Using a Link Metric to Improve Communication Mechanisms and Real-Time Properties in an Adaptive Middleware for Heterogeneous Sensor Networks*

Edison Pignaton de Freitas[1,2], Tales Heimfarth[2], Marco Aurélio Wehrmeister[2], Flávio Rech Wagner[2], Armando Morado Ferreia[3], Carlos Eduardo Pereira[2,4], and Tony Larsson[1]

[1] School of Information Science, Computer and Electrical Engineering, Halmstad University, Halmstad, Sweden
[2] Institute of Informatics, Federal University of Rio Grande do Sul, Brazil
[3] Military Institute of Engineering, Brazil
[4] Electrical Engineering Department, Federal University of Rio Grande do Sul, Brazil
{edison.pignaton,tony.larsson}@hh.se, {theimfarth,mawehrmeister,flavio}@inf.ufrgs.br, armando@ime.eb.br, cpereira@ece.ufrgs.br

Abstract. This paper presents middleware mechanisms to support real-time services in heterogeneous sensor networks, focusing on the evaluation of link metrics. Heterogeneous sensor networks require specific QoS (quality of service) guarantees in order to allow the coordination and cooperation among the different nodes that compose the system. In order to improve QoS, one of the first steps is to enhance the usage of the communication links, aiming at a more reliable and efficient message exchange. In this paper, key middleware features to address this concern are presented, in which a focus is given on the use of a link metric that, as part of a protocol, is used to optimize the message forwarding in relay communications across the network. Additionally, preliminary results are also presented.

Keywords: Heterogeneous Wireless Sensor Networks, QoS enhancement, Middleware support mechanisms, Link metric.

1 Introduction

Complexity in sensor network applications is increasing due to the use of different kinds of mobile sensors, which provide more advanced functionality and are deployed in scenarios where context-awareness is needed. In order to provide support for those emerging applications, an underlying infrastructure in the form of a middleware is necessary. The current main state-of-the-art middleware proposals, such as [1],

* E. P. Freitas thanks the Brazilian Army for the given grant to follow the PhD program in Embedded Real-time Systems in Halmstad University in cooperation with Federal University of Rio Grande do Sul.

J.H. Park et al. (Eds.): ISA 2009, LNCS 5576, pp. 422–431, 2009.

present some important drawbacks, which are mainly twofold: (i) the assumption that the network is composed only by a homogeneous set of basic or very constrained low-end sensors; and (ii) the lack of intelligence in the network, which hinders the adaptability required to deal with changes in operation conditions, e.g. lack of QoS management and control. Adaptability is a major concern that must be addressed due to: (a) long life time; and (b) deployment in highly dynamic environments. The first reason increases the probability of changes in user requirements through a systems life time, thus requiring flexibility in order to deal with changing demands. The second reason implies that applications have to be flexible enough in order to cope with drastic changes in the operation scenarios. In such environments, real-time requirements are especially hard to be met, because of variable operational conditions.

This paper presents the development of an adaptive middleware to support sophisticated sensor network applications, such as modern surveillance systems, that must adapt their behavior according to changes in their environment and in application demands. An overview of the middleware is presented, followed by some of the features for handling real-time requirements. The focus is then directed to the mechanisms that support these features, which is done by the use of a link metric that rates links and avoids the use of unstable or low-quality transitory communication links, thus reducing the negative impact from the dynamics of the topology. Preliminary results related to the link metric are also presented.

The remaining of the text is organized as follows. Section 2 presents the application scenario in which the middleware will be deployed and the main related issues. Section 3 presents an overview of the middleware structure, while Section 4 presents selected middleware features. In Section 5 details about the used link metrics are provided. Section 6 discusses related work, and Section 7 concludes the paper and gives directions for future work.

2 Application Scenario and Overview of Key Issues

The present work aims at contributing in the domain of modern surveillance systems. In this context, a sensor network composed by heterogeneous sensor nodes with different characteristics and capabilities is used. It is possible to indentify three dimensions when considering the heterogeneity of nodes: computer platform, sensing capabilities, and mobility. The first and second ones are closely related, considering rich sensor nodes using powerful sensor devices like radar, visible light cameras, or infrared sensors that are supported by moderate to high performance computing and communication resources. Low-end sensor nodes are those with constrained capabilities, such as piezoelectric resistive tilt sensors, with limited processing and communication support. The third dimension considers that sensor nodes can be static on the ground or can move, e.g. carried by a vehicle on the ground or flying at some height over the target area.

