Using Cooperation in QoS Selection to Reduce Service Cost

Stefan Fischer
University of Mannheim, Praktische Informatik IV
D-68131 Mannheim, Germany
Email: stefis@pi4.informatik.uni-mannheim.de

Hermann de Meer
University of Hamburg, Department of Computer Science
Vogt-Koelln-Str. 30, 22527 Hamburg, Germany
Email: demeer@informatik.uni-hamburg.de

Abstract

In some electronic commerce applications, not only the process of selling and buying is organized electronically, but also the transmission of the product, for example in video-on-demand or digital TV broad- and multicasting. For such applications, transmission quality plays an important role since the customer will only be satisfied if he receives the product in the quality he paid for. Therefore, the transmission medium used should be able to offer a guaranteed service. *Quality of Service management systems* are used to provide such guarantees. However, most existing systems are not suitable for electronic commerce applications since they (a) often do not scale, and (b) they are not concerned with the relation between cost and quality. In this paper, we present a new distributed QoS management system that solves these problems. It is especially interesting for electronic commerce because it is able to lower the communication service cost, based on regional information available due to the distributed approach.

Keywords: Quality of Service, Cost-related Quality, Distributed QoS Management, Agent Cooperation, Service Cost

1 Introduction

There is a number of electronic commerce applications where the *Quality of Service (QoS)* provided by the electronic transmission medium between seller and buyer plays an important role, namely those where not only the process of selling and buying, but also the transport of products uses this medium. Examples for this kind of applications are video-on-demand, teleseminars or digital television broadcasts, which all include continuous and thus time-critical media like video and audio. Buyer[1], seller[2] and communication service provider (CSP) negotiate a certain price and quality for the product (a soccer match on TV, for instance) and its transmission. The buyer then expects the negotiated quality to be delivered, and in general, he will, during the transmission, not accept a lower quality for the same price.

In order to be sure to deliver the product in the quality paid for by the buyer, the CSP has to use a communication service that *guarantees* a negotiated quality. This is achieved by employing a *QoS management system* the basic task of which is to manage the communication resources distributed throughout the network in such a way that a resource is available for processing a client request when necessary. Usually, this is automatically guaranteed in connection-oriented networks like ISDN, where a connection always implies a certain bandwidth (64 kbit/s in ISDN), but not in packet-switched ones like the Internet. QoS management systems consist of several management functions such as QoS specification, negotiation, renegotiation and adaptation. QoS negotiation, for instance, is necessary to find a common understanding between service user and service provider about the quality desired and the corresponding cost.

A number of QoS management systems has been developed [1]; however, most of them are not dealing with the notion of cost, but rather concentrate on technical issues of how to provide a certain quality (typically by reservation of resources). A further analysis reveals that they are also not very suitable for important future applications such as TV broadcasts because most of these systems work in a centralized way. Imagine a situation where hundreds of thousands of people want to watch a soccer match being transmitted in several qualities over a broadband network. Probably, they will all start to negotiate quality and cost at about the same time shortly before the match. As a result, the information seller will quickly be overloaded and not be able to communicate in time with all buyers. This type of QoS management does not scale.

Therefore, we developed a new QoS management system solving these problems: first, communication cost is an important issue in that the system tries to find a cost-optimal operation mode. Second, it is a *distributed management system* that is based on *QoS agents* throughout the network. From a point of view of a buyer, one such agent plays the role of the seller. In addition, agents *cooperate* in order to support further QoS functions such as QoS adaptation.

[1] In this paper, we use the terms "buyer", "sender" and "source" interchangeably, though the first one emphasizes the position in the elcetronic commerce scenario, while the latter are generally used when talking in terms of communication.

[2] Accordingly, we use the terms "seller" and "receiver" interchangeably.

Due to this cooperation, the system is called *Cooperative QoS Management* [4].

In this paper, we show how agent cooperation can be used to reduce the communication service cost to be paid for by the information buyer by optimizing resource usage. It allows CSPs to offer their guaranteed services for a lower price (at least to some service users) while getting some part of the profit themselves.

The paper is organized as follows: Section 2 gives a short introduction into the principles of Cooperative QoS Management. Section 3 then gives details of how agent cooperation may lead to service cost reduction. Section 4 is concerned with some monetary aspects of our approach, and Section 5 concludes the paper.

