Computer Communications 32 (2009) 1257–1262

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

QoS based scheduling in the downlink of multi-user wireless systems (extended) ☆

Feng Wang, Jian Pu *, Mounir Hamdi

Computer Science and Engineering Department, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong

ARTICLE INFO

Article history:
Available online 26 February 2009

Keywords:
Multi-user wireless networks
Frame aggregation
Scheduling

ABSTRACT

Frame aggregation is a MAC-layer technology proposed in 802.11n WLAN. The base station can serve two or more users in one frame simultaneously, which can improve MAC-layer efficiency by reducing the transmission time for preamble and frame headers, and the random backoff period for successive frame transmissions. This fact enables us to design a more QoS-aware scheduler from the MAC layer. In this paper, we first formulate the scheduling problem with frame aggregation into a knapsack problem that is shown NP hard. Then we propose a simple approximation algorithm (LUUF) based on the unit urgency concept. Our analysis shows that the complexity of LUUF is $O(n \log n)$ and it achieves an approximation ratio of $F'/F_{\max}$. We then show that in practice the complexity can be further reduced to $O(n)$ and the approximation ratio can be made very near to 1, which makes LUUF a promising candidate for wireless systems that support frame aggregation. We also conduct simulations comparing LUUF with the widely used Round-Robin scheduler and find that LUUF can significantly improve the quality of service for various numbers of users and different maximum aggregation frame sizes.

1. Introduction

IEEE 802.11n [1] is proposed as an amendment of the previous IEEE 802.11 wireless networking standard to significantly improve network throughput. It aims at providing a data transmission rate of up to 600 Mbps. The version 2.0 draft specification for the next generation IEEE 802.11n WLANs has been approved in March 2007. It has introduced substantial enhancements at both the PHY (physical) and MAC (medium access control) layers for high throughput, efficiency and robustness [2,3] for the wireless system. In the PHY layer, based on the MIMO-OFDM (Multiple Input Multiple Output – Orthogonal Frequency Division Multiplexing) technology, 802.11n can use spatial multiplexing to transmit two or more data streams simultaneously. It also provides transmitter spatial diversity to improve reception by spreading the spatial streams across multiple antennas [4]. Beamforming, specified as an optional feature, can further improve packet transmission efficiency. The 802.11n defines a new set of the modulation and coding schemes (MCS), and the MCS is an index value that determines the modulation, coding and number of spatial streams in MIMO-OFDM systems. The actual transmission scheme is composed of both the MIMO mode and the MCS. The efficiency improvements at the MAC layer are frame aggregation, block acknowledgment (block ACK, also backward compatible with 802.11e [5]), etc. Frame aggregation [6–8] can improve MAC-layer efficiency by reducing the transmission time for preamble and frame headers, and the random backoff period for successive frame transmissions. They are particularly applicable to voice traffic where the voice frame is short and continuous traffic such as video or large file transfers.

We tackle the wireless scheduling problem from a cross layer optimization angle. We have done much research on the link adaptation algorithms for opportunistic scheduling in [9]. However, while the link adaptation improves transmission on a physical link, the aggregate system performance is very much dependent on multi-user scheduling and cross layer optimization mechanisms, which are also heavily coupled with underlying link adaptation. This cross layer optimization becomes more imperative in the 802.11n wireless systems since the standard has introduced many significant options in the MAC layer. In this paper, we will take advantage of the frame aggregation in the 802.11n for designing a multi-user scheduling algorithm. The scheduler tries to improve the system performance, in particular, in terms of the quality of service (QoS) efficiency.

Different from the well studied opportunistic scheduling that monitors the channel continuously and decides the locally optimized strategy to send packets to one user, scheduler with frame aggregation can send packets to several users simultaneously. How to select this set of users to be serviced is a challenging problem, which can be easily modeled into a *knapsack* problem that is well known to be NP hard [10]. We argue that, however, in practice we do not have to find the optimal solution to the knapsack problem to achieve a good performance. A simple greedy algorithm that

---

☆ This research work is partially funded by The Hong Kong Research Grants Council under the Grant RGC 610307.

