A Unified Algorithm for Wireless MAC Protocols

Chao-Ming Teng and Kwang-Cheng Chen
Graduate Institute of Communications Engineering
College of EECS, National Taiwan University, Taipei 10617, Taiwan
Email: cmteng@santos.ee.ntu.edu.tw ; chenkc@cc.ee.ntu.edu.tw

Abstract— **Being confronted with numerous MAC protocols designed under a variety of networking considerations, we envision a softwared-defined MAC controller that can be "re-configured" to different MAC protocols. This paper presents a unified algorithm for wireless MAC protocols, a pioneer trial of this vision that benefits future wireless networks. This unified algorithm is based on the concept of MULCAR, a generalized model for MAC protocols proposed by Chen and Sun. We oberved that most protocols operate in a cycling fashion and identified major differences between each algorithm. Combined this with the concept of MULCAR, we unified several representative wireless MAC protocols into one parameterized algorithm. Among them are ALOHA with geometric backoff, binary exponential backoff, and Q-ary collision resolution algorithm, *p*-persistent CSMA, CSMA/CA, and GRAP. One can thus have the unified algorithm operate as different MAC algorithms with proper paremter setting, which enables the development of the software-defined MAC controller.**

I. INTRODUCTION

For more than three decades, researchers and protocol designers have proposed and analyzed numerous Medium Access Control (MAC) protocols with different considerations, which have resulted in changing dynamics such as different MAC protocols in each country and in each technology, competing standards that evolve with enhancements, ever-changing system requirements, growing cost and so on. Facing this fact, we envision a software-defined MAC controller that implements a "reconfigurable" MAC algorithm that can be configured to operate as different protocols by different parameter settings. Users may thus switch to different MAC protocols by different parameter configurations, which makes it possible to roam among different systems with the same device. Furthermore, enhancements or evolutions of standards can be achieved by software upgrade without hardware re-design. Therefore, working toward a reconfigurable MAC algorithm that unifies major MAC protocols greatly benefits the future wireless access technologies.

However, this work is challenging due to the fact that MAC protocols differ from one another in numerous aspects. In order to unify protocols in a systematic way, a fundamental understanding of the general rules of MAC protocols is vital. Inspired by Gallager's work [1], Chen and Sun proposed the MULCAR (MUlti-Layer Collision Avoidance/Resolution) model [6, 7] that serves as a general framework for different MAC protocols. MAC protocols, in general, differ mainly in the way of avoiding or resolving collisions. In terms of MULCAR, we say that different protocols employ different Collision Anticipation Tree Expansion (CATE) or Collision Resolution Tree Expansion (CRTE) to avoid or resolve collisions. CA/CRTE split contending/collided transmissions into different groups, and the splitting can be done in several domains including time, space, signaling, probability, and so forth [7]. Therefore, we adopt CA/CRTE to specify the mechanisms to avoid or resolve collisions among different protocols, which makes unification feasible.

This paper is organized as follows. Section II gives the network model we considered in this paper. We then re-write several representative MAC algorithms using MULCAR. Among them are the ALOHA protocol with geometric backoff, binary exponential backoff, and Q-ary collision resolution algorithm (Section III), p-persistent Carrier Sense Multiple Access (CSMA) and CSMA/CA (Section IV), and Group Randomly Addressed Polling (GRAP) (Section V). To ensure the correctness of these re-written algorithms, we show that each re-written algorithm and its original corresponding one share the same state transition diagram in Section VI and validate the unification model with an analytical example as well. Finally, the unified algorithm is present in Section VII with the parameter configuration. This is achievable since these re-written algorithms are based on the same framework, i.e., MULCAR. Section VIII concludes this work.

II. THE NETWORK MODEL

The network model we consider is as follows. We currently focus on slotted systems throughout this paper. We also assume that the channel is noiseless and there is no capture effect.

