QoS Multicast Routing in Partially Mobile Wireless TDMA Networks

Johann Gebhardt, Reinhard Gotzhein, Anuschka Igel, Christopher Kramer
Networked Systems Group
University of Kaiserslautern, Germany
Email: {j_gebhardt,gotzhein,igel,kramer}@cs.uni-kl.de

Abstract—In this paper, we present a novel QoS (quality of service) multicast routing protocol (QMRP) for partially mobile wireless TDMA (time division multiple access) networks. This protocol has been inspired by the typical requirements of industrial networked control systems, i.e., systems exhibiting real-time behavior and therefore requiring predictable communication bandwidths and transfer delays. QMRP is capable of iteratively constructing routing trees, and of assigning time slots along the branches of these trees such that a specific communication bandwidth is guaranteed and delays can be predicted. Special attention is given to mobile nodes, which require specialized QoS routing trees. A detailed evaluation gives insights into the operation and performance of QMRP.

I. INTRODUCTION

The use of wireless communication technologies for the flexible networking of sensors, actuators, and controllers in industrial automation is a crucial step on the way to the intelligent factory. These technologies provide a high-level application interface, e.g., a service interface abstracting from the distributed nature of the system. In the automation domain, real-time services such as periodical notification of sensor values with predictable communication delays are often required. To realize these services, deterministic communication protocols, i.e., protocols that can assert predictably collision-free resp. collision-protected medium access, timely delivery, and maximum delays, are required.

In a research project with several industrial partners, we are considering application scenarios involving a set of stationary nodes, such as sensors, actuators, and controllers, and a few mobile nodes, such as autonomous robots. The topology is such that the stationary nodes form a multi-hop network, i.e., there is a communication path between all pairs of nodes, and each mobile node is in range of some stationary node at all times. An application scenario in our research project is remote production monitoring and repair: e.g., when a malfunctioning of a machine is detected, an autonomous robot is sent to explore the defect. During operation, the robot dynamically subscribes to and unsubscribes from sensor services, and provides services to other stationary and/or mobile nodes itself.

In our research project, we are currently investigating and developing specialized protocols for production networks. So far, we have devised and implemented solutions for network-wide time synchronization [1], virtual medium slotting [2], deterministic arbitration [3], fast mode-signaling [4], and automatic topology discovery [5]. In this paper, we present QMRP, a novel QoS multicast routing protocol for partially mobile wireless TDMA networks. Based on the network status detected during automatic topology discovery (which considers interference links with a greater range than communication links), QMRP iteratively constructs routing trees and reserves time slots along the branches of these trees, exploiting multiple slot usage through SDMA (space division multiple access). QMRP is able to guarantee QoS requirements for bandwidth and delay by assigning exclusive slots. To avoid communication overhead, a centralized approach is used.

The paper is structured as follows. In Section II, we motivate and present our algorithms for the construction of QoS routing trees and schedules in stationary wireless multi-hop production networks. Section III extends these algorithms to support mobile nodes. Section IV evaluates the protocol, Section V elaborates on related work, and Section VI presents our conclusions.

II. CONSTRUCTION OF QoS ROUTING TREES IN STATIONARY WIRELESS PRODUCTION NETWORKS

A. Context description and design decisions

QMRP is a QoS routing protocol specifically devised to support production networks, consisting of a set of stationary and a few mobile nodes exchanging real-time control information over a wireless medium. Here, sensor and actuator nodes provide application services, such as alarm notifications, periodical temperature readings, and control value settings. Controller nodes use these services to collect sensor values, and to apply control values, thereby building feedback loops in the controlled system. In a typical usage scenario of the intelligent factory, sensor and actuator nodes register their services in a distributed service registry. Controller nodes look up, subscribe to and unsubscribe from services during operation. For each subscription to a service, a QoS route from service provider to subscriber has to be built. Often, several controllers subscribe to the same service. In this case, creating one multicast route from the service provider to all subscribers can decrease network load compared to one unicast route per subscription. Therefore, QMRP has been devised to construct QoS multicast routing trees.

