QoS over Real-Time Wireless Multi-hop Protocol

Domenico Sicignano*, Danilo Tardioli, and José Luis Villarroel

Grupo de Robótica, Percepción y Tiempo Real
Instituto de Investigación en ingeniería de Aragón
Universidad de Zaragoza, Zaragoza, Spain
{d.sicignano,dantard,jlvilla}@unizar.es

Abstract. This paper proposes a way to incorporate multimedia traffic in a real-time wireless communication network without jeopardizing the hard real-time traffic. This idea has been implemented and analyzed as an extension of the real-time multi-hop protocol (RT-WMP), a novel protocol that supports hard real-time traffic in relatively small ad-hoc networks. The protocol allows merging the real-time traffic coming from cooperative multitask robot teams and human communication such as video and voice. The quality of service (QoS) extension takes advantage of the bandwidth left free by the RT-WMP when it is not working in the worst-case situation. Real tests involving multi-robot data exchange and multimedia communication show that the extension can be perfectly integrated in the protocol and offers a suitable QoS transmission mechanism for real-time multi-hop networks.

Keywords: MANET, Real-Time Networks, Quality of Service (QoS).

1 Introduction

A Mobile Ad-hoc NETwork (MANET) is a collection of mobile nodes that communicate with each other over radio channels in the absence of an infrastructure. Nowadays, its widespread use confirms the need of supporting multimedia traffic. As a result, research has proposed several methods to offer some kind of Quality of Service (QoS) on MANETs.

In robotics applications, communication is generally provided by a wireless infrastructure. In certain fields of application such as emergencies, rescue situations, battlefields or hostile environments, cooperative teams of robots and humans need to be able to communicate using MANETs. To cooperate, robots must exchange information about their own state and environment, time constraints being the key feature. Moreover, human communication should use the same MANET as the robots, and the system must be able to support some class

* This work has been funded by the projects NERO DPI2006-07928 (MCYT, Spanish Gov.) and URUS IST-1-045062-URUS-STP (E.C.) and by the agreement between the Gobierno de Aragon and the Universidad de Zaragoza regarding the WALQA research laboratory.

J. Zheng et al. (Eds.): ADHOCNETS 2009, LNICST 28, pp. 110–128, 2010.

of QoS. In these situations the use of a real-time capable network is mandatory to allow distributed perception and prioritized information flows. In some situations like, for example, rescue tasks involving humans (in the event of collapsing buildings or fire) the possibility of establishing some type of communication with the victims is very useful both for easing the access to the disaster zone and obtaining information about the status of the people involved (e.g. audio streaming). In other situations, (e.g. telemanipulation, access to inaccessible zones, etc.) visual information (photo or video streaming) could be more effective. Both flows of information, however, have quite-strict time requirements and, at first sight, could be thought of as real-time flows. It might therefore be considered to be a good idea to take into account these flows at the planification time and treat them as normal real-time flows. However, on the one hand these flows are quite bandwidth consuming (even if some audio-streaming codec is capable of rates of about 3Kbps, time requirements force to reserve a wider bandwidth) and in some specific situations might not be possible (depending on the saturation of the real-time bandwidth) while, on the other hand, not all the frames necessarily have to be delivered. As an example, the iLBC [1] or speex [2] audio codecs guarantee, at low bit rate, a MOS (Mean Opinion Square) greater than 3.3 and 2.5 respectively with a packet delivery ratio (PDR) of about 95%. On the one hand, then, it seems there is little sense (or it may even impossible in some situations) the use of real-time bandwidth to transport information that does not have this type of requirements while on the other hand there are certain situations in which we need these types of flows.

In this paper we propose a novel solution to incorporate multimedia traffic in a real-time wireless communication network without jeopardizing the hard real-time traffic. This idea has been implemented and analyzed as an extension of the RT-WMP [3] hard real-time protocol. The rationale is to take advantage of the bandwidth left free by the protocol when it is not working in the worst-case situation and use it to send QoS frames to allow audio and video streaming flows.

In the following section we present a brief review of related work. Section 3 summarises the basic features of the RT-WMP protocol. The proposed QoS extension is introduced and explained in sections 4 and 5. Section 6 presents the Flow Admission Control scheme. The evaluation, by means of real experiments, is presented in section 7. Finally, section 8 sets out the conclusions.

2 Related Work

Token passing medium access control protocols for Ad-Hoc networks have been gaining popularity in recent years due to the advantages they offer such as deterministic network access guarantees, robustness against single node failure and support for flexible topologies.

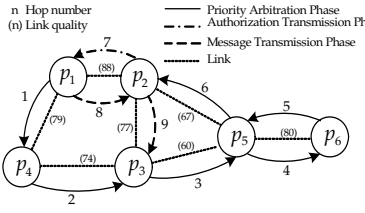
Several papers have proposed the token based scheme in Ad-Hoc networks. The majority of them implement similar MAC operations. The network generates a unique token that permits only the node currently holding it to transmit data.

In [4] the potential of achieving higher channel utilization using a token scheme with respect to CSMA based schemes is shown. More recently, in [5], the advantage has been analyzed of a token based scheme over contention based and centralized polling schemes to provide guaranteed priority for different traffic classes in WLAN. A similar analysis is conduced in [6] which shows how a token ring scheme applied in vehicular ad-hoc networks can outperform IEEE 802.11 DCF in terms of average throughput.

