A Fault Tolerant Efficient Scheduling Method for Improved Network Delay in Distributed Sensor Networks

Xiaojing Yuan, Haoying Liu, Jian Shen
Engineering Technology Department, University of Houston
xyuan@uh.edu,liuhaoying@gmail.com, jshen9@uh.edu

Abstract— Wireless sensor networks are now applied in many fields which deal with data-centric information or critical mission that need minimized network delay. At same time, it needs the capability of dealing with faulty situations. In this work, we propose scalable network architecture and an operating mechanism that in nature is tolerant to network structure change caused by failure, and extend our application level scheduling algorithm with the capability of healing from the failure. The algorithm achieves optimized network delay and gets close form solution that is computational simple to implement. We also look into the significance of different nodes' failure in both homogeneous and heterogeneous sensor network. Simulation results imply the effects of sensing and communication speed on significance of failure in heterogeneous sensor networks.

I. Introduction and Literature Review

Recent advances in MEMS (J.W. Gardner et al. 2001) technology make it possible to have compact and low-cost sensor nodes (xbow) with local sensing, processing and short range wireless communication capabilities. A network of such nodes is defined as wireless sensor network (WSN). Because of the flexibility and cost effectiveness of these sensor nodes, networked wireless sensor nodes (i.e., wireless sensor network) have been widely used in various monitoring systems, data collection, and process control applications (I.F.Akyidiz et al. 2002). However, due to the nature of the low-end embedded devices with limited energy budget, radio communication, and primitive user interface, WSNs are highly prone to hardware and software faults, security threats, and intrusion attacks (S.Gupta et al. 2007). Recent researches on fault tolerant sensor network focusing on the networking stack (B.Yu et al. 2006, B.Parnoy, et al. 2005, B.Bhargava et al. 2004, N. Ramanathan et al. 2005).

Various backup mechanisms have been developed to recover from the faulty nodes and communication connections. J. Bredin et al.(2005) developed multi-path routing method to maintain a k-connected network by allocating additional redundant sensor nodes. Others developed cooperative sensor processing to provide redundant copy of data and aim to optimize memory usage and delay overhead. Y. Wang (2005) and H. Wu (2005) developed a schema that make decision on which redundant message needs to be sent or dropped based on fault tolerance requirement and the time and destination of the message based on delivery condition. S. Chessa (2005) proposed a method that distributes the recovered information among the surviving sensors after fault detection. At the decision level, multi-sensor fusion techniques (F. Koushanfar et al., 2002) have also been developed to reduce the uncertainty caused by the failure.

With better and improved processing and storage capability, WSNs have spread to data-centric and mission critical applications which handles multimedia information such as sound, image, and video streams. The larger data packets and the mission critical real-time requirement put the challenge on minimizing the network delay when transmitting of existing wireless sensor network. In our previous work (H.Liu et al., 2007), we developed efficient scheduling method (instead of at networking stack) to study the dynamic changes of the network delay at the application layer. We define the "***network delay***" as the total response time from cluster head (that has better power, computation, and communication resources) assigning sensing tasks to each node in the network, each node accomplish its sensing task and on-board processing, and reporting back to the central node by uploading the raw measurements and/or results from local processing. In this paper, we extend the model to study the network delay when there are failures either at the sensor nodes or networking stack.

The organization of this paper is as follows. In section II, we discuss the system model and notations used in the paper. Section III gives the details of solving the network delay overhead caused by failure in homogeneous and heterogeneous networks. In section IV, we present simulation results of the delay overhead based on the model. The conclusion is reached in section V with future work discussion.

II. System Model and Notations used for Fault Tolerant Ad Hoc Wireless Sensor Network

The network structure is continuously changing for an ad hoc wireless sensor network because of the dynamic wireless connection. It can be linear, star, tree, or mesh at any instance based on the status of communication link and routing strategy. In order to model such dynamic behavior, we define a basic temporal unit "cycle" as the minimum time that the network structure stays fixed. The network structure can stay fixed for several cycles or change in the next cycle. To reduce the complexity of the problem, we assume any sensing task can be accomplished either within one cycle when there is no failure, or complete in two cycles when there is failure in the first cycle. We will show the model can be generalized when more than two cycles are needed to accomplish the sensing tasks.