Surveillance systems benefit from the interaction among heterogeneous sensors linked in a network, increasing the capability of data gathering and fusion and enhancing their efficiency by efficiently allocating resources according to the needs of a specific surveillance mission. For instance, in order to save resources, sophisticated and expensive sensors like radars can be deployed in mobile platforms and used on

demand. When alerts are issued by low-end, cheap sensor nodes such as piezoelectric tilt sensors, the use of the more expensive high-end mobile sensors is triggered in this area. Another possibility is that a fleet with a number of autonomous, small Unmanned Aerial Vehicles (UAVs) is used to survey an area in cooperation with low-end ground nodes in order to give faster response to events triggered by the ground nodes. This implies the need for a tradeoff between the quality of the data that a node may provide and the time to respond to an event. A fleet of small UAVs may not provide accurate data as a resource rich and larger UAV carrying a more advanced sensor device, but, as they are more numerous, it is likely that one of the small UAVs of the fleet will be close to the area where the event occurred and can come faster to that area. In any of these cases, the deployed system has different problems to solve in order to make the entire network work properly, allowing the cooperation among individual nodes.

Focusing the attention on the low-end ground sensor nodes, they have a constrained energy source and must spend it carefully in order to be able to run for a longer time. Communication is a key energy consumer in this case, so it is important to communicate as seldom as possible. The same holds for the small UAVs, which should also handle their energy resource carefully in order to not deplete it quickly. However, messages exchanged in the network must comply with QoS requirements, due to application or control mechanisms needs, thus demanding a strategy to optimize the use of communication links. Moreover, dynamic adaptations in the communication must take place due to changes in the network topology, which occur as nodes may come in and out of the network, as well as when their QoS status varies.

On the other hand, most of the richer nodes handle more sophisticated data, such as radar images and high resolution video. These nodes may require more sophisticated network handling, such as bandwidth reservation and more complex QoS priority handling over certain transmissions. These issues hold for the small UAVs as well, but resized for the quality of the data that they handle.

3 Middleware Overview

In order to support the dynamic nature of the operation scenarios in which the sensor networks are to be deployed, we propose the use of a set of mechanisms combined in a middleware that can address the required adaptation. These mechanisms are part of different services provided by the middleware, and many of them are supported by the use of a link metric. This metric rates the links, according to some parameters that will be presented further in Section 5, making possible the utilization of good quality links for the communications and preventing the use of transitory or bad communication channels. The middleware is divided in three layers, according to the goals of the services provided in each of them.

The bottom layer is called Infrastructure Layer. It is responsible for the interaction with the underlying operating system and for the management of the sensor node resources, such as available communication and sensing capabilities, remaining energy, etc. A component in this layer implements part of the link metric and is responsible for collecting the signal strength indication from the incoming packets as well as the bit error rate.

The intermediate layer is called Common Services Layer, which provides services that are common to different kinds of applications, such as cross layer QoS negotiation and control, message delivery control, among other. An additional component is responsible for keeping the abstract information about the link metric for the direct neighbors of the node. This module is used by other services to increase the communication quality. It uses the information captured from the network interface in the Infrastructure Layer to update the link metric (presented in Section 5).

The top layer is called Domain-Services Layer and has the goal of supporting domain specific needs, such as data fusion support and specific data semantic support to allow the production of application-related information by processing of raw data. Other reasoning capabilities related to the management of the sensing missions are also hosted in this layer. For more details about this topic readers are referred to [2].

4 Selected Middleware Features

The proposed middleware is based on the publish-subscribe paradigm and inspired on the Data Distribution Service for Real-time Systems (DSS) specification, standardized by OMG [3]. Although being inspired by this standard, the middleware does not follow the whole specification. It takes the standard as a guideline but uses other mechanisms to address the problems highlighted above and others discussed in [4]. In the following, some selected features of the middleware, those more related with the use of the link metric, are presented. For a complete list of the middleware features concerning real-time handling, interested readers are referred to [4].