Based on the ideas of the 802.4 token bus protocol, in [7] the authors propose the wireless token ring protocol (WTRP), a token ring network in which each node can transmit for a fixed time when it owns the token. Although multiple rings are allowed, a node only communicate with its neighbor, so the topology of the network is limited. The wireless dynamic token protocol (WDTP) [8] modifies the method to control the token transfer scheme of the WTRP. All nodes are clustered into subnets and the nodes of a subnet share a channel. This improves the adaptability to the network topology but the number of used channels increases. Some proposals are based on hybrid MAC Token CDMA policing mechanisms. Taheri and Scaglione's [9] proposal is based on a ring network where each token corresponds to a physical CDMA subchannel which is guaranteed to have a certain average rate and satisfies a probability of error bound. In [10] the authors propose an interesting spatial reuse solution based on CDMA modulation to allow a delay-bounded protocol. CDMA is also used in [11] which, based on the ideas of the 802.4 token bus protocol, offers a virtual ring topology where the combination of a token passing and CDMA scheme allows transmissions at the same time and avoid collisions. Unfortunately these solutions are based on uncommon consumer CDMA devices, normally used in mobile phones. As an extension to 802.11e, there are two interesting proposals that provide QoS over multi-hop traffic. In [12] the authors add a QoS mechanism to the enhanced distributed channel access (EDCA) scheme to allow a resource reservation. In [13] packets are prioritized using a combination of the laxity of the packet and the number of hops to the destination node to give higher priority to the packets that have to traverse many hops. However, these solutions suppose a modification of the 802.11e protocol and they have been designed to deal with multimedia traffic that has slightly different requisites than real-time traffic.

3 RT-WMP Overview

The Real Time Wireless Multi-hop Protocol (RT-WMP) is a protocol for MANETs. It works on top of the 802.11 protocol and supports real-time traffic. In fact, in RT-WMP, end-to-end message delay has a bounded and known duration and it manages global static message priorities as well. Besides, RT-WMP supports multi-hop communications. The protocol has been designed to connect a relatively small group (10-20 units) of mobile nodes. It is based on a token passing scheme and is designed to manage rapid topology changes through the exchange of matrix containing link quality amongst nodes. RT-WMP has an error recovery mechanism that can recover from certain types of errors without jeopardizing real-time behavior and a has also a technique for reincorporating lost nodes.



Link Quality Matrix

	p_1	p_2	p_3	p_4	p_5	p_6
p_1	-	88	0	79	0	0
p_2	88	-	77	0	67	0
p_3	0	77	-	74	60	0
p_4	79	0	74	-	0	0
p_5	0	67	60	0	-	80
p_6	0	0	0	0	80	-

Fig. 1. A hypothetical situation described by the network graph and the corresponding LQM. The hops sequence of the protocol is also shown.

Protocol Operations. The protocol works in three phases (see Fig. 1): *Priority Arbitration Phase (PAP)*, *Authorization Transmission Phase (ATP)* and *Message Transmission Phase (MTP)*. During the PAP, nodes reach a consensus over which of them holds the Most Priority Message (MPM) in the network in that moment. Subsequently, in the ATP, an authorization to transmit is sent to the node which holds the highest priority message. Finally, in the MTP, this node sends the message to the destination node. To reach a consensus over which node holds the highest priority message, in the PAP a token travels through all of the nodes. The token holds information on the priority level of the MPM in the network and its owner amongst the set of nodes already reached by the token. The node which initiates the PAP states that the highest priority message in its own queue is the MPM in the whole network and stores this information in the token. Then it sends the token to another node, which checks the messages in its own queue. If the node verifies that it holds a message with a higher priority level than the one carried by the token, it modifies the token data and continues the phase. The last node to receive the token, which knows the identity of the MPM holder, closes the PAP and initiates the ATP. In this phase, the node calculates a path to the MPM holder using the topology information shared amongst the members of the network (the Link Quality Matrix, see below) and sends an authorization message to the first node in the path. The latter will route the message to the second node in the path and so on, until the authorization reaches the MPM holder. This is when the MTP begins. The development of this phase is quite similar to the preceding one. The node that has received the authorization calculates the path to reach the destination, and sends the message to the first node of the path. The message follows the path and eventually reaches its destination. The phases are repeated one after another i.e., when the MTP finishes, the node destination of the message initiates a new PAP and so on. When none of the nodes has a message to transmit, the authorization and message transmission phase are omitted and the priority arbitration phase is continuously repeated.

The Link Quality Matrix. To describe the topology of the network, RT-WMP defines an extension of the network connectivity graph (as defined in [14]) adding nonnegative values on the edges of the graph. These values are calculated as functions of the *radio signal* between pairs of nodes and are indicators of the

link quality between them. These values are represented in a matrix called the Link Quality Matrix (LQM) (see Fig.1). Each column describes the links of a node with its neighbors. The nodes use this matrix to select which node to pass the token to and to take decisions on the best path to route a message from a source to a destination. All the nodes have a local copy of the LQM that is updated each time a frame is received. Besides, every node is responsible for updating its column of the LQM (both in the local copy and the shared copy) to inform the other nodes about local topology changes.

Error Handling in RT-WMP. RT-WMP is quite robust in the case of node failure. The *implicit acknowledgement* technique used dispenses with the necessity of monitor nodes to control the loss of the token. In RT-WMP, in fact, when a p_k node sends a frame of any type, it listens to the channel for a timeout. The receiver p_l node immediately processes the frame received and sends another frame to a third p_m node. The first sender listens to such a frame as well and interprets it as an acknowledgement. If the first sender does not hear the frame within timeout, it supposes that the p_l node has fallen or is out of its coverage range. In this case, the behavior depends on the phase that the protocol is in. If it is in the ATP or MTP, p_k discards the frame and starts a new PAP. However, if it is in the PAP, the p_k node sends the token to another node to continue the PAP without jeopardizing the temporizations (see [3] for details). Communication errors can produce another type of problem. Let us consider the situation where, in the PAP, the p_k node sends a token to the p_a node and waits for an implicit acknowledgment. Node p_a processes the frame and sends the frame to node p_b. As explained earlier, the last pass is also the acknowledgement for p_k. However, if node p_b hears the frame but p_k does not, a token duplication occurs. In fact both nodes p_k and p_b continue the PAP and at that moment there are two tokens in the network. This problem was solved introducing a *serial* field in the frames. In this way if a node receives frames with old serials, it discards them and informs the sender.