From our scenario, it follows that QoS routing trees are created and modified dynamically. For instance, when the first subscription to a particular service occurs, a routing tree consisting of a single branch is created. For subsequent subscriptions and unsubscriptions, the routing tree is extended and reduced, respectively. This calls for a routing algorithm that can iteratively construct and reduce QoS routing trees by...
adding and removing branches, and can handle reservations along the branches.

Another concern is the question whether the QoS routing algorithm should be centralized or decentralized. Both options have their benefits and drawbacks. For instance, a centralized algorithm based on global topology information can yield optimized QoS routing trees, but introduces a single point of failure. Decentralized solutions, on the other hand, produce high communication overhead and are prone to failure when reservations are required (see Sect. II-C). In the context of our industrial project, we use master-based time synchronization with Black Burst Synchronization (BBS) [1]. Furthermore, we run the Automatic Topology Discovery Protocol (ATDP) [5] to automatically discover and distribute the communication and interference topologies of stationary nodes when the network is powered up. Therefore, we have decided to devise a centralized QoS multicast routing algorithm, using the timing master as routing master.

As the problem of finding the optimal QoS multicast tree and its slot reservations is NP-complete, QMRP uses heuristics to either minimize delay or maximize slot-reuse.

When a node subscribes to a service, a QoS route request is triggered by the service provider. This request is then sent to the routing master, which determines and returns a new or extended routing tree. To route this request and its response, the forwarding nodes use the communication topology discovered and distributed by ATDP. Furthermore, they use exclusive pre-reserved management time slots (see Sect. II-C), which makes this exchange highly reliable.

B. Routing tree discovery

To create and extend routing trees, QMRP uses several heuristics. For this, the communication and interference neighborhood is formalized as follows: Let \( V \) be the set of network nodes. For \( a \in V \), \( \text{CN}(a) \) is the set of communication neighbors of \( a \) and \( \text{IN}(a) \) the set of interference neighbors, where \( \text{CN}(a) \subseteq \text{IN}(a) \).

Since routing trees are constructed dynamically, tree creation amounts to finding a path \( p \) from a source \( s \) (or the current tree) to a destination \( d \). Here, QMRP determines the set of shortest paths, measured in number of communication hops. If more than one shortest path is found, the neighborhood of each shortest path \( p \) is determined as follows:

\[
\text{neighborhood}(p) \triangleq \bigcup_{a \in p} \text{CN}(a)
\]
C. Reservation of time slots

In our application scenario, nodes exchange real-time control information over a wireless medium, which requires predictable bandwidths and transfer delays to achieve timely and reliable delivery of sensor values and actuator commands. To meet these QoS requirements, we use TDMA with slot reservations, and support multiple slot usage through SDMA.

In our approach, time is hierarchically structured into time slots as shown in Fig. 3. First, time is decomposed into a sequence of super slots. The start of a super slot is marked by a super tick, i.e., a reference point in time established during a network-wide (re-)synchronization phase. Super slots are subdivided into macro slots, which are in turn decomposed into micro slots. Within a super slot, micro slots are consecutively numbered, and are the time units used for slot reservation. The start of each macro slot is marked by a macro tick, established by network-wide (re-)synchronization using BBS [1]. Thus, synchronization is repeated during a super slot, keeping tick offsets small and thereby improving bandwidth usage.

In Sect. II-A, we have argued for a centralized QoS multicast routing algorithm, using the timing master as routing master. An additional benefit of a centralized solution is that mutual blocking of pre-reserved time slots caused by concurrent route discoveries can be ruled out. Furthermore, the enormous overhead of a decentralized solution to agree on slot reservations and to exchange reservation decisions is avoided. In particular, no exchange of reservation decisions with interference neighbors is required, which is very difficult to achieve in practice, since these nodes may not be in communication range. A centralized reservation algorithm supporting SDMA requires network-wide knowledge about both the communication and interference topologies of stationary nodes. As pointed out in Sect. II-A, we run ATDP [5] to automatically discover and distribute this information when the network is powered up.