* Corresponding author. Tel.: +852 6129 1847; fax: +852 2358 1477.
E-mail addresses: fwang@cse.ust.hk (F. Wang), pujian@cse.ust.hk (J. Pu), hamdi@cse.ust.hk (M. Hamdi).

0140-3664/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2009.02.013

has less complexity works sufficiently well with practical implementation of the frame aggregation.

The rest of this paper is organized as follows. In Section 2, we present the system model and conventions used throughout the paper. We then model the scheduling problem in the wireless system with frame aggregations into a knapsack problem in Section 3. In Section 4, we propose a simple greedy algorithm to do the scheduling and analyze its performance. In Section 5, we bring forward some practical considerations and argue that the greedy algorithm performs well in practice. In Section 6, some simulations and performance evaluation are present comparing our algorithm with the Round-Robin scheduling. Then, we conclude the paper in Section 7.

2. System model and conventions

In this section, we first describe the system model under consideration and put forward the scheduling problem. We also model the general frame aggregation scheme and formulate users' simple QoS requirements.

2.1. System model

We consider the downlink of a wireless system with n mobile stations (users) and one base station (BS). This general scenario applies to many wireless systems, such as 802.11 WLAN or the cell phone systems. We depict the framework in Fig. 1, where all the n users and BS are equipped with multiple antennas.

Users can join and leave. We assume the BS always has sufficient data for each user to download. The order and how much data of the users got serviced are based on their QoS requirements and the channel conditions, both of which are known at the central scheduler in the BS. Our task is to design such a scheduler to allocate the bandwidth to each user with the underlying physical constraints. If the BS can only select one user to service in one time, opportunistic algorithms work well. However, when the BS can select multiple users to service, things become different.

2.2. Frame aggregation

Different from the 802.11 a/b/g WLAN, we have an enhanced feature in the 802.11n downlink scheduling. That is, the BS can send data to multiple users *simultaneously*. And the batch of users selected to receive data is called a frame aggregation. For example, in Fig. 2, we can see that the first frame consists of packets to users 1, 3, ..., and n.

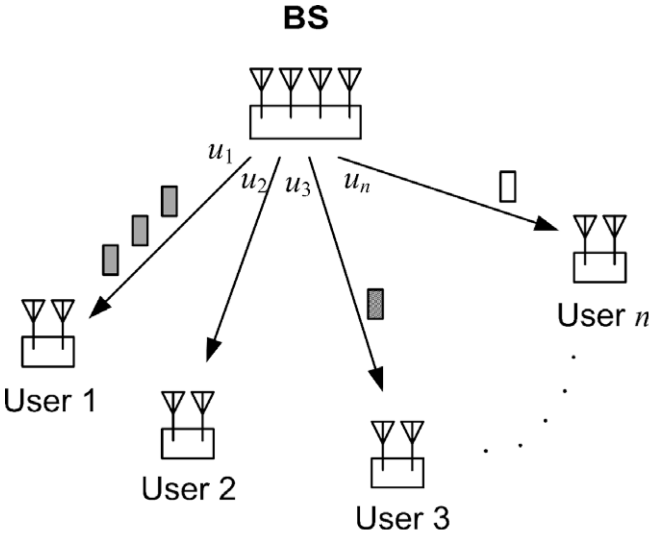

Fig. 1. A MIMO system with n users and one AP.

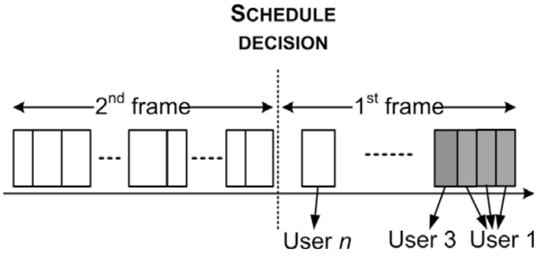

Fig. 2. Scheduling with frame aggregation.

In general, frame aggregation can increase the MAC-layer efficiency by not only reducing the transmission time for preamble and frame headers, but also reducing the waiting time during random backoff period for successive frame transmissions [11]. However, a larger aggregation frame will cause each station to wait for a longer time before its next chance for channel access. Furthermore, under error-prone channels, a larger aggregation frame may waste a longer period of channel time and lead to very low MAC efficiency. Therefore, there is a tradeoff between throughput and delay for frame aggregation at the MAC layer, and channel conditions should be taken into consideration when designing frame aggregation schemes. How to decide the optimal frame size, or even to find a way to adaptively adjust the frame size is out of the scope of this paper. For simplicity and put our paper in focus, we assume in the system under our consideration the frame size is variable and up-bounded by a preconfigured constant $F_{\max}$.

