A Fault-Tolerant Model of Wireless Sensor-Actuator Network

Keiji Ozaki, Kiyohiro Morita, Naohiro Hayashibara, and Makoto Takizawa
Tokyo Denki University, Japan
E-mail {kei, kiyo, haya, taki}@takilab.k.dendai.ac.jp

In a wireless sensor and actuator network (WSAN), a group of nodes, sensors, actuators, and actuation devices are geographically distributed and linked by wireless networks. Sensors gather information for an event occurring in the physical world and send them to actuators. Actuators perform appropriate actions on actuation devices by making a decision on receipt of sensed values from sensors. Sensors are low cost, low powered devices with limited energy, computation, and wireless communication capabilities. Messages may be lost due to collision and noise and sensors may be faulty. Here, nodes are required to reliably communicate with each other in realtime manner. In order to realize the reliability, we newly propose a multi-actuator/multi-sensor (MAMS) model where each sensor sends sensed values to multiple actuators and each actuator receives sensed values from multiple sensors in an event area. Even if messages are lost and sensors are faulty, actuators can surely receive sensed values. An actuator makes a decision on what actions to be performed on what order. We discuss centralized and decentralized protocol for reliable sensor-actuator communication.

1. Introduction

A wireless sensor and actuator network (WSAN) is a collection of nodes which are interconnected in wireless networks. There are types of nodes, sensors, actuators, and actuation devices to perform distributed sensoring and acting tasks [1, 2, 12]. Sensors gather values like temperature about physical world and send the sensed values to one or more than one actuator. Actuators are capable of making a decision on actions for the sensed values and perform the actions on actuation device. WSAN is used in sensing applications like microclimate control, home automation, environmental monitoring, target tracking [1, 2]. There are many discussions on how to reliably and efficiently broadcast messages among sensors [4, 12]. WSAN is one of the most significant technologies to realize ubiquitous computing systems [15].

Sensors are low-cost, low-power devices which are equipped with limited energy, computation, and wireless communication capabilities. Sensors may stop, even malfunction [11] due to the out-of-charge and fluctuation of observed phenomena in the physical world. In addition, messages are lost due to collisions and noise in a wireless channel. It depends on a type of each node how far the node can deliver a message, i.e. the strength of radio. That is, actuators deliver messages to nodes which are more distant than sensors. In our *multi-actuator/multi-sensor* (*MAMS*) model, each sensor sends sensed values to multiple actuators and an actuator receives sensed values of a same event from multiple sensors in order to be tolerant of faults of sensors and wireless networks. Even if some messages are lost in the wireless link and are delivered to actuators in different orders, each actuator can receive proper sensed values from the other proper sensors. For example, an actuator makes a decision like majority-based decision on sensed values from multiple sensors. We discuss semi-passive [3] protocols for coordinating multiple actuators to make a decision of a value.

In section 2, we present the model MAMS model. In section 3, we discuss what problems, to be resolved in the MAMS model. In section 4, we discuss how to realize reliable sensor-actuator communications.

2. System Model

2.1. Sensors and actuators

A wireless sensor and actuator network (WSAN) W is composed nodes interconnected with a wireless network [1, 2, 12]. There are three types of nodes, *sensors*, *actuators*, and *actuation devices*. A sensor gathers values about the physical world like temperature and sends it to actuators. An actuator makes a decision on what actions to be performed on actuation devices based on the sensed values from the sensors and then performs the actions on the actuation devices. Let $\mathbf{S}$ be a set of sensors, $\mathbf{A}$ be a set of actuators, and $\mathbf{O}$ be a set of actuation devices in a WSAN W, i.e. $W = \langle \mathbf{S}, \mathbf{A}, \mathbf{O} \rangle$.

A WSAN W is partitioned into event areas, $W_1, \cdots, W_m$ ($m \geq 1$). An event area W_i is a geographical unit of a WSAN W. Each event area

W_i is composed of sensors, actuators, and actuation devices. Let $\mathbf{S}_i$, $\mathbf{A}_i$, and $\mathbf{O}_i$ show sets of sensors $\{s_{i1}, \cdots, s_{il_i}\}$ ($\subseteq$ **S**), actuators $\{a_{i1}, \cdots, a_{im_i}\}$ ($\subseteq$ **A**), and devices $\{o_{i1}, \cdots, o_{in_i}\}$ ($\subseteq$ **O**) in an event area W_i, respectively.

