QoS Routing Protocols for Mobile Ad-hoc Networks – A Survey

Authors:
Philipp Becker

TU KL Technical Report 368/08
BelAmI Deliverable No. D2.5.1

Date: 6. August 2007
Version: 1.0
Status: Final
Classification: Public

A publication by University of Kaiserslautern

Fraunhofer IESE is an institute of the Fraunhofer Gesellschaft.
The institute transfers innovative software development techniques, methods and tools into industrial practice, assists companies in building software competencies customized to their needs, and helps them to establish a competitive market position.

Fraunhofer IESE is directed by
Prof. Dr. Dieter Rombach (Executive Director)
Prof. Dr. Peter Liggesmeyer (Director)
Sauerwiesen 6
67661 Kaiserslautern

© 2007 Fraunhofer IESE and TU Kaiserslautern.
All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means including, without limitation, photocopying, recording, or otherwise, without the prior written permission of the publisher. Written permission is not needed if this publication is distributed for non-commercial purposes.

Abstract

The provision of quality-of-service (QoS) on the network layer is a major challenge in communication networks. This applies particularly to mobile ad-hoc networks (MANETs) in the area of Ambient Intelligence (AmI), especiallay with the increasing use of delay and bandwidth sensitive applications. The focus of this survey lies on the classification and analysis of selected QoS routing protocols in the domain of mobile ad-hoc networks. Each protocol is briefly described and assessed, and the results are summarized in multiple tables.

Keywords: Quality-of-Service; QoS; ad-hoc; wireless networks; mobile networks; MANET; routing protocols; QoS routing; survey

© 2005 Fraunhofer IESE and TU Kaiserslautern

Technical Report 368/08

QoS Routing Protocols for Mobile Ad-hoc Networks – A Survey

Philipp Becker
Computer Science Department
University of Kaiserslautern
D-67653 Kaiserslautern, Germany
pbecker@informatik.uni-kl.de

1 Introduction

Mobile ad-hoc networks (MANETs) denote wireless networks that can form spontaneously as soon as multiple wireless nodes are in transmission range. Mobile nodes can join and leave or change their position inside the network, so its topology can change anytime in unpredictable ways. Another fundamental property is the absence of a centralized control to manage and assign resources. In addition, routing protocols in wireless networks have to cope with problems like the exposed and hidden terminal problem [1] or the usage of a shared medium, which can lead to frame collisions. Examples for mobile ad-hoc networks are ZigBee [2, 3] and Bluetooth [4] networks.

Routing is one of the core problems for data exchange between nodes in networks. In recent years, both the areas of providing quality-of-service and routing in mobile ad-hoc networks have massively increased in importance. Many routing protocols for wireless networks, e.g. AODV [5] or DSR [6], use best-effort routing, where all nodes within range compete for the shared medium. No guarantees or predictions can be given here on when a node is allowed to send. For quality-of-service (QoS) routing, it is not sufficient to only find a route from a source to one or multiple destinations. This route also has to satisfy one or more QoS constraints, mostly, but not limited to, bandwidth or delay. To guarantee these constraints after a route was found, resource reservations on the participating nodes are made.

Especially in the area of Ambient Intelligence (AmI) [7, 8], aiming for the improvement of everyday life activities through the application of additional computing devices, both mobile ad-hoc networks and support for QoS are often used in combination. In many cases, nodes in these networks can only be connected wirelessly because of their mobile character (wearable sensors, computers embedded in objects of everyday life, etc.). As the use of delay and bandwidth sensitive applications (e.g. voice or video streams) increases, so does the need for QoS routing protocols in MANETs.

Providing QoS in mobile ad-hoc networks is much more difficult than in most other types of network. First of all, because of the nature of radio links, reservations on links can influence each other in a 2-hop range and thus complicate the computation and management of bandwidth and delay restrictions. Additionally, even with reservations, resource availability cannot always be guaranteed due to the dynamic aspect of the network. In [9], this is denoted as *Soft QoS*. Protocols for QoS routing in MANETs have to take care of these problems.

This survey takes a closer look at selected QoS routing protocols for mobile ad-hoc networks with focus on their possible use in AmI networks.

The rest of this paper is organized as follows: In Section 2, information is given on how the surveyed protocols are classified. Section 3 contains the evaluation of the considered routing protocols, and Section 4 presents conclusions.

2 Classification

To compare different routing protocols, to show their strengths and weaknesses, common criteria have to be chosen. The protocols are then classified on the basis of these criteria. This section introduces the criteria used in Section 3.

Addressing Addressing defines what destination nodes will receive a packet sent out by a source node. *Unicast* means that exactly one destination node is addressed from a source node. To support unicast, a routing protocol discovers a path or multi-path between two nodes. With *multicast*, multiple nodes may be addressed. Here, a routing protocol discovers, e.g., a tree, with network nodes as tree nodes, network links between nodes as edges, the source node as root and the destination nodes as leaves, if they are not part of the route to another node in the same tree. The domain of multicast routing gained more importance during the last few years, partly because of the growing application of audio and video streams.

Broadcast means that a packet is addressed to all the nodes in a network, often realized by flooding, where each node repeats the packet. In wireless networks, this often leads to frame collisions and many unnecessary transmissions. Therefore, selective flooding tries to reduce the number of sent packets by just letting a subset of nodes forward the packet, such that every node in the network is in 1-hop range to one of these nodes.

Geocast addressing tries to reach all nodes within a certain geographical area, which makes it a special form of multicast addressing. *Anycast* addresses an arbitrary node from a group of destination nodes.

Prerequisites While the specification of some routing protocols also includes certain functionality like, e.g., resource reservation, others assume the existence of mechanisms to handle these tasks. These prerequisites have to be identified so that additional efforts to use a certain routing protocol can be estimated.

