

OFFERED LOAD ESTIMATION AND APPLICATIONS FOR USING SAME IN A COMMUNICATION NETWORK

5

Background

1. Field of the Invention

The invention relates generally to communication systems, and more particularly to offered load estimation and applications for using same in a communication network.

10

2. Discussion of Related Art

In today's information age, there is an increasing need for high-speed communication networks that provide Internet access and other on-line services for an ever-increasing number of communications consumers. To that end, communications networks and technologies are evolving to meet current and future demands. Specifically, new networks are being deployed which reach a larger number of end users, and protocols are being developed to utilize the added bandwidth of these networks efficiently.

20

One technology that has been widely employed and will remain important in the foreseeable future is the shared medium communication network. A shared medium communication network is one in which a single communications channel (the shared channel) is shared by a number of users such that uncoordinated transmissions from different users may interfere with one another. The shared medium communication network typically includes a number of secondary stations that transmit on the shared channel, and a single primary station situated at a common receiving end of the shared channel for, among other things, coordinating access by the secondary stations to the shared channel. Since communication networks typically have a limited number of communication channels, the shared medium communication network allows many users to gain access to the

25

30

network over a single communication channel, thereby allowing the remaining communication channels to be used for other purposes.

Many techniques are known which the primary station can use for coordinating access by the secondary stations to the shared channel. The ability of the primary station to meet specified performance goals depends on a number of factors, including the particular technique(s) employed and the number of secondary stations attempting to access the shared channel at any given time (often referred to as the "offered load"). Furthermore, the ability of the primary station to meet specified performance goals often depends on the ability of the primary station to adapt to changes in the offered load over time, and more specifically on how quickly the primary station can adapt to such changes. Thus, the primary station must be able to estimate the offered load of the network and react accordingly.

Brief Description of the Drawing

In the Drawing,

FIG. 1 is time line depicting a shared channel in accordance with a preferred embodiment of the present invention, with the shared channel divided into successive frames including a request interval for providing contention access;

FIG. 2 is a three-dimensional graph depicting a planar region ABC representing the set of possible contention outcomes in accordance with a preferred embodiment of the present invention;

FIG. 3A is a three-dimensional graph showing the locus of expected outcomes within the planar region ABC in accordance with a preferred embodiment of the present invention;

FIG. 3B is a two-dimensional graph showing the locus of expected outcomes within the planar region ABC in accordance with a preferred embodiment of the present invention;

FIG. 4 is a two-dimensional graph showing the planar region ABC divided into three regions based on the distance of points from the locus of expected outcomes in accordance with a preferred embodiment of the present invention;

5 FIG. 5 is a three-dimensional graph showing the planar region ABC intersected with three planes S_0 , I_0 , and C_0 in accordance with a preferred embodiment of the present invention;

FIG. 6 is a two-dimensional graph showing the three planes S_0 , I_0 , and C_0 intersecting at the point of maximum likelihood of
10 SUCCESS outcomes within planar region ABC in accordance with an embodiment of the present invention;

FIG. 7 is a two-dimensional graph showing the three planes S_0 , I_0 , and C_0 in accordance with a preferred embodiment of the present invention;

15 FIG. 8 is a block diagram showing a shared medium communication network in accordance with a preferred embodiment of the present invention;

FIG. 9 is a state diagram showing three possible states for a MAC User in accordance with a preferred embodiment of the
20 present invention;

FIG. 10 is a block diagram showing a primary station in accordance with a preferred embodiment of the present invention;
and

FIG. 11 is a block diagram showing a secondary station in
25 accordance with a preferred embodiment of the present invention.

Detailed Description

As discussed above, the primary station must be able to estimate the offered load of the network and react accordingly.
30 The present invention includes techniques for estimating offered load based on a history of contention outcomes. The present invention also includes applications for utilizing the estimated offered load for determining a request interval size and for

determining a contention access mode in a communication network. The present invention is described herein with reference to various embodiments.

5 **1. Offered Load Estimation Model**

 In accordance with the present invention, the shared channel is divided into discrete time slots, and is often referred to as a "slotted channel." The slotted channel is organized into successive frames, where each frame consists of a number of
10 slots. The number of slots in each frame can be fixed or variable. For convenience, T_k represents the number of slots in a frame k . A portion of each frame (referred to as the "request interval") is used for transmitting requests for contention access, and particularly for placing reservations for bandwidth. The number of
15 slots in each request interval can be fixed or variable. For convenience, M_k represents the number of slots in the request interval of the frame k (referred to as "request interval k "). Assuming that R slots are needed to transmit a request, the request interval k therefore provides M_k/R request transmission
20 opportunities in which requests can be transmitted. Although M_k is typically selected such that M_k/R is an integer, there is no requirement that M_k be so selected, and the value M_k/R is heuristically treated as being a real number for the purpose of discussion.

25 For each request transmission opportunity in a request interval, such as request interval k , there will be either (1) no request transmission; (2) a single request transmission; or (3) multiple request transmissions. When a single request is transmitted in response to a request transmission opportunity, it is
30 presumed for the sake of discussion that the request is successful. When multiple requests are transmitted, it is presumed that the requests collide and are therefore

unsuccessful. For convenience, the three outcomes are referred to as IDLE, SUCCESS, and COLLISION, respectively.

Offered load estimation was addressed in a paper written by Frits C. Schoute and published in the IEEE transactions on Communications, Vol. COM-31, NO.4, April 1983. The paper is related to the present invention because it provides a solution to a similar problem, yet, in a different environment and using a different technique. Schoute attempts to estimate the offered load in a Slotted Dynamic Frame Length ALOHA environment, where all data is transmitted in contention and hence reservation is totally absent. In summary, for each slot having a COLLISION outcome, Schoute computes the expected number of contending users based on the known maximum throughput of the system ($1/e$), and increments the offered load estimation accordingly. Schoute's solution can be readily extended to a contention-based reservation context in accordance with the present invention, provided that the goal is to maximize the contention throughput in the request interval.

However, the goal of the present invention is not to maximize the contention throughput in the request interval. Rather, the goal of the present invention is to estimate the offered load based on the number of observed IDLE, SUCCESS, and COLLISION outcomes in each request interval k . Therefore, the offered load estimation techniques of the present invention differ substantially from the offered load estimation technique of Schoute.

For the sake of simplicity, it is assumed that only certain requests are eligible for transmission during the request interval k . Specifically, only those requests that are available for transmission prior to request interval k (including "new" requests and requests made as part of a collision resolution scheme) are eligible for transmission in request interval k . Therefore, any requests that become available for transmission during request interval k cannot be transmitted in request interval k , and must

wait until request interval (k+1). A system that adheres to such a rule is often referred to as a "gated" system.

