CROSS-LAYER MAC DESIGN FOR BANDWIDTH ALLOCATION IN WIRELESS AD HOC NETWORKS

Xueyuan Su*
Department of Computer Science
Yale University
xueyuan.su@yale.edu

Sammy Chan
Department of Electronic Engineering
City University of Hong Kong
eeschan@cityu.edu.hk

ABSTRACT

To support rate allocation schemes for wireless ad hoc networks at the MAC layer, a cross-layer MAC protocol is proposed in this paper to coordinate transmissions among flows and guarantee their allocated rates. Two different algorithms are implemented in this protocol: the first one is able to support an arbitrary bandwidth allocation scheme but has a longer convergence time, while the second one is specifically designed for a category of rate allocation schemes and thus have a much better convergence performance.

I INTRODUCTION

The wireless ad hoc network is a type of wireless network that does not have a wired infrastructure to support communication among the wireless nodes. Wireless network is interference-limited and flows traversing different wireless channels could interfere each other. This characteristic is referred as clique constraint [1], which is illustrated in Fig. 1. Fig. 1(a) depicts an example network, in which \( f_1 \) and \( f_2 \) are two subflows of the multi-hop flow \( f_1 \) and \( f_2 \) are two subflows of the multi-hop flow \( f \). The contentsions among subflows are better represented in a flow contention graph as shown in Fig. 1(b). In a flow contention graph, a vertex represents a subflow. If two vertices have a common edge, it means that the two subflows cannot be active simultaneously. In this example, there are two maximal cliques, i.e., \( \Omega(f_1, f_2) \) and \( \Omega(f_1, f_2, f) \). In each clique, only one subflow can be active at any time.

![Figure 1: Maximal cliques example](image)

Many rate allocation schemes [1]-[3] have been proposed to achieve a specific allocation objective among contending flows. Based on the rate allocation scheme, the bandwidth is distributed to the contending flows and target flow rates are calculated, according to a certain objective criterion. After the target rates are calculated in higher layers by the rate allocation schemes, a MAC protocol should support these schemes by allowing each flow to obtain its entitled bandwidth through medium access control. Unfortunately, the current MAC protocols, such as IEEE 802.11 DCF and other variants, are usually based on random access or intuitive fairness, and thus cannot satisfy this requirement. Therefore, a specially designed MAC protocol is needed.

In order to support the rate allocation schemes at the MAC layer, we propose a new MAC protocol in this paper. The objective of this MAC protocol is, given the rates calculated by a rate allocation scheme at the network layer, to guarantee the allocated flow rates to be achieved. As a result, it is a cross-layer MAC protocol. It is noted that the MAC design proposed in this paper is not trying to differentiate service types for different traffic flows, which is being discussed by IEEE 802.11e standard. Instead, the aim of our work is to support the rate allocation schemes existing in higher layers. Readers are referred to [4] for detailed analysis and supplementary discussions.

The rest of this paper is organized as follows. Some related work are discussed in Section II. Section III discusses the framework of this MAC protocol, and in Section IV and V, two algorithms are described and analyzed, respectively. Section VI concludes the whole paper.

II RELATED WORK

Through adjusting the persistence probability, [5] tries to achieve MAC layer fairness. A mechanism for translating a given fairness model into a corresponding contention resolution algorithm is proposed. According to this translation procedure, by carefully choosing the adjustment function, some specific fairness criteria are proved to be achieved. This work provides the path for achieving fairness objectives over shared wireless channels; however, as the authors pointed out, as wireless networks move from the academic domain to the commercial domain, these mechanisms must support network level quality of service, which needs to be translated to MAC level fairness objective. In [1], max-min fairness is achieved by scheduling the access of single-hop flows to the medium, according to the calculated fair share. However, the achieved max-min fairness is for MAC layer subflows, not for the end-to-end multi-hop sessions. A two-tier distributed and iterative algorithm is proposed in [2]. The first tier algorithm completes the per-clique price calculation, and then the second tier algo-
rzithm performs per-node price calculation. At this point, the transmission of each node is scheduled according to per-node price. After careful choice of utility functions, proportional fairness and max-min fairness could be achieved. However, the iteration process of this algorithm has a relatively long convergence time, which might limit the application scope of this algorithm. The scheduling algorithm for arbitrary rate allocation adopted in [6] is based on tag marking and virtual clock. Each node maintains a virtual clock and uses the tag to estimate an approximate backoff value for each packet queued in its buffer. This algorithm only guarantees that subflows from the same flow receive approximately the same channel share. [7] combines a link scheduling that avoids collisions, a fair service discipline per link, and a hop-by-hop window flow control into one algorithm. The max-min fair scheduling is achieved by generating tokens at source nodes and max-min packets releasing process. The scheduling module is not built for arbitrary rate allocation schemes. IEEE 802.11e has been approved in late 2005 as a standard that defines a set of QoS enhancements for 802.11 Wi-Fi standard. Enhanced Distributed Channel Access (EDCA) is proposed as an improvement of DCF access method by defining priority for different type of traffic flows. However, it is not capable of ensuring the bandwidth entitlement for each multi-hop flow.