Phenomena in the physical world is changed on occurrence of an event. If an event occurs in some location, sensors in some distance from the location gather values of some attributes of the event. Let $SE(e)$ ($\subseteq \mathbf{S}_i$) be a subset of the sensors which sense an event e. Each event is characterized in terms of attributes like temperature and acceleration. Let Ω be a set of all the attributes of events. Let $\Omega(e)$ ($\subseteq \Omega$) show a *scheme* of an event e which is a subset of the attributes of the event e ($\Omega(e) \subseteq \Omega$). An event e is represented as a collection of values of attributes. Let *Dom(p)* be a set of possible values which an attribute a can take. Let *e.p* ($\in$ *Dom(p)*) be a value of an attribute a of an event e. For example, a tuple of values $\langle \cdots, 15[°C], N35°69'11.74"E139°22'19.33", \cdots \rangle$ shows values of an event e of a scheme $\Omega(e)$ = $\langle \cdots, temperature, location, \cdots \rangle$ where the event e occurs in Tokyo Denki Univ., Japan. *e.temperature* = $15[°C]$ and *e.location* = $N35°69'11.74"E139°22'19.33"$. There are *discreate* and *continuous* types of attributes. A discrete attribute takes discrete values like ON and OFF. A *continuous* attribute a takes a continuous value like *temperature*. A pair of continuous sensors may not take the same value, e.g. one sensor takes $15.1°C$ and the other takes $15.2°C$.

A *type* $\Omega(s_{ik})(\subseteq \Omega)$ of a sensor s_{ik} is a subset of the attributes where s_{ik} can sense. For example, a voice sensor v can gather voice information, $\Omega(v) = \{voice\}$. Let $e[s_{ik}]$ show a tuple of values for an event e which a sensor s_{ik} gathers, i.e. $\langle e.p \mid p \in \Omega(s_{ik})\rangle$. Multiple sensors sense a same event e. For a pair of sensors s_{ik} and s_{ih} in an event area W_i, a sensed value $e.p[s_{ik}]$ may not be the same as $e.p[s_{ih}]$, $e.p[s_{ik}] \neq e.p[s_{ih}]$ for an attribute $p \in \Omega(s_{ik}) \cap \Omega(s_{ih})$, e.g. due to sensor error and different sampling intervals.

Actuation devices which act to the physical world like robot arms and air conditioners are modeled to be objects [5] in this paper. An object is an encapsulation of state and methods for manipulating the state. Actuation of a device is modeled to be execution of methods on the device object. An actuator issues a method to an object. On receipt of a method, the method is performed on the device object. For example, cooler air is ventilated by the *air conditioner* object *ac* on receipt of a method (*turn*) *down*. Figure 1 shows the relations of sensors, actuators, and actuation device objects.

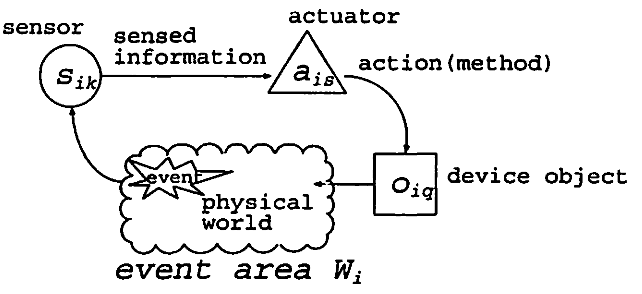

Figure 1. Sensor, actuator, and device object.

2.2. A wireless communication channel

There are multiple nodes, sensors, actuators, and device object in each event area W_i. A node sends a message in a broadcast channel of a wireless network. A message sent by a node can be received by nodes in W_i depending on the radio field intensity. Let $D(c)$ be a set of nodes which can receive a message sent by a node c in W_i. $D(c_1) = D(c_2)$ does not always hold for every pair of nodes c_1 and c_2. Every node in an event area W_i communicates with the others through a wireless channel. If multiple nodes send messages at the same time, the messages are lost due to the collision. In order to resolve the collision in the channel, some synchronization protocol like CSMA[9], CSMA/CA[7] and CSMA/CD[8] is used. For example, CSMA is used in the sensor MICA2 with TinyOS[14]. There are following properties on communication among nodes through one wireless channel:

1. If a pair of nodes c_1 and c_2 send messages at the same time, every node in $D(c_1) \cap D(c_2)$ loses the message due to the collision.
2. Let c_1 and c_2 be nodes which send messages m_1 and m_2 at the same time, respectively, in W_i. Every pair of nodes c_3 and c_4 in $D(c_1) \cap D(c_2)$ receive a pair of messages m_1 and m_2 in the same order if c_3 and c_4 receive both m_1 and m_2 [Figure 2].

