Distributed Optimal Contention Window Control for Elastic Traffic in Wireless LANs

Yaling Yang
Department of Computer Science
University of Illinois at Urbana-Champaign
Email: yyang8@uiuc.edu

Jun Wang
Department of Computer Science
University of Illinois at Urbana-Champaign
Email: junwang3@cs.uiuc.edu

Robin Kravets
Department of Computer Science
University of Illinois at Urbana-Champaign
Email: rhk@cs.uiuc.edu

***Abstract*— This paper presents a distributed contention window control control algorithm, GCA (General Contention window Adaptation), that achieves various bandwidth allocation policies and at same time efficient channel utilization. By modeling different bandwidth allocation policies as optimization problems of contention window assignment, we are able to design GCA and prove that it converges to the solution of the contention window assignment problem. By examining the stability properties of GCA, we identify the optimal stable point that maximizes channel utilization and provide solutions to control the stable point of GCA near the optimal point. Simulation results show that GCA achieves fairness and efficiency. Due to the generality of GCA, our work provides a theoretical foundation to analyze existing and design new contention window control algorithms.**

I. Introduction

Due to the shared nature of wireless communication channels and the intrinsic scarcity of bandwidth in wireless LANs, nodes must contend for the channel and compete for bandwidth. While both contention resolution and bandwidth allocation can be achieved through centralized scheduling at a wireless LAN access point, such centralized control is not scalable to a large number of nodes, suggesting the use of distributed algorithms for both contention resolution and bandwidth allocation. Distributed contention resolution in wireless LANs is quite common and can be achieved using various protocols [27], [15], [3], [11], [8]. In current distributed contention window based access protocols (e.g., IEEE 802.11 [27], MACA [15] and MACAW [3]), the contention window size is used to control the frequency at which a node accesses the channel to reduce contention in the network. However, contention window size also directly affects the share of bandwidth a node achieves during competition for the channel. Therefore, it is natural to extend such algorithms to support bandwidth allocation. The goal of such a combined algorithm is to allocate bandwidth to the nodes both "efficiently" and "fairly", where efficiency is defined by the level of bandwidth utilization and fairness is defined by the goals of the particular network. Given this observation, the goal of our research is to design a distributed contention window control algorithm to support fair bandwidth allocation and efficient channel utilization.

While efficiency can be defined in terms of the throughput of a network, fairness must be defined in the context of the requirements of the nodes using the network. For example, fairness may mean that every competing node in the network obtains the same bandwidth, that the share of bandwidth to a node is proportional to its priority, or that the highest priority node should obtain all of the bandwidth. Due to these different goals, it is desirable to support any definition of fairness. One major contribution of our research is bandwidth allocation with support for a large class of fairness definitions, including definitions used in current research.

There are two challenges to designing a fair and efficient distributed contention window control algorithm for wireless networks. First, the relationship between a node's bandwidth share and its contention window size is not straightforward. Second, the node's bandwidth share is also dependent on the contention window sizes of *all* competing nodes in the network. While a node can set its own contention window size, it has no control over the contention window sizes of other nodes. Additionally, by adjusting its own contention window size, a node directly affects other nodes' share of the bandwidth. Without careful design, adjusting contention window sizes at different nodes may result in an unstable system. Therefore, it is necessary to identify the properties of distributed contention window control algorithms that can guarantee fairness and stability.

Applications requiring bandwidth allocation can either have real time deadlines (i.e., streaming traffic) or be more tolerant of changes in service (i.e., elastic traffic [26]). While streaming traffic requires admission control to provide guarantees and ensure optimal bandwidth allocation, elastic traffic always has backlogged packets and adjusts the rate of the flows to fill the available bandwidth. Hence, competing flows with elastic traffic are more concerned about fairness and efficiency of bandwidth allocation. The focus of this paper is on supporting such elastic traffic using dynamic contention window control. Using contention window control for service guarantees for realtime traffic is beyond the scope of this paper and can be found in our technical report [29].

There have been extensive studies on contention window control in wireless LANs. However, none of these approaches can support both an arbitrary definition of fairness and efficient use of bandwidth. The first type of algorithm, including IEEE 802.11e [21] and [1], assigns different minimum contention window sizes to different types of nodes to achieve weighted fairness. However, since minimum contention window sizes are pre-configured and do not adapt to congestion, such approaches do not utilize the channel efficiently. The second type of algorithm, including AOB [5], MFS [18], [6] and [7], only focuses on efficient channel utilization in the context of uniform bandwidth allocation and the support for other definitions of fairness is limited. The third type of algorithm, including PFCR [24], tries to provide a more general definition of fairness by modeling fairness as an optimization problem of transmission rate allocation. However, the mapping between rate allocation and contention window adaptation in PFCR is only appropriate for a limited set of fairness definitions (See Section VIII-A.1). The final type of algorithm, P-MAC [25], tries to achieve both proportional fairness and efficient utilization by estimating the contention windows used by the competing nodes. Such estimation requires that every node, with or without packets for transmission, must start P-MAC simultaneously and calculate the contention window sizes for all other nodes synchronously. Nodes with outdated contention window sizes of the other nodes due to asynchronous starting time or temporary failure may cause the algorithm to fail.

Due to the limitations of existing approaches, we propose our distributed contention window control algorithm, called GCA (**G**eneral **C**ontention window **A**daptation), which can be used to achieve optimal bandwidth allocation for competing wireless nodes in terms of efficient channel utilization and various definitions of fairness. GCA is a fully distributed algorithm, where each node adjusts its contention window size individually based on purely local information. Additionally, GCA makes no assumptions about frame size. The goal of GCA is to provide a general solution for designing dynamic contention window control algorithms in wireless LANs. To our knowledge, GCA is the first such algorithm to be proved convergent and stable.

There are four major contributions of the research presented in this paper. First, we identify and model, for the first time, a variety of fair bandwidth allocation problems as an optimization problem for contention window assignment. Second, to solve this optimization problem, we present the design of GCA, a fully distributed contention window control algorithm. Additionally, we rigorously prove that GCA converges to the exact solution of the optimization problem, meaning that GCA can achieve fairness for any given fairness definition. Third, by studying the properties of the stable point of GCA, we show that efficient channel utilization can also be achieved. Therefore, by controlling this stable point, GCA can achieve both channel efficiency and fairness. We validate this claim through simulations of GCA with two different fairness definitions. Finally, we demonstrate that GCA provides a systematic scheme to generalize and evaluate related approaches by showing that many existing heuristic-based algorithms for dynamic contention window control can be analyzed by the GCA approach.

This paper is organized as follows. Section II discusses bandwidth allocation in wired and wireless networks. Section III reviews contention window based distributed medium access control in wireless networks and the relationship between bandwidth allocation and contention window size is established in Section IV. Section V presents contention window control as an optimal contention window assignment problem. Section V-B introduces our contention window control algorithm, GCA, that can be used to solve the optimization problem and Section VII discusses guidelines for implementing GCA. In Section VIII, we analyze several existing dynamic contention window algorithms. Section IX presents the evaluation of GCA. Finally, Section X concludes and discusses future research.

II. Bandwidth Allocation

Bandwidth allocation in wired networks has been researched for many years. Since wired networks are assumed to be either point-to-point and/or have high link bandwidth, competition for bandwidth happens at the transport layer where flows compete for buffer space in routers. Essentially, end hosts modulate their TCP congestion window size to achieve distributed control [23], [28], [19], [16], [12]. The theoretical foundation of this competition is work by Kelly, et al. [16], which describes a distributed rate control algorithm to solve the optimal bandwidth allocation problem. Given the above assumptions about wired networks, the sending rate of a node is essentially its own TCP congestion window size over its own round trip time. Therefore, Kelly's rate control algorithm can be directly used to design optimal congestion window control algorithms for TCP.

The shared nature and limited bandwidth of wireless links moves the competition for bandwidth from router queues to channel access time, limiting the direct application of research results in wired networks to wireless networks. In wired networks, the relationship between the sending rate and the congestion window size are simple, since sending rate is congestion window size over round trip time. Since each node has direct control over its sending rate, Kelly's rate control algorithm [16] can be directly applied to *congestion* window control. However, in wireless networks, the sending rate of a node is dependent on the contention window sizes of *all* competing nodes, and so no node has direct control over its sending rate. Therefore, the same rate control algorithm can not be directly applied to *contention* window control.

The problem of supporting efficient channel utilization and fair bandwidth allocation is currently being researched by the wireless networking community. However, most approaches sacrifice one goal to achieve the other.

The first type of approach focuses on fairness in terms of service differentiation, such as IEEE 802.11e [1], [21], assign different minimum contention window sizes to different types of nodes to achieve weighted fairness among nodes. However, since minimum contention window sizes are preconfigured and do not adapt to congestion, such approaches do not utilize the channel efficiently. Additionally, these approaches are limited to only one definition of fairness.

The second type of approach uses dynamic contention window control algorithms to achieve maximum channel utilization for networks with no service differentiation (i.e., competing nodes share the bandwidth equally) [5], [6], [7]. Additionally, these approaches assume the same average packet size at every node. The support for other definitions of fairness is limited in these approaches and the fairness that they target may be broken if packet sizes are different.

The third type of approach P-MAC [25] proposes a distributed contention window control algorithm to achieve weighted proportional bandwidth allocation given that the packet sizes of all nodes are the same. However, this algorithm requires that the number of traffic classes and their associated weights are preconfigured into all competing nodes. The algorithm also requires that every node, with or without packets for transmission, must calculate the most recent contention window sizes for all other nodes continuously and synchronously . Nodes with outdated contention window sizes for the other nodes, due to failures or interference, may cause the algorithm to fail to achieve fair bandwidth allocation. Essentially, this algorithm is very sensative to disturbence and is not a stable algorithm.

The final approach, PFCR [24], tries to provide a more general definition of fairness by modeling fairness as an optimization problem of transmission rate allocation. Then it tries to use the Kelly's rate control algorithm [16] to adjust the contention window of wireless networks. Unfortunately, this mapping between rate control and contention window adaptation is only appropriate for a limited set of fairness definitions.

Based on the limitations of existing approaches, we conclude that a distributed contention window control algorithm should satisfy three requirements. First, it should be flexible so that it can be configured to achieve arbitrary definitions of fairness. Second, it should utilize the channel efficiently. Third, it should not require synchronized calculations for nodes in the network and should be a stable algorithm. In the remainder of this paper, we present GCA (**G**eneral **C**ontention window **A**daptation) and show how it satisfies these requirements.

III. Distributed Contention Window Based Channel Access Control

Distributed contention window based channel access is a common approach for MAC protocols in wireless networks, the most widely used of which is IEEE 802.11 DCF mode. Although the design and validity of GCA is not dependent on any one MAC protocol, we present IEEE 802.11 DCF as an example protocol to set the stage for presenting GCA.

IEEE 802.11 DCF mode mediates between competing nodes using Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). If the medium is idle at the time of transmission, nodes can send immediately. However, when the medium is busy, nodes uses a random backoff procedure to resolve contention conflicts with other competing nodes. RTS and CTS messages provide medium reservation prior to data transmission, as well as fast collision resolution. Nodes may bypass RTS-CTS transmission using a DATA-ACK handshake.

Before transmission, a node must determine whether the medium is busy or idle. If the medium remains idle for *DCF inter-frame space* (DIFS) time units, the node can transmit. If the medium was initially busy or changed from idle to busy during the DIFS, the node must defer its transmission. The first part of the deferment period is determined by the success of the last transmission. If the last frame was successful, the node waits DIFS time units. If the last frame was not successful, the node waits *extended inter-frame space* (EIFS) time units. The second part of the deferment period is determined by a random backoff timer, calculated by the following equation:

$$\textit{Backoff Time} = \textit{Random()} \times \textit{aSlotTime}, \tag{1}$$

where $Random()$ is a pseudo-random integer uniformly distributed over [0,CW]. The *contention window*, CW, is an integer in the range [*minimum contention window* (CW^{min}), *maximum contention window* (CW^{max})]. After a successful transmission, CW is set to CW^{min}. After an unsuccessful transmission, CW is doubled, up to CW^{max}. When the timer expires, the node can transmit an RTS message. When the medium is idle, the backoff timer is decremented by $aSlotTime$, determined by the physical layer. However, the timer is stopped when the medium is busy and restarted after the medium is idle for a DIFS.

IV. Bandwidth Allocation and Contention Window Size

To realize fair bandwidth allocation by adapting the contention window size, it is essential to understand the relationship between a node's bandwidth allocation and its contention window size. The relationship can be found in two steps. The first step is to find the relationship between Node i's bandwidth allocation and its contention window size, assuming no exponential increase of the contention window size after a collision. Second, the analysis is extended to consider the exponential backoff algorithm used in IEEE 802.11, where the contention window size is doubled after each collision. For notation convenience, we use W_i for the contention window size of node i and W_i^{min} as the minimum contention window size for Node i when exponential backoff is used. Notation for the entire paper can be found in Appendix B.

The analysis in this section is similar to some existing work [4], [20], [10] and is presented here for completeness and later reference. We use this analysis to show that the relationship between Node i's bandwidth allocation and W_i is approximately the same as the relationship between its bandwidth allocation and W_i^{min} when exponential backoff is used. Therefore, any algorithm that successfully allocates bandwidth by controlling W_i can also be used to control W_i^{min}. While this relationship is only an approximation, we validate the accuracy of this approximation through simulation in Section IX. Additionally, in Section VII, we show that the proper form of GCA controls congestion in the network. Since the purpose of exponential increase is to control congestion, GCA actually eliminates the need for such exponential increase.

