# Ad Hoc Quality of Service Multicast Routing with Objection Queries for Admission Control

Kaan Bür[1], Cem Ersoy

*NETLAB, Department of Computer Engineering*
*Bogaziçi University, 34342 Bebek, Istanbul, Turkey*
*{burk, ersoy}@boun.edu.tr*

---

**Abstract**

Today's handheld computing devices equipped with novel wireless network technologies can provide their users with features such as mobility, multimedia support and group communication. On the other hand, the administrative effort required to enable all these features increases beyond the level that an ordinary user can manage. Ad hoc networks, being able to quickly organise themselves without user intervention, can overcome this problem. They are also suitable for group-oriented mobile multimedia communication. However, it is imperative for ad hoc networks to combine quality of service (QoS) and multicast routing strategies in order to utilise the wireless medium efficiently. This article defines the components of an ad hoc QoS multicast routing (AQM) protocol which addresses this issue. AQM achieves multicast routing efficiency by tracking the availability of resources for each node within its neighbourhood. Computation of free bandwidth is based on reservations made for ongoing sessions and the usage reported by neighbours. Current QoS status is announced at the initiation of a new session and updated periodically in the network to the extent of QoS provision. Thus, nodes are prevented from applying for membership if there is no QoS path for the session. When nodes wish to join a session with certain service requirements, a three-phase process consisting of request, reply and reserve steps ensures that the QoS information is updated and used to select the most appropriate routes. The allowed maximum hop count of the session is taken into account in order to satisfy the delay requirements of the multimedia applications. To cope with the continuous nature of streaming multimedia, AQM nodes check the availability of bandwidth within their neighbourhood not only for themselves but within a virtual tunnel of nodes. Objection queries are issued prior to reservation to avoid excessive resource usage due to allocations made by nodes which cannot detect each other directly. New performance metrics are introduced to evaluate the efficiency of AQM regarding the satisfaction level of individual members as well as the success rate of sessions. Simulation results show that, by applying novel QoS management techniques, AQM significantly improves multicast efficiency for members as well as for sessions.

*Keywords:* Ad hoc networks, mobile multimedia, multicast routing, quality of service, wireless communication.

---

## 1. Introduction

THE evolution of wireless communication technologies has reached such a point that it becomes popular and easy to integrate these technologies to handheld computing devices, which have initially been developed for personal use only. Today, a new generation of portable computers is being developed, offering users more computational power than ever, in addition to mobility. However, as these devices reach the mass market, they confront users with the heavy task of learning how to deal with such complicated items. In fact, the integration of new technologies into these devices requires configuration like computers, which is a challenge for the inexperienced user. Thus, it becomes an increasingly important feature that, once a mobile device is operational, it is able to configure itself with all its personal and networking capabilities, asking its users only for their personal preferences. Only by making the administrative work transparent to the end user can wireless technologies contribute to the penetration of mobile computing devices.

The widespread use of mobile and handheld devices is popularising ad hoc networks, which are self-organising communication groups formed by wireless mobile hosts. They make their administrative decisions in a distributed manner without any centralised control. They are free from the boundaries of any pre-existing infrastructure. They can be deployed anytime, anywhere [1]. They are considered for many applications, including group-oriented computing sessions characterised by close collaborative efforts, such as disaster relief, community events and game playing.

In order to meet the qualitative expectations of mobile users for such applications, ad hoc networks need support for multimedia, which makes quality of service (QoS) a fundamental requirement. QoS support is closely related to resource allocation, the objective of which is to decide how to reserve resources such that QoS requirements of all the applications can be satisfied [2]. Due to the dynamic nature of ad hoc networks characterised by variable link behaviour, node movements and topology changes, it is very important to design efficient methods of conserving the scarce resources [3]. Multicast routing is a promising technique to provide a subset of network nodes with the group-oriented service they demand while not jeopardizing the bandwidth requirements of others. The advantage of

---

Manuscript submitted to European Transactions on Telecommunications on August 30, 2004; revised version submitted on January 18, 2005.

This work was supported in part by the State Planning Organisation, Turkey, under Grant No. DPT98K120890 and the university research program of OPNET Technologies, Inc, USA.

[1] Corresponding author. *Tel:* +90-212-359-7170. *Fax:* +90-216-287-2461.

multicast routing is that packets are only multiplexed when it is necessary to reach two or more receivers on disjoint paths. Thus, multicast routing can improve wireless link efficiency by exploiting the inherent broadcast property of the wireless medium. Combining the features of ad hoc networks with the usefulness of multicast routing, a number of group-oriented applications can be realised [4].

Current research on the implementation of QoS to ad hoc networks is mainly limited to medium access control (MAC) and routing. Generally, a time division multiple access (TDMA) or a clustered code division multiple access (CDMA) over TDMA network synchronised on a frame and slot basis is assumed, where topologies do not change very fast and slot assignment is left to the underlying MAC layer [5, 6, 7, 8]. There is a control phase in each frame, whereby nodes exchange connectivity information while clusterheads synchronise slots and frames, in addition to assigning slots and code to connection requests. Each node broadcasts its QoS information during the control phase, at the end of which each node knows the channel reservation status of the next information phase [5]. A set of free and non-conflicting slots is calculated on three adjacent links and propagated towards the destination [6]. Bandwidth calculation is done end-to-end; only destinations reply to connection requests. Based on the free slot information in the path-searching packets they receive, destinations can either select single paths, or determine a multi-path route to satisfy their QoS requirements [7]. Imprecise QoS information is kept at each node for every other, whereas the state of immediate neighbours is traced more accurately [8]. Additional MAC assumptions include a mechanism of beacons, contention resolution, and local message broadcasting.

A QoS system consists of several components, including service differentiation, admission control, and resource allocation [9, 10]. Service differentiation schemes use QoS techniques such as priority assignment and fair scheduling. Priority-based mechanisms change the waiting times of the frames and assign smaller values to high-priority traffic. Fair scheduling algorithms partition resources among flows in proportion to a given weight and regulate the waiting times for fairness among traffic classes [9]. Since the exact condition of the wireless network is not known, on the other hand, an accurate decision is not possible for the admission of a new flow. Measurement-based admission control mechanisms are based on observations on the existing network status, whereas calculation-based mechanisms make use of performance metrics they define for evaluating the status of the network. Without admission control and bandwidth reservation, the provision of QoS only by differentiating flows and coordinating channel access order is not effective for high traffic loads [10]. A contention-aware admission control protocol (CACP) introduces the concept of an extended contention area covering the carrier sensing range of a node [11]. Admission decisions are based on the information collected from the neighbours in the contention area, which consists of the smallest local bandwidth available and the consumed bandwidth within that area. None of the nodes intentionally breaks QoS by admitting too many flows.

Another important feature of a QoS system is the congestion control scheme. Congestion occurs when the data sent exceeds the capacity of the network and causes excessive delay and loss. It can be avoided by predicting it and reducing the transmission rate. If congestion is local, it can also be handled by routing around the congested node without reducing the data rate [12]. A multicast congestion control scheme for multi-layer data traffic is proposed to be applied at the bottlenecks of the multicast tree using the queue states [13]. Some flow information is maintained at each node, and data layers are blocked and released to solve congestion and adjust the bandwidth rate.

It is not an easy task to incorporate QoS to ad hoc multicast routing. Incremental changes on existing schemes cannot efficiently address the critical issues mentioned above. This article presents the ad hoc QoS multicast routing (AQM) protocol as a composite solution to the problem. In this protocol, QoS availability is tracked within each node’s neighbourhood based on current resource reservations, and announced at session initiation. When a node wants to join a session with certain service requirements, a request-reply-reserve process is initiated to update this QoS information and select one of the routes which can meet the requirements of that session. Objection queries are utilised during this process to avoid excessive allocation of resources. Simulations show that AQM significantly improves multicast efficiency for members and sessions through QoS management.

This article is organised as follows. Previous research related to ad hoc multicast routing is summarised in Chapter 2. AQM is presented in Chapter 3. New metrics for the evaluation of the proposed QoS protocol are defined in Chapter 4. The performance of the system is interpreted in Chapter 5. Concluding remarks and a few suggestions on future research directions are made in Chapter 6.

## 2. A Brief History of Ad Hoc Multicasting

There are various protocols developed to build and maintain a multicast graph and perform routing in ad hoc networks, some of which are summarised below. However, they do not attempt to cover the QoS aspect of ad hoc multicast communication, which is becoming increasingly important as the demand for mobile multimedia increases.

