n, m)
            if m senses ACK(n, m) successfully then
                R_s(t) := R_s(t) ∪ (n, m)
```

We start by analyzing the performance of Randomized Maximal Matching Algorithm.

Proposition 2: If $C \geq 6$, then the Randomized Maximal Matching Algorithm has $O\left(e^{4\Delta^K} \log^2 N\right)$ time-complexity and can return a maximal schedule with the probability at least

δ satisfying **C1** of the randomized scheduling framework as below:

$$\mathbb{P}[\mathbf{N}(t) = \mathbf{R}^{\star}(t)] \geq \delta = 1 - \frac{1}{N^{C'}} \ (C' > 1). \tag{7}$$

Proof: Using the proof result of the first proposed algorithm, from (6), we have the probability that node n is not matched during an iteration i is $\mathbb{P}_i[n \, is \, not \, matched] \leq (1 - \frac{e^{-4\Delta^K}}{2})$. Thus, for one step, we have:

$$\mathbb{P}_s[n \, is \, not \, matched] \leq \left(1 - \frac{e^{-4\Delta^K}}{2}\right)^{C\, e^{4\Delta^K} \log N} \tag{8}$$

$$\leq e^{-\frac{e^{-4\Delta^K}}{2} \times C\, e^{4\Delta^K} \log N} \leq e^{-\frac{C \log N}{2}} = \frac{1}{N^{\frac{C}{2}}}. \tag{9}$$

Following the similar argument as in the proof of Proposition 1, we have exactly $|M|$ nodes of any maximal matching $M \in \mathcal{R}$ that are matched at the end of step s with a probability:

$$\mathbb{P}_s[M] \geq \left(1 - \frac{1}{N^{\frac{C}{2}}}\right)^{|M|} \geq 1 - \frac{|M|}{N^{\frac{C}{2}}} \geq 1 - \frac{1}{N^{\frac{C}{2}-1}}.$$

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

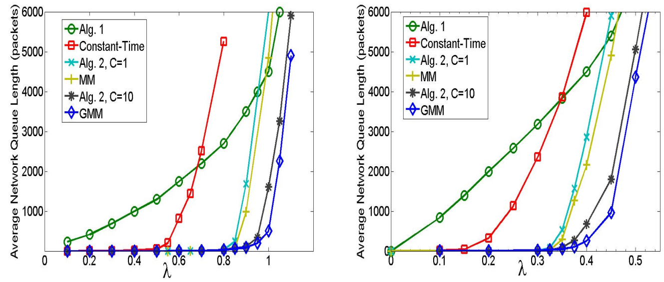
$$\mathbb{P}[M] \geq \left(1 - \frac{1}{N^{\frac{C}{2}-1}}\right)^{\log N} \geq 1 - \frac{\log N}{N^{\frac{C}{2}-1}}$$
$$\geq 1 - \frac{1}{N^{\frac{C}{2}-2}}.$$

Letting $C' = C/2 - 2$, and since $\mathbf{R}^{\star}(t)$ is also a maximal schedule so finally we have $\mathbb{P}[M = \mathbf{R}^{\star}(t)] \geq \delta = 1 - \frac{1}{N^{\frac{C}{2}-2}}$.

For time-complexity, the whole algorithm has at most $O(\log N)$ steps and each step includes $O(e^{4\Delta^K} \log N)$ rounds. So the Randomized Maximal Matching Algorithm takes $O(e^{4\Delta^K} \log^2 N)$ rounds of local message exchange totally. We complete our proof here. ■

V. Simulation Results

In this section, first we evaluate the throughput performance between Constant-Time (CT) [4], Maximal Matching (MM), Greedy Maximal Matching (GMM) algorithms and our proposed algorithms with Compare operation. The network topology we use for simulations is a 5×5 grid network. We implemented all algorithms under the single-hop traffic model. Every link has a capacity 1 packet/slot. There are 12 heavily-loaded links that are one-hop away from each other and they have Poisson traffic arrivals with mean 0.4λ. The remaining ones are lightly-loaded links with mean 0.2λ. In case of 1-hop interference model, Fig. 1a shows that with $C = 10$, Algorithm 2 has almost the same performance as GMM, which is close to the optimal and dominates other algorithms. This means that in this case Algorithm 2 can work at load very close to the threshold $\lambda = 1$ while in case of $\lambda \geq 1$, the queues become unstable. However, when $C = 1$, Algorithm 2 and MM are almost the same performance which is more than 80% capacity region. Empirically achieving more



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

Fig. 1: Capacity region of scheduling algorithms

than 80% and 60% capacity, the performance of MM and CT coincide with those in the literature [4], [5]. When the load is light (i.e. $\lambda \leq 0.6$), it shows that Algorithm 1 has the worst performance; however, when the load increases, there is a cross-over performance between Algorithm 1 and Constant-Time, MM and Algorithm 2 with $C = 1$. In case of 2-hop interference model (see Fig. 1b), the behaviors of all algorithms are similar to the first case except the capacity region approximately corresponds to $\lambda < 0.45$.

VI. Conclusion

In this work, we take into consideration maximizing throughput of wireless networks with distributed scheduling algorithms. So we choose Pick-and-Compare approach and generalize it into a randomized scheduling framework. We not only provide the analysis on how to achieve the optimal throughput with some constraints of this framework, but also give a upper bound for the delay of the whole system.

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] L. Tassiulas. Linear complexity algorithms for maximum throughput in radio networks and input queued switches, IEEE INFOCOM, 1998.

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

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

[5] S. Sanghavi, L. Bui, and R. Srikant, Distributed link scheduling with constant overhead, ACM Sigmetrics, 2007.

[6] E. Modiano, D. Shah, and G. Zussman, Maximizing Throughput in Wireless Networks via Gossiping,ACM Sigmetrics, 2006.