Quality-of-Service Quality-of-Service (QoS) in computer networks refers to the provision of guaranteed service on the networking layer, defined in form of performance contracts between application and service provider. To negotiate such a contract, the application defines QoS requirements that contain sufficient information about the required type and level of service.

Metrics To specify QoS requirements, metrics are needed. In networking, a metric associates a numerical value with a route, so different routes can be compared. QoS metrics can be divided into different groups, e.g. *additive* metrics, *multiplicative* metrics, and *concave* metrics. Let $d(n_i, n_j)$ be a metric for link (n_i, n_j) and

$p = (n_1, n_2, \ldots, n_m)$ be a path between nodes n_1 and n_m. Then the named metrics are defined as follows:

$$\begin{aligned} &\text{Additive:} && d(p) = d(n_1, n_2) + d(n_2, n_3) + \ldots + d(n_{m-1}, n_m) && (1) \\ &\text{Multiplicative:} && d(p) = d(n_1, n_2) \times d(n_2, n_3) \times \ldots \times d(n_{m-1}, n_m) && (2) \\ &\text{Concave:} && d(p) = \min\left(d(n_1, n_2), d(n_2, n_3), \ldots, d(n_{m-1}, n_m)\right) && (3) \end{aligned}$$

The most commonly used metrics in QoS networks are *bandwidth* and *delay*. Bandwidth (concave) denotes the bandwidth along a certain path and is limited by the link with the lowest bandwidth along this path. Delay (additive) indicates the time between sending out a packet from the source node and reception of this packet at the destination node. The metric *cost* does not belong to any of the groups above, as it is more abstract and can be defined by any function.

Besides these metrics, there are other interesting metrics for QoS networks. The *number of hops* (additive) represents the number of links in a path. *Jitter* (additive) denotes the variation between expected and actual reception time of a packet. *Energy* (additive) takes the energy needed to send a packet from source to destination into account. Alternatively, energy can also be handled as a concave metric, where a (mobile) node has to provide a certain energy level, to be considered as part of a route. *Loss probability* (multiplicative) refers to the probability of a packet to be lost on its way to the destination node, e.g. because of collisions, topology changes or weak radio signals. Further QoS metrics include, e.g., *signal strength* (concave) and *distance* (additive).

QoS routing protocols utilize subsets of these metrics. In many cases, only single metrics like bandwidth or delay or specific groups of metrics, e.g. additive metrics, are taken into account.

Constraints A QoS constraint is a lower or upper numerical bound referring to a QoS metric. If a path is feasible with respect to a QoS constraint, this means that the path's value regarding the chosen metric does not cross the given boundary.

This criterion refers to whether a QoS routing protocol is capable of finding a route satisfying a *single* QoS constraint only (even if the protocol allows the metric used for the constraint to be chosen from a set of metrics), or if it can take *multiple* constraints into account at the same time. Finding an optimal route that satisfies multiple constraints simultaneously is inherently hard and of complexity NP [10]. Therefore, most routing algorithms that consider multiple constraints do not try to find the optimal path but rather any path satisfying all constraints.

Reservations Guarantees for satisfaction of QoS constraints along a route can only be given if resources are reserved along this route. This classification criterion indicates whether a QoS routing protocol just determines a feasible route (*no*) or also takes reservations into account (*yes*), by providing own reservation functionality or by using other protocols, e.g. RSVP [11]. Of course, in wireless networks, the compliance of these guarantees also depends on the stability of the routes and the dynamics of the network topology.

Link properties Some routing protocols require *bidirectional* links. Two nodes a and b are linked bidirectionally, if there exist two *unidirectional* links between them, (a, b)

and (b, a). If a routing protocol relies on bidirectional links, this often means that either if a feasible path was found, the same path is used for backward communication, e.g. for confirmation of a path, or that the reception of packets has to be acknowledged.

Net state determination The term *net state* can cover topology information about the whole network or part of it, e.g. about all nodes in 1-hop range. This may include geographical information about a node's position and topology information about links between nodes, combined with information about QoS metrics for nodes or links. Many routing protocols need part of this information to determine a feasible route for given constraints.

> **Communication complexity** The communication complexity relates to the number of messages that need to be exchanged in a network consisting of n nodes in a worst case scenario, so that every node has up-to-date information. This premises a static topology, as the information will most probably never be up-to-date for all nodes in a network with high dynamics.

> **Packet size** This denotes the amount of information that is exchanged per packet to update other nodes in a worst case scenario. For example, if packets with net state information should include a full list of a node's 1-hop neighbors, the packet size complexity would be $O(n)$ for a network with n nodes, because they could be all in range of each other. $O(1)$ denotes a fixed packet size, independent of, e.g., the number of nodes in the network.
>
> Together with the communication complexity, information about packet size allows the estimation of an upper bound of data that has to be exchanged between the nodes to update net state information.

> **Storage Complexity** This denotes the amount of memory necessary to store net state information in a worst case scenario. This value can not be estimated by means of communication complexity and packet size, because received information does not necessarily need to be stored completely and storage complexity may also cover information gathered locally.

Route discovery A route between two nodes consists of a list of nodes $(n_1, n_2, \ldots, n_m)$, $m \geq 2$, where n_1 denotes the source node, n_m denotes the destination node, and a link exists between each two adjacent nodes in the list. Using this route, each packet from node n_1 with node n_m as destination will be sent to node n_2, which itself will send the packet to node n_3 and so on, until it reaches node n_m. These routes have to be discovered, either in advance or while sending the packet.