Multicast ad hoc on demand distance vector (MAODV) routing protocol is derived from AODV [14, 15]. The multicast group leader maintains a group sequence number and broadcasts it periodically to keep the routing information fresh. A node wishing to join a multicast group generates a route request. If the multicast group leader is in the request table, the request is unicast to it. Otherwise, the request is broadcast. Only the leader or members of the multicast group may respond to a join request by unicasting a route reply back to the requester. Nodes receiving join requests update their route and multicast tables with the downstream next hop information. Nodes receiving reply messages update their tables with the upstream next hop information. They increment hop counts and forward the reply to the requester, which selects the best from several replies in terms of highest sequence numbers and lowest hop count, and enables that route by unicasting a multicast activation message to its next hop. Intermediate nodes receiving the activation message enable their multicast table entries for the requester. If they are already multicast group members, further propagation of the message is not necessary. Otherwise, they unicast it upstream along the best route according to the replies they received previously. Nodes wishing to leave a group unicast a multicast activation message to their next hop with its prune flag set.

The core-assisted mesh protocol (CAMP) uses cores within a group to limit the control traffic caused by join requests, whereas anchors are supposed to rebroadcast data packets they receive to feed downstream routers [16, 17]. Each node maintains a set of tables for routing, core-to-group mapping, as well as anchor and multicast group management. When a node updates its anchor or multicast table, it sends a reporting message to all its neighbours. The basic join mechanism is initiated by a host asking its router to join a group. The router directly announces its membership if there are any data-forwarding members of that group among its neighbours. Otherwise, it broadcasts a join request. Member routers of the intended group send acknowledgements. The requesting router and its relays become part of the group as soon as they receive the first acknowledgement. A router leaves a multicast group if it has no member hosts and is not required as an anchor.

Bandwidth-efficient multicast routing (BEMR) finds the nearest forwarding multicast member for newly joining nodes [18]. When a new node broadcasts a join request, each node receiving the request adds its ID and increments the hop count before flooding it back to the network. Forwarding nodes receive some of these requests, choose the best alternative and send a reply along the selected path. The requester receives multiple replies and sends a reserve packet along the path with the best hop alternative. All nodes on this path become forwarding nodes. Routes are later optimised by removing unnecessary forwarding nodes.

The on-demand multicast routing protocol (ODMRP) introduces the concept of a forwarding group [19, 20]. Sources periodically broadcast join query messages to invite new members and refresh existing membership information. When a node receives a join query, it stores the upstream node address in its routing table. If the maximum hop count is not exceeded, it updates the join request using this table and rebroadcasts the packet. When a node decides to join a session, it broadcasts a join reply. When a node receives a join reply, it checks the table of next nodes to see if it is on the path to the source. If this is the case, it sets its forwarding group flag and broadcasts its own join reply after updating the table of next nodes. Periodic join requests initiated by the source must be answered by session members with join replies to remain in the group. Forwarding group nodes reset their flags if they do not receive any replies periodically.

Neighbour-supporting multicast protocol (NSMP) utilises node locality to reduce route maintenance overhead [21]. A mesh is created by a new source, which broadcasts a flooding request. Intermediate nodes cache the upstream node information contained in the request, and forward the packet after updating this field. When the request arrives at receivers, they send replies to their upstream nodes. On the return path, intermediate nodes make an entry to their routing tables and forward the reply upstream towards the source. In order to maintain the connectivity of the mesh, the source employs local route discoveries by periodically sending local requests, which are only relayed to mesh nodes and their immediate neighbours to limit flooding while keeping the most useful nodes informed. Replies are sent back to the source to repair broken links. Nodes more than two hops away from the source cannot join the mesh with local requests. They have to flood member requests.

Associativity-based ad hoc multicast (ABAM) builds a source-based multicast tree [22]. Association stability, which is achieved when the number of beacons received consecutively from a neighbour reaches a threshold, helps the source select routes which will probably last longer and need fewer reconfigurations. The tree formation is initiated by the source, whereby it specifically identifies its receivers. Valid receivers, which already know possible routes to the source, run a route selection algorithm to select and reply with routes of highest association stability. Upon receiving the replies, the source runs a tree selection algorithm to find common links, builds the shared-link multicast tree, and sends a setup message to its receivers. Tree reconfigurations occur when the associative property is violated. To join a multicast tree, a node broadcasts a request, collects replies from group members, selects the best route, and sends a confirmation. To leave a multicast tree, a notification is propagated upstream along the tree until a branching or receiving node is reached.

Differential destination multicast (DDM) lets source nodes manage group membership as admission controllers, and stores multicast forwarding state information encoded in headers of data packets to achieve stateless multicasting [23, 24]. Join messages are unicast to the source, which tests admission requirements, adds the requester to its member list, and acknowledges it as a receiver. The source needs to refresh its member list in order to purge stale members. It sets a poll flag in data packets and forces its active receivers to resend join messages. Leave messages are also unicast to the source, which removes the leaving member from its list. Forwarding computation is based on destinations encoded in the headers. During this process, a node has to check the header for any DDM block or poll flag intended for it and take the appropriate actions.

Independent-tree ad hoc multicast routing (ITAMAR) provides several heuristics to compute a set of independent multicast trees, such that a tree is used until it fails and then replaced by one of its alternatives [25]. Maximally independent trees are computed by minimising the number of common edges and nodes under the assumption that node movements are independent of each other. Some overlapping is allowed since totally independent trees might be less efficient and contain more links. Thus, the correlation between the failure times of the trees is minimal, which leads to improved mean times between route discoveries. New trees are computed when the probability of failure for the current set of trees rises above a threshold. Given a mobility pattern, it is important to estimate the time this happens. Instead of replacing a tree even if one link fails, an independent path algorithm finds a set of backup paths to replace the damaged part of the tree.

Lantern-tree-based QoS multicast (LTM) is a bandwidth routing protocol with an improved success rate by means of multipath routing [26, 27]. A lantern is defined as one or more subpaths with a total bandwidth between a pair of two-hop neighbouring nodes, whereas a lantern path is a path with one or more lanterns between a source and a destination. A lantern tree serves as the multicast tree with its path replaced by the lantern-path. The scheme provides a single path if bandwidth is sufficient or a lantern-path if it is not. The replying paths from the destination back to the source are merged together to construct the lantern tree.

Probabilistic predictive multicast algorithm (PPMA) tracks relative node movements and statistically estimates future relative positions to maximise the multicast tree lifetime by exploiting more stable links [28]. Thus, it tries to keep track of the network state evolution. It defines a probabilistic link cost as a function of energy, distance and node lifetime. The scheme tries to keep all the nodes alive as long as possible. It models the residual energy available for communication for each node, which is proportional to the probability of being chosen to a multicast tree. Nodes of low energy cannot join any more multicast trees.

## 3. The AQM Protocol

The motivation behind QoS support for multicast routing in ad hoc networks is the fact that mobile multimedia applications are becoming increasingly important for group communication. For an efficient ad hoc QoS multicast routing strategy, implementation of QoS classes, negotiations between the network and its users, bounded loss and delay, bandwidth reservation, and mobility management are very important. In the following sections, the structural components of AQM are defined, which address these issues. Design details include the usage of QoS classes, the management of sessions and members, resource allocation, neighbourhood maintenance, and dealing with mobility.

### *3.1. Usage of QoS Classes*

Different QoS classes are necessary to support various types of applications in an efficient manner. In any multimedia network, there may be multiple application types being run simultaneously, which need to be classified in terms of their varying QoS requirements. To represent such a generic networking environment in this work, a sample set of multimedia applications are suggested. Depending on the user profiles, network conditions and computational capabilities of the mobile multimedia devices, other applications with different QoS settings can easily be added to the set.

Defining QoS classes also limits the amount of information to be transmitted between network nodes. It is otherwise impossible to define and forward a best QoS combination without making some assumptions or disregarding some valuable data about the current QoS conditions being experienced by the network. Therefore, it may be preferable that nodes only inform others on the availability of a certain QoS support level and send updates only when this level changes.

### *3.2. Session Initiation and Termination*

A session is started by a session initiator (MCN_INIT), which can be any node that broadcasts a session initiation packet (SES_INIT) consisting of the identity number and the QoS class of the new session, which sets the bandwidth and hop count rules to join it. If necessary, the session definition can be extended with the duration and the cost of the application, the minimum number of users to activate the session, and the maximum number of acceptable users. A table of active sessions (TBL_SESSION) is maintained at each node to keep the information on session definitions. Figure 1 shows the phases of session initiation.

Using their session tables, nodes forward initiation packets of previously unknown sessions. A membership table (TBL_MEMBER) is used to denote the status of the predecessors (MCN_PRED) which have informed the node on the existence of a particular multicast session, and the QoS support status of the path from the session initiator up to this node via that predecessor. The hop count information in the packet is used to prevent loops in the forwarding process. The session initiation packet is forwarded as long as the QoS requirements are met. Before the packet is rebroadcast, each node updates its QoS information fields with the current QoS conditions experienced by that node. The packet is dropped if QoS requirements cannot be met any more, avoiding flooding the network unnecessarily.

