Quality-of-Service in Ad Hoc Carrier Sense Multiple Access Wireless Networks

João L. Sobrinho, *Member, IEEE*, and A. S. Krishnakumar, *Member, IEEE*

***Abstract*—Carrier sense multiple access (CSMA) is one of the most pervasive medium access control (MAC) schemes in ad hoc wireless networks. However, CSMA and its current variants do not provide quality-of-service (QoS) guarantees for real-time traffic support. This paper presents and studies black-burst (BB) contention, which is a distributed MAC scheme that provides QoS real-time access to ad hoc CSMA wireless networks. With this scheme, real-time nodes contend for access to the channel with pulses of energy—so called BB's—the durations of which are a function of the delay incurred by the nodes until the channel became idle. It is shown that real-time packets are not subject to collisions and that they have access priority over data packets. When operated in an ad hoc wireless LAN, BB contention further guarantees bounded and typically very small real-time delays. The performance of the network can approach that attained under ideal time division multiplexing (TDM) via a distributed algorithm that groups real-time packet transmissions into chains. A general analysis of BB contention is given, contemplating several modes of operation. The analysis provides conditions for the scheme to be stable. Its results are complemented with simulations that evaluate the performance of an ad hoc wireless LAN with a mixed population of data and real-time nodes.**

***Index Terms*—Ad hoc wireless networks, carrier sense multiple access (CSMA), quality-of-service (QoS), real-time traffic.**

I. Introduction

A KEY component in the development of single channel ad hoc wireless networks is the medium access control (MAC) protocol with which nodes share a common radio channel. Of necessity, such a protocol has to be distributed. It should provide an efficient use of the available bandwidth while satisfying the quality-of-service (QoS) requirements of both data and real-time applications. Carrier sense multiple access (CSMA) [1] is one of the most pervasive MAC schemes in ad hoc wireless networks [2]. CSMA is a simple distributed protocol whereby nodes regulate their packet transmission attempts based only on their local perception of the state—idle or busy—of the common radio channel.

Packet collisions are intrinsic to CSMA. They occur because each node has only a delayed perception of the other nodes' activity. They also happen due to hidden nodes [3]: two transmitting nodes outside the sensing range of each other may interfere at a common receiver. Many flavors of CSMA do exist, but invariably the nodes that participate in a collision schedule the retransmission of their packets to a random time in the future, in the hope of avoiding another collision. This strategy, however, does not provide QoS guarantees for real-time traffic support.

Recently, new MAC schemes for ad hoc wireless networks have been proposed, aimed either at improving the throughput over that of CSMA or at providing QoS guarantees for real-time traffic support. Among the first group of schemes is the multiple access collision avoidance protocol (MACA) [4], which forms the basis of several other schemes. With MACA, a source with a packet ready for transmission first sends a request-to-send (RTS) minipacket, which if successful elicits a clear-to-send (CTS) minipacket from the destination. Upon reception of the CTS minipacket, the source sends its data packet. In environments without hidden nodes, MACA may improve the throughput of the network over that attained with CSMA because collisions involve only short RTS minipackets rather than normal data packets as in CSMA. MACA also alleviates the hidden nodes problem because the CTS sent by the destination serves to inhibit the nodes in its neighborhood, i.e., exactly those nodes that may interfere with the ensuing packet transmission from source to destination. The floor acquisition multiple access (FAMA) class of protocols [5] includes several variants of MACA, one of which is immune to hidden nodes [6]. These protocols, however, have not been designed for QoS: control minipackets are subject to collisions, and their retransmissions are randomly scheduled.

The group allocation multiple access (GAMA) [7], [8] is a recent attempt to provide QoS guarantees to real-time traffic in a distributed wireless environment. In GAMA, there is a contention period where nodes use an RTS-CTS dialog to explicitly reserve bandwidth in the ensuing contention-free period. A packet transmitted in the contention-free period may maintain the reservation for the next cycle. The scheme is asynchronous and developed for wireless networks where all nodes can sense, and indeed receive, the communications from their peers. MACA/packet reservation (MACA/PR) [9] is a protocol similar to GAMA, but an acknowledgment follows every packet sent in contention-free periods to inform the nodes in the neighborhood of the receiver whether or not another packet is expected in the next contention-free cycle. These schemes deviate from pure carrier sensing methods in that every node has to construct channel state information based on reservation requests carried in packets sent onto the channel.

In this paper, we elaborate on the black-burst (BB) contention mechanism presented in [10]. With this mechanism,

Manuscript received May 14, 1998; revised February 11, 1999.

J. L. Sobrinho is with the Instituto de Telecomunicações, Lisboa 1049-001 Portugal (e-mail: joao.sobrinho@lx.it.pt).

A. S. Krishnakumar is with Bell Laboratories, Lucent Technologies, Murray Hill, NJ 07974 USA (e-mail: askbl@lucent.com).

Publisher Item Identifier S 0733-8716(99)04808-8.

real-time nodes contend for access to the common radio channel with pulses of energy—so called BB's—the lengths of which are proportional to the time that the nodes have been waiting for the channel to become idle. The scheme is distributed and is based only on carrier sensing. It gives priority access to real-time traffic and ensures collision-free transmission of real-time packets. When operated in an ad hoc wireless LAN, it further guarantees bounded real-time delays. In addition, the BB contention scheme can be overlaid on current CSMA implementations, notably that of the recent IEEE 802.11 standard [11] for wireless LAN's, with only minor modifications required to the real-time transceivers: the random retransmission scheme is turned off, and in substitution, the possibility of sending BB's is provided. This paper extends the initial work in a number ways. We prove the properties of BB contention for a general ad hoc wireless network without hidden nodes, but one in which the channel can be spatially reused. A scheme is devised that decouples the instants when real-time nodes gain exclusive access rights to the channel from the instants when real-time packets appear at the MAC layer for transmission over the channel. We also introduce enhancements to the efficiency of the scheme, when operated in an ad hoc wireless LAN, by allowing real-time packet transmissions from different nodes to be grouped into chains. Finally, a general analytical framework is presented to describe the dynamics of BB contention. The performance of BB contention is compared with that of the CSMA/collision avoidance (CSMA/CA) scheme of the IEEE 802.11 standard in an ad hoc wireless LAN with a mixed population of data and real-time nodes.

Section II presents a model for carrier sense wireless networks and briefly reviews the CSMA/CA protocol that is taken as a comparison example throughout this paper. The BB contention mechanism is described in Section III, and its properties are formally proved in Section IV. Section V addresses the interdependency between BB contention and the assembly of real-time packets. Chaining is described in Section VI. The analysis of BB contention is given in Section VII. The proofs in this section can be skipped without loss of continuity. The results are presented and discussed in Section VIII.

II. Carrier Sense Wireless Networks

Carrier sense wireless networks are engineered in such a way that the range at which a node can sense carrier from a given transmitter is different and typically larger than the range at which receivers are willing to accept a packet from that same transmitter. In addition, carrier from a transmitter can usually be sensed at a range beyond the range in which the transmitter may cause interference. To account for these differences, we model a wireless network as a set of nodes N, interconnected by links of three different types. Node i has a communication link with node j, if and only if in the course of time, it has packets to send to node j. Node i has an interfering link with node j if and only if any packet transmission with destination j that overlaps in time at j with a transmission from i is lost. The lost packets are said to have collided with the transmission from i. Finally, node i has a sensing link with node j, if and only if a transmission by node i prevents node j from starting a new transmission, i.e., node i inhibits node j. The communication, interference, and sensing graphs are denoted by $G_C = (N, L_C)$, $G_I = (N, L_I)$, and $G_S = (N, L_S)$, respectively, where L_C, L_I, and L_S are the edge sets (the links). The communication graph is a directed graph, whereas the interfering and sensing graphs are undirected. We assume that if node i has a communication link with node j, then i and j also have an interfering link between them. Similarly, an interfering link is also a sensing link, but not conversely. That is, $L_I \subset L_S$: G_I is a spanning subgraph of G_S. Any node has an interfering and sensing link with itself, since whenever a node transmits, it cannot simultaneously receive or start another transmission. In the wireless network of Fig. 1, node 9 has a communication link with node 10, and thus these nodes have both an interfering and a sensing link between them. Nodes 10 and 13 have an interfering link, and thus they also have a sensing link between them. Finally, nodes 9 and 13 have only a sensing link between them. We do not explicitly represent the links from a node to itself.

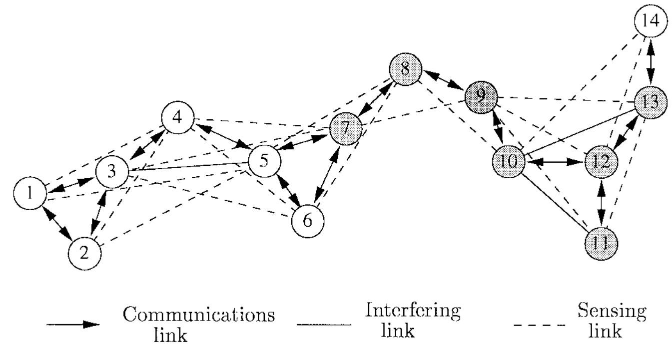


Fig. 1. A wireless network without hidden nodes. The shaded nodes form the set $N_S(9)$.

A path delay is associated with each sensing link to account for the propagation delay separating the nodes, the turn-around time of the wireless transceivers, and the sensing delay. The path delay of link ij is denoted by τ_{ij}. Since the sensing graph is undirected, $\tau_{ij} = \tau_{ji}$. The path delays further satisfy the two conditions $\tau_{ij} > 0$ and $\tau_{ik} + \tau_{kj} > \tau_{ij}$, for $ik, kj, ij \in L_S$. Let $\tau \triangleq \max(\tau_{ij})$.

The sets $N_I(i)$ and $N_S(i)$ represent the nodes that are neighbors of i, i included, in the interfering and sensing graphs, respectively. In Fig. 1, $N_I(10) = \{9, 10, 11, 12, 13\}$ and $N_S(9) = \{7, 8, 9, 10, 11, 12, 13\}$. For communication link ij, the set of nodes which are interfering neighbors of j but are not sensing neighbors of i, i.e., the set $N_I(j) \cap (N - N_S(i))$, is the set of nodes hidden from ij. A node in this set will not sense an ongoing packet transmission from i to j and may initiate its own packet transmission that will collide at j. In a wireless network without hidden nodes, we have $N_I(j) \subset N_S(i)$ for every $ij \in L_C$. The network of Fig. 1 does not have hidden nodes. Nevertheless, the common radio channel can be reused in space. For example, a packet transmission from node 9 to node 8 can coexist in time without collisions with a packet transmission from node 5 to node 7. The correctness results to be proven in this paper are for wireless networks without hidden nodes. We use the term wireless LAN for wireless networks in which $G_I = G_S$ forms a complete graph. In a wireless LAN, all nodes can sense each other's transmissions.

The CSMA/CA protocol of the IEEE 802.11 standard [11] defines three interframe spacings, $t_{\rm short}$, $t_{\rm med}$, $t_{\rm med} \geq 2\tau + t_{\rm short}$, and $t_{\rm long}$, $t_{\rm long} \geq 2\tau + t_{\rm med}$. If a node with a packet that is ready for transmission has perceived that the channel is idle during a long interframe spacing of length $t_{\rm long}$, the node immediately starts the transmission of the packet. Otherwise, it waits until that condition is satisfied and enters into backoff. Likewise, a node whose packet has experienced c consecutive collisions enters into backoff. In this mode, the node chooses a random number of slots s uniformly distributed between zero and $\min\{32 \times 2^c - 1, 255\}$ and sets a timer with an initial value $s \times t_{\rm slot}$ units of time, where $t_{\rm slot}$, $t_{\rm slot} \geq 2\tau$, is the length of a slot. The timer counts down only while the channel has been perceived idle for more than $t_{\rm long}$ units of time—it is frozen during a medium busy condition—and the packet is (re)transmitted as soon as the timer reaches zero. A node learns of the success or failure of its transmission through a positive acknowledgment scheme; the recipient of a correctly received packet sends back an acknowledgment minipacket within an interval of time of length $t_{\rm short}$.

