KNOWLEDGE-THEORETIC PROTOCOLS FOR WIRELESS SENSOR NETWORKS

Ioanna Symeou*

Real Time Systems Group
Department of Computer Science
University of York
UK
ioanna.symeou@cs.york.ac.uk
Fax: +44 1904 432789

Keywords: Wireless Sensor Networks, knowledge, protocols.

Abstract

The need for novel abstractions for Wireless Sensor Networks inspired the exploration of possible knowledge theoretic models for these networks. Abstracting Wireless Sensor Networks as knowledge theoretic multiagent systems could enable the systematic design and verification of protocols, by examining whether a required state of knowledge is reached in the network. Such an abstraction can also be used to derive certain properties and limitations for classes of protocols. The use of a knowledge theoretic model for Wireless Sensor Networks is thus investigated, along with examples of how such a model would facilitate protocol design and verification. An example knowledge-based protocol is examined, and evaluated through experimental simulation and off-line analysis.

1 Introduction

Sensor networks have been used in industry for many years now. Systems such as smoke detectors, CCTV cameras and so on, are forms of sensor based monitoring systems. Recent advances in wireless communications, MicroElectroMechanical systems, and the miniaturisation of energy capacity have enabled the emergence of Wireless Sensor Networks (WSNs). WSNs in general are viewed as a unique ad hoc distributed monitoring system. Groups of small nodes equipped with sensors, monitor the physical environment for some user defined events of interest, and use wireless communication to send and receive packets.

These networks differ from other kinds of ad hoc distributed systems since they have very limited resources, can be quite unpredictable due to their close relation with the physical environment, are more prone to failures, and use a data centric communication paradigm [1]. Research has focused on designing and implementing new protocols and services for WSNs that suit their special requirements. It is nevertheless equally important to provide novel design and modelling methodologies for these networks, in order to efficiently abstract all unique aspects of WSNs.

One such aspect is the notion of *knowledge* of the nodes in the network. Since a WSN is a distributed information gathering system, it seems natural to model and abstract what individual or groups of nodes in the network know about the monitored area and about each other. Nodes can thus be abstracted as *knowledge agents* and the network as a *knowledge-theoretic multiagent system*. Nonetheless, a potential drawback of such a methodology is the complexity of the models required to manipulate knowledge. The knowledge models could have quite a large overhead for knowledge-based WSN protocols. An example of such a protocol has been considered by the author, as an attempt to examine and investigate the benefits and drawbacks of this approach.

The rest of this paper is organised in the following manner. Section 2 provides some background information about related work in this area. In section 3, some of the basics of the knowledge theoretic methodology used are presented. Different potential uses of knowledge theoretic WSN protocols are explored in section 4. Section 5 discusses an example of a knowledge based WSN protocol and investigates its performance through experimental simulation and analysis of its memory requirements overhead. Finally, some conclusions and areas of future work are included in section 6.

2 Related Work

Knowledge-theoretic principles are used to analyse and model distributed message passing systems and prove properties such as: the infeasibility of simultaneous action under certain error conditions, the impossibility to guarantee termination of distributed knowledge-transfer protocols, upper message bounds for knowledge communication [6, 7, 11]. This methodology has also been used to verify and enhance various message transmission protocols [8], including the Internet's Transmission Control Protocol [19].

In WSNs quite a lot of research has been performed on enabling sensor nodes to form *beliefs* about the environment and occurring events. Forming beliefs in WSNs is based on *estimation theory* [15, 22]: given a set of observations and

*Supervised by Prof. Alan Burns

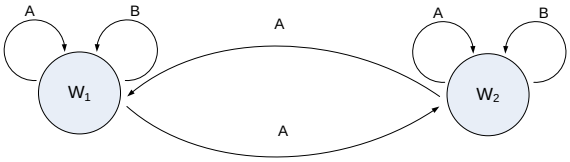

Figure 1: Kripke structure representing a system with two agents.

