A Service Oriented Framework for Wireless Communication in Mobile Multi Robot Systems

Alexander Tiderko and Thomas Bachran
Research Institute for Communication,
Information Processing, and Ergonomics (FKIE)
Research Establishment for Applied Sciences (FGAN)
Neuenahrer Straße 20, D-53343 Wachtberg
{tiderko, bachran}@fgan.de

***Abstract*—Multi robot systems (MRS) have been an active research area for many years. With the robot systems leaving the laboratory environment, the aspect of wireless communication is getting more important. Especially in multi robot systems wireless communication is crucial, but existing middleware for robot systems does not regard the challenges of wireless communication. Among other aspects the fluctuation of link delay, bandwidth and availability is characteristic for wireless communication. A communication system in a multi robot system must be capable to manage these challenges. Another issue is efficient point to multipoint (multicast) communication. This allows the availability of the same data to multiple recipients without an unnecessary high increase of network load compared to unicast communication schemes.**

We therefore propose a multicast communication framework for a multi robot system based on wireless communication. The framework is organised in two parts. The first part is a general framework for a wireless multicast communication protocol, which is optimized to offer efficient multicast communication capabilities on wireless networks and takes the available link qualities into account. The second part is a framework for a robot middleware, which benefits from the multicast communication optimizations of the first part. The framework assumes unreliable communication, thus taking fluctuating link qualities into account. Both parts are already implemented on a real system and are used in an experimental multi robot system.

I. Introduction

Currently many research groups investigate the design and development of robot software frameworks. Unfortunately most approaches concentrate on the higher network layers disregarding aspects from lower network layers like e.g. the fluctuating availability of communication links and their quality. With the robot systems leaving the laboratory environment, aspects of wireless communication must be taken into account to transfer robot applications to real world scenarios. Due to the increased importance of the lower network layers many framework approaches, applicable for the laboratory environment, are inefficient or even unusable in real world scenarios. Future robot software frameworks for multi robot systems should be able to manage the challenges arising from the mobility of robot systems and the usage of wireless communication even under adverse conditions for wireless communication. Such a framework should assume that connections between robot applications are not reliable and applications implemented for this framework should be robust against loss of messages.

Two important applications for multi robot systems are emergency management and military robotics. Both make high demands on the communication system. The demands for military applications were summarized during the *1st NATO Workshop on short-term realizable (multi-)robot systems in military domains* [1]. Beside evident requirements like mobile and wireless communication, high data rates and high ranges the communication group of this workshop pointed out the aspects of multicast communication, (link) quality-aware routing, quality of service (QoS), data prioritization, security and power awareness. Since current and short-term realizable communication systems cannot meet all these requirements, our work focuses on including multiple of these aspects into one communication system.

Based on these requirements we designed a framework for wireless mobile ad-hoc network protocols called WNet [2]. In section III we describe WNet with a focus on its multicast capabilities and quality-aware routing. Afterwards in section IV we present another framework, called Robot Services (*RoSe*), for the application layer. The main purpose of this framework is to replace existing middleware approaches to enable robot applications to efficiently support multicast communication and to benefit from lower layer improvements as presented with the WNet protocol. We therefore address only the communication aspects of the framework. In section V we summarize our work and point out some aspects of our future work.

II. Related work

The first part of this paper deals with wireless mobile ad-hoc networks (MANETs). MANET protocols have been actively studied in recent years. In these studies a lot of aspects were taken into account.

One of these aspects is quality-aware routing. Several metrics for the estimation of a link's quality have been proposed, e.g. the Expected Transmission Count (ETX) [3] or the Weighted Cumulative Expected Transmission Time (WCETT) [4] metric. Some unicast protocols support quality-aware routing by using these or other metrics. The QOLSR protocol [5] for example extends the OLSR protocol [6] by using a link quality derived from link bandwidth and delay

ROBOCOMM 2007, 14th – 16th Oct 2007, Athens, Greece.

ISBN 978-963-9799-08-0
DOI 10.4108/ICST.ROBOCOMM2007.2079

measurements. Another approach, the LQOLSR protocol [7], uses a link delay derived from the Signal to Noise Ratio (SNR) and packet delivery rate measurements. Another widely used implementation of the OLSR protocol [8] uses the ETX metric to improve the routing performance of OLSR.

Another aspect is the capability of multicast communication. Multicast capable protocols can be classified by a proactive or reactive behavior. Proactive protocols like MOLSR, SMOLSR (both proposed in [9]) or SMF [10] in conjunction with a neighborhood discovery protocol frequently flood the network with management frames to gather s information about the network topology. However reactive protocols like ODMRP [11] or MAODV [12] also flood the network but only when there is a demand, i.e. data to forward is available, and only to build up routes describing how to deliver the data to its destinations. Another important criterion to distinguish between these protocols is how the data is propagated in the network. While the SMF protocol uses the topology information to efficiently flood the network whenever a new multicast packet is available, the MOLSR protocol tries to reduce the load on the network by building a routing tree to the destinations of the packet and restricting forwarding to this tree.

