Multicast routing for ad hoc networks with a multiclass scheme for quality of service

Bür, Kaan; Ersoy, Cem

Published in:
Lecture Notes in Computer Science

DOI:
10.1007/978-3-540-30182-0_20

2004

Document Version:
Peer reviewed version (aka post-print)

Link to publication

Citation for published version (APA):
Bür, K., & Ersoy, C. (2004). Multicast routing for ad hoc networks with a multiclass scheme for quality of service. In C. Aykanat, T. Dayar, & I. Körpeoğlu (Eds.), *Lecture Notes in Computer Science* (Vol. 3280, pp. 187-197). Springer. https://doi.org/10.1007/978-3-540-30182-0_20

Total number of authors:
2

Multicast Routing for Ad Hoc Networks with a Multiclass Scheme for Quality of Service

Kaan Bür, Cem Ersoy

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

Abstract. As multimedia- and group-oriented computing becomes increasingly popular for the users of wireless mobile networks, the importance of features like quality of service (QoS) and multicasting support grows. Ad hoc networks can provide users with the mobility they demand, if efficient QoS multicasting strategies are developed. The ad hoc QoS multicasting (AQM) protocol achieves multicasting efficiency by tracking resource availability within a node's neighbourhood and announces it at session initiation. When nodes join a session of a certain QoS class, this information is updated and used to select the most appropriate routes. AQM is compared to a non-QoS scheme with emphasis on service satisfaction of members and sessions in an environment with multiple service classes. By applying QoS restrictions, AQM improves the multicasting efficiency for members and sessions. The results show that QoS is essential for and applicable to multicast routing in ad hoc networks.

1 Introduction

The increasing popularity of video, voice and data communications over the Internet and the rapid penetration of mobile telephony have changed the expectations of wireless users. Voice communication is accompanied by multimedia and the need for group-oriented services and applications is increasing. Therefore, it is essential that wireless and multimedia be brought together [1]. Ad hoc networks are communication groups formed by wireless mobile hosts without any infrastructure or centralised control, which can accompany these developments.

Quality of service (QoS) support for multimedia applications 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]. However, it is a significant technical challenge to provide reliable high-speed end-to-end communications in these networks, due to their dynamic topology, distributed management, and multihop connections [3]. In this regard, multicasting is a promising technique, the advantage of which is that packets are only multiplexed when it is necessary to reach two or more receivers on disjoint paths.

This work was supported in part by the State Planning Organisation, Turkey, under Grant No. DPT98K120890 and OPNET Technologies University Research Program.

It is not an easy task to incorporate QoS to ad hoc multicasting. Incremental changes on existing schemes cannot address the critical issues mentioned above efficiently. In this paper, the ad hoc QoS multicasting (AQM) protocol is presented to improve multicasting efficiency through QoS management. AQM tracks availability of QoS within a node's neighbourhood based on previous reservations in a network of multiple service classes, and announces it at session initiation. During the join process, this information is updated and used to select routes which can satisfy the QoS requirements of the session. Thus, AQM significantly improves the multicasting efficiency for members and sessions. The rest of this paper is organised as follows. Previous research related to ad hoc multicasting is summarised in Chapter 2. AQM is introduced in Chapter 3. The performance of the proposed system is evaluated in Chapter 4. Concluding remarks and future work are presented in Chapter 5.

2 An Overview to Ad Hoc Multicasting Protocols

Several protocols have been developed to perform ad hoc multicast routing. However, they do not address the QoS aspect of ad hoc communication, which is becoming increasingly important as the demand for mobile multimedia increases.

Associativity-based ad hoc multicast (ABAM) builds a source-based multicast tree [4]. Association stability, which results 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 identifies its receivers. To join a multicast tree, a node broadcasts a request, collects replies from group members, selects the best route with a selection algorithm, and sends a confirmation. To leave a tree, a notification is propagated upstream along the tree until a branching or receiving node is reached.

Neighbour-supporting multicast protocol (NSMP) utilises node locality to reduce route maintenance overhead [5]. 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.