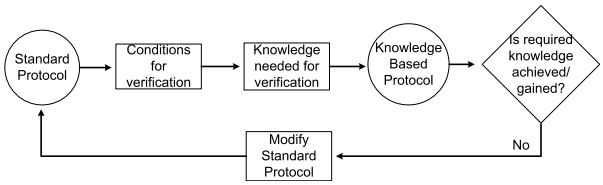

Figure 2: Designing and verifying KBPs.

hypotheses, estimate which hypothesis is true. However, to the best of the author's knowledge, explicit representation of knowledge and belief about *other* nodes has not been considered, neither have WSN protocols been extensively analysed from a knowledge-theoretic perspective. The need to model knowledge in a WSN was hinted at by Shi et al. [17]. They built a model of a WSN as a Bayesian network in order to examine the uncertainty knowledge in the network. However, Shi et al. only investigate the knowledge of a node about certain events, and not knowledge about its group. Recently, Cardell-Oliver et al. [5] have pointed out the need to use novel abstractions for WSNs to implement event detection software, and they developed a programming framework using spatio-temporal logic. They mention that it is possible to abstract nodes as agents in a knowledge-theoretic system, but do not elaborate the idea any further.

3 Epistemic Logic

To model and reason about knowledge in multiagent systems many researchers [6, 9, 14] use epistemic logic, introduced by Von Wright in 1951 [21].

3.1 Kripke Structures

One of the fundamentals of epistemic logic is the notion of a world [9] an agent considers possible in a system, according to a specific protocol. These possible worlds are usually represented by graph based *Kripke structures* [12]. The vertices of a Kripke structure represent different possible worlds, and the edges correspond to accessibility relations showing which worlds the agents consider indistinguishable. Moreover, it is trivially true that for any agent, any world is indistinguishable to itself. Consider for example two nodes A and B which exist in an environment with lossy communication. The nodes follow a protocol such that whenever a node detects an event it must broadcast a message. Assume that A detected an event, but did not receive a message from B. There are thus two possible worlds considered by node A: w_1 in which node B detected the event and its message was lost, and w_2 where B did not detect the event at all. The Kripke structure for this system is shown in Fig.1.

3.2 Knowledge Formulas

Epistemic logic uses the same formulas as propositional logic, with the addition of some knowledge formulas [6]. Of relation to this paper are the formulas of knowledge and common knowledge, $K_A\varphi$ and $C_G\varphi$ respectively. $K_A\varphi$ denotes that an agent A knows a fact φ. For any Kripke structure and in any world w, formula $K_A\varphi$ holds if and only if φ holds in all worlds accessible from w for agent A. In Fig.1 for example, considering φ to be "node B detected the event" then φ holds in w_1 but not in w_2. Since both worlds are indistinguishable for node A, then in neither w_1 or w_2 node A knows φ, meaning that $K_A\varphi$ does not hold. In order for $C_G\varphi$ to hold in a world w of a Kripke structure, then in w it must be true that all agents in group G know that all agents in G know that all agents in G know ... an infinite number of times, that φ holds. Common knowledge can be easily captured in a Kripke structure, by requiring that in all worlds of the structure φ is true.

3.3 Message passing systems

A message passing system is abstracted as a set of N agents, and a set of runs the system can perform [6, 7]. At each point in a run r, meaning at time t of r, any agent has a *local state* according to the events[1] it registered in the run thus far, such as "receive message" and "send message". Systems are represented by Kripke structures, by taking the worlds to be points in runs. Two points in a run are indistinguishable for an agent when its local state is the same in both points. Considering again Fig.1, worlds w_1 and w_2 are are indistinguishable for node A because in both worlds its local state is the same, which is *¡"detection of event","transmission of message"¿*. Modelling systems as Kripke structures enables researchers to reason about the agents' knowledge at different points in the runs. WSNs could therefore be defined in a similar manner to message passing systems, by making appropriate changes to the model used such as including "detection" events in the local state [3].

4 Knowledge-based protocols for Wireless Sensor Networks

Using a knowledge-based system model for WSNs, one can identify the state of knowledge of individual nodes, or groups of nodes, at points in the system run where a condition required by a protocol is met, and thus the protocol is verified. A Standard Protocol (SP) for WSNs can be viewed as a Knowledge-Based Protocol (KBP), where knowledge tests are performed to examine if certain formulas hold, and thus the desired knowledge state is reached. If at certain points some knowledge formulas do not hold, then the protocol can be enhanced so that the necessary knowledge is obtained. Fig.2 shows the process of designing and verifying protocols using knowledge-theoretic principles.