Worst-Case in RT-WMP. The PAP, ATP and MTP phases have a bounded duration. The PAP lasts, in the worst case, $2n-3$ hops. In fact, if the network is connected, a covering tree with $n-1$ arcs can always be found, so the tree can be covered by visiting all its nodes two times at the most. That would mean $2n-2$ hops, but a return to the first node can be avoided; therefore, there are only $2n-3$ hops. In the ATP and MTP, the path is determined using the Dijkstra algorithm. According to this algorithm, if the network is connected, the maximum number of hops to go from one node to another is $n-1$. From the global point of view, phases of the RT-WMP protocol are repeated one after another, with worst-case durations $t_{pa} = (2n-3)t_t$ for the PA phase, $t_{at} = (n-1)t_a$ for the ATP and $t_{mt} = (n-1)t_m$ for the MTP, t_t being the duration of a *token* pass, t_a the duration of an *authorization* pass and t_m the duration of a *message* pass. The absolute values of t_t, t_a and t_m depend on protocol parameters such as the number of nodes, the data rate and the maximum transmission unit (MTU) that the network has to carry, as well as on the underlying 802.11 protocol.

According to [3], the transmission of any frame takes:

$$t_{frame}(\mu s) = (192 + 50) + \frac{(28 + L) \cdot 8}{tx\_rate(bps)} \quad (1)$$

L being the size of the frame to be sent, now constituted by the standard RT-WMP frame plus the extra frame added by the QoS extension. So the RT-WMP protocol worst-case loop can be expressed as:

$$T_{WC} = t_{pa} + t_{at} + t_{mt} \quad (2)$$

4 System Overview

As mentioned earlier, the rationale is to take advantage of the bandwidth left free by the protocol when it is not working in a worst case situation. During the operations of RT-WMP, the PAP, ATP and MTP phases are continuously repeated one after the other. As explained in [3], their duration depends on the number of hops that the frame executes in each one of them. This value depends, in turn, on the network topology and on the position of the source and destination of any concrete message. In general it is unforeseeable but, of course, there exists an upper bound that represents the worst-case *loop* time (the latter defined as the succession of a consecutive PAP, ATP and MTP). As explained in [3], the worst-case end-to-end delay can be expressed as twice the worst-case loop. This is the value that must be used for the scheduling of a distributed real-time system using RT-WMP. During real-time analysis it must be assumed that the protocol could work all the time in its worst-case situation and, thus, offer its worst-case performance. In this way we are assuring that the deadline will always be honoured even in the worst operational conditions.

However, worst-case loops are unlikely to happen in practice and even with unfavorable network topologies they occur only in a small percentage of loops. In other words, in the majority of the cases the RT-WMP closes its loops in a time lower than the worst-case one and in some cases (if the real-time traffic bandwidth usage is below one-hundred percent) the loop consist uniquely of the best-case PAP. The rationale described in this paper consist of using, in any loop, the time slot between the real loop-duration and the worst case loop duration to send QoS information. In other words, we are forcing the protocol, when one or more QoS flows are present, to operate in the worst-case loop taking advantage of the fact that the worst case will take place in very few situations. It *does not worsen*, by design, the worst-case end-to-end so can be used in any real-time network to add QoS capabilities maintaining the same worst-case performances.

4.1 Available Time

The available time in any loop depends on several factors such as the relative position of source and destination, the network topology and so on. The duration

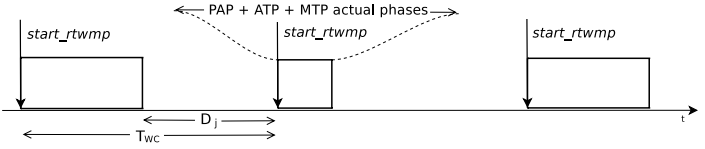


Fig. 2. Time intervals used by the QoS Extension

of the real RT-WMP loop t_j is *less than or equal* to the worst-case loop T_{WC}. Fig. 2 illustrates the situation. The available time D_j can be expressed as:

$$D_j = T_{WC} - t_j \tag{3}$$

While the minimum value of D_j is zero, the maximum corresponds to the situation in which the RT-WMP loop is only constituted by the PAP. In this case $D_j = T_{WC} - T_{PAP}$.

4.2 Protocol Operations

All the nodes use a single *shared* radio channel to exchange messages. To each node has been added a QoS transmission and reception queue (QTQ and QRQ respectively). Each QoS message has a deadline that is fixed by the application and that represents the time during the which the message is valid.

This QoS extension, has three phases: an arbitration phase, a QoS Authorization Phase (QAP) and a QoS Message Phase (QMP) that can be repeated one after the other for a limited amount of time. The arbitration phase is carried out during the PAP while the QAP and QMP, are added to the basic protocol.

In the arbitration phase *all* the nodes which have a QoS message waiting in the QTQ compete to gain the right to send it (during the PAP the token reaches all the nodes). One or more messages can be selected for transmission depending on their deadline and on the distance between the source and the destination nodes, as will be explained below. The address of the nodes owner of these messages are stored in the header of the frames. The first QAP starts when the standard MTP ends (or after the PAP if there is no real-time message to be sent). The node which ends the MTP (or the PAP), instead of restarting the successive PAP, sends an authorization to the owner of the first selected message (indicated in the header), using the same scheme used by the RT-WMP. The latter, then starts the QMP and sends the QoS message to the destination node that pushes the message into its QRQ. Successively, if the header specifies that there are other messages to be sent, it prepares an authorization and starts another QAP during which the message reaches the node owner of the selected message. This, in turn, sends its message during a further QMP and so on. As has been stated, the QAP and QMP are been repeated one after the other for a limited (and configurable) number of times, but in any case they stop when the worst-case loop time is reached. This behavior is obtained by loading a field of the header with the duration, expressed in milliseconds, of the worst-case loop

and subtracting, in any frame-pass, the time spent on this action. When this value is lower than the time needed to execute the next frame-pass, the QAP or QMP ends and a normal PAP restarts. If the PAP has to restart during a QMP, the transported message is stored in the QTQ of an intermediate node to be able to compete for selection again in the successive PAP.