In our approach we regard both aspects to develop a protocol that is capable of multicast routing and gains performance by using the advantages of quality-aware routing.

The second part of this paper discusses robot middleware approaches for multi robot systems. Although there exist many approaches, they unfortunately do not properly support multicast communication or handle the unreliability of wireless communication. A well-known example is the Player/Stage software [13]. As default it uses TCP communication [14]. Since TCP is designed for unicast transfers it cannot be used for efficient multicast communication. Beside this aspect several performance problems arise from the usage of the congestion control scheme of the TCP protocol when used over wireless links. TCP assumes that all losses indicate congestion, leading to a reduction of the transmission rate. This assumption is not adequate for wireless communication. This and other issues are discussed e.g. in [15].

Other robot middleware like MIRO [16] or ORCA [17] is based on a middleware commonly used in Internet applications like CORBA [18] or ICE [19]. Using these middlewares may ease implementing robot services but also incorporates the assumption and design decisions made for the middleware. Since they are designed for today‘s Internet these assumptions are usually not fully compatible with wireless communication. Both middleware approaches use TCP communication as default, leading to the problems already discussed.

Some of the robot middleware described before also support UDP communication as an alternative to TCP, but current applications implemented on these middleware assume the reliability of a TCP connection in the Internet. To really support unreliable communication, the applications have to be customized, too. A similar statement can be made for the multicast communication aspect. We therefore state that the middleware known to the authors are not fully compliant with wireless communications and that a new and special approach for efficient communication in multi robot systems is required.

An interesting middleware [20] is proposed in the context of the EU-funded IST-RUNES project. This middleware was developed parallel to our framework and is a generic approach which allows for different network types and communication schemes. It might be of interest to combine the RUNES middleware with our WNet framework to evaluate whether a similar performance as with the RoSe framework can be achieved. For such a combination it would be advisable to consider the assumptions made for the RoSe framework.

III. WNet

In this section we focus on the WNet framework for a wireless multicast capable MANET routing protocol. The framework is implemented on the data link layer of the ISO/OSI network stack. A WNet network can therefore be regarded like an IP subnet. This allows an easy reuse of existing applications, without the need of taking IP routing specific aspects into account, like e.g. the time to live (TTL) of IP packets.

The design of the WNet framework is based on the requirements for a wireless communication protocol in multi robot systems, which were already presented in section I. In this paper we concentrate on the multicast and quality based routing mechanisms used in the framework. While these mechanisms alone cannot be used to meet all requirements, they address multiple of these requirements and can lead to performance benefits. A first performance evaluation was presented in [21]. A more detailed analysis is considered future work.

The optimized multicast communication mechanisms without using retransmissions on communication links can reduce the load on the network therefore allowing more data to be transferred. Since the loss rate may be very high we introduced a retry mechanism to counter this problem and to increase the reliability of transmissions. The combination of the multicast mechanism with retries allows for robust multicast communication on low to medium network load rendering video communication over multiple hops possible. A side effect of the retry mechanism is an increased delay that gains importance when the load on the network increases. Having very high load conditions or even congestion the communication in mobile ad-hoc networks generally suffers from high delays and loss rates. Under these circumstances other protocols do perform better but this is considered to be of no great importance.

The quality based routing mechanism monitors fundamental attributes of the wireless links to estimate the quality of these links. The knowledge of link qualities is then used to influence the communication path chosen by WNet. Depending on how parameters for the link quality estimation are determined the loss rate and the delay experienced on the communication endpoints can be reduced. Furthermore paths may be chosen to avoid congested areas or to reduce the overall load on the network. Since WNet is integrated in the MAC layer, more link attributes for estimating a link's quality are available without the need of a cross layer approach. One example for such an attribute is the packet's signal strength measured at the receiver. We regard quality based routing as one key issue to meet the requirements for communication in multi robot systems. Our current work shows, that using optimized metrics for quality based routing results in significant performance gains.

The concept of the WNet framework is based on the OLSR protocol. Thus WNet inherits its proactive behavior. Like OLSR, WNet uses *HELLO* and *Topology Control (TC)* management frames to exchange information about the status of network nodes. This information is used to create and update the network topology graph stored in each network node. In contrast to OLSR, the *HELLO* and *TC* frames contain additional information about link attributes which is integrated into the topology graph. These link attributes are used to estimate the link quality. Based on the link quality information and the network topology data is forwarded using the Dijkstra algorithm [22]. Since WNet uses an explicit multicast mechanism, uni- and multicast routes are determined in the same manner. Unicast transmissions are regarded as multicast transmissions with only one recipient.

A. The WNet Topology Graph

The WNet topology graph is a node local data structure containing all protocol-relevant topology information. Since the data structure is maintained and updated on each node the topology information may differ slightly from node to node. All nodes in the graph are identified by their MAC address while the links are directed and identified by the source and destination MAC address of the corresponding nodes. Additionally a link ID is used to indicate the instance of a link. The instance may change if a link is temporarily not available.