Within each cycle, the cluster head first discover reachable nodes and set up the network topology. Then it performs the scheduling algorithm to find the optimal task distribution that will result in minimized network delay. The partitioned sensing tasks are then distributed to the smart sensor nodes (SSN) within the cluster. After each node finished their sensing task, they report the results back to the cluster head. The reporting sequence is implied by superscripts. For example, the node that reports last is represented as $SSN^1$and the one that reports second last is denoted as SSN^2. All these operations are completed within one cycle, during which the network structure does not change. Figure 2 shows the time diagram of the whole process for two cycles when there are failures in the network.

To describe the sensing task scheduling method, we summarize the notation and definitions used in the model as follows.

$\alpha_{i:}$ The fraction of sensing task that is assigned to sensor node SSN^i by the cluster head. It is assumed that every node will be assigned non-zero task, i.e., $0<\alpha_i<1$, and the task for all nodes sums to 1 ($\sum_{j=0}^{n} a_j = 1$).

y_i: A constant that is inversely proportional to the sensing speed of sensor node SSN^i in the network.

z_i: A constant that is inversely proportional to the communication speed of the communication $link_i$ in the network.

T_{ms}: Sensing intensity constant. This is the time it takes for the i^{th} sensor node to accomplish the whole sensing task when $y_i = 1$. The entire assigned sensing task can be sensed on the i^{th} sensor node in time y_iT_{ms}.

T_{cm}: Communication intensity constant. This is the time it takes to transmit the entire sensing task over a link when $z_i = 1$. The entire sub-task can be transmitted over the i^{th} link in time z_iT_{cm}.

T_i: The total time that elapses between the beginning of the scheduling process at $t = 0$ and the time when $sensor_i$ completes its reporting, $i = 1, \ldots, N$.

T_r: The time when the last sensor node finishes responding (finish time or make-span). $T_r = \max(T_1, T_2, \ldots, T_n)$.

ΔT: The delay overhead caused by node's failure.

While gathering data, the cluster head will continuously check the data integrity and detect any missing data and their source, i.e., fault detection and source identification. After the cluster head retrieves the routing and node availability information and decide that the missing data will affect the data integrity, a second cycle will be scheduled to complete the sensing task. If the faulty sensor node recovered from the failure and had all data still available, it will be requested to transmit the data upward while other nodes starting their new sensing task. When the sensor node is not recovered for the consequent cycle or the recovered node does not have all data available, the cluster head reschedules the remaining sensing task among all available sensor nodes like it is a new sensing task. To add in the complexity of the problem, the network structure may change from one cycle to another in ad hoc WSN, as shown in Fig. 4.

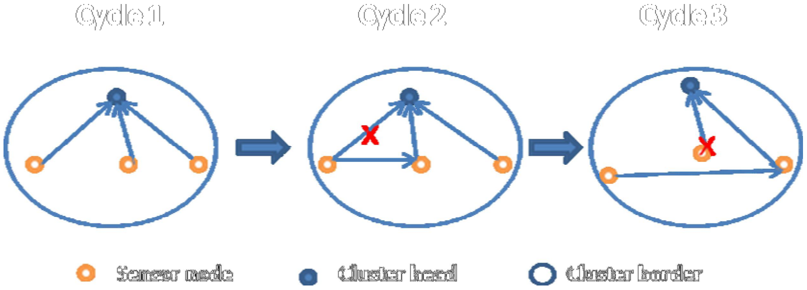

Figure 1. Different network structure between cycles because of fault

To quantitatively model the failure event, we define the failure time “t” as the time from the beginning of result reporting to the failure point. If t is less than 0, i.e. the node fails or loses connection before it begins to send, the entire data is lost. If it is greater than 0, i.e. the node fails or loses connection after sending started, the missing part is proportional to t.

In next section, we derive the closed form solution for single level tree network based on the fault tolerant strategy described above. We assume the network structure remains the same for two consecutive cycles.