Differential destination multicast (DDM) lets source nodes manage group membership, and stores multicast forwarding state information encoded in headers of data packets to achieve stateless multicasting [6]. 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. Forwarding computation is based on destinations encoded in the headers, where each node checks the header for any DDM block or poll flag intended for it.

Multicast ad hoc on demand distance vector (MAODV) routing protocol is derived from AODV [7]. The multicast group leader maintains a group sequence number and broadcasts it periodically to keep fresh the routing information. A node wishing to join a multicast group generates a route request. Only the leader or members of the multicast group may respond to a join request by unicasting a route reply back 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 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 on-demand multicast routing protocol (ODMRP) introduces the concept of a forwarding group [8]. 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.

3 The Ad Hoc QoS Multicasting Protocol

As mobile multimedia applications and group communication become popular for wireless users, ad hoc networks have to support QoS for multicasting. A QoS strategy should handle the reservation of resources, the optimisation of loss and delay to acceptable levels, and the implementation of QoS classes efficiently. In the following sections, the structural components of AQM are defined, which address these issues. Design details include the usage of QoS classes and levels, session initiation and destruction, membership management, and neighbourhood maintenance.

In this work, four QoS classes are suggested to represent a sample set of applications to be supported by the ad hoc network. Defining QoS classes limits the amount of information to be transmitted. It is otherwise impossible to forward a best QoS combination without making some assumptions or losing some valuable data. It is preferable that nodes inform others on the availability of certain QoS conditions and send updates only when they change.

3.1 Session Initiation and Destruction

A session is defined by its identity number, application type, QoS class and, if predictable, duration and cost. A node starts a session by broadcasting a session initiation packet (SES_INIT). Thus, it becomes a session initiator (MCN_INIT). A table of active sessions (TBL_SESSION) is maintained at each node to keep the information on the session definitions. Figure 1 shows the phases of session initiation.

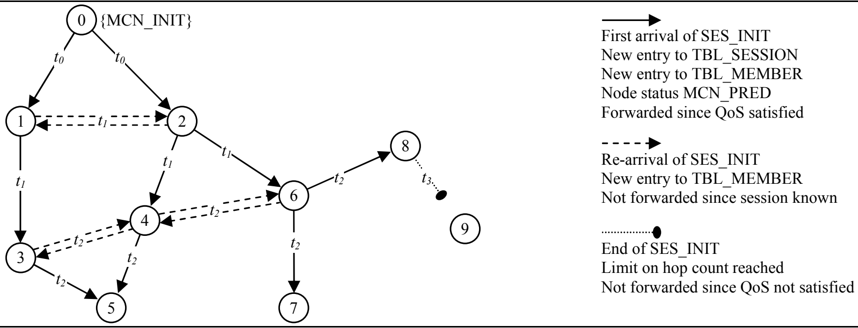


Fig. 1. The AQM session initiation process: SES_INIT is broadcast by MCN_INIT n_0 for a new session. It propagates through the network, informing all nodes from n_1 to n_8, which update their TBL_SESSION and TBL_MEMBER. n_9 is not informed since it is beyond the QoS limits in terms of hop count, which is used as a measure of end-to-end delay. $t_i < t_{i+1}$, $0 \leq i \leq 3$, represent the relative timing of the messages.

Using their session tables, nodes forward initiation packets of new sessions. A membership table (TBL_MEMBER) is used to denote the status of the predecessors (MCN_PRED) which have informed the node about a particular multicast session, and the QoS status of the path from the session initiator up to that node via that predecessor. Session initiation packets are forwarded as long as QoS requirements are met. Before a packet is rebroadcast, each node updates its QoS information fields with the current QoS conditions. The packet is dropped if QoS requirements cannot be met any more, avoiding flooding the network unnecessarily. Hop count information in the packets is used to prevent loops. Successful propagation of session initiation data is an important factor for the efficiency of subsequent session joining processes.