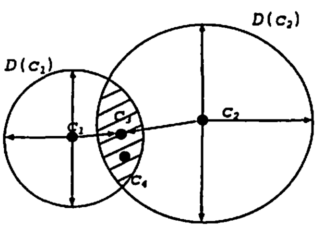

Figure 2. Contention.

A sensor s_{ik} cannot deliver messages to distant nodes due to less power supply. On the other hand, an actuator can deliver messages to distant nodes.

We make the following assumptions:

1. Each actuator a_t can deliver a message to every node in an event area W_i, i.e. $D(a_{it}) = \mathbf{A}_i \cup \mathbf{S}_i$.
2. Each sensor s_{ik} can deliver a message to only a subset od nodes in W_i, i.e. $D(s_{ik}) \subseteq \mathbf{A}_i \cup \mathbf{S}_i$.

We consider two communication models in WSAN:

1. One-channel model.
2. Two-channel model.

In the one-channel model, every node sends and receives mesages with one wireless channel. For example, if an actuator sends a message, every sensor cannot start sending messages. In the two-channel model, every pair of sensor and actuator nodes use different wireless channels while every sensor sends messages with other sensors and actuators and every actuators sends messages to sensors in one channel and every actuator sends messages to other actuators in another channel. That is, one channel is used for the actuator-actuator communication and another channel is used for sensor-sensor and sensor-actuator communication in CSMA. Actuators can communicate with each other freely from collisions with sensor-actuator communication. For example, actuators use the 11b Ethernet and deliver messages to other actuators and sensors can deliver messages to nodes.

3. MAMS Model

In the single-actuator (SA) model [1], there are one actuator and multiple sensors in an event area. If an event occurs in an event area W_i, sensors sense values of the event, like temperature. Sensors in an event area send the sensed values to one actuator. Here, the actuator may be a single point of failure. On the other hand, multiple sensors and multiple actuators in an event area are interconnected in the multi-actuator (MA) model [1] [Figure 3]. Each sensor sends sensed values to one actuator but some pair of sensors in an event area may send sensed values of an event to different actuators. If messages with sensed values from a sensor to an actuator are lost or some number of sensors are faulty, the sensed values of the event obtained by the sensor cannot be delivered to the actuator. In order for at least one actuator to make the decision, more number of sensors are required to send sensed values to the actuator.

In this paper, we propose a *multi-actuator/multi-sensor* (*MAMS*) model to realize the reliable sensor-actuator communication. Suppose a sensor s sends sensed values to an actuator a. Here, a is referred to as *parent* of the sensor s and s is in turn a *child* of the actuator a. Let $AA(a_{it})$ and $AS(a_{it})$ be sets of actuators and sensors, respectively, which can receive a message sent by an actuator a_{it} in an event area W_i. $D(a_{it}) = AA(a_{it}) \cup AS(a_{it})$. Let $SA(s_{ik})$ and $SS(s_{ik})$ be sets of sensors and actuators, respectively, which can receive a message sent by a sensor s_{ik} in W_i, $D(s_{ik}) = SA(s_{ik}) \cup SS(s_{ik})$.

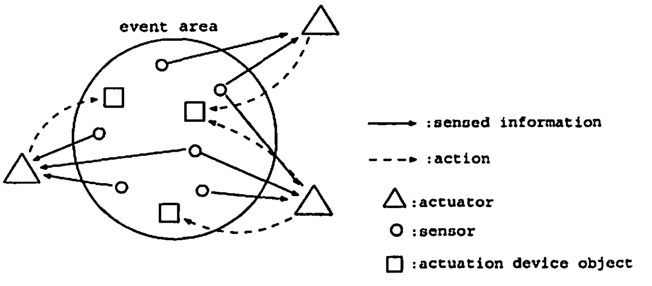

Figure 3. MAMS model.