The session information is refreshed periodically via session update packets (SES_UPDATE) sent by the session initiator. Similar to the session initiation packets, they are propagated throughout the network as long as the QoS requirements of the session can be fulfilled. Unlike the session initiation packets, however, each new update packet is forwarded once even if it belongs to a previously known session and come from a known predecessor. This ensures that all new nodes in a neighbourhood are informed on the existence of the ongoing sessions they can join.

The session is closed by its initiator with a session termination message (SES_TERMINATE). Upon receiving it, all nodes knowing that session clean their tables, whereas nodes forwarding multicast data (MCN_FWD) also free their resources allocated to it. A node receiving a session termination packet forwards it if it has also forwarded the corresponding initiation packet or is currently forwarding session data to at least one active session member. Thus, receivers of a closed session are forced to leave the session.

### *3.3. Membership Management*

Nodes can only join sessions known to them. A node can directly join a session if it is already a forwarding node in that session. Otherwise, it issues a join request. When a node broadcasts a join request packet (JOIN_REQ) for a session, only upstream neighbours which are aware of the session take the request into consideration. These are predecessors of the requester and propagate the packet upstream as long as QoS can be satisfied. The upstream flow of the request is guaranteed by comparing the hop count information of the packet with the distance to the server of the related session at intermediate nodes.

Ad hoc networks are highly dynamic, and available resources may change considerably after the arrival of the QoS conditions with the first session initiation packet. As explained in the following sections, greeting messages are exchanged between neighbours to update nodes on the bandwidth usage within a neighbourhood. However, nodes do not send session status update messages to avoid excessive control traffic. Instead, QoS is announced once by the session initiation packet and is updated only on demand. Intermediate nodes maintain a temporary request table (TBL_REQUEST) to keep track of the requests and replies they have forwarded and prevent false or duplicate packet processing.

A forwarded request eventually reaches some nodes which are already members of that session and therefore can directly send a reply (JOIN_REP) back to the requester. Members of a session are the initiator, the forwarders, and the receivers. Before replying, however, members send an objection query (JOIN_OBJ) to their immediate neighbours to check if a possible new resource allocation violates the bandwidth limitations of these. Such a query is necessary since it is otherwise impossible for a node to see the bandwidth usage beyond its direct neighbours. This concept is explained in more detail in the next section.

Downstream nodes, having forwarded join requests and waiting, aggregate the replies they receive at the end of a predefined time period, select the one offering the best QoS conditions, combine it with the QoS they can currently offer, and send it towards the requester. During this process, they exploit the objection query mechanism as well since it is possible that they qualify as forwarders. The information on the originator and the immediate forwarder of the reply is kept in the packet. The originator of the join request selects the one with the best QoS conditions among possibly several replies it receives. It changes its status from predecessor to receiver (MCN_RCV) and sends a reserve message (JOIN_RES) to the selected node which has forwarded the reply.

Upon receiving the reserve packet, intermediate nodes check whether they are among the intended forwarders on the path from the selected replier towards the requester. If this is the case, they change their status from predecessor to forwarder, reserve resources, and update their membership tables to keep a list of successors for that session. They forward the message upstream. Eventually, the reserve message reaches the originator of the reply, which can be the session initiator with some or without any members, a forwarder with one or more successors, or a receiver. If the

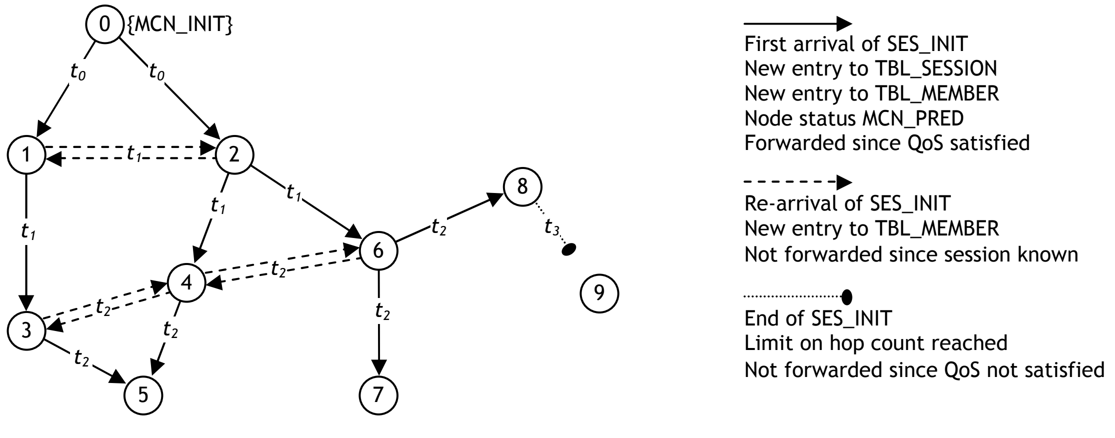


Fig. 1: The AQM session initiation process: SES_INIT is broadcast by MCN_INIT $n_0$ for a new session. It propagates through the network with time, informing all the nodes from $n_1$ to $n_8$, which update their TBL_SESSION and TBL_MEMBER. $n_9$ is not informed since it is beyond the QoS limits in terms of hop count. $t_i < t_{i+1}$, $0 \le i \le 3$, represent the relative timing of the messages. Upstream re-arrival arrows are not shown to keep the figure simple.

replier is the session initiator and this is its first member, it changes its status from initiator to server (MCN_SRV). If it is a receiver, it becomes a forwarder. In both cases, the replier records its successor in its member table and reserves resources to start sending multicast data. If the node is an active server or forwarder, it must have already reserved resources. It only adds the new member to its member table and continues sending the regular multicast data. Figure 2 shows the phases of joining a session.

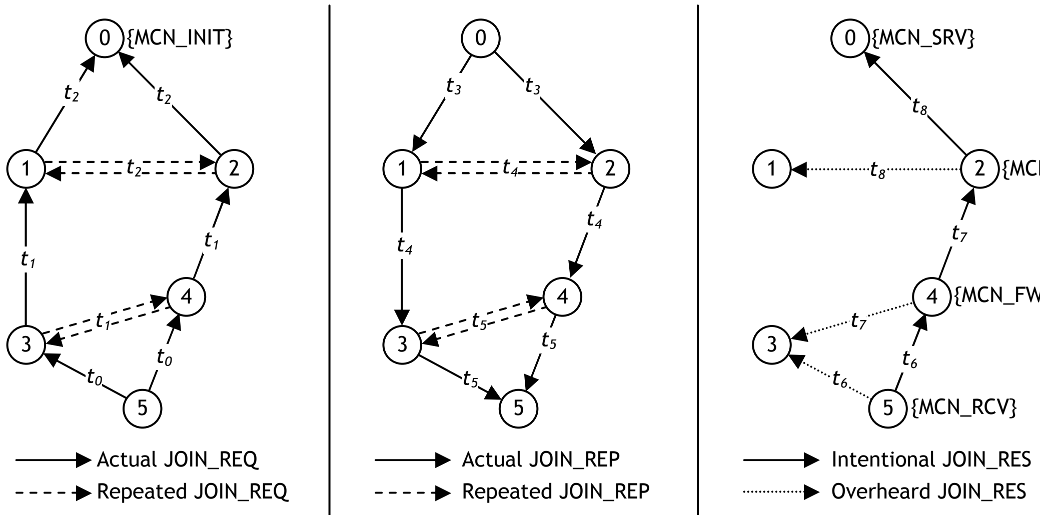
Each time a request-reply-reserve process completes successfully, having also passed the objection query phase, intermediate nodes gather enough routing and membership data to take part in the packet forwarding task. When a host sends multicast packets with a particular multicast session ID, its neighbours already know if they are involved in the session by checking their tables, one with information on their own membership status, and another with a list of multicast sessions they are responsible of forwarding. Nodes also make use of the replies they receive during a session join process. If the reply is sent by a previously unknown node in response to a request it has forwarded for a session, the intermediate node enters that predecessor into its member table for future routing operations.

A node needs to inform its forwarder on the multicast graph upon leaving a session. After receiving a quit notification (SES_LEAVE), the forwarding node deletes the leaving member from its member table. If this has been its only successor in that session, the forwarding node checks its own status regarding the session. If the forwarder itself is also a receiver, it updates its status. Otherwise, it frees resources and notifies its predecessor of its own leave.