The QoS extension implements eight flow priority classes where class zero corresponds to best-effort not-QoS traffic. Flows are served following their priority level. Audio flows, for example, usually have priority over video flows since audio information is more delay sensitive. The introduction of a flow in the protocol is regulated by the Flow Admision Control (FAC) that allows or denies access, taking into account the priority of the requesting flow and the estimated available bandwidth (EABW). The EABW is obtained estimating the global Loop Remaining Time (LRT) in the time units (the latter defined as the difference between the duration of the worst-case loop and the duration of a complete loop, including QAP and QMP phases). In other words, it is computing the percentage of free time that would be statistically available for extra QoS Flows, in the time unit. The FAC takes into account the priority of the flow, that is it only considers as unavailable the time occupied by higher priority flows.

5 The RT-WMP QoS Extension Details

In Real-Time distributed systems, bounded end-to-end delay and priority support are required for scheduling and time constraint guarantees. Since the plain RT-WMP is a real-time protocol, each event/phase protocol has a bounded and known duration even in the presence of the majority of errors. Loops have, for example, an upper bound on their duration that can be easily calculated and that is used as a base to calculate the loop remaining time, as will be explained with more details below. Several fields have been added to the header of the basic RT-WMP frames to implement the proposed extension.

5.1 Frame Header Modification

Fig. 3 shows the RT-WMP frame with the fields that support the QoS extension. The *qos_rem* field is a 2 byte field that is filled, at the beginning of any loop, with the duration, expressed in milliseconds of the worst-case RT-WMP loop. The *ac_loop_id*, *ac_pri*, and *ac_lot* are service fields used by the access control system to estimate the available QoS bandwidth. The next fields are used to identify the selected messages. All of them are (compile-time) configurable size

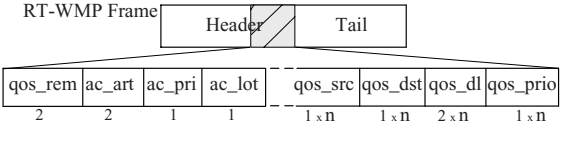


Fig. 3. Frame for the RT-WMP with QoS extension

vectors. Their size is application and network-size dependent and represents the maximum number of QoS messages that can be selected (and potentially delivered) in any loop. The *qos_dl* (2 bytes per message) contains the actual deadline of the packets (relative deadline to the actual moment) while the *qos_src* and *qos_dst* (1 byte per message) specify their source and destination. These last three fields are used to calculate the dynamic priority of the message that depends on the deadline and the distance between the source and destination of a message. The last *qos_pri* field (1 byte per message) carries the priority class of the selected message.

5.2 Phases of the Protocol

In this section a detailed description of the different phases of the protocol is presented including some implementation issues than condition the development of the protocol.

The Message Selection Phase. The first phase of the protocol takes place simultaneously with the PAP of the basic RT-WMP. In fact, in this phase the QoS extension does not alter the operations of the basic protocol (tokens are exchanged in the usual way). In this phase, the QoS messages to be transmitted in the successive phases are selected. In each node the QTQ contains all the QoS messages ordered by priority (see sec. 5.3).

The node that starts the PAP analyses the QTQ. Above all, it discards the expired messages. It then obtains the class flow, the deadline and the destination of the n most priority messages (n being the maximum number of QoS messages that can be delivered in any loop) and fills the correspondent fields of the token header. Moreover it calculates the worst-case loop duration and fills the field *qos_rem* of the header with this value expressed in milliseconds. Successively the basic protocol is responsible for sending the token to another node. The node that receives the token processes the basic part of the token as usual. It then actualizes the values of *qos_rem* and the *qos_dl* subtracting the time spent in the last token-pass. It successively calculates the new priority for the messages referenced by the token. This step is necessary because the change in the deadline implies a change in priority. It then again discards expired messages and compares the n most priority messages in the QTQ with those carried by the token. If figures out that owns one or more priority messages, and replaces the less priority with its own updating of the *qos_dl*, *qos_src* and the *qos_dst* fields. In the same way, the process is repeated up to the moment in which the last node of the network is reached. At this moment the node starts the ATP and the MTP (if real-time messages have been selected to be sent). In these two phases, there is no participation of the QoS extension except for the fact that the *qos_dl* field is decreased by the quantity corresponding to the time spent in any frame pass.

The QoS Authorization Phase. This phase starts after the conclusion of the MTP (if any) or the PAP or even after a QMP. The node that starts the

QAP prepares an authorization as in the basic protocol (see [3]), fills the *aut_src* with its address and *aut_dest* with the first element of the *qos_dst* vector, shifts by one the position of the *qos_dst*, *qos_dl*, *qos_pri* and *qos_src* vector elements (*qos_dl[0]=qos_dl[1], qos_src[0]=qos_src[1], etc.*) and sends the frame.

The authorization is propagated using the same routing algorithm of the basic protocol (Dijkstra based algorithm) until it reaches the destination. In any hop, however, the *qos_dl* and *qos_rem* fields are actualized subtracting the duration of any frame-pass. If at some moment the *qos_rem* field reaches a zero value (or a value that does not allow a further frame-pass), the QAP is immediately aborted and another PAP is started.

The QoS Message Phase. When a node receives a QoS Authorization, the QMP starts. It pops the most priority message from the QTQ and creates a new message frame placing data in the *message* field. It fills the *src* and *dest* fields with its address and with the destination address and calculates the path to the destination. Then it sends the message to the first member of the path as in the RT-WMP basic protocol. When the latter receives the message, it checks the *msg_dest* field. If it is not the destination node (i.e. if it is an intermediate node), it verifies if there is enough remaining time to forward the message to the next node of the path (i.e. the value of *qos_rem* is at least greater than the time needed for one message-hop). If this is the case, the node repeats the computation of the path and routes the message to the next member of the path, leaving the *dest* field unchanged. Otherwise, it pushes the message into the transmission QoS queue and starts a new PAP. In this case the message will compete to be selected for transmission again in the next PAP. If the message reaches the destination in a single loop, there is the chance of sending another QoS message. When the node receives the QoS message, it pushes it into the QRQ. It then looks at the *qos_rem* field. If it is assured that there is enough time to authorize another node and to allow at least one message-hop, it starts another QAP that, in turn, will cause another QMP and so on.