Figure 1 shows a sample topology graph. The local node, which stores the shown topology graph is denoted *L*. The other nodes are divided into different categories. Nodes which are reachable by an outgoing link are called neighbours and denoted with *N*. All nodes which are reachable by an outgoing link of a neighbour are called two-hop neighbours and are denoted *2*.

The topology graph is created and updated by *HELLO* and *TC* control messages. These message are sent periodically by each node and contain information about the vicinity of a node. *HELLO* messages are broadcasted but not forwarded. They contain information about the local node and all its incoming and outgoing links. Thus *HELLO* messages are used to keep the two-hop neighborhood of a node up-to-date.

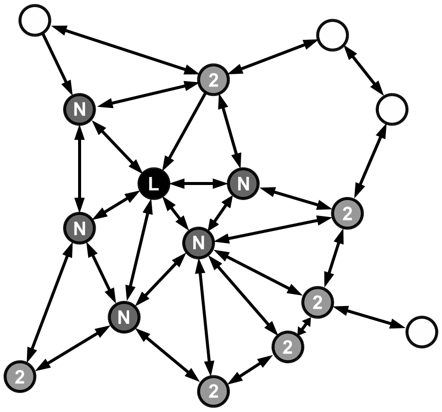

Fig. 1. A sample network topology graph

TC messages contain similar information, except that they are missing the outgoing links. Since it is sufficient to know the incoming links of all nodes and the since *TC* messages are flooded over the whole network it is reasonable to reduce the amount of data transmitted. For the same reason the *TC* messages are sent less frequent than the *HELLO* messages. To optimize the flooding process the fresh information about the two-hop neighborhood of a node is used for reducing the number of nodes which forward the data. These nodes are called *Multi Point Relays (MPRs)* and the flooding mechanism is called *MPR flooding* [23]. The set of *MPRs* is defined as the optimal (usually minimum) set of neighbours to reach all two-hop neighbours. The determination of such a minimum set is known to be a NP-hard problem [24]. For this reason we implemented a simple non optimal solution. For advanced methods it should be considered to take link qualities into account.

On the reception of *HELLO* or *TC* messages the source node and its link information are updated. The update process should also avoid updating link information with older information from other neighbors. If a node is not reachable anymore, e.g. due to moving out of range or shutting down the radio, no more control messages are received from this node. In order to avoid ghost nodes in the topology graph, the node and its incoming and outgoing links have to be deleted. To check whether link information is too old, a timestamp of the last update is assigned to each link. If a link is decided to be outdated it is removed from the graph. If a node has no more outgoing links, the node will be removed, too.

B. Link quality estimation

Routing in WNet is performed by using the Dijkstra algorithm with links weighted by a link quality value. This value is calculated using the information of the topology graph described in the previous section. For this purpose WNet exchanges additional information (node and link attributes) with the *HELLO* and *TC* messages. The most important attributes are the received signal strength (*Sig*) and the loss rate (*LR*) of a link and the utilization ($Util$) of the air interface as a node attribute. For each of these three attributes a single quality value (Q_{Sig}, Q_{LR} and Q_{Util}) is calculated and based on these values an overall quality value is determined by using a weighted geometric mean:

$$\overline{Q} = \sqrt[w]{Q_{\text{Sig}}^{w_1} \cdot Q_{\text{LR}}^{w_2} \cdot Q_{\text{Util}}^{w_3}} \qquad \text{with } w = \sum w_i \tag{1}$$

The performance of WNet strongly depends on how the link quality value is derived from these attributes and how these attributes are determined. In this work we present the original proposal of [25]. Our current work is to analyze the influence of the three attributes on the overall performance of the protocol. Simulation results presented in [21] have shown that enabling only the signal strength part of the metric by setting all other weights to zero can lead to performance benefits compared to classical hop-based routing metrics. Additional results from real world measurements confirmed this observation. Currently we investigate the influence of the loss rate metric part on the overall performance. We can already state that combining the original loss rate metric with the signal strength metric is suboptimal and that the metric presented here can be optimized by changing the attribute estimation and how the attribute is mapped to a quality value.

The signal strength attribute of a link is updated every time a new received signal strength estimate is available due to the reception of a frame. The attribute value is then mapped to the quality value Q_{Sig} using the following mapping function:

$$Q_{\text{Sig}}(x) = 1 - e^{\frac{-x+n}{10}} \tag{2}$$

The parameter n specifies the noise level measured when the signal strength attribute was determined. On most hardware assuming a constant noise level of -95 dBm seems to be sufficient. If this is not the case it can be more convenient to publish the signal to noise ratio (SNR) instead of the received signal strength. This mapping function allows for a small reaction to high signal strength values but a strong reaction to low signal strength values.