- **Flexibilibility**

The middleware provides full control of the communication. It does not use underlying control mechanisms available in the nodes' network layer. Instead, it provides its own communication control. This means that all parameters related to communication are controlled by the middleware, using only basic connectionless communication services offered by the network layer. The middleware handles parameters like number of retries, message priority, and memory usage for buffering and timing. Moreover, information about the error rate and signal strength of the incoming packets as well as the fusion of this kind of information into a high level metric are also concerns handled by the middleware.

This control over the communication provides more flexibility to manage the messages exchanged by each node, with direct impact on the reduction of latency.

- **QoS Control**

QoS control is performed through a contract between the data provider and the data requester. When a node publishes a data service, it informs also which QoS it is capable to offer. Nodes interested in the published data service and that also accepts the QoS offered may subscribe to the service. However, if a node is only interested in the data but does not agree with the offered QoS, it has two alternatives:

  - If the application that is requiring the data has a priority lower than other ones using the same service, it looks for another data provider;
  - If its priority is higher than other applications, it negotiates with the data provider, in order to obtain the desired QoS in spite of the bad consequences that this may imply to other lower priority applications.

Communication also plays an important role in order to provide a certain level of QoS. When a given node requires a certain QoS, the question whether the QoS can be provided depends on the availability of a suitable communication path between provider and requester. If just error-prone and overloaded links are available, a requested quality of service cannot be provided. Moreover, the link rating, explained in Section 5, is used in the middleware to select better links for the traffic of high priority applications when congestion is observed.

- **Network Decomposition**

Network decomposition is triggered by the situation of nodes within a group having a high communication flow between them and a low one outside the group. In this situation, they form a cluster in order to reduce the communication with outside nodes. A cluster-head, which is responsible for communications with nodes outside the cluster, is elected. Preferably, a cluster contains several good rated links between their members in order to prevent errors and low QoS in the communication. The rating of links is done using the link metric described later.

For the cluster-head election, a method based on a particular kind of self-organization is used: the division of labor and task allocation in swarms of social insects, described in detail in [5]. In social insects, different tasks are performed by specialized individuals. Each of the different morphological castes tends to perform a different task in the colony, ruled by a stimulus-threshold function, which is used in our case to assign the roles of cluster-head and cluster-member to suitable nodes.

The idea of the cluster-head election is that each node has probabilistic tendencies to assume this role: nodes with good connectivity and plenty of energy are very good candidates and thus have a higher probability of assuming the role. The complete description of the cluster-head election can be found in [6].

- **Data Segregation**

There are two kinds of data exchanged between nodes in the network: control data and application data. Control data are small and may not experience latency or unexpected delays to reach their destination. So, control data are segregated from application data by having higher priority to be forwarded. On the other hand, there are several kinds of application data, e.g. simple values (integers and floats), video streams, and character strings. Although this sort of data has a priority lower than control data, they must fulfill the QoS requirements of the application. Moreover, jitter is also reduced by the segregation, because the different kinds of data are handled by different buffers.

In our middleware, data segregation is improved by using the link metric rating. Therefore, depending on the priority and on the required QoS, different paths may be used between the data source and destination. High priority data and control data are always transmitted through links that better fit their needs. This brings a small error rate and high reliability to the transmissions.

5 Link Metric

In this section, we describe the mechanism used for the rating of links. As already seen, the link metric is a key feature of the middleware and is used extensively by

different other features, for example for selecting the links used for advertisements in the publish/subscriber paradigm. It is also very important for the QoS control of the middleware.

The need for an elaborated link metric arises from a very important difference between wired and wireless networks, which is the behavior of the network links. In a wired network, the links have a relatively stable quality. The parameter that has most influence on this type of link is the load of the network.

On the other hand, in ad hoc wireless networks, there are several parameters that influence the link quality. First of all, the propagation of the waves in a wireless medium is affected by phenomena like attenuation, distortion, exponential path loss, etc. Moreover, the environment is dynamic, with changing obstacles, temperature, and pressure that affect the transmission properties. The distortion caused by the physical effects over the radio waves introduces uncertainty at the receiver about the original signal, resulting in bit errors. Moreover, noise and interference lead also to reception errors.