Let $AE(e)$ ($\subseteq \mathbf{A}_i$) be a subset of the actuators which can receive sensed values sent by the sensors, i.e. $\{a_{it} | a_{it} \in SA(s_{ik})$ and $s_{ik} \in SE(e)\}$ in an event area W_i.

Each actuator a_{it} in the set $AE(e)$ may receive sensed values from multiple sensors for an event e occurring in an event area W_i. An actuator a_{it} can deliver a message to every node, i.e. actuator and sensor nodes in the event area W_i while a sensor s_{ik} can deliver a message to only a subset of the nodes. The more number of messages each actuator sends, the more number of messages may be lost due to collision. In this paper, we make the following assumptions on communications among actuators and sensors in an event area W_i:

1. Each actuator a_{it} can deliver a message to every actuator and sensor in an event area W_i, i.e. $AA(a_i) = \mathbf{A}_i$ and $AS(a_i) = \mathbf{S}_i$.
2. Each sensor s_{ik} can deliver a message to subsets of the actuators and sensors in W_i, i.e. $SA(s_{ik}) \subseteq \mathbf{A}_i$, $SS(s_{ik}) \subseteq \mathbf{S}_i$, and $D(s_{ik}) \subseteq \mathbf{A}_i \cup \mathbf{S}_i$.
3. Each actuator a_{it} is equipped with enough size of buffer but each sensor s_{ih} with only limited size of buffer.

From the assumptions 1 and 2, each actuator a_{it} receives every pair of messages sent by actuators and sensors in the same order if a_{it} receives both the messages. The assumption 2 means that even if an actuator a_{it} receives a message from a sensor s_{ik}, another actuator a_{iu} may not receive the message. The assumption 3 means even if a message arrives at a sensor without collision, the sensor may lose the message due to the buffer overflow. On the other hand, if a message arrives at an actuator, the actuator can surely receive the message.

Suppose an actuator a_{it} sends a message m_1 at the same time as a sensor s_{ik} sends a message m_2. In the one-channel model, m_2 is lost due to collision while m_1 may be delivered to some sensors and actuators

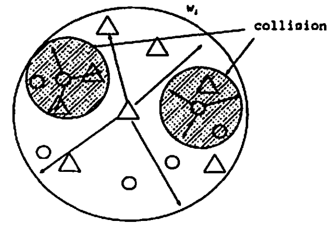

Figure 4. Collision.

as shown in Figure 4. In the two-channel model, m_1 and m_2 do not collide because the messages are sent in different channels.

Suppose a pair of events e_1 and e_2 occur in an event area W_i, which are sensed by a sensor s_{ik}. The sensor s_{ik} sends sensed values v_{ik1} and v_{ik2} for e_1 and e_2, respectively, to an actuator a_{it}. Another sensor s_{ih} senses a value v_{ih2} for the event e_2 and then v_{ih1} for e_1. The sensor s_{ih} sends the value v_{ik2} and then v_{ik1} to the actuators a_{it}. Here, a_{it} receives the values for the events e_1 and e_2 from the sensors s_{ik} and s_{ih} in different orders as shown in Figure 5. Here, suppose a method op_1 to be performed for the event e_1 and op_2 for e_2. The actuator a_{it} has to decide which method op_1 or op_2 to be first performed. If e_1 *happens before* e_2 ($e_1 \rightarrow e_2$)[10], op_1 has to be performed prior to op_2. If e_1 and e_2 concurrently occur ($e_1 \parallel e_2$), the actuator a_{it} has to decide which method to be issued prior to the other. Especially, if a pair of actuators a_{it} and a_{iu} receive the sensed values, a_{it} and a_{iu} have to take the same order of the methods op_1 and op_2.

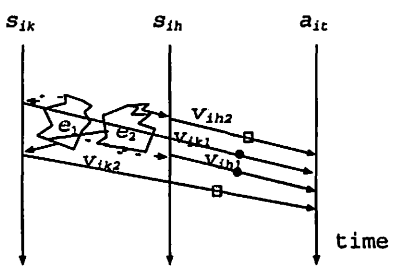

Figure 5. Order of events.

