Improving the Reliability of Event Reports in Wireless Sensor Networks

Wei Yuan, Srikanth V. Krishnamurthy, and Satish K. Tripathi
Department of Computer Science and Engineering, University of California, Riverside,
Riverside, CA, 92521
Email: {wyuan,krish,tripathi}@cs.ucr.edu

***Abstract*— In wireless sensor networks, data from sensors has to be transported to a central server or sink. Since the sensors are power-constrained devices, the data is typically fused en route, and an aggregated report of fused information is finally available at the sink. It is important that information from as many sensors as possible be fused in order to increase the credibility of the aggregated report. However, in sensor networks there may be faulty sensors or even malicious intruders that generate and report misleading information. Thus, it is important to collect and fuse enough *correct reports* that agree with each other; this would enable nodes that perform fusion to detect and ignore the effects of the faulty reports. In this work, we propose a protocol called Corroborative Aggregation Protocol (CAP), in which, each sensor that detects a report from its neighbor that contradicts its own findings generates its own report to dispute the faulty report. The idea is to increase the number of correct reports so as to effectively reduce the adverse effects of faulty reports. We show by simulations that CAP is effective in maintaining the credibility of the final fused content even if approximately 30 % of sensors within a detecting zone are wrong about an event.**

I. Introduction

Due to rapid advances in VLSI, RF and embedded processor technologies, the widespread use of wireless sensor networks to obtain physical quantities, such as temperature, pressure, etc., from the environment anywhere and at anytime is becoming the reality. Such a sensor network usually consists of hundreds or thousands of micro sensors with the capability of wireless communications and the ability to perform adequate processing to interpret the sensed data [1] [2] [3]. The information generated by the sensors is to be ultimately delivered to a central server or sink. Since the sensor network is typically energy and bandwidth constrained, it is important to process the data en route, and make local interpretations to reduce the amount of data flowing to the sink. This process is known as data fusion [4][6][7]. Typically, the region in which the sensors are dispersed is large in comparison with the limited transmission range of individual sensors and hence, a given sensor cannot directly communicate with all of the other sensors that might detect a common event. One might envision that at each sensor node at which data is received from multiple other sensors, information is fused to the extent possible. Thus, data will be fused multiple times, in stages. As data is relayed towards the sink, sensor nodes may collect data (probably already fused to a certain extent) from other sensors and perform data fusion. As an example in Fig. 1, node C fuses the data received from nodes E and F, and node B in turn fuses the information received from nodes C and D. The process wherein sensors detect an event and the content related to the event is eventually aggregated at the sink via multiple levels of fusion en route, is called a "round" of aggregation.

This work was partially supported by HRL Laboratories, LLC and the UC Micro.

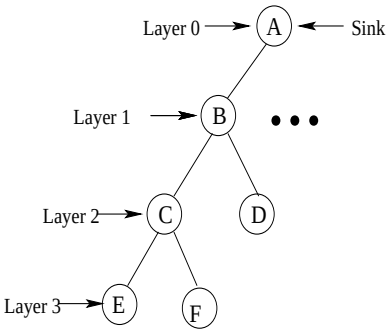

Fig. 1. A Structure to Perform Fusion at Multiple Levels

The credibility of the aggregated report at the sink is a measure of the accuracy of the sensed information. In this paper, we simply quantify the credibility of the aggregated report by the number of corroborative individual sensor reports that are fused in the aggregated report. Thus, the higher the number of the corroborative sensor reports, the higher the associated credibility of the aggregated report.

As mentioned earlier, multiple nodes need to collaborate and agree upon the detection of an event [3]. Typically, a collection of nodes, which are within the geographic vicinity of each other will be likely to detect the same event at approximately the same time. The reports from those nodes should support each other. However, some defective nodes or malicious nodes may generate faulty reports. In this paper, we propose a distributed protocol that facilitates co-ordination among sensor nodes to filter out these wrong reports. We name this protocol *the Corroborative Aggregation Protocol (CAP)*. In short, with CAP, each sensor node that hears a faulty report is stimulated to generate its own report disputing the faulty report. The goal is to increase the number of correct reports in the event of a faulty report to overcome adverse effects that may arise due to the wrong reports.