Because the quality of a link is an important factor in a wireless network, our model is based on a link rating provided by the common service layer of the middleware. This rating reflects the "usefulness" or "quality" of a link.

However, the various properties that influence the quality of a wireless link make the task of finding the appropriate link rating a challenge. How the quality of a wireless link may change under a very uniform environment can be seen in the experiment reported in [7]. In Figure 1, a scatter plot of how links vary over distance for a collection of nodes on the ground of a tennis court is shown.

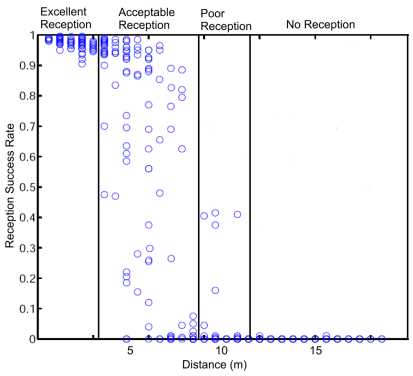

Fig. 1. Reception success rate versus distance of the transmitter/receiver (data source: [7])

Although in such an ideal environment a behavior near to the theoretical path loss curve was expected, the results depict a very different reality. After a certain distance (4m), the difference on the reception success rate between nodes at the same distance was very significant. This could be verified in the regions marked by "acceptable" and "poor" receptions (in the referred work the areas are called *transitional regions*). The labels in Figure 1 were assigned based on the average reception success rate (RSR). We can define thresholds for the lower limit of the defined regions, i.e., $RSR_{iexcellent}$, $RSR_{iacceptable}$, and RSR_{ibad}.

Besides this, many approaches are based on a bimodal link quality, where a link may exist or not. Although this may often be a true assumption for wired networks, it is not a reasonable approximation for wireless networks. Algorithms based on this simplistic assumption often choose low-capacity, long-range links instead of high-capacity, short-range links. This affects negatively the performance. This happens because bad links are good enough for control packet exchange, but during data transmissions, much of the capacity is consumed by retransmissions.

In this section, we define a link metric that summarizes the "goodness" of a link. Each link receives a real value that describes its quality. The quality of a link is estimated using the following variables: (1) Success Rate; (2) Received Signal Strength; (3) History; and (4) Energy Reserve. They are then summarized in our combined link metric.

- **Success Rate**

The idea is to use past samples of the success rate in order to estimate the quality of a link. Success rate is a relatively reliable method to predict the quality of a link. Nevertheless, there are also some drawbacks: at the beginning of the observation, there is no data to be used for the prediction; moreover, it reacts slowly to changes in the topology (a node has moved but the link rating still indicates a good link). In addition, very old measures can not accurately estimate the current situation.

- **Received Signal Strength**

The received signal strength indication (RSSI) as link metric is proposed as a substitute of the bimodal link metrics presented in some other approaches. The correlation between the received signal strength and the distance between two nodes is rather far from the ideal path loss curve, as presented in [8]. We argue that the signal strength may be used just as a rough indicator of the quality of the link, because, despite its low stability, it has a high agility. Therefore, we integrate it with other indicators in our combined metric.

- **History**

In the algorithms developed in this work, it is important to select trustworthy and stable links instead of newly created ones. In order to prevent the use of temporary links, an additional parameter is used in the metric. It measures how old the link is and penalizes very new links. This is especially important in networks with plenty topology changes.

- **Energy Reserve**

In a sensor network environment, the energy is a precious resource, and the pattern how energy is spent makes a real difference concerning the complete network life time. We decided to include the amount of energy of a node in the link metric to restrict the use of exhausted nodes, because the link metric tends to evaluate them worse than links between nodes with plenty of energy. This brings a more uniform consumption of energy. The energy reserve parameter of the link metric may improve a uniform energy use, especially by routing protocols.

- **Combined Metric**

As already mentioned above, the proposal combines the presented parameters in a link metric that indicates the goodness of a link. The combined metric is defined in the following equation:

$$M_{combined} = 1 - \left(k_1.M_{RSSI} + k_2.M_{RSR} + k_3.M_{history} + k_4.M_{energy}\right) \quad (1)$$

where $M_{RSSI} \in [0, 1]$ indicates the normalized signal strength indication, $M_{RSR} \in [0, 1]$ is the reception success rate, $M_{history} \in [0, 1]$ returns 0 for new links and 1 for old ones, and $M_{energy} \in [0, 1]$ returns 0 for depleted nodes and 1 for full nodes.