III. THE ALOHA ALGORITHM

A. The MULCAR_ALOHA Algorithm

In the slotted ALOHA protocol, a node with ready packets transmits freely at slot boundaries. When collision occurs, each involved node randomly chooses a time interval to "backoff" according to some specific rules. After the chosen time interval passes by, the node retransmits the collided packet, hoping that this time it may transmit successfully.

Figure 1 shows a typical operation of slotted ALOHA. Point A in Figure 1 is the time instant that previous transmission ends, and if it is a collision, each node has to choose a number of slots to backoff according to the backoff rule and retransmit. Consulting Figure 1, we made some observations:

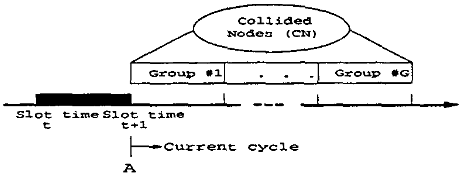


Fig. 1. The operation of the Slotted ALOHA protocol.

1. *Random backoff is a technique to perform tree expansion.* In MULCAR's point of view, randomly choosing a time interval to backoff represents a form of tree expansion. Check Figure 1 and say at time slot t, two or more nodes transmit their ready frames. Assume immediate feedback, nodes receive feedback at the end of the slot. Those who receive collision feedback join the set of *Collided Nodes* (CN) and as shown in Figure 1, are split into different groups that are numbered from 1 to G. CRTE splits collided nodes into different groups according to some specified rules such as geometric backoff, binary exponential backoff (BEB), Q-ary collision resolution algorithm and so forth. Nodes that are splitted by CRTE to group #g is scheduled to retransmit at slot g, counting from the end of the previous transmission (e.g. Point A in Figure 1).
2. *The protocol operates in a cycling style.* Consult Figure 1 again and say that group #3 is the first nonempty group after CRTE is performed. In this case, after two slots pass by, nodes in group #3 will transmit their frames and the whole process repeats all over again-that is, the ALOHA algorithm operates as cycles running indefinitely. In fact, most MAC protocols operate in a similar cycling style while they differ in the way of determing when to repeat the cycle. Therefore, identifying the timing to repeat a cycle plays an important role in the unification of algorithms.

With these observations, we may rewrite the slotted ALOHA algorithm below with the concept of cycles and CRTE.

The MULCAR_ALOHA Algorithm

```
RP_1  //Renewal Point 1
update G; // G is the maximum TE size
unmarked nodes in CN call
  CRTE(type_CRTE);
```

```
associate marked nodes in CN to group
  number #(original group number-g);
set g=1; //start to process each group
RP_2  //Renewal Point 2
nodes with new arrival packets during
  the processing of group #(g-1) -> TX(g);
nodes in group #g -> TX(g);
process group g with GP(gp_scheme);
if(there is no transmission){
  increase g by 1;
  if(g>G){
    goto RP_1;}
  else{
    goto RP_2;}}
elseif(there is a transmission){
   if(the transmission is a success){
      the successful node removes the
      transmitted packet from buffer;
      if(completeness is set){
         increase g by 1;
         if(g>G){goto RP_1;}
         else{goto RP_2;}}
      else{mark the losers;
           current cycle ends and
           goto RENEWAL_1;}}
   else{collided nodes -> CN;
        if(completeness is set){
        g++;
        if(g>G){goto RP_1;}
        else{goto RP_2;}}
      else{mark the losers;
           current cycle ends and
           goto RP_1;}}}
```

In the MULCAR_ALOHA algorithm, cycle starts whenever this algorithm reaches the RP_1 point. Afterward, collided nodes in CN are splitted by CRTE, which can be chosen by configuring the type_CRTE parameter. We consider three types of CRTE(type_CRTE) that are frequently incorporated in MAC protocols: the geometric_CRTE, the BEB_CRTE, and the Q-aryCRA_CRTE which are presented in detail in Section III-B, III-C, and III-D, respectively. Marked nodes are those who lose in previous cycle because their backoff intervals are not the shortest one and therefore, they will be promoted to a group that is g-level higher and enter the next cycle. In the original slotted ALOHA algorithm, this represents the situation that lost nodes decrease their backoff counter by g. New arrivals are assigned to groups according to their arrival time. Note that in the original slotted ALOHA, this operation is done implicitly, while a centralized and explicit way to do this is surely also applicable.