The rest of this paper is organized as follows. In section II, we describe related work. In section III, we describe our model of the sensor network; this forms the framework for the studies in this paper. In order to fuse data effectively the fusion operations at multiple levels ought to be synchronized. We describe a method (from our earlier work in [11]) for providing this synchronization in section IV. We describe our protocol, CAP in section V. Our simulation results and the evaluation of

our protocol are presented in section VI. Finally, we conclude our work in section VII.

II. Related Work

The focus of our paper is unique and different as compared with previous work on sensor data aggregation. In [9], The authors present *Directed Diffusion*, in which, a sensing task is disseminated from an aggregating sink throughout the network as an *interest*. A node sets up a *gradient* upon the receipt of each *interest* from its neighbor or neighbors. When a sensor node's profile matches the one indicated in the *interest*, it activates its sensor to collect data, and sends the data message to each of its neighbors for which it has *gradients*. In-network aggregation is performed during data diffusion. This work however does not address issues of the associated credibility of the aggregated content, our scheme can be considered as a possible *collaborative information processing extension* for *Directed Diffusion* that helps improve the tolerance to sensor failures.

In [10], the authors propose a collaborative calibration scheme, which exploits the redundancies in sensor measurements under dense deployment scenarios and. Although the focus of [10] is on the calibration of sensors, the overall philosophy of the approach of exploiting the redundancies in sensor reports in order to determine biases in sensor readings is similar to our objective. However, the authors do not investigate the effects of erroneous reports on the credibility of the final fused content.

Other Collaborative Signal and Information Processing (CSIP) approaches in sensor networks have also been studied under various application contexts [5], but these approaches focus more on detection, classification, tracking of targets, distributed compression, and active sensor querying, etc., rather than the reliability of sensors reports in the existence of sensor failures.

III. Sensor Network Model

In this section, we describe the basic framework of the sensor network considered in this paper. A sensor network consists of a large number of wireless micro-sensor nodes that are distributed over a certain area. We limit the sensor network we study to be homogeneous, which means all sensors are the same in terms of various specifications. After sensors have been dispersed, the distance between two neighbors can be measured by using the received signal strength indicator (RSSI), the time-of-arrival(ToA), the time-difference-of-arrival (TDoA), and the angle-of-arrival (AoA) [8]. The area may be divided into a number of regions based on the positioning precision requirements, sensing range of the sensors and other application specific requirements. Each micro-sensor node has at least one sensor, a computation unit and a radio transceiver. There are three circumstances that would cause a sensor node to send a report to the sink. First, sensor nodes periodically send reports to the sink and we call this periodical reporting. Second, the sink queries sensors in specific regions for current sensed information and we refer to reports generated in response as sink inquiry response reports. Third, due to the occurrence of certain events, reports are triggered from sensors in the particular region in which the event occurs; we call these reports event triggered reports. In this paper, we limit ourselves to event-triggered scenarios; however, note that the schemes that we propose can easily apply to the other scenarios as well, with minor modifications. Depending on the target event (application) and the types of sensors deployed (temperature sensors, pressure sensors, motion sensors, etc.), the way in which data is fused may vary. Data fusion models for various types of sensed data may be found in [12]. The protocol we propose in this paper is independent of the fusion model used; however, to simplify our analyses and simulations, we assume that the data that a sensor generates only represents whether or not an event occurs. A fused report would simply contain a count of the number of reports that either confirm or dispute the occurrence of the event[1].

Typically, the sink is distant from the area where the sensor nodes reside. The sensor data has to be ultimately relayed to the sink via multiple sensor node relays. One can then visualize the data being transferred via a structure that facilitates the *many-to-one* data transport. In some sense, building such a structure is akin to building a single source multicast tree; the difference is that instead of data propagating from the single source to the multicast group members, the data flows in the opposite direction, i.e., from the members to the sink. A second difference is that en route, data may be fused at various vantage points on the tree. The topology of this *aggregation tree* determines the efficiency with which data may be fused, to a certain extent. The discussion of the algorithms that generate and maintain this tree are beyond the scope of this paper. Our only requirement is that each sensor node is aware of its immediate neighbors; specifically it should know its *parent* i.e., the sensor node to which it sends data (either fused or raw) and its *children*, sensors from whom it receives data. For the purposes of this paper we simply create this tree using the breadth first search (BFS) technique [17]. Note however that, any other technique for building a tree (with arbitrary characteristics) may be used. Each *non-leaf* node or *internal node* is responsible for relaying (after possibly performing fusion) data received from its children towards the sink node. We assume that the aggregation tree is formed at the network initialization phase, and is dynamically re-organized as sensors sleep, wake up or fail. We note that in the aggregation tree, we refer to nodes as being at particular layers. The sink is at the lowest layer (Layer 0) whereas the leaves are at the highest layer. Furthermore, note that multiple trees may be formed for gathering information from multiple (possibly geographically separate) sensor sets.