Important notions for the rest of this paper. We are especially interested in *multimedia mass applications*, which we define to be applications with a huge number of participants, each of them receiving several media streams from one source at the same time[3]. Typical examples are TV broadcasts and teleseminars, and to some extent near-video-on-demand systems where client requests are synchronized at some point in time.

Multicast transmission plays an important role in such applications. Instead of sending a given media data stream to each user separately, the source sends only one stream, and this stream is only split up on routers where receivers wait on different outgoing links. As a result, resources only have to be reserved once per stream on a router and not dependent on the number of users. Compared to broadcast, only those parts of the network with receivers attached to them really receive the data stream. A *multicast tree* is the multicast equivalent to a route from sender to receiver in point-to-point communication. The information source is the root and all receivers build the leafs. Branches result from splitting a stream on routers in the network. The tree is set up by a multicast routing algorithm. From one router's viewpoint, its *upstream* router is the neighboring router on the way to the source, while the *downstream* routers are all neighboring routers on the way to the leafs.

Finally, our notion of a certain *quality* of a stream represents a vector of values for all relevant QoS parameters. For instance, a given stream may be of quality q_1 which represents the vector (size=640x480;rate=30fps;color=truecolor).

2 Cooperative QoS Management

The principle idea of the new QoS management scheme is to distribute the QoS functions by installing *QoS agents* in the network. These agents act on behalf of the data source (the seller) in that they negotiate and renegotiate qualities of data streams with the buyers. In addition, they are able, due to their knowledge

[3]There may also be channels on which receivers send data, as for instance in teleseminars.

about data stream distribution in the network area they are responsible for, to support more technical QoS functions such as QoS adaptation .

The architecture of this system is depicted in Fig. 1. The product is distributed in different qualities via multicast. Agents communicate only with neighboring agents in order to avoid communication overload situations in regions near to a source. They may also get information from and send information to local system components such as QoS monitoring functions, reservation and routing protocols.

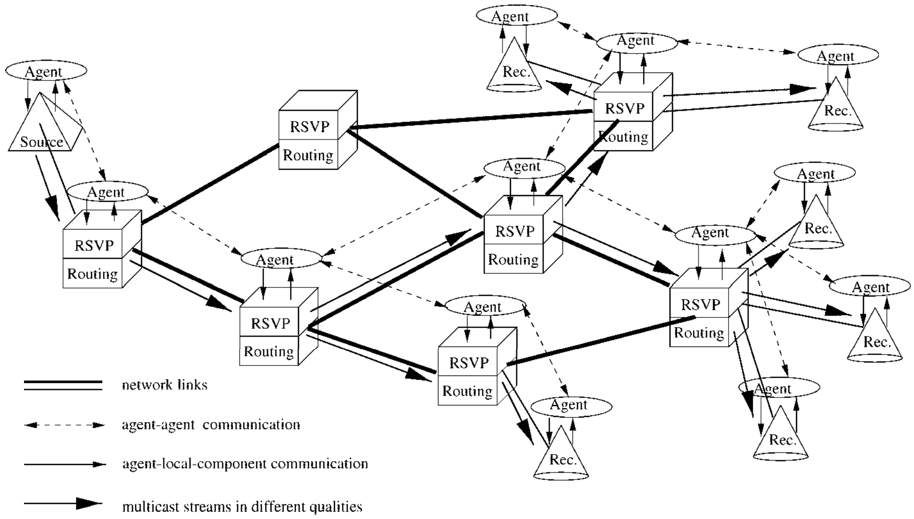

Figure 1: Architecture of Cooperative QoS Management

As soon as a router becomes part of the multicast tree for a given source (because a user wants to receive a product from this source), an agent is sent to this router and all other routers on the branch between the new user end system and the first router already part of the multicast tree. Also, one agent will be sent to the end system and will then contact its upstream agent in order to receive information about the available products and their qualities on this source. This agent, in turn, contacts its own upstream agent to ask for this information. This process continues until the first agent already part of the tree is contacted which then forwards the information about available qualities back downstream. On each link, the agents have to find out whether the link can support all available qualities by contacting the communication system. If a given quality cannot be supported on that link, it is removed from the list of available qualities and not offered to downstream agents. This is also the reason why agents do not already have the quality information when they are sent to their router. Finally, the end system agent will only receive information about those qualities supportable on the whole branch.

Based on the quality information available from the agent, a negotiation process takes place that includes the user, the end-system and router agents as well as the communication system. The negotiation process leads to one quality being selected by the user, and to resources being reserved in the network[4] in order to guarantee this quality. However, the reservation process does only have to cover that part of the tree where the stream in the desired quality is not yet available. On the remaining path to the source, resources are already reserved for this stream.