III THE FRAMEWORK OF MAC PROTOCOL FOR BANDWIDTH ALLOCATION

This section discusses the framework of our proposed MAC protocol for bandwidth allocation in wireless ad hoc networks. Recalling that flows from different maximal cliques could be active simultaneously, while flows within the same clique contend for the shared bandwidth, the MAC protocol would thus basically coordinate the data transmission of different subflows within each maximal clique. In this framework, the transmission time is slotted and packets are sent out at the beginning of each time slot. In each maximal clique, every $N$ time slots are defined as a transmission frame. Each transmission frame is regarded as a unit to estimate the share of the bandwidth within the corresponding maximal clique. The value of $N$ is relative to the precision of the calculated flow rates provided by the rate allocation schemes. Let $x_{i,j}$ and $c_l$ represent the flow rate of $f_{i,j}$ and effective bandwidth of link $l$ respectively, without losing generality, in the following discussion, we assume $N = 100$ so as to use 33 out of 100 time slots in a frame to represent the share of $x_{i,j}/c_l = 1/3$.

In each node, packets from different subflows are queued in different buffers. All the nodes maintain a clique occupancy table (COT) for each maximal clique with which they associate. Therefore, for some nodes in the network, they will maintain multiple COTs. For example, in Fig. 1, node A maintains a COT for maximal clique $\Omega(f_{1,1}, f_{1,2})$, while node C maintains two COTs respectively for maximal cliques $\Omega(f_{1,1}, f_{1,2})$ and $\Omega(f_{1,2}, f_{2,1}, f_{2,2})$. The length of each COT equals $N$, according to the length of the transmission frame. The content of the COT describes the occupancy result within the corresponding maximal clique. When a subflow $f_{i,j}$ successfully occupies a free time slot in a frame, all the nodes in the same maximal clique will update their COTs to mark the time slot with $f_{i,j}$.

Also, in subsequent frames, each subflow will send packets at the same time slots that it has already occupied, i.e., marked with its ID in COTs. If the number of occupied time slots is not enough to satisfy the entitlement (this situation happens when the allocated rate is not yet achieved or some flows have left the maximal clique), more time slots will be occupied by this subflow. Similarly, if the number of occupied time slots exceeds its entitlement (this situation happens when some new flows have joined the maximal clique), the subflow would release some occupied time slots accordingly. Therefore, the proposed MAC protocol is compatible with both static and dynamic flows.

Based on this framework of the MAC protocol, two different MAC algorithms will be proposed in the following sections.

IV GREEDY SELF-CONTENTION ALGORITHM

IVA Algorithm Description

Greedy Self-Contention (GSC) algorithm is a contention-based algorithm. In multiple access protocols, contention resolution has typically been achieved through two mechanisms: persistence and backoff [5]. The persistence probability and backoff window are functions of the estimated contention, and different contention resolution algorithms differ in terms of how they adjust these parameters in response to collisions and successful transmissions.