Given that the system is gated, all requests that become available for transmission during frame (k-1) will be transmitted during request interval k. For convenience, N_{k-1} represents the total number of requests that become available for transmission during frame (k-1) that are transmitted during request interval k. The N_{k-1} requests can be conceptualized as becoming available randomly over the T_{k-1} slots in frame (k-1), such that the average rate that requests become available during frame (k-1) is equal to:

$$\text{Eq. 1} \quad g_{k-1} = N_{k-1}/T_{k-1}$$

Thus, g_{k-1} represents the average number of requests per slot over the frame (k-1), such that:

$$\text{Eq. 2} \quad N_{k-1} = g_{k-1} \times T_{k-1}$$

Because the N_{k-1} requests are transmitted in the M_k/R request transmission opportunities in request interval k, it therefore follows that the average number of requests transmitted per request transmission opportunity during the request interval k is equal to:

$$\begin{aligned} \text{Eq. 3} \quad G_k &= N_{k-1}/(M_k/R) \\ &= (g_{k-1} \times T_{k-1})/(M_k/R) \\ &= (g_{k-1} \times T_{k-1} \times R)/M_k \end{aligned}$$

The probability distribution of the number of requests transmitted per request transmission opportunity can be approximated by the binomial distribution:

$$\text{Eq. 4} \quad P[m] =$$

where A is the number of requests that become available during frame (k-1) (i.e., $A = N_{k-1}$), B is the number of request transmission opportunities in frame k (i.e., $B = M_k/R$), and m is the random variable representing the number of request transmitted in a request transmission opportunity.

Therefore, the probability of SUCCESS, IDLE, and COLLISION outcomes during request interval k can be approximated as:

$$\text{Eq. 5} \quad P_k[S] = A \times \frac{1}{B} \times \left(1 - \frac{1}{B}\right)^{A-1}$$

$$\text{Eq. 6} \quad P_k[I] = \left[1 - \frac{1}{B}\right]^A$$

$$\text{Eq. 7} \quad P_k[C] = 1 - A \times \frac{1}{B} \times \left[1 - \frac{1}{B}\right]^{A-1} - \left[1 - \frac{1}{B}\right]^A$$

Provided that B is large, the binomial distribution $P[m]$ can be approximated by the Poisson distribution:

$$\text{Eq. 8} \quad P[m] = [(A/B)^m \times \exp(-A/B)] / m!$$

By definition, $G_k = A/B$. Substituting G_k for A/B in Eq. 8 results in the Poisson distribution:

$$\text{Eq. 9} \quad P[m] = [G_k^m \times \exp(-G_k)] / m!$$

Therefore, the probability of SUCCESS, IDLE, and COLLISION outcomes during request interval k can be approximated as:

$$\text{Eq. 10} \quad P_k[S] = G_k \times \exp(-G_k)$$

$$\text{Eq. 11} \quad P_k[I] = \exp(-G_k)$$

$$\text{Eq. 12} \quad P_k[C] = 1 - G_k \times \exp(-G_k) - \exp(-G_k)$$

5

Based on the above probabilities, the expected number of SUCCESS, IDLE, and COLLISION outcomes during request interval k is equal to:

$$10 \quad \text{Eq. 13} \quad E_k(S) = P_k[S] \times M_k/R$$

$$\text{Eq. 14} \quad E_k(I) = P_k[I] \times M_k/R$$

$$\text{Eq. 15} \quad E_k(C) = P_k[C] \times M_k/R$$

With respect to request interval k , the values M_k and R are known *a priori*, and the actual number of SUCCESS, IDLE, and COLLISION outcomes during request interval k can be measured. For convenience, the actual number of SUCCESS, IDLE, and COLLISION outcomes measured during request interval k are referred to as S_k , I_k , and C_k , respectively. Because S_k , I_k , and C_k are probabilistically equal to $E_k(S)$, $E_k(I)$, and $E_k(C)$, respectively, any one of Eq. 13, Eq. 14, and Eq. 15 can be used to determine the estimated offered load g_{k-1} during the request interval $(k-1)$.

Working from Eq. 14 and using the measured number of IDLE outcomes I_k to determine the offered load results in the following transformations:

$$\begin{aligned} \text{Eq. 16} \quad I_k &= E_k(I) \\ &= P_k(I) \times M_k/R \\ &= (1/R) \times M_k \times \exp(-G_k) \\ (R \times I_k)/M_k &= \exp(-G_k) \end{aligned}$$

30

Substituting G_k from Eq. 3 and solving for g_{k-1} gives the estimated offered load during the request interval $(k-1)$:

Eq. 17 $g_{k-1} = [M_k / (R \times T_{k-1})] \times \ln[M_k / (R \times I_k)]$

Thus, the estimated offered load g_{k-1} can be calculated
5 based on values that are either known *a priori* (i.e., M_k , R , and T_{k-1}) or measurable (i.e., I_k).

In many situations, the estimated offered load determined according to Eq. 17 may not be an accurate estimate of the actual offered load. This is because the Poisson distribution according
10 to Eq. 8 only approximates a binomial distribution if B is large. Depending on the number of request transmission opportunities in a single frame, the value B may or may not be large enough to ensure that the estimated offered load g_{k-1} is accurate. When the number of request transmission opportunities in a single frame is
15 not large enough to provide a statistically significant number of request transmission opportunities, the offered load estimation model must be adapted.

A. Offered Load Estimation Using Sample Window

20 A first adaptation of the offered load estimation model computes the estimated offered load over a number of consecutive frames. The number of consecutive frames must be large enough to provide a statistically significant number of request transmission opportunities, and yet not so large that the
25 offered load varies considerably within the number of frames. For convenience, the number of frames n over which the estimated offered load is calculated is referred to as the "sample window."

Assuming that I_i represents the number of IDLE outcomes in sample window frame i , then the total number of IDLE outcomes
30 over the sample window is equal to:

$$\text{Eq. 18} \quad I = \sum_{i=1}^n I_i$$

As in Eq. 16 above, the number of IDLE outcomes in the sample window frame i can be estimated by the corresponding expected number of outcomes as follows:

$$\begin{aligned} \text{Eq. 19} \quad I_i &= E_i(I) \\ &= P_i[I] \times M_i/R \\ &= M_i/R \times \exp(-G_i) \end{aligned}$$

10

such that:

$$\text{Eq. 20} \quad I = \sum M_i/R \times \exp(-G_i)$$

Using the transformations from Eq. 16 and Eq. 17 above, it is possible to calculate the instantaneous offered load g_i for each sample window frame i based on values that are either known *a priori* or measurable. By selecting an appropriate sample window, it is expected that the instantaneous offered load g_i does not vary considerably over the sample window. Therefore, the instantaneous offered load g_i can be approximated by an offered load g that is the same for each sample window frame i (i.e., $g_1 = g_2 = \dots = g_n = g$).