We assume a high density of sensor nodes and that multiple sensors detect each event. The *credibility* of the final report at the sink is directly reflected by the total number of reports that are fused and we denote this number by TN. Further, we define an additional metric which we call "Index of Credibility" (IC), as follows:

$$IC = \frac{TNP}{TN},$$

[1] In contrary, another example of data fusion might involve an attempt to estimate the precise location of a target by extrapolating its distance computations from multiple sensors whose co-ordinates are known. Even here, the more the reports the more precise will be the estimated position of the target.

where, TNP is the total number of nodes that indicate the occurrence of an event (defective nodes or malicious intruders may give opposite responses). Thus, the value of IC indicates the accuracy of the reported content at the sink, with regards to the occurrence of the event in terms of concurring reports.

We restrict ourselves to the occurrence of a single event in this work. Multiple events can be treated individually by using the same method. Note that by either including query identifiers or by associating time-stamps and geographical position with events, one could identify a particular event. If a sensor is unable to fuse data (application layer function), it may still be able to simply concatenate reports to the extent possible to save on header overhead.

The credibility of the aggregated data received by the sink is of great importance. A false alarm or an undetected alarm may be of significant consequence leading to either wastage of precious resources or an inappropriate reaction in response to an event. There are two types of effects that cause erroneous data. The first is due to a node's functional defects. As mentioned earlier, a micro-sensor node consists of sensors, a computation unit and an RF transceiver. The computation and transceiver operations are typically more robust than the sensing operations. Bit errors in a packet caused by a harsh noisy channel can be detected and even corrected through CRC, FEC or other error-detecting and correcting codes [14]. We are mainly concerned with sensor defects and erroneous sensor reports. One could also envision the presence of malicious sensor nodes that attempt to transmit misleading reports. Ideally, security protocols such as SPIN [13] are to be implemented in the entire network to ensure data confidentiality and integrity. However, this will introduce extra costs in terms of energy consumption, overhead and computation power. Obtaining multiple reports that corroborate each other can provide a "weak" notion of security against isolated malicious reports. We emphasize here that this is but a "weak" protection and compromise of aggregating nodes close to the sink can prove to be catastrophic in the absence of robust security mechanisms. The purpose of our protocol, CAP (to be described), is to provide some robustness to tolerate faulty sensors as opposed to overcoming security threats.

IV. Synchronization of Fusion Levels

In order to ensure that the sensed data be fused efficiently en route the sink, it is important that the fusion events at the various levels of the aggregation tree be synchronized. We propose a protocol called the Multilevel Fusion Synchronization (MFS) protocol that facilitates this synchronization in [11]. We use this protocol for synchronizing the fusion events in the sensor network in this paper and describe MFS here for completeness.

With MFS, the sink determines system parameters that are represented by MAX and Δ and during the phase when the aggregation tree is set up (or refreshed) indicates the values of these parameters to the relevant sensor nodes. The choice of MAX and Δ depend upon the trade-off that the user (sink) wishes to achieve between the credibility of the final fused report and the latency incurred during a round of aggregation.

Upon sensing an event, a leaf node on the tree immediately transmits a *raw* report to its parent on the tree. The receipt of this report triggers a timer at the parent node. We refer to all non-leaf nodes as *internal* nodes. An internal node, upon setting its timer, stimulates its neighbors by broadcasting a START message. The stimulus, in turn, causes the neighbor nodes to trigger their timers to indicate the beginning of the fusion operation for the associated event. Thus, the timer at an internal node is triggered by any of the three following events, (a) the detection of the event by the internal node, (b) the receipt of the first incoming report from one of its direct children, or (c) the receipt of a START message from a neighbor. The timer, thus triggered, will expire (MAX - K*Δ) seconds later, where K is the distance (in hops) from the internal node to the sink. Upon the expiry of the timer, data received will be aggregated and passed further up in the tree. We assume that late reports with regards to the event are simply discarded. Instead, another policy might be to send these late reports towards the sink without performing any fusion. This may be inefficient and expensive in terms of power consumption.