The loss rate attribute is calculated every TC interval based on the sequence numbers of *HELLO* and data messages. The loss rate obtained from *HELLO* messages is used to give a basic knowledge about a quality of a link even in the absence of data traffic. If data traffic is available, a second loss rate is determined and combined with the *HELLO* loss rate. If no *HELLO* or data frame was received during a TC interval, there is a great uncertainity of the availability of a link. To represent this uncertainty the loss rate of the former interval LR_{i-1} is used and aged to form the new loss rate attribute value LR_i:

$$LR_i = \frac{1 + LR_{i-1}}{2} \tag{3}$$

The loss rate attribute is mapped to the quality value Q_{LR} using a sigmoid function:

$$Q_{\text{LR}}(x) = \frac{1}{1 + b \cdot x^c} \tag{4}$$

The parameter set of this function was set to $b = 1\,000\,000$ and $c = 4$. These parameters were only loosely adjusted and simulations have shown that these parameters can be further optimized. By usage of the sigmoid mapping we ensure, that a low loss rate results in a high quality value until the specific threshold is reached. Then there is a strong reaction to the rising loss rate.

The utilization attribute of a node is calculated every *HELLO* interval. The calculation is based on the local utilization, the time used by a node to transmit data during the last *HELLO* interval. The local utilization values are then summed up for a node and all of its incoming links. The result is the utilization of a node. If it is possible to directly access a radio of a node, it may be possible to read the utilization information directly from the radio. This information would be more accurate.

To determine a links utilization Q_{Util} we first map the node attribute value of both endpoints and then combine them using their geometric mean. As the mapping function we use the same sigmoid function as for Q_{LR} using the parameters $b = 100$ and $c = 10$.

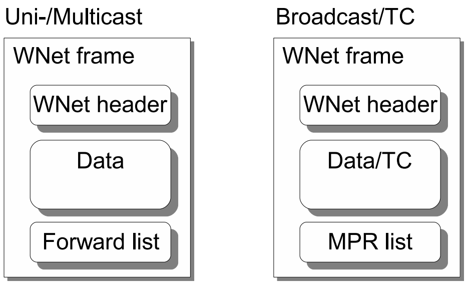

Fig. 2. Format of forwarded frames

C. Data forwarding

Data forwarding in WNet is performed using the MAC broadcast mechanism of IEEE 802.11. Therefore WNet has no access to features like RTS/CTS or the WLAN retry mechanism that can be used for MAC unicast transfers. In order to counter the disadvantages arising from this decision WNet implements its own retry mechanism and some other features. These features will be described in subsections III-D and the following. In this subsection we restrict ourselves to the basic forwarding mechanism.

WNet provides two frame types for forwarding. The first one is used for unicast and multicast data frames and forwarding is based on the Dijkstra shortest path calculation. The second one is used for broadcast data and TC frames and MPR flooding is used to publish the frame in the network. The structure of both WNet frame types is shown in figure 2.

The forward list used by uni- and multicast frames contains one entry per frame destination. Each entry holds a MAC address for the destination node and one address for the next hop on the path to the destination. Additionally it contains a link sequence number for the link to the next hop. Every time a frame is to be forwarded a new next hop is calculated. This next hop is determined by the Dijkstra algorithm using the link quality values as weights. Since the forward list entry does only contain the next hop and not the whole path to the destination, the best path is recalculated every hop. If the topology graph does not contain the destination node, the corresponding forward entry is removed from the forward list and if all entries are removed the whole frame is dropped. In order to avoid duplicate frames, a frame is only accepted by a destination node if the node is listed as a destination and a next hop in the same forward list entry.

The MPR list of TC and broadcast frames is used to specify the multi point relays (MPRs) that are authorized to forward the frame. It consists of one entry per MPR. Each entry contains the MAC addresses of the MPR and the link sequence number for the link of the local node to the MPR. If a broadcast frame is to be forwarded the forwarding node creates a new MPR list by determining a set of first hop neighbors that are capable of reaching all two-hop neighbors. To reduce the load in the network the set of neighbors should be as small as possible, but as stated before the calculation of a minimum set is a complex problem. WNet therefore uses only a simple approximation for this problem. If a node receives a broadcast frame, it checks whether the frame was already received before. If it was received before, the frame is dropped, otherwise the frame is sent to the upper network layer and if the node is listed as an MPR the frame is also forwarded. To check whether a broadcast frame was received before a broadcast sequence number was introduced in the WNet frame header. It is compared to the last received broadcast sequence number of the source of the frame node which is updated in the local network topology graph every time a new broadcast frame with this source is received.

D. Rate Selection

Other multicast protocols perform multicast communication by either using the unicast mechanism of 802.11 or the broadcast mechanism with one common data rate—usually the lowest data rate. In the first case multiple transmissions are necessary to reach multiple next hops resulting in an unnecessary high load. In the second case not all reachable nodes are reached if a high data rate is used, or transmission time is wasted if a low data rate is used. Since WNet knows about the link attributes to all next hops of a data frame, it can select the maximum data rate applicable to reach these next hops. WNet uses the signal strength attribute to decide which data rate should be used on a link. For each next hop this data rate is determined and the lowest data rate of these rates is selected for frame transmission. The rate selection mechanism is used for all unicast and multicast data frames. It may also be applicable to broadcast and TC frames but the selected rate should be considered during the MPR selection.