In each maximal clique, the normalized bandwidth bound never exceeds 1 and for each subflow $f_{i,j}$, it occupies the medium with a time fraction of $x_{i,j}/c_l$. This time fraction has a natural interpretation as a probability. Therefore, in GSC, each subflow uses $p_{i,j} = x_{i,j}/c_l$ as the persistence probability to contend for the free time slot. Initially, all COTs are empty, which means that all time slots in a transmission frame are free. At the beginning of a free time slot, each subflow transmits a packet with probability $p_{i,j}$. If subflow $f_{i,j}$ is the only subflow transmitting a packet, it would successfully obtain this time slot. All the other nodes in the same maximal clique that overhear this transmission mark the corresponding time slot in their own COTs with $f_{i,j}$. If collision happens, no subflow successfully obtains this time slot and thus no mark is made. Since the acquired bandwidth is estimated within each transmission frame, if some time slots are not occupied in the current frame, they could only be occupied in next frame. At this standpoint, in order not to affect the convergence time and to have a simple implementation approach, GSC algorithm does not adopt any backoff mechanism.

When the number of occupied time slots $n_{i,j}$ for subflow $f_{i,j}$ satisfies the equation:

$$\frac{n_{i,j}}{N} = \frac{x_{i,j}}{c_l},$$

it adjusts the persistence probability to zero. In other words, it would not contend for free time slots any more. This equation implicitly shows that the bandwidth acquired by subflow $f_{i,j}$ is already equal to its entitlement.

In subsequent frames, each subflow will transmit in the time slots marked with its ID in the COT. Finally, this contending
process converges to steady state, in which all the transmission frames have the same content, i.e., the active subflows in corresponding time slots are the same. In steady state, every subflow occupies a number of slots according to (1) in order to obtain its entitled bandwidth.

**IV.B Related Discussions**

From the process described above, it can be seen that given \( x_{i,j}/c \), each subflow tries to occupy \( n_{i,j} \) out of \( N \) time slots in a frame, where \( \frac{n_{i,j}}{c} = \frac{x_{i,j}}{c} \). After it successfully occupies \( n_{i,j} \) time slots, its persistence probability is set to zero and it will not contend for free time slot anymore. Therefore, no subflow could occupy more time slots than it is entitled to.

In order to simplify the notations, in the following discussion, we assume there are \( n \) subflows in the maximal clique \( \Omega \) and use \( f_i \) to represent the subflow. We define \( x_i \) as the rate of subflow \( f_i \) and \( n_i \) as the number of time slots it should obtain according to the allocated rate.

(i) **Clique occupancy table exchange**

**Proposition 1** If we call the two communicating nodes of each subflow as its transmission pair, for each subflow in a maximal clique, at least one end of its transmission pair can overhear the ongoing transmission in the maximal clique.

**Proof.** The transmission in our discussion refers to the bi-directional transmission including DATA packets and ACK packets. Let us assume that neither of the transmission pair can overhear the ongoing transmission within the maximal clique. It means that both of them are out of the transmission range of the ongoing active subflow. So they can perform the communication without causing interference in the maximal clique. This contradicts with the fact that no two subflows in a maximal clique can be active at the same time. Hence, at least one node of a transmission pair could overhear the active subflow in the network. \( \blacksquare \)

Based on Proposition 1, if COTs are exchanged between the two ends in each transmission pair, all the nodes in a maximal clique can maintain the COTs with the same content. Note that the COT exchange only occurs within a transmission pair, so the control information exchange is bounded locally by each maximal clique. In addition, COTs can be piggybacked in DATA/ACK packets to be exchanged between a transmission pair, thus no new packet format needs to be introduced.

(ii) **Mismatches of COTs between a transmission pair**

The COTs between a transmission pair should match each other. But sometimes mismatch may happen. Consider the situation in Fig. 2, where there is a maximal clique \( \Omega(f_1, f_2, f_3) \).

Consider subflows \( f_1 \) and \( f_2 \) are both trying to transmit a packet in the same free time slot. Of course collision happens and no one should mark the slot. However, node E is out of the range of subflow \( f_2 \) and just overhears the transmission of subflow \( f_1 \). It does not know collision has happened and according to the algorithm, it will mark the corresponding time slot in its own COT with \( f_1 \). Same thing happens to node F and it will mark the corresponding time slot with \( f_2 \). So mismatch between node E and F happens.