V. The Collaborative Aggregation Protocol (CAP)

Our objective in CAP, as mentioned earlier, is to provide robustness to faulty sensor reports. We allow sensor nodes to take advantage of the broadcast channel and operate in a *promiscuous* mode when possible, to overhear sensor reports generated by neighboring nodes. When a leaf node generates a report, it transmits the report to its parent immediately by means of a P-pac (Positive-packet). Its neighbors overhear this P-pac. If an overhearing neighbor is a leaf node and it disputes the report, it could *potentially* generate (to be discussed later) an N-pac (Negative-packet) and transmit the N-pac to its parent node. If the overhearing node is a leaf node and it corroborates the report, it simply ignores the P-pac. An internal node, upon the receipt of a P-pac, triggers its timer if it has not done so yet. If the P-pac is destined for this internal node, the internal node includes it for aggregation. An internal node also checks to see (a) if it has also detected the event reported by the P-pac; or (b) if it has received other P-pac messages corroborating the event, at the expiry of its timer. If neither of the above is in the affirmative, the internal node could *potentially* generate an N-pac and include it with the aggregated report. Thus, it would intentionally reduce the credibility of the aggregated report.

It is important to ensure that an N-pac generated by a leaf node does not trigger other nodes to generate new messages disputing the N-pac and that the process of corroboration be restrained to the area in which the event occurs. Thus, if a node wants to dispute a report, it does not generate an N-pac all the time, but only with a probability ρ. In another word, there is only ρ chance that the opinion of the disputing node will be taken into account. The probability ρ depends on the density of the sensor network, the sensor's sensing range R and the placement of sensors. We define a sensor S's sensing range R to be the radius of the circle area A with S at its center. All events that occur inside A can be detected by S while any event that occurs outside A can not be detected by S. We also refer to circle A to be S's *coverage*. As shown in Fig. 2, the probability ρ of a pair of nodes $S1$ and $S2$ detect the same event equal to the ratio of the area of the overlap between the coverages of the two nodes

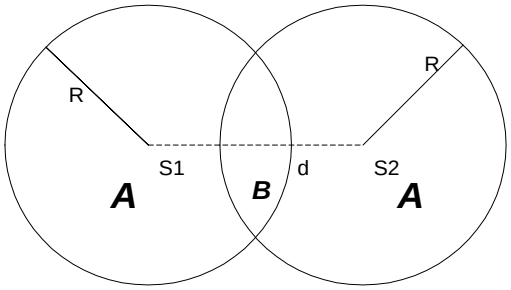

Fig. 2. Calculate ρ for a Pair of Neighbor Nodes

to the coverage of a single sensor node. Thus, we can calculate ρ to be

$$\rho = \frac{B}{A} = 2R^2 \arccos(\frac{d}{2R}) - \frac{d\sqrt{4R^2 - d^2}}{2}.$$

where d is the physical distance between nodes $S1$ and $S2$. As we mentioned in the previous section, it can be established by measuring one or more characteristics of the radio signal. By introducing the probability ρ, we weight the credibility of each *potential* disputing nodes basing on its distance to the sensing nodes. This policy is based on the intuition that nodes which are in close proximity are all likely to detect an event that occurs in the vicinity. Thus, if a faulty sensor transmits a false report, given that other sensor nodes in its proximity do not detect the event, one might expect that a number of disputing (correct) reports will be generated. In the other hand, a real event will also receive certain number of disputing (wrong) reports. However, because those disputing nodes which do not detect the event are not in the close vicinity of the event, thus are likely far away from those sensing nodes also, the values of ρ of the disputing node will be small, which will keep the number of disputing (wrong) reports low. Note that implicit in this discussion is an assumption that typically one might expect a high density of micro-sensors (as assumed in other literature [1]).

To summarize, CAP provides a method by which correct reports are confirmed by reinforcement and the credibility of isolated reports are reduced.