5.3 Message Priority Policy

Packet Deadline. Timing guarantees can be considered as a fundamental feature for MANETs able to support QoS applications. Each message has to be delivered before its deadline. Delay bounded service allows the protocol to know whether it is able to meet the deadlines or not.

Any QoS packet has an associated deadline by which it must be delivered. If this is not possible, the packet must be discarded. This deadline value is only needed until the packet is either successfully transmitted or discarded.

The mechanism to label and update the deadline on every QoS message is quite simple. Messages are labelled with their maximum admitted deadline, depending on the nature of message. Deadline values are usually 150 ms for voice and 400 ms for video traffic that correspond to maximum end-to-end delays admitted for multimedia traffic [15]. When this parameter reaches the zero value, the message expires. So, during multi-hop transmission, this value has to be

properly updated. Every node over the source-destination path updates packet deadlines taking into account the elapsed time and the transmission time of one-hop.

Packet Scheduling. The QoS extension implements a packet scheduler that assigns a dynamic priority to a packet taking into account the flow class, the deadline and the number of hops left to the destination, in this order. Above all, the scheduler sorts the packets according to their class flow. Messages in the same flow-class are sorted using the *laxity* that is a parameter that combines the deadline and the number of hops left to the destination as $laxity = deadline/hopleft$.

Instead of transmitting packets in the FIFO order (as in the case of 802.11e) or EDF order, we prioritize the packets with respect to the laxity. In fact, in 802.11e for example, a packet whose destination is one hop away has the same possibility of capturing the channel as a packet whose destination is several hops away. So, locally it does not take into consideration the number of hops a packet has to cross. However, the laxity gives us an estimate of how much delay the packet can tolerate at each hop. Hence, the packet with the lowest value of laxity is given the highest priority. If two packets have the same lowest value of laxity, we resolve the conflict by sending the packet which has more hops to travel. If the laxity value becomes zero, the packet is discarded since it is useless at the destination.

6 Flow Admission Control

The available bandwidth for QoS flows is limited and depends on different factors such as real-time flow saturation, network topology and so on. Thus, it is important to control the admission of new QoS flows in a real-time network since if the available bandwidth is not enough, it is possible to jeopardize the correctness of the whole set of flows. As an example, consider the situation in which in the network there already exists a 15Kbps flow and the global available bandwidth for QoS flow is about 20 Kbps. If we try to introduce another 15 Kbps flow of the same class, the system will distribute the available bandwidth between the two flows lowering the rate of both to 10 Kbps discarding the messages that cannot be delivered within the deadline. It would not be enough for a correct streaming and both flows would be useless. To avoid these problems we have developed a Flow Admission Control (FAC) system that estimates and manages the available bandwidth. The idea is to compute if there exist enough bandwidth for a given new flow. The admission control works in accordance with the flow *class*.

6.1 Available Resource Estimation

To estimate the available bandwidth for new QoS flows, the network is observed during a time window that contains several RT-WMP loops. We call this sliding window the *Available Bandwidth Estimation Interval* (ABEI). The width of the

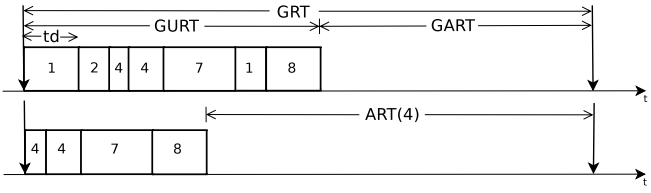


Fig. 4. Resource estimation mechanism

ABEI is configurable and a good choice is usually related to the hyperperiod of the underlying real-time distributed system.

Over an ABEI the *Global Remaining Time* (GRT) is calculated as:

$$GRT = \sum_{j:loop_j \epsilon ABEI} D_j. \tag{4}$$

The GRT represents the sum of all the remaining time D_j that is included in the ABEI. In other words, the GRT is a measure of the available time for QoS flows. Any QoS flow occupies a portion of this global available time. We call the sum of all the occupied portions Global Used Remaining Time (GURT) that can be expressed as:

$$GURT = \sum_{j:loop_j \epsilon ABEI, k \epsilon [0..7]} td_j(k) \tag{5}$$

$td_j(k)$ being the time consumed by a k class flow in a loop j (see Fig.4). In a similar way it is possible to define the Global Available Remaining Time (GART) as $GART = GRT - GURT$.

The GART represents the time still available subtracting the time occupied by already-active QoS flows. However, this access control scheme relies on flow classes, that is, higher priority flows can expel lower priority ones. In the light of this, the GART can be considered as the available time for the least-priority flow in the system at any moment. The available time for a given class flow is instead called the Available Remaining Time (ART). It can be expressed as:

$$ART(c) = GRT - \sum_{j:loop_j \epsilon ABEI, k \geq c} td_j(k) \tag{6}$$

c being the class of the flow that is requesting access.

If a flow requests access to the system, the FAC calculates the ART for the class flow of the flow requesting access and estimates (using a heuristic) whether there is enough bandwidth to allow the access.