[1]The term "event" here is used in the sense of system related events an agent locally observes, and is not to be confused with the environmental events detected by sensor nodes.

4.1 Protocol verification and enhancement

This section provides some examples of how certain groups of WSNs protocols can be verified and enhanced through the use of knowledge theoretic principles.

4.1.1 Tasking protocols

Tasking protocols for WSNs generally deal with which nodes in an area will be tasked to perform certain actions, such as sensing or relaying messages [4, 10]. The nodes that are chosen are those that meet certain criteria, such as maximize information gain, or proximity to a detected target. Such tasking protocols are thus verified if at any point of a run after tasks have been assigned, for any node n_i which should perform a task T the formula $K_{n_i}\varphi$ holds, where φ is "task T must be performed". If therefore a node does not receive information about which task to perform or cannot deduce this information, the protocol is not verified. Since in general any message can be lost in a WSN, an enhancement for these protocols would be to enable nodes to examine their knowledge and decide by themselves when they should perform certain tasks. Therefore, for any node n_i the formula $K_{n_i}\varphi$ would hold.

4.1.2 Routing protocols

WSN routing protocols try to find an optimal path to route messages between source nodes and gateway nodes gathering data [2, 22]. Which path is optimal depends on certain user defined criteria, such as which is the fastest path, or the least costly in terms of energy consumption. Routing protocols are thus verified if after a certain point in the system run there is always an optimal path from gateway nodes to sources and vice versa. For any node n_i on that path $K_{n_i}\varphi$ holds, where φ stands for "send message m to neighbour n_j". It thus follows that if at some point a node on the path does not know where to send messages, the protocol is not verified. An enhancement would be to enable nodes to use their knowledge to dynamically decide on the next-hop neighbour thus maintaining an optimal path. At any time therefore, there will be a path from the sources to the gateways such that for any node n_i on the path, $K_{n_i}\varphi$ would hold.

4.1.3 Aggregation protocols

To reduce the flow of data in the network, several aggregation protocols are used so that certain leader nodes fuse, or aggregate data packets [13, 22]. The leader node must wait to receive data messages so that the information quality in the aggregated message meets some criteria. If for example the leader node must aggregate the received data and send the maximum value, it must wait to receive from all nodes to decide which value to forward. Aggregation protocols are thus verified if at certain points in the run, and for any leader node n_i the formula $K_{n_i}\varphi$ holds, where φ is "information quality is met". If therefore certain packets are lost it is possible that the aggregation protocol is not verified. To thus enhance aggregation protocols, leader nodes could use their knowledge to deduce whether or not the required information quality is met, and if it is necessary to wait for any more packets. If for example the leader must forward the maximum value and uses its knowledge to deduce that the packet with the maximum value was received, then the required information quality is met and there is no need to wait for other messages.

4.2 Protocol properties

With the use of knowledge-theoretic principles it is also possible to prove certain properties for classes of WSN protocols. For example, any protocol that would require a group of nodes to perform some action such as transmit a message, or perform sensing, can be analysed using group knowledge formulas such as $C_G\varphi$. It can be shown that nodes in a group running a certain protocol must all have common knowledge of some protocol related formulas [3, 7]. Taking into account that message reception, or event detection cannot generally be guaranteed in a WSN, it can be shown that $C_G\varphi$, where φ is a protocol related formula, never holds and hence some protocols cannot be verified [3]. For instance, protocols requiring some sort of coordinated action amongst nodes cannot be verified, since it cannot be guaranteed that all nodes will receive the message or detect the event that initiates the action, and thus common knowledge of some protocol related formulas is never reached.

It is thus implied that several protocols requiring some sort of action from a group of nodes are effectively best effort, since no guarantees can be made for the verification of these protocols. It is possible nonetheless for designers to provide probabilistic guarantees for protocol verification. If for example the verification of a protocol depends on the group of nodes receiving a message, the designer can ensure that a message retransmitted a specific number of times is eventually received with a certain probability by all nodes. Provision for such probabilistic guarantees however, implies that perhaps nodes would have to consume more battery. Protocol designers can therefore choose depending on the requirements of the application running on the WSN, between best-effort protocols with no verification guarantees, or probabilistically verifiable protocols which could result in the nodes consuming more resources.