The algorithm then starts to process each group one by one until a transmission happens. To process a group #g with GP(gp_scheme), we mean to have the transmitter and the receiver complete a transmission. Figure 2 shows three possible types of GP(gp_scheme) including 2-way, 4-way handshaking, and polling (Note that they are all special cases of the procedure of reliable multiple access in [11]). Configuring the gp_scheme parameter means to change the way nodes interact with receivers (e.g., base stations). In ALOHA protocol, the 2-way handshaking is usually deployed. 4-way handshaking and polling will be discussed when we consider CSMA/CA and GRAP in Section IV and Section V, respectively.

The completeness in this algorithm represents whether the algorithm process every group in a cycle or not. If we choose geometric and BEB as the type_CRTE, the cycle repeats when an non-empty group is processed. While if the Q-ary collision resolution is chosen, the cycle repeats when all groups are completely processed. This is further explained in Section III-D when we consider the Q-ary collision resolution protocol.

If this transmission turns out to be a success, the cycle ends and the algorithm repeats. Otherwise, the algorithm collects collided nodes into set CN and similarly, the cycle ends and the algorithm repeats. Below we devote three sections to discuss the three types of CRTE we mentioned above.

B. Geometric Random Backoff

We can describe this backoff scheme in terms of MULCAR (tree expansion) by saying that each node with collided frame(s) is splitted into a group according to the geometric distribution. Thus a node joins group #1 with probability p, group #2 with probability $p(1-p)$, and so on. Below is the geometric_CRTE() process and please note that we consider here the *truncated* version geometric random backoff.

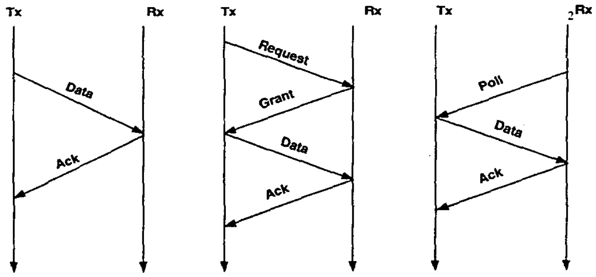


Fig. 2. Three different group process schemes.

```
geometric_CRTE(){
     for(each collided node){
        Choose a group #g to join
          according to the distribution:
        Pr{g=k}=p*(1-p)^k for 0<=k<max_delay
        Pr{g=k}=(1-p)^k   for k=max_delay}}
```

C. Binary Exponential Backoff

Similarly, we can see that BEB is another form of tree expansion in the language of MULCAR. Whenever the BEB_CRTE() (presented below) is invoked, collided nodes are splitted into different groups in order to resolve collisions.

```
BEB_CRTE(){
 for(each collided node){
  CW <- [((CW_min)+1)*
    2^((min(collision_count,
    max_delay))-1)]-1;
  Randomly choose a group #g to
    join according to an uniform
    distribution over the interval [0,CW];}}
```

D. Q-ary Collision Resolution

The forementioned geometric random backoff and BEB are both proved to be unstable in that when offered load is high, the channel is congested, delay tends to be unacceptable and the throughput drops significantly. Tree-splitting collision resolution techniques [3, 8] are thus developed to combat this vulnerability. We consider here the basic Q-ary collision resolution algorithm with free access proposed in [3]. The process is presented below.

```
Q-aryCRA_CRTE(Q,sets of collided nodes){
 for(each set s, s=[1,S]){
  Each node in set #s choose a
    group #g to join according
    to a uniform distribution over
    [(s-1)Q+1,sQ];}}
```

Up to now we present three possible forms of CRTE to split collided nodes into different groups. After any of these CRTE is invoked, the system is in the position to process each groups sequentially as the MULCAR_ALOHA algorithm shows.