Substituting G_k from Eq. 3 into Eq. 20 and substituting g for each g_i results in the following:

$$\text{Eq. 21} \quad I =$$

Except for the estimated offered load g , all of the elements of Eq. 21 are either known or measurable. An objective of the present invention is to derive from Eq. 21 an estimator function for g based on the known and measurable variables, such that $g =$

f(l). For now, it is assumed that there is such a function f(l), the details of which are presented later.

In certain situations, it is desirable to regularly update the estimated offered load in order to reflect the actual offered load of the network as it changes over time. In such situations, it is
5 important that the estimated offered load adapt quickly to changes in the actual offered load.

One way to update the estimated offered load is to consider consecutive disjoint sample windows of size n, and to update the
10 estimated offered load at the end of each sample window. This approach is simple, and requires relatively infrequent updates. However, it is also relatively slow to adapt to changes in the actual offered load, and can therefore be quite inaccurate if the actual offered load changes significantly between sample
15 windows.

A more accurate way to update the estimated offered load is to use a sliding sample window and to update the estimated offered load each frame. While this approach requires more frequent updates, it adapts more quickly to changes in the offered
20 load. However, it can still be inaccurate if the actual offered load changes significantly between frames.

In order to improve the performance of the sliding sample window approach, a weighting scheme is employed which assigns a higher weight to the x most recent frames in the sample window.
25 Thus, in a sample window having n frames, the x most recent frames are assigned a weighting factor α and the (n-x) "older" frames are assigned a weighting factor β , where $\alpha > \beta$. The total weight of the frames in the sample window is equal to:

30 Eq. 22 $n' = \alpha x + \beta(n - x)$

The weighting factor β can be arbitrarily set to one (1), so that the total weight of the frames in the sample window is equal to:

$$\begin{aligned} \text{Eq. 23} \quad n' &= \alpha x + n - x \\ &= n + (\alpha - 1)x \end{aligned}$$

The weighting factor α is selected so that the weight assigned to the x most recent frames is equal to a predetermined percentage X of the total weight n' as follows:

$$\text{Eq. 24} \quad \alpha x / n' \cong X$$

With the assumption that g is constant within the sample window, it can be expected that the ratio $Y = T_{i-1}/M_i$ (i.e., the ratio of the frame size to the request interval size) will also be constant for $i = 1$ to n . Applying this assumption to Eq. 21 results in:

$$\text{Eq. 25} \quad I =$$

For convenience, T represents the weighted average of the number of slots per frame over the entire sample window as follows:

$$\text{Eq. 26} \quad T = [T_0 + \dots + T_{n-x-1} + \alpha T_{n-x} + \dots + \alpha T_{n-1}] / n'$$

For convenience, M represents the weighted average of the number of slots per request interval over the entire sample window as follows:

$$\text{Eq. 27} \quad M = [M_1 + \dots + M_{n-x} + \alpha M_{n-x+1} + \dots + \alpha M_n] / n'$$

Substituting M into Eq. 25 results in:

Eq. 28 $I \cong \exp(-g \times Y \times R) \times n' \times M/R$

5

where $Y = T_{i-1}/M_i$. Heuristically, the ratio T_{i-1}/M_i can be approximated by the ratio T/M , such that $Y = T/M$. Substituting $Y = T/M$ into Eq. 28 results in:

10 Eq. 29 $I \cong$

The estimator function for g is obtained by taking the natural logarithm on both sides of Eq. 29 and solving for g as follows:

15 Eq. 30 $g' = f(I) =$

where g' is the estimator function for g.

B. Offered Load Estimation Using Single Frame

20 A second adaptation of the offered load estimation model computes the estimated offered load over a single frame. Estimating offered load using a single frame is desirable due particularly to its simplicity in not requiring the maintenance and evaluation of historical data as required when estimating offered
25 load over a sample window. One problem with estimating offered load over a single frame, though, is that the number of request transmission opportunities in a single frame does not represent a statistically significant sample, and therefore the observed outcomes for the frame may or may not be indicative of the actual
30 offered load. However, it is known that certain outcomes are more probable than other outcomes. For example, it is unlikely (but possible) that there will be all SUCCESS outcomes with no

IDLE or COLLISION outcomes, or an equal number of IDLE and COLLISION outcomes with no SUCCESS outcomes. Thus, the set of all possible outcomes can be divided into a set containing those outcomes that are likely and therefore "trusted," and a set
 5 containing those outcomes that are unlikely and therefore "untrusted." If an observed outcome falls within the set of "trusted" outcomes, then it is used to update the estimated offered load; otherwise, the observed outcome is ignored and is not used to update the estimated offered load. The problem then is to
 10 define the set of "trusted" and "untrusted" outcomes.

Since there are M_k/R request transmission opportunities in frame k and each request transmission opportunity results in either a SUCCESS, IDLE, or COLLISION outcome, then the sum of the number of outcomes is equal to M_k/R as follows:

15

$$\text{Eq. 31} \quad I_k + S_k + C_k = M_k/R$$

When mapped onto a three-dimensional graph in which I_k , S_k , and C_k represent the three axes, Eq. 31 defines a planar
 20 region ABC as shown in FIG. 2. Planar region ABC contains all of the possible states of request interval k such that any observed point $Z(I_k, S_k, C_k)$ falls on the planar region ABC.

Within the planar region ABC, certain points are more likely than other points as outcomes of request interval k . Assuming the
 25 offered load estimation model above is accurate, the most likely points within planar region ABC are those representing the expected number of SUCCESS, IDLE, and COLLISION outcomes according to Eq. 13, Eq. 14, and Eq. 15, respectively, shown as curve L in FIG. 3A. Thus, the curve L describes a locus of
 30 expected outcomes. For convenience, the planar region ABC and the curve L are shown in two-dimensional view in FIG. 3B. It should be noted that the maximum probability of SUCCESS is at

the point P^* , which maps to $S_k = 0.368$, $I_k = 0.368$, and $C_k = 0.264$ in a preferred embodiment described in detail below.

One important attribute of the planar region ABC is that the probability of a particular point is inversely proportional to its
 5 distance from the curve L (i.e., the closer the point is to the curve L, the higher the probability). Thus, the planar region ABC can be divided into region(s) having "trusted" points and region(s) having "untrusted" points based generally on the distance of each point from the curve L.