### *3.4. Allocation of Resources*

The nodes in an ad hoc network have to maintain their resource information with as much accuracy as possible to support QoS, which includes the ability to keep track of available bandwidth within their neighbourhood. This is how they are able to provide their neighbours with valid routes when asked to take part in a request-reply-reserve process of a node wishing to join a multicast session. In AQM, nodes behave proactively with regard to management of multicast session information by maintaining routing tables. They keep themselves and their neighbours aware of the changes in the QoS conditions and node connectivity regarding the multicast sessions known to them. The rationale behind this method is that QoS management in a highly dynamic environment such as wireless mobile networks cannot be achieved satisfactorily without informing the network on these issues in advance. However, the nature of a join process is on-demand, and AQM also checks the most up-to-date QoS conditions during this process and thus, it presents a hybrid approach.

During a request-reply-reserve process of a join request, the QoS conditions are checked at each intermediate node to make sure that current resource availability allows the acceptance of a new session. There are two important issues regarding the bandwidth allocation capability of a node that is about to decide whether or not to forward a reply it has just received towards the originator of the corresponding join request. First, the node has to check if, once selected by the requester as a forwarder for the multicast session, it can afford the bandwidth needed to support the streaming of the multimedia data of a certain QoS class. Second, it has to ensure that, by allocating



(a) Request phase (b) Reply phase (c) Reserve phase

Fig. 2. The AQM session joining process: (a) JOIN_REQ is issued by $n_5$. It propagates towards any member of the session as long as QoS can be satisfied. Nodes from $n_1$ to $n_4$ update their TBL_REQUEST as they forward the packet since they are not session members. (b) JOIN_REP is sent back from $n_0$ to $n_5$. It is forwarded by $n_1$, $n_2$, $n_3$, $n_4$. (c) $n_5$ sends JOIN_RES along the selected QoS path via $n_4$, $n_2$, $n_0$, which reserve resources and update their status. Other nodes ignore the message. $t_i < t_{i+1}$, $0 \le i \le 8$, represent the relative timing of the messages. Objection query messages are not shown in (b) to keep the figure simple.

bandwidth to a new request, it does not cause one of its neighbours suffer from overload as a result of excessive bandwidth usage in the neighbourhood of the latter.

The continuous nature of multimedia applications requires a new method of checking bandwidth availability. Concerning a session server about to allocate resources for its first member, twice as much bandwidth has to be available in the neighbourhood than the amount required by the QoS class of the session. The reason for this is that the forwarding node immediately following the server on the path to the member belongs to the same neighbourhood as the server. In other words, being within the transmission range of each other, they share the bandwidth of the same neighbourhood. Therefore, a session server has to ensure that its successor also has enough bandwidth available to forward multicast data packets that it receives. Following the path downstream towards the new member, an intermediate node about to take part in the packet forwarding process for the first time has to check for availability of three times the QoS bandwidth needed by the session, since it shares the bandwidth of one neighbourhood with the nodes immediately preceding and succeeding it. Once the multicast session starts, it receives packets from its predecessor, rebroadcasts them, and allows its successor to forward the packets further downstream. Thus, nodes have to check for availability of necessary bandwidth according to their position within the multicast tree before accepting a request. For a member already forwarding packets of that session, this requirement is met automatically since the node has already been through this allocation process. Figure 3 shows the virtual tunnel of nodes where to check the bandwidth availability in a pipelined fashion.

As mentioned previously, a node decides whether or not to take part in a session as a forwarder based on its current resource availability. While this approach prevents the node from overloading itself, it is not enough to help other nodes balance their loads. Although a node does not allocate more bandwidth than available in its neighbourhood, the overload problem arises as a result of the allocations made by its neighbours which cannot directly detect each other. In other words, a node can be surrounded by several neighbours, some of which are not within the transmission range of each other. In this case, the node experiences overload due to excessive resource usage in the neighbourhood, which cannot be foreseen since the surrounding nodes are not aware of each other's reservations. Thus, a particular kind of hidden terminal problem prevents nodes from making more accurate reservation decisions. To overcome the problem, an extension to the request-reply-reserve process is necessary, whereby each replying node first consults its neighbours to see if any of them becomes overloaded.

A node about to forward a reply first sends an objection query to its immediate neighbours. This one-hop message containing information on the requested bandwidth warns the neighbours to check whether they become overloaded as a result of this potential allocation. If the new reservation causes the limit to be exceeded, the neighbour sends the objection back to the node which has queried it. Otherwise the query is discarded. If the node having sent the query receives any objection, it cancels the forwarding of the reply. Otherwise the query times out, indicating that the reply can be safely sent. Only those neighbours who are serving one or more sessions may object to new allocations. It is not important that a silent node becomes overloaded.

A session initiator, which is about to get its first member, has to issue an objection query before replying. An intermediate node about to forward a reply towards its requester as a predecessor also has to send such a query. An

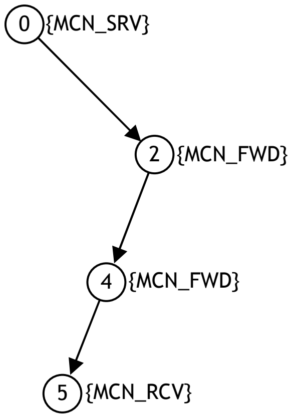


Fig. 3. The virtual tunnel approach to bandwidth availability. $n_0$ checks for two times QoS bandwidth since it has to make sure that $n_2$ can also forward packets. $n_2$ checks for three times QoS bandwidth since, in addition to its predecessor $n_0$ and itself, it has to make sure that $n_4$ can also forward the streaming data. Finally, $n_4$ checks again for two times QoS bandwidth since $n_5$ is only a receiver which does not send packets.

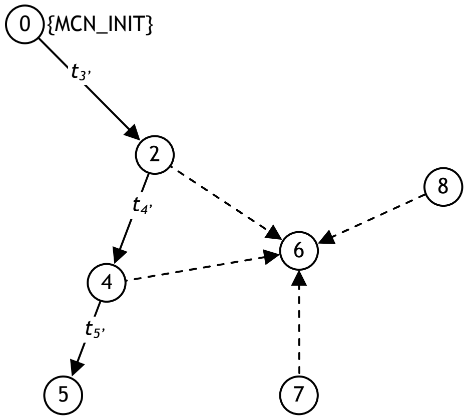


Fig. 4. The objection query mechanism. Revisiting the reply phase of a join process, the nodes $n_0$, $n_2$, $n_4$ send one-hop objection queries before sending their replies. This is necessary in order to allow $n_6$ to object to a possible allocation made by $n_2$ or $n_4$, if it starts suffering from overload. $n_6$ is already sharing its neighbourhood bandwidth with $n_7$ and $n_8$. However, $n_2$ and $n_4$ are not aware of this since they cannot directly detect the others.

active member forwarding packets of a session does not need to query objections for each new join request since it has previously consulted its neighbours. Figure 4 shows a situation where the objection query mechanism is utilised.

### *3.5. Neighbourhood Maintenance*

Each node periodically broadcasts greeting messages (NBR_HELLO), informing its neighbours on its existence as well as its bandwidth usage, which is determined by the QoS classes of the sessions being served or forwarded by that node. Greeting messages can be piggybacked to other control and data messages to reduce control overhead. In other words, nodes do not need to send greeting messages explicitly unless they have not sent any piggybacked greeting messages for a certain period of time. Each node aggregates the information it receives with these messages in its neighbourhood table (TBL_NEIGHBOUR). This table is used to calculate the total bandwidth currently allocated to multicast sessions in the neighbourhood, which is the sum of all used capacities of the neighbouring nodes for that time frame. Neighbourhood tables also help nodes with their decisions on packet forwarding. Session initiation packets are forwarded only if a node has neighbours other than its predecessors for that session.

Due to the broadcasting nature of the wireless medium, residual capacities are node-based, i.e., a node's available bandwidth is the residual capacity in its neighbourhood. In an ideal model, it is assumed that the bandwidth of a link can be determined on its neighbouring links [8]. Thus, a node can only use the remaining capacity not used by itself and its immediate neighbours.

### *3.6. Adaptation to Mobility*

One of the major concerns for ad hoc communications is the ability of the routing infrastructure to cope with the dynamics of node mobility. In order to maintain connectivity and support QoS with maximum possible accuracy and minimum overhead under mobility conditions within their neighbourhood, nodes perform periodic cleanup operations on their session, membership and neighbourhood tables. If a node does not receive any greeting messages from a neighbour for a while, it considers that neighbour lost. Lost neighbours are marked as such for a predefined short period of time, at the end of which they are deleted from the neighbourhood table if they do not reappear. To prevent unnecessary message exchanges, nodes need to detect new neighbours quickly and distinguish them from lost neighbours reappearing after a short period of time and do not necessitate any update. If the lost neighbour is related to a session, it is also removed from the session, membership and request tables. This is an essential operation to keep the nodes up-to-date regarding the sessions and ready for future membership management activities such as initiating a new join request or replying to other nodes' join requests.