On the other hand, we note that inside each frame cycle packets to different users do not have a relative order since they are actually sent *simultaneously*, which might be misinterpreted in Fig. 2. In detail, this is decided by the MCS in lower PHY layer. Therefore, in each frame cycle, we just care about the size of data to send the selected users, not the order of them. In simple terms, if selected users data size are denoted by

$$\left\{p_{i_1}, p_{i_2}, \ldots, p_{i_k}\right\},$$

then we have this simple constraint in each frame cycle:

$$\sum_{j=1}^{k} p_{i_j} \leqslant F_{\max}.$$

Please note that for simplicity we do not care about how many packets each user receives. We consider all the packets to a user as a whole and denote them as p_i. For example, in Fig. 2, user 1 receives three packets, but we still only use p_1 to denote the data it receives in this frame cycle.

2.3. Modeling users' QoS requirements

What makes scheduling in the 802.11n WLAN a challenging task is that different users have different quality of service (QoS) requirements. In wired networks, the QoS requirements are generally described in delay and throughput. Here we consider an additional QoS requirement that is very important in the wireless networks for future applications with strict time constraints. Examples of such applications include streaming multimedia, voice over Internet Protocol (VoIP), instant messaging (IM), and real-time video conferencing, all of which have not only the delay requirement but also the minimum data requirements of users at each scheduling time slot [12]. Therefore, we assume a minimum size of packets $p_{i\min}$ for each user i. That is to say, if user i is selected to receive data, it should receive at least $p_{i\text{min}}$ data for current frame cycle; otherwise it would rather not to receive any data.

To schedule packets according to users' QoS requirements, we use a concept of *user urgency*. The idea is explained as follows. Each user in the system has some assigned initial urgency u_i. These initial u_i's and their individual evolution along time reflects their QoS fulfillments by the scheduler that tries to first schedule the most urgent packets. A user's urgency is reduced when it gets serviced from the BS. And the reducing rate reflects user's QoS requirement and the channel conditions weighed by the scheduler.

It is surely difficult to precisely model the evolution of users' urgency. We simplify the model by the following two assumptions:

A user's urgency is reduced when it received packets from the BS. The amount of reduction is proportional to the size of packets received. Different users have different proportions that reflect their QoS requirements and channel conditions.

If a user does not receive any packets from the BS, its urgency stays unchanged.

It is easy to see that the urgency u is a non-increasing function. It is somehow counter-intuitive, since as time evolves, the user may become more urgent if it does not receive any packets. However, we note that our definition of urgency is only to help us to do scheduling. It is not necessarily to comply with the real meaning of urgency. The intuition is that in the long term, every user may have the chance to be idle. It is fair if all users do not increase their urgency whenever idle. The scheduler has no bias on letting any user be idle, if we exclude the QoS factor.

We use c_i to denote the decreasing rate for user i. Besides the users' QoS requirements, the decreasing rate is also adjusted by the scheduler according to the channel conditions from one frame to another.

According to the above two assumptions, we can derive user i's urgency evolution equation as follows

$$du_i = -c_i \cdot dp_i,$$

where du_i is the change of user i's urgency and dp_i is the size of data it receives from the BS in current frame cycle.

Solve this differentiate equation, we get

$$u_i = u_i' - c_i p_i,$$

where u_i' is user i's urgency value in previous frame cycle and p_i is the data serviced in current frame cycle.

3. Knapsack problem and the largest unit urgency first scheduling

In each frame cycle, the scheduler is responsible to select a set of users to grant receiving packets.