Principle of Operations. When a node closes a PAP, it stores in a local vector the value of *qos_rem* together with a timestamp. Next, it fills the *ac_lot* field with

the value of the *qos_rem* field. Successively, when a QoS Message is delivered, if k is the class flow of the message, the receiver computes the time spent to deliver the last message with the formula:

$$td(k) = ac\_lot - qos\_rem \tag{7}$$

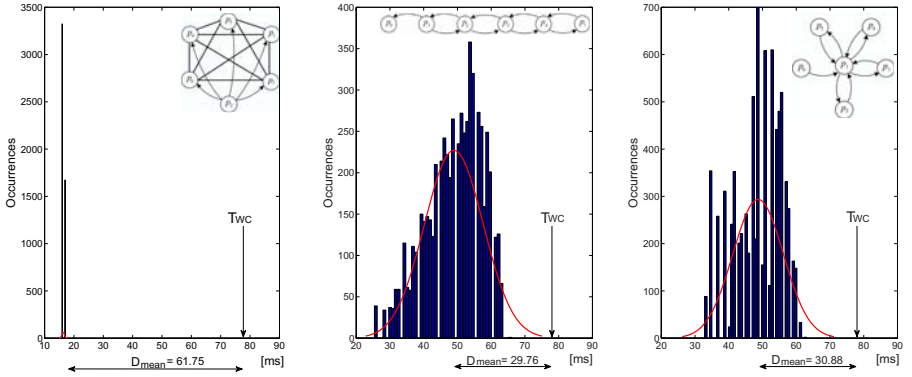
The node stores td in a local vector together with the class of the message just received. Then it actualizes the value of the *ac_lot* with the present value of *qos_rem* and continues the operations with another QAP or a new PAP. The process is repeated in any loop and the nodes accumulate, but in a distributed fashion, all the information about message delivery times and remaining times. In fact, none of the nodes has a global view of the available time in the network. However, the sum of all the elements of the first vector of all the nodes represents the GRT and the sum of all the elements of the second vector represents just the time consumed by all the active flows in a specific moment (i.e. the GART).

When a node needs to add a new flow into the network, it requests that it specify the class of the new flow in the *ac_pri* field (that normally contains a negative value). In the successive PAP, all the nodes analyse their vectors with respect to the values stored in the ABEI window. Specifically they sum all the values of the first vector whose timestamp is contained in the ABEI and subtract all the values of the second vectors whose timestamp is contained in the ABEI *and* class is greater or equal to the one requesting the flow. The result of this computation is added to the values of the *ac_art* field (that normally contains a null value). When the token has reached all the nodes, the *qos_art* field contains the Available Remaining Time (ART) of the given class flow.

Fig. 4 shows the rationale behind the procedure. The global time left free by the protocol in an estimation interval is consumed by the time spent to deliver QoS messages of any class. However, when a message requires access, only higher classes messages are considered in order to calculate the ART for this class flow. When in the next PAP the token again reaches the requesting node, it analyses the value contained in *ac_art*. Using a simple heuristic, the node decides if the requesting flow is admissible. If this is the case, it allows the application to begin the new stream.

Table 1. Parameters used in the real tests

	Parameter	**Values**
Scenario	Channel rate	1 Mbps
	Number of nodes	6
Data	Real-Time pkt	128-256-512 Byte
	QoS rate per flow	15 Kbps
Constraints	Deadline	150 ms
	Queues size	50 pkts



(a) (b) (c)

Fig. 5. Time spent for the RT-WMP in real test compared to worst case for different topologies

7 Evaluation

The aim of our experiments is to examine the impact of the proposed extension on the RT-WMP protocol. Several real tests have been made using an implementation of RT-WMP executed over the MaRTE OS [16] real-time operating system. A total of six nodes equipped with Intel Pentium IV CPU at 2.5GHz, 2GB RAM and Ralink RT61 chipset-based wireless cards have been used. To evaluate the correctness, the performance and the behavior of the protocol extension, we have performed some real experiments. Table 1 lists the parameter values used in the tests.

7.1 Available Time

The proposed extension uses the available time $D_j = T_{WC} - t_j$ to transmit the QoS traffic. Fig. 5 shows the time spent (t_j) by RT-WMP in the trasmisison of the real-time messages versus the worst case loop time (T_{WC}), for some network topologies. The protocol has been forced to work with saturated real-time traffic (i.e, there is a real-time message in each RT-WMP loop). Fig.5(a), 5(b), and 5(c) allow us to evaluate the actual time D_j that RT-WMP can make available to the QoS extension when the PAP, APT and MTP phases are finalized. The figure also shows the average available time D_{mean}.

In the following tests we show, with the parameters of table 1, the nodes configuration was made to simulate a chain (as shown in Fig. 5(b)). The set of RT-WMP source-destinations have been generated in a uniformly random manner whereas QoS traffic has been generated such that the first node sends traffic to the last node to test an end-to-end communication.

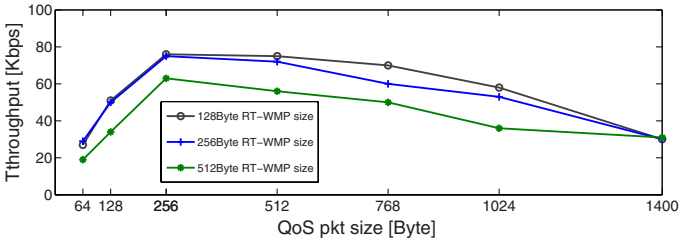


Fig. 6. QoS Cumulative Throughput vs. protocol packet size

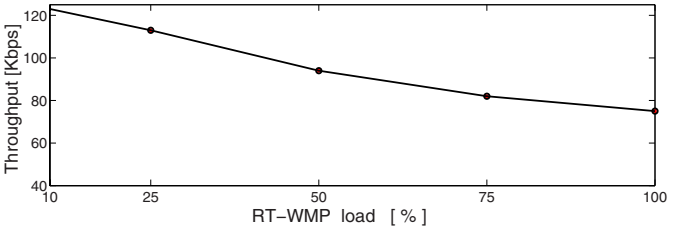


Fig. 7. QoS Cumulative Throughput vs. RT-WMP load percentile

7.2 RT-WMP Traffic Impact

QoS performance depends on real-time traffic. We want to consider the impact of real-time packet size and the total real-time traffic on QoS traffic. Fig. 6 shows the cumulative QoS throughput of 5 traffic flows (of 15 Kbps rate each one) in the presence of RT-WMP traffic generated to saturate the network resources versus the QoS packet size. The measurements were repeated for different real-time packet sizes, 128, 256 and 512 bytes respectively. The better throughput was obtained by a RT-WMP packet size of 128 bytes with QoS flow packet sizes of 256 bytes. The shape of the graphic is owing to the fact that small dimension packets increase the relative weight of authorization phases with negative consequences for efficiency. On the other hand, big packets (generated at the same rate) increase losses due to the being deadline.

