Modelling Routing Algorithms for Wireless Sensor Networks in *Creol*★

Wolfgang Leister[1] and Joakim Bjørk[2]

[1] Norsk Regnesentral, Oslo, Norway
[2] Institute of Informatics, University of Oslo, Norway

We show how to use *Creol* [3] to model routing and forwarding algorithms used in wireless sensor networks (WSN) [1]. A best-practice modelling in *Creol* as well as how to evaluate highly dynamic communication systems are shown at the example of the AODV routing algorithm. The purpose of this research is to evaluate the suitability of *Creol* to simulate and model-check larger models.

Background

Creol is an *object-oriented* modelling language, used to provide an abstract, and at the same time executable model of the implementation for individual components. *Creol* is part of the *Credo* methodology [2] that unifies several simulation and model checking tools. The *Credo* tools support an integrated formalism to model different aspects of highly re-configurable distributed systems, in terms of structural changes of a network, or changes of components.

A WSN consists of a number of spatially distributed autonomous sensor nodes that are communicating wirelessly. Each sensor node in a WSN is an electronic device that contains components to perform the tasks of sensing, processing, sending and receiving of sensor data. These sensors form a network, where data packets from several source nodes are forwarded towards a sink node, possibly via other nodes that serve as a forwarding device. In each node routing and forwarding algorithms are employed in order to contribute to transporting sensor data from the sender to the sink node. Routing is the process of selecting paths in a network along which to send network traffic.

Certain aspects of WSN have been studied previously using an experimental subset of the *Creol* runtime system [4]. Inter- and intra-component communication as well as modelling of time have been in focus there. Here, we emphasise modelling of forwarding and routing across several nodes.

AODV *(Ad hoc On Demand Distance Vector)* [6] is a reactive routing protocol that builds up the entries in the routing table only if needed. AODV can adapt to topology changes of the network. When a node wants to send a message to a sink node whose next hop cannot be retrieved from the routing table the node initiates a route discovery procedure by broadcasting a route request message. Other nodes receive this message and send route reply messages if they know a route to the destination. These messages contribute to build the routing tables. If any link in a route breaks due to node mobility, wireless channel interferences, etc., the node which detects a broken link will inform the source node, and the routing tables will be updated.

Modelling AODV in *Creol*

We created a model in *Creol* with the nodes and the network of a WSN as objects, depicted in Fig. 1. The model represents both flooding, and routing tables built with

★ This research is in the context of the EU project IST-33826 CREDO: *Modeling and analysis of evolutionary structures for distributed services.*

AODV as forwarding strategies. All objects have their own behaviour, and communicate with other objects. The internal structures, i.e., controller, sensors, and radio are manifested as a set of routines rather than objects.

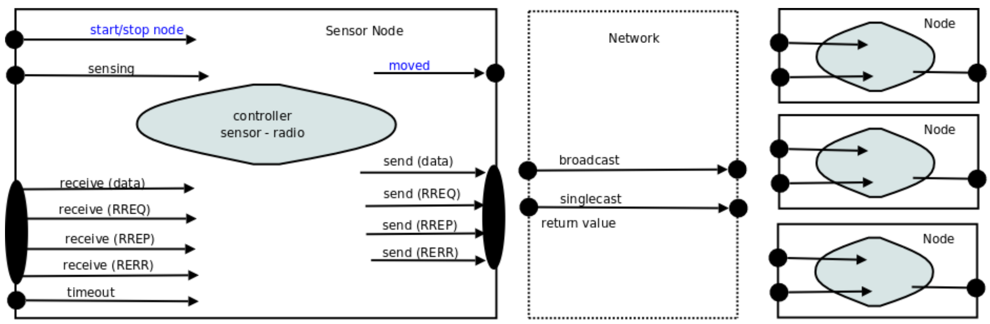

Fig. 1. Objects and their interfaces for Flooding and AODV

A model of simple flooding needs only to implement one message type with a rather simple structure. In contrast, AODV needs to handle four different message types, and additionally must provide storage space in the nodes for messages that need to wait for a route. Additionally, timeouts, failing nodes due to energy consumption, and the outcome of singlecast messages not delivered to the peer are modelled.

We have designed both untimed and timed versions of our model. The timed versions have local clocks in each node, and one clock in the network, which are synchronised in a similar manner as in the above mentioned model [4]. In the timed models we may reason about properties like collisions and timing; the untimed models contain fewer states, and are suited to reason whether routes are constructed properly, and messages arrive.

Experiences

The model of AODV in *Creol* is about 1400 lines of code, which is rather large for a model. However, different real world implementations of AODV are of about 5000-9000 lines of code [5]. The AODV algorithm contains many details to be handled for each of the four message types, which contributes to the code length. Additionally, some syntactical reasons in the current version of the *Creol* language, and the lack of scoped variables contribute to rather lengthy models.

We were able to create an executable formal model of AODV which represents by far the largest *Creol* model representing a real-world case study. Executing the model gives an insight into the mechanisms of AODV, especially regarding parallel execution of events in sensor nodes.

While modelling we found it rather easy to start with modelling in *Creol* once the run-time system and compiler were installed on the computer. We found a reasonable selection of language constructs suitable for the modelling task. As the *Creol* language is under development, we suggest as new features the possibility to define new data types and functions, as a means to both shorten the code and make programming more convenient. The underlying run-time system does not handle indeterminism in a satisfactory way in the sense that when simulating, we get the same execution path each time. Therefore, being able to control the execution from *Creol* both for simulation and model checking would be advantageous. We are not yet able to model check properties of the model by the underlying model checker, due to the large state space. The possibility to specify code fragments that need not to be model-checked, e.g., code being used to verify conditions which are not part of the analysis, would help us reduce the state space. We also conclude that building the timed model would be easier if *Creol* had timed objects, and several suggested extensions of *Creol* with time are dicussed.

Though the model of AODV for WSN resulted in a rather large model, measured in state space, lines of code, and number of objects, we could gain valuable experiences with modelling in *Creol* for real-world examples. For model-checking the state space is a clear limitation. As a next step of our research we are going to evaluate properties connected to functionality under various conditions, including changing topology, failure of nodes, energy consumption, time consumption, and collisions.

References

1. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, 2002.
2. I. Grabe, M. M. Jaghoori, B. Aichernig, T. Blechmann, F. de Boer, A. Griesmayer, E. B. Johnsen, J. Klein, S. Klüppelholz, M. Kyas, W. Leister, R. Schlatte, A. Stam, M. Steffen, S. Tschirner, X. Liang, and W. Yi. Credo methodology. Modeling and analyzing a peer-to-peer system in Credo. In *Proc. 3rd International Workshop on Harnessing Theories for Tool Support in Software, TTSS'09, Kuala Lumpur, Malaysia*, 2009.
3. E. B. Johnsen and O. Owe. An asynchronous communication model for distributed concurrent objects. *Software and Systems Modeling*, 6(1):35–58, 2007.
4. E. B. Johnsen, O. Owe, J. Bjørk, and M. Kyas. An object-oriented component model for heterogeneous nets. In F. S. de Boer, M. M. Bonsangue, S. Graf, and W. P. de Roever, editors, *FMCO*, volume 5382 of *Lecture Notes in Computer Science*, pages 257–279. Springer, 2007.
5. M. Musuvathi, D. Y. W. Park, A. Chou, D. R. Engler, and D. L. Dill. CMC: a pragmatic approach to model checking real code. In *OSDI, Usenix*, 2002.
6. C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561 (Experimental), July 2003.