

Design and Implementation of a Policy-based Management System for Data Reliability in Wireless Sensor Networks

Tuan D Le^{*†} Wen Hu[†] Sanjay Jha^{*} and Peter Corke[†]

^{*}School of Computer Science and Engineering, UNSW, Sydney, Australia
{dtle, sanjay}@cse.unsw.edu.au

[†]Autonomous Systems Laboratory, CSIRO ICT Centre, Brisbane, Australia
{tuan.ledinh, wen.hu, peter.corke}@csiro.au

Abstract—In this paper, we describe the design and the implementation of a management system called SRM for controlling data reliability in Wireless Sensor Networks. SRM is based on a hierarchical management architecture and policy-based network management paradigm. SRM consists of four modules: a user policy specification module, an evaluation module, a decision making module and an action module. The interaction among these modules ensures that the network provides adequate information to the users while reducing energy consumption. To demonstrate the effectiveness of the management framework, we design a policy for balancing energy consumption and data reliability. Our experimental results show that SRM can meet the reliability requirements, and reduces energy consumption by up to 50% compared to the case of no management.

I. INTRODUCTION

Data reliability of a sensor node is described by the probability of a data packet being delivered from the sensor node to the sink. Data reliability management is the task that ensures the delivery of data from a sensor node to the base station. Ensuring data reliability across many sensor nodes in a network is a challenging task because data transfer in Wireless Sensor Networks (WSN) is susceptible to loss when there are node failures, environmental interferences, nodes joining or leaving the network, power depletion, etc. For example, in the water quality monitoring application at Burdekin, Queensland, Australia [12], it is observed that the link qualities are considerably lower at night time than in the morning time. In this case, it is desirable to have multiple protocols which can run in different network conditions. Second, because of the scale of sensor networks, typically with tens, or even hundreds of nodes, coordinating the communication across these many nodes is complex.

On the other hand, in most applications, sensor nodes are battery-powered. The unattended nature of sensor nodes and hazardous sensing environments preclude battery replacement as a feasible solution while many sensor network applications demand that the network must operate for a long period of time. Minimizing energy consumption while ensuring the data reliability in dynamic conditions is a complicated task. Thus, it is nearly impossible for a single protocol to be appropriate all the time, even within a single sensor network application [14]. When the reliability of the network degrades, the network is no longer capable of delivering useful information to the users. Therefore, data reliability management, which is capable of

handling a range of possible parameter values or even handling a set of reliable communication protocols depending on node topology, network connectivity, and node status, to control the data reliability for each sensor node, is crucial.

In this paper, we propose, and implement a Sensor Reliability Management framework called **SRM** for WSNs. SRM is based on a hierarchical management architecture and policy-based network management paradigm formulated by IETF [3]. The primary contributions of paper are: 1) We design a management system called SRM specifically for data reliability management. SRM allows the network administrators to interact with the network by defining the management policies. SRM also provides a self-control capability to the network. Although SRM is designed for reliability management, it can easily integrate with other management services by providing the management policies. 2) We implement and evaluate SRM in a real-testbed. Our experimental results show that SRM can meet the reliability requirements and reduces significantly energy consumption.

The remainder of the paper is organized as follows. In Section II, we describe the SRM in detail. Section III describes the implementation details of SRM. In Section IV, we provide an example to demonstrate the benefits of using SRM for data reliability management. Section V describes the related work. Finally, Section VI summarizes the work presented in this paper.

II. SRM ARCHITECTURE

In this section, we present the SRM framework and its major modules in detail. Section II-A discusses the management architecture used in SRM. The following subsections discuss each module of SRM. Finally, we present an algorithm for controlling data reliability.

A. Management Architecture

To address the scalability issue, SRM uses a hierarchical management architecture. The management tasks are distributed to all sensor nodes including the cluster heads and the base station. Management policies are hierarchically distributed over the network. Management policies are divided into three levels: **node level**, which consists of a set of light-weight management rules that require less resources for estimation and can be performed locally; **cluster level**, which

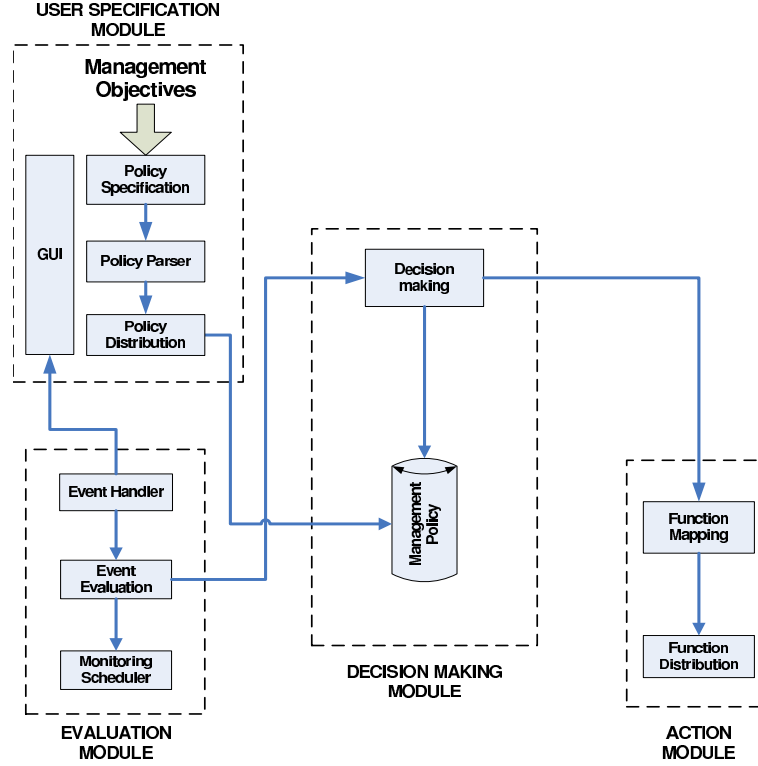


Fig. 1. The Sensor Reliability Management Framework, consisting of four modules: an evaluation module, a user policy specification module, a decision making module, and an action module.