> **Routing Type** There exist different strategies for route discovery in routing protocols. For *source routing*, the source node determines the route a packet will take on its own; for that, the node needs sufficient knowledge about the network's topology. While this is no problem in small- to middle-sized networks with static topology or low dynamics, it is in most cases not suitable for MANETs with higher dynamics due to scarce bandwidth for exchange of topology information and long propagation

times. Here, often, *distributed routing* on a hop-per-hop basis is used. This means that each intermediate node decides which of its 1-hop neighbors should receive the packet. This is not limited to sending the packet to exactly one neighbor, as different approaches may also flood the network or split up the route to increase the chance of successful delivery. A type of routing often used in large networks is *hierarchical routing*. Here, the complexity of the routing problem is reduced by dividing a network into a hierarchy of smaller networks, where each level is responsible for its own routing (divide and conquer-paradigm).

Most protocols surveyed in this paper discover the routes *reactively*, i.e. route discovery is done when a route is needed. If feasible routes are determined in advance, this is called *proactive* route discovery. While this method has the advantage that routes are already present when needed, it has severe drawbacks in mobile networks. Due to the dynamics of the network topology and long propagation times, the chance of outdated topology information and broken routes is too high to efficiently determine routes in advance. Some protocols use a *hybrid* approach, combining elements of both proactive and reactive methods.

Some protocols try to satisfy the QoS requirements (e.g. bandwidth) by finding a *multi-path* between source and destination node A multi-path denotes a path between two nodes that may split up and optionally reunite. This should not be mistaken for multicast routing, which will be addressed in the next paragraph. Depending on the QoS metrics in use, it is also possible to split a QoS constraint into different subconstraints. Fig. 1 gives an example of multi-path routing. There exists no path between nodes s and t that can satisfy a bandwidth constraint of, e.g., 4, so the bandwidth constraint and the path are split up at node n_2. As both paths meet at node n_5, they join again.

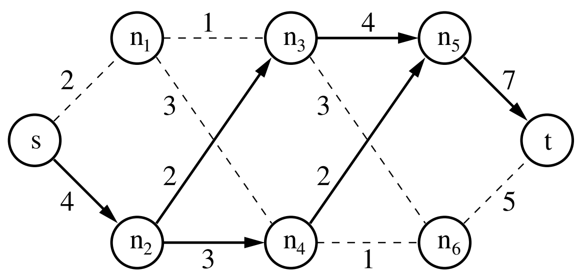

Figure 1: Example for multi-path routing with split up bandwidth requirement of 4

Communication complexity Communication complexity indicates the number of messages needed to discover a feasible route in a worst case scenario, if such a route exists. For distributed route discovery, the upper bound cannot be lower than $O(n)$, with n indicating the number of nodes in the network.

Packet size This denotes the upper bound for packet size during route discovery, and is quite similar to the packet size during net state determination.

Robustness Robustness describes the ability of a routing protocol to cope with unstable situations. Main problems in MANETs that have to be dealt with include outdated information about topology and resources, and broken routes due to mobility.

As mentioned before, a node's information about the state of the network might never be up to date, even at the moment an update by an adjacent node is received. So some routing protocols do not use the received data, but rather rely on *imprecise information* about the network. For example, imprecise information can be gathered by monitoring the behavior of other nodes and trying to predict how these nodes will behave in the near future.

If a route is broken, e.g. due to moving nodes, a new route has to be established. This can either be done by starting a new route discovery process at the source node or by initiating a *route repair* operation, if this is supported by the routing protocol. The approach for repairing a broken route may differ, depending on the routing protocol. One possible approach would be to bridge the broken links, so that the remaining path can still be used. Another possibility is to search for a new partial route from the last working node prior to the broken link to the destination node. Both possibilities are shown in Fig. 2.

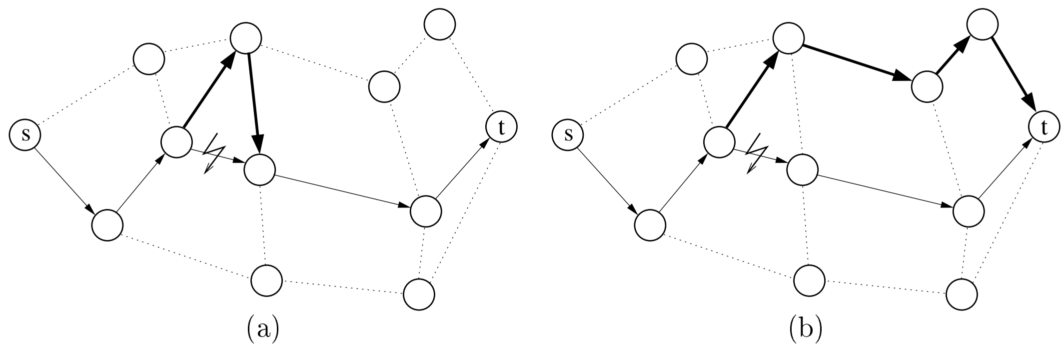

Figure 2: (a) bridging of a broken link, (b) new partial route to destination t

By keeping *secondary paths* in reserve, which also connect source and destination node, routing protocols can drastically shorten the time to reconnect these nodes in case of a broken route. Of course, this requires that the broken links must not be part of the secondary paths. If resources are already reserved for secondary paths, they are wasted as long as the secondary path is not in use. Without reservations, it is possible that a secondary path cannot be used due to missing resources when the path is needed.

If a route is broken, resources are still reserved along the route on both sides of the broken links. While most QoS routing protocols surveyed here use some sort of timer during route discovery to reserve resources only temporarily until a route is confirmed, some of them use a *route timer* for established routes as well. As a consequence, if a specific route is not used for a predefined amount of time, it is discarded and its reserved resources are released. Some protocols even abstain from a special procedure to free resources explicitly and only rely on timer mechanisms. Especially with concave QoS requirements like bandwidth, the use of *multi-path* routes can bring advantages in networks with very scarce resources. In addition, multi-path routing may increase robustness through overreservations.