10 In one embodiment, the planar region ABC is divided solely on the distance from the curve L. Fig. 4 shows a two-dimensional view of the planar region ABC, with the planar region ABC divided into three regions according to the distance from the curve L. Those points falling within a predetermined distance from the
 15 curve L (i.e., region 2) are considered to be "trusted" points, while all other points (i.e., regions 1 and 3) are considered to be "untrusted" points. While region 2 captures all points meeting at least a predetermined minimum probability, it is not an easy region to work with, since it is computationally complex to
 20 determine whether a particular point falls within the region.

In another embodiment, the planar region ABC is divided according to its intersection with three planes $S_k = S_0$, $I_k = I_0$, and $C_k = C_0$, as shown in three-dimensional view in FIG. 5. The three planes intersect at point P^* if $S_0 = 0.368 \times M_k/R$, $I_0 = 0.368 \times M_k/R$,
 25 and $C_0 = 0.264 \times M_k/R$, as shown in two-dimensional view in FIG. 6. Region $BB'P^*E$, corresponds to the state of obtaining many IDLE outcomes and few COLLISION outcomes in the request interval k, which is reasonably probable if the effective offered load within the frame is low. Region $CC'P^*D$ corresponds to the
 30 state of obtaining many COLLISION outcomes and few IDLE outcomes in the request interval k, which is reasonably probable if the effective offered load within the frame is high. Region EP^*D corresponds to the state of obtaining many COLLISION

outcomes, many IDLE outcomes, and few SUCCESS outcomes in the request interval k , which is improbable irrespective of the effective offered load. Region AB'C' corresponds to the state of obtaining many SUCCESS outcomes (i.e., with probability greater than 0.368) with few COLLISION and IDLE outcomes, which is desirable but improbable if the offered load estimation model is accurate.

Except for the point P^* , which falls in all regions, all points on the curve L fall within either region BB'P'E or CC'P'D. Thus, regions BB'P'E and CC'P'D are good candidates for containing "trusted" points. However, there are also points within regions AB'C' and EP'D that are close to the curve L and are therefore likely to be "trusted" points. In order to capture those "trusted" points, the three planes are redefined such that $S_0 = 0.4 \times M_k/R$, $I_0 = 0.4 \times M_k/R$, and $C_0 = 0.3 \times M_k/R$, as shown in two-dimensional view in FIG. 7. As a result, plane S_0 now falls above point P^* , and planes I_0 and C_0 intersect well below point P^* at point X . Points that fall within either region AB'C' (i.e., $S_k > S_0$) or region EXD (i.e., $C_k > C_0$ and $I_k < I_0$) are considered to be "untrusted" points, while points that fall within region BB'C'CDXE are considered to be "trusted" points.

2. Some Applications Utilizing Estimated Offered Load

As discussed above, the problem of estimating offered load in a communication network is a generic problem with many applications. One important application utilizes the estimated offered load to improve access performance in a shared medium communication network. Specifically, the estimated offered load is used for determining certain operating parameters such as the number of request transmission opportunities per frame and certain access mode parameters that affect how the network is accessed (described in more detail below).

FIG. 8 shows a shared medium communication network 100 in accordance with a preferred embodiment of the present invention. The shared medium communication network 100 allows a number of end users 110₁ through 110_N to access a remote external network 108 such as the Internet. The shared medium communication network 100 acts as a conduit for transporting information between the end users 110 and the external network 108.

The shared medium communication network 100 includes a primary station 102 that is coupled to the external network 108. The primary station 102 is in communication with a plurality of secondary stations 104₁ through 104_N (collectively referred to as "secondary stations 104" and individually as a "secondary station 104") by means of channels 106 and 107. Channel 106 carries information in a "downstream" direction from the primary station 102 to the secondary stations 104, and is hereinafter referred to as "downstream channel 106." Channel 107 carries information in an "upstream" direction from the secondary stations 104 to the primary station 102, and is hereinafter referred to as "upstream channel 107." Each end user 110 interfaces to the shared medium communication network 100 by means of a secondary station 104.

In a preferred embodiment, the shared medium communication network 100 is a data-over-cable (DOC) communication system wherein the downstream channel 106 and the upstream channel 107 are separate channels carried over a shared physical medium. In the preferred embodiment, the shared physical medium is a hybrid fiber-optic and coaxial cable (HFC) network. The downstream channel 106 is one of a plurality of downstream channels carried over the HFC network. The upstream channel 107 is one of a plurality of upstream channels carried over the HFC network. In other embodiments, the shared physical medium may be coaxial cable, fiber-optic cable, twisted

pair wires, and so on, and may also include air, atmosphere, or space for wireless and satellite communication. Also, the various upstream and downstream channels may be the same physical channel, for example, through time-division

5 multiplexing/duplexing, or separate physical channels, for example, through frequency-division multiplexing/duplexing.

In the shared medium communication network 100 of the preferred embodiment, the downstream channels, including the downstream channel 106, are typically situated in a frequency
10 band above approximately 50 MHz, although the particular frequency band may vary from system to system, and is often country-dependent. The downstream channels are classified as broadcast channels, since any information transmitted by the primary station 102 over a particular downstream channel, such
15 as the downstream channel 106, reaches all of the secondary stations 104. Any of the secondary stations 104 that are tuned to receive on the particular downstream channel can receive the information.

In the shared medium communication network 100 of a
20 preferred embodiment, the upstream channels, including the upstream channel 107, are typically situated in a frequency band between approximately 5 through 42 MHz, although the particular frequency band may vary from system to system, and is often country-dependent. The upstream channels are classified as
25 shared channels, since only one secondary station 104 can successfully transmit on a particular upstream channel at any given time, and therefore the upstream channels must be shared among the plurality of secondary stations 104. If more than one of the secondary stations 104 simultaneously transmit on a
30 particular upstream channel, such as the upstream channel 107, there is a collision that corrupts the information from all of the simultaneously transmitting secondary stations 104.

In order to allow multiple secondary stations 104 to share a particular upstream channel, such as the upstream channel 107, the primary station 102 and the secondary stations 104 participate in a medium access control (MAC) protocol. The MAC protocol
5 provides a set of rules and procedures for coordinating access by the secondary stations 104 to the shared upstream channel 107. Each secondary station 104 participates in the MAC protocol on behalf of its end users. For convenience, each participant in the MAC protocol is referred to as a "MAC User."