IV. THE CSMA ALGORITHM

The Carrier Sense Multiple Access (CSMA) protocol [13] improves the original ALOHA protocol by being "polite" when there is any ongoing transmission. With the knowledge of MULCAR, the carrier sensing part of CSMA protocol is just a way to split upcoming transmissions to avoid collisions. We may anticipate the collision if we sense the channel busy. What to do next is to split these nodes into different subsets and make them transmit at different time, using different codes, and so on. In Section IV-A and IV-B, we dicuss two popular CSMA variations and the type of CATE they invoke. An algorithm unified the two is presented in Section IV-C as the MULCAR_CSMA algorithm.

A. The p-persistent CSMA Protocol

To put p-persistent CSMA in the form of CATE, nodes that sense a busy channel are put into the designated set "Deferred Nodes (DN)" and at the beginning of the next cycle, CATE splits these nodes into different groups according to geometric distribution. In fact, we utilize the geometric_CRTE() we discussed in Section III-B here to perform the geometric CATE.

B. The CSMA/CA Algorithm

The CSMA/CA algorithm is the foundation of the MAC protocol for the IEEE 802.11 standard for wireless Local Area Networks (LANs) [5]. The "Collision Avoidance (CA)" part in CSMA/CA performs the function of CATE in the protocol. In the IEEE 802.11 standard, the system resolves collisions using binary exponential backoff, each node has to maintain a "Contention Window (CW)" that doubles whenever its transmission collides with others'. A node in IEEE 802.11 uses the value of "CW" to randomly select a backoff interval to defer a transmission, which is the collision avoidance operation in the standard. Below is the BEB_CATE(), which performs the collision avoidance in IEEE 802.11 we just mentioned.

```
BEB_CATE(){
 for(each node in DN){
  randomly choose a group #g to join
  according to an uniform distribution
  over interval [0,CW];}}
```

C. The MULCAR_CSMA Algorithm

Equipped with the two CATE processes above, we are in the position to integrate the p-persistent CSMA and CSMA/CA into one algorithm, that is-the MULCAR_CSMA algorithm below.

The MULCAR_CSMA Algorithm

```
RP_1  //Renewal Point 1
if(memoryless_after_lost is set){
 have all nodes in DN call CATE(type_CATE);}
else{
 unmarked nodes in DN call CATE(type_CATE);
 associate marked nodes in DN to group
 number #(original group number-g);}
have unmarked nodes in CN call
 CRTE(type_CRTE);
associate marked nodes in CN to group
 number #(original group member-g);
set g=1; //start to process each group
RP_2
nodes with new arrival packets during
 the processing of group #(g-1) -> TX(g);
nodes in group #g -> TX(g);
process group g with GP(gp_scheme);
if(there is no transmission){
  increase g by 1;
  if(g<G){
    goto RP_1;}
  else{
    goto RP_2;}}
elseif(there is a transmission){
 nodes in group #(g+1)
  to group #(g+t) -> DN;
 //t is the duration of the transmission
 if(memoryless_after_lost is set){
   mark the loser in CN;}
 else{
   mark the loser in CN;}
 if(the transmission is a success){
   the successful node removes the
    transmitted packet from buffer;
   current cycle ends and goto RP_1;}
  else{collided nodes -> CN;
       current cycle ends and goto RP_1;}}
```

Basically this algorithm resembles the MULCAR_ALOHA algorithm except that at the beginning of a cycle, CATE is introduced to reduce anticipated collisions. We add a parameter "memoryless_after_lost" to configure the system and when it is set, the algorithm clears the associated group number of losers in previous cycle and have them join CATE again, which is the case in p-persistent CSMA. Please note that in the case when nodes are capable of detecting the ongoing transmission (e.g. in Ethernet), CSMA/CD is usually employed. In the above MULCAR_CSMA algorithm, this means that the group processing time is reduced to the collision detection time and a new cycle begins right after the collision detection is done.