E. Multicast Capable Retry Mechanism

To counter the problem of high link loss rates WNet implements its own multicast capable retry mechanism. This retry mechanism is used for uni- and multicast frames and can optionally be enabled for broadcast and TC frames—but only for the links to the selected MPRs. Every WNet frame contains a "no acknowledgement" flag. This flag is set for every try which is not the last try. If this flag is not set, the receiver of a frame, which is listed in the forward or MPR list, sends an acknowledgement to inform the sender that the frame was received successfully. To distinguish between several tries a retry number is used so that the sender can determine the successful try that was acknowledged. This is important to determine a real loss rate of a link.

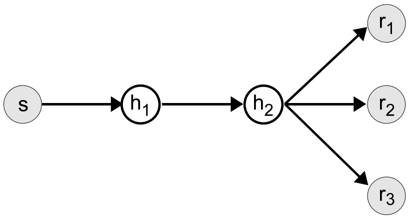

Fig. 3. Scenario for multicast test environment

Based on round trip times that are measured using the acknowledgement a sender can determine the expected time for receiving an acknowledgement. Based on this time the sender waits until it retransmits the frame with an increased retry number. In our work we usually use a maximum of three tries for each frame.

Every time a sender receives an acknowledgement the corresponding forward entries are removed from the frame. It is not necessary to forward the frame to these destinations again. This may allow a higher data rate and therefore a reduced transmission time reducing the load on the network. If all forward entries are removed the frame was transmitted successfully and no further tries are performed.

One problem which may arise from the usage of the retry mechanism is that unidirectional links do not allow to use acknowledgements so that a sender will use all tries. This may lead to congestion and therefore the retry mechanism should only be used if there is a low probability that such links exist.

Another problem which can occur due to the retry mechanism are duplicate frames. To avoid duplicates a receiver must monitor all received frames for a link. Frames containing a link sequence number which was already received are acknowledged but dropped. Such a situation can occur if an acknowledgement was lost or if an acknowledgement was late and the sender has already sent a retry.

F. Cross layer optimization

Even though WNet is designed to forward data on the MAC layer we added additional support to handle some IP signalling directly. We therefore added the IP address of a node to the *HELLO* and *TC* frames. This enables WNet to directly respond to ARP requests without broadcasting ARP messages on the network.

Additionally a WNet node interprets IGMP messages and knows about the multicast groups it is listening to. This information is also published by *HELLO* and *TC* frames. With the IP addresses and the multicast groups published in the network, every node can determine the receiver MAC addresses of every multicast group present in the network. If a multicast IP packet is received from the upper layer this capability is used to fill the forward list of the WNet frame with the corresponding destination MAC addresses.

G. Short Evaluation

The WNet framework is currently implemented for the ns-2 network simulator and for Linux. While a first performance

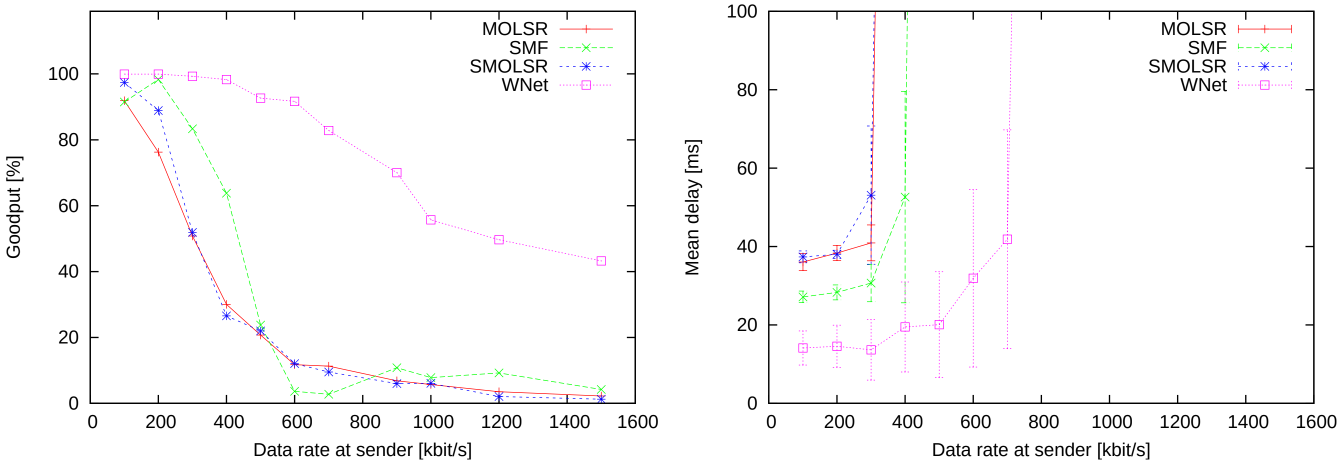

Fig. 4. Comparison of WNet with other protocols

analysis using simulations was presented in [21], we now introduce some results obtained from a simple static real world testbed. These results may lead to a first impression on the performance benefits of using the framework. The scenario is shown in figure 3.