III. BB Contention

BB contention is a MAC mechanism developed to provide QoS guarantees to real-time traffic over carrier sense wireless networks. The real-time applications considered are those like voice and video that require more or less periodic access to the common radio channel during long periods of time denominated sessions. The main performance requirement for these applications is bounded end-to-end delay, which implies a bounded packet delay at the MAC layer. This is the goal of BB contention.

The principles behind BB contention are more easily understood if we consider first a wireless LAN; the next section proves statements that are valid in more general cases. Real-time nodes contend for access to the channel after a medium interframe spacing of length $t_{\rm med}$, rather than after the long interframe spacing of length $t_{\rm long}$, used by data nodes. Thus, real-time nodes as a group have priority over data nodes. Instead of sending their packets when the channel becomes idle for $t_{\rm med}$, real-time nodes first sort their access rights by jamming the channel with pulses of energy, denominated BB's. The length of a BB transmitted by a real-time node is an increasing function of the contention delay experienced by the node, measured from the instant when an attempt to access the channel has been scheduled until the channel becomes idle for $t_{\rm med}$, i.e., until the node starts the transmission of its BB. To account for the path delays in the network, BB's are formed by an integral number of black slots, each of length $t_{\rm bslot}$, with $t_{\rm bslot}$ not smaller than the maximum round-trip path delay 2τ. Now, we would like the BB's sent by distinct real-time nodes when the channel becomes idle for $t_{\rm med}$ to differ by at least one black slot. To this end, we assume that every real-time packet transmission lasts at least a certain time $t_{\rm pkt}$ and that real-time nodes only schedule their next transmission attempts—to a time $t_{\rm sch}$ in the future—when they start a packet transmission. If a node starts a packet transmission at time u and that transmission is successful, that means that no other real-time node started a packet transmission during an interval of length $2t_{\rm pkt}$ around time u. Therefore, the next scheduled attempt made by the node in question is also staggered in time by $t_{\rm pkt}$ from the scheduled access attempts made by the other nodes. Counting the number of black slots to be sent in a BB in units of $t_{\rm pkt}$, we obtain the desired property that distinct nodes contend with BB's comprising different numbers of black slots. Following each BB transmission, a node senses the channel for an observation interval of length $t_{\rm obs}$ to determine without ambiguity whether its BB was the longest of the contending BB's. The winning node will transmit its real-time packet successfully and schedule the next transmission attempt. On the other hand, the nodes that lost the BB contention wait for the channel to once again become idle for $t_{\rm med}$, at which time they send new longer BB's. In conclusion, once the first real-time packet of a session is successfully transmitted, the mechanism ensures that succeeding real-time packets are also transmitted without collisions. In the end, real-time nodes appear to access a dynamic time division multiplexing (TDM) transmission structure without explicit slot assignments or slot synchronization. We follow with a detailed description of the access rules followed by every real-time node.

Every real-time packet lasts for at least a certain amount of time $t_{\rm pkt}$, $t_{\rm pkt} \geq 2\tau$, when transmitted on the channel. At the beginning of a session, a real-time node uses conventional CSMA/CA rules, possibly with a more expedited retransmission algorithm, to convey its first packet until it is successful. Subsequent packets are transmitted according to the mechanisms, herein described, until the session is dropped.

Whenever a real-time node transmits a packet, it further schedules its next transmission attempt to a time $t_{\rm sch}$ in the future, where $t_{\rm sch}$ is the same for all nodes. Suppose, then, that a real-time node has scheduled an access attempt for the present time. If the channel has been idle during the past medium interframe interval of length $t_{\rm med}$, the node starts the transmission of a BB. Otherwise, it waits until the channel becomes idle for $t_{\rm med}$ and only then starts the transmission of its BB. The length b of the BB sent by the node is a direct function of the contention delay it incurred, $d_{\rm cont}$

$$b(d_{\rm cont}) = \left(1 + \left\lfloor \frac{d_{\rm cont}}{t_{\rm unit}} \right\rfloor\right) t_{\rm bslot}$$

where $t_{\rm bslot}$ is the length of a black slot, the parameter $t_{\rm unit}$ is the unit of time used to convert contention delays into an integral number of black slots, and $\lfloor x \rfloor$ is the floor of x, i.e., the largest integer not larger than x. Correct operation of the scheme requires that $t_{\rm unit} \leq t_{\rm pkt}$. After exhausting its BB transmission, the node waits for an observation interval $t_{\rm obs}$, the length of which has to satisfy $t_{\rm obs} \leq t_{\rm bslot}$ and $t_{\rm obs} < t_{\rm med}$, to see if any other node transmitted a longer BB, implying that it would have been waiting longer for access to the channel. If the channel is perceived idle after $t_{\rm obs}$, then the node (successfully) transmits its packet. On the other hand, if

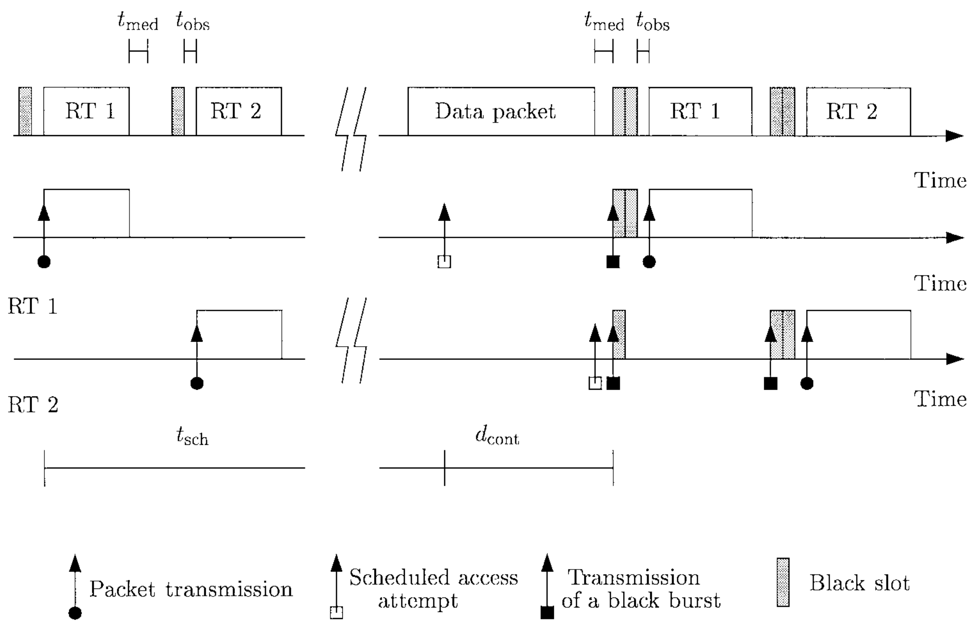


Fig. 2. Time diagram illustrating the BB contention mechanism.

the channel is busy during the observation interval, the node waits again for the channel to be idle for t_{med} and repeats the algorithm.

The start of packet transmissions from different nodes are shifted in time by at least t_{pkt}. Since it is only when a node initiates the transmission of a packet that it schedules its next transmission attempt to a time t_{sch} in the future, the contention delays of different nodes will likewise differ by at least t_{pkt}. Therefore, taking $t_{\mathrm{unit}} \leq t_{\mathrm{pkt}}$, the BB's of different nodes differ by at least one black slot, and thus every BB contention period produces a unique winner. That winner is the node that has been waiting the longest for access to the channel. The observation interval t_{obs} cannot last longer than the black slot time, i.e., $t_{\mathrm{obs}} \leq t_{\mathrm{bslot}}$, so that a node always recognizes when its BB is shorter than that of another contending node. It also has to be shorter than the medium interframe spacing, i.e., $t_{\mathrm{obs}} < t_{\mathrm{med}}$, to prevent real-time nodes from sending BB's by the time that a real-time packet transmission is expected.

The time diagram of Fig. 2 exemplifies the operation of the scheme. In this figure, nodes 1 and 2 have their attempts to access the channel delayed by a data packet transmission. After letting the channel go idle for t_{med}, both nodes contend for access with BB's. Host 1 has been waiting longer and so it transmits a longer BB with which it wins the contention; it successfully transmits its packet t_{obs} afterwards. Host 2 waits until the channel is once again idle for t_{med} and then sends its BB, which is now longer than before, as it reflects the increase in contention delay.

Overall, the BB contention scheme gives priority to real-time traffic, enforces a round-robin discipline among real-time nodes, and results in bounded access delays to real-time packets.

BB contention can also be used to support real-time sessions with different bandwidth requirements, which might be useful for multimedia traffic. On the one hand, distinct real-time sessions may have the corresponding nodes send packets of different sizes when they acquire access rights to the channel. On the other hand, the BB mechanism can be enhanced to accommodate real-time sessions with different scheduling intervals as long as the set of values allowed for the scheduling interval t_{sch} is finite and small. In the latter case, BB contention proceeds in two phases. Real-time nodes first sort their access rights based on contention delays as before. However, it is now possible for two nodes with different scheduling intervals to compute BB's with the same number of black slots. Hence, after this first phase, a real-time node contends again with a new BB, the length of which univocally identifies the scheduling interval being used by the node. This enhancement to BB contention is not pursued further in the paper.

IV. Correctness of BB Contention

In this section, we prove some properties of BB contention for a wireless network without hidden nodes: for every $ij \in L_C$, $N_I(j) \subset N_S(i)$. In this case, carrier sensing deferral rules imply that a transmission over communication link ij started by node i at time t_s^i collides at j with a transmission from node k, $k \in N_I(j)$, if and only if the latter transmission starts some time in the open interval $(t_s^i - \tau_{ik}, t_s^i + \tau_{ik})$. Recall the relations between the time intervals defined: $t_{\mathrm{bslot}} \geq 2\tau$, $t_{\mathrm{long}} \geq t_{\mathrm{med}} + 2\tau$, $t_{\mathrm{obs}} \geq 2\tau$, $t_{\mathrm{obs}} < t_{\mathrm{med}}$, $t_{\mathrm{pkt}} \geq 2\tau$, and $t_{\mathrm{unit}} \leq t_{\mathrm{pkt}}$.

Packets that contend for access to the channel using CSMA/CA, i.e., data packets and those packets that start a real-time session, may collide with one another and with BB's. However, the next two propositions show that all real-time packets that contend with BB's, i.e., all real-time packets except those that start a session, are transmitted without collisions.

Proposition 1: Any real-time packet that contends with BB's does not collide with either data packets or real-time packets that start a session.

Proof: Let real-time node i start a BB transmission comprising b^i black slots over link ij at time t_s^i in an attempt to acquire access rights for its packet. Any data node that can potentially collide with a packet transmitted over link ij must belong to $N_I(j)$. If data node k, $k \in N_I(j)$ does not start a packet transmission during the interval $(t_s^i - \tau_{ik}, t_s^i + \tau_{ik})$, then it cannot interfere later on with a packet from i transmitted $t_{\rm obs}$ after its BB. This is because a data node is required to see the channel idle for $t_{\rm long}$ before sending a packet and $t_{\rm obs} < t_{\rm long}$. Assume instead that data node k starts a packet transmission, of duration $t_{\rm data}$, at time t_s^k, $t_s^k \in (t_s^i - \tau_{ik}, t_s^i + \tau_{ik})$. If

$$t_s^k + t_{\rm data} + \tau_{ik} \geq t_s^i + b^i t_{\rm bslot} + t_{\rm obs}$$

the packet from node k ensures that node i sees the channel busy $t_{\rm obs}$ after its BB, and thus node i does not transmit its packet. On the other hand, if

$$t_s^k + t_{\rm data} + \tau_{ik} < t_s^i + b^i t_{\rm bslot} + t_{\rm obs}$$

the packet from node k may not be sufficient to prevent node i from transmitting its real-time packet. If node i does indeed transmit a packet $t_{\rm obs}$ past its BB, we have

$$\begin{aligned} t_s^i + b^i t_{\rm bslot} + t_{\rm obs} + \tau_{ij} &> t_s^k + t_{\rm data} + \tau_{ik} + \tau_{ij} \\ &> t_s^k + t_{\rm data} + \tau_{kj}. \end{aligned}$$

That is, there is no overlap between the packets sent by nodes i and k: the end of the packet sent by node k reaches j before the beginning of the packet sent by node i.