A. Channel Model

The channel model used to analyze the relationship between a node's bandwidth allocation and W_i was developed in [4]. This model assumes that every competing node can hear each other. Since GCA is designed for WLANs, this assumption is valid for GCA. The extension of GCA to more complex environments is future work. In this model, real time is divided into *virtual time slots*, where a node decrements its backoff timer

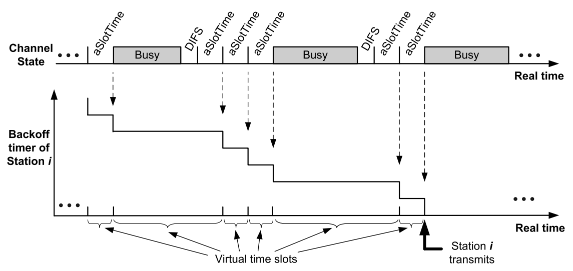


Fig. 1. Virtual time slots

once per virtual time slot and at most one packet can be transmitted in a virtual time slot. To visualize this model, Figure 1 shows the channel state and Node i's backoff timer. In this example, there are two types of virtual time slots. When the channel is idle, a virtual time slot is exactly $aSlotTime$ (e.g., Node i's first virtual time slot). When the channel is busy during the countdown of the backoff timer, a virtual time slot includes a busy period (e.g., Node i's second virtual time slot), but only decrements the backoff timer once. Such a virtual time slot extends from the start of the busy period until the end of the $aSlotTime$ period, since the backoff timer is not decremented until after the channel becomes idle for a DIFS period and the $aSlotTime$ period. This mapping of real time into virtual time slots allows the backoff process of a node to be modeled as a discrete Markov process (see [4] for details).

B. Bandwidth Allocation vs. Contention Window Size

For the set of transmitting nodes, $\mathcal{N}$, the absolute bandwidth allocated to Node i in bits per second, s_i, can be directly determined from the fraction of the total network capacity allocated to Node i, x_i. The general relationship between s_i and x_i can be stated in terms of P_i, the probability that Node i successfully transmits in a virtual time slot, and L_i, the channel bandwidth consumed by a successful packet transmission at Node i, as follows:

Theorem 1:

$$x_i = \frac{s_i}{\sum_{j\in\mathcal{N}} s_j} = \frac{P_i L_i}{\sum_{j\in\mathcal{N}} P_j L_j}$$

Proof: Since all nodes are assumed to carry elastic traffic and always have backlogged packets, the combined transmissions of all nodes consumes the entire network capacity, C. When the total throughput of all nodes achieves the capacity of the network, the following relationships hold:

$$\sum_{i\in\mathcal{N}} s_i = C, \tag{2}$$

$$x_i = \frac{s_i}{C}, \tag{3}$$

$$\sum_{i\in\mathcal{N}} x_i = 1. \tag{4}$$

For a period of real time, t, with m virtual time slots, Node i's expected number of sent packets is mP_i and Node i's expected throughput is $s_i = mP_iL_i$. Therefore, $s_i/\sum_{j\in\mathcal{N}} s_j = mP_iL_i/(\sum_{j\in\mathcal{N}} mP_jL_j)$. Canceling out m, $\frac{s_i}{\sum_{j\in\mathcal{N}} s_j} = \frac{P_iL_i}{\sum_{j\in\mathcal{N}} P_jL_j}$. Since x_i is defined as s_i/C and $\sum_{i\in\mathcal{N}} s_i = C$ according to Equations (2) and (3), $\frac{P_iL_i}{\sum_{j\in\mathcal{N}} P_jL_j} = x_i$. ∎

The following two subsections show the relationship between P_i and the contention window size, W_i, for all nodes. By linking P_i and W_i, Theorem 1 can be used to determine the relationship between W_i and s_i or x_i. We first discuss the case with no exponential increase after a collision and then relax this assumption and discuss the case with exponential increase.

1) Fixed Contention Window Sizes: Theorem 1 defines the relationship between x_i and P_i. Next, the relationship between P_i and W_i is derived so x_i can be expressed in terms of W_i.

Theorem 2: If W_i does not change after a collision,

$$x_i = \frac{P_iL_i}{\sum_{k\in\mathcal{N}} P_kL_k} = \frac{\frac{L_i}{W_i}}{\sum_{k\in\mathcal{N}} \frac{L_k}{W_k}}. \tag{5}$$

Proof: Let τ_i denote the probability that Node i transmits in a randomly chosen virtual time slot. When multiple nodes try to transmit in the same virtual time slot, a collision occurs and all transmissions fail. Therefore, the successful transmission probability, P_i, equals the probability that only Node i transmits in a virtual time slot [4]. Therefore,

$$P_i = \tau_i \prod_{j\neq i, j\in\mathcal{N}} (1-\tau_j) = \frac{\tau_i \prod_{j=1}^{n}(1-\tau_j)}{1-\tau_i}. \tag{6}$$

If W_i does not change after a collision, τ_i can be expressed as [4], [13]:

$$\tau_i = \frac{1}{W_i/2+1}. \tag{7}$$

Therefore, combining Equations (7) and (6) results in:

$$\frac{P_i}{P_j} = \frac{W_j}{W_i}. \tag{8}$$

Finally, combining Equation (8) with Theorem 1 results in:

$$x_i = \frac{P_iL_i}{\sum_{k\in\mathcal{N}} P_kL_k} = \frac{\frac{L_i}{W_i}}{\sum_{k\in\mathcal{N}} \frac{L_k}{W_k}}.$$

∎

Theorem 2 shows that a node's allocated bandwidth is determined by competing nodes' contention window sizes. Therefore, adaptation of the competing nodes' contention window sizes can achieve fair bandwidth allocation. Since Node i has no control over the contention window sizes of all other nodes, it is very important that the design of a contention window control algorithm ensures the stability of the system, even if each node adapts its own contention window size independently.

2) Exponentially Increasing Contention Window Sizes: In this section, IEEE 802.11-type protocols are considered, where the contention window size is doubled after a collision. Let m_i be the number of unsuccessful transmissions needed for the contention window size of Node i to reach W_i^{max}. The relationship between Node i's bandwidth allocation, s_i, and its minimum contention window size, W_i^{min}, can be expressed in the following theorem [20]:

Theorem 3:

$$x_i = \frac{P_i L_i}{\sum_{k\in\mathcal{N}} P_k L_k} \approx \frac{\frac{L_i}{W_i^{min}}}{\sum_{k\in\mathcal{N}} \frac{L_k}{W_k^{min}}}. \tag{9}$$

when the following two conditions are satisfied:

1) $W_i^{min} \gg 1, \forall i \in \mathcal{N}$, (e.g. $W_i^{min} = 16, 32, 64$ in the IEEE 802.11 standard),
2) $m_i \approx m_j, \forall i, j \in \mathcal{N}$.

Proof: The detailed proof in [20] shows that if $W_i \gg 1$, the probability of collision is approximately the same for all nodes. Therefore, assuming $m_i \approx m_j, \forall i, j \in \mathcal{N}$ and using the results developed in [4], the proof in [20] shows that,

$$x_i = \frac{P_i L_i}{\sum_{k\in\mathcal{N}} P_k L_k} \approx \frac{\frac{L_i}{W_i^{min}}}{\sum_{k\in\mathcal{N}} \frac{L_k}{W_k^{min}}}.$$

■

Since the relationship between x_i and W_i^{min} (Theorem 2) is approximately the same as the relationship between x_i and W_i (Theorem 1), in the rest of this paper, we design our dynamic contention window control algorithm assuming no exponential increase of the contention window size after a collision. Our algorithm can also be used to dynamically adjust the minimum contention window size when exponential increase is used, which we validate through simulation (see Section IX).

V. General Contention Window Adaptation Algorithm (GCA)

In this section, we present GCA, our distributed contention window control algorithm that achieves fair bandwidth allocation for any arbitrary definition of fairness. We first formulate the general fairness requirement as an optimization problem for contention window assignment. After the presentation of GCA, we prove the convergence and stability of GCA. Finally, the stable point for GCA is shown to provide fair bandwidth allocation.

A. Fairness Formulation

To design a general contention window control algorithm that supports various fairness definitions, it is necessary to identify a general formulation of fairness and translate it to a contention window assignment that achieves this fairness definition. In wired networks, fair bandwidth allocation has been modeled as an optimization problem [16], [22], [23]. We translate this optimization problem into a contention window assignment problem for wireless networks using the relationship between a node's allocated bandwidth and its contention window size. We provide solutions to the contention window assignment problem, which are window sizes that achieve fair bandwidth allocation.

The fairness of bandwidth allocation in wired networks can be formulated as an optimization problem for bandwidth allocation [16], [22], [23]. Essentially, each Node i has a utility of $U_i(s_i)$ when its bandwidth allocation is s_i. Given elastic traffic, the utility functions follow the following assumption [26]:

Assumption 1: For each i, $U_i(s_i)$ is an increasing, strictly concave and continuously differentiable function of s_i over the range $s_i \geq 0$.

Assuming that the channel capacity is C, the optimal bandwidth allocation problem can be formulated as [16], [22], [23]:

$OPT\_BW(U, C)$:

$$\max \textstyle\sum_{i\in\mathcal{N}} U_i(s_i)$$

subject to

$$\textstyle\sum_{i\in\mathcal{N}} s_i \leq C \text{ and } s_i \geq 0 \text{ for } i \in \mathcal{N}.$$

According to the Karush-Kuhn-Tucker optimality condition [2], it has been shown that the unique solution to the above OPT_BW problem is given by [16]:

$$U_i'(s_i) = \mu, \quad \text{for } i \in \mathcal{N} \tag{10}$$

$$\mu(C - \sum_{i\in\mathcal{N}} s_i) = 0, \tag{11}$$

$$\mu \geq 0, \tag{12}$$

where μ is the Lagrange multiplier.

GCA is designed around a general definition of utility that adheres to the above assumption. However, different definitions of utility result in different solutions to $OPT\_BW(U, C)$ and so achieve different definitions of fairness [22], [23]. In Section VII-B, we discuss the choice of the specific utility function and show some examples of how specific utility functions achieve specific definitions of fairness.

We now present the translation of $OPT\_BW(U, C)$ into a contention window assignment problem. First, the function $U_i(s_i)$ is mapped to a function of x_i by substituting $U_i(s_i)$ with $\tilde{U}_i(x_i)$, where $U_i(s_i) = U_i(x_i C) = \tilde{U}_i(x_i)$. Similar to $U_i(s_i)$, $\tilde{U}_i(x_i)$ is an increasing, strictly concave and continuously differentiable function of x_i over the range $x_i \geq 0$. Second, using the relationship between x_i and W_i expressed in Theorem 1 to replace x_i, we can translate $OPT\_BW(U, C)$ into a contention window allocation problem, $OPT\_WIN(\tilde{U}, C)$, as follows:

$OPT\_WIN(\tilde{U}, C)$:

$$\max \textstyle\sum_{i\in\mathcal{N}} \tilde{U}_i\left(\frac{\frac{L_i}{W_i}}{\sum_{k\in\mathcal{N}} \frac{L_k}{W_k}}\right)$$

over

$$W_i \geq 1 \text{ for } i \in \mathcal{N}.$$

Using the relationship between s_i, x_i and W_i expressed in Theorem 1 to replace s_i in Equations (10), (11) and (12), solutions to $OPT\_WIN(\tilde{U}, C)$ satisfy:

$$\tilde{U}_i'\left(\frac{\frac{L_i}{W_i}}{\sum_{k\in\mathcal{N}} \frac{L_k}{W_k}}\right) = \mu, \quad \text{for } i \in \mathcal{N} \tag{13}$$

$$\mu(1 - \sum_{i\in\mathcal{N}} \frac{\frac{L_i}{W_i}}{\sum_{k\in\mathcal{N}} \frac{L_k}{W_k}}) = 0, \tag{14}$$

$$\mu \geq 0. \tag{15}$$

Since $(1 - \sum_{i\in\mathcal{N}} \frac{\frac{L_i}{W_i}}{\sum_{k\in\mathcal{N}} \frac{L_k}{W_k}}) = 0$, Equation (14) is always satisfied and can be omitted. Therefore, solutions to

$OPT\_WIN(\tilde{U}, C)$ satisfy:

$$\boxed{\begin{array}{rl} \tilde{U}_i'\left(\frac{L_i/W_i}{\sum_{k\in\mathcal{N}} L_k/W_k}\right) = & \mu, \text{ for } \forall i \in \mathcal{N} \\ & \mu \geq 0. \end{array}} \tag{16}$$

Note that in OPT_WIN, we assume that each node has one utility function. Therefore, solutions to OPT_WIN achieve per-node fairness. We can build a similar model for per-flow utility functions to achieve per-flow fairness. However, solutions to OPT_WIN may require that flows in the same node use different W_i's. To implement per-flow fairness, every flow must keep a record of its own W_i and some scheduling algorithm must be implemented to ensure that fairness is not compromised by head-of-line blocking. Such an approach is taken in the design of IEEE 802.11e. For simplicity of presentation, we focus on fairness between nodes. However, our results can be used for per-flow fairness, assuming per-flow W_i state and per-flow scheduling.