consists of a set of medium-weight management policies that control the reliability of the cluster; and **base station level**, which consists of a set of heavy-weight management policies that control the reliability of the entire network. Next, we discuss the modules which constitute the SRM framework and their interdependencies. As shown in the Fig. 1, a management architecture is adopted for the reliability management framework, consisting of four modules: an evaluation module, a user policy specification module, a decision making module, and an action module. The interaction among these modules enables the management framework to efficiently adapt to the network dynamics. The following subsections discuss the individual role played by these modules in the SRM framework.

B. Evaluation Module

The evaluation module is responsible for collecting the management information required to estimate the reliability of the network. The evaluation module consists of three components: an event handler, an event evaluation, and a monitoring scheduler. The event handler component is a packetizer, which captures the management packets, translates them to an understandable data structure, and parses them to the event evaluation component. The event evaluation component evaluates the reliability of the network by comparing the actual reliability to the required reliability and make a conclusion on the health of the network. To minimize energy consumption, the monitoring scheduler component is responsible for controlling how frequently the management information is collected. Typically, management information is exchanged periodically. However, if there are not many interesting events occurring, the monitoring scheduler may

request to reduce the collection frequency for energy-saving. The monitoring scheduler component has two modes: a passive mode, where the system collects information every pre-defined period of time; and a reactive mode, where the system collects information if an interesting event occurs.

C. User Policy Specification

A policy is a set of rules governing decisions that will be implemented to achieve the objectives [3]. In SRM, we use the Policy Framework Definition Language (PFDL) [5] to express management rules. The PFDL simply expresses lists of IF<condition> THEN <action> SCOPE <scope> type of rules, where <condition> is a disjunctive normal form of condition expressions, <action> is a list of single action statements, and <scope> is the sensor node where the policy is enforced. Each rule comprises one or more terms joined by logical operators (e.g. AND, OR). A term comprises one management variable, a binary operator from the set {<, <=, =, >=, >}, and a reference value, which must be a real number or from a list of pre-defined constants. There are three scopes defined in SRM: LOCAL, for a single sensor node; CLUSTER, for a cluster; and NETWORK, for the entire network. A list of the rules will form a policy. If the evaluation of the condition expression request succeeds, the action list will be performed. In the SRM framework, a management policy is described by an XML schema document [8].

Table I shows an example of a rule in the reliability policy. In this example, E_{TH} , DR_{TH} denote the pre-defined thresholds of the remaining energy level of a sensor node and the desired reliability. The rule states that when the remaining

energy of a sensor node is lower than E_{TH} it should reduce its transmission rate if the delivery ratio is satisfactory, i.e., greater than the threshold DR_{TH} .

```
<?xml version='1.0' encoding='UTF-8'?>
<Rule>
  <CONDITION>
    <VARIABLE> Energy </VARIABLE>
    <OPERATOR> <= </OPERATOR>
    <REFERENCE> E_TH </REFERENCE>
    <VARIABLE> DELIVERY_RATIO </VARIABLE>
    <OPERATOR> > </OPERATOR>
    <REFERENCE> DR_TH </REFERENCE>
  </CONDITION>
  <ACTION>
    <FUNCTION> REDUCE_TRANSMISSION_RATE </FUNCTION>
  </ACTION>
  <SCOPE SCOPE='LOCAL' />
</Rule>
```

TABLE II
THE XML SCHEMA FOR THE DEFINED RELIABILITY RULE

Fig. II shows the corresponding XML schema. The condition is specified by `<CONDITION>` `</CONDITION>` tag. It consists of management variable, defined by `<VARIABLE>` `</VARIABLE>`, i.e., Energy; the logical operators `<OPERATOR>` `</OPERATOR>`, e.g., `<=`; and the reference value, defined by `<REFERENCE>` `</REFERENCE>`, e.g., `E_TH`. The action is specified by `<ACTION>` `</ACTION>` and is followed by a list of functions which need to execute. The scope is specified by `<SCOPE>` and is followed by its value.

The user policy specification module has four components: a policy specification, a policy parser, a policy distribution, and Graphical User Interfaces (GUIs). The policy specification component allows the users to define the management policies via a GUI or a web-page. The policy parser component is responsible to validate and translate the management policies into a data structure, which is understandable by the computer operating system. The policy distribution component distributes the management policies over the network. In SRM, we define GUIs as one of the components in the user policy specification module. By capturing and analyzing the packets received in the event handler component, the GUIs provide the visualization of the current state of the network for the users displaying coverage reliability, routing topology, link quality, etc.

D. Decision Making Module

The decision making module is the central layer of the framework. It has two components: a decision making component and a management policy component.