The session is closed by its initiator with a session destruction (SES_DESTROY) message. Upon receiving it, all nodes clean their tables. Member nodes forwarding multicast data also free their resources allocated to that session. A node receiving a session destruction 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.

3.2 Membership Management

A node can directly join a session if it is already a forwarding node in that session. Otherwise, it broadcasts a join request packet (JOIN_REQ) containing the session information. The predecessors of the requester propagate it upstream as long as QoS is satisfied. Ad hoc networks are highly dynamic, and available resources may change considerably since the arrival of the QoS conditions with the session initiation packet. Therefore, QoS conditions are checked at each node to make sure that current available resources allow the acceptance of a new session. 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.

The forwarded request reaches nodes which are already members of that session and can directly send a reply (JOIN_REP). Members of a session are the initiator, the forwarders, and the receivers. Downstream nodes, having initiated or forwarded join requests, thus waiting for replies, aggregate these and forward only the reply offering the best QoS conditions towards the requester. 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.

Intermediate nodes check the reserve packet to see whether they are forwarders on the path from the selected replier to the requester. If this is the case, they change their status from predecessor to forwarder (MCN_FWD), reserve resources, and update their membership tables to keep a list of successors. They send 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, or a receiver. If the 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 has 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 succeeds, intermediate nodes have enough routing and membership data to take part in the packet forwarding task. When a node sends multicast packets, 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.

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. If the forwarding node itself is not a receiver, it frees resources and notifies its forwarder of its own leave.

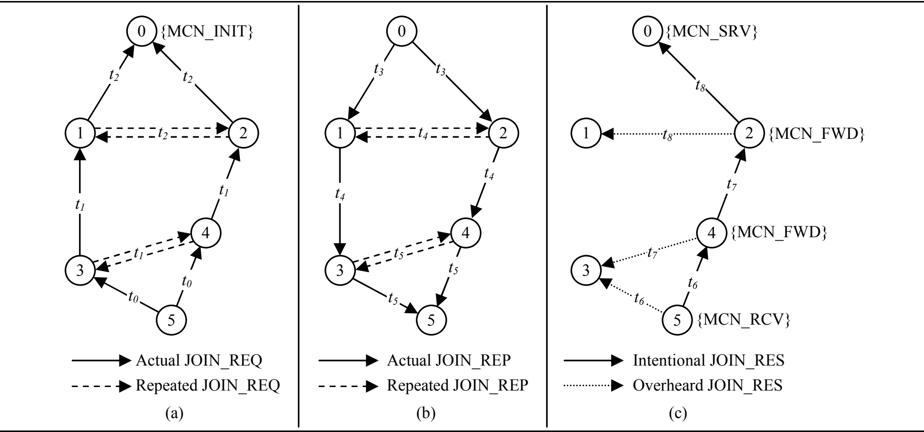


Fig. 2. The AQM session joining process: (a) JOIN_REQ is issued by n_5. It propagates through the network as long as QoS can be satisfied, until it reaches some members of the session. Nodes from n_1 to n_4 update their TBL_REQUEST as they forward the packet since they are not session members. (b) JOIN_REP is sent back from MCN_INIT 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.

3.3 Neighbourhood Maintenance

The nodes in an ad hoc network have to maintain their connectivity information with as much accuracy as possible to support QoS. This includes the ability to keep track of available bandwidth within their transmission range, and 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.

Each node broadcasts periodic greeting messages (NBR_HELLO), informing its neighbours on its bandwidth usage determined by the QoS classes of the sessions being served or forwarded by that node. To reduce overhead, greeting messages can be piggybacked to control or data messages. Each node aggregates the information in these messages to 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. If a node does not receive any greeting messages from a neighbour for a while, it considers that neighbour lost and deletes it from neighbourhood, session and membership tables.

Due to the broadcasting nature of the wireless medium, free bandwidth is node-based, i.e. a node's available bandwidth is the residual capacity in its neighbourhood. A node can only use the remaining capacity not used by itself and its immediate neighbours. This approach provides a sufficient method to measure bandwidth availability within a neighbourhood.