Clearly, the same proof applies if node k is a real-time node trying to initiate a session. □

Proposition 2: Real-time packets that contend with BB's do not collide with one another or with BB's.

Proof: Let real-time node i start a BB transmission comprising b^i black slots over link ij at time t_s^i, in an attempt to acquire access rights for its packet. As such, node i has seen the channel idle for at least $t_{\rm med}$ prior to t_s^i. Any real-time node with an ongoing session whose transmission can potentially collide with the packet transmitted over ij must belong to $N_I(j)$. If real-time node k, $k \in N_I(j)$ does not start a transmission during the interval $(t_s^i - \tau_{ik}, t_s^i + \tau_{ik})$, then it cannot interfere later on with a packet from i transmitted $t_{\rm obs}$ after its BB. This is because a real-time node is required to see the channel idle for $t_{\rm med}$ before sending a BB, $t_{\rm obs} < t_{\rm med}$, and it cannot send a packet without first transmitting a BB. Assume now that node k starts a transmission at time t_s^k, $t_s^k \in (t_s^i - \tau_{ik}, t_s^i + \tau_{ik})$. This transmission has to be a BB; it cannot be a packet because otherwise node i would have seen the channel idle for a maximum of only $t_{\rm obs}$, rather than $t_{\rm med}$, prior to t_s^i.

Let u_o^i, $u_o^i \leq t_s^i - t_{\rm sch}$, and u_o^k, $u_o^k \leq t_s^k - t_{\rm sch}$ be the last instants of time when nodes i and k transmitted a packet, respectively. Their contention delays are now $d_{\rm cont}^i = t_s^i - (u_o^i + t_{\rm sch})$ and $d_{\rm cont}^k = t_s^k - (u_o^k + t_{\rm sch})$, respectively. By hypothesis, the last packet transmitted by node i over link ij was successful. So, one of the following two conditions was satisfied: either i) $u_o^k + \tau_{kj} + t_{\rm pkt} \leq u_o^i + \tau_{ij}$ or ii) $u_o^i + \tau_{ij} + t_{\rm pkt} \leq u_o^k + \tau_{kj}$. The first condition means that the end of the packet transmitted by node k at time u_o^k reached node j before the beginning of the packet transmitted by node i at time u_o^i. In addition, for node i to have transmitted a packet back at u_o^i, the channel must have been idle at that time, and since $t_{\rm pkt} \geq 2\tau$, condition i) implies that $u_o^k + \tau_{ik} + t_{\rm pkt} \leq u_o^i$. That is, the end of the packet from node k reached node i before the latter started its packet transmission. Therefore

$$\begin{aligned} d_{\rm cont}^k - d_{\rm cont}^i &= t_s^k - t_s^i + u_o^i - u_o^k \\ &\geq t_s^k - t_s^i + t_{\rm pkt} + \tau_{ik} > t_{\rm pkt} \geq t_{\rm unit}. \end{aligned}$$

As a result, the BB of node k with b^k black slots has at least one more black slot than that of node i, i.e., $b^k \geq b^i + 1$. We have

$$\begin{aligned} t_s^i + b^i t_{\rm bslot} + t_{\rm obs} &< t_s^k + \tau_{ik} + b^i t_{\rm bslot} + t_{\rm bslot} \\ &\leq t_s^k + \tau_{ik} + b^k t_{\rm bslot} \end{aligned}$$

so node i will find the channel busy during its observation interval and refrains from transmitting its packet. *Mutatis mutandis*, if condition ii) is satisfied, we have the inequality

$$d_{\rm cont}^i - d_{\rm cont}^k > t_{\rm unit}.$$

The BB of node i will exceed that of node k by at least one black slot, i.e., $b^i \geq b^k + 1$. Thus, node k will find the channel busy during its observation interval and will defer any transmission until it again sees the channel idle for $t_{\rm med}$. If node i transmits its packet, it must do so $t_{\rm obs}$ after the end of the BB. Therefore, node k will see the channel idle for a maximum of $t_{\rm obs}$ and will not interfere with the packet sent by node i.

We have shown that if a real-time node transmits a packet without collisions with transmissions from other real-time nodes, then its next packet will also satisfy this condition. The statement of the proposition follows by induction. □

We can also show that real-time nodes have local priority over data nodes and that the priority attained by a real-time node increases with its contention delay, i.e., with the number of black slots in its BB.

Proposition 3: A real-time node that sees the channel idle for $t_{\rm med}$ after a medium busy condition will access the channel to transmit a BB and will prevent neighboring data nodes from transmitting a packet.

Proof: Let node k end a packet transmission at time t_e^k and real-time node i, $i \in N_S(k)$ see the channel idle for $t_{\rm med}$, starting at $t_e^k + \tau_{ik}$. Data node j, $j \in N_S(k) \cap N_S(i)$, will see the channel become idle no earlier than at $t_e^k + \tau_{jk}$ and at best could only start its packet transmission $t_{\rm long}$ afterwards. But

$$t_e^k + \tau_{ik} + t_{\rm med} + \tau_{ij} < t_e^k + \tau_{kj} + t_{\rm long}$$

which means that a BB sent by real-time node i reaches data node j before the latter sees the channel idle for $t_{\rm long}$, and so data node j defers to this BB. □

The proof of Proposition 2 already contains the following proposition as a corollary.

Proposition 4: A real-time node that sees the channel idle for $t_{\rm med}$ after a medium busy condition will access the channel to transmit a BB and will exclude from contention any neighboring real-time nodes that have a smaller number of black slots in their BB's.

For a wireless LAN, the two previous propositions imply that real-time packets access the channel with priority over data packets and that real-time nodes access the channel to transmit their packets in a round-robin order. Since those packets are not subject to collisions, real-time delays can be bounded, as shown in Section VII.

V. Interaction with Upper Layers

A. Operation with Feedback

If a real-time node were alone in the network, two consecutive real-time packet transmissions belonging to the same session would be separated in time by exactly $t_{\rm acc}$, $t_{\rm acc} \triangleq t_{\rm sch} + t_{\rm bslot} + t_{\rm obs}$. The access delays measure the deviation from this ideal situation. Specifically, an access delay is the time that elapses from the moment an access attempt occurs until the node is able to transmit the corresponding real-time packet, corrected for $(t_{\rm bslot} + t_{\rm obs})$. For $n \geq 2$, the nth access delay associated with a session is denoted by $d^{(n)}$ and is given by $d^{(n)} = (u^{(n)} - u^{(n-1)} - t_{\rm acc})$, where $u^{(n)}$ is the instant of time when the node started the transmission of its nth packet. As we discuss further in Section VII, given the maximum length of data packets, the rate of real-time sessions, and number of real-time nodes, the BB mechanism guarantees that the access delays are bounded and usually by a very small value $d_{\max}$.

When a node is the source node of a session, the contents of its real-time packets can reflect the access delays incurred in contenting for access to the channel. Typically, a real-time application generates blocks of information bits at regular intervals of time, of length much smaller than $t_{\rm acc}$. The block delay is the time interval that elapses from the moment an information block is made available by the application until it is successfully transmitted at the MAC layer (corrected for $t_{\rm bslot}+t_{\rm obs}$ and neglecting processing delays). The relation between access and block delays depends on how the application blocks of information are packetized for transmission at the MAC layer. One possibility is to have the MAC layer convey in a packet all the information blocks generated up to the instant when the node is about to start a packet transmission. The length of a real-time packet would thus grow with the access delay incurred by the node. The block delay of the oldest block conveyed in the packet would consist of $t_{\rm acc}$, plus the corresponding access delay: the block delay would never exceed $(t_{\rm acc} + d_{\max})$. In general, however, it is not feasible to assemble a packet at the time that its transmission should start, and further, the MAC layer usually contains a single buffer that we must ensure is filled with a packet by the time access to the channel is granted.

For a realistic alternative within the spirit of this section, consider a simplified communication architecture in which a real-time application puts its generated blocks of information into an application buffer. Whenever the node successfully transmits a packet it signals the application, which will assemble the next packet with all the blocks of information currently queued at the application buffer, plus the blocks that will be generated during the next interaccess interval of length $t_{\rm acc}$. At this later time, the packet is delivered to the MAC layer for transmission. With this procedure, the MAC layer always has a packet ready for transmission by the time it acquires undisputed access to the channel. When a node transmits its nth packet at time $u^{(n)}$, it leaves in the application buffer the blocks of information generated during the previous $d^{(n)}$ units of time; they will be part of the contents of the $(n+1)$th packet. The latter packet further incurs an access delay of $d^{(n+1)}$ at the MAC layer. Therefore, the block delay of the oldest block conveyed in the $(n+1)$th packet is not greater than $(d_n + t_{\rm acc} + d_{n+1})$: the block delay during a session never exceeds $(t_{\rm acc} + 2d_{\max})$.

B. Operation Without Feedback

In the previous section, the contents of a real-time packet depended on the access delays incurred by a node. There is a direct coupling between the MAC layer and the real-time application. A simpler communication architecture may be desired in which already assembled packets are passed onto the MAC layer for transmission one by one. This is also the situation encountered when a node is simply relaying real-time packets arriving from a distant source.

Suppose that real-time packets are presented to the MAC layer periodically, one every $t_{\rm rdy}$ units of time. The packet delay is the time that elapses from the moment a packet is available for transmission until it is successfully transmitted at the MAC layer (corrected for $t_{\rm bslot}+t_{\rm obs}$). The packet delay of the nth packet $w^{(n)}$ is given by $w^{(n)} = (u^{(n)} - t^{(n)} - t_{\rm bslot} - t_{\rm obs})$, where $t^{(n)}$ is the instant of time when the nth packet becomes ready for transmission, $t^{(n)} = t^{(1)} + (n-1)t_{\rm rdy}$. Clearly, we should not choose $t_{\rm sch} + t_{\rm bslot} + t_{\rm obs} = t_{\rm rdy}$. If that choice was made, the instants when the node accesses the channel would start drifting in relation to the arrival times of new packets, and the node cannot keep up with the packet arrival rate. Indeed, the packet delay of the nth packet would be $w^{(n)} = w^{(1)} + \sum_{i=2}^{n} d^{(i)}$, which grows monotonically with the number of packets already transmitted.

Consider instead a preventive approach whereby a real-time node schedules its next transmission attempt short of the inter-arrival time for packets $t_{\rm rdy}$. Specifically, when a real-time node transmits a packet it schedules the next transmission attempt to time $t_{\rm sch}$ in the future, now with $t_{\rm sch} = t_{\rm rdy} - t_{\rm bslot} - t_{\rm obs} - \delta$, where δ, $\delta > 0$, is called the slack time. At a scheduled access attempt, a real-time node will only start contending for access to the channel if a real-time packet is available for transmission. Otherwise, it waits for a ready packet and only then starts to contend for access to the channel. The correctness of the BB contention mechanism is preserved as long as the contention delays used to compute the lengths of BB's are always counted from the scheduled access attempts up to the time when the channel becomes idle for $t_{\rm med}$. Fig. 3 shows an example with two real-time nodes. Real-

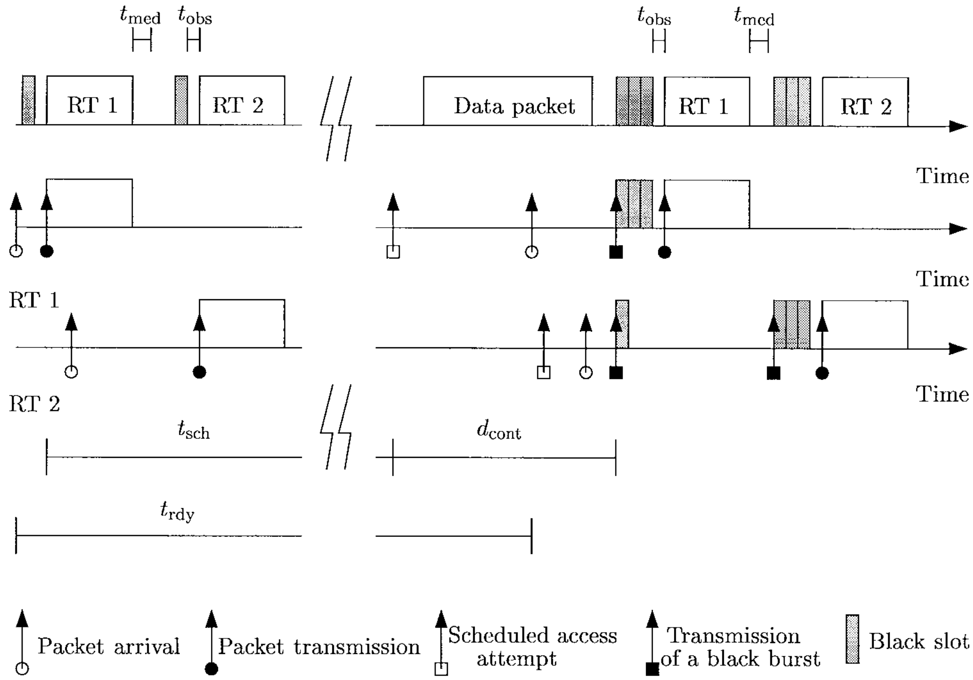


Fig. 3. Example of operation without feedback from the MAC layer to the upper layers.

time node 1 does not start to contend for access to the channel at its scheduled access attempt because no packet is ready at that time. When a packet arrives at this node, the channel is busy with a data packet transmission. As soon as the channel becomes idle for $t_{\rm med}$, both real-time nodes 1 and 2 contend with BB's. The contention delays used in the computation of BB's are measured from the scheduled access attempts rather than from the arrival times of packets. Although the packet arrivals belonging to nodes 1 and 2 are not separated in time by at least $t_{\rm pkt}$, note that their scheduled access attempts satisfy this condition. As suggested in the figure, operation of a BB mechanism in this way tends to produce longer BB's since the nodes may have to delay themselves on purpose until a packet is ready.