- Decision making component: Based on the estimated reliability from the evaluation module and the policies defined in management policy component, the decision making component determines appropriate actions to be executed and passes the request to the action module.
- Management policy component: The management policy component is similar to the Management Information Base (MIB) in traditional network management [6], which stores the management policies defined by the

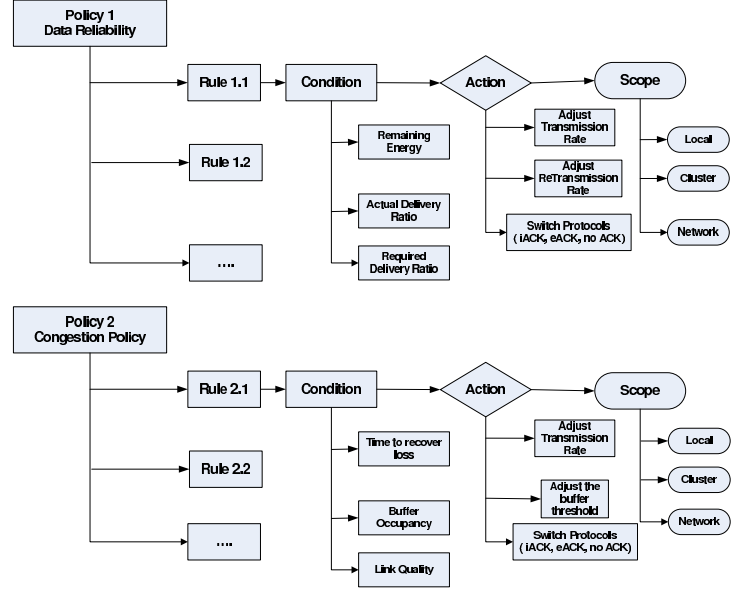


Fig. 2. The Management Policies in the SRM Framework

network administrator. In SRM, two policies are defined: a data reliability policy and a congestion policy. Other policies can be easily integrated into the framework. The data reliability policy contains a set of rules which ensures the end-to-end delivery ratios. The congestion policy contains a set of rules which detect and control the network congestion. As the energy is a critical resource in WSNs, both policies take energy-efficiency into account.

E. Action Module

The action module is responsible for performing the action assigned by the decision making module. The core of the action module is a management function, which can be a single command such as an alarm operation; an algorithm such as adjusting retransmission algorithm; or a protocol such as a transport protocol. An action may be executed by a set of functions on the sensor node in response to a call from the decision making module.

The action module consists of two components: a function mapping, and a function scheduling. The function mapping component maintains a function list for each action, which maps the action to a set of functions to be executed. The function scheduling component is responsible to distribute the assigned functions into the sensor nodes which run the functions.

After executing the action, the action module returns the state of execution to the decision making module. SRM defines three states: success, when the action is performed successfully; fail, when a failure occurs during the execution of the function; and executing, when the action is currently being executed.

Fig. 2 describes management policies and actions used in SRM. There are three actions defined in the data reliability policy called: `ADJUST_TRANSMISSION_RATE`, `ADJUST_RETRANSMISSIONS_RATE`, and `SWITCH_PROTOCOL`. `ADJUST_TRANSMISSION_RATE` is responsible to increase/decrease the transmission rate.

Rule ID	Condition	Action	Scope
1	IF ENERGY \leq E_TH AND DELIVERY_RATIO $>$ DR_TH THEN	REDUCE_TRANSMISSION_RATE	LOCAL

TABLE I
AN EXAMPLE OF A RELIABILITY RULE

Without introducing any extra protocol overhead, a sufficient quantity of data packets transmitted to the base station can ensure the end-to-end delivery ratio. For example, ESRT [7] can be used to implement this action. ESRT dynamically chooses a transmission rate that ensures event reliability for the application. Although ADJUST_TRANSMISSION_RATE action is simple and easy to implement, it is not energy-efficient when the link loss rates are high [18]. To reduce energy consumption, SRM also uses retransmission and acknowledgment schemes. ADJUST_RETRANSMISSIONS_RATE action is responsible to adjust the number of retransmissions to balance the data reliability and energy consumption. The last action is SWITCH_PROTOCOL, which allows the network to change the transport protocol on the fly. There are three protocols defined in SRM: noACK, eACK, and iACK. The choice of these transport protocols comes from the empirical studies in [14], which provides a baseline to compare energy consumption and data reliability between these three protocols. There is a trade-off between energy consumption and data reliability in these three protocols. An insufficient maximum number of retransmissions may cause a packet to be lost as it travels to the base station, wasting energy and network resources, as well as degrading end-to-end reliability. Conversely, there will be energy-inefficiency when the maximum number of retransmissions is too high.

There are three actions defined in congestion management policy called: ADJUST_TRANSMISSION_RATE, ADJUST_BUFFER_THRESHOLD, and SWITCH_PROTOCOL for congestion control. In addition, other actions are defined in SRM such as: ALERT, send the alarm message to the base station, RETURN, return the value after executing the action, etc.