10 MAC protocols fall into two basic categories: contention-free and contention-based protocols.

In contention-free protocols, end users access a shared channel in a controlled manner such that transmissions are scheduled either statically, or adaptively so that collisions are
15 completely avoided. With static scheduling, such as that of a Time Division Multiple Access (TDMA) scheme, a predetermined transmission pattern is repeated periodically. The users may access channel resources only during the time intervals assigned to them individually. Contention-free protocols with static scheduling for
20 resource allocation is inefficient for a cable network supporting a large number of users where, typically, only a fraction of the users are active at any time. With adaptive scheduling, the transmission pattern may be modified in each cycle to accomodate dynamic traffic demand, via reservations or token passing. A fraction of the multiple
25 access channel, or a separate channel, is used to support the overhead due to reservation, or token passing. A reservation scheme typically requires a centralized controller to manage the reservations. A token passing scheme, on the other hand, is usually implemented in a distributed manner. Contention-free protocols with adaptive
30 scheduling are sometimes referred to as demand assignment multiple access.

In contention-based protocols, users contend with one another to access channel resources. Collisions are not avoided by design,

but are either controlled by requiring retransmissions to be randomly delayed, or resolved using a variety of other contention resolution strategies. The broadcast capability of a network, such as an HFC cable network, can often be taken advantage for simplified control in the MAC layer. One approach for delaying retransmissions is a binary exponential backoff approach, wherein a backoff window limits the range of random backoff, and an initial backoff window is doubled in successive attempts for retransmission. As the binary exponential backoff approach is known to lead to instability in heavy load, the maximum number of retransmissions for a request can be used to truncate the otherwise indefinite backoff.

Most contention-based protocols resolve collisions by using feedback information on the number of users involved in the collisions. If the number of conflicting transmissions can be determined from the feedback, then channel throughput arbitrarily close to one packet per packet transmission time is known to be achievable in principle, but with intractable complexity. More often than not, for the sake of simplicity, feedback information used is ternary indicating zero, one, or more transmissions, or binary indicating exactly one transmission or otherwise.

An example of a contention-based protocol is known as an ALOHA multiple access protocol. Its original version, which operates with continuous or unslotted time, is referred to as Unslotted ALOHA. Another version, which operates with discrete or slotted time, is referred to as Slotted ALOHA. The behavior and performance of Unslotted and Slotted ALOHA have been studied widely, and their maximum throughputs are well known to be $1/(2e)$ and $1/e$, respectively.

One type of MAC protocol suitable for HFC cable networks utilizes a reservation system in which each MAC User that wants to transmit data on the shared channel is required to make a reservation. In contention-free protocols with adaptive scheduling, users with pending transmissions must reserve transmission

resources. The protocol for reservation is itself a multiple access protocol.

In the shared medium communication network 100, many users share the upstream channel 107 for transmissions to the primary station 102. However, at any time, it is likely that only a fraction of these users are actually busy. If a contention-free protocol with static scheduling (e.g., TDMA) is used, resources allocated to idle users are wasted. This inefficiency is particularly intolerable when the load in the system is light. Contention-based protocols behave well under light load, but have limited throughput when offered load is high due to excessive collisions.

The throughput of a reservation-based system is limited by the percentage of available bandwidth that is allocated to the reservation control channel. One approach for reducing the demand on the reservation channel is to allocate a small field in the data packets for piggy-backing additional requests (i.e., including a request along with the transmitted data).

During steady-state operation, the number of users waiting to make reservations is typically small, particularly when piggybacking is permitted. It is therefore advantageous for the reservation protocol be a contention-based protocol with contention resolution. Unlike in a conventional contention-based protocol, the users typically do not contend with data packets, but instead contend with special reservation packets that are considerably smaller than data packets.

In a multiple access system with contention-based reservation, each MAC User that has data to transmit but has not already made a reservation waits for contention opportunities provided by the primary station 102. Each contention opportunity is provided by the primary station to a selected group of MAC Users, and allows each of the MAC Users in the specified group to contend for a reservation at a specific time, provided the MAC User has data to send.

Following each contention opportunity, the primary station 102 monitors for contention by the MAC Users and determines the

outcome of contention for each contention opportunity, specifically whether no MAC User contended, exactly one MAC User contended, or more than one MAC User contended. For convenience, the contention outcomes are referred to as IDLE, SUCCESS, and
 5 COLLISION, respectively. The primary station 102 then sends feedback information to the MAC Users indicating the outcome of contention for each contention opportunity. The feedback information allows each MAC User to determine, among other things, whether or not its own contention attempt was successful, and hence whether its
 10 request for bandwidth reservation has been accepted.

The advantage of a reservation system over a simple TDMA system is derived from the fact that request packets used for reservation, either in contention mode or a time division mode, are considerably smaller than most data packets. In a contention-based
 15 reservation system, the bandwidth wasted due to IDLE or COLLISION outcomes is relatively small compared to the bandwidth used for actual data transmission. Suppose that the ratio of the size of a request packet to that of a data packet is $v \ll 1$, and that a simple Slotted ALOHA multiple access scheme is used in the logical
 20 control channel for contention-based reservation. Then, it can be verified that the maximum throughput, S , in data packets per time unit, achievable in such a system is given by:

$$\text{Eq. 32} \quad S = \frac{1}{1+v/S_r} = \frac{1}{1+ve}$$

25

where S_r is the maximum throughput for Slotted ALOHA, which is equal to $1/e$ (see Bertsekas and Gallager, Data Networks, Section 4.5, Prentice-Hall, 1987).

A reservation-based MAC protocol can be represented at a high
 30 level by a state diagram for each MAC User as shown in FIG. 9. The MAC User starts in the INACTIVE state, and remains there so long as it has no data to transmit or it is waiting for an opportunity to transmit

a request. When the MAC User receives data to be transmitted, the MAC User transitions into the ACTIVE state upon receiving a contention-free opportunity to transmit a request, provided it is not required to contend for upstream bandwidth, as in the case of unicast polling. Otherwise, the MAC User transitions into the CONTENTION state upon receiving and transmitting a request in a transmission opportunity for contention. In the CONTENTION state, the MAC User contends for access to the channel until it is able to make a successful reservation for itself, or until its request is rejected due to system overload. Upon making a successful reservation in this state, the MAC User transitions into the ACTIVE state. Here, the MAC User receives opportunities to transmit its data, and remains in the ACTIVE state so long as it has data to transmit. During system overload, a requests pending contention resolution may be denied further retransmission opportunities after a predetermined number of attempts. When this happens, the MAC User moves from the CONTENTION state to the INACTIVE state. If new data arrives while the MAC User is in the ACTIVE state, the MAC User may be permitted to include a piggyback request in the data it transmits. Upon transmitting all of its data, the MAC User transitions back into the INACTIVE state.