Next, suppose a pair of sensors s_{ik} and s_{ih} sense both values v_{ik1} and v_{ih2} for a pair of events e_1 and e_2, respectively. The sensors s_{ik} and s_{ih} are heterogeneous, i.e. $\Omega(s_{ik}) \neq \Omega(s_{ih})$. An actuator a_{it} receives v_{ik1} from s_{ik} before v_{ih2} from s_{ik}. Suppose $e_1 \parallel e_2$, the actuators a_{it} and a_{iu} have to make a decision which event precedes the other event.

In sensor-actuator communications, each actuator a_{it} has to make the following decisions:

1. The actuator a_{it} makes a decision on a value v_{ik} from sensed values sent by child sensors. Then, one value v is taken from a set of the values v_{i1},

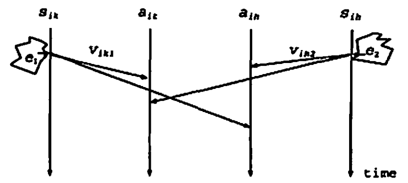

Figure 6. Order of events.

..., v_{im_i}, where each value v_{it} is taken by an actuator a_{it} (t = 1, ..., m_i).
2. The actuator a_{it} makes a decision on which method to be performed on which actuation device for the sensed value v.
3. The actuator a_{it} makes a decision on in what order methods to be issued to actuators.

Every actuator has to make an agreement on one discrete value in the domain for a discrete attribute. On the other hand, a *continuous* attribute takes a continuous value like temperature. Even if a pair of sensors s_{ik} and s_{ih} are proper, the sensors s_{ik} and s_{ih} may send different values to actuators. For example, a sensor s_{i1} sends temperature value 14.01 [°C] and another sensor s_{i2} sends 14.02 [°C] depending on the precision and position of the sensor.

4. Reliable Sensor-Actuator Communications

We discuss how the actuators cooperate to make a decision on a value for a collection of sensed values from the sensors. There are ways, active [16], passive [16], semi-passive [3], and semi-active [16] ways. In the passive way, one actuator plays a role of a coordinator. Let a_{i1} be the primary actuator. The other actuators $a_{i2}, \cdots, a_{im_i}$ are secondary in an event area W_i. On receipt of a sensed value from a sensor s_{ik}, an actuator a_{it} forwards the value to the primary actuator a_{it}. Then, a_{i1} makes a decision on a value from the sensed values collected by the actuators. The primary actuator a_{i1} performs methods for the sensed value on actuation devices. If the primary actuator is faulty, one of the secondary actuators takes over the primary one. In the active way, every actuator receives sensed values and performs same actions on devices. As discussed [13] multiple redundant executions of an action have to be resolved. In the passive and semi-passive ways, the secondary actuator neither make a decision nor perform methods. On the other hand, each secondary actuator makes a decision on what methods to be performed in what order but does not perform the methods in the semi-active way. In the semi-active and semi-passive ways, every secondary actuator has to receive the same sensed values. In order to realize the realtime communication, we take the semi-passive way since the semi-passive way implies larger availability than the passive and semi-active way.

[Semi-passive coordination protocol]

1. On receipt of sensed values from some number of sensors, an actuator a_{it} sends pairs of the sensor identifier and the sensed value from the sensor to the primary actuator a_{i1}.
2. On receipt of sensed values from sensors and secondary actuators, every actuator a_{it} makes a decision on a value and methods $op_{it1}, \cdots, op_{itk_{it}}$ for the value, v_{it}. The primary actuator broadcasts the value v and the methods $op_{it1}, \cdots, op_{itk_{it}}$ to every secondary actuator.
3. On receipt of the value v_{i1} and the methods from the primary actuator a_{i1}, the secondary actuator a_{it} check if $v_{it} = v_{i1}$ and $op_{itl} = op_{i1l}$ $(l = 1, \cdots, k_{it})$. If not, a_{it} logs v_{i1} and $op_{il1}, \cdots, op_{ilk_{it}}$.

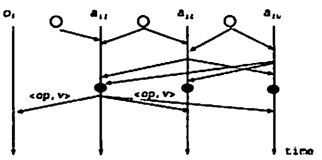

Figure 7. Sensor-actuator communication.

Since every actuator is interconnected with one wireless channel, every actuator can receive a message if some actuator sends the message. In addition, every actuator can receive messages in the same order. Each secondary actuator a_{it} can just receive the method *op* and the value v without making the decisions n the value and the method.