III. Fault Tolerant Task Scheduling and Response Time Delay Overhead

Figure 6 shows the time diagram of two cycles for single failure at SSN^k that is either recoverable (a) or un-recoverable (b) during the second cycle.

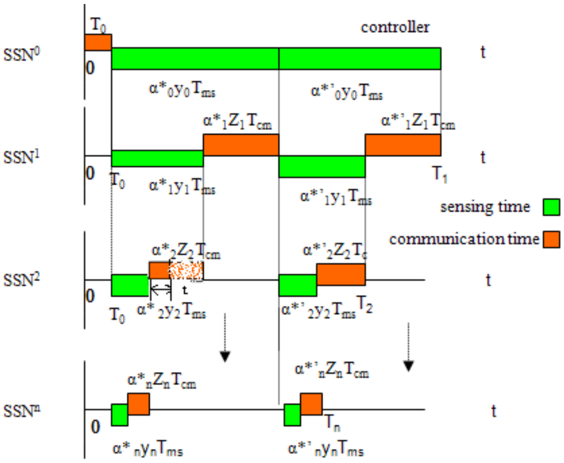

Figure 2(a). The two scheduling cycle when failure node heal before next scheduling cycle

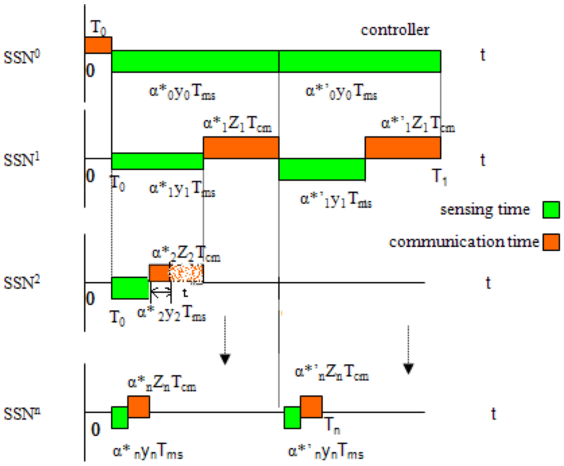

Figure 2(b). The two scheduling cycle when failure node heal after next scheduling cycle

Based on divisible load theory (V. Bharadwaj et al., 2003) and the time diagram shown in Fig. 6, the first cycle of the fault tolerance WSN can be described by a set of linear equations (Eq.1). The solution of the optimum task partition (Eq. 2) and the optimum response time T$_f$ (Eq. 3) the same as the results presented in (H.Liu et al. 2007).

$$\begin{aligned} a^*{}_0 y_0 T_{ms} &= a^*{}_1 y_1 T_{ms} + a^*{}_1 z_1 T_{cm} \\ a^*{}_1 y_1 T_{ms} &= a^*{}_2 y_2 T_{ms} + a^*{}_2 z_2 T_{cm} \\ &\dots \\ a^*{}_{n-1} y_{n-1} T_{ms} &= a^*{}_n y_n T_{ms} + a^*{}_n z_n T_{cm} \end{aligned} \tag{1}$$

$$a^*{}_k = \frac{\prod_{j=k+1}^{n} f_j}{1+\sum_{i=1}^{n}\prod_{j=i}^{n} f_j}, \quad \text{k=0, 1, ..., n-1} \qquad a^*{}_n = \frac{1}{1+\sum_{i=1}^{n}\prod_{j=i}^{n} f_j} \tag{2}$$

$$T_r^* = T_0 + \frac{\prod_{j=1}^{n} f_j}{1+\sum_{i=1}^{n}\prod_{j=i}^{n} f_j} y_0 T_{ms}, n > 0 \qquad (3) \qquad \text{in which } f_k = \frac{y_k + z_k d'}{y_{k-1}},\ d' = \frac{T_{cm}}{T_{ms}}.$$

$$T_r^* = T_0 + y_0 T_{ms}, n = 0$$

The remaining task due to the single failure occurred from node SSN^k which is the base of the second cycle scheduling is: $\Delta a = a_k - t / z_k T_{cm}$ (4)

To study the response delay caused by the faulty node and / or faulty connection, we look into the relationship between the sensing (y_k) and communication (z_k) capability and the failure node position in the reporting sequence.