It is very important to note that, although the solution to $OPT\_WIN(\tilde{U}, C)$ is unique in terms of $x_i = \frac{L_i/W_i}{\sum_{k\in\mathcal{N}} L_k/W_k}$, it is not unique in terms of W_i. Consider contention window sizes $\mathbf{W} = \{W_i : i \in \mathcal{N}\}$ that solve $OPT\_WIN(\tilde{U}, C)$. When $\mathbf{W}$ is multiplied by a constant factor a, the resulting contention window assignment $a\mathbf{W} = \{aW_i : i \in \mathcal{N}\}$ is also a solution to $OPT\_WIN(\tilde{U}, C)$. Among the possible solutions to $OPT\_WIN(\tilde{U}, C)$ that satisfy the fairness requirement, channel utilization can be quite different. Therefore, the identification of the solution of $OPT\_WIN(\tilde{U}, C)$ that maximizes channel utilization is important and is discussed in Section VI-A.

B. The Design of the GCA Algorithm

GCA is used to control the contention window size of all nodes in the network to achieve fair and efficient bandwidth allocation. The design of GCA is fully distributed and so each node need only collect local information and adjust its contention window size accordingly. In addition, unlike previous work on dynamic contention window control [5], [6], [7], [25], GCA can be used in a network where nodes have different average packet sizes.

In GCA, a Node i adapts its W_i according to the following differential equation:

$$\boxed{\dot{W}_i(t) = -\alpha W_i(t)[\tilde{U}_i'(x_i) - f(\mathbf{W}, \mathbf{L})],} \tag{17}$$

where α is a positive constant factor, $\dot{W}_i(t)$ is the time derivative of W_i, $\mathbf{W} = \{W_i : i \in \mathcal{N}\}$ and $\mathbf{L} = \{L_i : i \in \mathcal{N}\}$. $f(\mathbf{W}, \mathbf{L})$ is a function of some observable state of the channel.

Various choices for channel state characteristics can be used in $f(\mathbf{W}, \mathbf{L})$ and the corresponding $f(\mathbf{W}, \mathbf{L})$ can have many different forms. However, GCA does make two assumptions about $f(\mathbf{W}, \mathbf{L})$. First, the channel state that $f(\mathbf{W}, \mathbf{L})$ uses must be observable by all nodes sharing the channel and all nodes must be configured with the same $f(\mathbf{W}, \mathbf{L})$. Since in the networks targeted by GCA every node can hear each other and hence see the same channel state, the first assumption is not very restrictive. Second, to guarantee system stabilization at a unique point (See details in Section V-C), $f(\mathbf{W}, \mathbf{L})$ must be strictly increasing with respect to $\sum_{i\in\mathcal{N}} \frac{L_i}{W_i}$ inside a certain set of system states. Since there are many channel states that depend on the window sizes and packet lengths of all nodes (e.g., packet transmission delay, average length of an idle period or collision probability), a $f(\mathbf{W}, \mathbf{L})$ that uses any of these channel states is by definition a function of $\mathbf{W}$ and $\mathbf{L}$. By choosing the right form of the function for these channel states, $f(\mathbf{W}, \mathbf{L})$ can easily meet the second assumption. As long as $f(\mathbf{W}, \mathbf{L})$ satisfies these assumptions, GCA is not limited to any specific observed channel state or type of function for $f(\mathbf{W}, \mathbf{L})$. We demonstrate in Section VIII that these assumptions are not too restrictive and GCA can be used to model current dynamic contention window control algorithms.

To implement GCA in real system, the update algorithm in Equation (17) must be translated into to its discrete counterpart. Therefore, according to Taylor Series expansion, Equation (17) can be expanded over a small time interval ϵ as follows:

$$\begin{aligned} W_i(t+\epsilon) &= W_i(t) + \dot{W}_i(t)\epsilon + o(\epsilon), \\ &\approx W_i(t) - \alpha W_i(t)[\tilde{U}_i'(x_i(t)) - f(\mathbf{W}(t), \mathbf{L})]\epsilon, \end{aligned}$$

where $o(\epsilon)$ represents the second and higher order terms of ϵ. Therefore, if a node updates its contention window size at every virtual time slot, the discrete version of GCA can be expressed as the following iterative algorithm:

$$W_i^{k+1} = W_i^k - \alpha W_i^k[\tilde{U}_i'(x_i^k) - f(\mathbf{W}^k, \mathbf{L})]. \tag{18}$$

If a node only performs the window size update for each packet transmission, which means that the average number of slots between each update is $\frac{W_i}{2}$, the discrete version of GCA becomes:

$$W_i^{k+1} = W_i^k - 0.5\alpha(W_i^k)^2[\tilde{U}_i'(x_i^k) - f(\mathbf{W}^k, \mathbf{L})]. \tag{19}$$

The GCA algorithm itself is simple and only requires local information about the state of the network. Despite this simplicity, GCA converges to the solution of OPT_WIN, and so also achieves efficient allocation for any arbitrary fairness definition. In the remainder of this section, we present the proof of GCA's convergence to OPT_WIN using control theory [17].

C. The Convergence and Fairness of GCA

In this section, we prove that GCA, as expressed in Equation (17), asymptotically converges to a unique point that is a solution to OPT_WIN given the two assumptions about $f(\mathbf{W}, \mathbf{L})$. Our proof includes two theorems. Theorem 4 states that regardless of the specific form of $f(\mathbf{W}, \mathbf{L})$ (as long as every node observes the same $f(\mathbf{W}, \mathbf{L})$), GCA converges to an *invariant set* [17] where each element of the set is a solution to OPT_WIN. The second theorem takes $f(\mathbf{W}, \mathbf{L})$ into consideration and shows that given the two assumptions for $f(\mathbf{W}, \mathbf{L})$, GCA converges to a unique point that solves OPT_WIN.

Theorem 4: Starting from any initial state of $W_i(0) > 0$, the system described in Equation (17) converges to an invariant set $\Gamma = \{\mathbf{W} : \frac{\dot{W}_i}{W_i} = \frac{\dot{W}_j}{W_j}, \text{ for } \forall i, j \in \mathcal{N}\}$ and every element in Γ is a solution to OPT_WIN. In addition, inside Γ, the $\tilde{U}_i'(x_i)$

in Equation (17) remains a constant; i.e., $\tilde{U}_i'(\frac{L_i/W_i}{\sum_{k\in\mathcal{N}} L_k/W_k}) = \tilde{U}_j'(\frac{L_j/W_j}{\sum_{k\in\mathcal{N}} L_k/W_k})$, for $\forall i, j \in \mathcal{N}$.

Proof: The proof consists of four steps. At step one, for notation simplicity, we translate the update algorithm of W_i to the update algorithm of its reciprocal $Z_i = \frac{1}{W_i}$. At step two, we find a Lyapunov function $V(\cdot)$ of the system and prove that $\dot{V} \geq 0$ [1]. At step three, we prove that the set of points that satisfy $\dot{V} = 0$ is an invariant set. Therefore, using the La Salle Invariant Set Principle [17], we conclude that GCA converges to this invariant set. At step four, we prove that every point in this invariant set is a solution to OPT_WIN.

Step 1: Let $Z_i = 1/W_i, \forall i \in \mathcal{N}$. By Theorem 2 and replacing W_i with Z_i, the update algorithm in Equation (17) is:

$$\begin{aligned}\dot{W}_i &= -\alpha W_i(t)\left[\tilde{U}_i'(\frac{\frac{L_i}{W_i}}{\sum_{k\in\mathcal{N}}\frac{L_k}{W_k}}) - f(\mathbf{W},\mathbf{L})\right], \quad (20)\\ &= -\frac{\alpha}{Z_i(t)}\left[\tilde{U}_i'(\frac{Z_iL_i}{\sum_{k\in\mathcal{N}} Z_kL_k}) - f(\mathbf{Z},\mathbf{L})\right], \quad (21)\end{aligned}$$

where $\mathbf{Z} = \{Z_i : i \in \mathcal{N}\}$. Note that $\dot{Z}_i$ has the following relationship to $\dot{W}_i$:

$$\dot{Z}_i = \frac{d\frac{1}{W_i}}{dt} = -\frac{1}{(W_i)^2}\dot{W}_i = -Z_i^2\dot{W}_i.$$

After replacing $\dot{W}_i$ by $\dot{Z}_i$, the algorithm expressed in Equation (17) is equivalent to:

$$\dot{Z}_i = \alpha Z_i\left[\tilde{U}_i'(\frac{Z_iL_i}{\sum_{k\in\mathcal{N}} Z_kL_k}) - f(\mathbf{Z},\mathbf{L})\right]. \quad (22)$$

Step 2: In this step, we define a scalar function $V(\mathbf{Z})$ as:

$$V(\mathbf{Z}) = \sum_{i\in\mathcal{N}} \tilde{U}_i(\frac{Z_iL_i}{\sum_{k\in\mathcal{N}} Z_kL_k}). \quad (23)$$

According to Lemma 1 in Appendix A, V is a Lyapunov function with $\dot{V} \geq 0$. The zero values of $\dot{V}$ are obtained for the set:

$$R = \left\{\mathbf{Z} : \frac{\dot{Z}_i}{Z_i} = \frac{\dot{Z}_j}{Z_j}, \text{ for } \forall i, j \in \mathcal{N}\right\}. \quad (24)$$

Step 3: From Equation (22),

$$\frac{\dot{Z}_i}{Z_i} = \alpha\left[\tilde{U}_i'(\frac{Z_iL_i}{\sum_{k\in\mathcal{N}} Z_kL_k}) - f(\mathbf{Z},\mathbf{L})\right].$$

Therefore, the equality condition in Equation (24) is equivalent to:

$$\tilde{U}_i'(\frac{Z_iL_i}{\sum_{k\in\mathcal{N}} Z_kL_k}) = \tilde{U}_j'(\frac{Z_jL_j}{\sum_{k\in\mathcal{N}} Z_kL_k}), \text{ for } \forall i, j \in \mathcal{N}. \quad (25)$$

Hence, R can also be defined as:

$$R = \left\{\mathbf{Z} : \tilde{U}_i'(\frac{Z_iL_i}{\sum_{k\in\mathcal{N}} Z_kL_k}) = \tilde{U}_j'(\frac{Z_jL_j}{\sum_{k\in\mathcal{N}} Z_kL_k}), \forall i, j \in \mathcal{N}\right\}. \quad (26)$$

[1]Note that we are discussing a maximization problem. Therefore, the convergence condition is $\dot{V} \geq 0$ [16], [28].

The following proof shows that R is an invariant set of the system described in Equation (22). First, note that:

$$\begin{aligned}\frac{d}{dt}\tilde{U}_i'(\frac{Z_iL_i}{\sum_{k\in\mathcal{N}} Z_kL_k}) &= \sum_{j\in\mathcal{N}}\left[\frac{\partial}{\partial Z_j}\tilde{U}_i'(\frac{Z_iL_i}{\sum_{k\in\mathcal{N}} Z_kL_k})\right]\dot{Z}_j,\\ &= [\frac{\partial}{\partial Z_i}\tilde{U}_i'(\frac{Z_iL_i}{\sum_{k\in\mathcal{N}} Z_kL_k})]\dot{Z}_i +\\ &\quad \sum_{j\in\mathcal{N},j\neq i}[\frac{\partial}{\partial Z_j}\tilde{U}_i'(\frac{Z_iL_i}{\sum_{k\in\mathcal{N}} Z_kL_k})]\dot{Z}_j,\\ &= \tilde{U}_i''(\frac{Z_iL_i}{\sum_{k\in\mathcal{N}} Z_kL_k})\frac{L_i}{(\sum_{k\in\mathcal{N}} Z_kL_k)^2}\times\\ &\quad \left[\sum_{j\in\mathcal{N},j\neq i} Z_j\dot{Z}_iL_j - \sum_{j\in\mathcal{N},j\neq i} Z_i\dot{Z}_jL_j\right],\\ &= 0. \ (\dot{Z}_iZ_j = \dot{Z}_jZ_i \text{ (Equation (24)).)}\end{aligned}$$

Therefore, if at any time the condition in Equation (25) is satisfied, $\tilde{U}_i'(\frac{Z_iL_i}{\sum_{k\in\mathcal{N}} Z_kL_k})$ for $\forall i \in \mathcal{N}$ remains constant for all future time. Therefore, Equation (25) holds for all future time. Hence, R is an invariant set of the system, which means that whenever the system state evolves into R, it remains in R. Together with the result from *Step 2*, we conclude that the update algorithm of Equation (22) converges to an invariant set R.

Step 4: Since the update algorithm of Equation (22) is equivalent to the algorithm in Equation (17), we conclude that the update algorithm in Equation (17) converges to an invariant set:

$$\begin{aligned}\Gamma &= \left\{\mathbf{W} : \frac{\dot{W}_i}{W_i} = \frac{\dot{W}_j}{W_j}, \forall i, j \in \mathcal{N}\right\},\\ &= \left\{\mathbf{W} : \tilde{U}_i'(\frac{\frac{L_i}{W_i}}{\sum_{k\in\mathcal{N}}\frac{L_k}{W_k}}) = \tilde{U}_j'(\frac{\frac{L_j}{W_j}}{\sum_{k\in\mathcal{N}}\frac{L_k}{W_k}}), \forall i, j \in \mathcal{N}\right\}.\end{aligned} \quad (27)$$

Clearly, Γ matches the optimality condition for OPT_WIN in Equation (16) and so any element in Γ solves OPT_WIN. ■

This proof shows that as long as every node observes the same $f(\mathbf{W},\mathbf{L})$, GCA converges to a set Γ, whose elements all solve OPT_WIN. The next theorem shows that given the second assumption about $f(\mathbf{W},\mathbf{L})$, GCA converges to a unique point in Γ, hence the system stabilizes at a single point.

Theorem 5: Inside Γ, if $f(\mathbf{W},\mathbf{L})$ is strictly increasing with respect to $\sum_{i\in\mathcal{N}}\frac{L_i}{W_i}$, the update algorithm in Equation (17) converges to a unique point $\widehat{\mathbf{W}} \in \Gamma$ that solves OPT_WIN.