The slack time δ has to be determined so that packet delays remain bounded at all times. Design criteria are given in Section VII. Clearly, δ has to be longer than the channel time occupied by a maximum length data packet transmission. Suppose that every time a real-time node is ready to contend, a maximum length data packet gets hold of the medium. At best, the real-time node will be able to send packets one every $(t_{\rm sch} + t_{\rm data} + t_{\rm med} + t_{\rm bslot} + t_{\rm obs})$ units of time, where $t_{\rm data}$ is the duration of a data packet transmission. In order to obtain a bounded packet delay, we need to ensure that $(t_{\rm sch} + t_{\rm bslot} + t_{\rm obs} + t_{\rm data} + t_{\rm med}) < t_{\rm rdy}$.

VI. Chaining

In a wireless LAN, the number of real-time nodes contending for access to the channel can be reduced by grouping real-time packet transmissions into chains. A chain is a sequence of real-time packets where each packet invites the next for transmission. In order to support chains, each real-time packet is endowed with two new fields: a send node ID (SID), which contains the identity of the node transmitting the packet, and a next node ID (NID), which contains the identity of the node invited to transmit next. The special value NIL is used to denote an empty field. The SID field is set to NIL in the first and last packets of a session. A real-time node relies on the round-robin discipline enforced by BB contention to choose a temporary ID to be used during a session. After sending the first packet of a session, a real-time node observes the channel during the ensuing round to determine the identity of all the other active sessions. Therefore, by the time it transmits its second packet, it is able to choose a unique identifier for itself which it keeps for the duration of the session. The last packet of a session sent by a real-time node also has the SID field set to NIL. The NID field is NIL at every packet that is at the tail of a chain.

A node has to respond within an interval of length $t_{\rm short}$ to an invitation from another real-time node in order to ensure that the real-time packets comprising a chain are transmitted in sequence without being disturbed by either BB's or data packet transmissions. The dynamics of chain creation and segregation are achieved through a distributed algorithm running at each node. Two basic operations can be performed on chains: splitting and concatenation. Principally, splitting occurs when a node ends a session and leaves the chain to which it belongs, possibly dividing it into two new chains. It may also occur when a packet is corrupted, e.g., due to a link outage. Since real-time nodes are always prepared to contend with BB's at every scheduled access attempt, even when they are part of a chain, an abrupt break in a chain does not deprive them of their access rights to the channel: it only reduces the efficiency with which the channel is used. Fig. 4 shows a split of a chain into two new subchains.

Concatenation occurs when two distinct chains are merged into a longer one for the purposes of efficiency. The algorithm

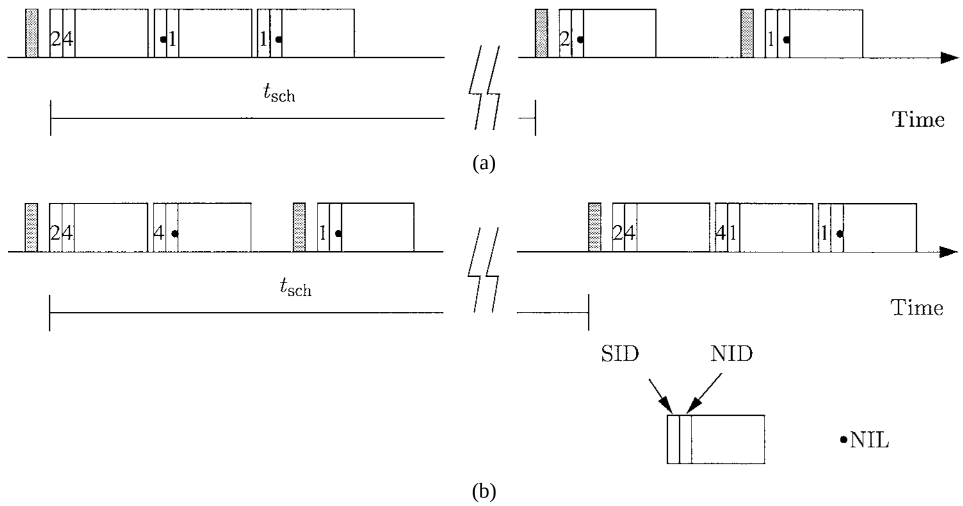


Fig. 4. Operations on chains: (a) splitting and (b) concatenation.

for the concatenation of chains has to ensure that closed loops of packet transmissions never occur and that the number of packets in a chain does not exceed a prespecified maximum, so as to also provide timely access to data traffic. It is up to the tail node of a chain to decide whether or not to pull toward itself the next chain that comes onto the channel. To this end, the tail node monitors the channel during a round. It first identifies the candidate node to be invited in the next round by looking for the first packet with an SID field not NIL. Then, it counts the number of packets comprising the chain to which the candidate node belongs, i.e., it counts the number of real-time packets observed on the channel until it finds either an SID or a NID field set to NIL. Finally, the tail node keeps a running counter with the number of nodes present in the chain being currently observed on the channel. At the end of the round, when the tail node finally transmits another real-time packet, it tests whether the current count of nodes (which reflects the number of nodes in its own chain) plus the number of nodes in the candidate chain are below the maximum allowed. If this condition is satisfied, the tail node invites the candidate node immediately after sending its real-time packet. Otherwise, concatenation does not take place.

Fig. 4 exemplifies the concatenation of two chains into a longer one. Tail node 4 identifies node 1 as a candidate for invitation in the next round, and it finds out that node 1 is in a chain all by itself $(\mathrm{NID} = \mathrm{NIL})$. At the end of the round, when tail node 4 is invited by node 2, it knows from the running counter that the chain to which it belongs comprises two nodes (2 and 4). In this example, the maximum number of nodes allowed in a chain is greater than two, so node 4 invites node 1 to transmit its packet.

VII. Analysis of BB Contention

A. Operation with Feedback

When there is no data traffic, real-time nodes appear to be accessing a TDM transmission structure without incurring access delays. Two consecutive packet transmissions from any given real-time node are separated in time exactly by an interaccess interval of length $t_{\rm acc}$. A data packet transmission may perturb this state of affairs, delaying the access instants of real-time nodes and forcing the nodes to contend among themselves with BB's. Eventually, the real-time nodes recover from the perturbation and reorganize themselves into a state in which they no longer incur access delays. This is the behavior that we formalize mathematically in this section.

For simplicity of presentation, the path delays between nodes are ignored. There are n chains in the system, labeled from 1 through n. The total channel time occupied by chain i when it is not delayed is denoted by l_i, and the idle time per round is y

$$\sum_{i=1}^{n} l_i + y = t_{\rm acc}, \quad y > 0. \tag{1}$$

The length of a chain grows in proportion to the access delay experienced by its head node. The growth rate associated with chain i is denoted by η_i, and it is a function of the number of nodes comprising the chain as well as their individual bandwidth requirements. For example, if the channel bit rate is r_c, each real-time session has coding rate r_s, the overhead per real-time packet is h, and there are m nodes per chain; then $l_i = t_{\rm bslot} + t_{\rm obs} + m(h + r_s t_{\rm acc})/r_c + (m-1)t_{\rm short} + t_{\rm med}$ and $\eta_i = mr_s/r_c$, for all i, $i = 1, \ldots, n$. Every BB contains at least one black slot; beyond that, the length of a BB increases linearly with the contention delay experienced by the real-time node. The proportionality constant is denoted by α, $\alpha = t_{\rm bslot}/t_{\rm unit}$. Let $\gamma_i \triangleq \alpha + \eta_i + \alpha\eta_i$, $\gamma \triangleq (\gamma_1, \ldots, \gamma_n)^t$ and $\Gamma \triangleq \mathrm{diag}(\gamma_1, \ldots, \gamma_n)$.

Let u_{ik}, $i = 1, \ldots, n$, $k = -1, 0, \ldots,$ denote the time at which the head node of chain i accesses the channel to transmit a packet in round k; at that time, it also schedules the next transmission attempt to time $(u_{ik} + t_{\rm sch})$. The access delay of the head node of chain i, $i = 1, \ldots, n$ in round k, $k = 0, 1, \ldots,$ is denoted by d_{ik} and is given by

$$d_{ik} = u_{ik} - u_{i,k-1} - t_{\rm acc}. \tag{2}$$

The vector of access delays in round k is denoted by d_k, $d_k = (d_{1k}, \ldots, d_{nk})^t$.

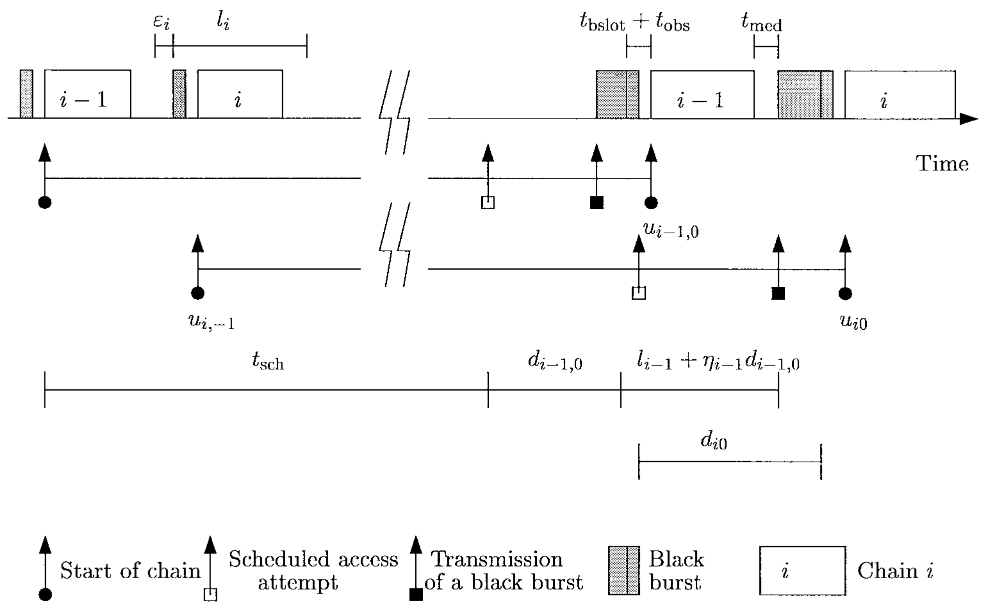


Fig. 5. Initial access delays.

Definition 1: The BB contention mechanism is stable if and only if there is a value $d_{\max}$ such that $d_{ik} \leq d_{\max}$, for $i = 1, \ldots, n$ and $k = 0, 1, \ldots$.