For each request transmission opportunity provided by the primary station 102, the primary station receives either (1) no transmission, indicating that no MAC User transmitted a reservation request; (2) a reservation request, indicating that a single MAC User transmitted a reservation request and identifying that MAC User; or (3) a collision, indicating that more than one MAC User transmitted a reservation request. For convenience, the three feedback states are referred to as IDLE, SUCCESS, and COLLISION, respectively.

The primary station 102 schedules future request transmission opportunities and data transmission opportunities based on the result of the contention-based reservation. If a successful reservation is made (i.e., if the result of the contention is SUCCESS), then the

primary station 102 allocates bandwidth to the MAC User based on the QoS requirements of the corresponding end user so that the MAC User can transmit user information contention-free over the shared channel. On the other hand, if multiple MAC Users respond (i.e., if
5 the result of the contention is COLLISION), then the primary station 102 attempts to aid in resolving the collision by providing additional request transmission opportunities.

In a preferred embodiment, the MAC protocol includes a protocol commonly referred to as Multimedia Cable Network
10 System (MCNS), which is defined in the document entitled MCNS Data-Over-Cable Service Interface Specifications Radio Frequency Interface Specification SP-RFI-102-971008 Interim Specification (hereinafter referred to as the "MCNS Protocol Specification"), incorporated herein by reference in its entirety. In
15 the MCNS Protocol Specification, the primary station 102 is referred to as a Cable Modem Termination System (CMTS), and the secondary stations 104 are referred to as Cable Modems (CMs). The CMTS is responsible for packet processing, resource sharing, and management of the MCNS MAC and Physical layer
20 functions. Each CM operates as a slave to the CMTS. MAC Protocol Data Units (PDUs) transmitted on the downstream channel 106 by the CMTS may be addressed to an individual CM via unicast, or to a selected group of CMs via multicast or broadcast. In the upstream channel, a MAC PDU may be sent by
25 any CM to the CMTS. MCNS supports variable length MAC PDUs.

The MCNS Protocol Specification utilizes a slotted upstream channel, such that the upstream channel 107 is divided into successive time slots. The MAC protocol supports a plurality of
30 slot types for carrying different types of information. Each time slot is capable of transporting a unit of information (for example, a data packet or a control packet). The MCNS Protocol Specification further divides the upstream channel 107 into

successive frames, where each frame includes a number of slots. The CMTS allocates bandwidth to a group of CMs by transmitting on the downstream channel 106 a control message containing a bandwidth allocation information element known as a MAP. The
5 MAP specifies the allocation of transmission opportunities within a given transmission frame. Bandwidth is allocated, frame by frame, in terms of transmission opportunities for contention-based reservation requests (or simply requests) as well as for user data. A successful transmission in a contention opportunity results in
10 the reservation of a future data transmission opportunity.

More specifically, the upstream channel 107 is modeled as a stream of mini-slots, providing for TDMA at regulated time ticks. The use of mini-slots implies strict timing synchronization between the CMTS and all the CMs. Hence, the CMTS is responsible for
15 generating the time reference to identify these mini-slots and periodically allow for ranging opportunities so that all CMs maintain their synchronization. The access to the mini-slots by the CMs is controlled by the CMTS. To accomplish that, the CMTS transmits on the downstream channel a MAP describing the use of each upstream
20 mini-slot in a specified future time interval. This message, in a way, "maps" in a future time interval each mini-slot to its use. Of course, the MAP has to be sent by the CMTS earlier than the effective time interval that it describes in order to allow enough time for the CMs to transmit in the mapped mini-slots.

25 In the MCNS Protocol Specification, each frame is organized into discrete intervals. At least three different interval types are defined. A request interval includes a number of mini-slots that are allocated for transmitting requests (or small data packets) in contention mode. A maintenance interval includes a number of mini-
30 slots allocated for registration of CMs. A data grant interval includes a number of mini-slots allocated for transmitting data packets. The MAP includes a number of information elements (IEs) that define the different intervals in the frame.

FIG. 10 is a block diagram showing an exemplary primary station 102 in accordance with a preferred embodiment of the present invention. In the preferred embodiment, the primary station 102 includes a number of functional modules implemented on individual cards that fit within a common chassis. In order to enable communication within the shared medium communication network 100, the primary station 102 requires at least a minimum set of functional modules. Specifically, the minimum set of functional modules comprises an Adapter Module 210, a MAC Module 220, a Transmitter Module 240, and a Receiver Module 230. In the preferred embodiment, the minimum set of functional modules allows the primary station 102 to support a single downstream channel and up to eight upstream channels. For the sake of convenience and simplicity, the exemplary embodiments described below refer to the single upstream channel 107, although it will be apparent to a skilled artisan that multiple upstream channels are supportable in a similar manner.

The Adapter Module 210 controls the flow of data and control messages between the primary station 102 and the secondary stations 104. The Adapter Module 210 includes Control Logic 218 that is coupled to a Memory 212. The Control Logic 218 includes, among other things, logic for processing data and control (e.g., request) messages received from the secondary stations 104, and logic for generating data and control (e.g., MAP) messages for transmission to the secondary stations 104. The Memory 212 is divided into a Dedicated Memory 216 that is used only by the Control Logic 218, and a Shared Memory 214 that is shared by the Control Logic 218 and MAC Logic 224 (described below) for exchanging data and control messages.

The Control Logic 218 and the MAC Logic 224 exchange data and control messages using three ring structures (not shown) in the Shared Memory 214. Data and control (e.g., request) messages received from the secondary station 104 are

stored by the MAC Logic 224 in a Receive Queue in the Shared Memory 224. Control (e.g., MAP) messages generated by the Control Logic 218 are stored by the Control Logic 218 in a MAC Transmit Queue in the Shared Memory 214. Data messages for transmission to the secondary station 104 are stored by the Control Logic 218 in a Data Transmit Queue in the Shared Memory 214. The Control Logic 218 monitors the Receive Queue to obtain data and control (e.g., request) messages. The MAC Logic 224 monitors the MAC Transmit Queue to obtain control (e.g., MAP) messages, and monitors the Data Transmit Queue to obtain data messages.

The MAC Module 220 implements MAC functions within the primary station 102. The MAC Module 220 includes MAC Logic 224 that is coupled to a Local Memory 222 and to the Shared Memory 214 by means of interface 250. The MAC Logic 224 monitors the MAC Transmit Queue and the Data Transmit Queue in the Shared Memory 214. The MAC Logic 224 transmits any queued data and control (e.g., MAP) messages to Encoder/Modulator 241 of Transmitter Module 240 by means of interface 253. The MAC Logic 224 also processes data and control (e.g., request) messages received from the Receiver Module 230 by means of interface 255. The MAC Logic 224 stores the received data and control messages in the Receive Queue in the Shared Memory 214 by means of interface 250.

