A Decentralized Fault Diagnosis System for Wireless Sensor Networks

Anmol Sheth, Carl Hartung and Richard Han
Department of Computer Science
University of Colorado, Boulder
Email: anmol.sheth@colorado.edu, carl.hartung@colorado.edu, rhan@cs.colorado.edu

Abstract

The irregularities of a low cost wireless communication interface, changing environmental conditions, in-situ deployment and scarce resources make management, monitoring and troubleshooting performance of a sensor network a challenging task. In this paper we present the design of a decentralized fault diagnosis system for a wireless sensor network. Our system distinguishes between multiple root causes of degraded performance and provides efficient feedback into the network to troubleshoot the fault.

1 Introduction

This paper presents the design and implementation of a distributed fault diagnosis and troubleshooting system for wireless sensor networks (WSNs). A diagnostic system is important for WSNs so that an end user can clearly monitor, manage, and troubleshoot the performance of the network. As WSN deployments increase, the ability to diagnose the network becomes especially critical, since WSNs are deployed in situ and are therefore subject to an even larger variety of environmentally induced faults than the Internet.

This paper focuses on diagnosing the most common or first-order problems in WSN deployments that have been reported in the literature. In particular, recent work has observed that one of the first-order problems most commonly experienced in deployed WSNs is reduced data throughput. The root causes of this reduced sensor data throughput have been attributed in large part to either hidden terminal conflicts, congestion, and/or wireless coverage/connectivity that exhibits irregular, asymmetric, and/or time-varying behavior. However, no coherent framework has been offered to methodically and efficiently differentiate between these prominent root causes of the same first-order problem. This paper focuses on providing such a framework, so that an end user of a WSN can clearly distinguigh between these quite different causes. Misdiagnosis and misapplication of a solution results not only in wasted time, energy, and communication bandwidth - all scarce resources in a WSN - but also can exacerbate the problem, resulting in lower data throughput.

Existing architectures for providing diagnosis in a WSN are largely centralized . Existing systems like Sympathy[2], SCALE[1] and EmStar/EmTos [3] require diagnosis data to be periodically reported back to a centralized base station where diagnosis takes place. There are several limitations with this approach. First, the centralized architecture of reporting diagnostic information back to a base station is not a scalable solution. As the size of the sensor network increases, more and more diagnostic information is fed back to the base station. This control messaging decreases the longevity of the sensor network, and also results in more packet overhead and collisions that will interfere with timely delivery of sensor data. Second, in many cases, summaries of diagnostic data are insufficient to accurately diagnose the problem from a remote base station. For example, detailed information in the form of local packet traces is needed in order to accurately diagnose hidden terminals as the root cause of reduced data throughput. Such a high reporting volume will exacerbate the energy exhaustion effects of a centralized approach. Third, the existence of a high data rate backchannel cannot be assumed for many deployed WSNs. In many cases, a wired backbone is assumed to be available to provide the high data rate backchannel, e.g. motelab, which clearly is not feasible in widely distributed outdoor WSNs such as GDI.

In this paper we introduce a new *decentralized* architecture for diagnosing faults in a sensor network and a new algorithm for effectively differentiating between three root causes of commonly experienced problem of reduced data throughput. The following two sections provide the details of the architecture and the diagnosis algorithm.

2 Decentralized Diagnostic Architecture

The architecture of the diagnostic system takes a decentralized approach. Nodes monitor behavior locally. When abnormal behavior is detected, nodes execute an in-network

0-7803-9466-6/05/$20.00 ©2005 IEEE

diagnostic procedure that is local in nature, i.e. nodes query their neighbors for diagnostic information. These localized single hop interactions allow the diagnostic framework to scale easily to much larger and denser sensor networks.