Scalability This indicates whether a routing protocol can still be used efficiently with increased network size. Especially protocols with high complexity can experience difficulties with growing network size. As values for this criterion, we use *low*, *medium*, and *high*.

Suitable for ad-hoc Each of the surveyed protocols is given a rating on whether it is suitable for application in mobile ad-hoc networks or not. This rating results from the classification of a protocol and its analysis, and can be either *yes* or *limited*.

Performance Assessments To assess a routing protocol, additional information beyond the protocols description and its algorithms is needed. This can be a theoretical *analysis* of a protocols complexity, a *simulation* (or series of simulations) of the protocols behavior in a simulated environment, or an *implementation* of the routing protocol used on real hardware.

Other Everything that does not fit into one of the other categories, but should be included nevertheless, is listed here.

3 Survey and classification of QoS routing protocols

In this section, selected QoS routing protocols for MANETs are surveyed. For each protocol, the functionality and main features are described briefly, followed by an assessment. The results are summarized in Tables 1 to 3, using the criteria described in Section 2. Each protocol is referred by its name and a reference to the paper where it was defined. If no explicit name was given to a protocol, the authors are named instead.

Besides this survey, there also exist other surveys covering the same topic or at least part of it. In [12], Chen and Nahrstedt discuss QoS routing problems in general and present a number of different routing algorithms. Khetrapal [13] concentrates on routing protocols for MANETs without QoS support. In [14], Guimarães et al. give a short overview over some existing QoS routing mechanisms and protocols in MANETs. Kumar et al. [15] survey MAC protocols for ad-hoc networks, including protocols with QoS support.

Ticket-Based Probing (TBP) [9] TBP tries to solve the delay- and bandwidth-constrained least-cost routing problems using imprecise state information. The source node sends out a number of probes to some of its neighboring nodes to discover a feasible path. Each probe contains at least one *ticket*. Nodes may split up a single probe into multiple probes, distributing the received tickets among these new probes. Therefore, the total number of tickets used for path discovery is constant. The information used to distribute the tickets among a node's neighbors is based on the up-to-date information about the links to adjacent nodes as well as on aggregated information about the neighbors' view of the net concerning bandwidth, delay, and cost, which may be less accurate.

The maximum number of probes at any time and the number of paths searched is bounded by the number of tickets. Thus, the routing overhead can be controlled, giving the choice to use more tickets to increase the chance of finding a feasible path.

As a disadvantage, each node has to keep the complete state information for each of its neighbors. So the memory requirements do not only depend on the number of nodes in the network, but also on the number of a node's neighbors.

Forward algorithm (FA) [16] In ad-hoc networks based on time division multiple access (TDMA), the forward algorithm tries to find a feasible solution for bandwidth constrained routing. For route discovery, it uses a modified version of the AODV [5] routing protocol, where additional information is appended to each routing packet. During route discovery, the set of time slots that each link along the route should use is determined. To measure the bandwidth of a path, the algorithm does not search for a global maximum which would be NP-complete [17], but does instead try to find a suboptimal solution by calculating local maxima for adjacent links and to propagate this calculation during route discovery along the path to the destination. Paths with shorter delay are favored over those with more bandwidth, as long as the bandwidth constraint is still satisfied.

Advantages are that, as in AODV, nearly no state information has to be stored by the nodes except for active routes, and that there is no proactive detection of the net state, resulting in fewer control messages. FA is not limited to using AODV, for route discovery, but could also use other on-demand routing protocols like, e.g., DSR [6] or TORA [18]. Also included is a route repair mechanism where parts of the old route can be reused if not already timed out.

On the other side, in the worst case while using AODV, a route request can result in flooding the entire network, generating much communication overhead. Therefore, the main application area of FA is restricted to small networks with low network mobility.

Adaptive Proportional Routing (APR) [19] APR abstains from the exchange of QoS state information among routers and uses only locally gathered information. For each source-destination-pair of nodes, one or multiple explicit-routed paths have to be set up in advance, e.g. with MPLS [20]. These are the candidate paths for routing. The maximum capacity for each path is known to the routers. Each flow routed along a candidate path has a certain probability of being blocked. By knowing the capacity and measuring the blocking rates when trying to route a flow along a path, a *virtual capacity* for this path is computed. This capacity may change over time, as new local information is gathered about current blocking rates. APR tries to equally distribute the flows among the available paths w.r.t. the virtual capacity of each path and with preference of minhop (i.e. shortest) paths over alternative (i.e. non-minhop) paths.

No QoS information is exchanged between the nodes, reducing protocol overhead. Core routers (i.e. non-source routers) do not need to keep and update any QoS state database necessary for global QoS routing since the paths are already defined and no reservations are made.

On the other side, APR is not suitable for mobile networks, as paths have to be set up in advance. Additionally, no hard QoS guarantees are possible, since no reservations are made.

Liao et al. 01 protocol [21] Liao et al. propose a routing protocol that tries to detect a multi-path route to a destination node to fulfill a bandwidth requirement. For that, it uses a scheme of sending out probing packets with tickets, similar to that of TBP. But, other than in TBP, this protocol is based on an on-demand manner, so no global link state information has to be collected in advance, and single tickets may be split up into sub-tickets, each trying to find a path with lower bandwidth requirement. The destination node will pick one ticket or a set of sub-tickets forming a whole ticket and send a reply to the source node, confirming the bandwidth reservations.