As soon as agents are set up on a router, they start to perform QoS management functions. Usually, they *monitor* the current QoS. Whenever they detect a QoS problem, they may become active by informing their neighboring agents. In [4, 2] we describe how this kind of agent cooperation can be used to support technical QoS function like QoS adaptation. [5] gives details about inter-agent protocols. In addition, agents may become active on their own in order to optimize system performance and resource usage. In the following section, we concentrate on the function *QoS renegotiation* and show how it can be used to reduce communication service cost.

3 Using cooperation to reduce service cost

In the applications we investigate here, namely multimedia mass applications, the use of multicast transmission is an important means to reduce communication cost. If, however, a source lets the users arbitrarily select the value of single QoS parameters, the advantage of multicasting quickly disappears, since resources have to be reserved for every single quality[5]. The more resources an application has to reserve, the more expensive the communication service will be.

The basic idea of reducing communication cost is to use the QoS renegotiation function in the following way: the information provider has the complete information about which receiver obtains which quality. He may then start renegotiation processes with single receivers in order to convince them to switch their quality. If he is able to convince all receivers of this quality, this stream does not have to be transmitted anymore, and the corresponding resources can be released. As a result, the communication cost will decrease. The ideal result would be that all buyers receive the product in the same quality such that resource usage is optimal from the point of view of the whole application.

This idea sounds fine, but with conventional QoS management systems, it is not very likely to be successful due to two reasons:

[4]This is the task of some underlying protocol and not part of an agent's work.

[5]The strength of this effect depends on the media coding scheme to be used: in hierarchical coding [6] it will be less important than in independent coding of every single stream ("simulcast").

1. Usually, a receiver has the possibility to tune single QoS parameters. As result, a huge number of qualities will be selectable. It is then hard to decide which receiver to convince to switch to which other quality.
2. In such systems, the information provider (the source of the multicast tree(s)) has the information on which receiver receives which quality. This information is of global type because it includes all receivers. The source does not know about the local distribution of receivers and qualities. It can thus only try to find a global solution in that it tries to convince all receivers of a certain quality to switch to another one. In some subtrees, this may be the wrong strategy, in case there are far more receivers of this quality than of the one to switch too. It will then be much more unlikely that all receivers accept the offer to switch (only if all receivers switch, the corresponding resources can be released). As a result, it is quite unlikely that a global optimization process leads to better resource usage in any part of the tree.

Consider now the new Cooperative QoS Management. Here, the information on which qualities are received by which users is *localized.* Each router agent with a distance of one to a leaf in the multicast tree knows for his usually small (in terms of the number of receivers) region which receivers get which quality. Since the region is small, it is much easier to decide if it makes sense to try to convince receivers. If, of 20 receivers, two receive quality q_1 and 18 receive quality q_2, it will be much more likely that the two receivers accept to switch to quality q_2 than in a global situation as described above where of, say, 50,000 receivers, 45,000 receive quality q_1 and 5000 quality q_2. If a cooperative agent convinces all his receivers to switch, the resources for quality q_1 can be released on his subtree. This does not necessarily influence other parts of the multicast tree. Thus, such agents initiate local rather than global optimization processes which are much more likely to be successful. The process of local optimization is shown in Figure 2 for an example with six receivers two of which receive quality q_1 and four receive quality q_2 (Part (a)). Agent a_1 decides to convince receivers r_1 and r_2, and if it is successful, all receivers in his subtree will receive q_2 (Part (b)). As a result, the application needs less buffer space on the router of a_1 and less bandwidth on the upstream link.

In order to improve the local optimization process, downstream agents can use the information they get from upstream agents in the decision of which quality is the one to favor from a more global viewpoint. Fig. 3(a) shows a typical situation: agent a_1 (the upstream agent of a_2, a_3 and a_4) knows that its router distributes quality q_1 to the routers of a_2, a_3 and a_4, but q_2 only to the router of a_2. From its viewpoint, it is desirable to convince all receivers attached to the router of a_2 to switch to q_1. It communicates this information to a_2. A_2 may now decide whether it tries to convince all receivers to switch

