A Distributed Scheduling Algorithm for Real-time (D-SAR) Industrial Wireless Sensor and Actuator Networks

Pouria Zand[1] Supriyo Chatterjea[1] Jeroen Ketema[2] Paul Havinga[1]
Pervasive Systems Group[1], Formal Methods and Tools Group[2],
Faculty of EEMCS, University of Twente, P.O. Box 217, 7500AE, Enschede, The Netherlands
{p.zand, s.chatterjea, j.ketema, p.j.m.havinga}@utwente.nl

Abstract

Current wireless standards and protocols for industrial applications, such as WirelessHART and ISA100.11a, typically use centralized network management for communication scheduling and route establishment. However, due to their centralized nature, these protocols have difficulty coping with dynamic large-scale networks. To address this problem, we propose D-SAR, a distributed resource reservation algorithm that allows source nodes to meet the Quality-of-Service requirements for peer-to-peer communication. D-SAR uses concepts derived from circuit switching and Asynchronous Transfer Mode (ATM) networks and applies them to wireless sensor and actuator networks. Simulations show that latency in connection setup is 93% less in D-SAR compared to WirelessHART and that 89% fewer messages are sent during connection setup in case the distance from source to destination is 12 hops

1. Introduction

Industrial wireless technologies such as WirelessHART [1] and ISA100.11a [2] use centralized network management techniques for communication scheduling and establishing routes. While such an approach may be easy in terms of implementation and can generate optimal results for static networks, centralized systems often perform poorly in terms of management reaction time: All updates need to be sent first to a centralized network manager (i.e. a gateway[1]) for further processing. The network manager then performs recalculations and disseminates updated instructions to the relevant nodes in the network. As the round-trip time for such decision-making actions can be very high, centralized approaches are unable to cope with highly dynamic situations (e.g. numerous link or node failures). This problem is further exacerbated as the network is scaled up.

To mitigate the above problem, the current paper presents D-SAR, a distributed scheduling algorithm for enabling real-time, closed-loop control. The distributed nature of our approach allows the system to adapt to disturbances or changes within the network in a timely manner. D-SAR focuses on allocating bandwidth resources and is based on concepts derived from Asynchronous Transfer Mode (ATM) networks. We take ATM signaling protocols [3] as a starting point, as these address certain performance issues in terms of reliability and timeliness of packet delivery, similar to what is of importance in industrial applications that require closed-loop, real-time control.

By means of simulation, we compare the resource reservation and connection establishment procedures of D-SAR, which is distributed, and WirelessHART, which is centralized, in terms of latency and message overhead. The simulations show that latency in connection setup is 93% less in D-SAR and that 89% fewer messages are sent during connection setup compared to WirelessHART in case the distance from source to destination is 12 hops. Hence, in dynamic situations, where e.g. link or node failures occur, D-SAR will establish new connections significantly faster

Section 2 provides background on circuit switching and ATM networks. We provide details on the D-SAR algorithm in Section 3. Section 4 describes a partial verification of the D-SAR algorithm and evaluates performance. Section 5 concludes.

2. Circuit Switching and ATM Networks

Large-scale, distributed, real-time control applications require data to be transmitted over long distances through a multi-hop network in a timely manner. As argued in the introduction, a distributed resource reservation algorithm is needed that allows source nodes, based on the requirements of the application and the traffic characteristics, to reserve network resources for their peer communications addressing different Quality-of-Service (QoS) needs.

Distribution will allow the system to adapt quickly to disturbances and changes within the network in a timely manner. While such distributed mechanisms do not exist for present day sensor networks, relevant techniques from other networking-related domains could potentially be adapted to develop solutions suitable for wireless sensor and actuator networks. For example, QoS in multi-hop networks could be supported by mechanisms borrowed from circuit and packet switching protocols and from the ATM protocol. Some

[1] In this paper we consider a gateway and a network manager as a single component.

of these mechanisms allow a source node to request a special end-to-end QoS for specific data flows or classes of data by reserving the resources and setting up a path between the source and destination(s).

Circuit switching is primarily designed for telecommunication networks. It establishes a dedicated link between the source and destination for the duration of communication by reserving network resources, thus guaranteeing a certain level of QoS. The reservation mechanism can play an important role in transferring real-time traffic. However, reserving routes and resources only for certain specific flows means that the routes cannot be used by other flows. In other words, the route remains reserved even if it is not being actively used. This makes it unsuitable for bursty traffic conditions. Packet switching, on the other hand, is specifically designed for delivering bursty traffic over a shared network by using statistical multiplexing, but it does not provide any QoS guarantees.