This protocol may find routes satisfying the bandwidth requirement even if no single path exists with sufficient bandwidth. But it relies on the existence of multiple transceivers per hop to effectively avoid collisions. The number of split-ups per ticket is not limited. This may help in discovering feasible routes even if there is a large number of links with narrow bandwidth. On the other hand, this may also result in a very large communication overhead, not only during route discovery, but during normal data transmission as well.

QoS-aware Multicast Routing Protocol (QMPR) [22] The goal of QMPR is to reduce the communication overhead when constructing a multicast tree by switching between single-path routing and multi-path routing. When a node n wants to join an already existing multicast tree, a single path to the tree's core is searched using a unicast routing algorithm. During route discovery, the QoS constraint is checked at every intermediate node. Consider two intermediate nodes a and b with a being part of the already discovered path. If b is the next node chosen by the unicast algorithm, but the link (a, b) violates the QoS constraint, then instead a would send messages to its other neighboring nodes to split up the search process. If more than one feasible path is detected, a chooses the best one (e.g. by smallest number of hops). The number of split ups can be restricted by specifying a maximum branching level.

Compared to other QoS multicast routing protocols like the spanning join protocol by Carlberg and Crowcroft [23] or QoSMIC [24] by Faloutsos et al., QMPR avoids flooding to reduce the communication overhead.

QMPR was not explicitly designed for MANETs, so it does not take mobility into account. However, because of its high-level design, it can be used on top of arbitrary unicast routing protocols, so it can be used in MANETs nevertheless.

Liao et al. 02 protocol [25] Liao et al. devised a routing protocol for MANETs to reserve bandwidth in a time-framed medium while solving the hidden- and exposed-terminal problems. Each node keeps several tables of information, e.g. about the time slots of all nodes within a 2-hop range and their current usage (send, receive, free). This information is used to find free slots when reserving bandwidth and avoid the hidden- and exposed-terminal problems. To find a feasible route, a route request packet is sent out that includes, among other things, a list of 1-hop neighbors that may rebroadcast this request, if they have sufficient collision-free time slots. Time slot reservation is done during route acknowledgment on the way back to the source node.

This routing protocol is rather simple and can be implemented with low effort. Additionally, it avoids the hidden- and exposed-terminal problems.

But as the memory requirements of this protocol are rather high, it is suitable only for smaller networks. Additionally, a route request may also result in flooding the entire network. The chance of flooding the network increases with the number of free time slots at each node.

Lin-Liu protocol [26] This routing protocol is based on DSDV [27] on a time-slotted medium, but with support for reservations and bandwidth calculation. Each time frame, consisting of a fixed number of slots, is divided into two phases, the control phase and the data phase. The information exchanged by DSDV is broadcasted in a predefined time slot during the control phase and extended by bandwidth information, that consists of listings of free slots for a specific node. With this additional information, the end-to-end bandwidth for a specific path can be calculated and used to determine a feasible path. Along this path, bandwidth is reserved in form of specific slots. If the bandwidth requirement cannot be fulfilled at any participating node during reservation, a RESET message is sent back to free already reserved slots hop-by-hop. To weaken the consequences of broken paths, secondary routes are maintained that can be used immediately.

This protocol works on basis of another well-know protocol and avoids the hidden-terminal problem. Besides that, it has the same disadvantages as DSDV as, e.g., large overhead and long propagation time.

Optimally partitioned most probable path (OP-MP) [28] Here, the delay constrained routing problem consists of a combination of two other problems, MP (most probable path) and OP (optimally partitioned path). Problem MP is about finding the most probably satisfying a given delay constraint D. Because this problem is NP-hard, it is combined with problem OP, that wants to find an optimal partition of a given path p. It is assumed that net state information in MANETs cannot be accurate due to dynamics. Let the network be modeled as a graph $G(V, E)$. Then for each link $l \in E$, there exists a function $f_l(d)$ that returns the probability that l can guarantee a delay bound d. Other than the net state, these functions $f_l(d)$ do not change rapidly, rather the dynamics are incorporated in the probabilities. It is assumed that delays on the links are independent and all functions $\{f_l(d)\}_{l \in E}$ are known. Information for these has to be gathered and exchanged independently from this protocol.

While this protocol uses a novel principle, it lacks helpful mechanisms for application in MANETs like route repair. Also, there is neither any form of analysis nor are there any simulations of this protocol to support its assessment.

Ad hoc QoS on-demand routing (AQOR) [29] This protocol uses limited flooding to discover the best route available in terms of smallest end-to-end delay with bandwidth guarantee. A route request packet includes both bandwidth and end-to-end delay constraints. Let T_{max} denote the delay constraint. If a node can satisfy both constraints, it will rebroadcast the request to the next hop and switch to *explored* status for a short period of $2\,T_{max}$. If multiple request packets arrive at the destination, it will send back a reply packet along each of these routes. Intermediate nodes will only forward the reply, if they are still in *explored* state. However, the bandwidth reservation for each flow is only activated by the arrival of the first data packet from the source node. Delay is measured during route discovery. The route with the least delay is chosen by the source.

No mechanism for connection tear-down is needed or integrated, since all reservations are only temporary. Timers are reset every time a route is used. So there is an upper time bound after which broken routes are detected. To further reduce communication overhead during route discovery, AQOR can work with some location aided routing protocols [30].

For delay violation detection, the estimated time offset between the system clocks of source and destination node has to be known.