In addition to a decentralized architecture, our diagnostic framework must be able to differentiate among multiple root causes of reduced data throughput. The most common root causes of reduced data throughput are hidden terminals, congestion, and link asymmetry. These root causes manifest themselves identically at the higher layers of the network stack as the single symptom of a drop in data rate. In order to distinguish between these quite distinct root causes, the diagnosis needs lower layer information. Thus, our solution incorporates cross-layer metrics from multiple layers in the network stack. In particular, our solution incorporates diagnostic information from both the MAC and network layers.

Existing literature in WSN's have proposed solutions to resolve each of the individual problems in isolation. However, these problems in practice can occur simultaneously in time and space. Our key contribution is providing a general diagnostic framework that distinguishes between these problems so that the correct troubleshooting operation(s) can be performed.

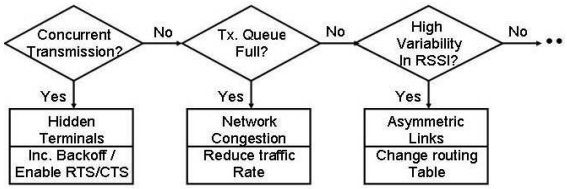

Figure 1. Fault diagnosis algorithm

Figure 1 illustrates the distributed fault diagnosis algorithm that executes on each sensor node. The algorithm is subdivided into multiple stages, with each stage diagnosing a particular root cause. Each stage consists of two components, namely the detection algorithm for determining whether the associated root cause is the source of the degraded performance, and the remedy or solution to execute to mitigate the fault. The stages are ordered so that the diagnosis can be performed in a single pass. The *order of the fault detection* is very important. For example, both hidden terminals and network congestion have the same symptom at the network layer of causing overflowing of transmit buffer queues and high contention of the wireless channel. If the diagnosis algorithm initiates detection of congestion prior to detection of hidden terminals, then the algorithm will conclude erroneously that the root cause was congestion. Fortunately, there is additional diagnostic information that can be gleaned from the MAC layer in the case of hidden terminals to differentiate between the two root causes. Thus, in the figure, hidden terminals are diagnosed prior to congestion.

Each node monitors the observed traffic forwarded from all nodes below its position in the WSN routing tree, i.e. all nodes further from the base station. The diagnosis algorithm is initiated only at the parent node of the faulty link. The parent node executes the single pass fault detection algorithm when it detects an abnormal drop in data rate. To improve the efficiency of this approach, the parent node probes its children to see if they also observe this reduced throughput. If the answer is yes, then the parent node refrains from initiating the diagnosis. Thus, only the immediate parent of the faulty link initiates the diagnosis, which iterates through each fault and checks for its presence. Once a fault is detected, corrective measures are taken locally. A summary of the action can optionally be sent up to the base station where this information is logged. Section 3 discuss the details of the diagnosis algorithm.

A variety of other scenarios are also handled by our algorithm. If there are multiple root causes, then after the first stage has executed its remedy, there will still be a persistent drop in throughput. At this point the algorithm executes again, finds that the first stage's detection algorithm is passed, and then executes the second stage, and so on. Eventually, the algorithm will iteratively eliminate all root causes by applying the appropriate remedies. Another scenario is if the algorithm observes a drop in throughput but is unable to identify the root cause. In this case, a summary may be optionally sent back to the base station.

Over time, the corrective measures taken by a particular node may no longer apply. In this case, the algorithm checks periodically for the fault that triggered the corrective measure, and increases the period exponentially if the problem persists. This ensures that the network is not too aggressive in disabling the fault mitigation procedures. If the condition no longer holds, then the node disables the troubleshooting action and reverts to its previous state.

3 Diagnosis Algorithm

The low cost wireless communication radio interface, in-situ deployment, changing environmental conditions and scarce resources cause a wide range of anomalies in a sensor network. Diagnosis of the root cause of sensor network faults requires synthesis between multiple layers of the stack and fine-grained diagnostic information. In this section we discuss the details of the diagnosis algorithm outlined in section 2.