The ATM protocol uses a switching technique that combines the concepts from circuit and packet switching. For example, similar to circuit switching, before initiating data transfer, a virtual circuit is established between source and destination. This is achieved by ensuring that communication resources[2] are available at each of the nodes along the route from source to destination. Connection establishment fails if the required portion of the bandwidth cannot be allocated for any of the links along the route. The protocol also includes admission control mechanisms that help to determine whether the required QoS guarantees can be provided. ATM uses statistical multiplexing techniques, similar to those used in packet switching in order to cope with variable bit rates (i.e. bursty traffic).

Our approach is based on techniques used in ATM networks, as our ultimate aim is to develop techniques supporting both constant rate and bursty traffic. However, in this paper we only consider constant rate traffic. That is, data traffic between sensors and actuators has a constant rate.

3. The D-SAR Algorithm

As we focus on applications that require constant data rates, we allocate a virtual circuit for each traffic flow. This implies that the resources reserved for each end-to-end connection will depend on the expected traffic characteristics.

There are two separate approaches to carrying out resource reservation. One approach, based on circuit switching, is to dedicate specific communication resources in the network to particular traffic flows. The second approach, based on ATM networks, is to allow communication resources in the network to be shared between multiple traffic flows. This second approach allows for better utilization of individual communication resources and, hence, is our approach of choice to build our D-SAR algorithm upon.

We now give a high-level overview of our distributed algorithm for resource reservation D-SAR; more details are provided in [4]. The algorithm assumes that the network has already been established, that all nodes are joined to the network, and that the routing layer has constructed routes between the network nodes.

D-SAR algorithm is responsible for allocating bandwidth resources based on the traffic characteristics requested by source nodes. The message exchanges used to set up connections are similar those followed by the ATM signaling protocol [3]. The source node initiates the setup phase by sending a SETUP message. The format of this message is similar to a Contract Request in ISA100.11a or a Service Request in WirelessHART. However, unlike ISA100.11a and WirelessHART, in which a source node sends the request to a centralized system manager, in D-SAR the source node sends the SETUP message to the following node along the route to the destination (where the route was established previously by the routing layer). The message includes input parameters such as the selected bandwidth resource for communication with the next hop when communication is established, destination address, connection priority, end-to-end transit delay, traffic ID, and requested publishing period. The sender of the SETUP message sets a Timer T1 and waits for a response in the form of a CALL PROCEEDING message, which will be sent by the next node along the defined route, as shown in Figure 1.

The receiver of the SETUP message performs a check of available resources by performing an admission control operation based on requested connection parameters included in the SETUP message such as the connection priority and publishing period. This operation checks, e.g., whether the incoming resource requested by the sender is available and if any free outgoing resource to the next hop is available. If the required communication resources are available, a CALL PROCEEDING message is sent back to the sender. Upon receiving this message, the sender stops Timer T1 and starts a Timer T2. The receiver of the SETUP message forwards the SETUP message to the next hop along the route. This process continues until the SETUP message reaches the destination node. If, however, the receiver of the SETUP message is unable to accommodate the new connection, it refuses the connection by responding with a RELEASE COMPLETE message.

When the destination node receives the SETUP message all communication resources along the route are only temporarily reserved. The destination can now either accept or decline the connection request. In case the destination node accepts the connection, it sends a CONNECT message to the source node. In case a des-

[2] A communication resource can refer either to a timeslot or timeslot-channel cell depending on the data link layer definitions.

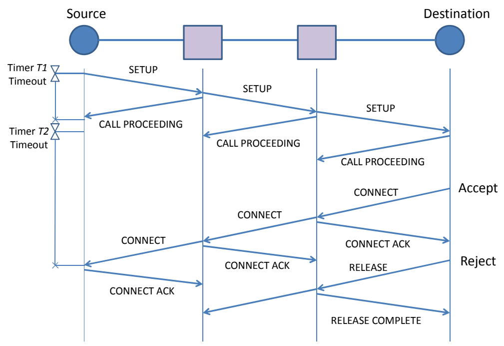

Figure 1 Overview of connection establishment protocol

tination node declines the connection request, it sends a RELEASE COMPLETE message to the source node instead.