V. THE GROUP RANDOMLY ADDRESSED POLLING (GRAP) PROTOCOL

In this section, we discuss another representative family of wireless MAC algorithm: the Randomly Addressed Polling (RAP) family [9]. Chen and Lee designed RAP with the idea that instead of polling every node in the network for transmssion, the Base Stations only poll those nodes with ready frames. Here we consider a typical example protocol in the family, the GRAP protocol (check the detailed operation in [10])

There are several differences between GRAP and ALOHA/CSMA we discussed previously.

- We note that in the GRAP above, collided nodes and contending nodes choose group to join "uniformly" (between 1 to G). We may thus design the uniform_CATE and uniform_CRTE to split nodes, just like what we do in CA/CRTE we proposed in previous sections.
- In GRAP, new arrivals during a cycle is not allowed to access the channel, which is termed as "blocked access" method. In ALOHA and CSMA protocols discussed previously, new arrivals may join the contention right after they are ready, and this is termed as "free access" method.
- After calling CA/CRTE, GRAP processes "all" groups, then the cycle ends. ALOHA or CSMA only processes the first non-empty group, then the cycle ends. This difference is the completeness we mentioned in Section III-D.

The MULCAR_GRAP Algorithm

```
RP_1
all new arrivals during previous cycle -> DN;
have all nodes in DN call uniform_CATE;
have all nodes in CN call uniform_CRTE;
all nodes report the results of CATE and
  CRTE back to BSs;
set g=1; //start to process each group
RP_2
nodes in group #g -> TX(g);
process group #g with GP(gp_scheme);
if(there is no transmission){
   g++;
   if(g>G){ //G is the group size in GRAP
        goto RP_1;}
   else{goto RP_2;}}
elseif(there is a transmission){
   if(the transmission is a success){
      the successful node removes the
       transmitted packet from buffer;
      current cycle ends and goto RP_1;}
   else{collided nodes -> CN;
        current cycle ends and goto RP_1;}}
```

As one can see, the blocked access method forces the system to collect new arrivals during the previous cycle into the set DN and have them transmit in the current cycle. The results of CA/CRTE are reported back to the Base Stations (BSs).

VI. THE VALIDNESS AND EQUIVALENCE

To further demonstrate analytically the validness of unification, We provide a general throughput formula by analyzing the unified algorithm directly (with assumptions to ease the analysis) and show how this general throughput can be reduced to throughputs of several algorithms. We consider the case of a MAC algorithm employs geometric CATE and CRTE, with free access method, the 2-way handshaking as the group process scheme, and both the completeness and memoryless after lost are disabled. Detailed anaylisis is presented in [12] due the limited space here. In short, we derive the throughput of the system, S, according to renewal theory. Thus S is the fraction of time that a successful transmission takes within a cycle time. It follows that,

$$S = \frac{\overline{U}}{\overline{T}} \tag{1}$$

where,

$$\overline{U} = \sum_{0 \leq i \leq M} \pi_i \cdot P_{succ}(i) \cdot L, \tag{2}$$

$$\begin{aligned} \overline{T} = & \sum_{0 \leq i \leq M} \{\text{E[Idle Period]+E[Transmission Period]}\} \\ = & \sum_{0 \leq i \leq M} \pi_i \{ \frac{1}{[1-(1-p)^i(1-\sigma^{M-i}]} + P_{succ}(i) \cdot T_{TX} \\ & + [1-P_{succ}(i)] \cdot T_{CD} \}. \end{aligned} \tag{3}$$

$\Pi = [\pi_0, \pi_1, \ldots, \pi_M]$ is the stationary probability distribution of the number of backlogged nodes at a specific time. $P_{succ}(i)$ represents the probability of successful transmission if the number of backlogged nodes at the beginning of the cycle is i. T_{TX} is the duration of the GP(gp_scheme) process if this transmission is a success. While the transmission turns out to be a collision, we use T_{CD} to denote the length instead. L is the length of a data frame. Since we are dealing with a slotted system, we assume that L is an integer multiple of one slot time defined by protocol designers. In addition, each node in idle state generates a new frame with probability σ within a slot and p denotes the parameter of the geometric distributioin in both geometric_CATE() and geometric_CRTE().