We present now how the sub-metrics used in the equation are calculated. The value of M_{RSSI} is adjusted upon reception of any packet (addressed to the node or acquired in promiscuous mode). An average of the received values and the current M_{RSSI} with an aging factor α is calculated, i.e., $M_{RSSI} = \alpha \cdot M_{RSSI} + (1- \alpha) \cdot AM_{RSSI}$, where AM_{RSSI} denotes the adjusted measured signal strength. The adjustment in the signal strength is done in order to improve its performance by cutting out extremes where the signal is either excellent ($RSSI_{excellent}$) or very poor ($RSSI_{verypoor}$).

The metric M_{RSR} is just the combination of the current measured reception success rate with the existing one, i.e., $M_{RSR} = \alpha \cdot M_{RSR} + (1 - \alpha)\ meas_{RSR}$. The measured reception success rate ($meas_{RSR}$) is calculated based on the monitoring of packet transmissions and correlated acknowledgment in the middleware Infrastructure Layer.

The history metric ($M_{history}$) is calculated using the number of received packets. C_{rx} is the number of received packets of the link. This counter is decremented periodically (down to 0) in order to cope with extinguishing links. We define:

$$M_{history} = \left(1, \frac{C_{rx}}{\text{stable\_link\_count}}\right) \quad (2)$$

where stable_link_count is the number of packets necessary to consider a link as fully active.

Finally, the energy reserve measures how much energy a node has, i.e. M_{energy} returns one when the battery is full and zero when depleted.

5.1 Preliminary Results

Preliminary results of the proposed link metric usage in a clustering algorithm are presented in this sub-section. In summary, two versions of the emergent clustering algorithm [6] were tested. One of them used the proposed link metric and the other one is exactly the same, but without the link metric.

The algorithm has two main phases: the first one is the cluster-head election while the second one is the selection of the cluster members. The cluster-head election is not relevant for the link metric evaluation provided herein, since both versions (with and without link metric) use the same cluster-head election procedure. The second phase is based on a membership fitness function that evaluates the suitability of a node to be member of the cluster.

Every time a node becomes cluster-head, it starts to search for suitable members using a broadcast message. When a candidate node receives this message, it calculates its own fitness to join the given cluster. This fitness function uses parameters as remaining energy, connection to the cluster (based on the link metric in one of the versions), and number of neighbors. Based on this fitness, the node starts a timer to reply to the cluster-head call. In this way, good candidates reply faster and are included in the cluster. When a node is included in the cluster, it also broadcasts the call for members' message.

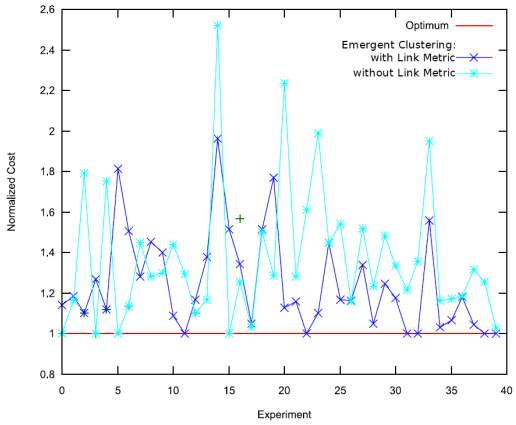

Fig. 2. Normalized results for clustering with and without use of the link metric

The difference between the two versions is on using or not the link metric in the fitness function. Figure 2 shows the results of forty simulation runs. For each run, the cost of the solution is calculated using the following procedure: the distance between all nodes inside the cluster is calculated based on the link metric; when there is no direct connection between two nodes, the shortest path is used and the distance is the sum of the link metric of this path. The cost of one cluster is the sum of the distance of all nodes to all nodes. For the complete network decomposition, the total cost is given by the sum of all individual cluster costs.

In the figure, one notices that the use of the link metric increases the performance of the algorithm considerably for most of the simulation cases. This happens because the cluster-head could select the nodes with higher link quality to be member, thus decreasing the individual cluster cost and resulting in a smaller total cost. It is important to remark that the figure shows the normalized costs.