Fig. 7 shows the effect of the RT-WMP load on the QoS traffic. 100% means that all RT-WMP loops have a real-time message to send in the MTP phase. Obviously, a lower RT-WMP load benefits the QoS throughput.

7.3 Fairness

For several reasons (partially connected topology, channel access method and hidden terminals), the contention among stations in an ad-hoc network is not homogeneous. Some nodes can suffer severe throughput degradation especially when the load is high. This is known as the *fairness problem* and IEEE 802.11 does not resolve it [17]. The RT-WMP QoS extension, working over 802.11,

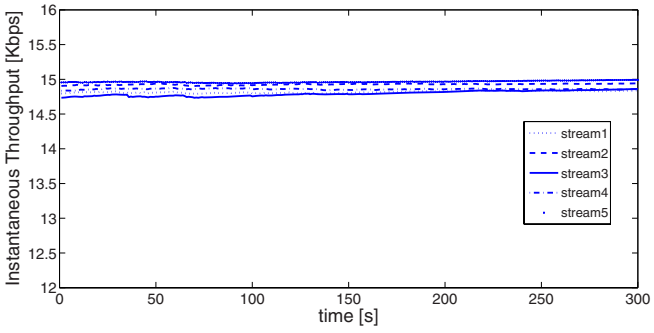


Fig. 8. Instantaneous Throughput of 5 flow of same class

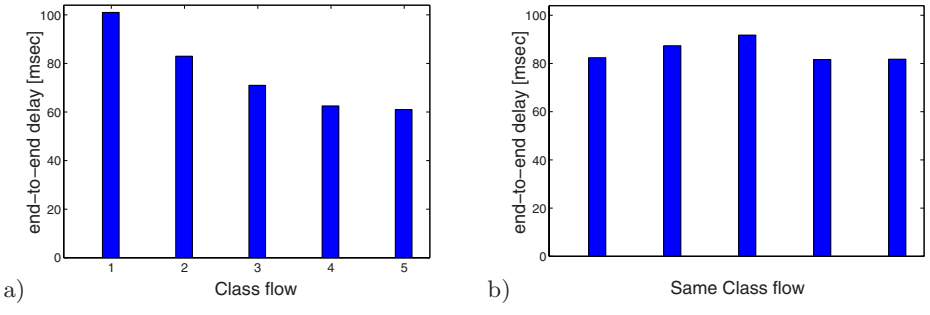


Fig. 9. End-to-end delay against class flow traffic

guarantees fairness amongst nodes. So we verified that sender nodes achieve very similar instantaneous throughput. In other words, throughput of same class flows shows very small deviations. This is showed in Fig. 8.

7.4 End-to-End Delay

Delay can degrade the audio and video applications. For such applications end-to-end delays have to be limited.
We have evaluated end-to-end from the point of view of flow class. Fig. 9 shows end-to-end delay values obtained over 5 QoS flows. Fig. 9.a compares delays of different class flows. As can be seen, delays turn out to be limited and, in the case of same class flows (Fig. 9.b), they suffer a very small variation. The system thus becomes more responsive to the high priority traffic granting similar delivery times to same class flows.

7.5 Multi-hop Transmission

As was stated in section 5.3, the source and destination may be several hops away and most protocols do not take into consideration the number of hops

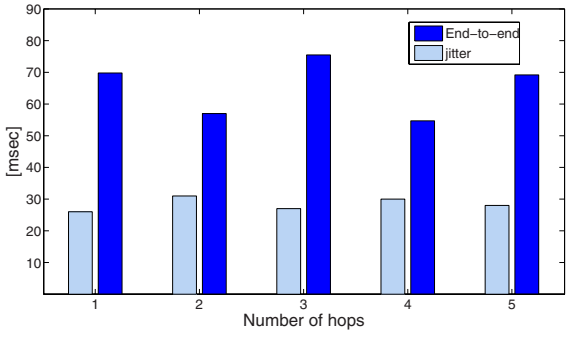


Fig. 10. Delay and jitter vs hop count

a packet has to cross. Since QoS priority policy takes into account the hops remaining before a packet's destination, we have evaluated the end-to-end delay and the jitter of 5 flows when each node generates traffic with the same rate and same class flow to the furthest node. So the *hop left* of each one is different. The results are showed in Fig. 10. Delays and jitters are small and the priority policy provides similar performances for the nearest and furthest nodes.

7.6 PDR Evaluation

Packet Delivery Ratio (PDR) is a measure of the percentage of packets that reach the destination within the specified deadline. The PDR is calculated as the ratio of the number of packets received within the deadline by the destination application layer, and the number of packets sent by the application layer at the source node. Fig. 11 shows PDR values for 5 flows of 15Kbps. In the first test (Fig. 11.a), we generated flows with different classes. We repeated the test with the same class flows(Fig. 11.b). The results show that the scheme gives acceptable PDR values and also show the effect of the priority policy.

7.7 Real Scenario Experiments

In our research we are interested in providing a reliable network for a hostile environment such as a tunnel. A team of four mobile robots (as shown in Fig. 12.a) equipped with microphone and speakers, were sent into the Somport tunnel (the railroad linking Canfranc, Spain with Pau, France (Fig. 12.b)). Robot activity was controlled by a laptop that worked as a base station. Several tasks were involved including telemanipulation, autonomous formation movement and maintaining inter-node connectivity and end-to-end voice communication between the laptop and the farthest mobile robot, simulating a rescue mission. In this scenario, the real-time multi-hop protocol RT-WMP supports the delay sensitive messages amongst nodes and the QoS extension is responsible for allowing end-to-end voice stream that allows the communication with victims. User voices were encoded in packets using the very efficient open source variable bit codec Speex [2], specifically designed for speech compression in VoIP applications.