VI. Simulation and Results

We implemented CAP and MFS in ns-2 [15]. For our simulations we make use of the CMU Monarch group's mobility extensions [15]. The existing implementation of the IEEE 802.11 [16] MAC layer protocol is used.

Δ(sec)	0.03	0.08	0.16	0.22	0.28
IC(%)	81.35	95.72	95.02	97.65	89.03

Table.1 Index of Credibility(IC) of Data with CAP

We uniformly randomly distributed one hundred nodes over a 100m*100m square area, The sink resides outside the field and close to one of the corners of the square region. We assume the radio range of sensor node to be 50m and the sensing range to be 25m. An event happens at a randomly chosen point within the field of interest. We include the MFS protocol to synchronize

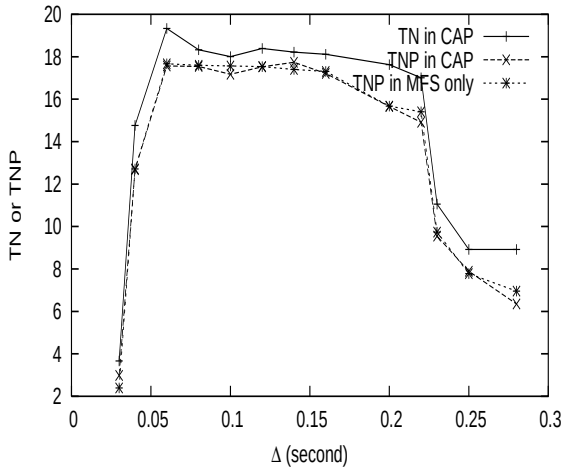

Fig. 3. Credibility of Data with and without CAP

the fusion operations at various levels on the tree as described earlier. We set MAX to be 1.2 seconds (as in [11]), and we run tests over different values of Δ.

We performed the following simulation experiments for each random distribution of the sensors:
(a) We use a scenario with no faulty sensors to ensure that implementing CAP does not degrade the performance in terms of the credibility of the aggregated report under normal conditions;
(b) we introduce faulty sensors that report wrong information about an event that actually occurs;
(c) faulty sensors report the occurrence of a *phantom* event by means of P-pacs.

In Fig. 4, we compare the performance of the MFS protocol with and without being complemented by CAP. Note that the average total number of aggregated individual reports indicating the occurrence of the events (denoted by TNP to denote "Total Number of Positive packets") with CAP is almost identical to the average total number of the individual reports (TN) when CAP is not included. The inclusion of CAP however, produces additional reports from nodes that dispute the occurrence of the events. Since, in this experiment, we had no faulty sensors, these disputing reports are from nodes that are not within the sensing zone and thus, do not sense the events. Since these nodes do not detect the events, they generate N-pacs disputing credibility of the reported information. Let us call such nodes *mislead nodes*. However, we observe that the number of such *mislead nodes* is fairly small, and a high level of Index of Credibility is associated with the fused content ($> 80\%$) in spite of these contradictory reports (Table. 1).

Next, for each of the above events in each of the above setups, we randomly pick certain number of nodes which generate a P-pac in the above simulation, and make them the faulty nodes, i.e. force them to drop the report. We vary the number of the faulty nodes that produce no report. Our metrics of interest are TNP, TN, and the Credibility of the fused content.

Fig. 4 and 5 show that even when the number of faulty sensors increases to 7 (which is more than 30% of the total number of nodes in the sensing zone), the aggregated data still has a IC of above 70%. Thus, the sink will have enough corroborative

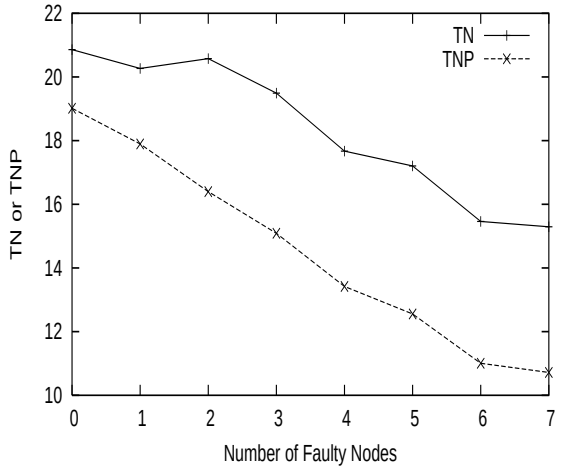

Fig. 4. Credibility of data with CAP when there are faulty Sensors that miss an event.