3.1 Detection of Hidden Terminals

Hidden terminals arise when neighbor nodes cannot correctly sense the medium to be busy and transmit simultaneously. This results in collisions of the transmitted packets

which leads to either both, or one of the packets being received incorrectly. Hidden terminals can be avoided by selectively enabling collision avoidance like RTS/CTS on the nodes which are hidden from each other.

Enabling RTS/CTS based collision avoidance on every node in the affected part of the network, or globally in the entire network is not ideal because of the significant overhead of transmitting a RTS and CTS for each data frame. We propose a much more lightweight mechanism where collision avoidance is only enabled on the nodes which are actually involved in concurrent transmissions.

As part of the diagnosis algorithm, the node selected by the base station first commands its one-hop neighbors to synchronize their clocks and also start recording the time stamps of the packets being transmitted. This synchronization need not be very accurate and even an error up to a few milliseconds is tolerable. Upon receiving this request neighboring nodes start logging the timestamp of packets transmitted in a fixed length buffer. After a fixed interval, the leader node requests the array of timestamps sequentially from each neighbor node. On receiving these timestamps the leader can very simply identify the neighbor nodes that are involved in concurrent transmissions. Nodes whose transmissions overlapped were not able to sense the channel to be busy correctly, and hence are hidden from each other. The above *in-network* diagnosis reduces the overhead of transmitting fine-grained information like packet timestamps over a multi-hop network up to the base station. Also, by selectively enabling RTS/CTS only at the selected leader node reduces the overhead of the system significantly. Collisions are minimized while the bandwidth is efficiently utilized for transmitting useful data packets.

It is important to note that this approach is not limited to detecting hidden terminals only among the nodes which share the same upstream parent. This method allows detecting hidden terminals in the *entire* radio range of the selected leader. The leader could then selectively enable collision avoidance only on the nodes which are hidden from each other. All nodes adhere to the RTS/CTS control messages and defer their transmission, however the control messages are only transmitted by the specific nodes which are hidden from each other. This selective enabling of collision avoidance significantly reduces the overhead on the system.

3.2 Detection of Network Congestion and Asymmetric Links

A large number of algorithms have been proposed to mitigate congestion in sensor network. As an initial implementation we employ the hop-by-hop flow control algorithm. This technique requires the node with a nearly full transmit queue to signal its neighbors to stop generating traffic by setting a congestion bit in its header. Although this technique incurs minimal overhead in form of control messages, it should not be enabled by default. As discussed before, hidden terminals and network congestion have the same symptoms. Having congestion avoidance enabled by default could lead to nodes unnecessarily applying backpressure to their child nodes, where the root cause of the problem is hidden terminals. This would potentially slow down the traffic generation rate of the entire network and still not mitigate the problem.

Detection of asymmetric links can be achieved by each node maintaining link qualities to its neighbor nodes. Thus in a tree topology, asymmetric links can be diagnosed by observing a mis-match between data received and transmitted by the child nodes. Troubleshooting an asymmetric link requires the parent node to command the child node to change its parent node.

4 Conclusion

In this paper we presented the design of a decentralized fault diagnosis system for WSN. The system enables efficient management of a WSNs by diagnosing the true root cause of a degraded performance by combining multiple sensor observations. This diagnosis is performed *in-network*, and hence requires minimal data collection at the centralized base station. Differentiation between multiple root causes aids in effective troubleshooting of the fault. Finally we propose a single pass algorithm to diagnose the root cause faults. The selective reactive troubleshooting of faults results in a significant improvement in performance of the network as compared to existing mechanisms of proactive troubleshooting.

References

[1] A. Cerpa et al. “SCALE: A tool for Simple Connectivity Assessment in Lossy Environments”. CENS Technical Report 0021, September 5, 2003.

[2] N. Ramanathan et al., “Towards A Debugging System for Sensor Networks”, In Proceedings of International Journal for Network Management, 2004

[3] L. Girod et al., “A System for Simulation, Emulation, and Development of Heterogenous Sensor Networks”, In Proceedings of the 2nd ACM Conference on Embedded Networked Sensor Systems (SenSys’04), 2004.