For a homogenous WSN in which all sensor nodes have the same sensing and communication speed, i.e., $y_i=y_{i+1}$, $z_i=z_{i+1}$, i=0…n, we derive the closed form solution for the fault tolerant task scheduling as shown in Fig. 6.

When the failure recovered before the beginning of the second cycle (Fig. 6(a)), the remaining task Δα is redistributed to all nodes using Equation 1 to 3. The optimal response time can be derived straightforwardly at $\Delta T = \Delta\alpha \cdot T_r$

When the failure cannot be recovered before the beginning of the second cycle (Fig.6(b)), the remaining task will be partitioned without the node SSN^k:

$$\begin{aligned}
a^{*}{'}_{0}\, y_0 T_{ms} &= a^{*}{'}_{1}\, y_1 T_{ms} + a^{*}{'}_{1}\, z_1 T_{cm} \\
a^{*}{'}_{1}\, y_1 T_{ms} &= a^{*}{'}_{2}\, y_2 T_{ms} + a^{*}{'}_{2}\, z_2 T_{cm} \\
&\dots \\
a^{*}{'}_{k-1}\, y_{k-1} T_{ms} &= a^{*}{'}_{k+1}\, y_{k+1} T_{ms} + a^{*}{'}_{k+1}\, z_{k+1} T_{cm} \\
&\dots \\
a^{*}{'}_{n-1}\, y_{n-1} T_{ms} &= a^{*}{'}_{n}\, y_n T_{ms} + a^{*}{'}_{n}\, z_n T_{cm}
\end{aligned} \qquad (5)$$

Together with $\sum_{j=0}^{n-1} a^{*}{'}_{j} = \Delta a$, $y_0=y_1=\ldots=y$, $z_0=z_1=\ldots=z$, the closed form solution for response time delay overhead ΔT_r is:

$$\Delta T_r = \Delta a\,(T_0 + \frac{f^{n-1}}{1+\sum_{i=1}^{n-1} f^{n-i}} y T_{ms}) \qquad (6)$$

From Eq. 6, we conclude that the response time delay overhead ΔT_r is proportional to the remaining sensing task Δα. Together with Eq. 4, it is straightforward to see that the earlier the node fails, the larger its first cycle task portion is, the bigger the response time delay overhead. Based on single level tree network property and the characteristics of the

S4R strategy, we conclude that when fails at the same time t, the latter a sensor node reports, the more impact its failure will have on the response time delay overhead.

IV. Simulation Results for Fault Tolerant Homogeneous WSN

In all of the simulations, we assume a WSN with 10 sensor nodes SSN_0, …, SSN_{10}. We also assume $T_{cm}= T_{ms}= 1$, and $T_0 = 0$, in order to clearly show the effects of $y_{i,}$ $z_{i,}$ on the delay overhead ΔT_r.

In homogeneous network, every node is identical with respect to their sensing and communication capability. Because of this, the response time delay is determined by the parameters such as when the failure happens (t), which node fails, the goodness of the network (y, z), and whether or not the failure is recoverable.

Fig 7 shows the delay overheads caused by the failure of SSN^2 at time of $0.01*T_{cm}$, $0.02*T_{cm}$, and $0.05*T_{cm}$. X-axis denotes the size of the cluster. We can see clearly that as the size of the cluster increases, the effect of failure recovery decreases. That is because task portion for SSN^2 is smaller in a larger cluster, so do the lost packets. It is also shown that for a smaller cluster, even if the failure happens earlier, the packet loss is more severe, a failure recovery can neutralize it and yield smaller delay overhead.