With the modeling of users' urgency, it is obvious for a good scheduler to maximize the total users' urgency in each frame. Since the frame size is bounded by F_{max}, we formulate the scheduler problem as follows:

$$\begin{array}{ll} \max & \sum_{i=1}^{n} x_i u_i \\ \text{subject to} & \begin{cases} \sum_{i=1}^{n} x_i p_i \leqslant F_{\text{max}} \\ p_i \geqslant p_{i\text{min}}, \quad \text{for } i = 1, 2, \dots n \end{cases}, \end{array}$$

where $x_i = 0/1$, for $i = 1, 2, \dots n$. If $x_i = 1$, user i is selected for current frame cycle; otherwise it is not.

This is a combinatorial optimization problem. If we let $p_i = p_{i\text{min}}$, it becomes a typical *knapsack* problem. The knapsack problem is stated as follows. Given a set of items, each with a *cost* and a *value*, determine the number of each item to include in a collection so that the total cost is less than a given limit and the total value is as large as possible. For the 0/1 knapsack problem, each item can only be selected one copy. Returning to our scheduling problem, the cost is the packet size a user receives and the value is user's urgency.

It is well known that the 0/1 knapsack problem is an NP hard problem and therefore is computationally intractable when n is large.

In this section, we design a simple scheduler in each frame cycle. We call our proposed algorithm Largest Unit Urgency First (LUUF).

Algorithm 1. LUUF scheduling algorithm

```
Input
  Frame size upper bound $F_{\max}$,
  Users' urgency vector $\{u_1, u_2, \dots, u_n\}$,
  Users' minimum data vector $\{p_1, p_2, \dots, p_n\}$
Output
  A set of users $\{i_1, i_2, \dots, i_k\}$
Procedure
  FOR $i = 1, 2, \dots, n$
    Calculate $t_i = u_i/p_i$
  Sort the vector $\{t_1, t_2, \dots t_n\}$ in decreasing order to
  $\{t_{i_1}, t_{i_2}, \dots, t_{i_n}\}$
  Set $T = 0; k = 1$
  FOR $k = 1, 2, \dots, n${
    $T = T + p_{i_k}$
    IF $T \leqslant F_{\max}$
      Output $i_k$
      Reduce $u_{i_k}$ proportional to $p_{i_k}$
    ELSE
      $T = T - p_{i_k}$
      Continue loop
}
```

In simple terms, the LUUF first sort the users in a decreasing order according to their unit urgency u_i/p_i. Then the LUUF starts to add users from the largest unit urgency until the frame size up-bound is overflow.

Obviously, the selected users cannot always maximize the total urgency. We analyze the performance of LUUF in the following section and argue that with practical considerations, the LUUF achieves a rather good balance between the algorithm complexity and system performance.

4. Analysis of the largest unit urgency first scheduler

We first analyze the complexity of LUUF. It is easy to see that the complexity is $O(n \log n)$ since the LUUF basically involves two phases: sorting the unit urgencies and selecting users. It is well known that the sorting complexity is $O(n \log n)$ [10] and the selecting process costs $O(n)$ since it is a sequential process. Overall, the LUUF complexity is $O(n \log n)$.

To analyze the performance of LUUF, we should find out how close the solution given by LUUF is near the optimal solution.

We first define an approximation ratio of LUUF. If the maximal total urgency can be achieved in the frame cycle is U, and the total urgency given by LUUF is $\sum u_{i_k}$, we say the approximation ratio of LUUF is

$$\frac{\sum u_{i_k}}{U}.$$

It is easy to see that the approximation ratio cannot be larger than 1.

We have the following two arguments.

Argument 1: If the selected users by LUUF occupy the whole frame size F_{max}, the LUUF achieves the maximum total urgency, i.e., the approximation ratio is 1.

Argument 2: If the selected users occupy a size of $F' < F_{\max}$, the approximation ratio is larger than

$$\frac{F'}{F_{\max}}.$$

Proof. We prove argument 1 first.