Definition 2: The BB contention mechanism is unconditionally stable if and only if it is stable, whatever the size of the data packets.

Consider an initial setting whereby

$$u_{1,-1} = -t_{\text{sch}} \quad \text{and} \tag{3}$$

$$u_{i,-1} - u_{i-1,-1} = l_{i-1} + \varepsilon_i, \quad i = 2, \ldots, n \tag{4}$$

with $\sum_{i=2}^{n} \varepsilon_i \leq y$. Define the vector $\varepsilon^t \triangleq (0, \varepsilon_2, \ldots, \varepsilon_n)$. The head node of chain 1 has a transmission attempt scheduled for time $t = 0$, and it would transmit a BB if the channel were idle at that time. Assume instead that there is a data packet transmission that extends from $t = 0$ to $t = t_{\text{data}}$, and define $z \triangleq (t_{\text{data}} + t_{\text{med}})$. Then, we have the following.

Proposition 5: The vector of access delays in round 0 is given by

$$d_0 = (1+\alpha)[A(-\varepsilon + z e_1)]_+$$

with

$$A = [I - F(I+\Gamma)]^{-1} = \sum_{i=0}^{n-1} [F(I+\Gamma)]^i$$

$$F \triangleq \begin{bmatrix} 0 & & & & \\ 1 & \ddots & & & \\ 0 & \ddots & \ddots & & \\ & \ddots & \ddots & 0 & \\ & & \ddots & 1 & 0 \end{bmatrix}$$

$$e_1 \triangleq (1, 0, \ldots, 0)^t.$$

Proof: Clearly

$$d_{1,0} = (1+\alpha)z. \tag{5}$$

Consider the condition

$$u_{i-1,0} + \eta_{i-1} d_{i-1,0} + l_{i-1} - t_{\text{bslot}} - t_{\text{obs}} > u_{i,-1} + t_{\text{sch}}, \quad i = 2, \ldots, n \tag{6}$$

which from (2) and (4) is equivalent to

$$(1 + \eta_{i-1}) d_{i-1,0} - \varepsilon_i > 0, \quad i = 2, \ldots, n. \tag{7}$$

If this condition is satisfied, then the channel is busy when the scheduled access instant of the head node of chain i occurs in round 0. Then (see Fig. 5)

$$\begin{aligned} d_{i0} &= (1+\alpha)(u_{i-1,0} + \eta_{i-1} d_{i-1,0} + l_{i-1} - u_{i,-1} - t_{\text{acc}}) \\ &= (1 + \gamma_{i-1}) d_{i-1,0} - (1+\alpha)\varepsilon_i, \quad i = 2, \ldots, n. \end{aligned} \tag{8}$$

Otherwise, if (7) is not satisfied, $d_{i0} = 0$. Combining the two cases yields

$$d_{i0} = [(1+\gamma_{i-1}) d_{i-1,0} - (1+\alpha)\varepsilon_i]_+, \quad i = 2, \ldots, n \tag{9}$$

where $[x]_+ \triangleq \max(x, 0)$. In matrix form, we write

$$d_0 = [F(I+\Gamma) d_0 - (1+\alpha)\varepsilon + (1+\alpha) z e_1]_+ \tag{10}$$

from which the proposition follows. Note that F is a "downward/forward shift" matrix: left multiplication by F shifts the lines of a matrix downwards, and right multiplication by F shifts the columns of a matrix to the left. □

The entries of matrix A a_{ij} are given by

$$a_{ij} = \begin{cases} 1 & i = j \\ \prod_{l=j}^{i-1} \bar{\gamma}_l & i > j \\ 0 & i < j \end{cases} \tag{11}$$

with $\bar{\gamma}_l \triangleq 1 + \gamma_l$.

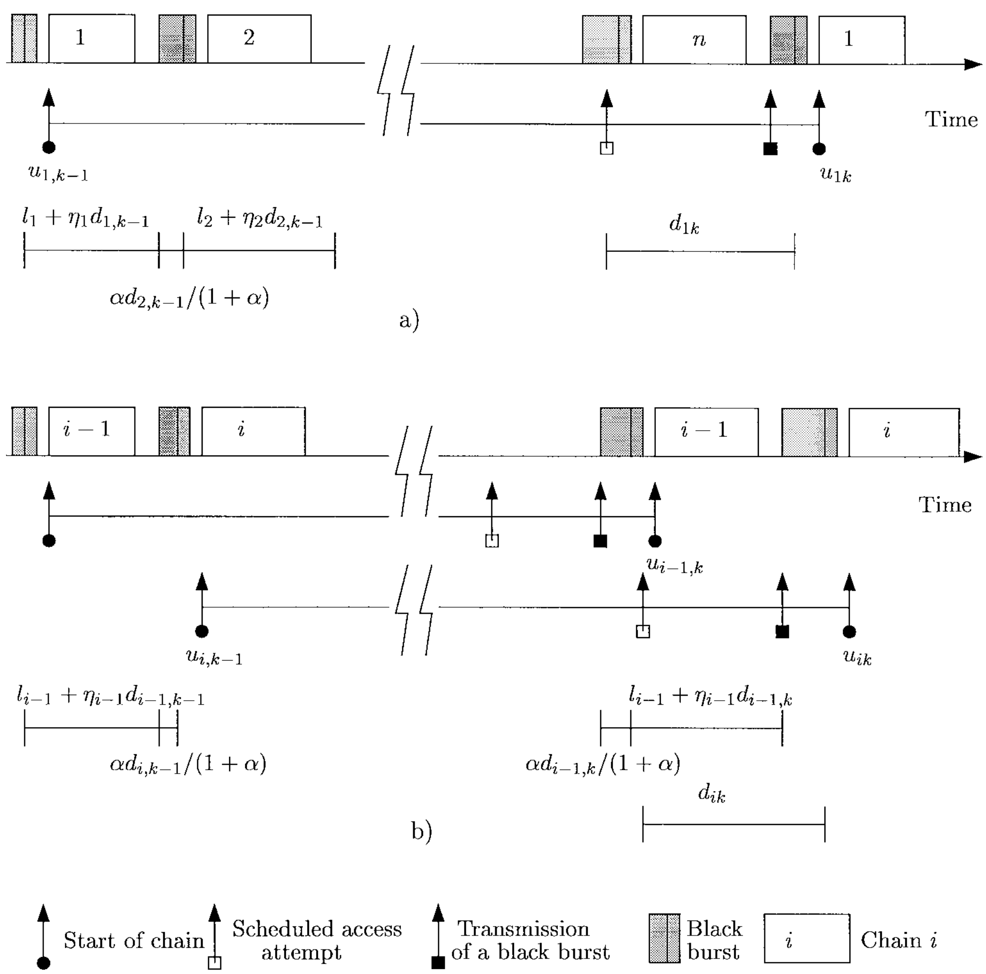


Fig. 6. Recursive relation for the access delays. (a) Access delay of the head node of chain 1. (b) Access delay of the head node of chain i, $i \neq 1$.

Proposition 6: The vectors of access delays in consecutive rounds are related by

$$d_k = [Cd_{k-1} - (1+\alpha)yAe_1]_+$$

where

$$C = Ae_1\gamma^t - (1+\alpha)A(I-F) + I.$$

Proof: Using the same methodology as in Proposition 5, we get (see Fig. 6)

$$d_{1k} = \left[\sum_{j=1}^{n} \gamma_j d_{j,k-1} - \alpha d_{1,k-1} - (1+\alpha)y\right]_+ \tag{12}$$

$$d_{ik} = [(1+\gamma_{i-1})d_{i-1,k} - (\gamma_{i-1} - \alpha)d_{i-1,k-1} - \alpha d_{i,k-1}]_+, \quad i = 2,\ldots,n. \tag{13}$$

In matrix form, we write

$$\begin{aligned} d_k = \{&F(I+\Gamma)d_k + [I - F(I+\Gamma) - (1+\alpha)(I-F)]d_{k-1} \\ &+ \gamma^t d_{k-1}e_1 - (1+\alpha)ye_1\}_+ \end{aligned} \tag{14}$$

from which the proposition follows. □

The entries of matrix C are given by

$$c_{ij} = \begin{cases} \gamma_i \prod_{l=1}^{i-1} \bar{\gamma}_l - \alpha, & i = j \\ \gamma_j \prod_{l=1}^{i-1} \bar{\gamma}_l - \gamma_j \bar{\alpha} \prod_{l=j+1}^{i-1} \bar{\gamma}_l, & i > j \\ \gamma_j \prod_{l=1}^{i-1} \bar{\gamma}_l, & i < j \end{cases} \tag{15}$$

with $\bar{\alpha} \triangleq 1 + \alpha$. Clearly, $c_{ij} \geq 0$, matrix C is nonnegative, and we bound the spectral radius of C, denoted by $\rho(C)$.

Proposition 7: The spectral radius of C is bounded as follows.

1) If $\sum_{i=1}^{n} \gamma_i - \alpha < 1$, then
$$1 - \left(1 - \sum_{i=1}^{n} \gamma_i + \alpha\right) \prod_{i=1}^{n-1} \bar{\gamma}_i \leq \rho(C) \leq \sum_{i=1}^{n} \gamma_i - \alpha.$$

2) If $\sum_{i=1}^{n} \gamma_i - \alpha = 1$, then
$$\rho(C) = 1.$$

3) If $\sum_{i=1}^{n} \gamma_i - \alpha > 1$, then
$$\sum_{i=1}^{n} \gamma_i - \alpha \leq \rho(C) \leq 1 + \left(\sum_{i=1}^{n} \gamma_i - \alpha - 1\right) \prod_{i=1}^{n-1} \bar{\gamma}_i.$$

Proof: Compute the row sums

$$Ce = \left(\sum_{i=1}^{n} \gamma_i\right) Ae_1 - (1+\alpha)Ae_1 + e = \left(\sum_{i=1}^{n} \gamma_i - \alpha - 1\right) Ae_1 + e \tag{16}$$

with $e \triangleq (1,\ldots,1)^t = (I-F)^{-1}e_1$ and apply the Row Sums Theorem [12, p. 492]. □

This proposition shows, in particular, that $\rho(C) < 1$, $\rho(C) = 1$, and $\rho(C) > 1$, for $\sum_{i=1}^{n} \gamma_i - \alpha < 1$, $\sum_{i=1}^{n} \gamma_i - \alpha = 1$, and $\sum_{i=1}^{n} \gamma_i - \alpha > 1$, respectively. If $\gamma_1 > \alpha$, then $C > 0$. On the other hand, if $\gamma_1 = \alpha$, the first column of C is zero. The eigenvalues of C in that case are zero, together with the eigenvalues of the principal submatrix obtained by deleting the first row and first column of C. The latter submatrix is positive. In general, we use Perron's Theorem [12, p. 500] to state the next proposition.

Proposition 8: We have that:

1) $\rho(C)$ is an algebraically simple eigenvalue of C;
2) there is v, $v > 0$, such that $Cv = \rho(C)v$;
3) $|\lambda| < \rho(C)$ for every eigenvalue λ, $\lambda \neq \rho(C)$;
4) $\lim_{k\to\infty}[\rho(C)^{-1}C]^k = L$, $L = (w^t v)^{-1} v w^t$, $Cv = \rho(C)v$, $w^t C = w^t \rho(C)$.

Consider the following equation in $v(x)$:

$$Ae_1 - (1+\alpha)A(I-F)v(x) = (x-1)v(x). \tag{17}$$

This equation will be of paramount importance in deriving our next results.

Proposition 9: The equation $Ae_1 - (1+\alpha)A(I-F)v(x) = (x-1)v(x)$ has solution

$$v_i(x) = \frac{\prod_{l=1}^{i-1}[x + \alpha + \gamma_l(x-1)]}{(x+\alpha)^i}$$

where $v_i(x)$ denotes the ith component of $v(x)$.