6 Related Work

MiLAN [9] is an adaptive middleware that explores the concept of proactive adaptation, in order to respond to the needs in terms of QoS imposed by changes in the operational environment. MiLAN allows the specification of the required QoS for data, adjusting the network to increase its lifetime, by efficiently using energy. The major difference is that in our work we consider levels of quality of the link state, by using the described link metric, to choose the best nodes to forward messages. In MiLAN, there is no such fine grain consideration of the link status; instead, a bimodal link quality approach is used.

Quality Objects (QuO) [10] proposes the addition of a QoS adaptive layer on an existing middleware, such as RT-CORBA [11]. It provides means for specifying, monitoring, and controlling QoS and also for adapting the middleware behavior according to the QoS variations during runtime. However, as this framework relies on an existing middleware such as RT-CORBA, it has the same drawback regarding its use in low-end nodes. Besides, it also uses a bimodal link quality approach.

As far as we searched in the literature, no middleware uses link metrics to improve QoS in a way similar to the one presented in this paper.

7 Conclusions and Future Work

This paper presented a proposal to use a link metric to support different mechanisms to address real-time issues in a middleware for heterogeneous sensor networks. An overview of the possible operational scenarios was discussed, highlighting the need for an efficient strategy to handle and adapt the communication, considering the QoS demands. Then the middleware was presented and some of its selected features described. Finally the link metric was described and preliminary results presented.

As current work we are integrating the link metric mechanisms with the middleware services, as well as providing additional simulations to assess the efficiency of the technique in different scenarios.

References

1. Madden, S., Franklin, M.J., Hellerstein, J.M., Hong, W.: TinyDB: An Acquisitional Query Processing System for Sensor Networks. ACM Transactions on Database Systems 30(1), 122–173 (2005)
2. Freitas, E.P., Wehrmeister, M.A., Pereira, C.E., Ferreira, A.M., Larsson, T.: Multi-Agents Supporting Reflection in a Middleware for Mission-Driven Heterogeneous Sensor Networks. In: Proc. of 3rd ATSN, in conjunction with 8th AAMAS (2009)
3. Object Management Group (OMG). Distribution Service for Real-time Systems (DSS) Specification. Version 1.2 (January 2007)
4. Freitas, E.P., Wehrmeister, M.A., Pereira, C.E., Larsson, T.: Real-time Support in Adaptable Middleware for Heterogeneous Sensor Networks. In: Proceedings of International Workshop on Real Time Software (RTS 2008), pp. 593–600. IEEE, Los Alamitos (2008)
5. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Sta Fe I. Studies in the Sciences of Complexity. Oxford University Press, Oxford (1999)
6. Heimfarth, T., Janacik, P., Rammig, F.J.: Self-Organizing Resource-Aware Clustering for Ad Hoc Networks. In: Obermaisser, R., Nah, Y., Puschner, P., Rammig, F.J. (eds.) SEUS 2007. LNCS, vol. 4761, pp. 319–328. Springer, Heidelberg (2007)
7. Woo, A., Culler, D.: Evaluation of Efficient Link Reliability Estimators for Low-power. Technical Report, UC Berkeley (2002)
8. Janacik, P.: Service Distribution in Wireless Sensor Networks. Master's Thesis, University of Paderborn (2005)
9. Heinzelman, W., Murphy, A., Carvalho, H., Perillo, M.: Middleware to Support Sensor Network Applications. IEEE Network Magazine Special Issue (2004)
10. Vanegas, R., Zinky, J., Loyall, J., Karr, D., Schantz, R., Bakken, D.: QuO's Runtime Support for QoS in Distributed Objects. In: Proc. of Middleware 1998, the IFIP International Conference on Distributed Systems Platform and Open Distributed Processing (1998)
11. Schantz, R.E., Loyall, J.P., Schmidt, D.C., Rodrigues, C., Krishnamurthy, Y., Pyarali, I.: Flexible and Adaptive QoS Control for Distributed Real-time and Embedded Middleware. In: Proc. of 4th Intl Conf. on Distributed Systems Platforms. Springer, Heidelberg (2003)