Proof: The proof of Theorem 4 shows that the system described by Equation (17) converges to the invariant set Γ, every element of which is a solution to OPT_WIN. To prove this theorem, we only need to show that starting from any point in Γ, the algorithm of Equation (17) converges to a unique point $\widehat{\mathbf{W}} \in \Gamma$. Since the algorithm of Equation (17) is equivalent to the algorithm of Equation (22), we can show convergence of the algorithm in Equation (17) by showing convergence of the algorithm in Equation (22) to a unique point $\widehat{\mathbf{Z}} \in R$.

For ease of notation, we define $\theta = \sum_{k\in\mathcal{N}} Z_kL_k$. Since, $\sum_{i\in\mathcal{N}}\frac{L_i}{W_i} = \sum_{i\in\mathcal{N}} Z_iL_i$, the assumption that $f(\mathbf{W},\mathbf{L})$ is strictly increasing with respect to $\sum_{i\in\mathcal{N}}\frac{L_i}{W_i}$ is equivalent to the assumption that $f(\mathbf{Z},\mathbf{L})$ is strictly increasing with respect to θ.

The rest of the proof consists of three steps. First, we translate $f(\mathbf{Z},\mathbf{L})$ to a function of θ. Then, using the property that $f(\mathbf{Z},\mathbf{L})$ is strictly increasing with respect to θ, we identify the unique equilibrium point of GCA, $\widehat{Z}$. Finally, we prove that $\widehat{Z}$ is the unique stable point of GCA, and so GCA converges to $\widehat{Z}$.

Step 1: By defining $\hat{\gamma} = \tilde{U}_i'(\frac{Z_i L_i}{\sum_{k\in\mathcal{N}} Z_k L_k})$ for $\forall i \in \mathcal{N}$, the proof of Theorem 4 shows that when GCA converges to R, $\hat{\gamma}$ remains constant. Therefore, Equation (25) can be expressed as:

$$\tilde{U}_i'(\frac{Z_i L_i}{\theta}) = \hat{\gamma}, \text{ for } \forall i \in \mathcal{N}. \tag{28}$$

Next, by defining $\mathbf{g}(\cdot) = \{\tilde{U}_i'^{-1}(\cdot) : i \in \mathcal{N}\}$ and $\mathbf{L}^{-1} = \{\frac{1}{L_i} : i \in \mathcal{N}\}$, $\mathbf{Z}$ can be expressed as ($\{\cdot\}$ is the notation for set):

$$\begin{aligned}\mathbf{Z} &= \{Z_i\} = \left\{\frac{\tilde{U}_i'^{-1}(\hat{\gamma})\theta}{L_i}\right\} = \theta\{\tilde{U}_i'^{-1}(\hat{\gamma})\}^T\{\frac{1}{L_i}\} && (29)\\ &= \theta[\mathbf{g}(\hat{\gamma})]^T\mathbf{L}^{-1}, && (30)\end{aligned}$$

Therefore, $f(\mathbf{Z},\mathbf{L}) = f(\theta[\mathbf{g}(\hat{\gamma})]^T\mathbf{L}^{-1},\mathbf{L})$.

Step 2: From Equations (22) and (28), when $\dot{Z}_i = 0$,

$$f(\theta[\mathbf{g}(\hat{\gamma})]^T\mathbf{L}^{-1},\mathbf{L}) = \tilde{U}_i'(\frac{Z_i L_i}{\theta}) = \hat{\gamma}.$$

If $\hat{\theta}$ is the solution to the above equation,

$$f(\hat{\theta}[\mathbf{g}(\hat{\gamma})]^T\mathbf{L}^{-1},\mathbf{L}) = \hat{\gamma}. \tag{31}$$

Since $f(\mathbf{Z},\mathbf{L}) = f(\theta[\mathbf{g}(\hat{\gamma})]^T\mathbf{L}^{-1},\mathbf{L})$ is strictly increasing with respect to θ, $\hat{\theta}$ is unique. Therefore, the unique equilibrium point of the system can be defined as $\widehat{\mathbf{Z}} = \hat{\theta}[\mathbf{g}(\hat{\gamma})]^T\mathbf{L}^{-1} = \{\hat{Z}_i : i \in \mathcal{N}\}$, where:

$$\begin{aligned}\hat{Z}_i &= \frac{\tilde{U}_i'^{-1}(\hat{\gamma})\hat{\theta}}{L_i}, && (32)\\ \hat{\theta} &= \sum_{k\in\mathcal{N}} \hat{Z}_k L_k. && (33)\end{aligned}$$

Step 3: To show that $\widehat{Z}$ is the unique stable point of GCA, we define a scalar function $V_2(\mathbf{Z})$ as follows:

$$V_2(\mathbf{Z}) = \sum_{i\in\mathcal{N}} \int_{\hat{Z}_i}^{Z_i} \frac{1}{\sigma}(\hat{Z}_i - \sigma)L_i d\sigma.$$

The following proof shows that $V_2(\mathbf{Z})$ is a Lyapunov function, and therefore GCA converges to $\widehat{Z}$.

$$\begin{aligned}\dot{V}_2 &= \textstyle\sum_{i\in\mathcal{N}}(\frac{\partial V_2}{\partial Z_i})\dot{Z}_i,\\ &= \textstyle\sum_{i\in\mathcal{N}} \frac{1}{Z_i}(\hat{Z}_i - Z_i)L_i\dot{Z}_i.\end{aligned}$$

Note that in the invariant set R,

$$\begin{aligned}\dot{Z}_i &= \alpha Z_i(\tilde{U}_i'(\tfrac{Z_i L_i}{\theta}) - f(\mathbf{Z},\mathbf{L})) && \text{Equation (22) and definition of } \gamma\\ &= \alpha Z_i(\hat{\gamma} - f(\mathbf{Z},\mathbf{L})). && \text{from Equation (28)}\end{aligned}$$

Therefore, using Equations (31) and (33),

$$\begin{aligned}\dot{V}_2 &= \textstyle\sum_{i\in\mathcal{N}} \alpha(\hat{Z}_i - Z_i)L_i(\hat{\gamma} - f(\mathbf{Z},\mathbf{L})),\\ &= \alpha(\textstyle\sum_{i\in\mathcal{N}} \hat{Z}_i L_i - \sum_{i\in\mathcal{N}} Z_i L_i)(\hat{\gamma} - f(\mathbf{Z},\mathbf{L})),\\ &= \alpha(\hat{\theta} - \theta)\left[f(\hat{\theta}[\mathbf{g}(\hat{\gamma})]^T\mathbf{L}^{-1},\mathbf{L}) - f(\theta[\mathbf{g}(\hat{\gamma})]^T\mathbf{L}^{-1},\mathbf{L})\right].\end{aligned}$$

Since $f(\theta[\mathbf{g}(\hat{\gamma})]^T\mathbf{L}^{-1},\mathbf{L})$ is strictly increasing with respect to θ, $\dot{V}_2 \geq 0$. This equality holds if and only if $\theta = \hat{\theta}$. Therefore, we have shown that the algorithm in Equation (22) converges to a unique point $\widehat{\mathbf{Z}} \in R$. Since the algorithm of Equation (17) is equivalent to the algorithm of Equation (22), we conclude that the algorithm in Equation (17) converges to a unique point $\widehat{\mathbf{W}} \in \Gamma$. ■

The proofs of Theorems 4 and 5 demonstrate that the system is stable under the control of GCA and that GCA converges to a unique point that solves OPT_WIN. Therefore, GCA achieves our goal of support for arbitrary fairness definitions. Next, we present how GCA can be used to achieve high channel utilization.

VI. Channel Utilization of GCA

Theorem 4 shows that utility functions determine the invariant set Γ (see Equation 27), where the ratios between any two W_i's are fixed and invariant. Essentially, the choice of utility functions defines the ratios of W_i's at the stable point of GCA, and, therefore, the fairness between nodes. However, multiple assignments of $\mathbf{W}$ may satisfy the same ratio condition and their channel utilization may be quite different.

In Sections V-B and V-C, we note that the choice of $f(\mathbf{W},\mathbf{L})$ controls the stable point of the system and so determines channel utilization. If $\mathbf{W}$ at the stable point is too large, channel bandwidth is not fully utilized since idle periods are too long. If $\mathbf{W}$ at the stable point is too small, collisions increase, which also results in inefficient use of bandwidth. Therefore, the problem of maximizing channel utilization is essentially the problem of choosing the right $f(\mathbf{W},\mathbf{L})$, which, together with the choice of utility functions, should enable the system to stabilize at a point that supports the fairness definition and achieves high channel utilization.

A. Optimal Stable Point

To investigate how to choose $f(\mathbf{W},\mathbf{L})$ such that the system stables at a point that maximizes channel utilization, we need to identify this optimal stable point. In this section, we analyze the property of the optimal stable point of GCA and show that at the optimal stable point, the sum of the reciprocals of all W_i's is quasi-constant regardless of the number of competing nodes and therefore can be pre-calculated. Hence, using this property, we can design $f(\mathbf{W},\mathbf{L})$ to ensure that GCA converges around the optimal stable point (details in Section VI-B).

To identify the property of ω at the optimal stable point, assume there are n competing nodes belonging to m classes $c_1, c_2, \ldots c_m$ and nodes in the same class have the same utility function. The fraction of nodes in each class is $\beta_1, \beta_2, \ldots \beta_m$, where $\sum_{i=1}^m \beta_i = 1$. Therefore, ω can be expressed as

$$\omega = \sum_{i=1}^n \frac{1}{W_i}. \tag{34}$$

To capture the fact that the ratios between contention window sizes are determined by the choice of utility functions and the fact that the nodes in the same class share the same utility function, we define a new variable φ_i as follows:

$$\varphi_i = \frac{1/W_i}{\sum_{j=1}^n 1/W_j} = \frac{1/W_i}{\omega}, \forall 1 \leq i \leq n, \tag{35}$$

$$\varphi_i = \varphi_j = \varphi_{c_k}, \forall i,j \in c_k, \tag{36}$$

where $\sum_{i=1}^{n}\varphi_i = \sum_{k=1}^{m} n\beta_k\varphi_{c_k} = 1$.

To maximize channel utilization, the average time between each successful packet transmission, denoted as F, must be minimized. Given the probability that a successful transmission occurs in a slot, P_s, the average number of slots between each successful transmission can be expressed as $(1/P_s - 1)$. Given the virtual time channel model from Section IV, virtual time slots between two consecutive successful transmissions may be idle slots or busy slots containing collisions. Therefore,

$$\begin{aligned} F &= (\frac{1}{P_s} - 1)(aSlotTime\frac{P_I}{1-P_s} + T_c\frac{P_c}{1-P_s}), &(37)\\ &= \frac{1}{P_s}(aSlotTime \cdot P_I + T_c P_c), &(38)\end{aligned}$$

where $aSlotTime$ is the duration of an idle virtual time slot, T_c is the duration of a busy virtual time slot containing collisions, P_I is the probability of a virtual time slot being idle and P_c is the probability that a collision happens in a virtual time slot. Equation (37) follows because given the condition that a slot does not include a successful transmission, the probability that the slot is an idle slot is $\frac{P_I}{1-P_s}$ and, similarly, given the condition that a slot does not include a successful transmission, the probability that the slot has a collision is $\frac{P_c}{1-P_s}$. Note that P_I equals the probability that no node transmits in a slot. Combining P_I with Equations (7) and (35), we get:

$$\begin{aligned} P_I &= \prod_{i=1}^{n}(1-\tau_i) = \prod_{i=1}^{n}(1 - \frac{1}{W_i/2+1})\\ &= \frac{1}{\prod_{k=1}^{m}(1+2\omega\varphi_{c_k})^{n\beta_k}}. \end{aligned} \quad (39)$$

Since the probability that Node i successfully transmits a packet in a slot is P_i and a slot can have at most one successful transmission, according to the additive rule of probability for mutually exclusive events: $P_s = \sum_{i=1}^{n} P_i$. Combining this with Equations (6), (7), (39) and the definition of ω results in:

$$P_s = \sum_{i=1}^{n}\frac{\tau_i}{1-\tau_i}\prod_{j=1}^{n}(1-\frac{1}{W_i/2+1}) = \sum_{i=1}^{n}\frac{2}{W_i}P_I = 2\omega P_I. \quad (40)$$

Given that P_c equals the probability that a slot is neither idle nor contains a successful transmission,

$$P_c = 1 - P_I - P_s. \quad (41)$$

Using Equations (39), (40) and (41), the expression of F in Equation (38) can be translated to a function of ω as follows:

$$\begin{aligned} F(\omega) = \frac{1}{2\omega}[T_c\prod_{k=1}^{m}(1+2\omega\varphi_{c_k})^{n\beta_k}\\ - T_c(1+2\omega) + aSlotTime]. \end{aligned} \quad (42)$$

According to optimization theory [2], the ω that minimizes F, denoted as ω_{opt}, satisfies $F'(\omega_{opt}) = 0$. By calculating $F'(\omega)$ and setting it to zero,

$$\begin{aligned} \left[1 - \sum_{k=1}^{m}(\frac{2\omega_{opt}n\beta_k\varphi_{c_k}}{1+2\omega_{opt}\varphi_{c_k}})\right]\prod_{k=1}^{m}(1+2\omega_{opt}\varphi_{c_k})^{n\beta_k}\\ = 1 - \frac{aSlotTime}{T_c}. \end{aligned} \quad (43)$$

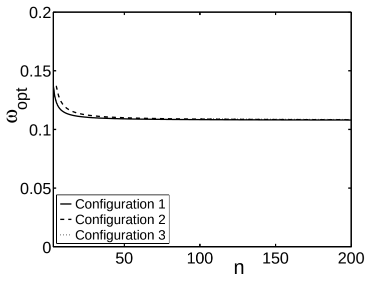


Fig. 2. ω_{opt} in three network configurations. Configuration 1 has 1 class, Configuration 2 has 2 classes with $\varphi_{c_1} : \varphi_{c_2} = 1 : 5$ and $\beta_1 : \beta_2 = 0.5 : 0.5$ and Configuration 3 has 4 classes with $\varphi_{c_1} : \varphi_{c_2} : \varphi_{c_3} : \varphi_{c_4} = 1 : 5 : 10 : 20$ and $\beta_1 : \beta_2 : \beta_3 : \beta_4 = 0.5 : 0.3 : 0.15 : 0.05$

Since $\sum_{k=1}^{m} n\beta_k\varphi_{c_k} = 1$, when $n \to \infty$, Equation (43) becomes:

$$(1-2\omega_{opt})e^{2\omega_{opt}} = 1 - \frac{aSlotTime}{T_c}. \quad (44)$$

Solving this equation gives the lower bound of ω_{opt}. Figure 2 depicts how ω_{opt} changes as n increases. As we can see, for a large n, ω_{opt} is a quasi-constant and the differences between different configurations of classes are hard to distinguish. Therefore, we can pre-calculate this quasi-constant and pre-configure GCA to converge around this value by a proper design of $f(\Omega)$.