Without losing generality, we assume the sorted unit urgency is

$$\frac{u_1}{p_1} \geqslant \frac{u_2}{p_2} \geqslant \cdots \geqslant \frac{u_n}{p_n}$$

And LUUF selects k users $\{1, 2, \ldots, k\}$, which enables

$$p_1 + p_2 + \cdots + p_k = F_{\max}.$$

Assume *any* other set of users $\{j_1, j_2, \ldots, j_m\}$, which also observes the constraint:

$$\sum_{h=1}^{m} p_{j_h} \leqslant F_{\max}.$$

We now compare their total urgencies

$$\sum_{i=1}^{k} u_i \quad \text{and} \quad \sum_{h=1}^{m} u_{j_h}.$$

Examine the two sets of users

$$\{1, 2, \ldots, k\} \quad \text{and} \quad \{j_1, j_2, \ldots, j_m\}.$$

Single out *all* different users in these two sets. Without losing generality, we assume

$$i_1, \ldots, i_t \in \{1, 2, \ldots, k\}, \text{ but } i_1, \ldots, i_t \notin \{j_1, j_2, \ldots, j_m\}, \text{ and}$$
$$j_1, \ldots, j_s \notin \{1, 2, \ldots, k\}, \text{ but } j_1, \ldots, j_s \in \{j_1, j_2, \ldots, j_m\}.$$

It is easy to see that

$$i_1, \ldots, i_t \leqslant k \text{ while } j_1, \ldots, j_s \geqslant k.$$

Since the unit urgencies are sorted in a decreasing order, we know that

$$\frac{u_{i_1}}{p_{i_1}}, \ldots, \frac{u_{i_t}}{p_{i_t}} \geqslant \frac{u_k}{p_k} \quad \text{and} \quad \frac{u_{j_1}}{p_{j_1}}, \ldots, \frac{u_{j_s}}{p_{j_s}} \leqslant \frac{u_k}{p_k}.$$

Using a simple property of the inequality, we have

$$\frac{u_{i_1} + \cdots + u_{i_t}}{p_{i_1} + \cdots + p_{i_t}} \geqslant \frac{u_k}{p_k} \quad \text{and} \quad \frac{u_{j_1} + \cdots + u_{j_s}}{p_{j_1} + \cdots + p_{j_s}} \leqslant \frac{u_k}{p_k}.$$

Therefore,

$$\frac{u_{i_1} + \cdots + u_{i_t}}{p_{i_1} + \cdots + p_{i_t}} \geqslant \frac{u_{j_1} + \cdots + u_{j_s}}{p_{j_1} + \cdots + p_{j_s}}.$$

It is obvious that

$$p_{i_1} + \cdots + p_{i_t} \geqslant p_{j_1} + \cdots + p_{j_s}$$

since the users selected by LUUF occupy all the $F_{\max}$.

Therefore,

$$u_{i_1} + \cdots + u_{i_t} \geqslant \frac{p_{i_1} + \cdots + p_{i_t}}{p_{j_1} + \cdots + p_{j_s}} (u_{j_1} + \cdots + u_{j_s}) \geqslant u_{j_1} + \cdots + u_{j_s}.$$

Together with the overlapped users, we know that the set of users selected by LUUF maximize the total urgency, compared with any other set of users observing the frame size constraint.

This finishes the proof of argument 1. □

We now move on to the proof of argument 2.

Still assume the LUUF selects a set of k users $\{1, 2, \ldots, k\}$. But now they only occupies a size of

$$F' = \sum_{i=1}^{k} p_i < F_{\max}.$$

To obtain the lower bound of the approximation ratio, we add an additional user to the system, user 0, who has

$$u_0 = \frac{u_k}{p_k} p_0 \quad \text{and} \quad p_0 = F_{\max} - F'.$$

It is easy to see that user 0 has the same unit urgency as user k.

Perform LUUF on the new set of users again, we know that the LUUF selects users $\{1, 2, \ldots, k, 0\}$ and the total urgencies of these $k + 1$ users are the maximum urgency according to argument 1, since the set of users $\{1, 2, \ldots, k, 0\}$ just occupies the whole frame size $F_{\max}$. And the maximum total urgency for these $k + 1$ users is

$$\sum_{i=1}^{k} u_i + u_0.$$

It is obvious that adding a new user can only increase the maximum urgency a system can achieve. Assume that without the new user the maximum total urgency is U. Therefore, U observes the following:

$$U \leqslant \sum_{i=1}^{k} u_i + u_0.$$

Denote the total urgency by LUUF to be $U' = \sum_{i=1}^{k} u_i$. Then

$$U \leqslant U' + u_0.$$

It is easy to see that

$$U' = \sum_{i=1}^{k} u_i = \sum_{i=1}^{k} \frac{u_i}{p_i} p_i \geqslant \sum_{i=1}^{k} \frac{u_k}{p_k} p_i = \frac{u_k}{p_k} \sum_{i=1}^{k} p_i = \frac{u_k}{p_k} F'.$$