5 Example Knowledge-based protocol and evaluation

In this section, an example KBP for data transmission is presented. The example KBP serves as a mean to investigate: (i) if knowledge-theoretic models can be used by sensor nodes at run time to make decisions evaluating their knowledge, (ii) if such a knowledge-theoretic structure could be used to verify WSN protocols under certain assumptions, (iii) what would the performance and overhead of a simple KBP be, and (iv) how one could design a KBP based on a simple SP for data transmission.

The example KBP and the assumptions for the network were kept simple, since this work focused on an experi-

mental approach.[2] Also, the simplicity of the protocol enables the derivation of evaluation results related more to the knowledge-theoretic part of the algorithm which are less affected by the details of the underlying protocol. The KBP is evaluated through experimental simulation against a SP for data transmission, and through analysis of the protocol overhead in terms of memory and storage requirements. Due to lack of space, details about the two protocols were omitted. Full details can be found in the York University Yellow Report [20].

5.1 Network and node assumptions

The following properties were assumed for the WSN: the network consists of groups of nodes, and each group has several subgroups of N nodes, which sleep and wake up at approximately the same times. Nodes in a subgroup are *neighbours* and know each other's unique ID. It is moreover assumed that all nodes in a subgroup can detect an event, and receive a message broadcasted to the subgroup. Nodes have dual-radios [16, 18] and can use a low power radio for communicating small control packets. A message can be lost with probability p_m. Nodes periodically wake up and monitor a specific value of the environment, for instance temperature. If a node senses a change in the value over a threshold since the last time it took a measurement, it detects an event of interest. It is assumed that an event lasts long enough for all nodes in a subgroup to detect it, but a node could miss an event with probability p_e. Finally, any event is always detected by at least one node in the subgroup.

5.2 Standard and knowledge-based protocols

To provide for battery savings, it is assumed that it is sufficient for one node to send a data message for an occurring event.[3] A data transmission protocol should hence satisfy the property that *for each occurring event, exactly one node in a subgroup sends a data packet to a leader node*: if no nodes transmit, the event is not communicated, and if more than one node sends a packet the overall battery of the subgroup is needlessly reduced.

To ensure that nodes equally consume energy, in both protocols nodes take turns in sending data messages. The SP each node is running is the following: if an event of interest is detected, the node performs a *standard test* to decide based on a fixed increasing ID order, if it is its turn to send a data message. If an event however is not detected by any node the sending order is jeopardized, since the node has a different view than its neighbours of when it is its turn to send. The SP therefore cannot guarantee to satisfy the condition of one message being sent for each event, even with the assumption that every event is detected by at least one node in each subgroup.

A KBP is thus considered, taking into account *what the nodes must know* so that the condition is satisfied. Nodes evaluate certain facts using a Kripke structure-based *knowledge model*, which reflects all the possible realities (worlds) for the subgroup up to the specific round of the KBP. Each world in the model is assigned with a probability indicating how possible it is for that world to be the reality.

It was observed that to meet the protocol condition of one data message being sent for each occurred event, it is sufficient that: nodes in each subgroup which missed an event come to know, or become *aware*, that the event occurred and thus the sending order is maintained. In the KBP therefore, when nodes detect an event they broadcast a *small Low Power Message* (LPM). The LPM is thus a mean for the nodes to communicate knowledge between them. Subsequently, a node is aware of an event e if it either detected e, or it became aware of e by receiving a LPM from some neighbour that identified the event. Since it is assumed that any event e is detected by at least one node, and if it is further presumed that nodes which did not detect e receive at least one LPM, all nodes are aware of e. The KBP therefore seems to be verified if a node which did not detect an event receives at least one LPM. This is a more relaxed requirement than all nodes detecting each event, which is needed for the SP to be verified.

Since battery is a limited resource in WSNs, the KBP was designed such that the node with the most battery will transmit the data message for an occurring event. Therefore, nodes use the knowledge model to probabilistically decide who the next sender is, based on: which neighbour has the most battery, *and* if the neighbour under consideration *also knows it should sent*.[4] In each round therefore, the node with the most battery is probabilistically chosen as the sender. If there are more than one nodes with the same battery, then the one with the least ID is chosen.