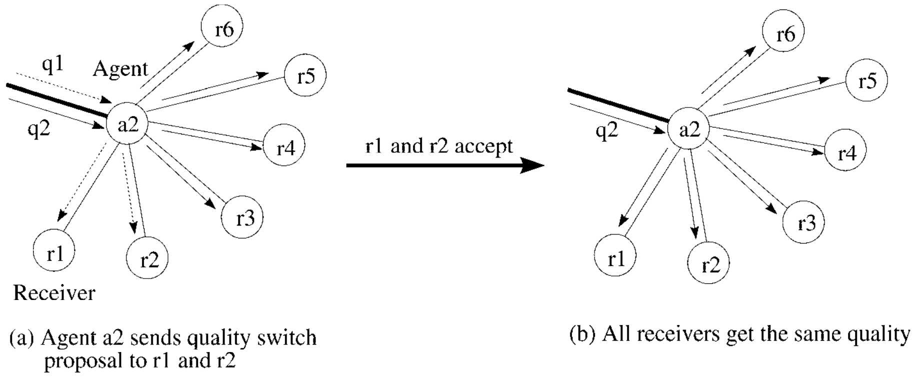

Figure 2: The local optimization process

to q_1, or if it rather is interested in a local optimization which seems to be more likely to be successful here. This flexibility of an agent is due to its autonomy. Fig. 3(b) shows the results in case the decision to optimize locally (upper picture) and "regionally" (lower picture) leads to a success. From the global viewpoint, the regional optimization is better, but it is less likely that it is successful. Obviously, such a decision is very complex since it depends on many parameters. One further parameter, for instance, which has not been mentioned so far, is the fact that some qualities may not be supportable on dwonstream links. Asking receivers to switch to such a quality makes thus no sense. To support agents in the decision process, several ways to find strategies have been developed [3].

4 Making quality switching attractive

The question now is why a receiver should be interested in switching the quality and why it is also interesting for the communication service provider to support the optimization process.

For the service provider, the answer is quite simple. The resources that have been released by means of a quality switch of several receivers can be used for other applications. In some parts of the network, such a release will even enable the start of new applications or the participation of new users, since without it, there may have not been sufficient resources available. As a result, the CSP will be able to serve more applications and more users, which in turn will result in additional revenue. The more unified a multicast tree of a given application with respect to supported qualities is, the better will be the resource utilization and the possible additional revenue.