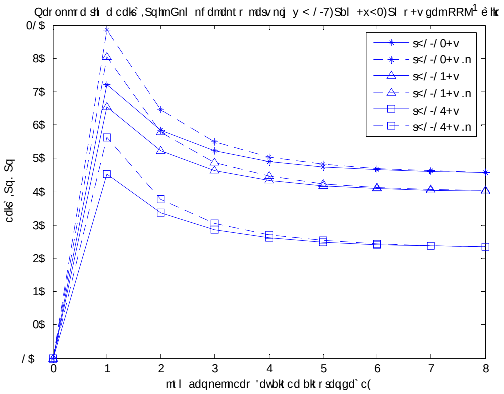

Figure 7. $\Delta Tr/Tr$ when SSN^2 fails at different time z=0.8, y=1

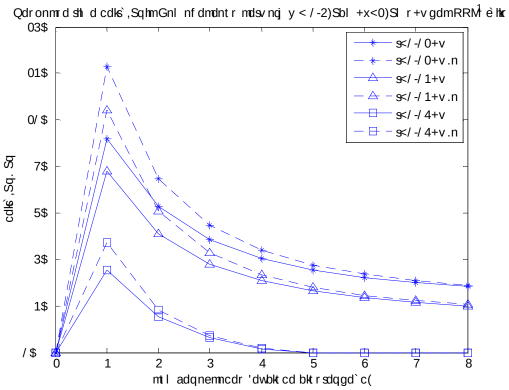

Figure 8. $\Delta Tr/Tr$ when SSN^2 fails at different time z=0.3, y=1

Fig 8 is the same situation for a cluster that has better communication link of $z=0.3*T_{cm}$. In this case, the earlier failure results in more delay overhead but the latter failure works much better. The better link quality has magnified the effect significance of failure. However, a better sensing speed does not have a similar effect, but only decreases the delay overhead because of a faster re-sensing for the missing task. This is shown in Fig 9.

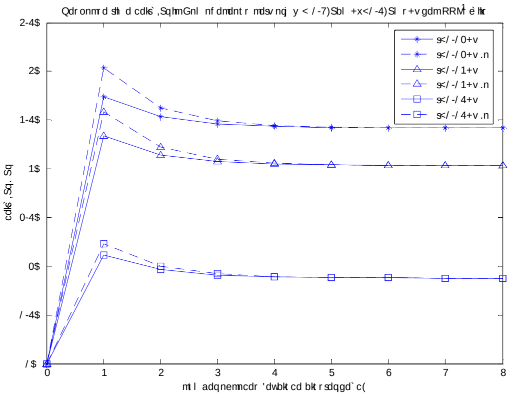

Figure 9. ΔTr/Tr when SSN^2 fails at different time z=0.3, y=1

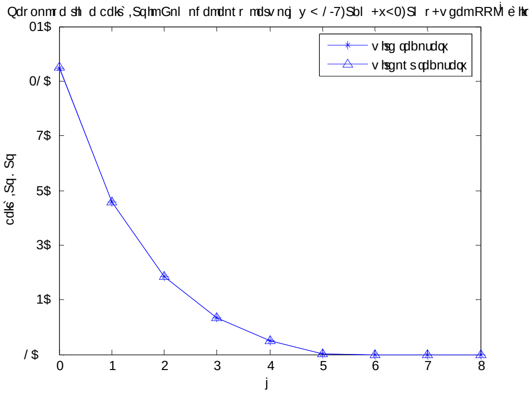

Figure 10. ΔTr/Tr when different SSN^k fails at t=0.01*T$_{cm}$, z=0.8, y=0.5

Fig 10 exams the effect of position in reporting sequence. In a cluster of size 9 (9 nodes and a cluster-head), we look at the different results of failure of SSN^1, SSN^2, …, SSN^9. Like we discussed before, the effect of failure recovery is almost invisible when the cluster size is 10. However, which position the failure node is in the reporting queue is very important. The latter it reports to the cluster-head, a much bigger significance its failure has.

V. Conclusion

In this paper, we presented a fault tolerant task scheduling algorithm and derive the closed form solution for optimal task partitioning that results in minimum response time delay overhead caused by single failure. Two scenarios are considered: failure that can recover within time and those that cannot. The closed form solution and simulation results for a 10 node homogeneous WSN shows that the response time delay overhead is linear proportional to the time of the failure, i.e. the amount of missing data/information that affect the final decision making process. Future work include expand the fault tolerant framework / model developed here to heterogeneous WSN as well as ad hoc WSN where the network structure changes from one cycle to another.