Finally, the KBP is designed assuming the following fault model: it is not possible for any node to loose more than m LPMs transmitted for an event e, for any node not to receive more than m' consecutive LPMs from the same sender, for more than v nodes to miss an even, and for any node to miss more than v' consecutive events.

5.3 Protocol evaluation

The KBP was evaluated using experimental simulations and off-line analysis of the size of the knowledge model used.

5.3.1 Experimental simulation

The two protocols were simulated using Java, for a WSN with one group and one subgroup, consisting of N = 3 nodes. The small number of nodes in the simulation *does not* negatively affect the evaluation: it was desired to test if

[2] Current research is concentrating in performing knowledge-theoretic analysis and enhancement to an actual WSN protocol, with more realistic network assumptions.

[3] It is true that the message might be lost, but our main concern is to verify that a protocol satisfies a given condition, and the protocols discussed can be expanded to satisfy a condition saying that $k > 1$ messages must be sent.

[4] There is no point in deciding a neighbour will be the sender, if the neighbour itself does not decide to transmit.

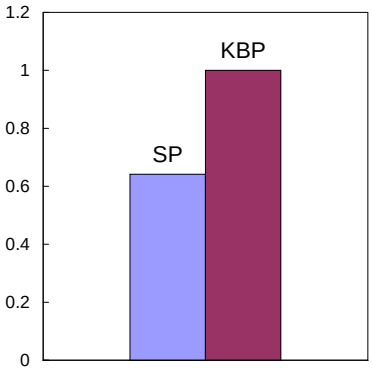

Figure 3: Distinct data messages transmitted to events occurred ratio.

each node could evaluate a knowledge model and make decisions. Under the KBP nodes behave similarly no matter what N is. This implementation corresponds to simulating a WSN of any size, with any number of groups consisting of subgroups of 3 nodes. In all test cases implemented it is assumed that an event e is detected by at least one node, and if a node n_i does not identify e, it receives at least one LPM. The results presented here are obtained from several test cases where events and messages were randomly missed, according to the fault model.

The simulation results showed that nodes successfully maintained the knowledge model and made decisions based on their knowledge. For all test cases, when a node running the KBP missed an event e, it became aware of e through LPM reception. If a node n_i was the sender node according to the sending order it would decide itself, and subsequently transmit a data message. Any other node was able to decide that n_i was the sender and thus just one data message was sent for each event occurred. Under the SP nodes could not identify if they, or a neighbour, missed an event and thus the sending order was jeopardized. In all test cases therefore several events were not communicated. It was noticed that event non-communication can propagate: the perception of the sending order of each node can change such that from a point on there are always some events which are not communicated. Therefore, even if nodes actually miss few events, quite many in effect are not communicated.

Fig.3 shows the average *distinct messages*[5] *transmitted to events occurred* ratio for both protocols. This ratio measures the effectiveness of the protocol in terms of communicating information for each event that has occurred in the network. As expected, the KBP is more efficient in communicating unique pieces of information about occurring events.

A drawback of the KBP is that nodes consume more battery due to LPM communication. Nonetheless, the *total* battery of all nodes in the group reduces gracefully under the KBP, whereas the reduction is unpredictable under the SP. With the KBP the user can be sure about the relative battery units of the nodes at any point in the run. Fig.4 shows the average

[5]Data messages are distinct if just one node sent a data message. In any round where two or more messages are sent, the number of distinct data messages is one.

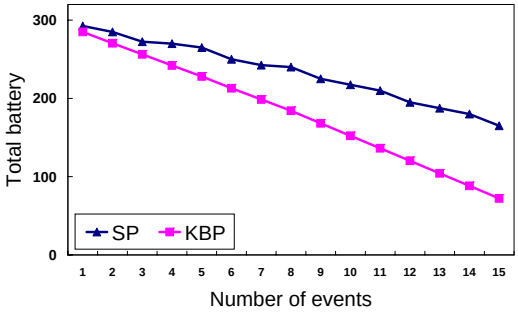

Figure 4: Total battery consumption.