4 Performance Evaluation

Simulations are repeated multiple times in a network with four service classes as defined in Table 1. Nodes generate their own sessions or join other nodes' sessions with certain probabilities, which belong to one of these four classes. All simulation parameters are given in Table 2. The simulations are conducted using OPNET Modeler 10.0 Educational Version and Wireless Module [9]. The usage scenarios consist of open-air occasions such as search and rescue efforts or visits to nature in an area with boundaries, where a wired network infrastructure is not available. 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.

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 class, and making decisions on session initiation/destruction or join/leave. The session manager is responsible for declaring new sessions initiated by its application manager, sending requests for sessions it wishes to join, keeping lists of sessions, members and requests of other nodes, processing and forwarding their messages, and taking part in their join processes when necessary. The network manager is responsible for packet arrival and delivery, and for broadcasting periodic greeting messages to make the derivation of free bandwidth information possible.

Table 1. QoS classes and requirements

QoS Class	Bandwidth Requirement	Average Duration	Delay Tolerance	Application Type
0	128 Kbps	1,200 s	10 ms	High-quality voice
1	256 Kbps	2,400 s	100 ms	CD-quality streaming audio
2	3 Mbps	1,200 s	10 ms	TV-quality video conference
3	4 Mbps	4,800 s	90 ms	High-quality video

Table 2. Simulation parameters

Parameter Description	Value
Area size	400 m x 400 m
Greeting message interval	10 s
Maximum available bandwidth	10 Mbps
Node distribution (initial)	Uniform
Node idle times	Exponential (300 s; 600 s; 900 s; 1,200 s)
Service class distribution	0: 40%; 1: 20%; 2: 30%; 3: 10%
Session generation / joining ratio	1 / 9
Simulation duration	8 h
Wireless transmission range	200 m

Previous research efforts have mostly been evaluated through the use of important metrics which give a notion about the internal efficiency of a protocol. Two of these are data delivery ratio and control overhead [10]. However, the evaluation of QoS performance in ad hoc networks necessitates additional metrics. The main concern of this work is to evaluate the efficiency of AQM in providing multicast users with QoS and satisfying application requirements. Therefore, two new performance metrics, member- and session-level satisfaction grades, are introduced.

4.1 The Grade of Member Satisfaction

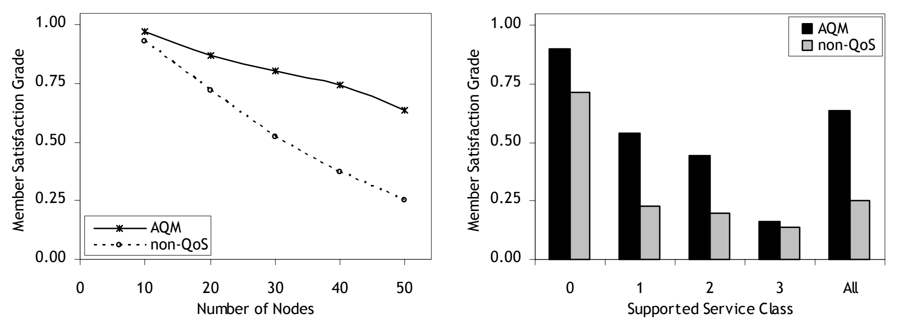
An important aspect of the QoS-related multicasting decisions made by AQM is the improvement in the ratio of overloaded member nodes, which has a direct impact on the satisfaction of session members regarding the multicasting service provided. On the other hand, the same decisions lead the network to reject more join requests than a non-QoS scheme. The member satisfaction grade S_{Member} is defined as the weighted sum of these two components to evaluate the member-level success ratio of AQM:

$$S_{Member} = \beta\left(1 - \frac{o}{s + \alpha f}\right) + (1 - \beta)\frac{r}{q} \quad . \tag{1}$$