F. An algorithm for the ADJUST_RETRANSMISSION_RATE action

We propose an algorithm for the ADJUST_RETRANSMISSIONS_RATE action for controlling the end-to-end delivery ratios. Let us denote R^* and Δ as the desired average reliability requirement and the deviation, respectively, where $0 < \Delta$ and $R^* < 1$. Let us also denote $NR(i)$ as the current number of retransmissions allowed at node i and the MAX_NR as the maximum number of retransmissions allowed at each node. We define \bar{R} as:

$$\bar{R} = \frac{\sum_{i=1}^N R(i)}{N} \quad (1)$$

where $R(i)$ is the actual delivery ratio obtained at node i and N is the number of sensor nodes in the network. The objective of the algorithm is to maintain the delivery ratio \bar{R} such that:

$$R^* - \Delta < \bar{R} < R^* + \Delta \quad (2)$$

The algorithm is described as follows. After a predetermined period of time, the evaluation module activates and estimates the delivery ratio $R(i)$ for every sensor node i . If the condition of the Equation (2) is held, SRM will do nothing. Otherwise, the decision making module needs to perform a management action to adjust the reliability. There are two cases:

- If $R(i) \leq R^* - \Delta$, the decision making module will increase the number of retransmissions $NR(i)$ at node i . However, if $NR(i) == MAX_NR$, node i is no longer capable of improving the reliability. In this case, the decision making module will find a node on the routing path of node i toward the sink, say node j , in which $NR(j) \leq MAX_NR$ is held and ask the node j to increase $NR(j)$. If there is no node j available, it will inform the base station to prevent further requests.
- If $R(i) \geq R^* + \Delta$, the decision making module will check if the node i is required to increase the $NR(i)$ by its children. If it is not, then it will decrease $NR(i)$.

III. IMPLEMENTATION

A. Overview

The implementation has two parts: the base station side and the sensor node side. The base station side was implemented in a high-level programming language, Java, and was run on a PC. On the sensor node side, the SRM was implemented in C programming language under the Fleck Operating System (FOS) in a 30-node Fleck-3 [2] testbed. FOS is an operating system for WSN nodes developed by CSIRO [10]. It provides a priority-based, non-preemptive (cooperative) threading environment with separate stacks for each thread, which has the advantage of providing a simple concurrent programming model which does not require semaphores. The implementation was run on a Fleck-3 platform, but it is straightforward to port it to other platforms such as Mica2 [1] or Telos [4]. In this work, we only evaluate the network which comprises a base station and sensor nodes. Thus, the base station maintains the management policies at both the base station and the cluster head levels. The performance under cluster topology will be investigated in our future work.

B. Protocol Engine

Surge Reliable was used (for brevity we called Surge) [25] as the routing protocol. Surge is a reliable multi-hop routing protocol that uses link quality as its routing metric. In Surge, each node dynamically selects its parent based on the link quality to construct a stable routing tree to the sink. In this work, the focus is not on optimizing the data transmissions for the downstream direction (from the sink to the sensor nodes). Thus, a flooding protocol is used for the downstream communication. Flooding implements the best-effort flood: each node rebroadcasts packet exactly once and prevents retransmissions by maintaining a duplicate packet list.

Packet Field	Size	Description
address	uint16_t	The local address
group	uint8_t	The group ID
type	uint8_t	Opcode: 0xfa = data packet, 0xfe= management packet
length	uint8_t	The data length
crc	uint16_t	The cycle redundancy code
ack	uint16_t	The acknowledgment
rsi	uint16_t	The received signal strength indication
time	uint16_t	The time
originaddr	uint16_t	The original packet address
seqNo	uint8_t	The sequence number
nodeID	uint16_t	The destination address
function ID	uint8_t	The function ID

TABLE III
THE PACKET HEADER

To avoid packet collision, a packet is rebroadcast with a small random delay. Although we use flooding, any communication protocols from the sink to the sensor nodes can be used. To minimize energy consumption, multiple commands are bundled into a single message. Each command consists of nodeID (2 bytes) and commandID (1 byte). Table III shows the packet header used in FOS. As the packet length in FOS is 32-bytes, each management packet can hold up to 5 commands.

C. Sensor Node Software Component

In the SRM framework, each sensor node has the responsibility of providing the information requested by the base station, and performing the functions as specified by the command requested from the base station. The sensor node also maintains its own management policies which will be executed when the conditions are fired. The example of an event handler component and a monitoring scheduling component in the evaluation module are:

```

1 void * fos_network_rx_thread (void * arg)
2 { fos_message_t rxm;
3   ....
4   while (1) {
5     if (fos_macfilter_read (0, f, &rxm) == 1) {
6       if (rxm.type == FOS_MAC_TYPE_MANAGEMENT) {
7         fos_management_handler (&rxm);
8       }
9     }
10    ....
11  }
12 }

1 void * fos_network_management_send(
2 uint16_t addr, uint8_t * content, int8_t n, uint8_t flags)
3 {
4   ....
5 }

```

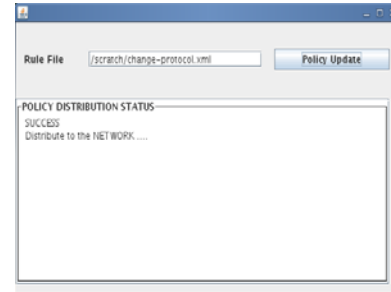
The sensor nodes also have the implementation of other modules: the decision making modules, and the action modules as described in Section II.