A CONNECT message traverses along the multi-hop network back to the source node. Every intermediate node that receives the message stops Timer T2 and sends a CONNECT ACK message back to the node it received the CONNECT message from. When an intermediate node confirms the connection using a CONNECT ACK message, it switches all the temporary resource reservations over to permanent ones. Performing the reservation in two steps ensures that resource reservations are not carried through in case a connection request is unsuccessful.

We allow the network to cope with network dynamicity by preventing established connections from remaining even in case the source and destination node no longer require the connection or in case an intermediate node wishes to terminate the connection due to resource constraints. A node that wishes to terminate a connection transmits a RELEASE message. This message ensures that all nodes along the route release all the resources previously allocated for the connection.

4. Evaluation of the D-SAR Algorithm

In order to increase our confidence in the design of D-SAR, we constructed a formal specification of the connection establishment protocol in mCRL2 [5]. Using this formal specification, we were able to verify almost fully automatically that in the case of normal operation, i.e. when no message loss occurs, a connection is always eventually established and that D-SAR is deadlock free.

We implemented D-SAR and WirelessHART [6] in the network simulator NS-2 to allow for performance comparisons of end-to-end connection establishment between our approach and WirelessHART.

As the data link layer, we implemented IEEE 802.15.4e (Time Slotted Channel Hopping (TSCH) mode) [7] in NS-2 [8]. Moreover, for the routing layer we implemented the Routing Protocol for Low power and lossy networks (RPL) [9].

In our simulation, we assume that the simulation area is 150m×150m, that the transmission range is 15 meters, and that the distance between neighbors is around 10 meters. The network consists of 45 wireless nodes.

To perform the evaluation of connection establishment, 29 pairs of sensors and actuators were considered in the network. These pairs are chosen such that the total hop distance of sensor to gateway and of gateway to actuator is spread in different hop levels. Figure 2 displays a sample end-to-end connection between sensor node 37 and actuator node 45 for the D-SAR algorithm.

In WirelessHART, each sensor node sends out a Service Request to the network manager, which includes parameters such as the actuator address, publishing period, and service/connection ID. When receiving a Service Request, the network manager reserves the requested resources along an uplink graph from the sensor to the gateway and from the gateway, along a downlink graph, to the actuator. Figure 3 shows a sample connection in which the network manager has allocated the resources from sensor node 37 to actuator node 45. The network manager defines reliable routing graphs to ensure robust communication. If communication between a node and its next hop is disrupted due to interference, an alternative path can be used to transport the data. Note that the D-SAR algorithm currently does not support this kind of path diversity.

Figure 4 and Figure 5 display, respectively, the delay in establishing a connection and the number of required communications (number of messages sent) to establish an end-to-end connection for pairs of nodes based on their unique connection/service ID. Data for both D-SAR and WirelessHART is presented and connections are classified based on the total hop distance of sensor to gateway and of gateway to actuator.

As expected, the increase in total hop distance for pairs in both D-SAR and WirelessHART results in more delay, and in a larger number of communications to establish a connection. Moreover, Figure 4 and Figure 5 indicate a considerable difference in delay and required number of communications between D-SAR and WirelessHART. For example, when the total hop distance of sensors to the gateway and from the gateway to actuators comprises 12 hops, the average of the connection configuration delay is around 93% less for D-SAR compared to WirelessHART, while the average number of required communications for connection establishment is 89% less. Part of this difference can be explained by the fact that in WirelessHART the network manager has to define more links to provide a reliable uplink and downlink graph. However, the difference is mostly due to the difference in management approaches between D-SAR and WirelessHART. Where D-SAR relies on a distributed approach, WirelessHART makes use of a centralized management

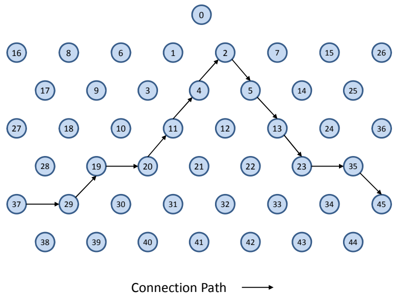

Figure 2 An end-to-end connection between nodes 37 and 45 in D-SAR

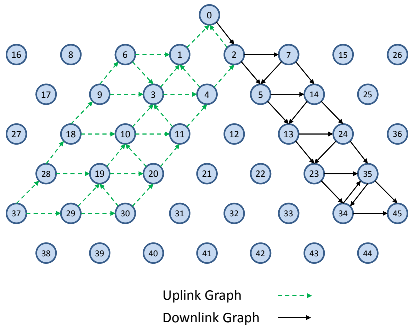

Figure 3 An end-to-end connection between nodes 37 and 45 in WirelessHART

approach, which is far more expensive in terms of time and resources.