In (1), 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 streaming nature of multimedia applications and the broadcasting nature of the wireless medium necessitate that session servers and forwarders have different bandwidth requirements within their neighbourhood. A server only takes its successors into consideration whereas a forwarder deals with its predecessor as well as its successors in terms of overload. Thus, the impact of overloaded neighbours on these nodes is not the same. To reflect this difference, *f* is multiplied by a coefficient *α*, which is set to 1.5 in the simulations. The division *o/(s+α f)* gives the ratio of overloaded nodes to all serving and forwarding nodes. Thus, the first term of the summation, multiplied by a relative weight coefficient *β*, represents a member overload prevention rate. Continuing with the second term, *r* is the number of receivers, and *q* is the total number of join requests issued by all mobile nodes. Their ratio reflects the success of the scheme in satisfying a node's request to join a session. The purpose of *β*, which can be varied between 0 and 1, is to adjust the relative weight of one term over the other according to the preferences of the ad hoc network. To give equal weights to overload prevention and member acceptance, *β* is set to 0.5 in the simulations. Other values are possible to change the network preferences.

Figure 3(a) compares the member satisfaction grades of AQM to a non-QoS scheme. In AQM, nodes do not accept more traffic than the bandwidth available in their neighbourhood. However, overloaded members still occur due to the hidden terminal problem. When QoS support is deactivated, nodes do not check their bandwidth limitations before replying to join requests. As a result of this, some of the serving or forwarding nodes become heavily overloaded, and their successors start suffering from collisions and packet losses. As the number of nodes grows, more requests are accepted per node without considering the available bandwidth, which causes a drastic decrease in member satisfaction grades. It can be concluded that the application of QoS restrictions significantly increases member satisfaction.

Figure 3(b) compares AQM to the non-QoS scheme with regard to the supported QoS class in a 50-node network. In each of the first four simulation pairs, all generated sessions belong to a single QoS class. AQM outperforms the non-QoS scheme in all of these classes. Moreover, AQM's overall performance increases as the network starts supporting multiple QoS classes. The reason for this improvement is that in AQM, sessions of lower classes can still be managed efficiently even if a join request for a higher-class has been rejected due to QoS restrictions. While the application of QoS causes more users to be rejected, the lack of these restrictions forces users to experience difficulties in getting any service as network population grows and bandwidth requirements increase.



(a) (b)

Fig. 3. Comparison of the member satisfaction grades of AQM and a non-QoS scheme: (a) Under support for multiple service classes. (b) Under support for single vs. multiple service classes with 50 nodes.

4.2 The Grade of Session Satisfaction

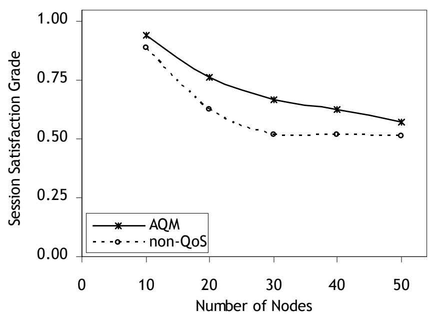
Rejection of some join requests and excessive bandwidth occupation by single nodes in a session affects all its members. It is necessary to observe the implications of these events on sessions. The session satisfaction grade $S_{Session}$ is defined as the weighted sum of these two components to evaluate the session-level success ratio of AQM:

$$S_{Session} = \gamma\left(1-\frac{l}{m}\right)+(1-\gamma)\left(1-\frac{j}{m}\right) . \quad (2)$$

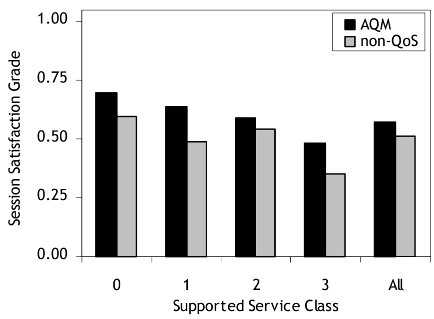
In (2), l is the number of sessions with at least one overloaded member, j is the number of sessions with at least one rejected join request, and m is the total number of sessions. The first term is the ratio of sessions without any overloaded members, whereas the second term reflects the success of AQM with regard to sessions without any rejections. The purpose of γ, which can be varied between 0 and 1, is to adjust the relative weight of one term over the other according to the preferences of the ad hoc network. To explicitly stress the effect of overloaded sessions on AQM, γ is set to 0.8 in the simulations. Other values are possible to change the network preferences.