Therefore, the approximation ratio is

$$\frac{U'}{U} \geqslant \frac{U'}{U' + u_0} = \frac{1}{1 + \frac{u_0}{U'}} \geqslant \frac{1}{1 + \frac{u_0}{\frac{u_k}{p_k} F'}} = \frac{1}{1 + \frac{\frac{u_k}{pk}(F_{\max} - F')}{\frac{u_k}{p_k} F'}} = \frac{F'}{F_{\max}}.$$

This finishes the proof of argument 2. □

5. Discussions

In the previous section, we analyzed the LUUF performance under the assumption that the frame size is fixed at $F_{\max}$. In fact, in practice we can have a more flexible scheduler. According to the LUUF, if the selected users only occupy a frame size of $F' < F_{\max}$, it is unnecessary for LUUF to wait for a period of $F_{\max} - F'$ to start next frame cycle. Therefore, we can adjust current frame size to F' and transmit packets according to the results of LUUF. According to argument 1 in Section 4, we know that the set of users maximizes the total urgency within frame size of F'. However, this does not necessarily mean that the LUUF finds optimal solution for the frame aggregation scheduling, because the frame size $F_{\max}$ or any other value is pre-determined by other tradeoffs. While we use the F' for actually transmission, the marginal gain of reducing the frame size is positive, but does not leads to a total system optimization. Nevertheless, we can safely say that using F' for actually frame aggregation size does improve the approximation ratio to be larger than $F'/F_{\max}$.

Another discussion is on the complexity improvement of LUUF. We have seen that its major complexity is due to the sorting process that is at least $O(n \log n)$ [10]. We show here that we can improve this complexity to $O(n)$ if all users have the same QoS requirements.

We use the idea of online algorithms and in each succeeding frame cycle LUUF uses the information from previous frame cycle. Revisit the LUUF. After each frame cycle, it selects a set of users $A = \{i_1, i_2, \ldots, i_k\}$ and leave with another set of users $B = \{1, 2, \ldots, n\} - \{i_1, i_2, \ldots, i_k\}$. We know that the users have already been ordered according to their unit urgency in each set. In the succeeding frame cycle, the users' urgency in B do not change, while in A, each user changes its urgency to

$$u_{i_k} = u'_{i_k} - c_{i_k} p_{i_k}$$

according to the urgency modeling in Section 2.3. Therefore, the unit urgency change is

$$\frac{u_{i_k}}{p_{i_k}} = \frac{u'_{i_k}}{p_{i_k}} - c_{i_k}.$$

Since all users have the same QoS requirements, we can assume the urgency decreasing rate for all users are the same, which means

$$c_{i_k} = c.$$

Therefore, the unit urgency order of users in A does not change in the succeeding frame cycle, since all users in it will decrease their unit urgency by a same amount. That is to say, when performing the sorting for LUUF in the succeeding frame cycle, we already have two sets of users sorted according to their unit urgency. We can simply use a *merging* operation on these two sets and get a full set of sorted all users. This merging operation simply cost a complexity of $O(n)$ [10]. Therefore, the total LUUF complexity can be reduced to $O(n)$ in each frame cycle except for the first frame.

6. Simulations and performance evaluation

In this section, we conduct some simulations comparing our scheduling algorithm LUUF with the Round-Robin scheduling algorithm. Round-Robin is a simple but widely used scheduling algorithm. In order to decide which users will get served in the next time slot, the Round-Robin scheduler keeps all users in a circular queue and visits them one by one. If adding the current user's frame does not lead to overflow the maximum aggregated frame size $F_{\max}$, the current user will be served. Otherwise, the scheduler will not add the current user's frame into the aggregated frame and continues to visit the next user in order. If all users in the circle have been visited but no more user frames can be added, the scheduling task is done. This end condition can be implemented using a counter variable (counting how many users have been skipped since the last successful frame addition). The problem of the Round-Robin scheduling is that it does not consider the urgency of the users and the order of users in the queue is kept as the same all the time. So the aggregated frame size maybe approaches the maximum frame size (as no more sub-frames can be added) but the quality of service (i.e., the total served urgency $\sum u_i$) may not be optimal.