In order to motivate a receiver to switch to another quality (in fact, the user

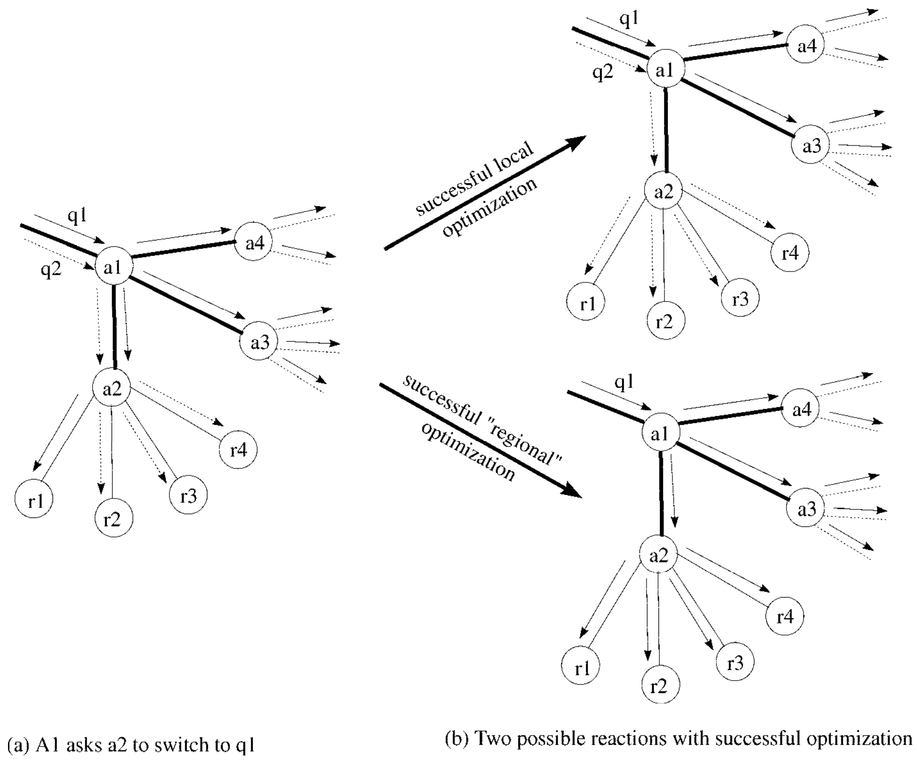

Figure 3: Agent cooperation to globalize the optimization process

is satisfied with the current quality-cost relation since he accepted it during negotiation), there has to be some *reward*. Since quality is strongly related to cost, the reward should be some *monetary* advantage. The money necessary to finance such a reward can be generated from the advantage a CSP gains from releasing resources and using them for additional applications and users. This also depends on the question whether a switch from high to low or from low to high quality is executed. "Low to high" normally leads to a certain increase in copyright (to be paid to the information provider) and transmission cost (to be paid to the CSP), and vice versa in the "high to low" case. The general question is how the additional value generated by the CSP should be distributed among the participants. Possible recipients are

- *all receivers that switch quality*
  Their reward has to be relatively high because without their action, there will be no additional revenue at all. A reward in the "low to high" may thus consist of a relatively low increase of the overall cost, while in the "high to low" case, the reward may be a relatively strong decrease of the

cost.

- *all other receivers*
  It may be fair to let the other receivers participate, but it is not necessary since they accepted their current settings for quality and cost and do not know that some cost will decrease. In a market-driven environment, their share will probably be zero. As a result, there are receivers (those who switch) who pay (much) less for a certain quality than others. A real-world example comparable to this are "last-minute" offers for vacation packages which are usually much less expensive than regularly booked packages. If one traveller accepts such an offer, that does not mean that others who already booked will get some money back.

- *the CSP*
  Actually, the provider has the opportunity to control the switching process by asking a certain price for the communication cost for the new quality. He has to find a solution between a fair share for himself and a good enough reward for the receivers to switch qualities. The more receivers have to be convinced, the less reward will be available per single receiver. In order to make an attractive offer, the CSP may have to reduce his own reward to some extent.

- *the information provider*
  The information provider will often be interested in more receivers selecting a higher quality since such a stream usually generates more revenue without further cost (if it is sent anyway). Probably, he will make switching from low to high quality more attractive by not asking the full price for the high quality stream. On the other hand, he may try to avoid receiver switching from high to low since this will decrease his revenue. His advantage is that he controls the QoS agents and can define their strategies.

Obviously, the players in this scenario may have conflicting goals. Market laws will decide which of the players can realize his goals to the highest degree. Receivers may not switch if the reward is not high enough, or the information provider may instruct his agents not to offer a switch from high to low quality, if the CSP does not offer him a certain compensation. Typically, a compromise will be found which is acceptable for all players.

The following example illustrates this discussion: assume that a source offers a stream in two qualities q_1 (high) and q_2 (low). Copyright cost for Q_1 is 8 currency units (CU) and 5 CU for q_2. Transmission cost is 4 CU for q_1 and 2 CU for q_2, since q_1 needs twice as much resource units as q_2 (say 4 and 2 resource units, respectively). The overall "normal" price to be paid for a stream in agiven quality then is 12 CU for q_1 and 7 CU for q_2. At a point in time t_1, two receivers get the q_2-stream and 10 get the q_1-stream. The agent now tries to convince the q_2-receivers to switch. This will be a switch from low to high. The CSP estimates that, with 2 resource units being released in case of success, he can serve three more users of another application and can probably generate 9 additional CU. He is ready to spend 6 CU as a reward for switching receivers which means 3 CU per receiver. The information provider does not give any rebate and still asks for 8 CU for the higher quality. With this information, the agent makes the following offer to both q_2-receivers: If you switch to q_1, you just pay 9 CU (8 CU for the information, $4 - 3 = 1$ CU for the transmission) instead of your current 7 CU.

It should be noted that there is a risk that, even if the switch of one receiver is successful, the result may not be a release of resources, since other receivers may not accept the quality switch. In order to be able to convince receivers to switch, CSPs have to accept this risk and guarantee a reward even if not all receivers switch, because a receiver will probably not switch if he cannot be sure to receive a reward. One way to minimize the risk is to make improve the offer to receivers who do not accept the given one.

5 Conclusion

In this paper, we described a QoS management system, namely Cooperative QoS Management, that is especially well-suited to support certain large-scale electronic commerce applications. After a short description of the whole approach, we concentrated on the QoS renegotiation function and showed that due to its distribution throughout the network and the cooperation between QoS agents and users, it is extremely useful to reduce communication service cost and to improve system performance.