Additional action can be necessary depending on the status of the lost neighbour as well as that of the node itself. When an active session member, e.g., a forwarder or a receiver, loses its preceding forwarder or server, this means that it loses its connection to the session. It changes its own status to a predecessor, i.e., a regular node which is aware but not an active member of the session. It also informs its successors with a lost session message (SES_LOST) if it is a forwarding member of the session. When, on the other hand, a server or a forwarder loses a receiver, it updates its status depending on the existence of other receivers in that session. It does not need to inform other nodes. When a node loses its only predecessor for a specific session due to changes in network topology, it notifies its successors of the lost session and lets them know that it should be deleted from the list of predecessors for that session. Downstream nodes receiving the lost session messages interpret them in a similar way to update their status regarding the lost session and forward the message if necessary. This mechanism, combined with the periodic updates mentioned previously, keeps nodes up-to-date regarding the QoS status of the sessions and prevents them from making infeasible join requests in terms of resource allocation.

## 4. Performance Metrics

Distributed, loop-free, on-demand operation is a very important qualitative property for ad hoc networks [29]. However, the evaluation of ad hoc routing protocols also necessitates quantitative metrics, which can be measured to give a notion about their internal efficiency. The goodput, i.e., the packet loss rate defined by the ratio of the number of packets received to the number of packets transmitted, the average end-to-end delay and throughput as well as the control overhead are widely used to evaluate ad hoc routing protocols. These are also adopted by ad hoc QoS routing protocols [5, 6], in addition to QoS-oriented metrics such as the success ratio defined by the number of accepted connections divided by the number of connection requests, the average path cost [7, 8], and the incompleteness ratio defined by the number of broken connections divided by the number of successful QoS requests [5, 7].

The efficiency of ad hoc multicast routing protocols is further measured by their data packet delivery ratio, data forwarding, i.e., packet replication efficiency defined by the number of data packets transmitted per original data packet delivered, and control overhead [14, 18, 20, 21, 23, 24]. Other metrics used are the average delay, the percentage of packet loss, the number of control packets received by each node [16, 17], multicast tree lifetime [25], and energy consumption [28]. Finally, the multicast session success rate is defined by the number of accepted receivers divided by the number of requests to join a session [26, 27].

On the other hand, the evaluation of QoS multicast routing performance in ad hoc networks requires criteria that are both qualitative and measurable. The main concern of this article is to test the efficiency of AQM in providing multicast users with QoS and satisfying the service requirements of multimedia applications. Therefore, it is necessary to focus on user satisfaction both at member and session levels. Thus, two new performance metrics are introduced in the following sections in addition to two of the conventional metrics used in previous research.

### *4.1. Metrics of Member Satisfaction*

The success rate of accepting new members to multicast sessions is an important criterion for the performance of a multicast routing protocol. It is a good measure of member satisfaction what percentage of the users who wish to join a multicast session can be admitted to the requested multimedia application. Thus, the member acceptance ratio $A_{Member}$ is formulated as follows:

$$A_{Member} = \frac{r}{q} \tag{1}$$

where $r$ represents the number of receivers, and $q$ is the total number of join requests issued by all ad hoc nodes. Their ratio reflects the success rate of AQM in accepting a node's request to join a session.

An important aspect of the QoS-related multicast routing decisions made by AQM is the improvement in the ratio of overloaded member nodes, which has a direct impact on the satisfaction of session members regarding the multicast service provided. It is one of AQM's main concerns that network resources are not excessively utilised to avoid possible collisions and packet loss due to overload and keep the QoS conditions at a satisfactory level. The member overload ratio $O_{Member}$ is formulated as follows:

$$O_{Member} = \frac{o}{s+f} \tag{2}$$

where $o$ represents the number of overloaded nodes, which have decided to serve and forward more sessions than is possible without exceeding the maximum available bandwidth, $s$ is the total number of session servers, and $f$ is the total number of session forwarders. The division gives the ratio of overloaded nodes to all serving and forwarding nodes and represents a member-level rate of overload.

It is inevitable that the computational overhead of a routing protocol increases with its complexity. However, it is possible to keep this overhead at an acceptable level while adding QoS functionality to a protocol, especially in order to deal with the effects of mobility, the changes in topology, and the issues of scalability. Thus, the member control overhead $C_{Member}$ is formulated as follows:

$$C_{Member} = \frac{p}{s+f+r} \tag{3}$$

where $p$ represents the total number of packets received and processed by the nodes of the ad hoc network. The sum of $s$, $f$ and $r$ gives the total number of active nodes in the network, participating in at least one multicast session as a server, a forwarder, or a receiver. Thus, the division gives the number of control packets per multicast member to manage and maintain the AQM system.

### *4.2. Metrics of Session Satisfaction*

Excessive bandwidth occupation by single nodes during the course of a session has also an effect on other members of that session. Therefore, it is necessary to observe the implications of these events on sessions as well. The session overload ratio $O_{Session}$ is defined to evaluate the session-level success ratio of AQM to prevent overload, and formulated as follows:

$$O_{Session} = \frac{l}{m} \tag{4}$$

where $l$ is the number of sessions with at least one overloaded member, and $m$ is the total number of sessions. The term gives the percentage of sessions with one or more overloaded members, which can be interpreted as a session-level overload rate experienced by the ad hoc network.

## 5. Computational Experiments

Four important criteria are introduced in the previous chapter in order to evaluate the performance of AQM. Two of these, namely the member acceptance ratio (1) and the control overhead (3), are inspired by previous research. Since a QoS-based scheme needs additional criteria to show its own quality, however, two new metrics are defined. These are the member and session overload ratios (2, 4). In the following sections, AQM is compared to a non-QoS scheme under realistic scenario assumptions with regard to these four criteria.

Table 1
QoS Classes and Requirements

| Class | Application | Bandwidth | Duration | Delay |
|---|---|---|---|---|
| 0 | High-quality voice | 128 Kbps | 1 200 s | 10 ms |
| 1 | CD-quality audio | 256 Kbps | 2 400 s | 90 ms |
| 2 | Video conference | 2 Mbps | 1 200 s | 10 ms |
| 3 | High-quality video | 3 Mbps | 4 800 s | 90 ms |

Table 2
Simulation Parameters

| Parameter Description | Value |
|---|---|
| Greeting message interval | 10 s |
| Maximum available bandwidth | 10 Mbps |
| Mobility model | Random waypoint |
| Node speed | 1-4 m/s (uniform) |
| Node pause time | 10-400 s (uniform) |
| Node idle time | 300 s (exponential) |
| Service class distribution | 0:40%; 1:20%; 2:30%; 3:10% |
| Session generation / joining ratio | 1 / 9 |
| Simulation duration | 4 h |
| Square area edge lengths | 200-400-600-800-1000-1200 m |
| Transmission ranges | 50-100-150-200-250-300 m |

### 5.1. Simulation Settings

The simulations are conducted using OPNET Modeler 10.5 Educational Version and Wireless Module [30]. AQM nodes are modelled in three layers with application, session, and network managers. The application manager is responsible for selecting the type of application to run, setting its QoS requirements, as well as making decisions on session initiation, termination, join and leave. The session manager is responsible for declaring new sessions initiated by its application manager to other nodes, sending requests for sessions its application manager wishes to join, keeping lists of sessions, members and requests of other nodes, processing and forwarding their information messages, and taking part in their join processes when necessary. The network manager is responsible for packet arrival and delivery, in addition to broadcasting periodic greeting messages and receiving other nodes' greeting messages in order to process them to derive free bandwidth information.

The non-QoS scheme is basically a modified version of AQM which does not make any intelligent decisions based on QoS availability when responding to session join requests. In the non-QoS scheme, all sessions are announced along the network, and all nodes can join all sessions regardless from bandwidth and delay limitations.

Simulations are repeated 20 times for each data point and results are aggregated with a 95% confidence interval for a multicast scenario with four QoS classes to represent a sample set of applications. Nodes initiate or join sessions according to a certain probability. Generated sessions are assigned randomly to one of the four QoS classes defined in Table 1. To comply with the sample bandwidth and delay bounds given as part of these QoS class definitions, nodes are restricted to certain minimum bandwidth and maximum hop count regulations. In other words, a node is allowed to join a session only if it can find a path to the server with more bandwidth available than the minimum amount and less hops away than the maximum allowed. Apart from that, there is no limit to the size of the multicast groups. The effect of mobility on the performance of AQM is observed under the random waypoint mobility model. In contrast to previous performance evaluations in the research literature, which limit their simulations to a few minutes, four hours of network lifetime have been simulated to get a realistic impression of the aggregated behaviour of multiple multicast sessions being maintained simultaneously in a distributed manner. The parameters of the mobility model and other simulation settings are given in Table 2. Two sets of simulations are conducted using these parameters. In the first set, the network population is selected as variable and the wireless transmission range is fixed at 250 m in an area of 1000 m by 1000 m. In the second set, the range is selected as variable whereas the population is fixed at 100 nodes. In order to keep the node densities of the networks in the second set close to each other, the size of the network area is varied accordingly.