The Transmitter Module 240 provides an interface to the downstream channel 106 for transmitting data and control (e.g., MAP) messages to the secondary stations 104. The Transmitter Module 240 includes a Transmitter Front End 242 that is operably coupled to the downstream channel 106 and an Encoder/Modulator 241. The Encoder/Modulator 241 includes logic for processing data and control (e.g., MAP) messages received from the MAC Logic 224 by means of interface 253. More specifically, the Encoder/Modulator 241 includes encoding

logic for encoding the data and control (e.g., MAP) messages according to a predetermined set of encoding parameters, and modulating logic for modulating the encoded data and control (e.g., MAP) messages according to a predetermined modulation mode. The Transmitter Front End 242 includes logic for transmitting the modulated signals from the Encoder/Modulator 241 onto the downstream channel 106. More specifically, the Transmitter Front End 242 includes tuning logic for tuning to a downstream channel 106 center frequency, and filtering logic for filtering the transmitted modulated signals. Both the Encoder/Modulator 241 and the Transmitter Front End 242 include adjustable parameters, including downstream channel center frequency for the Transmitter Front End 242, and modulation mode, modulation symbol rate, and encoding parameters for the Encoder/Modulator 241.

The Receiver Module 230 provides an interface to the upstream channel 107 for receiving, among other things, data and control (e.g., request) messages from the secondary stations 104. The Receiver Module 230 includes a Receiver Front End 232 that is operably coupled to the upstream channel 107 and to a Demodulator/Decoder 231. The Receiver Front End 232 includes logic for receiving modulated signals from the upstream channel 107. More specifically, the Receiver Front End 232 includes tuning logic for tuning to an upstream channel 107 center frequency, and filtering logic for filtering the received modulated signals. The Demodulator/Decoder 231 includes logic for processing the filtered modulated signals received from the Receiver Front End 232. More specifically, the Demodulator/Decoder 231 includes demodulating logic for demodulating the modulated signals according to a predetermined modulation mode, and decoding logic for decoding the demodulated signals according to a predetermined set of decoding parameters to recover data and control (e.g., request)

messages from the secondary station 104. Both the Receiver Front End 232 and the Demodulator/Decoder 231 include adjustable parameters, including upstream channel center frequency for the Receiver Front End 232, and modulation mode, modulation symbol rate, modulation preamble sequence, and decoding parameters for the Demodulator/Decoder 231.

In the preferred embodiment, the primary station 102 includes a configuration interface 254 through which the adjustable parameters on both the Receiver Module 230 and the Transmitter Module 240 are configured. The configuration interface 254 operably couples the MAC Logic 224 to the Demodulator/Decoder 231, the Receiver Front End 232, the Encoder/Modulator 241, and the Transmitter Front End 242. The configuration interface 254 is preferably a Serial Peripheral Interface (SPI) bus as is known in the art.

FIG. 11 is a block diagram showing an exemplary secondary station 104 in accordance with a preferred embodiment of the present invention. The secondary station 104 includes a User Interface 310 for interfacing with the End User 110. Data transmitted by the End User 110 is received by the User Interface 310 and stored in a Memory 308. The secondary station 104 also includes a Control Message Processor 304 that is coupled to the Memory 308. The Control Message Processor 304 participates as a MAC User in the MAC protocol on behalf of the End User 110. Specifically, the Control Message Processor 304 transmits data and control (e.g., request) messages to the primary station 102 by means of Transmitter 302, which is operably coupled to transmit data and control (e.g., request) messages on the upstream channel 107. The Control Message Processor 304 also processes data and control (e.g., MAP) messages received from the primary station 102 by means of Receiver 306, which is operably coupled to receive data and control (e.g., MAP) messages on the downstream channel 106.

An important consideration that affects performance in the MCNS MAC protocol is the number of mini-slots allocated to the request interval in each frame. Assuming that the number of slots per frame T_k is substantially constant, the number of mini-slots allocated to the request interval affects the number of mini-slots allocated to the other intervals, particularly the data interval. A large number of mini-slots allocated to the request interval decreases the likelihood of collisions, but also decreases the number of mini-slots allocated for transmitting data and therefore decreases the data throughput of the system. Furthermore, a small number of mini-slots allocated to the request interval can increase the likelihood of collisions and therefore decrease the data throughput of the system by preventing successful requests from reaching the CMTS. Preferably, the number of slots in the request interval is selected to maximize the likelihood of SUCCESS outcomes. This typically involves increasing the number of slots in the request interval if the offered load is high, and decreasing the number of slots in the request interval if the offered load is low. Thus, the offered load is a key consideration in selecting the number of slots per request interval.

Another important consideration that affects performance in the MCNS MAC protocol is the type of contention access used. In accordance with the MCNS Protocol Specification, at least two types of contention access is supported. In a first type of contention access, the secondary stations 104 are only permitted to transmit request messages during the request interval. In a second type of contention access, the secondary stations 104 are permitted to transmit either request messages or small data messages during the request interval. The second type of contention access can improve performance when there are few collisions, but can decrease performance when there are many collisions. Therefore, the second type of contention access would only be utilized when the actual offered load is low, where the first type of contention access would be used when the actual offered load is high. Thus, the offered load is a

key consideration in selecting the type of contention access in the MCNS MAC protocol.

In the MCNS MAC protocol, the offered load is not known *a priori*. Therefore, the offered load must be estimated using either the sample window technique or the single frame technique described herein, typically by the Control Logic 218 as an element its MAP generating logic.

A. Application of the Sample Window Technique

The estimator function g' from Eq. 30 was derived using a number of variables. These include the variable n representing the sample window size; the variable x representing the number of weighted frames in the sample window; and the variable X representing the percentage variable that is used in Eq. 24 to derive the weighting factor α .

In accordance with a preferred embodiment of the present invention, the variable n is equal to 16 frames. The variable n to be large enough that there is a statistically significant number of request transmission opportunities in the sample window, and yet not so large that the offered load varies significantly over the sample window. With reference to Eq. 21, it was accepted heuristically that the instantaneous offered load g_i for each frame in a sample window does not vary considerably over the sample window, and therefore that the instantaneous offered load g_i can be approximated by an offered load g that is the same for each sample window frame i (i.e., $g_1 = g_2 = \dots = g_n = g$). Simulations of the MCNS MAC protocol have shown that a frame seldom exceeds 5 milliseconds in duration, and the offered load variation during a 100 millisecond time interval is typically less than ten percent of its original value. Therefore, the instantaneous offered load can be approximated by g provided that the sample window size n is less than approximately 20 frames.