5. Conclusion

In this paper we proposed D-SAR, a distributed resource reservation algorithm as an alternative to the centralized approached of ISA100.11a and WirelessHART. As WirelessHART can construct optimal schedules, it is a good choice for networks with low dynamicity. However, when the dynamicity of the network increases, the centralized solution becomes inefficient.

D-SAR is developed to cope with dynamic situations. The algorithm uses concepts from ATM networks to fulfill real-time requirements. Since the protocol uses a distributed approach, it needs less time to (re-)establish connections, as supported by the simulations we performed. As such, D-SAR can cope with disturbances or changes within the network in a timely manner, and large-scale networks can also be better supported. In addition, the use of temporary connections in D-SAR, which may be terminated at any time, also ensures the algorithm can cope better with network dynamicity and disturbances in the network.

Acknowledgement: This work is supported by WiBRATE: "Wireless, Self-Powered Vibration Monitoring and Control for Complex Industrial Systems" (wibrate.eu), a Collaborative Project supported by the European 7th Framework Programme, contract number 289041.

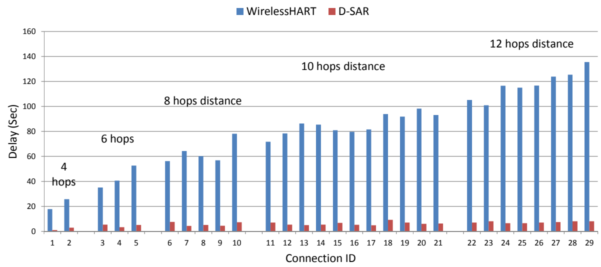

Figure 4 Connection establishment delays (D-SAR vs. WirelessHART)

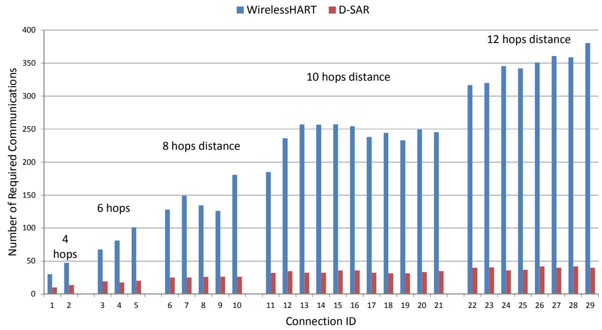

Figure 5 Number of required communications for connection establishment (D-SAR vs. WirelessHART)

References

[1] Industrial communication networks - Fieldbus specifications, WirelessHART communication network and communication profile IEC/PAS 62591 Ed. 1.0, Publication date: January 22, 2009.

[2] Wireless Systems for Industrial Automation: Process Control and Related Applications, International Society of Automation (ISA) Standard 100.11a, Draft 2a, 2009.

[3] Harry G. Perros, Ed., “Connection-oriented Networks SONET/SDH, ATM, MPLS and Optical Networks”. Wiley; 1 edition, May 6, 2005.

[4] P. Zand, S. Chatterjea, J. Ketema, and P. Havinga, “D-SAR: A Distributed Scheduling Algorithm for Real-time, Closed-Loop Control in Industrial Wireless Sensor and Actuator Networks”. Technical Report.

[5] J. Groote, A. Mathijssen, M. A. Reniers, Y. S. Usenko, M. Weerdenburg, “The Formal Specification Language mCRL2,” Proc. of Methods for Modelling Software Systems, 2007, Dagstuhl Seminar Proceedings, 06351

[6] P. Zand, A. Dilo, and P. Havinga, “Implementation of WirelessHART in NS-2 simulator”. ETFA 2012, 17-21 Sept. 2012.

[7] IEEE Standard for Local and metropolitan area networks--Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer, Date of Publication: April 16 2012.

[8] The network simulator ns-2. http://www.isi.edu/nsnam/ns/.

[9] Winter, T., et al. (2011) RPL: IPv6 Routing Protocol for Low power and Lossy Networks - draft-ietf-roll-rpl-19: http://tools.ietf.org/html/draftietf-roll-rpl-19.