Given ω_{opt}, the optimal stable point of $\mathbf{W}$ is:

$$W_{i,opt} = 1/(\varphi_i\omega_{opt}), \forall i.$$

Since $\sum_{i=1}^{n}\varphi_i = 1$, φ_i decreases as n increases. Because ω_{opt} tends to be a constant as n increases, it can be concluded that $\mathbf{W}$ at the optimal stable point tends to increase as n increases.

In summary, this analysis shows that ω_{opt}, which is the sum of the reciprocals of all W_is at the optimal stable point of GCA, is a quasi-constant whose range can be pre-calculated. In addition, as the number of competing nodes increases, the W_i's at the optimal stable point also increase. Therefore, if we use the pre-calculated range of ω_{opt} in the design of $f(\mathbf{W}, \mathbf{L})$ so that the stable point of GCA is around the optimal value, we ensure that W_i's increase as the number of competing nodes increases. Essentially, in this way, GCA can provide congestion control that prevents the system from heavy collision loss. On the other hand, when the number of competing nodes decreases, GCA decreases every node's W_i to avoid long idle periods.

B. Choice of $f(\mathbf{W}, \mathbf{L})$

Since ω_{opt} is a quasi-constant, if at a stable point of the system, the value of ω is inside the small range of ω_{opt}, this stable point is not far from the optimal stable point. Therefore, the channel utilization of the system is close to the maximum value. The simplified case, as we have discussed above, can be used to get an approximation of the range of ω_{opt}. If the range of ω_{opt} is known, we can design some $f(\mathbf{W}, \mathbf{L})$ functions to confine the stable point of the system so that its ω is inside the range of ω_{opt}. When the ω of the system is larger than the upper bound of ω_{opt}, indicating a too small $\mathbf{W}$, the value of $f(\mathbf{W}, \mathbf{L})$ is a large negative value. According to Equation (17), this large negative value of $f(\mathbf{W},\mathbf{L})$ forces the system to increase its $\mathbf{W}$, driving its ω back inside the range of ω_{opt}. Similarly, when the ω of the system is smaller than the lower bound of ω_{opt}, $f(\mathbf{W},\mathbf{L})$ becomes a large positive value, which drags the system back inside the range of ω_{opt}. Examples of $f(\mathbf{W},\mathbf{L})$ are presented in Section VIII. Our simulation results in Section IX verify the effectiveness of this approach.

VII. Implementation Considerations

In the previous section, we introduced our general contention window control algorithm, GCA, and showed how it can be used to achieve fair and efficient channel utilization. However, we need to address two implementation issues of GCA. First, estimation of the fraction of the total network capacity allocated to Node i, x_i, in Equation (17). Second, choosing the utility functions. In this section, we address these two issues.

A. Estimation of x_i

If the network capacity C is known, a node can simply observe its own sending rate, s_i, to obtain x_i, since $x_i = \frac{s_i}{C}$. However, the capacity of a wireless channel is not always constant since it may be affected by outside interference, such as conflicting technologies or a microwave. Therefore, this method is not practical for use in real networks.However, a node can directly estimate its x_i by observing two states of the channel, the average number of idle virtual slots between two busy virtual slots, I, and the average length of a busy virtual slot, T_b. Since in IEEE 802.11 networks a node monitors the channel continuously, I and T_b can be obtained at the MAC layer easily. In the rest of this section, IEEE 802.11 DCF RTS/CTS mode is used as an example to show how this can be done.

Let P_b be the probability that a virtual slot is a busy slot. Since I is the average number of idle slots between two consecutive busy slots,

$$P_b = 1 - P_I = \frac{1}{I+1}. \quad (45)$$

Since a busy virtual slot is caused either by a successful transmission or by a collision, the average length of a busy slot, T_b, can be expressed as:

$$T_b = T_s \frac{\sum_{i\in\mathcal{N}} P_i}{P_b} + T_c \frac{P_c}{P_b}, \quad (46)$$

where P_i is the probability that Node i successfully transmits in a virtual slot, T_s is the average length of a virtual slot with a successful transmission and T_c is the duration of a virtual slot with a collision. T_s can be expressed as $T_s = RTS + CTS + 3 \times SIFS + DATA + ACK + DIFS + aSlotTime$. Since RTS/CTS exchange is used, collisions usually happen between RTS packets. Hence, $T_c = RTS + EIFS + aSlotTime$. Based on IEEE 802.11 configurations, T_c is much smaller than T_s. Therefore, as long as P_c is not much larger than $\sum_{i\in\mathcal{N}} P_i$, using Equation (45), Equation (46) can be simplified to:

$$T_b \approx T_s \frac{\sum_{i\in\mathcal{N}} P_i}{P_b} = T_s(I+1)\sum_{i\in\mathcal{N}} P_i. \quad (47)$$

Note that T_s also satisfies $T_s = \sum_{i\in\mathcal{N}} \frac{L_i P_i}{S \sum_{j\in\mathcal{N}} P_j}$, where S is the channel transmission rate. Therefore, from Equation (47), we can get $\sum_{i\in\mathcal{N}} P_i L_i = T_b S/(I+1)$. Since $\sum_{i\in\mathcal{N}} P_i L_i$ is the average network throughput per virtual slot and $P_i L_i$ is Node i's average throughput per virtual slot,

$$x_i = \frac{P_i L_i}{\sum_{j\in\mathcal{N}} P_j L_j} \approx \frac{P_i L_i (I+1)}{T_b S}. \quad (48)$$

To calculate P_i from I, note that Node i transmits in a slot successfully if and only if it is the only node that transmits in that slot. Therefore, $P_i = \frac{1}{W_i/2+1} \prod_{j\in\mathcal{N}, j\neq i}(1 - \frac{1}{W_j/2+1})$. Combining the fact that $P_b = 1 - P_I$ and using Equations (39) and (45), P_i becomes $P_i = \frac{2}{W_i}(1 - \frac{1}{I+1})$. Integrating this with Equation (48), we finally obtain the estimation of x_i based on I and T_b:

$$\boxed{x_i \approx \frac{2 L_i I}{W_i T_b S}}. \quad (49)$$

Since in IEEE 802.11 networks a node monitors the channel continuously, the value of I and T_b can easily be obtained at the MAC layer. Therefore, the approximation in Equation (49) provides a practical estimation of x_i.

B. Choice of Utility Functions

Depending on the system design policies, the choices of utility functions can be quite different. The strength of GCA is that it is very flexible and can be used with a large range of utility functions that define a variety of fairness definitions, as long as the utility functions are strictly increasing concave functions. In this section, we briefly review several common utility functions and their corresponding fairness definitions.

1) Strict Priority: For a system that needs to achieve strict priority (i.e., the highest-priority nodes get all the bandwidth while other nodes get no bandwidth), we can use a weighted linear utility function like $\tilde{U}_i(x) = \rho_i x_i$, where ρ_i is node's weight as defined by its priority. Using this utility function, the optimal bandwidth allocation problem becomes:

$$\begin{array}{rl} & \max \sum_{i\in\mathcal{N}}(\rho_i x_i) \\ \text{over} & \sum_{i\in\mathcal{N}} x_i \le C \\ \textit{and} & x_i \ge 1 \text{ for } i \in \mathcal{N}, \end{array}$$

with the optimality condition:

$$\begin{array}{l} \sum_{i:\rho_i = max\{\rho_i, i\in\mathcal{N}\}} x_i = C, \\ x_i = 0 \text{ for } \forall i \text{ such that } \rho_i < max\{\rho_i, i\in\mathcal{N}\}. \end{array}$$

From Equation (17), the corresponding update algorithm is:

$$\dot{W}_i = -\alpha W_i[\rho_i - f(\mathbf{W},\mathbf{L})]. \quad (50)$$

Note that this utility function does not satisfy the stability conditions (i.e., $\tilde{U}(\cdot)$ is not strictly concave). Therefore, the system is not stable, meaning that our update algorithm will never converge to a certain $\mathbf{W}$. However, an interesting behavior of this update algorithm is that the highest-priority node with $\rho_i = \max\{\rho_i, i\in\mathcal{N}\}$ essentially drives $f(\mathbf{W},\mathbf{L})$ to be equal to $\max\{\rho_i, i\in\mathcal{N}\}$. The other competing nodes infinitely increase their W_i's. Therefore, the nodes with high priority quickly obtain all the bandwidth of the channel and our update algorithm achieves this strict priority between nodes.

2) Weighted Proportional Fairness: Some systems aim to achieve weighted proportional fairness [16] between nodes (i.e., bandwidth allocations satisfy $\frac{x_i}{\rho_i} = \frac{x_i}{\rho_j}, \forall i, j \in \mathcal{N}$, where ρ_i is the weight of a node). The utility function that can be used to achieve such proportional fairness is a weighted log function like $U_i(x_i) = \rho_i \log x_i$. Our update algorithm for this system is:

$$\dot{W}_i = -\alpha W_i[\frac{\rho_i}{x_i} - f(\mathbf{W}, \mathbf{L})].$$

3) Minimum Potential Delay: If the policy of the system is to minimize the total delay of file transfers in the system, the utility function can be expressed as $U_i(x) = -\frac{\rho_i}{x_i}$, where ρ_i is the size of the file that Node i is transmitting. Our update algorithm for this system is:

$$\dot{W}_i = -\alpha W_i[\frac{\rho_i}{x_i^2} - f(\mathbf{W}, \mathbf{L})].$$

4) Mixed Utility: It is also possible that in a system, different nodes have different goals and hence different utilities. In such situations, each node simply updates its contention window according to its own utility function. The system automatically converges to a stable point where the aggregated utility of all competing nodes is maximized. In general, the variety of choices of the utility functions give GCA the flexibility to achieve many different kinds of fairness definitions and so GCA can be used in systems that have different fairness policies.

VIII. Case Study

In the previous sections, we have analyzed the optimality, stability and optimal stable point of GCA. Since GCA is a general algorithm for contention window control, these analyses can be used as a powerful tool to examine existing approaches and design new algorithms. Due to space limitations, we can only present a brief analysis of three examples. The first example shows how to use GCA to check the fairness of an existing algorithm. The second case shows how to use GCA to analyze the stability and efficiency of an existing algorithm. The final case shows how to use GCA to design a new contention window control algorithm.

A. Case 1: Fairness Analysis

1) PFCR: In [24], it is proposed to directly translate the rate adaptation algorithm $\dot{s}_i = \alpha - \beta\frac{P_c}{\tilde{U}_i'(s_i)}$ to a contention window control algorithm,

$$\dot{Z}_i = \alpha - \beta\frac{P_c}{\tilde{U}_i'(Z_i)}, \tag{51}$$

to solve $OPT\_BW(U, C)$ (α and β are positive constants). A special case of the algorithm with a weighted log utility function is named PFCR. Assuming uniform packet size, it can be shown that this algorithm can not achieve an arbitrary fairness definition.

At the equilibrium point of the algorithm, $\dot{Z}_i = 0$, which results in: $\tilde{U}_i'(Z_i) = \tilde{U}_j'(Z_j) = \frac{\beta P_c}{\alpha}, \forall i, j \in \mathcal{N}$. By replacing Z_i with $\frac{1}{W_i}$, we get:

$$\tilde{U}_i'(\frac{1}{W_i}) = \tilde{U}_j'(\frac{1}{W_j}) = \frac{\beta P_c}{\alpha}, \forall i, j \in \mathcal{N}, \tag{52}$$

which does not satisfy the optimality condition for OPT_WIN in Equation (16), and hence can not achieve an arbitrary fairness. Although, for log utility functions (e.g. PFCR), when Equation (52) is satisfied, the fairness condition in Equation (16) is also satisfied. However, such a property does not hold for many utility functions (e.g $\tilde{U}_i(x_i) = \rho_i x_i + log x_i$).