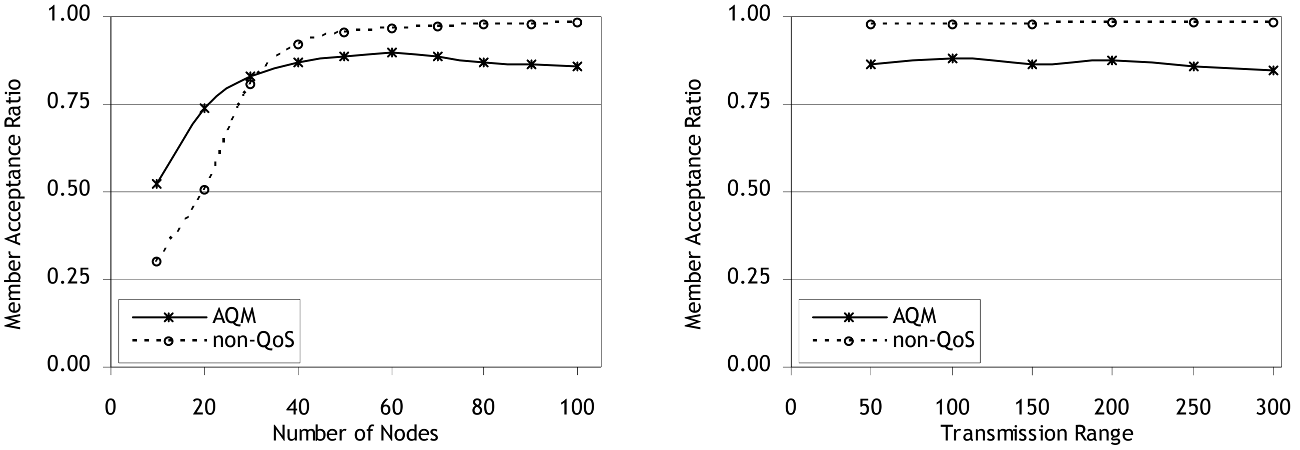
A node can take part at only one application at a time as a server or receiver. However, it can participate in any number of sessions as a forwarder as long as QoS conditions allow. The usage scenarios consist of open-air occasions such as search and rescue efforts and visits to nature in an area with boundaries, where a network infrastructure is not available.

### 5.2. Evaluation of Member Satisfaction

The success of a QoS multicast routing system depends primarily on the satisfaction of its members. In this regard, two important indications of member satisfaction are a high probability of being accepted by a requested session, and the sustainability of the application at an acceptable quality level once accepted by that session. In addition, a system which is developed to utilise scarce resources effectively has to produce little overhead itself. In this section, AQM is evaluated with regard to these three factors.

Figure 5 compares the member acceptance ratio of AQM to the non-QoS scheme with regard to: (a) rising network population; and (b) increasing transmission range. In sparse networks with a small number of nodes and low connectivity, AQM performs better than the non-QoS scheme since it informs its nodes periodically on the availability of ongoing sessions and prevents them from making requests for sessions that are not reachable. As the network density grows, the non-QoS scheme starts performing better since multicast sessions accept almost all join requests they receive. Only a small percentage of the requests fail, mainly due to the changes in the status of the intermediate nodes that reply to the requests. In AQM, where QoS restrictions apply, nodes do not accept new requests if they cannot afford the required free bandwidth. Thus, not all requests are granted an acceptance and the member acceptance ratio is lower than the non-QoS scheme. This ratio decreases slightly as the number of nodes in the network increases, due to the fact that more nodes try to join sessions simultaneously while the network capacity is the same. However, AQM is still able to achieve an acceptance ratio close to the non-QoS scheme due to its ability to eliminate infeasible join requests preliminarily by keeping its nodes up-to-date regarding the QoS conditions in the network and the status of the existing sessions. Similar trends are observed in the variable transmission range experiments, where there is a slight decrease in the member acceptance ratio of AQM as the range increases, which can be explained by the interference between the nodes causing larger neighbourhoods. Thus, the same resources have to be shared by more nodes. In summary, AQM rejects some of the join requests to prevent existing session members from being overloaded, and provides better conditions for them.

Figure 6 compares the member overload ratio of AQM to the non-QoS scheme. In AQM, where QoS support is active, nodes do not make allocations exceeding the maximum bandwidth available in their neighbourhood. The number of overloaded members is kept to a minimum with



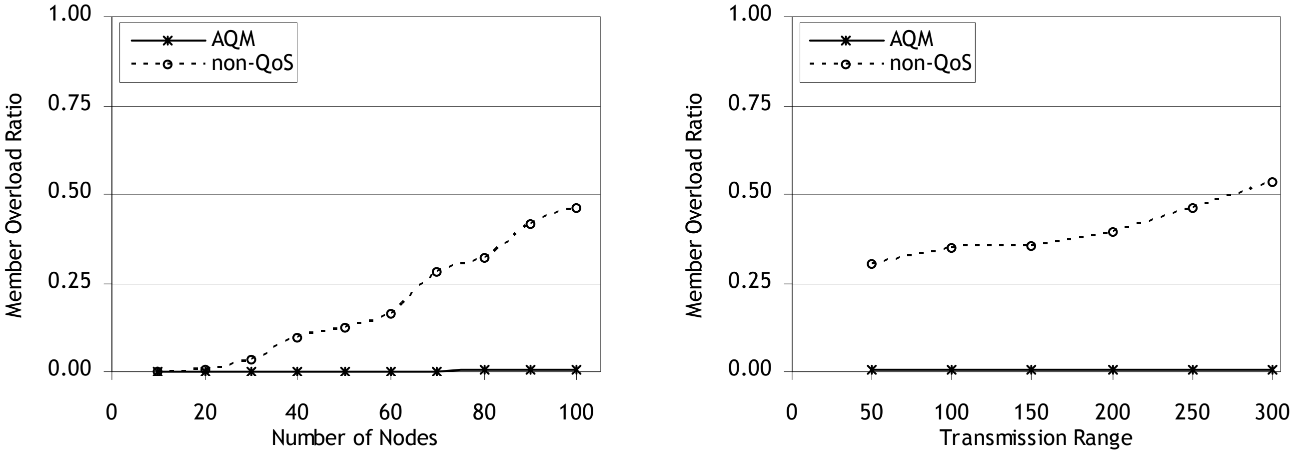
(a) Variable network population, transmission range fixed at 250 m.

(b) Variable transmission range, network population fixed at 100 nodes.

Fig. 5: Comparison of the member acceptance ratio of AQM to a non-QoS scheme.

the introduction of the objection query mechanism. In the non-QoS scheme, nodes directly accept join requests as soon as they are aware of any path towards the session server. Since they are not limited by available resources, they soon become overloaded. As the number of nodes in the network grows, more sessions are initiated, and more requests are accepted per node without taking care of the available bandwidth, which causes a drastic increase in the ratio of overloaded members for the non-QoS network. As the transmission range increases and neighbourhoods grow, on the other hand, ad hoc nodes start suffering from collisions and packet losses due to interference. The results show that AQM outperforms the non-QoS scheme with its ability to prevent members from being overloaded. In fact, the ratio of overloaded members is practically 0 for AQM.