We can reduce Equation (3) to throughput of Slotted ALOHA (analyzed in [2]) by setting $T_{TX} = T_{CD}$ which both equal to one slot time (one frame legth). Similarly, we can get the throughput of p-persistent CSMA/CD (presented in [4]) by setting one slot time to the length of one-way propagation delay and $T_{TX} = L+1$ and T_{CD} to the necessary time to detect collision. This general throuhgput formula validates the idea of re-writting algorithms in a cycling fashion.

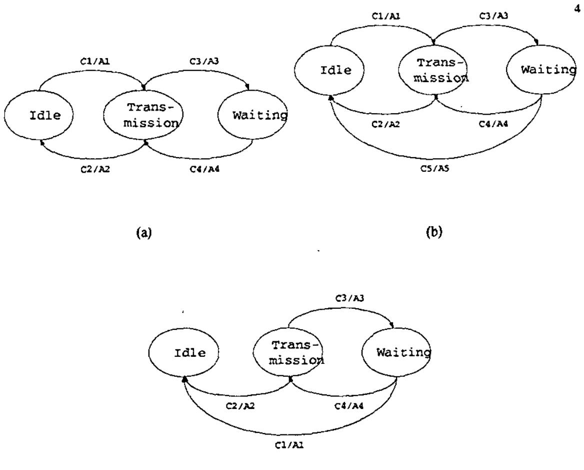
In addition to the validness, the equivalence between the re-written algorithm and its original one needs verificaiton. Those re-written algorithms are equivalent to the original corresponding algorithms in that one can map each operation that a node may take in one algorithm to an operation in another algorithm and the two operations have the the same effect. Here we show the equivalence between algorithms using the state transition diagrams of different algorithms. Only after the equivalence is shown, the integration of these re-written algorithms is correct and make sense.

The trasistion diagrams of both the slotted ALOHA protocol and the MULCAR_ALOHA protocol are of the same form as shown in Figure 3-(a). Figure 3-(b) and Figure 3-(c) illustrate the common transition diagram for CSMA and MULCAR_CSMA and GRAP and MULCAR_GRAP, respectively. Of course, the definitions of conditions (Cx in short) and actions (Ax in short) (Cx/Ax in Figure 3) are different for the corresponding equivalent algorithms, which is shown in Table I to III.

VII. THE UNIFICATION AND METHODOLOGY

A. The Unified Algorithm

Integrating the above three algorithms, we present the unified algorithm including ALOHA with three types of random backoff (geometric, binary exponential backoff, and Q-ary CRA), p-persistent CSMA, CSMA/CA, and GRAP protocols.



(a)

(b)

(c)

Fig. 3. The common state transition diagrams

TABLE I

CX AND AX FOR STATE TRANSITION IN ALOHA AND MULCAR_ALOHA

	CSMA	MULCAR_CSMA
C1	new frame arrives	new frame arrives
C2	transmission succeeds	transmission succeeds
C3	collision feedback	collision feedback
C4	backoff timer timeout	process of associated group starts
A1	transmit the frame	process with GP(gp_scheme)
A2	remove the transmitted frame	remove the transmitted frame
A3	choose a backoff delay	call CRTE()
A4	transmit the frame	process with GP(gp_scheme)