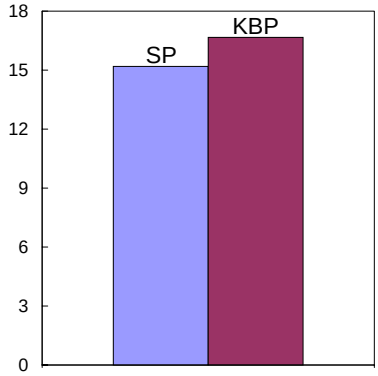

Figure 5: Energy consumed to distinct data messages transmitted ratio.

total battery under the two protocols for 15 occurred events. Considering moreover the average *total energy consumed to distinct data messages sent* ratio shown in Fig.5, it is shown that even though the KBP is more energy consuming, both protocols are similar in terms of energy being efficiently consumed for sending distinct data messages.

5.3.2 Analytic evaluation

The size of the knowledge model is key for the implementation and analysis of the KBP. It is crucial to ensure that nodes have sufficient memory to store and process the model used. The knowledge-model was based on Kripke structures and implemented as a tree with three levels. Each level corresponds to a depth of knowledge for the node. The node's knowledge was represented by several possible worlds. Details about this choice of design and the analytic evaluation can be found in the technical report [20]. Tables 1 and 2 summarize the maximum bounds for worlds in the three levels of the model (L0, L1 and L2) for various parameters of the fault model. The calculations were made for subgroups of 3 and 10 nodes.

Parameters			**World Bounds**		
			L0	L1	L2
$v = 1$	$m = 1$	$m' = 2$	2	12	4
		$m' = 4$	8	30	6
$v = 2$	$m = 0$	$m' = 2$	1	3	4
		$m' = 4$	1	4	6
	$m = 1$	$m' = 2$	6	16	4
		$m' = 4$	54	40	6

Table 1: Maximum world bound when N = 3.

Parameters			World Bounds		
			L0	L1	L2
$v = 5$	$m = 4$	$m' = 2$	126	1464	4
		$m' = 4$	126^3	5856	4
$v = 9$	$m = 0$	$m' = 2$	1	3	4
		$m' = 4$	1	4	4
	$m = 4$	$m' = 2$	131×256	1984	4
		$m' = 4$	131×256^3	7936	4

Table 2: Maximum world bound when N = 10.

This analytic evaluation shows that the knowledge model could have quite a large overhead. It thus appears that it might not be such a good design policy, for the nodes to maintain and evaluate a full-scale knowledge model at run time. An attractive alternative would be to perform protocol analysis off-line with a knowledge-theoretic model, and use the results to enhance the protocol. The nodes would not need to use a knowledge model at run time and thus save precious space and memory. Another solution would be to define light-weight knowledge-models that would store less information but be efficient enough for nodes to make knowledge-based decisions at run time.

5.3.3 Evaluation discussion

Comparing the results of the evaluation with the goals set in the beginning of this section, the following are observed. Through the use of knowledge theoretic methodologies, it is possible to develop a KBP based on a SP, where nodes evaluate a knowledge model at run time. The KBP can improve the performance of the network in terms of information transfer but with increased energy consumption. For all test cases the KBP was verified and the condition met, under the considered assumptions of event detection.[6] Nevertheless, those assumptions might not be satisfied in a real WSNs and there could be some nodes that will not be aware of certain events. It would thus be interesting to consider probabilistic protocol verification, considering more realistic assumptions. Finally, quite a large overhead is associated with the model used in the KBP. More efficient models for knowledge manipulation would thus be a possible area of future work.

6 Conclusions and Future Work

This paper investigates a novel knowledge-theoretic abstraction for WSNs. Several uses of such a representation were proposed and an example of a KBP is discussed. Even though the example KBP can improve the performance of the network, a serious criticism of the protocol is the potential large size of the knowledge-model used, since WSN nodes are severely memory and storage constrained. This demonstrates a potential drawback of designing KBPs for WSNs. Areas of future research therefore include investigation of light-weight knowledge-models that could be more suitable for resource constrained nodes.

Another potential alternative would be to perform off-line knowledge-theoretic analysis of a protocol and use the results to enhance protocol performance. This eliminates the need for the nodes to maintain a possibly large knowledge-model during run time. Future work also concentrates on performing such off-line analysis for existing and more complex WSN protocols. The development of a general knowledge-theoretic framework is thus envisioned, that will enable designers to methodically develop WSN protocols, and explore tradeoffs between protocol performance, resource consumption, and knowledge related overhead.