The parameter settings of our simulations are:

1. maximum aggregation frame size $F_{\max}$: 1000–10,000 bytes, but the value is fixed for each simulation run;
2. number of users N: 5–100, fixed for each run;
3. users' urgency u_i: 10–100, randomly generated for each user;
4. users' sub-frame size p_i: 100–1000 bytes, randomly generated for each user.

For each simulation run, we choose different maximum frame sizes and different numbers of users. After the two algorithms (LUUF and Round-Robin) finish their scheduling, we compare their urgency summary $\sum u_i$ of served users, which is the quality of service considered in this paper. For each set of $\{F_{\max}, N\}$, we repeat the simulations for at least 100 times using different random seeds, and then average the performance improvement ratios over the repetitions.

We cannot include all of the extensive simulations we have conducted in this paper for the space limitation. Here we will present two typical results. The first one is about the effect of the number of users. We fix the maximum aggregation frame size at 3000 bytes and run the simulations for 5, 10, 20, ..., 100 users. As shown in Fig. 3, the performance improvement ratio of LUUF over Round-Robin is about 14–43%. And as the number of users increases, the improvement ratio grows, too. This is because more users give more choices of scheduling and better chance to find an optimal combination of sub-frames. The second result is about the effect of the maximum aggregation frame size. The number of users is fixed at 20 but the maximum aggregation frame size varies from 1000 to 10,000 bytes. As shown in Fig. 4, the performance improvement ratio of LUUF over Round-Robin is about 10–180%. The interesting thing is, when the maximum aggregation frame size is small, a bad choice of which sub-frame to serve may lead to the starvation of other more urgent users, which may be the reason of the large performance ratio in the figure.

Generally speaking, the Round-Robin scheduling is able to fill the available frame space as well as LUUF. The difference of the final assembled frame sizes of the two algorithms is minor in all cases (all approaching $F_{\max}$, not shown in the figures). But when we consider the quality of service (i.e., the total urgency of served users), LUUF is much better than the Round-Robin scheduling. The performance improvement ratio of LUUF over Round-Robin is about 10–180% as a result from the simulations.

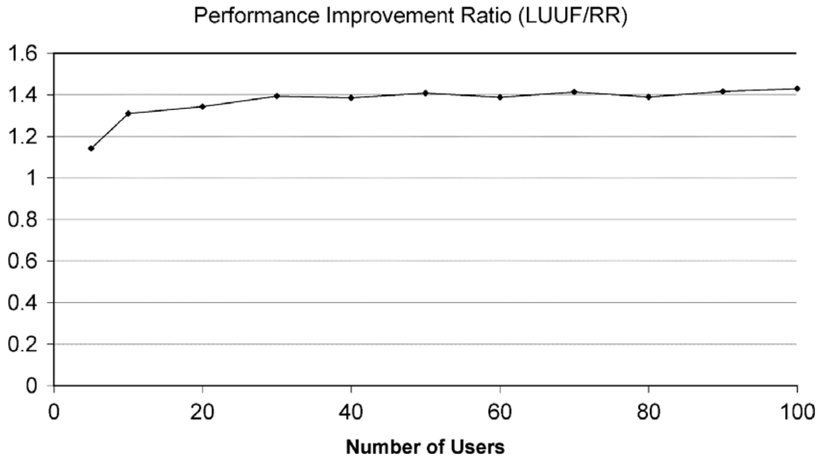

Fig. 3. The performance improvement ratio of LUUF over Round-Robin for different numbers of users.