In this scenario the sender node s streams multicast UDP data with a constant bitrate and a packet size of 1000 bytes to three receivers r_1, r_2 and r_3. We measured the goodput and the delay at all receivers by varying the data rate. Figure 4 shows the result for WNet with its signal strength metric part enabled and all other metric parts disabled. These results are compared to the protocols MOLSR, SMOLSR and SMF.

Figure 4 shows that WNet can provide a higher goodput especially when the data rate is increased. The higher performance of WNet can be put down to the retry mechanism. Since the retry mechanism leads to a higher delay of received frames we expected a higher delay compared to the other protocols, but even delay is lower. The results also suggest that WNet runs into congestion at data rates higher than 600 kbit/s. Regarding the other protocols we observe an acceptable behavior at rates up to 300 kbit/s.

IV. RoSe

In section II we introduced robot middleware approaches. Since these approaches are currently not applicable to real world scenarios, such a middleware must be altered to fulfill the requirements of outdoor multi robot systems. One of the greatest challenges is to address the problem of wireless communication. Using TCP communication should be avoided. Even though some approaches can use the UDP protocol, the communication scheme remains the same. They are designed to use permanent and reliable connections even though the network cannot provide them. Additionally they do not efficiently support multicast communication.

In this section we briefly present a simple framework, that addresses these above-mentioned problems. This framework is described in more detail in [26]. It does not assume that permanent and reliable communication is available. Instead it assumes that services are robust against the loss of connections, can cope with fluctuating arrival times of data packets and incomplete information due to partial loss of data. As an example a simple solution to counter fluctuating arrival times can be to buffer data. If such a mechanism is not applicable, alternative solutions must be chosen. Another example for a robust service could be a tracking algorithm using multiple sensor nodes. If a sensor node cannot deliver data the algorithm should still be able to track objects even though the precision may be reduced. Since the robustness is an issue for specific robot services, this issue is not addressed by our framework. It can only provide services with an unreliable UDP communication scheme.

The other aspect which should be considered when implementing a framework is the multicast support. Multicast communication in multi robot systems is motivated by the fact that sensor and processed data is often required by multiple services. For example the data of a laser sensor can be used for a collision avoidance service by a single robot, for distributed mapping algorithms and for a service that visualizes the vicinity of a robot for an operator. Multiple unicast connections would lead to multiple laser data packets therefore increasing the load in the network. If multicast communication is efficiently supported by the lower network layers, a single packet is sufficient to publish the laser data to all services therefore saving valuable network resources.

A. Communication scheme

The communication scheme used in our framework is a simple publish-subscriber approach. An unicast and a multicast address is assigned to each service. While the unicast address is used to directly address the service, e.g. used in commands to the service, the multicast address is used to publish information of the service. Every service that wants to receive data from another service has to register itself to the specific multicast group.

In figure 5 the concept of the communication scheme is illustrated by the example of three services. Service B wants

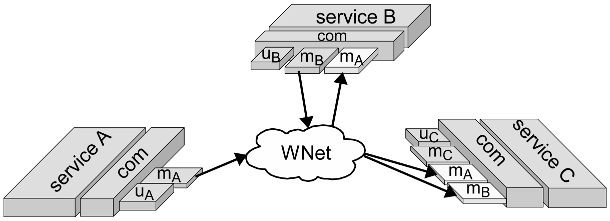

Fig. 5. Communication scheme in RoSe

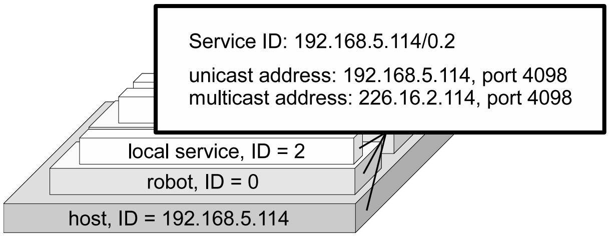
Service ID: 192.168.5.114/0.2
unicast address: 192.168.5.114, port 4098
multicast address: 226.16.2.114, port 4098
local service, ID = 2
robot, ID = 0
host, ID = 192.168.5.114

Fig. 6. Service ID, IP addresses and port number of a service

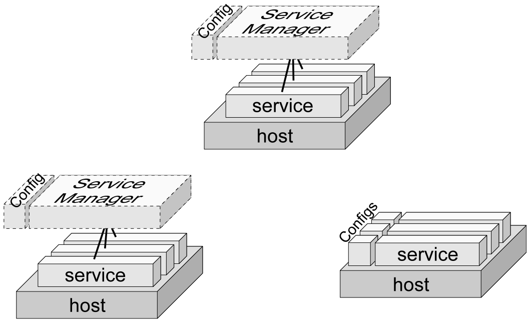

Fig. 7. Decentralized architecture with and without a Service Manager

to receive data from service *A*, so *B* registers itself to the multicast group of *A*. Service *C* registers itself to the multicast groups of *A* and *B* and therefore automatically receives data from both services. Since the service sending multicast data does not need to care about where to send the data, there is no management overhead for managing incoming and outgoing connections of the service. The data is automatically transported by the lower network layers. While the communication scheme is also applicable to other communication protocols it benefits of using WNet as the lower network layer protocol. Since a WNet network node only transmits multicast data if a multicast recipient is available, the network load is automatically kept low if no recipient is available.