2) AOB: In this section, we analyze the fairness property of AOB, Asymptotically Optimal Backoff Algorithm. AOB is proposed to dynamically adjust contention window sizes to achieve maximum bandwidth utilization. In AOB, for Node i with priority ρ_i, at every packet transmission, the node set its contention window size as:

$$Z_i^{k+1} = 0.5\left[1 - \min(1, \frac{1}{(I^k+1)2\omega_{opt}})^{m_k \rho_i}\right], \tag{53}$$

where ω_{opt} is pre-computed and m_k is the number of transmission attempts for the current packet. At the stable point of the network, we have $Z_i^{k+1} - Z_i^k = 0$, which indicate that:

$$Z_i = 0.5\left[1 - (\frac{1}{(I+1)2\omega_{opt}})^{m_i \rho_i}\right]. \tag{54}$$

Note that the average number of transmission attempts for a packet from Node i can be expressed as

$$m_i = 1/(1 - \phi_i),$$

where ϕ_i is the probability that when Node i transmits in a slot, this transmission fails due to a collision. ϕ_i can be expressed as:

$$\phi_i = 1 - \prod_{j=1, j\neq i}^{n}(1 - \frac{1}{W_i/2+1}) = 1 - (1 + 2/W_i)P_I.$$

Therefore, using Equations (39) and (45),

$$m_i = \frac{1}{(1+2/W_i)P_I} = \frac{I+1}{(1+2/W_i)I}$$

Combining this with Equation (54), we get:

$$Z_i = 0.5\left[1 - (\frac{1}{(I+1)2\omega_{opt}})^{\frac{I+1}{(1+2/W_i)I}\rho_i}\right]. \tag{55}$$

For large number of nodes, since AOB control the congestion level of the network, the W_i of each node is large. Therefore, $2/W_i \ll 1$ Hence,

$$\begin{aligned} Z_i &\approx 0.5\left[1 - (\frac{1}{(I+1)2\omega_{opt}})^{\frac{I+1}{I}\rho_i}\right] \\ &= 0.5[1 - \beta^{\rho_i}], \end{aligned} \tag{56}$$

where $\beta = (\frac{1}{(I+1)2\omega_{opt}})^{\frac{I+1}{I}}$. Therefore, we finally obtain the specific fairness achieved by AOB as:

$$\frac{s_i}{s_j} = \frac{Z_i}{Z_j} = \frac{1 - \beta^{\rho_i}}{1 - \beta^{\rho_j}} \forall i, j \in \mathcal{N}. \tag{57}$$

Obviously, the fairness achieved by AOB is not arbitrary.

B. Case 2: Stability and Efficiency Analysis

1) IEEE 802.11e: In this section, we show that IEEE 802.11e can be treated as a special form of GCA with weighted log utility function $\rho_i log(x_i)$ and $f(\mathbf{W}, \mathbf{L}) = \frac{\lambda}{I}$, where λ is a positive constant. The following derivation shows that this $f(\mathbf{W}, \mathbf{L})$ is strictly increasing with respect to $\theta = \sum_{i \in \mathcal{N}} \frac{L_i}{W_i}$ in invariant set Γ. Therefore, this form of GCA is stable. From Equations (7), (**??**) and (45),

$$\frac{1}{I+1} = (1 - \prod_{j=1}^{n}(1-\tau_j)) = (1 - \prod_{j=1}^{n}(\frac{W_j/2}{W_j/2+1})). \quad (58)$$

Combining the definition $Z_i = 1/W_i$ and Equations (30) and (58), it can be derived that in the invariant set Γ, the following relationship holds:

$$f(\mathbf{W}, \mathbf{L}) = \frac{\lambda}{I} = \lambda \left[\left(\prod_{j=1}^{n} \frac{L_j + 2U_j'^{-1}(\hat{\gamma})\theta}{L_j} \right) - 1 \right],$$

where $\hat{\gamma}$ is a positive constant. The derivative of $f(\mathbf{W}, \mathbf{L})$ to θ then can be shown as:

$$\frac{\partial}{\partial \theta} f(\mathbf{W}, \mathbf{L}) = \lambda \sum_{i=1}^{n} \frac{2U_i'^{-1}(\hat{\gamma})}{L_i + 2U_i'^{-1}(\hat{\gamma})\theta} \prod_{j=1}^{n} \frac{L_j + 2U_j'^{-1}(\hat{\gamma})\theta}{L_j} > 0.$$

Therefore, $f(\mathbf{W}, \mathbf{L})$ is strictly increasing with respect to θ inside Γ, which guarantees the stability of the system according to Theorem 5.

To examine the property of the stable point for this example, we combine the weighted utility function $f(\mathbf{W}, \mathbf{L}) = \frac{\lambda}{I}$ and Equations (49) and (17):

$$\dot{W}_i = -\alpha W_i \left[\frac{\rho_i W_i T_b R}{2 L_i I} - \frac{\lambda}{I} \right].$$

At the stable point of the system, where $\dot{W}_i = 0$,

$$W_i = \frac{2\lambda L_i}{\rho_i T_b S}.$$

Note that in the above expression, W_i is not related to the channel status and is purely decided by the packet size and weight. Essentially, this form of GCA results in static W_i's, similar to the behavior of IEEE 802.11e. Also note that the discussion in Section VI-B shows that as the number of competing node increases, the W_i's at the optimal stable points of GCA must also increase. Since the W_i's in IEEE 802.11e are fixed, this system can not achieve efficient channel utilization.

C. Case 2: Asymptotically Optimal Backoff Algorithm (AOB)

A heuristic algorithm called AOB, Asymptotically Optimal Backoff Algorithm, has been proposed to dynamically adjust contention window sizes to achieve maximum bandwidth utilization [5]. AOB assumes that every node has the same priority and the same average packet size. At every packet transmission, a node sets its contention window size to:

$$Z_i^{k+1} = 0.5 \left[1 - \min(1, \frac{1}{(I^k+1)2\omega_{opt}})^{m_k} \right],$$

where ω_{opt} is precomputed, $Z_i = \frac{1}{W_i}$ and m_k is the number of transmission attempts for the current packet. By the following calculations, we can translate AOB's contention window update algorithm to a special form of GCA.

Note that the update algorithm of AOB can be written as:

$$Z_i^{k+1} - Z_i^k = 0.5 \left[1 - \min\left(1, \frac{1}{(I_k+1)2\omega_{opt}}\right)^{m_k} \right] - Z_i^k. \quad (59)$$

Let $\tilde{U}(\cdot)$ be $\tilde{U}(x) = x - 0.5x^2$, which is a strictly increasing concave function in the range of $[0, 1]$. Since $\tilde{U}'(x) = 1 - x$, Equation (59) can be written as:

$$Z_i^{k+1} - Z_i^k = \frac{1}{2(I+1)} [\tilde{U}'(\frac{2Z_i^k}{\frac{1}{I_k+1}}) - \min(1, \frac{1}{(I_k+1)^{m_k-1}2\omega_{opt}^{m_k}} - I)]. \quad (60)$$

Note that the proof of Theorem 4 shows that GCA is equivalent to the update algorithm Z_i in Equation (22). By approximating $\sum_{k \in \mathcal{N}} Z_k \approx \frac{1}{2(I+1)}$, the discrete form of GCA is:

$$Z_i^{k+1} - Z_i^k = \alpha[\tilde{U}_i'(2Z_i^k(I+1)) - f(\mathbf{Z}^k, \mathbf{L})],$$
$$(\text{Note } L_i = L_j \forall i, j \in \mathcal{N}),$$

where each iterative step is a packet transmission. Comparing this to Equation (60), we find that the AOB algorithm is a special case of GCA with its $f(\mathbf{W}, \mathbf{L})$ defined as:

$$f(\mathbf{W}, \mathbf{L}) = f(\mathbf{Z}, \mathbf{L}) = \min(1, \frac{1}{(I+1)^{m-1}2\omega_{opt}^m} - I).$$

Using a similar method to the analysis of Case 1, it is easy to verify that this version of $f(\mathbf{W}, \mathbf{L})$ is also a strictly increasing function of $\sum_{i \in \mathcal{N}} \frac{L_i}{W_i}$ in Γ. Therefore, AOB is a stable algorithm that converges to a unique point.

To analyze the property of the stable point of AOB, note that every node has the same priority in AOB. Therefore, at the stable point, each node should have the same contention window size. Assume there are n competing nodes, $Z_i = \frac{1}{W_i} = \frac{\omega}{n}, \forall i \in \mathcal{N}$. Also note that at the stable point, $\dot{Z} = 0$ (i.e., the right side of Equation (59) becomes zero). Therefore, at the stable point of the system, ω satisfies:

$$\frac{2\omega}{n} = 1 - (\frac{\frac{1}{I+1}}{2\omega_{opt}})^m,$$

where:

$$\frac{1}{I+1} = 1 - \prod_{i=1}^{n} \frac{1}{1 + \frac{2\omega}{n}}.$$

It is obvious that ω is related to n. However, by setting $n = 1$ and $n \to \infty$, we obtain the bounds of ω as $[\omega_1, \omega_2]$, where:

$$1 - 2\omega_1 = (\frac{\omega_1}{(1+2\omega_1)\omega_{opt}})^m, \quad (61)$$

$$(\frac{1 - e^{-2\omega_2}}{2\omega_{opt}})^m = 1. \quad (62)$$

Essentially, AOB bounds the ω of the system inside a range that includes ω_{opt}, which explains why AOB can almost achieve maximum channel utilization. Therefore, AOB is a stable algorithm and achieves high channel utilization.

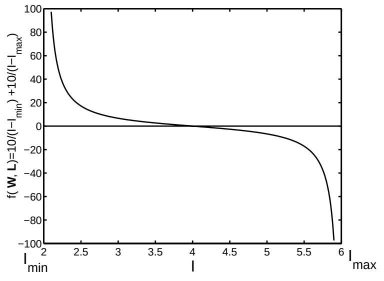


Fig. 3. $f(\mathbf{W},\mathbf{L})$ for Case 3

D. Case 3: New Algorithm Design

Since we have shown the analysis of existing approaches, we next present an example of the process of designing a special case of GCA. In this example, we assume any utility function that is strictly increasing and concave and that the observed channel state for $f(\mathbf{W},\mathbf{L})$ is I. To ensure that $f(\mathbf{W},\mathbf{L})$ can control the system operating near the maximum channel utilization point, we first give an approximation of I_{opt} corresponding to ω_{opt}. Using the same simplifications as in Section VI-A and according to Equation (58), it is easy to derive that $I = \frac{1}{(1+\frac{2\omega}{n})^n - 1}$. For large n, I at the optimal stable point is $I_{opt} \approx \frac{1}{e^{2\omega_{opt}}-1}$. This approximation allows us to calculate I_{opt} and define $f(\mathbf{W},\mathbf{L})$ as:

$$f(\mathbf{W},\mathbf{L}) = \lambda/(I - I_{min}) + \lambda/(I - I_{max}),$$

where $I_{min} < I_{opt} < I_{max}$ and the range of $[I_{min}, I_{max}]$ is small. Figure 3 shows the shape of $f(\mathbf{W},\mathbf{L})$ with $I_{min} = 2$ and $I_{max} = 6$. According to Section VI-B, this function $f(\mathbf{W},\mathbf{L})$ bounds the stable point of the system around the point that maximizes channel utilization. Using this method, we ensure that the ω of the system converges in the range around the value of ω_{opt}. The performance of this algorithm is evaluated in Section IX.

IX. Evaluation

In this section, we evaluate the performance of two variants of GCA using simulations in ns2 [9]. In GCA-EXP, GCA is used to adjust the minimum contention window size of IEEE 802.11, where contention window size is exponentially increased after a collision. In GCA-DIRECT, GCA is used to directly adjust the contention window size, without exponential increase. The evaluation of these two variants of GCA focuses on three aspects: (1) support for different definitions of fairness, (2) maintaining fairness and (3) maintaining efficiency. Although GCA is a general algorithm that can be used to achieve many different kinds of fairness, we only present the performance of GCA for strict priority and proportional fairness in this paper. These two types of fairness represent opposite extremes, where strict priority requires that all bandwidth is allocated to the node with the highest priority while proportional fairness requires that every node get a fraction of bandwidth proportional to its priority.

The $f(\mathbf{W},\mathbf{L})$ used in all simulations is the one discussed in Section VIII-D. Additionally, we use a simple implementation of GCA where a node only updates its contention window size when it transmits a packet. While other update options are possible, such as updating every virtual time slot or every short period of time, updating the contention window size at each packet transmission is the simplest and imposes minimal computational overhead. Finally, channel bandwidth is always 11Mbps.

In the first part of the evaluation, two simple simulations are used to illustrate the evolution of the system under the control of GCA. Next, the fairness of GCA is evaluated for proportional and strict priority fairness. Finally, GCA's channel utilization is evaluated for both fairness definitions.

A. System Evolution

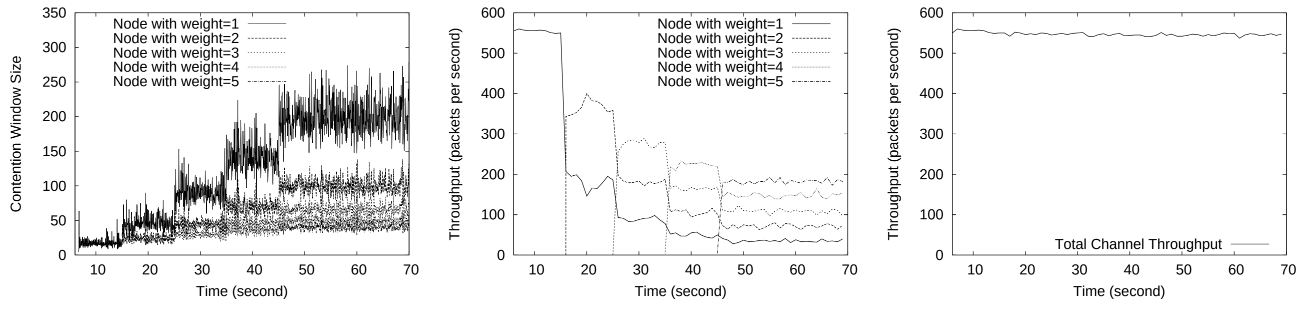
To illustrate how GCA adapts the contention window size to support fair and efficient channel utilization, simulation results from two simple cases are presented.