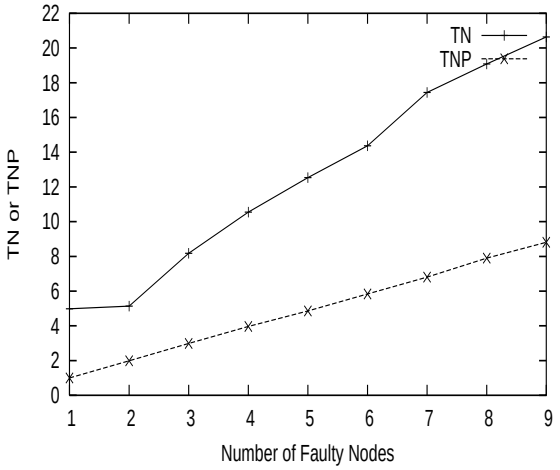

Fig. 6. Credibility of data with CAP when there are faulty sensors that generate false alarms

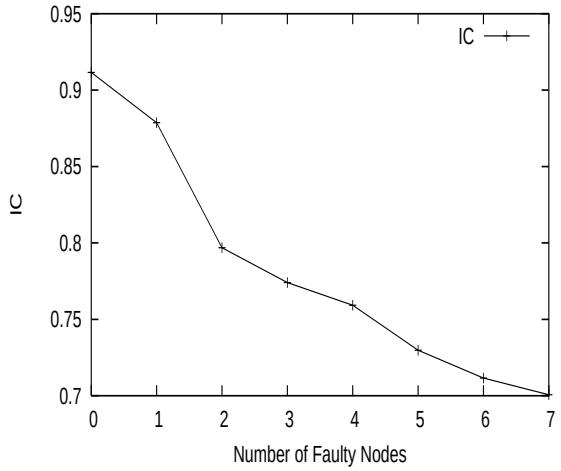

Fig. 5. Index of Credibility of data with CAP with faulty Sensors that miss an event.

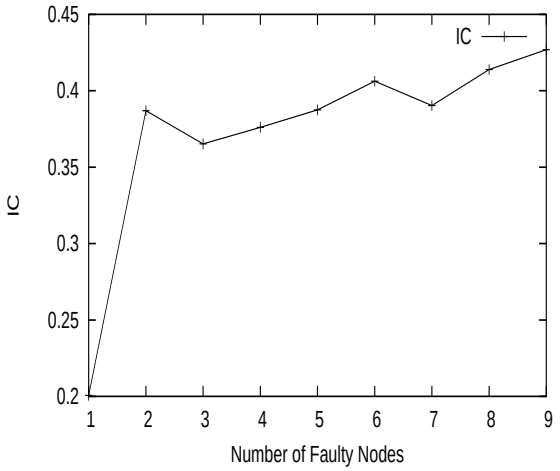

Fig. 7. Index of Credibility of data with CAP when there are faulty sensors that generate false alarms

information to make a good decision.

Note here though, since we assume only the sensing components to be faulty, these faulty nodes can generate disputing wrong results upon overhearing P-pacs from the defect-free nodes, due to CAP. Furthermore, note from Fig. 4, that as the number of faulty sensor nodes increases, the values of TNP and TN actually go down. The decrease in TNP is because of a lower number of defect-free sensors. The decrease in TN is because of the obvious decrease in TNP, and a possible reduction in the number of *mislead nodes*. The second factor is because some *mislead nodes* in the previous test may not hear any P-pac in the current tests, since their sources of P-pacs are faulty nodes in this test and do not generate P-pacs any more. Thus, the total number of reports fused, TN, decreases.

Finally, we randomly choose a certain number of nodes to be faulty nodes, which wrongly report the occurrence of a phantom event. We increased the number of such faulty sensors to up to 9. Fig. 7 shows that the Index of Credibility of such reports stays fairly low (under 45%) due to disputing reports from the *good nodes* that overhear the faulty reports. Also note in Fig. 6, that the value of TN increases much more steeply as compared with the value of TNP (note that P-pacs are now generated by the faulty nodes). This is because of a significantly higher increase in the number of reports from the disputing defect-free sensors. Thus, CAP provides tolerance to faulty sensors.