In accordance with a preferred embodiment of the present invention, the variable x is equal to 3, and the variable X is equal to

0.4 such that α is equal to 3. The selection of these parameter values will become apparent by understanding why a sliding window approach with weighting is used.

It should be recalled that a first possibility was to update the
5 estimated offered load at the end of each sample window. This approach is very inaccurate, and can lead to oscillations of the estimated offered load around the actual offered load. To understand why, assume that at the end of some sample window j , the offered load is underestimated. Since the estimated offered load is used to
10 select the request interval size for each frame in the next sample window ($j+1$), the CMTS will set the request interval size for each frame in the next sample window ($j+1$) to a value smaller than it should be. This will lead to collisions in frame 1 of sample window ($j+1$). As these collided requests will be retransmitted in the frame 2
15 of sample window ($j+1$), the probability of collision is further increased in frame 2. The process will go on until the end of the sample window ($j+1$), eventually leading to an extremely high probability of collision at the end of the window, and hence a very small number of IDLE outcomes. As a result, the estimated offered load will increase
20 significantly, leading to an over-estimation of the offered load. Similarly, this over-estimation yields request interval sizes in sample window ($j+2$) larger than they should be, resulting in a large number of IDLE outcomes at the end of sample window ($j+2$). The result is an under-estimation of the offered load, causing the cycle to repeat and
25 therefore causing the estimated offered load to fluctuate around the actual offered load.

It should also be recalled that a second possibility was to use a sliding sample window and to update the estimated offered load at the end of each frame without using weighting. While
30 simulation results have shown this approach to be significantly better than updating the estimated offered load at the end of each sample window, this approach is also susceptible to fluctuations around the actual offered load. To understand why, assume that

at the end of some frame k , the offered load is under-estimated. Therefore, the request interval in the next frame $(k+1)$ will be smaller than it should be, resulting in a higher probability of COLLISION outcomes in frame $(k+1)$. However, because the
5 estimated offered load is determined over a window of 16 frames, the increased probability of COLLISION outcomes in frame $(k+1)$ will have little impact on the estimated offered load, so that the estimated offered load is likely to increase slightly but remain under-estimated. This under-estimation will cause more collisions
10 in frame $(k+2)$ and subsequent frames, as the actual offered load increases faster than the estimated offered load due to an increased number of retransmissions. After several frames, the sample window will start including more and more of those frames having a large number of COLLISION outcomes, leading to an
15 over-estimation of the offered load. Once the estimated offered load is over-estimated, the request interval sizes will be set bigger than they should be, resulting in many IDLE outcomes. Again, because the estimated offered load is determined over a window of 16 frames, the increased number of IDLE outcomes in the
20 subsequent frames will have little impact on the estimated offered load, so that the estimated offered load is likely to decrease slightly but remain over-estimated. After several frames, the sample window will start including more and more of those frames having a large number of IDLE outcomes, leading again to an
25 under-estimation of the offered load causing the cycle to repeat and therefore causing the estimated offered load to fluctuate around the actual offered load.

These examples demonstrate that fluctuations can occur about the actual offered load when the response to changes in the
30 actual offered load is too slow. With reference to the sample window approach, the slow response problem would not exist if the sample window is small. However, reducing the size of the sample window also reduces the number of request transmission

opportunities in the sample window, which degrades the accuracy of the estimation. Thus, the sample window must remain relatively large, but the estimated offered load must adapt quickly.

The solution of course is to add a weighting factor that
5 emphasizes the most recent frames and yet still considers a large number of frames. As discussed above, the x most recent frames are given a weighting factor of α , while the older $(n-x)$ frames are given a weighting factor of β that is arbitrarily set to one. The weighting factors must be selected appropriately to obtain fast
10 response and an accurate estimated offered load. If the ratio α/β is too small or if x is too large, then not enough weight is given to the most recent frames, and therefore the estimated offered load will adapt too slowly. On the other hand, if the ratio α/β is too large, then too much weight is given to the most recent frames,
15 and therefore the estimated offered load will be inaccurate.

Simulation results have shown that good performance is obtained when x is equal to 3 and X is equal to 0.4, such that α is equal to 3. This selection of x is consistent with the expected operation of the MCNS MAC protocol. Because the starting
20 backoff window size does not exceed M_k/R and the ending backoff window size does not exceed $2xM_k/R$, any retransmissions from collisions in frame $(n-2)$ will all show, with a high probability, in frames $(n-1)$ and n . For the same reason, the effect of collisions in frames prior to $(n-2)$ is minimal in frame n . Therefore, frames
25 $(n-2)$, $(n-1)$, and n are most representative of the actual offered load for frame $(n+1)$, and therefore are weighed more heavily than the prior frames.

B. Application of the Single Frame Technique

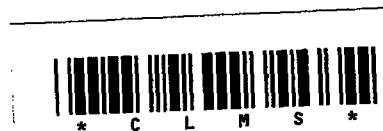
30 In accordance with a preferred embodiment of the present invention, the Control Logic 218 determines the number of IDLE, SUCCESS, and COLLISION outcomes during the request interval k ,

referred to as I_k , S_k , and C_k , respectively. Based on the number of IDLE, SUCCESS, and COLLISION outcomes during the request interval k , the Control Logic 218 decides whether the outcomes represents a "trusted" or "untrusted" point. Specifically, the outcomes
5 are deemed to represent an "untrusted" point and are ignored if $S_k > S_0$ or if $C_k > C_0$ and $I_k < I_0$ (where $S_0 = 0.4 \times M_k/R$, $I_0 = 0.4 \times M_k/R$, and $C_0 = 0.3 \times M_k/R$). Otherwise the outcomes are deemed to represent a "trusted" point and are used to update the estimated offered load.

All logic described herein can be embodied using discrete
10 components, integrated circuitry, programmable logic used in conjunction with a programmable logic device such as a Field Programmable Gate Array (FPGA) or microprocessor, or any other means including any combination thereof. Programmable logic can be fixed temporarily or permanently in a tangible
15 medium such as a read-only memory chip, a computer memory, a disk, or other storage medium. Programmable logic can also be fixed in a computer data signal embodied in a carrier wave, allowing the programmable logic to be transmitted over an interface such as a computer bus or communication network. All
20 such embodiments are intended to fall within the scope of the present invention.

The present invention may be embodied in other specific forms without departing from the essence or essential characteristics. The described embodiments are to be considered
25 in all respects only as illustrative and not restrictive.

We claim:





1. A method for selecting a request interval size in a communication network, the method comprising the steps of:
 - estimating an offered load in the network based on at least one set of contention outcomes; and
- 5 selecting the request interval size based on the estimated offered load.