D. Base Station Software Component

In the SRM framework, the base station has the responsibility of managing the sensor nodes and providing management information of the sensor network for its users. The base station sensor node is required to provide a bridge between the PC and other sensor nodes. The base station receives the packets from sensor nodes, either management packets or data packets, and forwards them to the PC via a serial port. The network administrator forwards management commands to PC via its TCP port and they eventually reach the sensor nodes via the base station.

ID	NodeID	HopCo...	NextHop	SeqNo	TxCount	AckCount	TotTxCo...	sendEst	receive...	Deliver...
4	3	1	100	144	258	0	593	64	98	0.851...
5	30	1	100	144	1749	0	2849	84	97	0.946...
6	5	1	100	144	161	0	248	72	99	1.0
7	25	1	100	144	1577	0	2769	97	98	1.0
8	22	2	20	144	176	0	264	81	98	0.970...
9	8	2	20	144	176	0	282	92	95	1.0
10	16	2	20	144	173	0	247	99	95	0.970...
11	23	2	25	144	360	0	980	81	98	0.860...
12	18	2	25	144	173	0	421	78	98	0.656...
13	29	2	30	144	151	0	272	99	98	0.992...
14	19	2	25	144	180	0	448	88	99	0.831...
15	26	2	30	144	176	0	275	79	98	0.992...
16	15	2	30	144	195	0	366	94	97	1.0
17	28	2	30	144	182	0	299	79	99	0.731...
18	17	2	30	144	175	0	447	81	96	0.8
19	14	2	25	144	557	0	1242	93	99	0.930...
20	9	2	30	144	445	0	931	94	99	1.0
21	10	2	30	144	186	0	301	80	97	0.946...
22	24	2	25	144	172	0	173	83	98	0.758...
23	27	2	30	144	181	0	360	96	97	0.879...
24	11	3	14	144	176	0	269	98	99	0.489...
25	13	3	14	144	183	0	375	68	98	0.481...
26	7	3	9	144	174	0	298	94	98	0.949...
27	6	3	9	144	176	0	254	94	98	0.949...
28	12	3	14	144	176	0	288	98	99	0.909...

(a) Network Statistic GUI



(b) Management Policy GUI

Fig. 3. Network Statistic and Management Policy GUIs

E. Graphical User Interfaces (GUIs)

A set of GUIs have been developed at the base station side for the network administrators, as shown in Fig. 3. Each tool is triggered by a received packet from a sensor node via the TCP port or the requests from the network administrator. The GUI in Fig. 3(a) provides statistical information about the network such as the next-hop ID, the sequence number of a received packet, the link qualities, etc. The GUI in Fig. 3(b) allows the administrator to load pre-defined management policies in XML format and distribute them to the network.

IV. EXAMPLES

A. Summary

In this section, we provide an example to demonstrate the data reliability policy in SRM with the action ADJUST_RETRANSMISSIONS_RATE. A data streaming application, where the sensor nodes report data to the sink periodically, is considered. The following metrics are used:

- Number of data transmissions: is the total number of data packets exchanged in the network and the entire experiment. It includes the retransmitted data packets and the acknowledgment packets.
- Number of management transmissions: is the total number of packets used for the management activities over the entire experiment.
- Delivery Ratio: is the ratio between the number of packets received at the base station and the number of packets originally sent from the sensor node.

```

<?xml version='1.0' encoding='UTF-8'?>
<Rule>
  <CONDITION>
    <VARIABLE> DELIVERY_RATIO </VARIABLE>
    <OPERATOR> < </OPERATOR>
    <REFERENCE> DR_TH1 </REFERENCE>
    <VARIABLE> RETRANSMISSION </VARIABLE>
    <OPERATOR> < </OPERATOR>
    <REFERENCE> MAX_RETRANSMISSION </REFERENCE>
  </CONDITION>
  <ACTION NAME='INCREASE_RETRANSMISSION' />
  <SCOPE SCOPE='LOCAL' />
</Rule>
<Rule>
  <CONDITION>
    <VARIABLE> DELIVERY_RATIO </VARIABLE>
    <OPERATOR> > </OPERATOR>
    <REFERENCE> DR_TH2 </REFERENCE>
    <VARIABLE> RETRANSMISSION </VARIABLE>
    <OPERATOR> > </OPERATOR>
    <REFERENCE> 1 </REFERENCE>
  </CONDITION>
  <ACTION NAME='DECREASE_RETRANSMISSION' />
  <SCOPE SCOPE='LOCAL' />
</Rule>

```

TABLE V
THE XML SCHEMA FOR THE DEFINED RELIABILITY RULES

In data streaming applications, there are usually different interests at each period of time. For example, in the Burdekin water quality monitoring application [12], there is more interest during the period that the farmers pump the water to the field. Thus, it is desirable to have a better delivery ratio during this period than at other periods. When the farmers pump the water, the SRM decides to use a transport protocol which provides good delivery ratio, e.g. explicit acknowledgment, but possibly with high energy consumption, and uses an energy-efficient transport protocol, e.g. no acknowledgment, at the other period of time to reduce energy consumption, but still provide a certain level of reliability.