To deal with this problem, the following rule is used to regulate the COT exchange between a transmission pair. When the COT from the other end arrives, this end first compares it with the one of its own. If it finds that for a corresponding time slot, two COTs are marked with two different subflows, it concludes collision has happened. It then deletes the subflow marked in this time slot and inform the other end this situation. Only if its own COT is empty in a time slot, it copies the corresponding entry in the COT of the other end.

**IV.C Convergence Time Evaluation**

The upper bound of convergence time in each maximal clique is derived in this subsection. We assume in a maximal clique \( \Omega \), there are \( n \) subflows. Let \( \omega \) be the set of these \( n \) subflows. Before obtaining the entitled bandwidth, each subflow \( f_i \) will contend for every free time slot with probability \( p_i = x_i/c_i \). The probability to obtain a free time slot for this subflow is \( p_i \prod_{j \neq i} (1 - p_j) \). On average, after contending for \( 1/[p_i \prod_{j \neq i} (1 - p_j)] \) free time slots, subflow \( f_i \) can successfully transmit once. Let \( N_a \) denotes the number of free time slots that subflow \( f_i \) has contended for before obtaining \( n_i \) time slots.

Assume subflow \( f_a \) is the first one to obtain its entitled bandwidth, i.e., obtaining \( n_a \) time slots it needs. Then the average number of free time slots it has contended for should be

\[
N_a = p_a \prod_{i \in \omega, i \neq a} (1 - p_i) \tag{2}
\]

After subflow \( f_a \) gets \( n_a \) time slots, its persistence probability is set to zero. If subflow \( f_b \) is the second one to obtain its entitled bandwidth, then on average the number of free time slots it has contended for should be

\[
N_b = p_b \prod_{i \in \omega, i \neq b} (1 - p_i) + p_{ba} \prod_{i \in \omega, i \neq a, b} (1 - p_i), \tag{3}
\]

where \( n_{ba} \) is the number of time slots \( f_b \) obtains before subflow \( f_a \) obtains its entitled bandwidth, \( n_{bb} \) is the number of time slots \( f_b \) obtains between the time \( f_a \) obtains its entitled bandwidth and the time it obtains its entitled bandwidth. It is obvious that \( n_{ba} + n_{bb} = n_b \).

Then subflow \( f_b \) also stops contending for more time slots. Following this way, for the last subflow \( f_h \), the average number...
of free time slots it has contended for is

\[ N_k = \frac{n_{ka}}{p_k} \prod_{i \in \xi_k} (1 - p_i) + \frac{n_{kb}}{p_k} \prod_{i \in \omega, j \neq i} (1 - p_i) + \cdots + \frac{n_{kb}}{p_k} \prod_{i \in \omega, j < k} (1 - p_i), \]  

(4)

where \( n_{ka} + n_{kb} + \cdots + n_{kb} = n_k \).

Let us consider the worst situation: all the subflows occupy the required time slots strictly one by one, i.e., before subflow \( f_a \) obtains \( n_a \) time slots, the other subflows do not obtain any time slot, then subflow \( f_b \) obtains \( n_b \) time slots followed by \( f_c \) obtaining \( n_c \) time slots and so on. In this situation, the convergence time is the longest one. Hence, (3) becomes

\[ N_b = \frac{n_b}{p_b} \prod_{i \in \omega, j \neq i} (1 - p_i), \]  

(5)

which means that before subflow \( f_a \) obtains all the \( n_a \) time slots, \( f_b \) does not obtain any time slot. And (4) becomes

\[ N_k = \frac{n_k}{p_k}, \]  

(6)

Let subflow \( f_a \) be the first one to obtain its entitled bandwidth. In the first frame, since all the time slots are free, the number of time slots it can obtain in this frame is

\[ n_a(1) = N \cdot p_a \prod_{i \in \omega, j \neq a} (1 - p_i) \]  

(7)

Note that it is a statistical analysis here, so \( n_a(1) \) may not be an integer.

Then in the second frame, \( n_a(1) \) time slots are already occupied by \( f_a \) and the number of free time slots is \((N - n_a(1))\). So it can obtain

\[ n_a(2) = (N - n_a(1)) \cdot p_a \prod_{i \in \omega, j \neq a} (1 - p_i) \]  

(8)

more time slots in the second frame.