When a packet is transferred from node $a$ to node $b$ in communication range of $a$ in time slot $s$, we assume that node $b$ returns an acknowledgment in the same time slot $s$. For collision-free transmission, it is therefore required that no other node in interference range of $a$ or $b$ is scheduled to send or receive in slot $s$. This constraint can be formalized as follows: Let $s$ be a time slot and $TX^s$ be the reservation status of $s$, defining, for all pairs of nodes $a, b \in V$, whether $s$ is reserved for transmission from $a$ to $b$, provided $b \in CN(a)$. To reserve time slot $s$ for collision-free packet exchange between two nodes $a, b \in V$, $s$ must be free for transmission from $a$ to $b$, as expressed by the predicate $F^s(a, b)$:

$$F^s(a, b) \overset{\text{def}}{=} b \in CN(a) \land \forall c \in IN(a) \cup IN(b) : \neg \exists d \in V : (TX^s(c, d) \lor TX^s(d, c))$$

For slot reservation, QMRP applies two strategies. The first strategy aims at small transfer delays. The lower bound of the transfer delay is given by the product of routing tree depth and micro slot duration. This lower bound is reached if consecutive slots can be reserved for the nodes along the paths of the routing tree. In general, QMRP should keep the gaps between reserved slots small to achieve small transfer delays.

A minimal delay can be achieved by always choosing the next free slot, i.e., a slot neither blocked nor used at the sender or receiver. In Tab. I, an example schedule illustrates this strategy for a transmission from node $k$ to node $h$ over nodes $i$ and $g$ (path $p_1$ in Fig. 1). To keep the example small, it is assumed that the interference range equals the communication range. In the example, some already scheduled transmission are depicted in gray. The minimal delay strategy first checks if the first slot $0$ is free for transmission from node $k$ to node $i$, which is the case, and therefore schedules this transmission in this slot. The nodes $j$ and $l$ in interference range of $i$ and $k$ are blocked; nodes $g$ and $f$ are already blocked by another transmission. The next transmission from node $i$ to node $g$ cannot be scheduled in slot 1 because node $g$ is blocked by another transmission. Slot 2 is still free, so it is used for the transmission from node $i$ to node $g$, blocking nodes $c, f, h, j, k$ and $l$. The next hop from node $g$ to node $h$ cannot be done in slot 3 as node $h$ is already blocked. Therefore, this transmission is scheduled in slot 4, blocking $c, d, i$ and $j$. With this strategy, the transmission is scheduled with a delay of 5 slots.

<table>
<thead>
<tr>
<th>Slot</th>
<th>$0$</th>
<th>$1$</th>
<th>$2$</th>
<th>$3$</th>
<th>$4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>node $a$</td>
<td>$TX^0_a$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
</tr>
<tr>
<td>node $b$</td>
<td>$TX^1_b$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
</tr>
<tr>
<td>node $c$</td>
<td>$RX^2_c$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
</tr>
<tr>
<td>node $d$</td>
<td>$TX^2_d$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
</tr>
<tr>
<td>node $e$</td>
<td>$TX^2_e$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
</tr>
<tr>
<td>node $f$</td>
<td>$RX^2_f$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
</tr>
<tr>
<td>node $g$</td>
<td>$RX^2_g$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
</tr>
<tr>
<td>node $h$</td>
<td>$TX^2_h$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
</tr>
<tr>
<td>node $i$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
</tr>
</tbody>
</table>

Legend:

$RX^s_i$ Blocked due to transmission in interference range
$TX^s_i$ Reception from node $i$
$TX^s_i$ Sending to node $i$
$s / RX^s_i / TX^s_i$ Already scheduled transmissions

3Some of these time slots are pre-reserved management slots, which are used, among other things, to forward route requests (see Sect. II-A).