First, we examine the behavior of GCA for proportional fairness. In this simulation, there are five competing nodes with weighted log utility functions with weights 1, 2, 3, 4 and 5, respectively. The simulation runs for 70s. The node with weight 1 starts first at 5s, the node with weight 2 starts at 15s, the node with weight 3 starts at 25s, and so on. All packet sizes are 512B.

By examining the evolution of the contention window sizes for all competing nodes, we can see how GCA adapts the contention windows sizes as each new node starts transmitting (see Figure 4(a)). As the number of competing nodes increases, GCA increases the contention window sizes for all competing nodes to prevent congestion and so keeps the system operating near its optimal point. Additionally, GCA quickly adapts and at the same time maintains the ratio between contention window sizes to provide each competing node its weighted fair share of bandwidth (see Figure 4(b)). Finally, GCA maintains high network utilization (see Figure 4(c)), indicating that the throughput of the network is not greatly affected by the changes in the number of competing nodes. Essentially, GCA avoids congestion and maintains high throughput in the network.

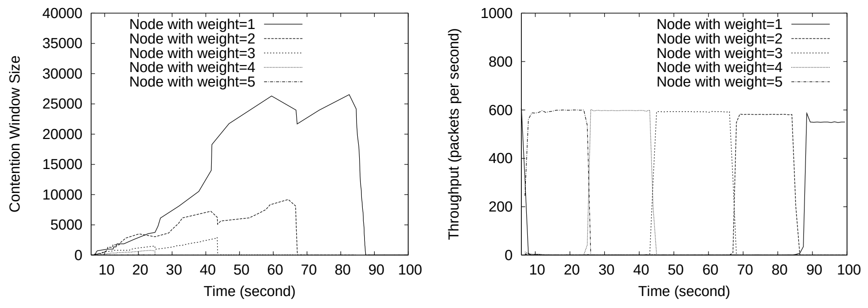
For the second example, we examine the behavior of GCA for strict priority fairness. There are five competing nodes with linear utility functions with weights 1, 2, 3, 4 and 5, respectively. All competing nodes have bulk data to be transmitted and start at 5s. The simulation runs for 100s. All packets are 512B.

Similar to the evaluation for weighted proportional fairness, Figure 5(a) shows the evolution of the contention window size as nodes finish their transmissions one by one and Figure 5(b) shows the throughput of the nodes. At the beginning of the simulation, the node with weight 5 has a very small contention window size while the other nodes with lower weights keep on increasing their contention window sizes. Therefore, the node with weight 5 soon obtains the whole channel bandwidth. After the node with weight 5 finishes its transmission, the contention window size of the node with weight 4 drops down and grabs the bandwidth of the channel. After the node with weight 4 finishes its transmission, the node with weight 3 gets the channel. The process goes on until only the node with the lowest priority is left in the network. These results show that GCA can achieve strict priority fairness between competing nodes using weighted linear utility functions.



(a) (b) (c)

Fig. 4. Evolution of contention window size, throughput and total throughput for weighted proportional fairness



(a) (b)

Fig. 5. Contention window size and throughput evolution for strict priority

B. Fairness

Next, we evaluate GCA's performance in terms of accuracy of the achieved fairness, measured in terms of Jain's fairness index [14], a common measure of fairness for bandwidth allocation. Given n competing node, Jain's fairness index is expressed as:

$$\Psi = \frac{(\sum_{i=1}^{n} \frac{s_i}{r_i})^2}{n \sum_{i=1}^{n} (\frac{s_i}{r_i})^2},$$

where r_i is Node i's share of bandwidth proportional to its weight and s_i is Node i's achieved bandwidth. The fairness index is a real value between 0 and 1 with values closer to 1 indicating better proportional fairness. When perfect proportional fairness is achieved, the fairness index equals 1. If, on the other hand, only one node out of n is allocated bandwidth, the fairness index is $1/n$. Since bandwidth allocation based on strict priority fairness aims to give all the bandwidth to the node with the highest priority, the fairness index should be $1/n$ for a perfect strict priority fairness based bandwidth allocation.

1) Weighted proportional fairness: GCA achieves weighted proportional fairness using the weighted log utility functions as discussed in Section VII-B.2. Competing nodes have weights from 1 to 5, while the number of competing nodes ranges from 5 to 50. All competing nodes start in the first 10s. We evaluate GCA for both fixed and heterogeneous packet sizes.

For heterogeneous packets sizes, per-node packet sizes are randomly picked between 400B and 1000B. By looking at the fairness indexes for GCA-EXP, GCA-DIRECT and IEEE 802.11e (see Figure 6(a)), we can see that both GCA-EXP and GCA-DIRECT achieve a fairness index that is much larger than the fairness index of IEEE 802.11e and very close to 1 regardless of the number of competing nodes. The main reason for IEEE 802.11e's unfairness is that the contention window size is independent of the packet size. In essence, nodes that send larger packets obtain more bandwidth than their fair share.

When all packets are 512B, the fairness for IEEE 802.11e is greatly improved (see Figure 6(b)), although its performance is still worse than GCA. The fairness index of GCA-EXP is also slightly smaller than for GCA-DIRECT because the exponential increase of the contention window after a collision changes the ratio between contention window sizes and hence degrades the fairness of GCA-EXP's bandwidth allocation. Therefore, both GCA-EXP and IEEE 802.11e are less fair than GCA-DIRECT. However, since GCA-EXP is able to adjust the minimum contention window to avoid excessive collisions, it essentially controls the effects of collisions on fairness. Therefore, GCA-EXP has better fairness performance than IEEE 802.11e.

2) Strict priority: To examine the ability of GCA to achieve strict priority fairness, we vary the number of competing nodes from 5 to 50, with weights ranging from 1 to 5. All nodes start in the first 10s. In one set of simulations, all nodes have 512B packets, while in the other set of simulations, each node has a different packet size randomly generated between 400B and 1000B. The simulations run for 100s.

Regardless of packets size, both GCA-EXP and GCA-DIRECT achieve a fairness index that is very close to that of the ideal allocation based on strict priority fairness (see Figure 6(c)). This demonstrates GCA's ability to support strict priority fairness based bandwidth allocation.

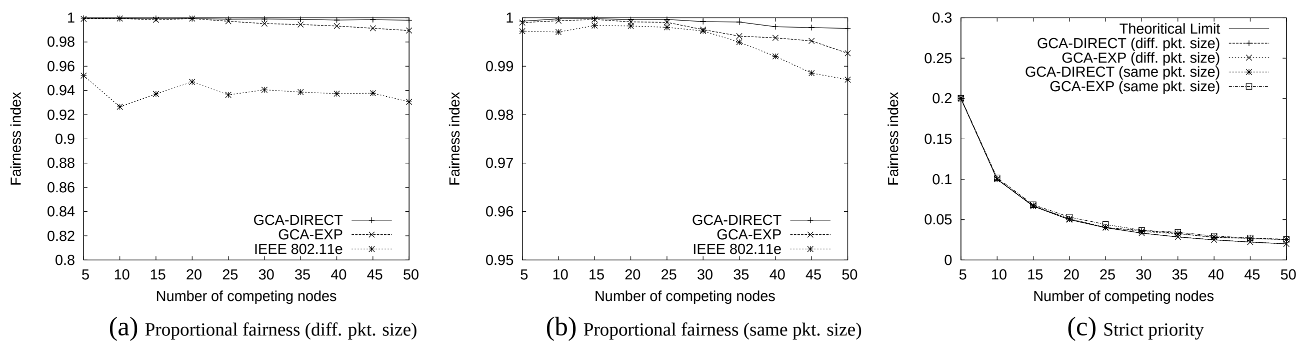


Fig. 6. Fairness index for weighted proportional fair with different packet size and same packet sizes and for strict priority with different or same packet size

C. Channel utilization

Finally, we evaluate GCA's ability to achieve high channel utilization by comparing it with IEEE 802.11 or IEEE 802.11e and the theoretical capacity of the network.

1) Weighted proportional fairness: In this set of simulations, the channel utilization of GCA is compared with IEEE 802.11e and the theoretical maximum network capacity where the contention window sizes are assigned to ensure maximum network throughput. For GCA, the weights of the log utility functions range from 1 to 5. For IEEE 802.11e, there are five classes of traffic, with the minimum contention window sizes of the classes being 30, 37, 50, 75 and 150, respectively. These contention window sizes for IEEE 802.11e are selected to ensure similar weighted bandwidth allocation as GCA. The number of competing nodes range from 5 to 50. All packets are 512B. Each simulation runs for 100s. All competing nodes start in the first 10s.

Figure 7(a) depicts the throughput of GCA and IEEE 802.11e, normalized to the theoretical maximum capacity of an IEEE 802.11 network. Essentially, the channel utilization of GCA is very close to the theoretical limit of IEEE 802.11, indicating efficient channel usage. Since IEEE 802.11e does not have the ability to dynamically adjust its minimum contention window size according to the congestion level, the channel utilization of IEEE 802.11e degrades as the number of competing nodes increases.

2) Strict priority: Since strict priority requires that only the node with the highest priority wins the bandwidth, we compare the channel utilization of GCA to an IEEE 802.11 network with only one sending node and the theoretical maximum network capacity for a single sending node. The performance is evaluated both for fixed and heterogeneous packet sizes. Each simulation runs for 100s. All nodes start in the first 10s.

Figure 7(b) depicts the throughput of GCA (with multiple competing nodes) and compares it to the throughput of an IEEE 802.11 network with one sending node, normalized to the theoretical maximum capacity of the IEEE 802.11 network with one sending node. These results show that the channel utilization of GCA is very close to the theoretical limit of a single-node IEEE 802.11 network, indicating an efficient use of the channel. Since IEEE 802.11 does not have the ability to dynamically decrease its minimum contention window size when the congestion level of the network is low, the channel utilization of IEEE 802.11 is much lower than GCA.

X. Conclusion and Future Work

In this paper, we provide a systematic method for designing dynamic contention window control algorithms that can be used to achieve fair and efficient bandwidth allocation. We decompose the requirement for both fairness and efficiency to the problem of choosing proper utility functions and observable functions of the channel state. Due to the inclusion of a wide diversity of both of these types of functions, we essentially broaden the scope of designing dynamic contention window control algorithms. In response to the limitations of current algorithms, we present a general form of dynamic contention window control (GCA) that can be used to achieve both arbitrary fairness and efficient channel utilization and prove its stability.

It is also interesting to note that our analysis of dynamic contention window control to achieve fair and efficient bandwidth allocation can also be used to formulate dynamic packet size control for achieving the same goal. The algorithm design and analysis is similar to GCA and due to space limitations, is not presented in this paper. However, in general, dynamic packet size control is inferior to contention window control since it introduces additional complexities due to the high packet error rate associated with longer packets, which may affect the fairness of bandwidth allocation.

For future work, we plan on comparing the performances of different choices of $f(\mathbf{W}, \mathbf{L})$. Additionally, we plan to extend GCA into the domain of multihop wireless networks, with the goal of supporting fair bandwidth allocation along with efficient channel utilization in such environments.

References

[1] Imad Aad and Claude Castelluccia. Differentiation Mechanisms for IEEE 802.11. In *Proceedings of INFOCOM*, 2001.

[2] Dimitri Bertsekas. *Nonlinear Programming: Second Edition*. Athena Scientific, 1999.

[3] Vaduvur Bharghavan, Alan J. Demers, Scott Shenker, and Lixia Zhang. MACAW: A media access protocol for wireless LAN's. In *SIGCOMM*, pages 212–225, 1994.

[4] Giuseppe Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3), 2000.

[5] Luciano Bononi, Marco Conti, and Enrico Gregori. Run-time optimization of IEEE 802.11 wireless LANs performance. *IEEE Transactions on Parallel and Distributed Systems (IEEE TPDS)*, 15(1), January 2004.

[6] Frederico Cali, Marco Conti, and Enrico Gregori. Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit. *IEEE Transactions on Networking*, 8(6), December 2000.

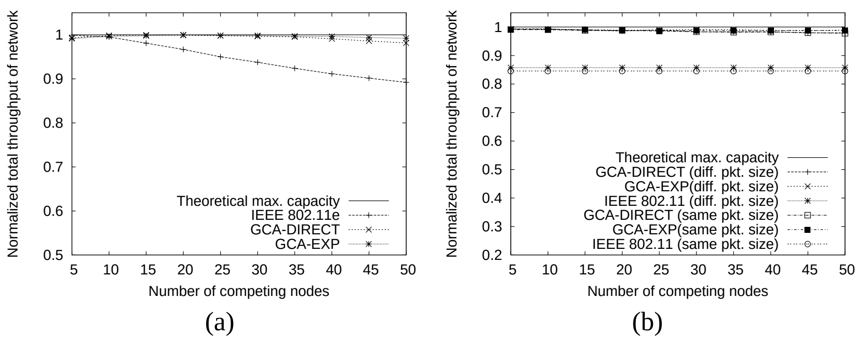


Fig. 7. Channel Utilization for proportional fairness and strict priority

[7] Frederico Cali, Marco Conti, and Enrico Gregori. IEEE 802.11 protocol: Design and performance evaluation of an adaptive backoff mechanism. *IEEE Journal on Selected Areas in Communications*, 18(9), September 2000.