VI. References

Julian W. Gardner, and Vijay K. Varadan, and Osama O. Awadelkarim, Microsensors, (2001), MEMS and Smart Devices, Wiley 2001, ISBN 0-4718-6109-X

Crossbow Technology Inc. http://www.xbow.com

I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci (2002). Wireless Sensor Networks: a Survey. New York, NY, USA: Computer Networks: The International Journal of Computer and Telecommunications Networking, 15 March 2002. p 393 – 419.

S. Gupta, R. Zheng, M.Cheng (2007), ANDES: an Anomaly Detection System for Wireless Sensor Network, MASS 2007, 1-9, Oct. 2007, Pisa, Italy.

Bo Yu, Bin Xiao (2006), Detecting Selective Forwarding Attacks in Wireless Sensor Networks. Greece: IPDPS, 2006. p 3 – 6.

Bryan Parnoy, Adrian Perrigz, Virgil Gligor (2005). Distributed Detection of Node Replication Attacks in Sensor Networks. Proc. of the 2005 IEEE Symposium on Security and Privac, Washington DC USA: IEEE Computer Society, p 1 – 15.

Bharat Bhargava, Weichao Wang (2004), Visualization of Wormholes in Sensor Networks. New York, NY, USA: ACM Press, 2004.

Nithya Ramanathan, Kevin Chang, Rahul Kapur, Lewis Girod, Eddie Kohler, Deborah Estrin (2005), Sympathy for the Sensor Network Debugger. New York, NY, USA: ACM Press, November 2-4, 2005. p 1 – 12.

W. L. Lee, A. Datta, and R. Cardell-Oliver (2006), Network Management in Wireless Sensor Networks, to appear in Handbook on Mobile Ad Hoc and Pervasive Communications, edited by M. K. Denko and L. T. Yang, American Scientific Publishers, 2006

Jonathan L. Brediny, Erik D. Demainez, MohammadTaghi Hajiaghayiz and Daniela Rus (2005), “Deploying Sensor Networks with Guaranteed Capacity and Fault Tolerance”, *MobiHoc’05*, May 25–27, 2005, UrbanaChampaign, Illinois.

Y. Wang and H. Wu (2005), “DFT-MSN: The Delay Fault Tolerant Mobile Sensor Network for Pervasive Information Gathering.” Tech Report, CACS, University of Louisiana at Lafayette, 2005. http://www.cacs.louisiana.edu/˜wu/paper/DFT-MSN.pdf.

Y. Wang, F. Lin, and H. Wu (2005), “Efficient Data Transmission in Delay Fault Tolerant Mobile Sensor Networks (DFT-MSN)”, *13th IEEE International Conference on Network Protocols*, Boston, Massachusetts, November 6-9, 2005.

S. Chessa and P. Maestrini (2005), “Fault recovery mechanism in single-hop sensor networks”, *Computer communications*, **28**(17), 1877-1886.

F. Koushanfar and M. Potkonjak and A. Sangiovanni-Vincentelli (2002), "Fault tolerance techniques in wireless ad-hoc sensor networks", *UC Berkeley technical reports*. citeseer.ist.psu.edu/article/koushanfar02fault.html

G. Chockler, M. Demirbas, S.Gilbert, C. Newport and T. Nolte (2005), “Consensus and Collision Detectors in Wireless Ad Hoc Networks”, *Proceedings of the 24th ACM Symposium on Principles of Distributed Computing*, July 17-20, 2005, Las Vegas, Nevada

H. Liu, X. Yuan and M. Moges (2007), “An Efficient Scheduling Method for Improved Network Delay in Distributed Sensor Networks”, *3rd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities*, Orlando, Florida, May 21-23, 2007.

Kemal Akkaya, Mohamed F. Younis (2005), “A survey on routing protocols for wireless sensor networks”, *Ad Hoc Networks*, **3**(3): 325-349

V. Bharadwaj, D. Ghose, T.G. Robertazzi (2003), “Divisible Load Theory: A new paradigm for load scheduling in distributed systems”, *Cluster Computing*, **6**(1), 7-18.