VII. Conclusions

In wireless sensor networks, raw data from sensors is fused and transported to a central sink. In the presence of faulty sensors the credibility of this final fused content can be seriously affected. In this paper, we propose a protocol (Corroborative Aggregation Protocol) that helps improve the credibility of the fused content in the presence of faulty sensors. CAP works on the principle of having multiple sensors corroborate each other and by having each sensor, upon the receipt of a report, generate a negative re-enforcement report contradicting the original report at a probability ρ which is a measurement of proximity of itself to the original reporting node. Our simulation results quantify the performance of our protocol and show that use of CAP is a viable option for sensor networks. In particular we

show that CAP does not degrade performance of the network during normal operations (no faulty sensors). We also show that in typical scenarios, with CAP, the credibility of the fused content (quantified in terms of the percentage of correct raw reports that are fused) is as high as 80 %.

REFERENCES

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "A Survey on Sensor Networks." *IEEE Communications Magazine*, Vol. 40, No. 8, pp. 102-116, August 2002

[2] D. Estrin, L. Girod, G.Pottie and M.Srivastava, "Instrumenting the World with Wireless Sensor Networks." In *Proceedings of ICASSP 2001*, May, 2001.

[3] G.J. Pottie, and W.J. Kaiser, "Wireless Integrated Netwrok Sensors." *Communications of the ACM*, Vol. 43, No. 5, May 2000.

[4] G.J. Pottie, "Hierarchical Information Processing in Distributed Sensor Networks." In *Proceedings of the IEEE International Symposium on Information Theory, 1998*, pp.163, 1998.

[5] IEEE Signal Processing Magazine special issue on Collaborative Signal and Information Processing for Microsensor Networks, S. Kumar, F. Zhao and D. Sheperd(eds.), Vol. 19, no. 2, March 2002.

[6] W.R. Heinzelman, A.C. Chandrakasan and H. Balakrishnan, "Energy Effi cient Communication Protocol for Wireless Microsensor Network." In *Proceedings of the IEEE Hawaii International Conference on System Sciences*, Jan, 2000.

[7] C. Intanagonwiwat, D. Estrin, R. Govindan and J. Heidemann, "Impact of Network Density on Data Aggregation in Wireless Sensor Networks." In *Proceedings of International Conference on Distributed Computing Systems*, July, 2002.

[8] J. D. Gibson, edittor-in-chief, "The mobile communications handbook." *Boca RAton, CRC Press*. New York, IEEE Press, 1996.

[9] C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks." In *Proceedings of the sixth annual international conference on Mobile computing and networking 2000*, Aug, 2000.

[10] V. Bychkovskiy, S. Megerian, D. Estrin, and M. Potkonjak, "Colibration: A Collaborative Approach to In-Place Sensor Calibration."In *Proceedings of the 2nd International Workshop on Information Processing in Sensor Networks (IPSN'03)*, April, 2003.

[11] W. Yuan, S. V.Krishnamurthy, and S. K. Tripathi, "Synchronization of Multiple Levels of Data Fusion in Wireless Sensor Networks." In *Proceedings of IEEE 2003 Global Communications Conference (GLOBECOM)*, Dec, 2003.

[12] D.L.Hall and J.Llinas, *Handbook of Multisensor Data Fusion*, CRC Press, 2001.

[13] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J.D. Tygar , "SPINS:Security Protocols for Sensor Networks."In *Proceedings of the seventh annual international conference on Mobile computing and networking 2001*, July 2001.

[14] A.S. Tanenbaum, "Computer Networks, 3rd Edition.", *Prentice Hall PTR*. pp.183-190, Upper Saddle River, New Jersey, 1996.

[15] The Network Simulator - ns-2 "http://www.isi.edu/nsnam/ns/".

[16] "IEEE 802.11 Standard for Wireless LAN: Medium Access Control (MAC) and Physical Layer (PHY) Specifi cation.", New York, Approved on 26 June, 1997.

[17] T.H. Cormen, C.E. Leiserson and R.L. Rivest, "Introduction to Algorithms.", *The MIT Press*. pp.469-477, Cambridge, Massachussatts, 1990.