Repeat the calculation in this way, finally in the \( \xi_a \) frame, after obtaining

\[ n_a(\xi_a) = (N - \sum_{j=1}^{\xi_a-1} n_a(j)) \cdot p_a \prod_{i \in \omega, j \neq a} (1 - p_i) \]  

(9)

more time slots, subflow \( f_a \) obtains \( n_a = \sum_{j=1}^{\xi_a} n_a(j) \) time slots in total and sets its persistence probability to zero.

As we assumed above, in the next frame after \( \xi_a \), subflow \( f_b \) begins to obtain time slots. In this frame, since subflow \( f_a \) has already occupied \( n_a \) time slots, the number of free time slots is \((N - n_a)\). So in its first frame, \( f_b \) can get

\[ n_b(1) = (N - n_a) \cdot p_b \prod_{i \in \omega, j \neq b} (1 - p_i) \]  

(10)

In the same way, for its second frame, \( f_b \) gets \( n_b(2) = (N - n_a - n_b(1)) \cdot p_b \prod_{i \in \omega, j \neq b} (1 - p_i) \) time slots, and in its last frame,

\[ n_b(\xi_b) = (N - n_a - \sum_{j=1}^{\xi_b-1} n_b(j)) \cdot p_b \prod_{i \in \omega, j \neq b} (1 - p_i). \]  

At this moment, \( f_b \) obtains its entitled bandwidth with \( n_b = \sum_{j=1}^{\xi_b} n_b(j) \) and sets its persistence probability to zero.

If we repeat the calculation on the last subflow \( f_k \) in the same way, until after a specific frame \( \xi_k \), the remaining time slot for \( f_k \) is one.

\[ n_k(\xi_k) = (N - \sum_{i=1}^{b} n_i - \sum_{j=1}^{\xi_k-1} n_k(j)) \cdot p_k \]  

(11)

Then \( f_k \) will spend \((1/p_k)\) more frames to obtain the last time slot and obtain its entitlement with \( n_k = \sum_{j=1}^{\xi_k} n_k(j) + \frac{1}{p_k} \).

After \( f_k \) obtains \( n_k \) time slots, all the subflows have obtained their entitled bandwidth and bandwidth allocation is completed. As we assumed that subflows obtain the time slot one by one, so the overall convergence time is

\[ \xi = \sum_{i=1}^{\omega} \xi_i + \frac{1}{p_k} \]  

(12)

This means that after \( \xi \) frames, each subflow \( f_i \) has obtained required number of time slots \( n_i \) and the allocated entitlement is achieved. Note that Equation (12) provides an upper bound of the convergence time. The usual convergence time should be less than this value since subflows are not constrained to obtain the time slots strictly one by one.

We calculate the upper bound of convergence time following the approach described above in three different cases and carry out 1000 simulations for each case to obtain the average convergence times. The results are shown in Table 1. As mentioned before, the convergence time is usually less than that in the worst situation.

<table>
<thead>
<tr>
<th>Upper Bound</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N = 10, f_1 = \frac{1}{4} )</td>
<td>10</td>
</tr>
<tr>
<td>( N = 100, f_1 = \frac{1}{4} )</td>
<td>12</td>
</tr>
<tr>
<td>( N = 100, f_1 = \frac{1}{3}, f_2 = f_3 = \frac{1}{4} )</td>
<td>18</td>
</tr>
</tbody>
</table>

V COOPERATIVE TOKEN FORWARDING ALGORITHM

VA Algorithm Description

The deficiency of GSC algorithm is the relatively long convergence time due to its contention operation. Some recently proposed rate allocation schemes, for example, the schemes in [2][3][6], have an inherent characteristic that the local flow contention graph and the local contention information are available among all nodes in the same maximal clique. In other words, nodes in the maximal clique could be regarded as quasi-synchronized with the same flow contention information.