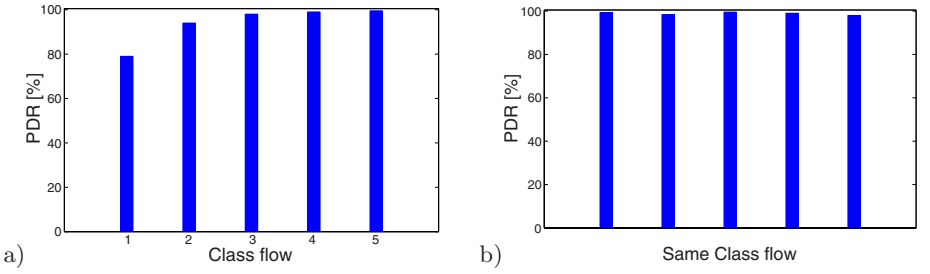


Fig. 11. Variation of PDR

Fig. 12. A robot used in real experiments in the Somport tunnel

8 Conclusions

In this paper we have proposed a way to incorporate multimedia flows in a real-time wireless communication network without jeopardizing the real-time traffic. This idea has been implemented and analyzed as an extension of the RT-WMP hard real-time protocol. This technique allows merging the real-time traffic with human communication such as video and voice over a MANET. The rationale is to take advantage of the bandwidth left free by the protocol when it is not working in the worst-case situation and use it to send QoS frames to allow audio and video streaming flows.

This QoS extension of the RT-WMP is perfectly integrated in the protocol and keeps real-time and QoS traffic separate and independent from each other. QoS messages are delivered following a priority policy based on flow class and laxity. The extension implements a flow admission control that estimates the available bandwidth using a distributed approach and allows or denies the entering of new flows into the system. The solution has been implemented over the RT-WMP and evaluated in a controlled environment, and the results show that it is a valid solution for adding QoS capabilities to real-time protocols. In fact, tests show that many audio and video flows can be supported simultaneously. Moreover, further tests have been performed in real applications involving cooperative multi-robot teams.

References

1. Andersen, S.V., Kleijn, W.B., Hagen, R., Linden, J., Murthi, M.N., Skoglund, J.: iLBC - a linear predictive coder with robustness to packet losses, pp. 23–25 (2002)
2. Valin, J.: The speex codec manual the speex project (2003), `http://www.speex.org`
3. Tardioli, D., Villarroel, J.L.: Real time communications over 802.11: RT-WMP. In: IEEE International Conference on Mobile Adhoc and Sensor Systems, MASS 2007, pp. 1–11 (2007)
4. Ergen, M., Lee, D., Sengupta, R., Varaiya, P.: Wireless token ring protocol-performance comparison with IEEE 802.11. In: Proceedings of Eighth IEEE International Symposium on Computers and Communication (ISCC 2003), vol. 2, pp. 710–715 (2003)
5. Wang, P., Zhuang, W.: A token-based scheduling scheme for WLANs and its performance analysis. In: IEEE International Conference on Communications, ICC 2007, pp. 3716–3721 (2007)
6. Zhang, J., Liu, K.H., Shen, X.: A novel overlay token ring protocol for inter-vehicle communication. In: IEEE International Conference on Communications, ICC 2008, pp. 4904–4909 (2008)
7. Ergen, M., Lee, D., Sengupta, R., Varaiya, P.: WTRP — Wireless token ring protocol. IEEE Transactions on Vehicular Technology 53(6), 1863–1881 (2004)
8. Zhai, H., Wang, J., Fang, Y.: Ducha: A new dual-channel MAC protocol for multihop ad hoc networks. IEEE Transactions on Wireless Communications 5(11), 3224–3233 (2006)
9. Taheri, S.A., Scaglione, A.: Token enabled multiple access (tema) for packet transmission in high bit rate wireless local area networks. In: IEEE International Conference on Communications, ICC 2002, vol. 3, pp. 1913–1917 (2002)
10. Donatiello, L., Furini, M.: Ad Hoc Networks: A Protocol for Supporting QoS Applications. In: IPDPS 2003: Proceedings of the 17th International Symposium on Parallel and Distributed Processing, Washington, DC, USA, p. 219.2. IEEE Computer Society, Los Alamitos (2003)
11. Liu, I.-S., Takawira, F., Xu, H.-J.: A hybrid token-CDMA MAC protocol for wireless ad hoc networks. IEEE Transactions on Mobile Computing 7(5), 557–569 (2008)
12. Hamidian, A., Krner, U.: Providing QoS in Ad Hoc Networks with Distributed Resource Reservation. In: 20th International Teletraffic Congress (ITC-20), Ottawa, Canada (2007)
13. Bheemarjuna Reddy, T., John, J.P., Siva Ram Murthy, C.: Providing MAC QoS for multimedia traffic in 802.11e based multi-hop ad hoc wireless networks. Comput. Netw. 51(1), 153–176 (2007)
14. Facchinetti, T., Buttazzo, G., Almeida, L.: Dynamic resource reservation and connectivity tracking to support real-time communication among mobile units. EURASIP J. Wirel. Commun. Netw. 5(5), 712–730 (2005)
15. Vlaovic, B., Brezocnik, Z.: Packet based telephony. In: International Conference on EUROCON 2001, Trends in Communications, vol. 1, pp. 210–213 (2001)
16. Rivas, M.A., Harbour, M.G.: Marte OS: An ada kernel for real-time embedded applications. In: Proceedings of the International Conference on Reliable Software Technologies, Ada-Europe-2001 (2001)
17. Wang, Y., Bensaou, B.: Achieving fairness in IEEE 802.11 DFWMAC with variable packet lengths. In: Global Telecommunications Conference, GLOBECOM 2001, vol. 6, pp. 3588–3593. IEEE, Los Alamitos (2001)