Chen-Heinzelman protocol [31] This protocol tries to provide soft QoS or better than best-effort service for a bandwidth constraint, rather than guaranteed hard QoS. It utilizes two kinds of schemes, a *feedback scheme* and an *admission scheme*. While the admission scheme searches for a route satisfying the bandwidth constraint, the feedback scheme updates the constraint if a node does not have enough residual bandwidth and returns the final value to the application. The latter can be used when applications can scale the data they transmit to meet the channel conditions, e.g. by reducing quality of video streams. The route discovery function of this protocol is based on AODV [5] with a modified packet format. The residual bandwidth can be either estimated by listening to the channel and the ratio of free and busy times, or by appending a node's current bandwidth and that of its 1-hop neighbors to AODV's periodic *hello*-messages. AODV's route request (RREQ) packets include additional information about the used scheme and either the bandwidth constraint (admission scheme) or the minimum of bandwidth constraint and detected bandwidth on the partial path (feedback scheme).

This protocol has all the advantages and disadvantages of AODV. Additionally, it is QoS-aware, but does not give any QoS guarantees, as no bandwidth is reserved for a route.

Its feedback scheme adapts to the available bandwidth and gives feedback to applications. Both adaptation and feedback are only done once, during route discovery. If any changes occur for an already discovered route, no further feedback is given or adaptation is made.

Table 1: Classification of routing protocols – Capability and prerequisites

Protocol	**Addressing**	**Prere-quisites**	**Quality of Service**			**Link Properties**
			Metrics	**Con-straints**	**Reser-vations**	
TBP [9]	Unicast	P1, P2	Delay, Bandwidth, Cost	Multi	yes	Bidirectional
FA [16]	Unicast	P3, P4	Bandwidth	Single	yes	Bidirectional
APR [19]	Unicast	P5, P6	Bandwidth	Single	no	no info given
Liao01 [21]	Unicast	P3, P7	Bandwidth	Single	yes	Bidirectional
QMPR [22]	Multicast	P2, P8	non-additive metrics	Multi	no	Bidirectional
Liao02 [25]	Unicast	P3, P9	Bandwidth	Single	yes	Bidirectional
Lin-Liu [26]	Unicast	P3, P10	Bandwidth	Single	yes	Bidirectional
OP-MP [28]	Unicast	P11	Delay, increasing Cost[a]	Single	no	no info given
AQOR [29]	Unicast	(P12), P13	Delay, Bandwidth	Multi	yes	Bidirectional
Chen [31]	Unicast	–	Bandwidth	Single	no	Bidirectional

[a] The cost function associated with each link has to increase with the QoS required from it

Prerequisites:

P1	MAC protocol with resource reservation
P2	Resource reservation protocol
P3	Synchronized, time-slotted medium
P4	On-demand routing protocol, e.g. AODV [5]
P5	Explicit-routed paths for each source-dest-pair
P6	Network topology information
P7	Multiple transceivers per host that can work simultaneously
P8	Arbitrary unicast routing protocol
P9	Information about net state and 2-hop neighbors
P10	Global clock or time synchronization mechanism
P11	Information about links and their delay guarantees
P12	Synchronized clocks or information about clock offset
P13	Contention-based medium access mechanism

Observations

Most of the surveyed protocols have some similarities:

- Assumption of bidirectional links:
  All surveyed protocols assume bidirectional links, except OP-MP and APR, where no explicit or implicit info was given for this topic. Especially in case of distributed routing, acknowledgments of discovered routes are often sent back to the source node along the newly discovered routes, sometimes together with resource reservation on the way back (e.g. Liao et al. 01). These mechanisms require bidirectional routes in order to work. Unidirectional links can be used, e.g., in case of source routing, as long as the protocol does not require bidirectional links for means of resource reservation, notification about broken routes, and so on.

Table 2: Classification of routing protocols – Complexity

Protocol	**Net state determination**			**Route discovery**		
	Commun. Complexity	**Packet Size**	**Storage Complexity**	**Routing Type**	**Commun. Complexity**	**Packet Size**
TBP [9]	F1	F1	$O(n^2)$	Distributed, Hybrid	$O(t \times n)$	$O(n)$
FA [16]	AODV	$O(1)$	$O(n \times s)$	Distributed, Reactive	$O(n)$	$O(1)$
APR [19]	not defined	–	$O(p)$	Source, (Reactive), Hierarchical	–	–
Liao01 [21]	$O(n)$	$O(s)$	$O(n \times s)$	Distributed, Reactive, Multi-Path	$O(n \times t \times u)$	$O(t)$
QMPR [22]	F2	F2	F2	Distributed, Reactive	F2, F3	F2
Liao02 [25]	not defined	not defined	$O(n \times \max(n, s))$	Distributed, Reactive	$O(n)$	$O(n \times s)$
Lin-Liu [26]	DSDV	O (DSDV+s)	$O(\text{DSDV} + n \times s)$	Distributed, Reactive	$O(n)$	$O(s)$
OP-MP [28]	not defined	not defined	$O(n^2)$	Source, Reactive	–	–
AQOR [29]	$O(n)$ every second	$O(1)$	$O(n)$	Distributed, Reactive	$O(n)$	$O(1)$
Chen [31]	AODV	$O(1)$	$O(n)$	Distributed, Reactive	$O(n)$	$O(1)$

Footnotes:

F1	Protocol for net state determination (not part of TBP); DSDV-like protocol suggested
F2	depends on unicast routing protocol
F3	depends on # of branchings

Variables:

n:	# nodes
t:	# tickets
s:	# time slots per frame (slotted medium)
p:	# of candidate paths connected to node
u:	max. # of ticket split ups

- Unicast routing:
  Most protocols support only unicast routing. Although there are many different approaches for QoS routing in MANETs, none of these protocols is widely spread, and there still is a lot of research going on in this domain. The development of QoS multicast routing protocols cannot be less complex than the development of QoS unicast routing protocols, as all multicast routing protocols have to consider the case where the routing tree consists of only two nodes. One exception to this is the development of QoS multicast routing protocols that utilize other QoS unicast routing protocols, as seen with QMPR. So multicast routing protocols will most likely have to wait until some progress has been achieved in the domain of QoS unicast routing in MANETs.