The Unified Algorithm

```
RP_1
if(access method=blocked){
 all new arrivals during previous cycle -> DN;}
if(memorless_after_lost is set){
 have all nodes in DN call CATE(type_CATE);}
else{
 unmarked nodes in DN call CATE(type_CATE);
 associate marked nodes in DN to group
  number #(original group number-g);}
unmarked nodes in CN call CRTE(type_CRTE);
associate marked nodes in CN to group
 number #(original group number-g);
if(report grouping result is set){
 all nodes report the grouping result back;}
set g=1; //start to process each group
RP_2
if(access method=free){
   nodes with new arrival packets during the
    processing of group #(g-1) -> TX(g);}
nodes in group #g -> TX(g);
process group #g with GP(gp_scheme);
if(there is no transmission){
 g++;
 if(G is set){ //G is the maximum TE size
   if(g>G){goto RP_1;}
   else{goto RP_2;}}
 else{goto RP_2;}}
elseif(there is a transmission){
 if(access method=free){
   nodes in group #(g+1)
    to group #(g+t) -> DN;}
   //t is the duration of the transmission
   if(the transmission is a success){
    the successful node removes the
```

TABLE II

CX AND AX FOR STATE TRANSITION IN CSMA AND MULCAR_CSMA

	CSMA	MULCAR_CSMA
C1	new frame arrives	new frame arrives
C2	transmission succeeds	transmission succeeds
C3	collision feedback	collision feedback
C4	backoff timer timeout	process of associated group starts
C5	sence the channel busy	collected to DN
A1	transmit the frame	process with GP(gp_scheme)
A2	remove the transmitted frame	remove the transmitted frame
A3	choose a backoff delay	call CRTE()
A4	transmit the frame	process with GP(gp_scheme)
A5	choose a backoff delay	call CATE()

TABLE III

CX AND AX FOR STATE TRANSITION IN GRAP AND MULCAR_GRAP

	GRAP	MULCAR_GRAP
C1	new frame arrives	new frame arrives
C2	transmission succeeds	transmission succeeds
C3	collisioin feedback	collision feedback
C4	backoff timer timeout	*process of associated group starts*
A1	wait for [READY] and choose a group to join	collected to DN and wait until new cycle begins and call CATE()
A2	remove the transmitted frame	remove the transmitted frame
A3	*choose a backoff delay*	*call CRTE()*
A4	transmit the frame	process with GP(gp_scheme)

```
  transmitted packet from buffer;
 if(completeness is set){
  g++;
  if(g>G){goto RP_1;}
  else{goto RP_2;}}
 else{
  if(memoryless_after_lost is set){
   mark the loser in CN;}
  else{mark the loser in CN and DN;}
       current cycle ends and goto RP_1;}}
else{
 collided nodes -> CN;
  if(completeness is set){
   g++;
   if(g>G){goto RP_1;}
   else{goto RP_2;}}
  else{
   if(memoryless_after_lost is set){
    mark the loser in CN;}
   else{mark the loser in CN and DN;}
        current cycle ends and goto RP_1;}}}
```

To configure the unified algorithm into the original protocols, we list the parameter setting in Table IV.

B. The Methodology of Unification

With the above trials presented in previous sections, one can see that our methodology of unifying MAC protocols generally consists several steps:

1. Recognize the type of CATE and CRTE each protocol uses by observing how contending and collided nodes are separated to avoid and resolve collisions. For example, identify the probability distribution used for splitting nodes.
2. Determine the way each protocol employs to process each group. Possible choices are 2-way or 4-way handshaking, polling, and so forth.
3. Find out under what kind of condition each protocol renews a cycle. A cycle may renew whenever there is a transmission occurs, or whenever all groups are processed.
4. Subtle differences between these protocols are then identified and are specified as parameters which control the flow of the unified algorithm. For instance, we defined the "access method" to decide whether new arrivals are permitted to access the channel in current cycle, "completeness" as the parameter that determines when a cycle renews, and the "memoryless after lost" to say whether or not the un-transmitted nodes remember their associated group number in previous cycles.

TABLE IV

PARAMETER CONFIGURATIONS FOR DIFFERENT WIRELESS MAC PROTOCOLS.