The experiment is described as follows. Each sensor node periodically sends a data packet to the base station at the rate of one packet every 10 seconds. The ADJUST_RETRANSMISSIONS_RATE action proposed in Section II-F is used to balance the energy consumption and the delivery ratios. The desired data delivery requirement R^* is 60%, 70%, and 90%, respectively, with $\Delta = 0.02$. The algorithm maintains the reliability ratio between $[R^* - \Delta, R^* + \Delta]$. The experiment was run for 35 minutes (2200 seconds). In order to evaluate the performance, artificial losses are introduced for all the links in the network for a period of time. Specifically, the link layer randomly dropped packets with 35% probability during the period between 15 minutes (1000 seconds) and 25 minutes (1600 seconds), as shown in Fig. 5(a). This effectively increased the expected energy consumption of the sensor nodes.

B. Results

Fig. 4 is a snapshot of one of routing trees constructed during the experiment. Due to the changes in wireless link quality, the routing tree was dynamic with significant routing variability.

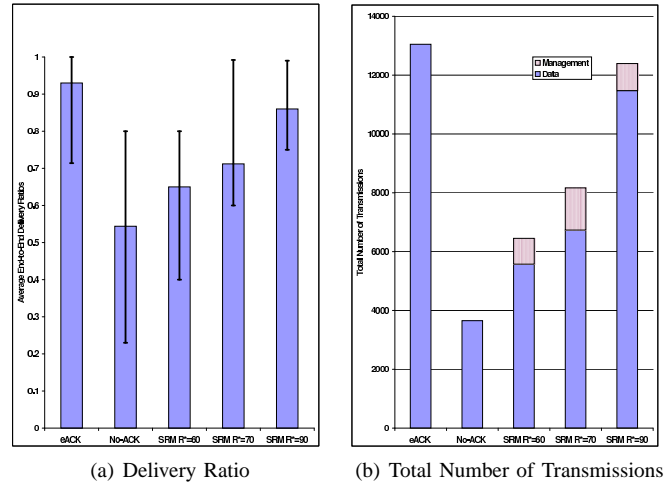


Fig. 6. The End-to-End Delivery Ratios and Total Number of Transmissions

Fig. 5(b) shows the delivery ratio of 30-node versus time when the required delivery ratio is $R = 70\%$. The error bars in the figure show the minimum, the average, and the maximum of delivery ratio for all 30 nodes over a period of 100 seconds. With no-ACK, the average delivery ratios dramatically reduce when the link loss rates are high whereas with eACK (with 7 maximum number of retransmissions), the average delivery ratios are reasonably high, even under high link loss. The reason is that the high number of retransmissions can compensate the packet loss, thus increases the delivery ratio. It is observed that on average SRM achieves delivery ratios at about 71.2%. When the link loss increases, SRM decides to increase the retransmissions, and is constantly adjusting to maintain the delivery ratio at the rate of 70%.

We also ran the SRM with different delivery ratio requirements of $R^* = 60\%$ and $R^* = 90\%$. Fig. 6(a) shows the minimum, average, and the maximum delivery ratios achieved for the entire period. It is observed that on average SRM meets the reliability requirement with the small error of $\pm 5\%$. Fig. 6(b) presents the quantity of data and management traffic transmitted during the experiment with and without management. When $R = 60\%$, SRM reduces the number of transmissions by more than 50%. This result suggests that SRM can reduce energy consumption significantly if the delivery ratio of 60% is satisfactory. Although the number of transmissions using no-ACK is small, the delivery ratio of no-ACK is only 54%, which is lower than the requirement of 60%. When $R = 90\%$, the number of transmissions in SRM is 13.4% lower than eACK. The total number of transmissions of SRM in these three cases, including both the data and management packets, is still less than for eACK. Thus, SRM can balance the energy consumption and data delivery and adapt to the network dynamics.

Although the SRM can provide adaptivity, it comes at a cost. As shown in Fig. 6(b), the management traffic is from 7% to 18% of the total messages transmitted. The reason is that in this example, the monitoring scheduler component

Rule ID	Condition	Action	Scope
1	IF DELIVERY_RATIO < DR_TH1	INCREASE_RETRANSMISSION	LOCAL
2	AND RETRANSMISSION < $MAX_RETRANSMISSION$ THEN IF DELIVERY_RATIO > DR_TH2 AND RETRANSMISSION > 1 THEN	DECREASE_RETRANSMISSION	LOCAL