Proof: Multiplying both sides of the equation by $A^{-1} = [I - F(I+\Gamma)]$ and rearranging the terms yields

$$\{(x+\alpha)I - F[(x+\alpha)I + (x-1)\Gamma]\}v(x) = e_1 \tag{18}$$

from which the proposition follows. □

Proposition 10: The characteristic polynomial of matrix C, called $p(x)$, is given by

$$p(x) = \frac{x(x+\alpha)^n - \prod_{l=1}^{n}[x + \alpha + \gamma_l(x-1)]}{x-1}.$$

Proof: By definition, v is an eigenvector of C, corresponding to eigenvalue λ, if and only if $Cv = \lambda v$. This condition can be expressed as $(\gamma^t v)Ae_1 - (1+\alpha)A(I-F)v = (\lambda - 1)v$. Using Proposition 9, we get $v = (\gamma^t v)v(\lambda)$. If v is indeed an eigenvector $\gamma^t v \neq 0$ and hence $\gamma^t v(\lambda) = 1$. Conversely, if $\gamma^t v(\lambda) = 1$, then λ is an eigenvalue with eigenvector $v(\lambda)$. In short, λ is an eigenvalue of C if and only if $\gamma^t v(\lambda) = 1$. This condition can be written as $p(\lambda) = 0$ with

$$p(x) = (x+\alpha)^n - \sum_{i=1}^{n} \gamma_i (x+\alpha)^{n-i} \prod_{l=1}^{i-1}[x + \alpha + \gamma_l(x-1)] \tag{19}$$

from which we conclude that $p(x)$ is the characteristic polynomial. The form presented in the statement of the proposition is obtained through algebraic manipulations. □

One interesting consequence of Proposition 10 is the fact that the spectrum of C remains invariant to a permutation of the γ_i. For the next proposition, let $\lambda = \rho(C)$ and v and w^t be the associated right and left eigenvectors, respectively, such that $\gamma^t v = w^t Ae_1 = 1$.

Proposition 11: The vector of access delays in round k can be written as

$$d_k = \left[\frac{\lambda^k}{w^t v}\left(w^t d_0 - \frac{(1+\alpha)y}{\lambda - 1}\right)v + M^k d_0 - \frac{(1+\alpha)y}{1-\gamma^t v(1)}(I - M^k)v(1)\right]_+$$

for $\lambda \neq 1$, $k \geq 1$, and

$$d_k = \left[\frac{w^t d_0 - k(1+\alpha)y}{w^t v}v + M^k d_0 + \frac{(1+\alpha)y}{w^t v}(I - M^k)\left(\frac{w^t v'(1)}{w^t v}v - v'(1)\right)\right]_+$$

for $\lambda = 1$, $k \geq 1$.

Proof: From Propositions 6 and 9, we write $(xI - C)v(x) = (1 - \gamma^t v(x))Ae_1$, and $(x - \lambda)w^t v(x) = 1 - \gamma^t v(x)$, from which the following equalities are deduced:

$$(I-C)v(1) = (1 - \gamma^t v(1))Ae_1 \tag{20}$$

$$(I-C)v'(1) = -\gamma^t v'(1)Ae_1 - v(1) \tag{21}$$

$$w^t v(1) = \frac{1 - \gamma^t v(1)}{1 - \lambda} \tag{22}$$

$$w^t v = -\gamma^t v'(\lambda) \tag{23}$$

$$w^t v'(\lambda) = -\frac{\gamma^t v''(\lambda)}{2}. \tag{24}$$

These equations will prove useful in deriving our results. It is easily shown by induction that

$$d_k = \left[C^k d_0 - y\left(\sum_{l=0}^{k-1} C^l\right)Ae_1\right]_+, \quad k = 0, 1, \ldots. \tag{25}$$

Decompose C in the form $C = \lambda L + M$ with $L = (w^t v)^{-1} v w^t$. If $\lambda \neq 1$, then using (20) and (22), we get

$$\begin{aligned}\left(\sum_{l=0}^{k-1} C^l\right)Ae_1 &= (I - C^k)(I-C)^{-1}Ae_1 \\ &= -\lambda^k L\frac{v(1)}{1-\gamma^t v(1)} + \frac{1}{1-\gamma^t v(1)} \cdot (I - M^k)v(1) \\ &= \frac{\lambda^k}{w^t v}\frac{1}{\lambda - 1}v + \frac{1}{1-\gamma^t v(1)}(I - M^k)v(1)\end{aligned} \tag{26}$$

which when inserted into (25) yields the statement of the proposition. If $\lambda = 1$, then we make use of (21) and (23) to write

$$\begin{aligned}\left(\sum_{l=0}^{k-1} C^l\right) Ae_1 &= (k-1)LAe_1 + (I-M^k)(I-M)^{-1}Ae_1 \\ &= \frac{kv}{w^tv} + \frac{1}{w^tv}(I-M^k)\left(v'(1) - \frac{w^tv'(1)}{w^tv}v\right). \end{aligned} \quad (27)$$

Inserting this result into (25) yields the statement of the proposition for $\lambda = 1$. The product $w^tv'(1)$ can be computed via (24). □

Proposition 12: The BB contention mechanism is unconditionally stable if and only if

$$\sum_{i=1}^{n} \gamma_i - \alpha \leq 1.$$

In addition, if the condition above is not satisfied, the BB contention mechanism is still stable provided that

$$z \leq \frac{y}{\lambda - 1}$$

where λ is the largest root of the characteristic polynomial.

Proof: If $\sum_{i=1}^{n} \gamma_i - \alpha \leq 1$, then $\lambda \triangleq \rho(C) \leq 1$. Referring to Proposition 11, we conclude that $\lim_{k\to\infty} d_k = 0$, independent of the value of d_0. Consider now the case $\sum_{i=1}^{n} \gamma_i - \alpha > 1$ and note that, from Proposition 5, $d_0 \leq (1+\alpha)zAe_1$, with the equality holding for $\varepsilon = 0$. Hence, $w^td_0 \leq (1+\alpha)z$, and using Propositions 8 and 11, we conclude that $\lim_{k\to\infty} d_k = 0$, if $z < y/(\lambda-1)$, and $\lim_{k\to\infty} d_k = \infty$, if $z > y/(\lambda-1)$. For the special case $z = y/(\lambda-1)$, it can be shown that all the eigenvalues of M have modulus less than unity. Therefore, referring again to Proposition 11, we have $\lim_{k\to\infty} d_k = (1+\alpha)ye/(\sum_{i=1}^{n} \gamma_i - \alpha - 1)$. □

Although the access delays are bounded at the limiting case $\sum_{i=1}^{n} \gamma_i - \alpha > 1$, $z = y/(\lambda-1)$, data traffic would never be allowed onto the channel.

B. Operation Without Feedback

The model for operation without feedback differs from the one of the previous section in a number of aspects. First, the total channel time occupied by chain i is always l_i, independent of access delays; that is, $\eta_i = 0$, $\gamma_i = \alpha$, for every i, $i = 1, \ldots, n$. Second, the head node of a chain is only ready to contend for access to the channel when both a scheduled access attempt has occurred, and a packet is waiting. Third, the concept of stability refers to bounded packet delays, rather than to bounded access delays. We have derived the following sufficient conditions for the packet delays to remain bounded.

Proposition 13: The BB mechanism is stable if the following conditions are met.

1) There is an y, $y > 0$, such that $\sum_{i=1}^{n} l_i + y = t_{\rm sch} + t_{\rm bslot} + t_{\rm obs}$.
2) Either $\alpha(n-1) \leq 1$ or, if $\alpha(n-1) > 1$, $z + \delta < y/(\lambda - 1)$.
3) $\delta \geq \max(d_{ik})$, for $i = 1, \ldots n$ and $k = 0, 1, \ldots$.

In these conditions, λ denotes the largest eigenvalue of C, and $\max(d_{ik})$ is the largest value attained by the linear system of Propositions 5 and 6.

We omit the proof because of space limitations, but we will provide some insight into the conditions of the proposition. The first condition states that the chain transmissions do not occupy a full scheduling interval plus $(t_{\rm bslot} + t_{\rm obs})$. That is, there is some idle time left even if all nodes at the heads of a chain transmit at their scheduled access attempts. The second condition is sufficient to ensure that the real-time nodes recover from the most unfavorable initial setting. That happens when a packet arrival occurs a slack time after the scheduled access attempt but a data packet transmission has just started at that time. Thus, the initial contention delay is $(\delta + z)$, rather than just z. Last, if the system reaches an overload point where the head nodes always have a packet ready at the scheduled access attempts, we would like the packet delays to decrease from one access instant to another, so that the system effectively drifts away from the overload point. This is expressed by the third condition.

VIII. Results

A. Parameters and Assumptions

We present results for a wireless LAN with a mixed population of data and real-time nodes: all nodes are assumed to sense each other's transmissions. In this case, node mobility does not impact our channel access scheme and therefore was not explicitly modeled. Nominal values for the parameters of the system are given in Table I, and when applicable, they have been taken from the direct sequence spread spectrum version of the IEEE 802.11 standard. The channel bit rate is denoted by r_c, and each packet transmission has a physical (PHY) layer header $t_{\rm phy}$ and a MAC-layer header h. Packets arrive at a data node according to a Poisson process with rate λ, and each has payload denoted by $b_{\rm pkt}$. Real-time sessions have a constant coding rate r_s. We take $t_{\rm unit} = t_{\rm pkt}$, $t_{\rm pkt} = t_{\rm phy} + (h + r_st_{\rm acc})/r_c$ (operation with feedback), $t_{\rm pkt} = t_{\rm phy} + (h + r_st_{\rm rdy})/r_c$ (operation without feedback). The data and real-time loads are given by $\rho_{\rm dh} = N_{\rm dh}\lambda b_{\rm pkt}/r_c$ and $\rho_{\rm rth} = N_{\rm rth}r_s/r_c$, where $N_{\rm dh}$ and $N_{\rm rth}$ denote the number of data and real-time nodes, respectively.

The overall performance of the wireless LAN when it operates with both data and real-time traffic was studied with simulations. Each simulation point was obtained by running ten independent replicas of the system and averaging the results. Each simulation run corresponds to 4–10 min of system operation. The error bars in the plots mark the 95% confidence intervals.

B. Stability

The operation of BB contention is stable whenever the maximum real-time delay, either access delay (feedback) or packet delay (no feedback), is finite. On the other hand, CSMA/CA does make guarantees on maximum delays. A traffic stream, either real-time or data, has a stable operation under CSMA/CA if its average packet delay is finite.

TABLE I
NOMINAL VALUES FOR THE PARAMETERS OF THE SYSTEM

Parameter	Symbol	Nominal value
Channel rate	r_c	2 Mbits/s
Short spacing	t_{short}	10 μs
Medium spacing	t_{med}	30 μs
Long spacing	t_{long}	50 μs
Slot time	t_{slot}	20 μs
PHY header per pkt	t_{phy}	192 μs
MAC header per pkt	h	34 bytes
Black slot time	t_{bslot}	20 μs
Observation interval	t_{obs}	20 μs
Data nodes	N_{dh}	10 nodes
Inter-access interval	t_{acc}	30 ms
Inter-packet arrival	t_{rdy}	30 ms

TABLE II
NUMBER OF SUPPORTED REAL-TIME NODES FOR $r_s = 64$ KBIT/S, WITH AND WITHOUT FEEDBACK FROM THE MAC LAYER TO THE APPLICATION. AN IDEAL TDM STRUCTURE ACCOMMODATES 23 NODES

b_{pkt}	Feedback	No feedback
825 bytes	21	18 ($\delta = 5$ ms)
1500 bytes	21	16 ($\delta = 8.2$ ms)
∞	21	

TABLE III
NUMBER OF SUPPORTED REAL-TIME NODES FOR $r_s = 32$ KBIT/S AND VARIOUS NUMBERS OF REAL-TIME NODES PER CHAIN. AN IDEAL TDM STRUCTURE ACCOMMODATES 37 NODES

	Real-time nodes per chain		
b_{pkt}	1	2	4
825 bytes	31	35	36
1500 bytes	30	35	36
∞	24	35	36

Since BB contention gives nonpreemptive priority access to real-time traffic, the maximum number of real-time nodes supported in the network does not depend on the data load but only on the largest length of a data packet, as shown in Section VII. Table II gives the maximum number of supported real-time nodes for a coding rate $r_s = 64$ Kbit/s, an interaccess interval $t_{\text{acc}} = 30$ ms (feedback) and an interpacket arrival interval $t_{\text{rdy}} = 30$ ms (no feedback). The row entry labeled ∞ gives the number of real-time nodes supported in an unconditionally stable system. For the chosen parameters, an ideal TDM strategy would accommodate 23 real-time nodes. The table indicates that even without chaining, BB operation coupled with feedback to the application allows for almost the same number of nodes to be supported as an ideal TDM scheme. Without feedback to the application, the slack time δ has to be designed. In the table, the values chosen for δ are indicated together with the maximum number of real-time nodes that satisfy the sufficient stability conditions of Section VII. Simulations have shown that more nodes can be accommodated than predicted from the analysis.