4QMRP uses symmetrical links only, which are detected by ATDP.

5QMRP uses the interference topology automatically detected by ATDP.
TABLE II. EXAMPLE SCHEDULE WITH A TRANSMISSION FROM NODE k TO NODE h OPTIMIZED FOR MAXIMUM UTILIZATION.

<table>
<thead>
<tr>
<th>Slot</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>node a</td>
<td>–</td>
<td>x</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>node b</td>
<td>RX</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>node c</td>
<td>RX</td>
<td>RX</td>
<td>–</td>
<td>x</td>
<td>–</td>
</tr>
<tr>
<td>node d</td>
<td>x</td>
<td>RX</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>node e</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>RX</td>
<td>–</td>
</tr>
<tr>
<td>node f</td>
<td>x</td>
<td>x</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>node g</td>
<td>x</td>
<td>x</td>
<td>TX</td>
<td>RX</td>
<td>x</td>
</tr>
<tr>
<td>node h</td>
<td>RX</td>
<td>–</td>
<td>–</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>node i</td>
<td>RX</td>
<td>x</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>node j</td>
<td>RX</td>
<td>–</td>
<td>–</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>node k</td>
<td>TX</td>
<td>RX</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>node l</td>
<td>x</td>
<td>–</td>
<td>–</td>
<td>TX</td>
<td>x</td>
</tr>
</tbody>
</table>

Legend: see Tab. I

The utilization of a schedule $S$ is defined as the number of transmission and reception slots of $S$ divided by the total number of slots used or blocked in $S$:

$$ utilization(S) \equiv \frac{TX_{slots}(S) + RX_{slots}(S)}{TX_{slots}(S) + RX_{slots}(S) + blocked_{slots}(S)} $$

For the considered schedule in Tab. I, the utilization is $\frac{6+6}{6+6+22} \approx 0.35$. The utilization of a schedule reaches the theoretical optimum of 1 only if all slots are either used for transmission or reception, i.e., there are no slots blocked due to transmissions in interference range.

The second strategy aims at optimizing slot utilization. High slot utilization is achieved by giving priority to SDMA, i.e., by giving preference to reserving slots such that they are still free on as many other links as possible. This is useful if the network needs to cope with a lot of traffic and delay is not the most important criterion.

This strategy is based on three Slot Decision Policies (SDPs). The SDPs originating from [7] have been adjusted for QMRP due to the fact that QMRP uses a master node for scheduling and thus can make use of global status information.

The first policy is to start the scheduling with the link with the fewest interference-free slots (fewest possibilities first policy). This policy increases the chances to find a schedule for the given route. We consider a similar example as for the first strategy (see Tab. II). Again, we try to find a schedule from node $k$ to node $h$ over nodes $i$ and $g$ (path $p_1$ in in Fig. 1). Link $[k, i]$ can be scheduled in four slots not already blocked by other transmissions, link $[i, g]$ in two slots and link $[g, h]$ in two slots. Therefore, we start with one of these links, here with link $[i, g]$.

The second policy is to choose free slots for the considered link which have the least conflicts with other links still to schedule (least conflict first policy). For the considered link $[i, g]$, slots 3 and 4 are free. Slot 3 is also free for link $[k, i]$, but not for link $[g, h]$, as it is blocked at $h$. Slot 4 is free in both other links. Therefore, link $[i, g]$ is scheduled for slot 3, which blocks nodes $f, j, k$ and $l$; nodes $c$ and $h$ are already blocked.

Now the first policy is applied to the remaining links. Link $[k, i]$ has three free slots left and node $[g, h]$ only two, so $[g, h]$ is scheduled next. By the least conflict first policy, slot 2 is chosen as it is blocked for link $[k, i]$, whereas slot 4 is free for both.