Figure 4(a) compares the session satisfaction grades of AQM to the non-QoS scheme. Since AQM prevents single nodes from being overloaded more efficiently, it also achieves improvements in session satisfaction. However, unsatisfied sessions still occur. Some nodes become overloaded as a result of the allocations made by their neighbours that cannot be aware of each other's reservations due to the hidden terminal problem. When QoS support is deactivated, on the other hand, the lack of bandwidth restrictions causes more nodes to become overloaded, and as the network grows, more sessions are affected and session satisfaction decreases.

Figure 4(b) compares AQM to the non-QoS scheme with regard to the supported QoS class in a 50-node network. In each of the first four simulation pairs, all generated sessions belong to a single QoS class. AQM outperforms the non-QoS scheme in all of these classes. Thus, AQM achieves better performance by decreasing the number of overloaded members and sessions, at the cost of an acceptably increased number of rejected nodes.



(a)



(b)

Fig. 4. Comparison of the session satisfaction grades of AQM and a non-QoS scheme: (a) Under support for multiple service classes. (b) Under support for single vs. multiple service classes with 50 nodes.

5 Conclusion

AQM is designed to improve multicasting efficiency through the management of resources within each node's neighbourhood. It is compared to a non-QoS scheme in a realistic network scenario where multiple application service classes are supported. The primary evaluation criteria for AQM are service satisfaction grades defined both for members and sessions. Simulations show that, by applying QoS restrictions to the ad hoc network, AQM achieves significantly better results than a non-QoS scheme. Without QoS support, 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. It is not a realistic assumption that a mobile network can afford a pure on-demand scheme if it has to support QoS. AQM proposes a hybrid method in terms of multicasting with table-driven session management and on-demand verification of QoS information.

An important research direction is keeping the QoS data up-to-date, which is a major concern for a node in AQM, and involves the handling of lost neighbours, data exchange, and interpretation of a node's QoS status. A second issue closely related to QoS data accuracy is the hidden terminal problem. An extension to the request-reply-reserve process is necessary, whereby each replying node consults its neighbourhood to see if there are any objections. Neighbour awareness and discovery are typically handled by a periodic mechanism of beacons at the medium access control (MAC) layer. However, reliable MAC broadcast is a challenging 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. However, AQM is independent of the design of lower layers.

References

1. Van Nee, R, and R. Prasad, *OFDM for Wireless Multimedia Communications*, Artech House Publishers, 2000.
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. Walrand, J, and P. Varaiya, *High-Performance Communication Networks*, 2nd edition, Morgan Kaufmann Publishers, 2000.
4. Toh, C.K, Ad Hoc Mobile Wireless Networks: Protocols and Systems, Prentice Hall, 2002.
5. Lee, S, and C. Kim, "A new wireless ad hoc multicast routing protocol," *Elsevier Science Computer Networks*, Vol. 38, pp. 121-135, February 2002.
6. Ji, L, and M.S. Corson, "Explicit Multicasting for Mobile Ad Hoc Networks," *Mobile Networks and Applications (MONET)*, Vol. 8, No. 5, pp. 535-549, October 2003.
7. Royer, E.M, and C.E. Perkins, "Multicast Ad Hoc On-Demand Distance Vector (MAODV) Routing," *IETF MANET WG Internet Draft*, work in progress, July 2000.
8. Lee, S.J, W. Su, J. Hsu, and M. Gerla, "On-Demand Multicast Routing Protocol (ODMRP) for Ad Hoc Networks," *IETF MANET WG Internet Draft*, work in progress, January 2000.
9. OPNET Technologies Inc, Bethesda, MD, USA, available at http://www.opnet.com.
10. Corson, S, and J. Macker, "Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," *RFC 2501*, January 1999.