Cooperative Token Forwarding (CTF) algorithm takes advantage of this inherent characteristic, and thus eliminates the
contention among nodes in a maximal clique. In CTF, each node is assigned with a priority determined by its MAC address. Since all the nodes in the same maximal clique are aware of the flow information within the clique according to the rate allocation scheme, the assigned priorities are also available to all of them. Nodes with smaller MAC addresses are assigned with higher priorities (the reverse ordering is also applicable). In other words, if the sending node of a subflow has the smallest MAC address, this subflow has highest priority to acquire the time slots. Some subflows may share the same sending node. In this case, there is no need to distinguish these subflows from different priorities. When such a sending node has acquired the right for transmission, it will internally schedule the transmission of its subflows. From the discussion above, it is noted that the priority assignment is naturally based on the quasi-synchronous characteristic of the rate allocation scheme, and thus no central coordinator is needed.

When a node joins the maximal clique, it generates a token. The token is a logical symbol with which a node could transmit data packets. After this node overhears the medium and exchanges information with one-hop neighbors, if it finds another node with higher priority, it would destroy the token generated by itself. In this way, at the beginning of the algorithm, all the nodes generate a token themselves, and then only one token which is generated by the node with the highest priority is maintained in each maximal clique.

Only the node with the token is eligible to transmit data packets, other nodes would just listen to the medium and update their COTs. When a node obtains the token, it should first check whether it still needs more time slots in this transmission frame, and whether it is ready for transmission. If both conditions are satisfied, it will begin transmission; otherwise, it would forward the token to the next node. Initially, the node with highest priority obtains the token and keeps on transmitting data packets and occupies continuous time slots. When the number of occupied time slots satisfies Equation (1) for all subflows, the node forwards the token to another node with the second highest priority. Then this node repeats the procedure and forwards the token to the next one. When the node with the lowest priority satisfies its entitlement, it returns the token to the node with highest priority so as to launch a new round of transmission. When the token is lost due to unexpected node or link failures, the node with highest priority is eligible to generate a new token and continue the transmission. The process continues until all subflows satisfy their bandwidth entitlement.

V.B Token forwarding strategy

With CTF in an individual maximal clique, the bandwidth allocation would be completed within one transmission frame. However, since a node only forwards the token after occupying enough time slots to obtain its entitled bandwidth, nodes with lower priorities are expected to occupy the latter portion of the transmission frame. This may lead to the situation that packets accumulate and cause congestion at those nodes.

In order to solve this potential problem, the token forwarding strategy could be modified as follows: when a subflow successfully occupies \( n'_{i,j} \) time slots so that \( \frac{n'_{i,j}}{N} = p\% \times \frac{c_{i,j}}{c_{i}} \), it would forward the token to the next subflow. Here \( p\% \) is a tunable parameter. For example, if \( p\% = 20\% \), the subflow with the token will forward it to the next subflow after it satisfies 20% of its bandwidth entitlement. This strategy could give opportunity to nodes with lower priorities to transmit data packets at earlier time and thus help solve the potential problem. In addition, this strategy will not affect the ultimate bandwidth allocation since the rate of each subflow is still bounded by its entitlement.

We carry out 1000 simulations for a network with three maximal cliques. The average convergence times under different values of \( p\% \) are shown in Table 2.

<table>
<thead>
<tr>
<th>( p% )</th>
<th>20%</th>
<th>60%</th>
<th>100%</th>
</tr>
</thead>
<tbody>
<tr>
<td>convergence time</td>
<td>7.4</td>
<td>3.2</td>
<td>1.9</td>
</tr>
</tbody>
</table>

As expected, the convergence time achieved by CTF algorithm is much shorter than GSC algorithm. Moreover, smaller value of \( p\% \) would more effectively solve the potential congestion at nodes with lower priorities, but it would prolong the convergence time.

VI CONCLUSION

A novel MAC protocol for bandwidth allocation in wireless ad hoc networks is proposed in this paper. Given the flow rates allocated at the network layer, this MAC protocol ensures flows to achieve the target rates. Two different algorithms are implemented in this protocol. The first one is a contention-based algorithm and could achieve arbitrary bandwidth allocation. In order to accelerate the convergence speed, the second algorithm is specifically designed for the category of quasi-synchronous rate allocation schemes and thus its convergence performance is much better. Both of the MAC algorithms are distributed and easy to implement.

REFERENCES