- Distributed, reactive routing:
  Most selected protocols follow a distributed, reactive routing approach. Protocols relying on source routing need to have information about the net state that are sufficiently up to date. So source routing has severe disadvantages in MANETs with high dynamics, because of the increased communication overhead to exchange information about net state, even when no routes need to be discovered.

Table 3: Analysis of routing protocols

Protocol	**Robustness**	**Scalability**	**Suitable f. ad-hoc**	**Performance Assessments**	**Other**
TBP [9]	route repair, imprecise inf.	low	yes	Analysis, Simulation	–
FA [16]	imprecise inf., route timer, route repair	low – medium	yes	Analysis, Simulation	–
APR [19]	imprecise inf., local inf.	medium	limited	Analysis	Adaptive
Liao01 [21]	multi-path	medium	yes	Simulation	–
QMPR [22]	–	high	F2	Analysis, Simulation	–
Liao02 [25]	–	low	yes	Simulation	–
Lin-Liu [26]	route repair through secondary paths	low – medium	limited	Simulation	–
OP-MP [28]	imprecise inf.	low	limited	Analysis of computational complexity	–
AQOR [29]	imprecise inf., route repair, route timer	high	yes	Simulation	–
Chen [31]	imprecise inf.	medium	yes	Simulation	Adaptive, Feedback

- QoS metrics are either delay or bandwidth:
  As these are the most commonly used metrics in QoS networks, it can be assumed, that they are widely supported. Additionally, some routing protocols supporting one of these can also be adapted with little effort to use other metrics from the same category, e.g. additive or concave metrics. For example, the delay constraint used in TBP could be replaced by energy or jitter constraints, if the necessary net state information is also available at all nodes.

  But this adaptation is not always possible, as can be seen with AQOR. Here, delay constraints are used to initialize timers to change a node's state for a certain period of time. This principle would not work with, e.g., energy constraints, which also belong to the group of additive metrics.

4 Conclusion

In this paper, selected QoS routing protocols were presented. Each protocol was briefly described, together with some of their advantages and disadvantages. All protocols were classified, with main focus on their use in wireless ad-hoc networks.

While some of the protocols expand widely used routing mechanisms or protocols (e.g. FA [16], Lin-Liu [26], Chen-Heinzelman [31]), others present novel approaches to solving the QoS routing problems in MANETs (e.g. TBP [9], OP-MP [28]).

Although the existence of bidirectional links cannot always be guaranteed in wireless networks, they have a great impact on the complexity of distributed routing protocols. If unidirectional links have to be considered, other mechanisms have to be found for route acknowledgment, route repair and so on. Route acknowledgment could be sent back to the source node via an alternative route, that would have to be discovered as well. Route repair mechanisms initiated by the destination node possibly wouldn't work anymore. So if a sufficient number of bidirectional links in a mobile network can be assumed and at least one of the mechanisms mentioned above is used, the existence of bidirectional links can be made mandatory for a protocol to improve its efficiency and reduce its complexity.

The use of distributed routing will most likely achieve better results than source routing in most cases. In order for source routing to work, up-to-date information about the net state is needed at the source node. In networks with either high dynamics or with many short-lived connections, net state information will change quickly, and thus need to be updated very frequently. Especially in larger networks, due to the high amount of net state information and long propagation time, the chance of outdated net state information at a node increases, and so does the chance of determining a route at the source node that cannot satisfy the given QoS requirements. With distributed routing and depending on the protocols route discovery mechanism, a route may use information about its neighbors within a certain (hop) range, but does not necessarily need net state information for the whole network. So source routing protocols will most likely focus on an efficient way to aggregate and distribute net state information among the network, whereas the communication complexity for distributed routing protocols lies more on efficient distributed route discovery mechanisms.

The question whether a route repair mechanism should be included in a protocol depends on the characteristics of the application domain. In networks with low mobility, i.e. seldom broken routes, or if the route discovery mechanism has a low complexity in means of communication, route repair mechanisms can be omitted for the simplicity of the protocol. If, however, the named conditions do not apply, the inclusion of a route repair mechanism can improve the protocol.

Route timers for already established routes represent an easy way to cope with the problem of releasing reserved resources in case of broken routes. If resources are already reserved in a soft state during route discovery on the way to the destination node, route timers for route discovery as seen in AQOR also make sense. The inclusion of soft state reservations has both advantages and disadvantages. Without soft state reservations, resources needed for a discovered route may have been reserved by a parallel route discovery process. But if a network is flooded for route discovery and every node tries to reserve resources in a soft state for a short time, other route discovery processes may fail during that time, although enough resources are available.

Applications with scalable data can benefit from protocols with feedback mechanisms like Chen-Heinzelman. To improve this principle, periodic feedback for already existing routes could be included, as well as hard lower bounds for QoS metrics. This allows applications to scale their data up or down on the fly, while still having guaranteed resources for their minimal requirements.

Multi-paths as in Liao et al. 01 increase the chance of finding a feasible path for QoS requirements with metrics that can be split up, e.g. bandwidth, in networks with scarce resources. On the other hand, multi-paths can lead to increased communication overhead and also raise the chance of broken (partial) routes, as more nodes are involved in routing.

References

[1] Chunhung Richard Lin and Mario Gerla. Asynchronous Multimedia Multihop Wireless Networks. In *INFOCOM (1)*, pages 118–125, 1997.