Finally, the remaining link $[k, i]$ has to be scheduled, which could be scheduled in slot 0, 1 or 4. As there is no other link to be scheduled, the least conflict first policy has no effect. Therefore, the third policy is applied: It chooses the slot in which the least nodes in interference range are free (most reuse first policy). In slot 0, interference neighbors $l$ and $j$ are free, in slot 1, $i$, $j$ and $f$, and in slot 4 all four. Therefore, link $[k, i]$ is scheduled in slot 0.

The resulting schedule starts in micro slot 0 and ends in micro slot 2 of the next super slot, thus the delay is 8 slots. The utilization is $\frac{7+7}{8+20} \approx 0.42$. As expected, this strategy yields schedules with a higher delay as the first one but achieves a higher utilization.

When a new destination is added to an existing tree, the nodes along the added branch have to be scheduled. At the node where the new branch starts, QMRP tries to use local multicasting. This means that the sent frame can be received by up to three receivers in communication range of the sender, which acknowledge reception. QMRP supports up to three ACKs in one micro slot. The order of the ACKs is determined by the node-ID, i.e., the node with the lowest ID sends its ACK first. From the schedule all nodes get from the routing master, they know the other multicast receivers and can therefore determine in which order the ACKs are sent. If local multicast cannot be used – either because the new receiver is blocked due to a transmission in its interference range, or because there are already three multicast receivers – the same frame is scheduled again in the next free slot. The rest of the scheduling of new tree branches works as described using either the minimum-delay strategy or the maximum-utilization strategy.

III. EXTENSION TO PARTIALLY MOBILE NETWORKS

A. Design decisions

We now extend QMRP to support QoS multicast routing in partially mobile production networks. In our scenario, mobile nodes, e.g., autonomous robots, may act as service providers and/or as service users, i.e., as senders and/or receivers. As service provider, a mobile node transmits data to a set of (mobile or stationary) subscribers; as service user, it receives data from a (mobile or stationary) provider.

A possible approach is to include mobile nodes in QoS routing trees, as already described in Sect. II. This, however, yields routes that are potentially unstable and tend to break when mobile nodes change their location, which is a serious problem in a real-time environment. To solve this problem, we adopt a different routing strategy, at the price of consuming more resources. To start with, we assume that a mobile node is always in range of some stationary node. For better performance, we determine a minimal subset $V_{acc} \subseteq V_{stat}$ of stationary nodes, called access nodes, such that this assumption is satisfied.

B. Routing tree discovery

To determine routing trees covering mobile nodes, we distinguish receive routes and send routes. In case of a receive route, there is a source node transmitting data to the mobile node. To establish a receive route, we start by creating a multicast tree from the source node to all access nodes. Each
branch of this tree is then extended by a last hop to the mobile node. This last hop, however, is only used by one access node in range of the mobile node at the time of transmission. To identify the access node in charge, we use beacon frames between access nodes and mobile nodes, which are exchanged in pre-reserved management slots.

An example is illustrated in Fig. 4. A multicast tree is built from the source node \( k \) to the access nodes \( b, h \) and \( i \) to reach the mobile node \( M \). We start the tree with the route to the access node nearest to the source, which is \( i \) (see path \( p_i \) in Fig. 4). We then extend the tree as explained in Sect. II-B with the branch to node \( b \) (path \( p_2 \)) and then to node \( h \) (path \( p_3 \)). In Fig. 4, the mobile node \( M \) is currently in range of access node \( b \), so the last hop would be from \( b \) to \( M \).

The case of adding a mobile node to an already existing routing tree is equal to adding all access nodes of the mobile node to the tree.

In case of a send route, the mobile node transmits data to a set of (stationary and/or mobile) destination nodes. To establish a send route, we start by determining a distributor node, i.e., a node that receives data from the mobile node and forwards them to the final destinations. This simplifies the routing significantly, as otherwise, we would need one routing tree from each access node to all destinations.

There are several possible strategies to choose a good distributor node. One strategy is to choose the access node through which the routing request was issued. This strategy yields optimal routes as long as the mobile node stays mostly in range of this access node, but once the node starts to move, inefficient routes need to be taken. Another strategy is to choose a central distributor node, which has short routes to most nodes. We decided to adopt this strategy and choose the node with the minimum average distance to all other nodes as distributor node. The best strategy for selection of the distributor node is still due to further study.