TABLE IV
RELIABILITY RULES FOR BALANCING THE DELIVERY RATIO AND ENERGY CONSUMPTION

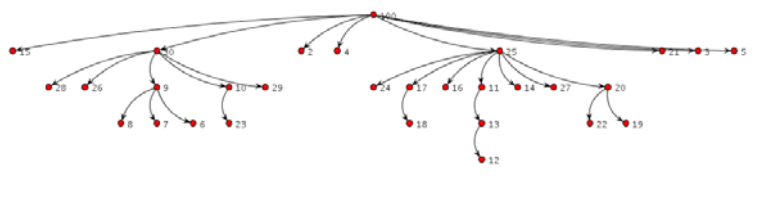


Fig. 4. Network Topology

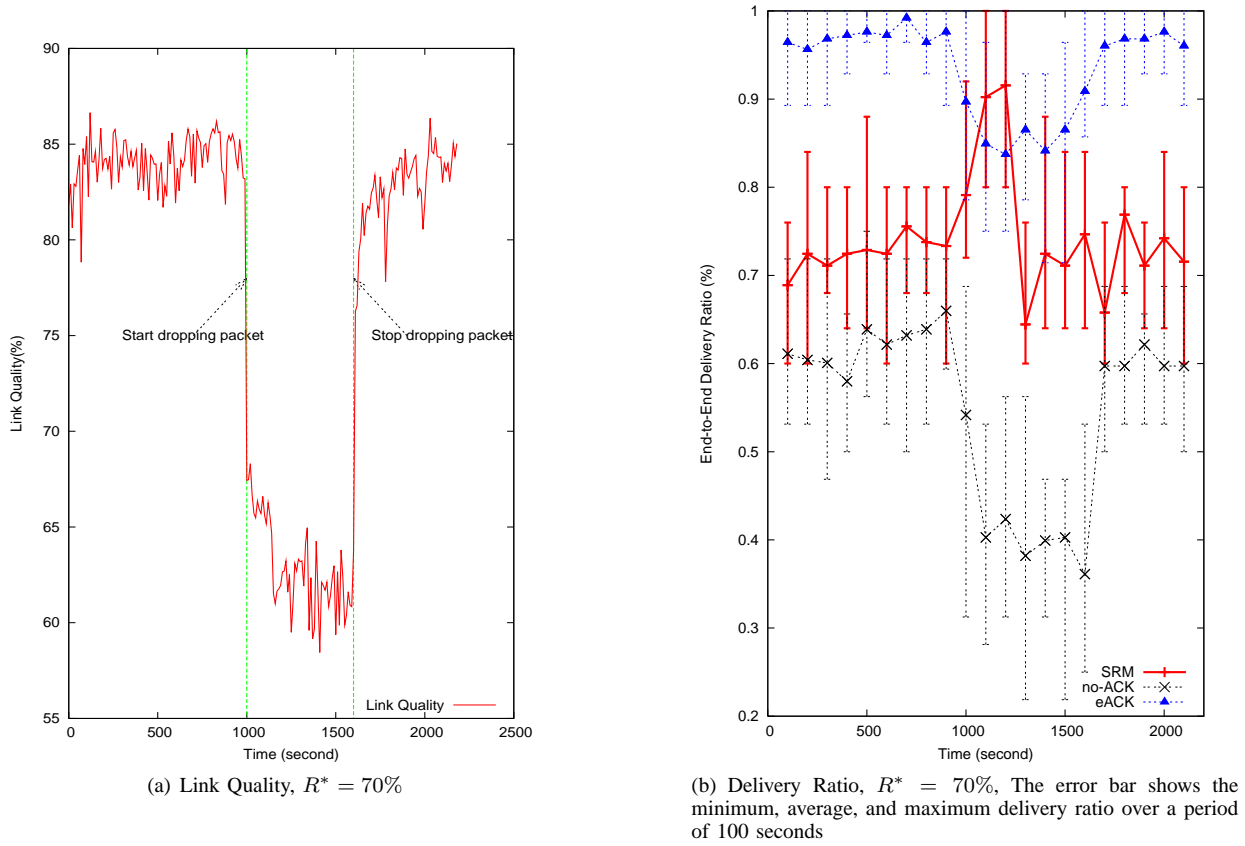


Fig. 5. The Link Quality and End-to-End Delivery Ratios

was in a passive mode, where the management information is periodically collected. It implies that the management solution may be potentially expensive for the management task which requires a lot of management information. Therefore, our future work will investigate the reactive mode of the monitoring scheduler component, which dynamically selects how often the management information is collected based on network conditions for reducing management traffic.

V. RELATED WORK

Cha et al. [9] proposed a policy-based network management approach for sensor nodes to organize and manage themselves. Their work is closest in spirit to our work and both are based on the policy based network management paradigm [3]. However, their work only provides the concepts for designing policy-based sensor network management and primarily focuses on designing an energy-efficient clustering algorithm for policy distribution. In contrast, we provide details of the

management system for data reliability management. Further, we evaluated the system in realistic environment. There are many other management systems have been proposed for WSNs such as BOSS [20], Moteview [22], SNMS [21], etc. However, these systems require network administrators to participate in network management tasks. Thus, they do not provide a capability of self-control network themselves. SRM is complementary to these work which provides both manual and automatic management services. Kim et al. [16] proposed SenOS, a finite state machine based operating system for WSNs. However, the proposed network management protocol is limited to only SenOS operating system. Ruiz et al. [19] proposed MANNA, a management architecture for wireless sensor networks. MANNA provides generic concepts for designing sensor network management. Deb et al. [11] proposed STREAM, a distributed algorithm for sensor topology retrieval at multiple resolutions. The STREAM retrieves network state for multiple resolutions at different communication cost. By selecting a subset of nodes and merging their neighborhood lists, an approximate topology can be constructed. Hsin et al. [13] proposed the two-phase self-monitoring system for WSNs. Two phase self-monitoring system aims to detect malfunctioning nodes in the network. However, none of these work takes data reliability into account.

In the context of data reliability, there are many transport protocols in literature such as CODA [24], ESRT [7], PSFQ [23], RCRT [17], Flush [15], etc. However, none of the existing solutions takes the data reliability problem from the management point of view. Each protocol is developed for a specific application. For example, ESRT [7] is designed for ensuring event reliability; PSFQ [23] is designed for network reprogramming; Flush [15] is designed for delivering a large bulk of data, etc. SRM, on the other hand, provide a framework which allows different reliable transport protocols to run the network.