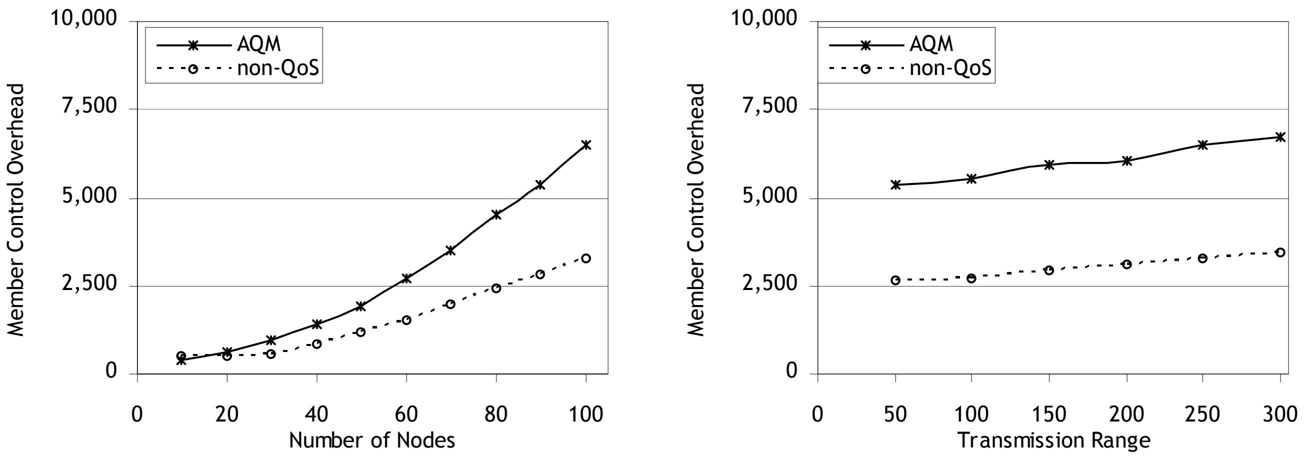
Figure 7 compares the member control overhead of AQM to the non-QoS scheme. For both schemes, the number of control messages per member increases smoothly as the network population or a node's transmission range grows. In addition to its periodic greeting and session update packets, AQM necessitates more one-hop objection queries to ensure that none of the session members becomes overloaded since more multicast paths are possible between neighbours in a more crowded network. It also generates more lost session notifications since there are more nodes in each neighbourhood which become aware of each others existence as well as disappearance. It can be concluded from the figure that AQM has a control overhead twice as much as the non-QoS scheme. However, AQM provides QoS with an acceptable



(a) Variable network population, transmission range fixed at 250 m.

(b) Variable transmission range, network population fixed at 100 nodes.

Fig. 6: Comparison of the member overload ratio of AQM to a non-QoS scheme.



(a) Variable network population, transmission range fixed at 250 m. (b) Variable transmission range, network population fixed at 100 nodes.

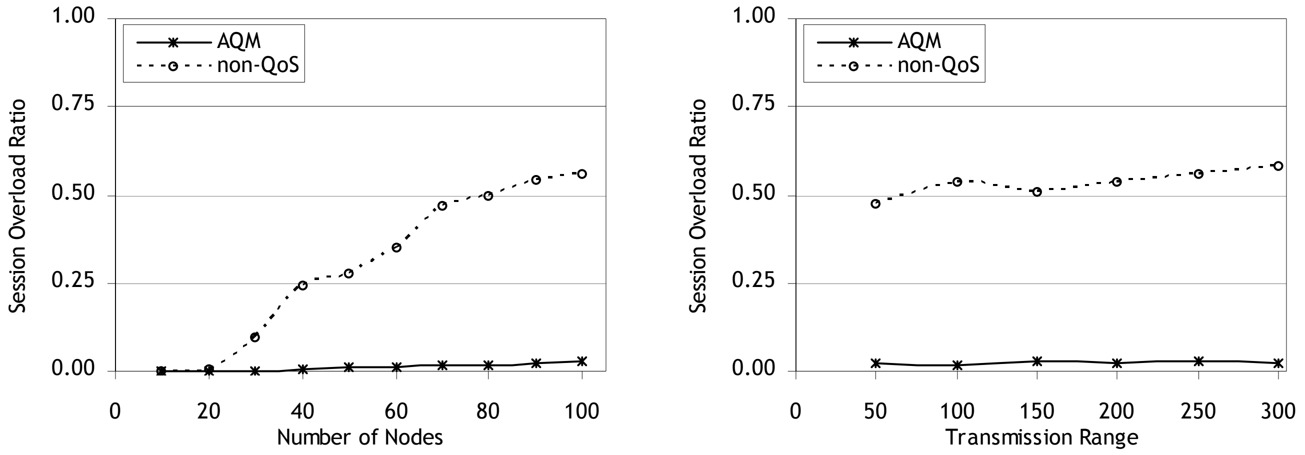
Fig. 7: Comparison of the member control overhead of AQM to a non-QoS scheme.

overhead. In fact, by rejecting some of the join requests, AQM cuts further communication with those nodes, whereas the non-QoS scheme communicates with all requesters until their routing information is delivered.

Figures 5, 6 and 7 show that the QoS support provided by AQM increases member satisfaction during multicast sessions significantly, especially when the number of nodes in the network grows or interference is high due to an increase in the transmission range. It is a widely accepted assumption that dropped connections are generally more annoying than rejected ones. Thus, the member overload ratio is clearly more important than the member acceptance ratio since an overloaded node affects all its neighbours as well as all related sessions and causes collisions, packet losses and intolerable delays. While the application of QoS restrictions causes more users to be rejected than a non-QoS scheme, the lack of these restrictions yields to catastrophic results. Without a policy to manage network resources effectively, users experience difficulties in getting any service as bandwidth requirements increase.

### 5.3. Evaluation of Session Satisfaction

In addition to evaluating the satisfaction of individual members, it is also necessary to make sure that AQM improves the QoS of a session with all its members as a whole. The session overload ratio is an indication of the number of sessions experiencing QoS violations caused by their members due to resource allocations exceeding the network limitations. This is an important criterion since an overloaded intermediate session member has a negative



(a) Variable network population, transmission range fixed at 250 m. (b) Variable transmission range, network population fixed at 100 nodes.

Fig. 8: Comparison of the session overload ratio of AQM to a non-QoS scheme.

effect on all the downstream members of that session.

Figure 8 compares the session overload ratio of AQM to the non-QoS scheme, once again with regard to: (a) rising node population; and (b) increasing wireless transmission range. The overload experienced by AQM sessions barely increases as the number of nodes in the network grows, whereas the non-QoS scheme starts suffering from heavy overload in most of its sessions. The difference becomes clearer as the increased wireless transmission range of the nodes starts causing collisions. The reason for this impact is the hidden terminal problem mentioned previously. AQM solves this problem by using objection queries whereas the non-QoS scheme suffers from it in an even larger neighbourhood.

Figure 8 shows that QoS support also increases session satisfaction significantly. An overloaded member causes performance decrease on all the sessions it is serving. AQM achieves better performance by decreasing the number of overloaded members and sessions while keeping the number of rejected join requests to an acceptable level.

## 6. Conclusion

Novel wireless networking technologies embedded into portable computing devices enable an ever-growing number of users to communicate with each other while on the move, i.e., without being connected to a wired infrastructure. As soon as the user becomes part of such a wireless network, however, a series of heavy administrative tasks have to be accomplished to configure the device. Ad hoc networks are self-organising communication entities which take over this burden and make the user enjoy full wireless functionality.

On the other hand, the increasing amount of multimedia content shared over various communication media today makes QoS-related, resource-efficient routing strategies very important for ad hoc networks. The multicast routing scheme presented in this article, AQM, provides ad hoc networks with these features. It keeps the network up-to-date on the availability of sessions with regard to QoS considerations. It controls the availability of resources throughout the network and ensures that the users of an application do not suffer from QoS degradation due to bandwidth allocations exceeding the limits of the shared wireless medium. In its bandwidth calculations, AQM takes the continuity property of multimedia data into consideration and checks bandwidth availability along a virtual tunnel of nodes. It also facilitates an objection query mechanism to inform nodes on possible overload on others that cannot be directly detected due to the hidden terminal effect. AQM also sets limits to path length in terms of hop count and checks them in order to satisfy the delay requirements of multimedia applications. By applying these instruments, AQM is able to eliminate infeasible requests for membership preliminarily at their sources.

Service satisfaction is the primary evaluation criterion for a QoS-related scheme. Therefore, two new metrics are defined to compare AQM to a non-QoS scheme, in addition to two regular metrics such as the routing success rate and the control overhead. These new metrics are the ratio of overloaded session members, and the ratio of sessions with overloaded members. The results give a good insight to the quality of AQM. Simulations show that there are significant performance differences between the two schemes for members and sessions. By applying QoS restrictions to the ad hoc network, AQM achieves lower overload on members and improves the multicast efficiency for members and sessions. Without a QoS scheme, users experience difficulties in getting the service they demand as the network population grows and bandwidth requirements increase. AQM proves that QoS is essential for and applicable to ad hoc multimedia networks.

A possible future research direction is the assessment of the recent multicast routing protocols in the literature using the same criteria as above to have an alternate view to their performance in terms of QoS as experienced by the user. A second topic is the efficient rerouting of multicast sessions when changes occur in the network topology as a result of mobility or varying QoS conditions. On the other hand, the implementation of a realistic mobility model is also very important for an accurate evaluation of these protocols. Mobility changes the network topology constantly, which has a profound effect on the network characteristics. It is a good idea to evaluate ad hoc network protocols with multiple mobility models. Ad hoc applications with team collaboration and real-time multimedia support necessitate group mobility, which improves performance if protocols take advantage of its features such as multicast routing.