One difficulty for this communication scheme is to provide security if intruders are possible. If this is the case all service data should be encrypted by a key known to the sender service and all possible recipients. A method for exchanging multicast keys can be found in [27].

B. Addressing concept

A service is identified by a single Service ID (*SID*). The *SID* is used to derive the corresponding unicast and multicast addresses. It consists of three parts: The Host ID (*HID*), the Robot ID (*RID*) and the Local Service ID (*LSID*). The *HID* identifies the host where the service is located. It is given by the IP address of the network device of the host. The *RID* is used to distinguish between multiple robots controlled by the same host. While this should be a rare case for real robot systems, it can be used for simulating multi robot systems. The *LSID* is then used to distinguish between multiple service on a robot, e.g. between a laser sensor service and a collision avoidance service. The *SID* is specified in the format 'HID/RID.LSID'. An example is given in figure 6.

The unicast address of a service is given by the unicast IP address that is equal to the *HID* and the service specific port number. This port number is calculated by the following function:

$$port = (RID+4)*1024+LSID \tag{5}$$

Using this function the number of robots on one host is restricted to 32 robots and the number of services per robot is restricted to 1024.

For the multicast address of a service an unique IP multicast group is calculated. The multicast group in RoSe is given by a prefix that is common to all multicast groups used by RoSe. The prefix is followed by the port number in network byte order and the remaining bits are specified by the last bits of the *HID*. In the example of figure 6 the multicast prefix is 226. Taking the 16 bits of the port number into account, the remaining part is 8 bits. In the example the value is 114. Using the multicast prefix of our example the number of robots is restricted to 254 and the subnet mask of the unicast IP address should cover the first 24 bits. If more robot systems should be available in the framework IPv6 should be used as an alternative, but our current implementation is limited to IPv4.

C. Service configuration and management

The RoSe framework provides flexible configuration options for its services. A service can be configured in two different ways. The first method is to configure a service with a configuration file, which is stored in a XML format. The second method is to configure a service with configuration messages through the network. Since the first approach would lead to many configuration files on one host the concept of a *Service Manager* was introduced. A *Service Manager* is used to control all services on a host. It is started as a normal RoSe service and can be configured as a normal service. The configuration of the *Service Manager* contains the configuration of all of its controlled services. If the *Service Manager* starts a service it also configures the service using the hosts loopback device.

In the decentralized approach, which is shown in figure 7, the services are either configured by configuration files or by a *Service Manager* that is configured by a local configuration file. The advantage of the decentralized approach is that each host can be configured without being dependant of other hosts.

In a centralized approach an additional service can be introduced. This service is called the *Configuration Manager*.

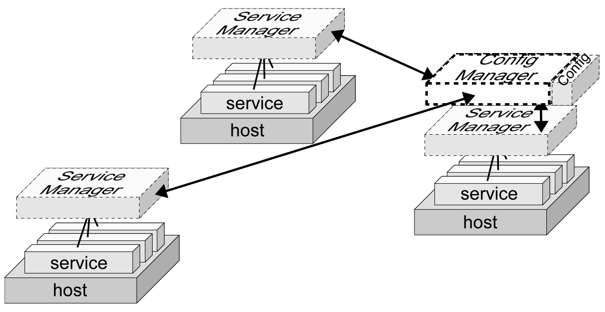

Fig. 8. Centralized architecture with Service and Configuration Managers

In this approach the services of each host are controlled by a *Service Manager*, but the *Service Manager* is configured through the network. This allows for a single configuration file on the host where the *Configuration Manager* is located. The disadvantage of this approach is that each host which should be configured must be reachable through the network. The concept of this approach is illustrated in figure 8.

Under the assumption that connections are not reliable the decentralized approach should be favored if possible, to ensure that robot systems obtain their full configuration timely.

V. Conclusion

We presented two frameworks for communication in multi robot systems. The first framework WNet tries to ensure the best connectivity that is possible. The framework operates on the MAC network layer and presents the network as an IP subnet. It is therefore transparent to upper layer protocols. The second framework RoSe is designed as an application layer middleware for robot services.

WNet has been implemented for a real testbed and a simulation environment. It can provide the application with good communication performance. Our future work will be to further improve the WNet framework, especially regarding its quality-aware routing capabilities.

The RoSe framework was presented as a novel approach for a robot middleware, which takes wireless communication aspects into account. Using this framework robot services can be implemented respecting the challenges of wireless communication but services must still be designed to cope with unreliable network communication. Future work regarding the RoSe framework will be to implement existing services for the framework and make these services robust enough for wireless communication.

References

[1] NATO IST, "Reports on 'short term realisable (multi-)robot systems in military domains', IST-052/RWS-008 and 'multi-robot-systems in military domains', IST-032/RTG-014," 2004.