The effect of chaining on the number of supported real-time nodes is reported in Table III. Real-time sessions have a coding rate $r_s = 32$ Kbit/s, and there is feedback from the MAC layer to the application. Although real-time traffic is unconditionally stable for a maximum of 24 nodes when there is no chaining, just grouping the real-time packets in pairs increases the number of supported real-time nodes to 35, independent of the length of data packet transmissions.

Figs. 7 and 8 show the maximum number of real-time nodes that can be supported in the wireless LAN with stable data and real-time traffic operation. It is clear that BB contention improves the total amount of traffic that can be carried in

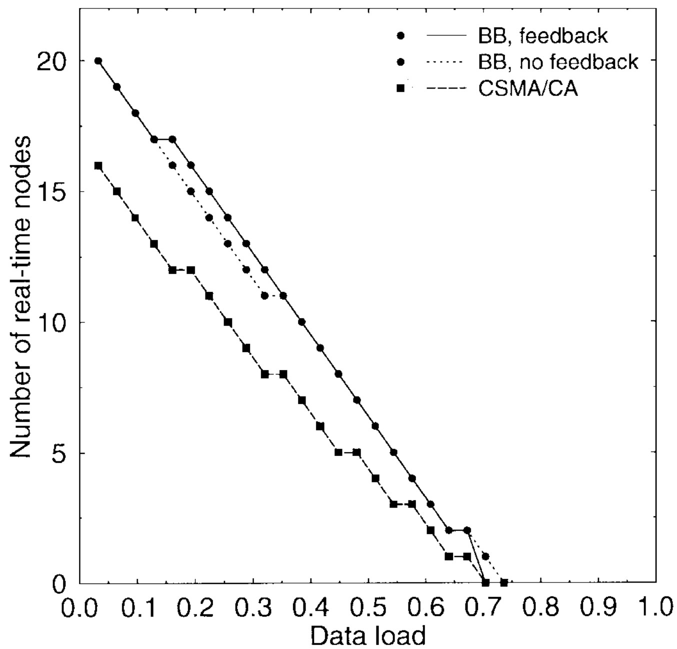


Fig. 7. Maximum number of real-time nodes versus data load for stable operation $r_s = 64$ Kbit/s, $b_{\text{pkt}} = 825$ bytes.

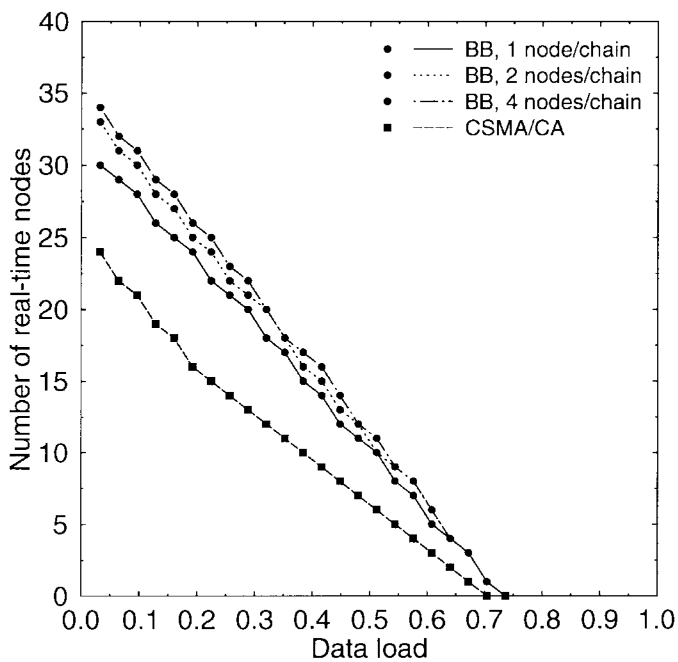


Fig. 8. Maximum number of real-time nodes versus data load for stable operation $r_s = 32$ Kbit/s, $b_{\text{pkt}} = 825$ bytes.

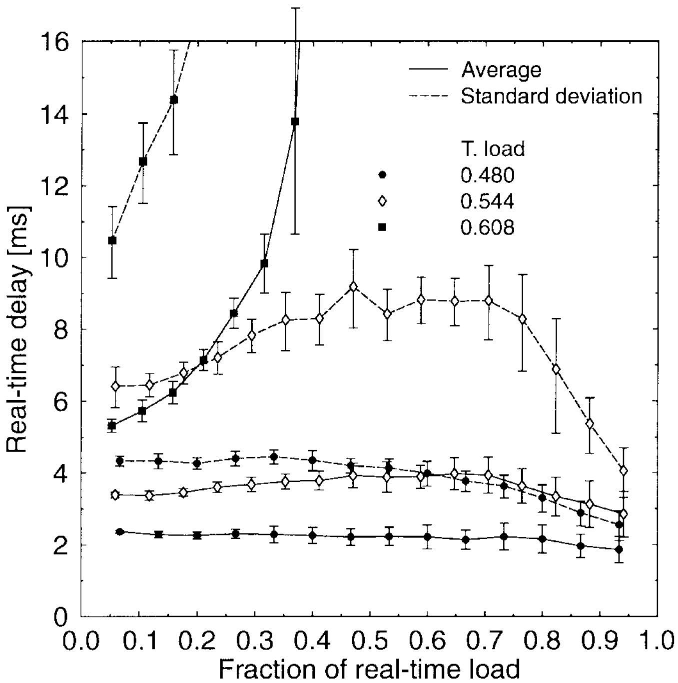


Fig. 9. Average and standard deviation of real-time packet delays under CSMA/CA versus a fraction of real-time load $r_s = 64$ Kbit/s, $b_{\rm pkt} = 825$ bytes.

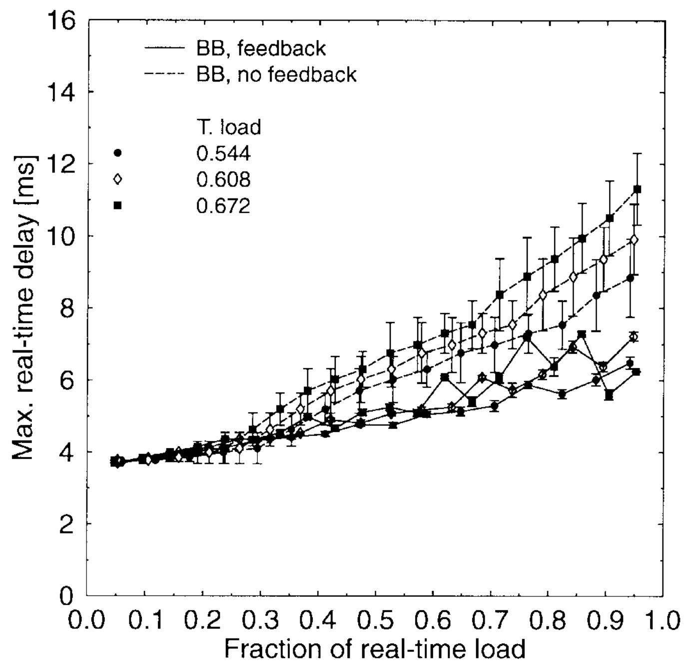


Fig. 10. Maximum real-time delay under BB contention versus a fraction of real-time load $r_s = 64$ Kbit/s, $b_{\rm pkt} = 825$ bytes.

the network, i.e., it improves the throughput. This is because real-time packets are never subject to collisions. On the other hand, under CSMA/CA data and real-time packets collide with one another, bringing down the utilization of the channel. For a data load of 0.128 and a real-time coding rate of 64 Kbit/s, BB contention supports 30% more real-time nodes than CSMA; for a coding rate of 32 Kbit/s and no chaining, the improvement is 37%.

C. Delay Performance

Real-time packet delay performance under CSMA/CA can only be characterized with statistical metrics, since no delay guarantees are made. In Fig. 9, we plot both the average and the standard deviation of real-time packet delays as a function of the fraction of real-time load for constant total (real-time plus data) load. The real-time coding rate is 64 Kbit/s, and the initial phase of a real-time packet stream associated with a session was drawn from a uniform distribution in an interpacket arrival interval of length $t_{\rm rdy} = 30$ ms. We see that the standard deviation can in many cases be twice the average, and that for practical purposes, the network cannot be operated at loads above 0.544. Fig. 10 shows the maximum real-time delay when real-time traffic contends for access to the channel with BB's. In the operation without feedback, the slack time was chosen equal to $\delta = 5$ ms. We confirm that the maximum real-time delay is typically small, even at network loads as high as 0.672. It increases with the fraction of real-time load as it reflects BB contention among real-time nodes. Table IV further shows the maximum real-time packet delay for the case of no feedback and total load 0.544 together with the percentage of real-time packets that exceed that delay when CSMA/CA is used.

It is also important to assess the extent to which the priority attained by real-time traffic with BB contention affects data delay performance. In Fig. 11, we plot the average data packet delay as a function of the fraction of real-time load for constant total load and the same parameters as before. First, we note that under CSMA/CA, the average data packet delay increases as we trade data load for real-time load. This is mainly because the relative packet overhead brought onto the channel by real-time traffic is much larger than that brought by data traffic: real-time packets and data packets contain 240 and 825 bytes of payload, respectively, but the overhead per packet is always 82 bytes. In addition, a mixture of packet lengths brings down the performance of CSMA/CA because the channel time wasted is a collision that depends on the longest packet involved in that collision, rather than on the average length of the colliding packets. More interestingly, Fig. 11 shows that when real-time packets contend with BB's, the average data packet delay does not increase as much as with CSMA/CA. Indeed, we could expect otherwise since real-time traffic is served with priority over data traffic. However, the data access procedures anticipate collisions, and consequently use the channel less efficiently than the collision-free access procedures of real-time traffic. As we trade data for real-time load, a larger volume of traffic gets priority over data, but that new traffic is efficiently served through BB contention. Not surprisingly, we also observe that operation without feedback produces larger data packet delays, since the BB's used to assert real-time access priority tend to be longer.

For the operation with feedback, the choice of the interaccess interval $t_{\rm acc}$ sets the size of real-time packets and has a significant impact on performance. As discussed in Section V-A, the real-time block delay is the sum of $t_{\rm acc}$ and the access delay. The former term is dominant in determining the overall

TABLE IV
MAXIMUM PACKET DELAY AND PERCENTAGE OF REAL-TIME PACKETS THAT EXCEED THAT DELAY UNDER CSMA/CA, FOR A TOTAL LOAD OF 0.544

Frac. of real-time load	0.12	0.24	0.3	0.47	0.59	0.71	0.82	0.94
Max. pkt. delay [ms]	3.78	4.00	4.63	5.70	6.30	6.98	7.54	8.84
CSMA delay > Max. pkt. delay	26.3%	26.8%	24.1%	19.7%	17.3%	15.3%	10.8%	6.1%

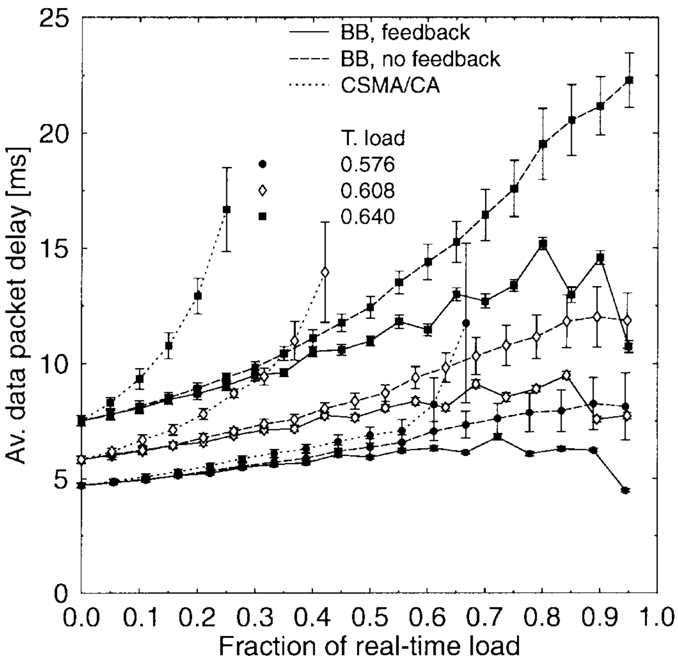


Fig. 11. Average data packet delay versus a fraction of real-time load $r_s = 64$ Kbit/s, $b_{\text{pkt}} = 825$ bytes.