	ALOHA w/ geometric or binary *exponential backoff*	*Basic Q-ary* CRA	*p-persistent* CSMA	*CSMA/CA*	*GRAP*
slot time	one transmission *+one feedback*	one transmission *+one feedback*	one propagation *delay*	defined in *Spec.*	one propagation *delay*
access method	free	free	free	free	blocked
completeness	no	*no*	*no*	*no*	*yes*
memoryless after *lost*	no	yes	no	yes	no
report grouping result	no	no	no	no	yes
group process scheme	2-way handshaking	2-way handshaking	2-way handshaking	4-way handshaking	polling
type of CATE	none	none	geometric_CATE	BEB_CATE	uniform_CATE
type of CRTE	geometric_CRTE or BEB_CRTE	Q-aryCRA_CRTE	geometric_CRTE	BEB_CRTE	uniform_CRTE

Based on the cycling idea, the unification becomes straightforward and systematic.

VIII. CONCLUSION

This paper presents a piorneer trial of a unified algorithm for several MAC protocols for wireless access networks. The ALOHA protocol with geometric backoff, binary exponential backoff, and Q-ary CRA, the p-persistent CSMA, CSMA/CA, and the GRAP protocols are considered. We re-write these protocols on the basis of MULCAR, a powerful tool that models most existing MAC protocols. Those re-written algorithms are then integrated into one unified MAC algorithm that can be configured to operate as different MAC algorithms with proper parameter setting. We validate the unification model with an analytical example. The equivalence between the re-written protocol and the original one are shown with transition diagrams' operation. We have successfully designed a unified MAC algorithm that is the core and the basis of a software-defined MAC controller, which satisfies the need for a single hardware platform that can simultaneously support different and evolving protocols.

IX. ACKNOWLEDGEMENT

The authors would like to thank the support from National Science Council and Ministry of Education under Grant 89-E-FA06-2-2 and USENIX Association's Scholastic Committee.

REFERENCES

[1] R. G. Gallager. A perspective on multiaccess channels. *IEEE Trans. on Information Theory*, March 1985.

[2] L. Kleinrock and S. S. Lam. Packet switching in a multiaccess broadcast channel: performance evaluation. *IEEE Trans. on Communications*, April 1975.

[3] P. Mathys and P. Flajolet. Q-ary collision resolution algorithms in random-access systems with free or blocked channel access. *IEEE Trans. on Information Theory*, March 1985.

[4] F. A. Tobagi and V. B. Hunt. Performance analysis of carrier sense multiple access with collision detection. *Computer Networks*, 4:245-259, 1980.

[5] *Institute of Electrical and Electronics Engineers, Inc.. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications.* 1997

[6] K. C. Chen and Y. K. Sun. A general multi-layer collision resolution multiple access protocol for wireless networks. In *Proc. of VTC*, 1996.

[7] Y. K. Sun, K. C. Chen, and D. C. Twu. Generalized Tree Multiple Access Protocols in Packet Swtiching Networks. In *Proc. of PIMRC*, 1997.

[8] J. I. Capetanakis. Tree algorithm for packet broadcast channels. *IEEE Trans. on Information Theory*, pages 505-515, September 1979.

[9] K. C. Chen and C. H. Lee. RAP-a novel medium access control protocol for wireless data networks. In *Proc. of GLOBECOM*, 1993.

[10] C. S. Chang, K. C. Chen, M. Y. You, and J. F. Chang. Guaranteed quality-of-service wireless access to ATM networks. *IEEE Journal on Selected Areas in Communications*, January 1997.

[11] K. C. Chen. Medium access control of wireless LANs for mobile computing. *IEEE Network* Sep./Oct. 1994.

[12] C. M. Teng. A Unified Algorithm for Medium Access Control Protocols. Master Thesis, National Taiwan University, June 2001.

[13] L. Kleinrock and F. A. Tobagi. Packet switching in radio channels: Part I-carrier sense multiple-access modes and theri throughput delay characteristics. *IEEE Trans. on Communications*, 23(12):1400-1416, December 1975.