[8] Jing Deng and Zygmunt Haas. Dual busy tone multiple access (dbtma): A new medium access control for packet radio networks. In *IEEE ICUPC*, 1998.

[9] Kevin Fall and Kannan Varadhan. NS notes and documentation. In *The VINT Project, UC Berkely, LBL, USC/ISI, and Xerox PARC*, 1997.

[10] Chuan Heng Foh and Moshe Zukerman. Performance Analysis of the IEEE 802.11 MAC Protocol. In *Proceeding of the European Wireless*, 2002.

[11] Chane L. Fullmer and J. J. Garcia-Luna-Aceves. Solutions to hidden terminal problems in wireless networks. In *SIGCOMM*, pages 39–49, 1997.

[12] S. Golestani and S. Bhattacharyya. A class of end-to-end congestion control algorithms for the internet. In *Proceedings of the Sixth International Conference on Network Protocols*, 1998.

[13] T.S. Ho and K. C. Chen. Performance evluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LAN's. In *Proceeding of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 1996.

[14] Raj Jain. *The Art of Computer Systems Performance Analysis: Techniques for ExperimentalDesign, Measurement, Simulation and Modeling*. John Wiley and Sons, Inc, 1991.

[15] Phil Karn. A New Channel Access Method for Packet Radio. In *Proceedings of the 9th ARRL Computer Networking Conference*, 1996.

[16] F. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. In *Journal of the Operational Research Society*, volume 49, 1998.

[17] Hassan Khalil. *Nonlinear systems*. Prentice Hall, 1996.

[18] Hwangnam Kim and Jennifer Hou. Improving Protocol Capacity with Model-based Frame Scheduling in IEEE 802.11-operated WLANs. In *ACM Mobicom*, 2003.

[19] Srisankar Kunniyur and R. Srikant. End-to-end congestion control schemes: Utility functions, random losses and ECN marks. In *INFOCOM (3)*, pages 1323–1332, 2000.

[20] Bo Li and Roberto Battiti. Performance Analysis of An Enhanced IEEE 802.11 Distributed Coordination Function Supporting Service Differentiation. In *International Workshop on Quality of Future Internet Service*, 2003.

[21] Stefan Mangold, Sunghyun Choi, Peter May, Ole Klein, Guido Hiertz, and Lothar Stibor. IEEE 802.11e Wireless LAN for Quality of Service. In *Proceedings of European Wireless*, 2002.

[22] Laurent Massoulié and James Robert. Bandwidth sharing: Objective and algorithms. *IEEE/ACM Transaction on Networking*, 10(3), June 2002.

[23] Jeonghoon Mo and Jean Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Transaction of Networking*, 8(5):556–567, 2000.

[24] Thyagarajan Nandagopal, Tae-Eun Kim, Xia Gao, and Vaduvur Bharghavan. Achieving MAC Layer Fairness in Wireless Packet Networks. In *ACM Mobicom*, 2000.

[25] Daji Qiao and Kang G. Shin. Achieving efficient channel utilization and weighted fairness for data communications in IEEE 802.11 WLAN under the dcf. In *IWQoS*, 2002.

[26] Scott Shenker. Fundamental design issues for the future internet. *IEEE Journal on Selected Areas in Communication*, 13(7), September 1995.

[27] IEEE Computer Society. 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.

[28] Rayadurgam Srikant. *The mathematics of Internet Congestion Control*. Birkhauser, 2004.

[29] Yaling Yang and Robin Kravets. Distributed QoS Guarantees for Realtime Traffic in Ad Hoc Networks. Technical Report UIUCDCS-R-2004-2446, June 2004.

APPENDIX

A. Proof of Lemma 1

Lemma 1: If scalar function V is defined as:

$$V(\mathbf{Z}) = \sum_{i\in\mathcal{N}} \tilde{U}_i\Big(\frac{Z_iL_i}{\sum_{k\in\mathcal{N}} Z_kL_k}\Big), \tag{63}$$

V is a Lyapunov function for the system described by Equation (22) with $\dot{V} \geq 0$. The zero values of $\dot{V}$ are obtained for the set $R = \{\mathbf{Z} : \frac{\dot{Z}_i}{Z_i} = \frac{\dot{Z}_j}{Z_j}, \forall i, j \in \mathcal{N}\}$.

Proof: Note that the derivative of $V(\mathbf{Z})$ to Z_i is:

$$\begin{aligned}\frac{\partial V}{\partial Z_i} &= \frac{\partial}{\partial Z_i}\tilde{U}_i\Big(\frac{Z_iL_i}{\sum_{k\in\mathcal{N}} Z_kL_k}\Big) + \frac{\partial}{\partial Z_i}\sum_{j\in\mathcal{N},j\neq i}\tilde{U}_j\Big(\frac{Z_jL_j}{\sum_{k\in\mathcal{N}} Z_kL_k}\Big),\\ &= \tilde{U}_i'\Big(\frac{Z_iL_i}{\sum_{k\in\mathcal{N}} Z_kL_k}\Big)\frac{L_i\sum_{k\in\mathcal{N}} Z_kL_k - Z_iL_i^2}{(\sum_{k\in\mathcal{N}} Z_kL_k)^2}\\ &\quad -\sum_{j\in\mathcal{N},j\neq i}\tilde{U}_j'\Big(\frac{Z_jL_j}{\sum_{k\in\mathcal{N}} Z_kL_k}\Big)\frac{Z_jL_jL_i}{(\sum_{k\in\mathcal{N}} Z_kL_k)^2}.\end{aligned}$$

The first item after the last equal sign can be expressed as:

$$\begin{aligned}&= \tilde{U}_i'\Big(\frac{Z_iL_i}{\sum_{k\in\mathcal{N}} Z_kL_k}\Big)\frac{L_i\sum_{k\in\mathcal{N},k\neq i} Z_kL_k}{(\sum_{k\in\mathcal{N}} Z_kL_k)^2},\\ &= \sum_{j\in\mathcal{N},j\neq i}\tilde{U}_i'\Big(\frac{Z_iL_i}{\sum_{k\in\mathcal{N}} Z_kL_k}\Big)\frac{Z_jL_jL_i}{(\sum_{k\in\mathcal{N}} Z_kL_k)^2}.\end{aligned}$$

Therefore,

$$\begin{aligned}\frac{\partial V}{\partial Z_i} &= \sum_{j\in\mathcal{N},j\neq i}\tilde{U}_i'\Big(\frac{Z_iL_i}{\sum_{k\in\mathcal{N}} Z_kL_k}\Big)\frac{Z_jL_jL_i}{(\sum_{k\in\mathcal{N}} Z_kL_k)^2}\\ &\quad -\sum_{j\in\mathcal{N},j\neq i}\tilde{U}_j'\Big(\frac{Z_jL_j}{\sum_{k\in\mathcal{N}} Z_kL_k}\Big)\frac{Z_jL_jL_i}{(\sum_{k\in\mathcal{N}} Z_kL_k)^2},\\ &= \frac{L_i}{(\sum_{k\in\mathcal{N}} Z_kL_k)^2}\sum_{j\in\mathcal{N},j\neq i}\Big[\tilde{U}_i'\Big(\frac{Z_iL_i}{\sum_{k\in\mathcal{N}} Z_kL_k}\Big)\\ &\quad -\tilde{U}_j'\Big(\frac{Z_jL_j}{\sum_{k\in\mathcal{N}} Z_kL_k}\Big)\Big]Z_jL_j.\end{aligned}$$

From Equation (22), $\tilde{U}_i'(\frac{Z_iL_i}{\sum_{k\in\mathcal{N}} Z_kL_k})$ can be expressed as:

$$\tilde{U}_i'\Big(\frac{Z_iL_i}{\sum_{k\in\mathcal{N}} Z_kL_k}\Big) = \Big[\frac{\dot{Z}_i}{\alpha Z_i} + f(\mathbf{Z}, \mathbf{L})\Big].$$

Substituting $\tilde{U}_i'(\frac{Z_iL_i}{\sum_{k\in\mathcal{N}} Z_kL_k})$ in the expression of $\frac{\partial V}{\partial Z_i}$, we get:

$$\begin{aligned}\frac{\partial V}{\partial Z_i} &= \frac{L_i}{(\sum_{k\in\mathcal{N}} Z_kL_k)^2}\sum_{j\in\mathcal{N},j\neq i}\Big\{\Big[\frac{\dot{Z}_i}{\alpha Z_i} + f(\mathbf{Z}, \mathbf{L})\Big]\\ &\quad -\Big[\frac{\dot{Z}_j}{\alpha Z_j} + f(\mathbf{Z}, \mathbf{L})\Big]\Big\}Z_jL_j,\\ &= \frac{L_i}{\alpha(\sum_{k\in\mathcal{N}} Z_kL_k)^2}\sum_{j\in\mathcal{N},j\neq i}\Big[\frac{\dot{Z}_i}{Z_i} - \frac{\dot{Z}_j}{Z_j}\Big]Z_jL_j.\end{aligned}$$

If $\eta = \frac{1}{\alpha(\sum_{k\in\mathcal{N}} Z_k L_k)^2}$, it is obvious that $\eta > 0$. Therefore,

$$\begin{aligned}
\frac{\partial V}{\partial Z_i} &= \eta L_i \textstyle\sum_{j\in\mathcal{N},j\neq i}[\frac{\dot{Z}_i}{Z_i} - \frac{\dot{Z}_j}{Z_j}]Z_j L_j, \\
&= \eta L_i \textstyle\sum_{j\in\mathcal{N},j\neq i}[(\frac{Z_j L_j}{Z_i})\dot{Z}_i - \dot{Z}_j L_j], \\
&= \eta[(\textstyle\sum_{j\in\mathcal{N},j\neq i} Z_j L_j)\frac{L_i}{Z_i}\dot{Z}_i - (\sum_{j\in\mathcal{N}} \dot{Z}_j L_j)L_i].
\end{aligned}$$

Therefore, the time derivative of $V(\mathbf{Z})$ can be shown as:

$$\begin{aligned}
\dot{V} &= \textstyle\sum_{i\in\mathcal{N}} \frac{\partial V}{\partial Z_i}\dot{Z}_i, \\
&= \textstyle\sum_{i\in\mathcal{N}} \eta[(\sum_{j\in\mathcal{N},j\neq i} Z_j L_j)\frac{L_i}{Z_i}\dot{Z}_i - (\sum_{j\in\mathcal{N},j\neq i} \dot{Z}_j L_j)L_i]\dot{Z}_i, \\
&= \eta \textstyle\sum_{i,j\in\mathcal{N},j\neq i}[\frac{Z_j L_j L_i}{Z_i}\dot{Z}_i^2 + \frac{Z_i L_i L_j}{Z_j}\dot{Z}_j^2 - 2\dot{Z}_i\dot{Z}_j L_i L_j], \\
&= \eta \textstyle\sum_{i,j\in\mathcal{N},j\neq i}[(\sqrt{\frac{Z_j L_j L_i}{Z_i}} \mid \dot{Z}_i \mid - \sqrt{\frac{Z_i L_i L_j}{Z_j}} \mid \dot{Z}_j \mid)^2 \\
&\quad +2 \mid \dot{Z}_i \mid\mid \dot{Z}_j \mid L_i L_j - 2\dot{Z}_i\dot{Z}_j L_i L_j], \\
&\geq \eta \textstyle\sum_{i,j\in\mathcal{N},j\neq i}(\sqrt{\frac{Z_j L_j L_i}{Z_i}} \mid \dot{Z}_i \mid - \sqrt{\frac{Z_i L_i L_j}{Z_j}} \mid \dot{Z}_j \mid)^2, \\
&\geq 0.
\end{aligned}$$

Finally, the equality holds if and only if:

$$\frac{\dot{Z}_i}{Z_i} = \frac{\dot{Z}_j}{Z_j}, \text{ for } \forall i, j \in \mathcal{N}.$$

■

B. Notation

- $\mathcal{N}$: the set of transmitting stations
- C: the network capacity
- P_i: the probability that Node i successfully transmits in a virtual slot
- τ_i: the probability that Node i attempts to transmit in a virtual slot
- s_i: the sending rate of Node i
- L_i: the channel bandwidth consumed for a successful packet transmission.
- x_i: the fraction of channel bandwidth of Node i
- W_i: the abbreviation for contention window size of Node i
- W_i^{min}: the abbreviation for minimum contention window size of Station i
- $\mathbf{W}$: $\{W_i : i \in \mathcal{N}\}$
- $\mathbf{L}$: $\{L_i : i \in \mathcal{N}\}$
- $\mathbf{P}$: $\{P_i : i \in \mathcal{N}\}$
- Z_i: $\frac{1}{W_i}$
- Γ: the invariant set of GCA
- R: the invariant set of GCA-Z
- F: the average time between successful packet transmissions
- I: the average number of idle virtual slots between two busy virtual slots
- ω: $\sum_{i\in\mathcal{N}} \frac{1}{W_i}$
- φ_i: $\frac{1/W_i}{\omega}$
- θ: $\sum_{i\in\mathcal{N}} \frac{L_i}{W_i} = \sum_{k\in\mathcal{N}} Z_k L_k$
- T_b: the average length of busy virtual slot
- T_c: the average duration of a virtual slot including a collision
- P_I: the probability that a virtual slot is an idle slot
- P_s: the probability that a successful transmission happens in a virtual slot
- P_c: the probability that a collision happens in a virtual slot
- S: the channel sending rate of IEEE 802.11