When establishing a send route, we determine a concast tree, i.e., a tree from several sources (access nodes) to one destination (distributor node), with all branches prefixed by a first hop from the mobile node to the corresponding access node. To obtain this tree, we create a multicast tree from the distributor to all access nodes, as in case of a receive route, and use it in the opposite direction\(^6\). Finally, we create a multicast tree from the distributor node to all destinations, and attach it to the concast tree.

In Fig. 5, an example illustrates a send route from mobile source \( M \) to destinations \( e \) and \( l \) through access nodes \( b, f, h \) and distributor node \( g \). First, note that the concast tree is not built exactly the same way as multicast trees. When adding a new destination to a concast tree, only the shortest path from source to destination is optimized, not the total number of links in the tree. If the total number of links would be optimized, as for multicast trees, node \( b \) would have been connected with the tree using the link to node \( f \) instead of the separate path through node \( c \). This is more efficient, because different from multicast trees, only one branch of a concast tree will be actually used at a time. Second, note that the route might use some links in both directions, e.g., from access node \( f \) to destination \( l \), the route uses the link from \( g \) to \( i \) in both directions. Obviously, this is inefficient in this case, but the route also needs to be functional when the mobile node is connected to access node \( b \) or \( h \), in which case the link from \( g \) to \( i \) is necessary.

\(^6\)This approach is feasible as QMRP only uses bidirectional links detected by ATDP.

C. Reservation of time slots

For routing trees covering mobile nodes, time slots have to be reserved. As before, we distinguish receive and send routes. In case of a receive route, we have already created a QoS multicast tree from the source node to all (stationary) access nodes. In case the source node is stationary, slot reservations for this multicast tree are covered by the algorithm in Sect. II-C. Then, each branch of this tree has been extended by a last hop to the mobile node. Since only one of the access nodes will forward the data on this last hop to the mobile node, it is sufficient to reserve exactly one time slot. However, since the location of the mobile node is not fixed, this reservation must be exclusive, i.e., multiple slot usage through SDMA is not feasible.

In case of a send route, several kinds of slot reservations are needed. First, there is the concast tree from the access nodes to the distributor node. Since the mobile node will send its data only to the access node currently in charge, only one branch of the concast tree will be activated. Thus, the different branches of the tree do not block each other. Therefore, slots are scheduled the same way as described in Sect. II-C, with the exception that blocking entries are ignored if they are only caused by another branch of the same concast tree. In addition, a reservation for the first hop from the mobile node to the
success 7111216 5 12 slots 2 success 50 6 slots 9 8 12 3 15 50

Tree depth 20 2 3 3 4 5 10 4 15 0 50 10 5
delay 15 3 7 0

are fundamentally different in functionality and/or QoS support performance of QMRP to existing routing protocols, as they difference is not very high, possibly because in these compara-
corresponding simulations with the minDelay strategy. The
slightly higher with the maxUtil strategy compared to the
in Tab. III and Tab. IV.

In a series of simulations, we have compared the strategies of QMRP to minimize transfer delay (minDelay) and to max-
imize slot utilization (maxUtil). Furthermore, we have assessed the benefits of using access nodes to support send and receive
routes to and from mobile nodes. We have not compared the performance of QMRP to existing routing protocols, as they are fundamentally different in functionality and/or QoS support (see Sect. V).

For our simulations, we have used different topologies. In this paper, we elaborate on two of these topologies, which are shown in Fig. 6 and 7. Topology 1 represents dense networks with rather evenly distributed nodes and links; topology 2 represents stretched networks as they would typically occur in a production line. We assume that the interference neighborhood of a node (not shown in the figures) equals its 2-hop communication neighborhood.