The primary actuator a_{i1} may stop by fault. We assume every actuators can recognize that another actuator is faulty by using some kind of fault detector [6]. then, one of the operational secondary actuator s selected, e.g. an actuator whose identifier is the minimum. If a secondary actuator is faulty, the faulty actuator is removed by the other actuators.

5. Concluding Remarks

In this paper, we discussed how to make a wireless sensor and actuator network (WSAN) fault-tolerant where sensor and actuator nodes stop by fault and messages are lost. A WSAN is decomposed with event areas. Each event area is composed of sensors, actuators, and actuation device objects. Each sensor communicates with multiple actuators and each actuator receives sensed values from multiple sensors by using one wireless communication channel. This is the multi-actuator/multi-sensor (MAMS) model which we proposed to make WSAN fault-tolerant in this paper. An actuator makes a decision on what method, i.e. action to be performed on which actuation device object in an event area and then issues the method to the object. Sensors are less reliable and may be arbitrarily faulty, due to low-energy, low cost devices. In addition, messages are lost due to collision in a wireless channel. Actuators are assumed to only stop by fault. We discuss a semi-passive way on how multiple actuators communicate with each other to collect sensed values and make a decision on a value and occurrence order of events.

Acknowledgment

This research is partially supported by Research Institute for Science and Technology [Q05J-04] and Academic Frontier Research and Department Center [16-J-6], Tokyo Denki University.

References

[1] I. F. Akyildiz and I. H. Kasimoglu. Wireless sensor and actor networks: research challenges. *Ad Hoc Networks journal (Elsevier)*, 2:351–367, 2004.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks journal (Elsevier)*, 38:393–422, 2002.

[3] X. Defago, A. Schiper, and N. Sergent. Semi-passive replication. In *Proceedings of the 17th IEEE Symposium on Reliable Distributed Systems (SRDS-17)*, pages 43–50, 1998.

[4] A. Durresi and V. Paruchuri. Geometric broadcast protocol for heterogeneous sensor networks. *Proc. of 19th IEEE International Conf. on Advanced Information Networking and Applications (AINA2005)*, 1:343–348, 2005.

[5] A. Goldberg and D. Robson. *Smalltalk-80: The interactive programming environment.* Addison-Wesley, 1983.

[6] N. Hayashibara, X. Défago, R. Yared, and T. Katayama. The φ accural failure detector. *Proc. of the 23rd IEEE International Symposium on Reliable Distributed Systems (SRDS-23)*, 1:68–78, 2004.

[7] IEEE. *IEEE Std 802.11b – Wireless LAN Medium Access Control (MAC) and Physical layer (PHY) Specifications: High Speed Physical Layer (PHY) in the 2.4 GHz Band.* IEEE, 1999.

[8] IEEE. *Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications.* IEEE, 2000.

[9] S. S. Lam. A carrier sense multiple access protocol for local networks. Technical report, Austin, TX, USA, 1979.

[10] L. Lamport. Time, clocks, and the ordering of events in a distributed system. *Comm. ACM*, 21(7):558–565, 1978.

[11] L. Lamport, R. Shostak, and M. Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, 1982.

[12] T. Melodia, D. Pompili, V. C. Gungor, and I. F. Akyildiz. A distributed coordination framework for wireless sensor and actor networks. *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, 1:99–110, 2005.

[13] K. Ozaki, K. Watanabe, S. Itaya, N. Hayashibara, T. Enokido, and M. Takizawa. A fault-tolerant model of wireless sensor-actor network. In *Proceedings of the 9th International Symposium on Object and Component-Oriented Read-Time Distributed Computing (ISORC 2006)*, pages 186–193. IEEE Computer Society, 2006.

[14] TinyOS. http://www.tinyos.net.

[15] M. Weiser. Hot topics: Ubiquitous computing. *IEEE Computer*, pages 71–72, 1993.

[16] M. Wiesmann, F. Pedone, A. Schiper, B. Kemme, and G. Alonso. Understanding replication in databases and distributed systems. In *Proceedings of 20th International Conference on Distributed Computing Systems (ICDCS'2000)*, pages 264–274, Taipei, Taiwan, R.O.C., Apr. 2000. IEEE Computer Society Technical Commitee on Distributed Processing.