The scope of this article is the design of a QoS scheme for ad hoc multicast routing and to validate it as a feasible and useful one at the higher layers. However, further study is necessary to implement a realistic MAC layer and simulate ad hoc network environments closer to real life scenarios. Reliable MAC broadcasting is a hard task due to the request-to-send/clear-to-send (RTS/CTS) signalling problem. The MAC layer is also responsible for resource reservation and the acquisition of available link bandwidth information, which is another significant issue involving infrastructure decisions. On the other hand, AQM is independent of the design of lower layers, and within the scope of this work, efforts have been made to maintain its integrity by addressing these issues in higher layers.

AQM has a simple, flat network structure where all nodes are equal. It avoids complicated network topologies such as hierarchical or clustered structures, which are challenging in terms of design and maintenance and present points of failure. However, it is possible to adapt AQM to a clustered network to scale with network size. Intra-cluster multicast sessions can be handled by AQM, whereas inter-cluster communication can be managed by a higher-layer, hierarchical version of it, still providing the network with QoS features. It is not a realistic assumption that a mobile network can afford a pure on-demand scheme if it has to support QoS. Therefore, AQM proposes a hybrid method in terms of multicast routing with table-driven session management and on-demand verification of QoS information upon the initialisation of a join process.

## References

[1] Chakrabarti, S, and A. Mishra, "QoS Issues in Ad Hoc Wireless Networks," *IEEE Communications Magazine*, Vol. 39, No. 2, pp. 142-148, February 2001.

[2] Zhang, Q, W. Zhu, G.J. Wang, and Y.Q. Zhang, "Resource Allocation with Adaptive QoS for Multimedia Transmission over W-CDMA Channels," *Proceedings of IEEE WCNC,* Chicago, USA, September 2000.

[3] Mohapatra, P, J. Li, and C. Gui, "QoS in Mobile Ad Hoc Networks," *IEEE Wireless Communications,* Vol. 10, No. 3, pp. 44-52, June 2003.

[4] Cordeiro C.M, H. Gossain, and D.P. Agrawal, "Multicast over Wireless Mobile Ad Hoc Networks: Present and Future Directions," *IEEE Network*, Vol. 17, No. 1, pp. 52-59, January 2003.

[5] Lin, C.R, "On-Demand QoS Routing in Multihop Mobile Networks," *Proceedings of IEEE INFOCOM,* Alaska, USA, April 2001.

[6] Zhu, C, and M.S. Corson, "QoS Routing for Mobile Ad Hoc Networks," *Proceedings of IEEE INFOCOM,* New York, USA, June 2002.

[7] Chen, Y.S, Y.C. Tseng, J.P. Sheu, and P.H. Kuo, "An on-demand, link-state, multi-path QoS routing in a wireless mobile ad-hoc network," *Elsevier Science Computer Communications*, Vol. 27, No. 1, pp. 27-40, January 2004.

[8] Chen, S, and K. Nahrstedt, "Distributed Quality of Service Routing in Ad Hoc Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 8, pp. 1488-1505, August 1999.

[9] Pattara-Atikom, W, and P. Krishnamurthy, "Distributed Mechanisms for Quality of Service in Wireless LANs", *IEEE Wireless Communications*, Vol. 10, No. 3, pp. 26-33, June 2003.

[10] Zhu, H, M. Li, I. Chlamtac, and B. Prabhakaran, "A Survey of Quality of Service in IEEE 802.11 Networks", *IEEE Wireless Communications*, Vol. 11, No. 4, pp. 6-14, August 2004.

[11] Yang, Y, and R. Kravets, "Contention-Aware Admission Control for Ad Hoc Networks", *IEEE Transactions on Mobile Computing,* to appear.

[12] Barry, M, B.J. Hogan, and S. McGrath, "Congestion Avoidance in Source Routed Ad Hoc Networks", *Proceedings of the IST Mobile Summit*, Lyon, France, June 2004.

[13] Peng, J, and B. Sikdar, "A Multicast Congestion Control Scheme for Mobile Ad-Hoc Networks", *Proceedings of IEEE Globecom*, San Francisco, USA, December 2003.

[14] Royer, E.M, and C.E. Perkins, "Multicast Operation of the Ad-Hoc On-Demand Distance Vector Routing Protocol," *Proceedings of ACM/IEEE MOBICOM,* Seattle, USA, August 1999.

[15] Royer, E.M, and C.E. Perkins, "Multicast Ad Hoc On-Demand Distance Vector (MAODV) Routing," *IETF MANET Working Group Internet Draft*, work in progress, July 2000.

[16] Garcia-Luna-Aceves, J.J, and E.L. Madruga, "A multicast Routing Protocol for Ad-Hoc Networks," *Proceedings of IEEE INFOCOM,* New York, USA, March 1999.

[17] Madruga, E.L, and J.J. Garcia-Luna-Aceves, "Scalable Multicasting: The Core-Assisted Mesh Protocol," *Kluwer Mobile Networks and Applications*, Vol. 6, No. 2, pp. 151-165, March 2001.

[18] Ozaki, T, J.B. Kim, and T. Suda, "Bandwidth-Efficient Multicast Routing for Multihop, Ad Hoc Wireless Networks," *Proceedings of IEEE INFOCOM,* Alaska, USA, April 2001.

[19] Lee, S.J, W. Su, and M. Gerla, "On-Demand Multicast Routing Protocol (ODMRP) for Ad Hoc Networks," *IETF MANET Working Group Internet Draft*, work in progress, January 2000.

[20] Lee, S.J, W. Su, and M. Gerla, "On-Demand Multicast Routing Protocol in Multihop Wireless Mobile Networks," *ACM Mobile Networks and Applications*, Vol. 7, No. 6, pp. 441-453, December 2002.

[21] Lee, S, and C. Kim, "A new wireless ad hoc multicast routing protocol," *Elsevier Science Computer Networks*, Vol. 38, No. 2, pp. 121-135, February 2002.

[22] Toh, C.K, Ad Hoc Mobile Wireless Networks: Protocols and Systems, 1st edition, Prentice Hall PTR, 2002.

[23] Ji, L, and M.S. Corson, "Differential Destination Multicast – A MANET Multicast Routing Protocol for Small Groups," *Proceedings of IEEE INFOCOM,* Alaska, USA, April 2001.

[24] Ji, L, and M.S. Corson, "Explicit Multicasting for Mobile Ad Hoc Networks," *ACM Mobile Networks and Applications*, Vol. 8, No. 5, pp. 535-549, October 2003.

[25] Sajama, S, and Z.J. Haas, "Independent-Tree Ad Hoc Multicast Routing (ITAMAR)," *ACM Mobile Networks and Applications*, Vol. 8, No. 5, pp. 551-566, October 2003.

[26] Chen, Y.S, Y.W. Ko, and T.L. Lin, "A Lantern-Tree-Based QoS Multicast Protocol for Wireless Ad Hoc Networks," *Proceedings of IEEE ICCCN,* Miami, USA, October 2002.

[27] Chen, Y.S, and Y.W. Ko, "A Lantern-Tree-Based QoS On-Demand Multicast Protocol for a Wireless Mobile Ad Hoc Network," *IEICE Transactions on Communications,* Vol. E87-B, No. 3, March 2004.

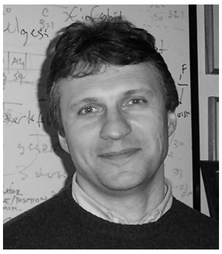
[28] Pompili, D, and M. Vittucci, "A Probabilistic Predictive Multicast Algorithm in Ad Hoc Networks (PPMA)," *Proceedings of MED-HOC-NET,* Mugla, Turkey, June 2004.

[29] Corson, S, and J. Macker, "Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," *RFC 2501*, January 1999.

[30] OPNET Technologies Inc, Bethesda, MD, USA, available at http://www.opnet.com.

## Vitae

*Kaan Bür received his BS degree in control and computer engineering from Istanbul Technical University in 1995 and his MS degree in computer engineering from Bogaziçi University in 1998. He worked first as an I.T. engineer and then as an R&D engineer in Beko Elektronik A.S. between 1995 and 2002. Currently, he is a PhD candidate in the Computer Engineering Department of Bogaziçi University. His research interests include high-speed networks, wireless, mobile and multimedia communications. Kaan Bür is a student member of IEEE.*



*Cem Ersoy received his BS and MS degrees in electrical engineering from Bogaziçi University, Istanbul, in 1984 and 1986, respectively. He worked as an R&D engineer in NETAS A.S. between 1984 and 1986. He received his PhD in electrical engineering from Polytechnic University, Brooklyn, New York in 1992. Currently, he is a professor in and head of the Computer Engineering Department of Bogaziçi University. His research interests include performance evaluation and topological design of communication networks, wireless communications and mobile applications. Dr. Ersoy is a senior member of IEEE.*