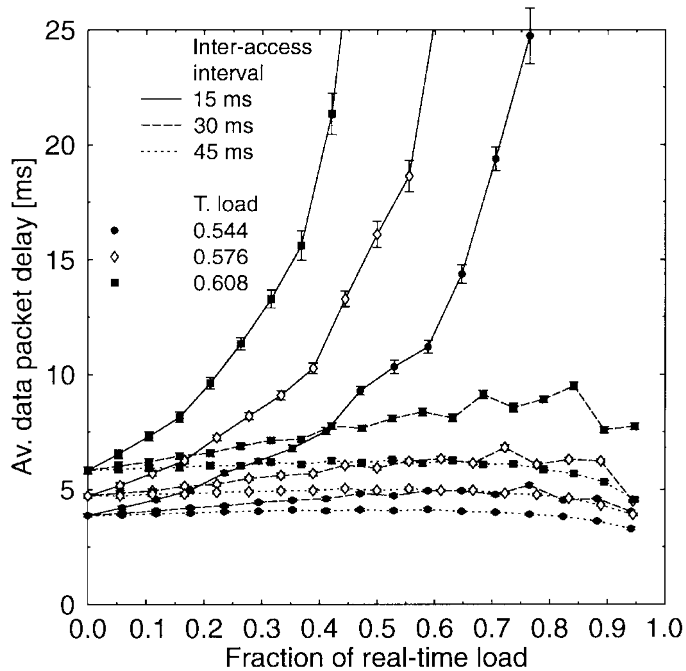


Fig. 12. Average data packet delay versus a fraction of real-time load for three values of the interaccess interval $r_s = 64$ Kbit/s, $b_{\text{pkt}} = 825$ bytes.

block delay: the smaller t_{acc}, the smaller the block delay. However, a small value of t_{acc} means small real-time packets, and those bring a large relative overhead onto the channel. As such, long data packet delays may be induced. In Fig. 12, we show the average data packet delay as a function of the fraction of real-time traffic for three values of the interaccess interval $t_{\text{acc}} = 15, 30, 45$ ms. For $t_{\text{acc}} = 15$ ms, the average data delay increases rather steeply with the fraction of real-time load, whereas for t_{acc} greater than 30 ms, the small improvement in average data delay does not justify the increase in real-time block delay.

The effect of chaining is reported in Fig. 13, where we present the average data packet delay as a function of the fraction of real-time load for a real-time coding rate $r_s = 32$ Kbit/s. Operation without chaining is compared to scenarios with two and four real-time nodes per chain. We note that for this set of parameters, chaining provides a moderate improvement in data delay performance. The improvement brought by chaining is not very significant because the system is constrained by the PHY and MAC overheads rather than by the overhead that results from BB contention. At 32 Kbit/s, each real-time packet has only 120 bytes of payload. We expect the performance to improve and the benefits of chaining to become more apparent, if either the interaccess interval is increased or the PHY plus MAC overhead is decreased. The former solution implies longer block delays for real-time applications. The latter possibility requires a better transceiver design or a simpler MAC header. As an experiment, we depict in Fig. 14 the average data packet delay as a function of the fraction of real-time load, assuming that the PHY plus MAC overhead can be reduced to 34 bytes. We verify that the average data packet delays are smaller than before and that chaining real-time packet transmissions improves the performance substantially, especially at high real-time loads.

IX. CONCLUSION

BB contention is a distributed MAC scheme designed for QoS real-time traffic support in carrier sense ad hoc wireless networks. The scheme can be overlaid on current CSMA implementations, notably those that comply with the IEEE 802.11 standard, without requiring changes to the access procedures of data nodes. Real-time nodes contend for access to the channel by sending pulses of energy, the durations of which are a direct function of the delay they experienced until the channel became idle. BB contention guarantees that real-time packets are transmitted without collisions and with priority over data packets. For an ad hoc wireless LAN, it can further be shown that BB contention enforces a round-robin discipline among real-time nodes and results in bounded real-time delays.

A mode of operation has been devised that decouples the instants when a node acquires access rights to the channel from the instants when real-time packets arrive at the MAC layer for transmission. Chaining of real-time packets was introduced as a means of increasing the performance of the system. The

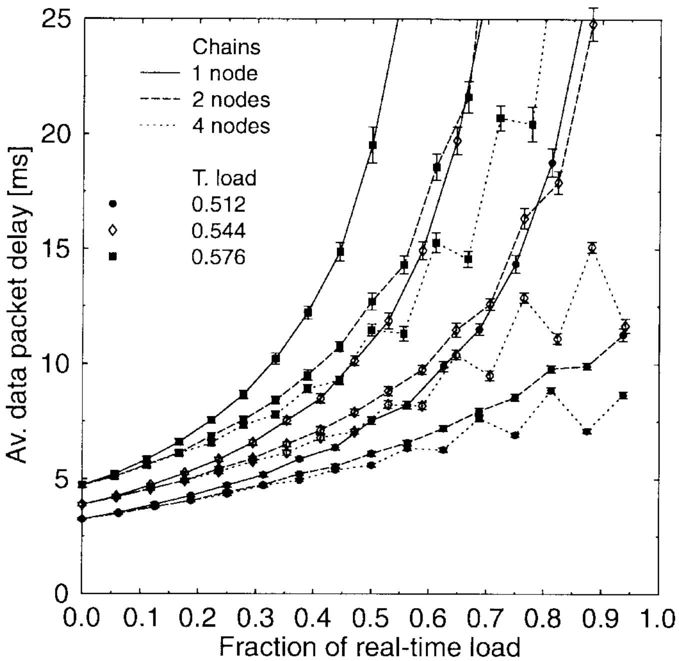


Fig. 13. Average data packet delay versus a fraction of real-time load $r_s = 32$ Kbit/s, $b_{\rm pkt} = 825$ bytes.

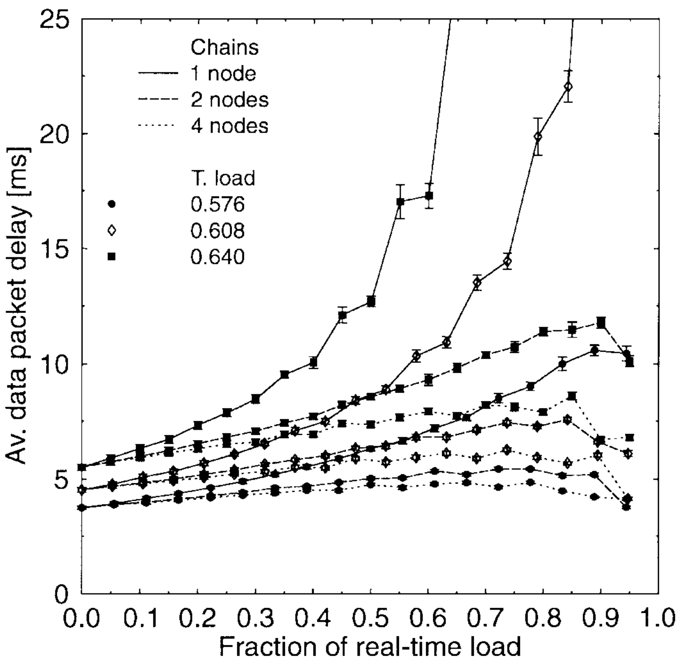


Fig. 14. Average data packet delay versus a fraction of real-time load, $r_s = 32$ Kbit/s, $b_{\rm pkt} = 825$ bytes, and reduced PHY plus MAC overhead.

processes of chain creation and segregation are distributed and resilient to failures.

A general analytical framework to study the dynamics of BB contention has been presented. In particular, the analysis yields conditions for the BB mechanism to be stable. Simulations were also conducted to compare the performance of BB contention with that of CSMA/CA. The former mechanism can carry more traffic than CSMA/CA. More importantly, it makes QoS guarantees in terms of maximum real-time delay without significantly deteriorating average data delay performance.

ACKNOWLEDGMENT

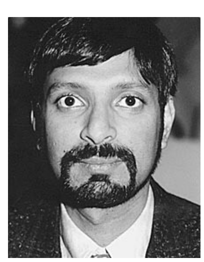
The authors would like to thank the reviewers for their helpful comments. Their remarks greatly improved the presentation of the paper.

REFERENCES

[1] L. Kleinrock and F. A. Tobagi, “Packet switching in radio channels, part I—Carrier sense multiple-access modes and their throughput-delay characteristics,” *IEEE Trans. Commun.*, vol. COM-23, pp. 1400–1416, Dec. 1975.
[2] B. Leiner, D. Nielson, and F. A. Tobagi, Eds., *Proc. IEEE*, Special issue on packet radio networks, vol. 75, Jan. 1987.
[3] L. Kleinrock and F. A. Tobagi, “Packet switching in radio channels, part II—The hidden terminal problem in carrier sense multiple access and the busy tone solution,” *IEEE Trans. Commun.*, vol. COM-23, pp. 1417–1433, Dec. 1975.
[4] P. Karn, “MACA—A new channel access method for packet radio,” in *Proc. ARRL/CRRL Amateur Radio Ninth Computer Networking Conf.*, ARRL, 1990, pp. 134–140.
[5] C. Fullmer and J. Garcia-Luna-Aceves, “Floor acquisition multiple access (FAMA) for packet-radio networks,” in *Proc. SIGCOMM’95*, Cambridge, MA, pp. 262–273.
[6] C. Fullmer and J. Garcia-Luna-Aceves, “Solutions to hidden terminal problems in wireless networks,” in *Proc. SIGCOMM’97*, vol. 2, Cannes, France, pp. 39–49.
[7] A. Muir and J. Garcia-Luna-Aceves, “Supporting real-time multimedia traffic in a wireless LAN,” in *Proc. SPIE Multimedia Computing and Networking 1997*, San José, CA, pp. 41–54.
[8] R. Garcés and J. Garcia-Luna-Aceves, “Collision avoidance and resolution multiple access with transmission groups,” in *Proc. IEEE INFOCOM’97*, Kobe, Japan, pp. 134–142.
[9] C. Lin and M. Gerla, “Asynchronous multimedia multihop wireless networks,” in *Proc. IEEE INFOCOM’97*, Kobe, Japan, pp. 118–125.
[10] J. L. Sobrinho and A. S. Krishnakumar, “Real-time traffic over the IEEE 802.11 medium access control layer,” *Bell Labs Tech. J.*, vol. 1, pp. 172–187, Autumn 1996.
[11] *IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard 802.11, 1997.
[12] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge: Cambridge Univ. Press, 1985.

João L. Sobrinho (M’97) received the Engenheiro Electrotécnico degree in 1990 and the Ph.D. degree in electrical and computer engineering in 1995, both from Instituto Superior Técnico (IST), Lisbon, Portugal.

He is presently an Assistant Professor at IST and a Researcher at the Telecommunications Institute (IT), Lisbon, Portugal. From 1995 to 1996, he was a Member of the Technical Staff at the Wireless Networking and Mobile Computing Research Department, Bell Laboratories, Lucent Technologies. His current research interests are in quality-of-service over packet-switched networks, wireless and wired alike.



A. S. Krishnakumar (M’91) received the B. Tech. degree in electrical engineering from the Indian Institute of Technology, Madras, India, the M.S. degree in electrical engineering from Northwestern University, Evanston, IL, and the M.S. degree in statistics and the Ph.D. degree in electrical engineering, both from Stanford University, Palo Alto, CA.

He is the Head the Networked Multimedia Research Department at Lucent Tecnologies, Bell Labs, Murray Hill, NJ. His research interests include high-speed wireless data networks and wireless multimedia integration.