To compare the performance of the strategies minDelay and maxUtil for stationary nodes, we have randomly generated 50 scenarios with three multicast trees each, where each tree has one source and three destinations. These scenarios were run on both topologies, with both strategies, and a super slot size varied from 10 to 15 micro slots. The results are summarized in Tab. III and Tab. IV.

First, we observe that the arithmetic mean utilization is slightly higher with the maxUtil strategy compared to the corresponding simulations with the minDelay strategy. The difference is not very high, possibly because in these compara-
dense topologies, transmissions always block many other

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Slots</th>
<th>Utilization</th>
<th>Tree depth</th>
<th>Delay</th>
<th>Success</th>
</tr>
</thead>
<tbody>
<tr>
<td>maxUtil</td>
<td>10</td>
<td>0.19 ± 0.02</td>
<td>2.59 ± 1.22</td>
<td>4.66 ± 4.10</td>
<td>27</td>
</tr>
<tr>
<td>minDelay</td>
<td>10</td>
<td>0.18 ± 0.02</td>
<td>2.52 ± 1.21</td>
<td>3.16 ± 2.01</td>
<td>20</td>
</tr>
<tr>
<td>maxUtil</td>
<td>12</td>
<td>0.17 ± 0.02</td>
<td>2.74 ± 1.24</td>
<td>4.36 ± 4.08</td>
<td>48</td>
</tr>
<tr>
<td>minDelay</td>
<td>12</td>
<td>0.17 ± 0.01</td>
<td>2.74 ± 1.24</td>
<td>3.61 ± 2.48</td>
<td>47</td>
</tr>
<tr>
<td>maxUtil</td>
<td>15</td>
<td>0.18 ± 0.02</td>
<td>2.74 ± 1.24</td>
<td>6.06 ± 6.39</td>
<td>50</td>
</tr>
<tr>
<td>minDelay</td>
<td>15</td>
<td>0.17 ± 0.01</td>
<td>2.75 ± 1.23</td>
<td>3.56 ± 2.18</td>
<td>50</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Slots</th>
<th>Utilization</th>
<th>Tree depth</th>
<th>Delay</th>
<th>Success</th>
</tr>
</thead>
<tbody>
<tr>
<td>maxUtil</td>
<td>10</td>
<td>0.24 ± 0.01</td>
<td>3.54 ± 2.30</td>
<td>7.33 ± 7.28</td>
<td>8</td>
</tr>
<tr>
<td>minDelay</td>
<td>10</td>
<td>0.23 ± 0.01</td>
<td>3.33 ± 2.12</td>
<td>4.41 ± 3.63</td>
<td>5</td>
</tr>
<tr>
<td>maxUtil</td>
<td>12</td>
<td>0.24 ± 0.01</td>
<td>3.96 ± 2.22</td>
<td>9.08 ± 8.34</td>
<td>45</td>
</tr>
<tr>
<td>minDelay</td>
<td>12</td>
<td>0.23 ± 0.01</td>
<td>3.63 ± 2.06</td>
<td>6.36 ± 5.14</td>
<td>26</td>
</tr>
<tr>
<td>maxUtil</td>
<td>15</td>
<td>0.24 ± 0.01</td>
<td>3.99 ± 2.21</td>
<td>11.22 ± 11.18</td>
<td>50</td>
</tr>
<tr>
<td>minDelay</td>
<td>15</td>
<td>0.23 ± 0.01</td>
<td>3.99 ± 2.21</td>
<td>5.7 ± 4.34</td>
<td>50</td>
</tr>
</tbody>
</table>

In conclusion, the strategy should be chosen based on the QoS requirements of the transmissions. For time-critical transmis-
sions, the minDelay strategy is recommended. If there is a lot of traffic in the network, the maxUtil strategy allows to schedule more transmissions and thus is recommended for high-traffic networks with lower requirements on delay.