VI. CONCLUSION

We proposed, implemented, and evaluated SRM: a Reliability Sensor Management framework for WSNs. SRM is based on a hierarchical management architecture and on the policy-based network management paradigm. Although SRM is designed for data reliability management, it can be easily integrated with other management services as a part of a WSN self-management architecture. In addition, SRM also allows the network administrators to interact with the network by providing management policies. The provided examples show that the SRM can provide enough data reliability while minimizing energy consumption.

There are still open issues as far as SRM is concerned. The promising results obtained here motivate a further investigation on other components in SRM. In the future work, we will implement RCRT [17] and CODA [24] for the network congestion policy using the SRM framework. Another interesting question here is how the decision making module handles the conflicts among management rules. A linear programming approach may be useful to identify how the maximization of number of rules could be satisfied which maximizes the objective benefits. Finally, the future work may investigate the

learning of the daily trends in data reliability and come up with an adaptive management rule for reliability management.

REFERENCES

- [1] *Crossbow Technology, Mica2*. <http://www.xbow.com/Products/>.
- [2] *Fleck Platform*. <http://www.sensornets.csiro.au/fleck1.htm>.
- [3] *IETF, Terminology for Policy-Based Management*. <http://www.ietf.org/rfc/rfc3198.txt>.
- [4] *Moteiv, Telos*. <http://www.moteiv.com/>.
- [5] Policy framework definition language, <http://www.ietf.org/proceedings/98dec/i-d/draft-ietf-policy-framework-pfdl-00.txt>.
- [6] Simple network management protocol, <http://www.snmp.com/protocol/>.
- [7] O. Akan and I. Akyildiz. Event-to-sink reliable transport in wireless sensor networks. *ACM Transactions on Networking*, 13:1003–1016, 2005.
- [8] P. Biron, K. Permanente, and A. Malhotra. A xml schema part 2: Datatypes. <http://www.w3.org/tr/xmlschema-2/>, 2001.
- [9] S. H. Cha, J. E. Lee, M. Jo, H. Y. Youn, S. Kang, and K. H. Cho. Policy-based management for self-managing wireless sensor networks. *IEICE Transactions on Communications*, pages 3024–3033, 2007.
- [10] P. Corke. Fos a new operating system for sensor networks. In *EWSEN: The 5th European conference on Wireless Sensor Networks*, 2008.
- [11] B. Deb, S. Bhatnagar, and B. Nath. Stream: Sensor topology retrieval at multiple resolutions, in *journal of telecommunications systems, kluwer publications (special issue on wireless sensor networks)*.
- [12] T. L. Dinh, W. Hu, P. Sikka, P. Corke, L. Overs, and S. Brosnan. Design and deployment of a remote robust sensor network: Experiences from an outdoor water quality monitoring network. In *SenseApp: Second IEEE Workshop on Practical Issues in Building Sensor Network Applications*, 2007.
- [13] C. fan Hsin and M. Liu. A distributed monitoring mechanism for wireless sensor networks. In *Proceedings of the 1st ACM workshop on Wireless security*, 2002.
- [14] W. Hu, S. Rothery, and P. Corke. An empirical study of data collection protocols for wireless sensor networks. In *REALWSN: Workshop on Real-World Wireless Sensor Networks*, 2008.
- [15] S. Kim, R. Fonseca, P. Dutta, A. Tavakoli, D. E. Culler, P. Levis, S. Shenker, and I. Stoica. Flush: A reliable bulk transport protocol for multihop wireless network. In *Proceedings of the ACM Conference on Embedded Networked Sensor Systems (Sensys)*, pages 351–365, 2007.
- [16] T.-H. Kim and S. Hong. Sensor network management protocol for state-driven execution environment. In *Proceedings of the International Conference on Ubiquitous Computing (ICUC)*, 2003.
- [17] J. Paek and R. Govindan. Rcrt: Rate-controlled reliable transport for wireless sensor networks. In *SenSys '07: Proceedings of the 5th international conference on Embedded networked sensor systems*, pages 305–319. ACM, 2007.
- [18] Z. Rosberg, R. Liu, T. Le, S. Jha, A. Y. Dong, and J. Zic. Energy efficient statistically reliable hybrid transport protocol for sensed data streaming. Technical report, CSIRO ICT Centre Pub. no. 07/213, June 2007, 2007.
- [19] L. B. Ruiz. *MANNA: A management architecture for wireless sensor networks*. PhD thesis, Dec, 2003.
- [20] H. Song, D. Kim, K. Lee, and J. Sung. Upnp-based sensor network management architecture. In *International Conference on Mobile Computing and Ubiquitous Networking*, April, 2005.
- [21] G. Tolle and D. Culler. Snms: application-cooperative management for wireless sensor networks. In *Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems*, 2004.
- [22] M. Turon. Mote-view: a sensor network monitoring and management tool. In *Proceedings of the 2nd IEEE workshop on Embedded Networked Sensors*, 2005.
- [23] C.-Y. Wan, A. Campbell, and L. Krishnamerthy. Psfq: A reliable transport protocol for wireless sensor networks. In *Proceedings of the first ACM International Workshop on Wireless Sensor Networks and Applications*, pages 862–872, 2002.
- [24] C.-Y. Wan, S. B. Eisenman, and A. T. Campbell. Codas: congestion detection and avoidance in sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 266–279. ACM, 2003.
- [25] A. Woo, T. Tong, and D. Culler. Taming the underlying challenges of reliable multihop routing in sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 14–27. ACM Press, 2003.