We are currently designing the implementation environment for our approach. Since we need to install our own code on routers in the network, the use of *active networks* [7] seems especially promising. There are several active networks environments available for experiments, see for example[8], and there are already first ideas to integrate popular resource reservation protocols like RSVP [10] and active networks [9]. Our environment will allow to send QoS agents to arbitrary routers in the active network and let them contact local system components like RSVP or QoS-based multicast routing instances as well as neighboring agents [2]. An overview of the prototype can be seen in Fig. 4.

References

[1] C. Aurrecoechea, A. Campbell, and L. Hauw. A Survey of QOS Architectures. *Multimedia Systems Journal, Special Issue on QoS Architectures*, 1998. To appear.

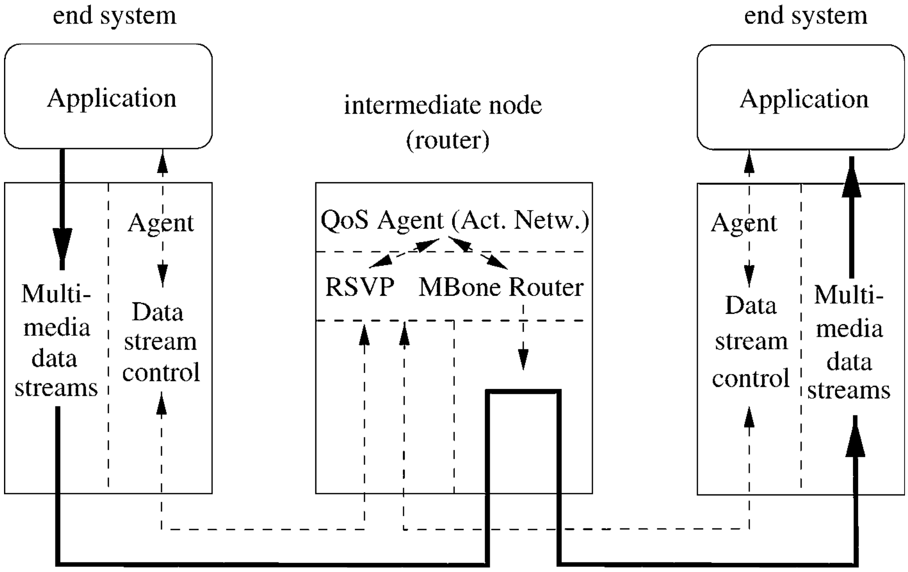

Figure 4: Overview of the implementation design

[2] H. de Meer, A. Puliafito, and O. Tomarchio. Software Agents for QoS Management. *Cybernetics and Systems: An International Journal*, 27(5), 1998. To appear.

[3] S. Fischer and H. de Meer. Decision Support in Cooperative QoS Management. Technical Report 4/97, University of Mannheim, 1997. URL=http://www.informatik.uni-mannheim.de/~ stefis/documents/tr-97-004.ps.gz.

[4] S. Fischer, A. Hafid, G. v. Bochmann, and H. de Meer. Cooperative QoS Management in Multimedia Applications. In *4th IEEE Int. Conf. on Multimedia Computing and Systems (ICMCS'97), Ottawa, Canada*, pages 303–310. IEEE Computer Society Press, June 1997.

[5] A. Hafid and S. Fischer. A multi-agent architecture for cooperative qos management. In *Proc. Int. Conf. on Management of Multimedia Networks and Services (MMNS'97), Montreal, Canada*, July 1997.

[6] N. Shacham. Multipoint communication by hierarchically encoded data. In *IEEE Infocom'92*, pages 2107–2114, 1992.

[7] D. Tennenhouse et al. A Survey of Active Networks Research. *IEEE Communications Magazine*, 35(1):80–86, Jan. 1997.

[8] D. J. Wetherall, J. V. Guttag, and D. L. Tennenhouse. ANTS: A Toolkit for Building and Dynamically Deploying Network Protocols. Submitted to IEEE OpenArch'98, San Francisco. See URL=http://www.tns.lcs.mit.edu/publications/openarch98.html.

[9] R. Wittmann and M. Zitterbart. AMnet: Active Multicast Network. In *Proc. 4th COST237 Workshop 'From Multimedia Services to Network Services', Lisboa, Portugal*, Dec. 1997. To appear.

[10] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala. RSVP: A New Resource Reservation Protocol. *IEEE Network*, 7(5), Sept. 1993.