[2] T. Bachran, H. H.-J. Bongartz, and A. Tiderko, "A framework for multicast and quality based forwarding in MANETs.," in *Communications and Computer Networks*, pp. 120–125, IASTED/ACTA Press, 2005.

[3] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *MobiCom '03: Proceedings of the 9th annual ACM/IEEE international conference on Mobile computing and networking*, pp. 134–146, 2003.

[4] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, (New York, NY, USA), pp. 114–128, ACM Press, 2004.

[5] H. Badis, A. Munaretto, K. A. Agha, and G. Pujolle, "QoS for ad hoc networking based on multiple metrics: Bandwidth and delay," in *IFIP/IEEE MWCN 2003*, (Singapore), Oct. 2003.

[6] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR)," Request for Comments 3626, IETF, 2003.

[7] K. J. Lee, M. S. Kim, S. Y. Cho, and B. I. Mun, "Delay-centric link quality aware OLSR," in *The IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05)*, (Los Alamitos, CA, USA), pp. 690–696, IEEE Computer Society, 2005.

[8] A. Tønnesen, "Impementing and extending the optimized link state routing protocol," tech. rep., University of Oslo, Department of Informatics, 2004.

[9] A. Laouiti, P. Jacquet, P. Minet, L. Viennot, T. Clausen, and C. Adjih, "Multicast optimized link state routing," Tech. Rep. 4721, Institut National de Recherche en Informatique et en Automatique (INRIA), Feb. 2003.

[10] SMF Design Team and IETF MANET Working Group, "Simplified Multicast Forwarding for MANET," internet draft, IETF Network Working Group, draft-ietf-manet-smf-04.txt, Mar. 2007.

[11] S.-J. Lee, W. Su, and M. Gerla, "On-demand multicast routing protocol (ODMRP) for ad hoc networks," internet draft, IETF MANET Working Group, draft-ietf-manet-odmrp-02.txt, Jan. 2000.

[12] E. M. Royer and C. E. Perkins, "Multicast operation of the ad-hoc on-demand distance vector routing protocol," in *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, (New York, NY, USA), pp. 207–218, ACM Press, 1999.

[13] B. P. Gerkey, R. T. Vaughan, and A. Howard, "The player/stage project: Tools for multi-robot and distributed sensor systems.," in *Proceedings of the International Conference on Advanced Robotics*, (Coimbra, Portugal), pp. 317–323, ICAR, June 2003.

[14] D. E. Comer, *Internetworking with TCP/IP. Principles, Protocols, and Architectures.* Prentice Hall, 2000.

[15] G. Xylomenos, G. C. Polyzos, G. P. Mähönen, and M. Saaranen, "Tcp performance issues over wireless links," *Communications Magazine, IEEE*, vol. 39, pp. 52–58, April 2001.

[16] A. Makarenko, A. Brooks, and T. Kaupp, "Orca: Components for robotics," 2006.

[17] S. Enderle, H. Utz, S. Sablatnög, S. Simon, G. Kraetzschmar, and G. Palm, "Miro: Middleware for autonomous mobile robots," 2001.

[18] C. McHale, *CORBA EXPLAINED SIMPLY.* Xhaus.com, 2004.

[19] M. Henning, "A new approach to object-oriented middleware," *IEEE Internet Computing*, vol. 8, pp. 66–75, Jan. 2004.

[20] P. Costa, G. Coulson, C. Mascolo, G. P. Picco, and S. Zachariadis, "The RUNES Middleware: A Reconfigurable Component-based Approach to Networked Embedded Systems," in *Proceedings of the 16^{th} Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC'05)*, (Berlin (Germany)), Sept. 2005.

[21] H. H.-J. Bongartz, T. Bachran, C. de Waal, and M. Frank, "Performance evaluation of a proactive link quality based uni- and multicast routing protocol for MANETs," in *KIVS 2007: Kommunikation in Verteilten Systemen*, (Berlin, Germany), pp. 169–175, VDE Verlag, 2007.

[22] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Num. Math.*, no. 1, pp. 269–271, 1959.

[23] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint relaying: An efficient technique for flooding in mobile wireless networks," Tech. Rep. Research Report RR-3898, INRIA, Feb. 2000.

[24] R. M. Karp, "Reducibility among combinatorial problems," in *Reducibility Among Combinatorial Problems*, pp. 85–103, VDE Verlag, 1972.

[25] T. Bachran, H. H.-J. Bongartz, and A. Tiderko, "A framework for multicast and quality based forwarding in MANETs," in *CCN '05: Proceedings of the 3rd IASTED International Conference on Communications and Computer Networks*, ACTA Press, 2005.

[26] A. Tiderko, F. Hoeller, and T. Bachran, "A framework for multicast communication over unreliable networks in multi robot systems," *to be published in Towards Autonomous Robotic Systems (TAROS 2007)*, 2007.

[27] T. Aurisch, "A daemon for multicast internet key exchange," in *28th Annual IEEE International Conference on Local Computer Networks (LCN)*, pp. 368–376, 2003.