To access the benefit of using access nodes, simulations with both send and receive routes were run on both topologies. For receive routes, we evaluated the advantage of using access nodes. Without access nodes, a broadcast to all stationary nodes would be necessary as it cannot be known in advance which node is in connection with a mobile node. Fig. 6 shows topology 1 with a multicast tree from source node 3 to four access nodes. Scheduling this multicast tree produces a maximum delay of 6 micro slots and blocks 69 slots (evaluated with minDelay). In comparison, scheduling a multicast tree reaching all nodes results in a maximum delay of 11 micro slots and 128 blocked slots, despite exploiting local multicast. This shows that the usage of access nodes reduces the network load significantly, even in comparably small networks.

For send routes, the scheduling improves slot reusage by scheduling several branches of a concast tree concurrently, exploiting the fact that the mobile node only sends to one access node. To evaluate the effect of this optimization, we have scheduled a concast tree from four access nodes to distributor node 6 in topology 2 (see Fig. 7) with and without this optimization. Without the optimization, the delay is 7 micro slots whereas with the optimization, only 3 micro slots are needed (also evaluated with minDelay).
V. RELATED WORK

The advantage of multicast routes in comparison to several unicast routes is well known and several multicast routing protocols exist (e.g., [8]–[10]). In our research project, these protocols have been found unsuitable, mainly because they lack QoS support. Other routing protocols like the one described in [11] focus on QoS support, but only consider unicast. PSLCB [12] and the Hexagonal-Tree-Routing protocol [13] are TDMA-based multicast routing protocols with QoS support, but fail to prevent interferences in cases where the interference range is larger than the communication range [14]. Other protocols like [15] and [16] combine CDMA (code division multiple access) with TDMA, but also do not consider that the interference range may be larger than the communication range. Furthermore, the use of CDMA introduces additional overhead, e.g., for computation and distribution of the codes. Using the interference topology that is automatically detected by ATDP, QMRP is able to reliably prevent interference while still exploiting SDMA.

Existing routing protocols usually don’t make use of topology information gathered by another protocol, as QMRP does, but do their own topology detection. To this end, reactive protocols like AODV [17] flood the network using route request packets whenever a new route is needed. Proactive protocols like OLSR [18], on the other end, regularly detect and exchange topology information. In contrast to these protocols, we decided to separate the problem of topology detection from the routing and scheduling problem. This allows us to reuse the gathered topology information, e.g., for clustering.

Regarding the support of mobile nodes, routing protocols like AODV often allow all nodes to move, producing unnecessary overhead for stationary nodes. On the other hand, reservation protocols like [7] assume that the topology is stable. If nodes move, reservations need to be repaired, which can make communication with mobile nodes temporarily unreliable. By handling stationary and mobile nodes differently, QMRP is able to handle the grade of mobility that is needed in our scenario while ensuring reliable and efficient communication for both mobile and stationary nodes.

VI. CONCLUSIONS

In this paper, we have presented QMRP, a novel protocol for QoS multicast routing in partially mobile wireless TDMA networks. This protocol has been inspired by the typical requirements of industrial networked control systems, i.e., systems calling for predictable transfer bandwidths and transfer delays. We have described the functionality and strategies of QMRP to iteratively construct routing trees, and of assigning time slots. A detailed evaluation has given insights into the performance of the protocol.

To gain practical experience, we are currently integrating QMRP into ProNet 4.0 (Production Network 4.0), a prototypical communication system for the intelligent factory. Due to its extensive functionality, its real-time support, its service interface, and its high degree of flexibility, ProNet 4.0 is particularly suitable for use in Industry 4.0, a future project in this context.

A topic for further study is the tuning of some of the strategies used by QMRP. For instance, when constructing a routing tree, QMRP chooses the shortest path with the largest neighborhood size to optimize future tree extensions. Another choice could be to use the smallest neighborhood size to improve route discovery success rates. Also, the strategy for selection of distributor nodes for send routes in mobile scenarios can be tuned. The evaluation showed that the strategies for minimum delay and maximum utilization both have their benefits. We therefore plan to automate the choice of the strategy based on QoS requirements and network usage.

REFERENCES