[6] Even though the KBP was verified for the test cases implemented, there is a proof in the York University Yellow Report that the protocol is verified for any run under the considered assumptions.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, 2002.

[2] J. N. Al-Karaki and A. E. Kamal. Routing Techniques in Wireless Sensor Networks: a survey. *IEEE Wireless Communications*, 11(6):6–28, 2004.

[3] A. Burns and I. Symeou. On the infeasibility of coordinated reaction in Wireless Sensor Networks. *Submitted to ACM Transactions of Sensor Networks*, N/A, 2006.

[4] J. Byers and G. Nasser. Utility-based Decision-Making in Wireless Sensor Networks. In *ACM international symposium on Mobile ad hoc networking & computing (MobiHoc)*, pages 143–144, Boston, MA, 2000.

[5] R. Cardell-Oliver, M. Reynolds, and M. Kranz. Space and Time Logic for Programming Sensor Networks. In *Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISOLA)*, 2006.

[6] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning About Knowledge*. (The MIT Press, 1995, 1^{st} edn.).

[7] J. Y. Halpern and Y. Moses. Knowledge and Common Knowledge in a Distributed Environment. In *Proceedings of the 3^{rd} annual ACM Symposium on Principles of distributed computing (PODC)*, 1984.

[8] J. Y. Halpern and L. D. Zuck. A Little Knowledge Goes a Long Way: Simple knowledge-based derivations and correctness proofs for a family of protocols. In *Proceedings of the 6^{th} annual ACM Symposium on Principles of distributed computing (PODC)*, 1987.

[9] J. Hintikka. *Knowledge and Belief*. (Cornell University Press, 1962, 1^{st} edn.).

[10] M. Jones, S. Mehrotra, and J. H. Park. Tasking Distributed Sensor Networks. *International Journal of High Performance Computing Applications*, 16(3):243–257, 2002.

[11] R. Koo and S. Toueg. Effects of Message Loss on the Termination of Distributed Protocols. *Information Processing Letters*, 27(4):181–188, 1988.

[12] S. Kripke. Semantical analysis of modal logic. *Zeitschrift fur Mathematische Logik und Grundlagen der Mathematik*, 9:67–96, 1963.

[13] B. Krishnamachari, D. Estrin, and S. B. Wicker. The Impact of Data Aggregation in Wireless Sensor Networks. In *International Conference on Distributed Computing Systems (ICDCSW)*, pages 575–578, Washington, DC, 2002.

[14] J. J. Meyer and W. van der Hoek. *Epistemic Logic for AI and Computer Science*. (Cambridge University Pressm 1995, 1^{st} edition.

[15] G. J. Pottie and W. J. Kaiser. Wireless Integrated Network Sensors. *Communications of the ACM*, 43(5):51–58, 2000.

[16] C. Schurgens, V. Tsiatsis, S. Ganeriwal, and M.B. Srivastava. Topology Management for Sensor Networks: Exploiting Latency and Density. In *Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2002.

[17] D. Shi, J. You, and Z. Qi. Building Graphical Model Based System in Sensor Networks. *Springer Lecture Notes in Computer Science*, 3823:218–227, 2005.

[18] E. Shih, P. Bahl, and M. J. Sinclair. Wake on wireless: an event driven energy saving strategy for battery operated devices. In *Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2002.

[19] F. Stulp and R. Verbrugge. A Knowledge-Based Algorithm for the Internet Transmission Control Protocol (TCP). *Bulletin of Economic Research*, 54(1):69–94, 2002.

[20] I. Symeou. Knowledge Theoretic Design for Data Transmission Protocols of Wireless Sensor Networks. *York University Yellow Report, YCS-2007-413*, 2007.

[21] G.H. Von Wright. *An Essay on Modal Logic*. (North-Holland Publishing Company, 1951, 1^{st} edn.).

[22] F. Zhao and L.Guibas. *Wireless Sensor Networks: An Information Processing Approach*. (Elsevier, 2004, 1^{st} edn.).