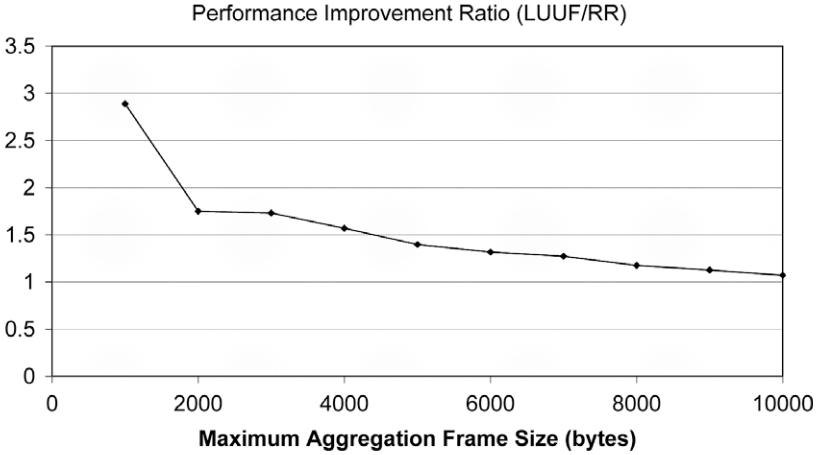

Fig. 4. The performance improvement ratio of LUUF over Round-Robin for different maximum aggregation frame sizes.

7. Conclusions and future work

In this paper, we investigate the multi-user scheduling in the MAC layer of wireless systems with support of frame aggregation. We first model the scheduling problem into a knapsack problem that is NP hard and computational intractable. We propose a simple and efficient algorithm (LUUF) to approximate the optimal solution. Our analysis shows that LUUF exhibits rather good performance when combined with practical considerations. In particular, the LUUF works better with variable frame size and can further reduce the algorithm complexity if all users in the system have the same QoS requirements. We believe LUUF is one of the promising MAC schedulers for the new approved 802.11n WLAN and other wireless systems with frame aggregation schemes.

Our future work along this line of research includes simulations of the LUUF with underlying link adaptation algorithms [9] and the finer modeling of the users' urgency. We thank anonymous reviewers for all their valuable comments.

References

[1] Y. Xiao, IEEE 802.11n: enhancements for higher throughput in wireless LANs, IEEE Wireless Commun. 12 (2005) 82–91 (see also IEEE Personal Communications).

[2] IEEE 802.11 WG, Draft PAR for High Throughput Study Group, May 2002.

[3] IEEE 802.11n/D2.0, Draft amendment to STANDARD [for] information technology-telecommunications and information exchange between systems – Local and Metropolitan networks – specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: enhancements for higher throughput, Std., March 2007.

[4] S. Nanda, R. Walton, J. Ketchum, M. Wallace, S. Howard, A high-performance MIMO OFDM wireless LAN, IEEE Commun. Mag. 43 (2005) 101–109.

[5] L.A.N. Wireless, Medium Access Control (MAC) and Physical Layer (PHY) specifications, part 11, IEEE Std., vol. 802.11e, 2005.

[6] D. Skordoulis, Q. Ni, H. Chen, A.P. Stephens, C. Liu, A. Jamalipour, IEEE 802.11n MAC frame aggregation mechanisms for next-generation high-throughput WLANS, IEEE Wireless Commun. 15 (2008) 40.

[7] Y. Kim, S. Choi, K. Jang, H. Hwang, Throughput enhancement of IEEE 802.11 WLAN via frame aggregation, in: Vehicular Technology Conference, 2004 (VTC2004-Fall), 2004 IEEE 60th, vol. 4, 2004.

[8] B.S. Kim, H.Y. Hwang, D.K. Sung, MAC 15-3-effect of frame aggregation on the throughput performance of IEEE 802.11n, in: Wireless Communications and Networking Conference, 2008 (WCNC 2008), IEEE, 2008, pp. 1740–1744.

[9] Q. Xia, M. Hamdi, Cross layer design for IEEE 802.11 WLANs: joint rate control and packet scheduling, in: Proceedings of the Fifth IEEE International Workshop on Wireless Local Networks, pp. 624–631.

[10] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, second ed., The MIT Press, 2002.

[11] Y. Lin, V.W.S. Wong, Frame aggregation and optimal frame size adaptation for IEEE 802.11n WLANs, GlobeCOM, San Francisco, November 2006.

[12] S. Pal, S. Kundu, M. Chatterjee, S. Das, Combinatorial reverse auction based scheduling in multi-rate wireless systems, IEEE Trans. Comput. 56 (2007) 1329–1341.