[2] J.A. Gutierrez, M. Naeve, E. Callaway, M. Bourgeois, V. Mitter, and B. Heile. Ieee 802.15.4: a developing standard for low-power low-cost wireless personal area networks. *IEEE Network*, 15(5):12–19, 2001.

[3] Patrick Kinney. Zigbee technology: Wireless control that simply works. `http://www.zigbee.org/resources`, October 2003. Whitepaper.

[4] Bluetooth Specification Documents. `http://www.bluetooth.com/Bluetooth/Learn/Technology/Specifications/`.

[5] C. E. Perkins and E. M. Royer. Ad-hoc On-Demand Distance Vector Routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applicatons*, pages 90–100, New Orleans, LA, Feb. 1999.

[6] D. Johnson, D. Maltz, and J. Broch. *DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks*, chapter 5, pages 139–172. Addison-Wesley, 2001.

[7] E. Aarts, R. Harwig, and M. Schuurmanns. Ambient Intelligence. The Invisible Future: The Seamless Integration of Technology into Everyday Life. McGraw Hill, 2002.

[8] IST Advisory Group, Software technologies, embedded systems and distributed systems. A European strategy towards an Ambient Intelligent environment. Report, 2002.

[9] Shigang Chen and Klara Nahrstedt. A Distributed Quality-of-Service Routing in Ad-Hoc Networks. *IEEE Journal on Selected Areas in Communications*, 17(8), August 1999.

[10] Zheng Wang and Jon Crowcroft. Quality-of-Service Routing for Supporting Multimedia Applications. *IEEE Journal of Selected Areas in Communications*, 14(7):1228–1234, 1996.

[11] R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, and S. Jamin. RFC 2205: Resource ReSerVation Protocol (RSVP) — Version 1 Functional Specification, September 1997.

[12] Shigang Chen and Klara Nahrstedt. Distributed Quality-of-Service Routing in High-Speed Networks Based on Selective Probing. In *LCN*, pages 80–89, 1998.

[13] Ankur Khetrapal. Routing Techniques for Mobile Ad Hoc Networks Classification and Qualitative/Quantitative Analysis. In *ICWN*, pages 251–257, 2006.

[14] Rafael Guimarães, Julián Morillo, Llorenc Cerdà, José-M. Barceló, and Jorge García. Quality of Service for Mobile Ad-hoc Networks: an Overview. Technical report, Technical University of Catalonia, Barcelona, Spain, June 2004.

[15] Sunil Kumar, Vineet S. Raghavan, and Jing Deng. Medium Access Control protocols for ad hoc wireless networks: A survey. *Ad Hoc Networks*, 4(3):326–358, 2006.

[16] Chenxi Zhu and M. Scott Corson. QoS routing for mobile ad hoc networks. Technical report, Flarion Technologies, Bedminster, New Jersey 07921, 2002.

[17] Chenxi Zhu. *Medium Access Control and Quality-of-Service Routing for Mobile Ad Hoc Networks*. PhD thesis, Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20906, 2001.

[18] Vincent D. Park and M. Scott Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proceedings of INFOCOM '97*, pages 1405–1413, April 1997.

[19] Srihari Nelakuditi, Zhi-Li Zhang, Rose P. Tsang, and David H. C. Du. Adaptive Proportional Routing: A Localized QoS Routing Approach. *IEEE/ACM Transactions of Networking*, 10(6):790–804, December 2002.

[20] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol Label Switching Architecture, 2001.

[21] Wen-Hwa Liao, Yu-Chee Tseng, Shu-Ling Wang, and Jang-Ping Sheu. A Multi-path QoS Routing Protocol in a Wireless Mobile ad Hoc Network. In *ICN '01: Proceedings of the First International Conference on Networking-Part 2*, pages 158–167, London, UK, 2001. Springer-Verlag.

[22] Shigang Chen, Klara Nahrstedt, and Yuval Shavitt. A QoS-Aware Multicast Routing Protocol. In *INFOCOM (3)*, pages 1594–1603, 2000.

[23] Ken Carlberg and Jon Crowcroft. Building shared trees using a one-to-many joining mechanism. *SIGCOMM Comput. Commun. Rev.*, 27(1):5–11, 1997.

[24] Michalis Faloutsos, Anindo Banerjea, and Rajesh Pankaj. QoSMIC: Quality of Service Sensitive Multicast Internet Protocol. In *SIGCOMM*, pages 144–153, 1998.

[25] Wen-Hwa Liao, Yu-Chee Tseng, and Kuei-Ping Shih. A TDMA-based Bandwidth Reservation Protocol for QoS Routing in a Wireless Mobile Ad Hoc Network. In *Proceedings of IEEE ICC*, April–May 2002.

[26] Chunhung Richard Lin and Jain-Shing Liu. QoS Routing in Ad Hoc Wireless Networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1426–1438, August 1999.

[27] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proc. ACM SIGCOMM '94*, pages 234–244, 1994.

[28] Dean H. Lorenz and Ariel Orda. QoS Routing in Networks with Uncertain Parameters. *IEEE/ACM Transactions on Networking*, 6(6):768–778, 1998.

[29] Qi Xue and Aura Ganz. Ad hoc QoS on-demand routing (AQOR) in mobile ad hoc networks. *Journal of Parallel and Distributed Computing*, 63(2):154–165, 2003.

[30] Young-Bae Ko and Nitin H. Vaidya. Location-Aided Routing (LAR) in mobile ad hoc networks. *Wireless Networks*, 6(4):307–321, 2000.

[31] Lei Chen and Wendi B. Heinzelman. QoS-Aware Routing Based on Bandwidth Estimation for Mobile Ad Hoc Networks. *IEEE Journal on Selected Areas in Communications*, 23(3):561–572, March 2005.