Real-time Forest Fire Detection with Wireless Sensor Networks

Liyang Yu, Neng Wang
Dept. of Computer Science and Technology
East China Normal University
Shanghai, China
{lyyu, nwang}@cs.ecnu.edu.cn

Xiaoqiao Meng
Dept. of Computer Science
University of California, Los Angeles
Los Angeles, U.S.A.
xqmeng@cs.ucla.edu

***Abstract*—In this paper, we propose a wireless sensor network paradigm for real-time forest fire detection. The wireless sensor network can detect and forecast forest fire more promptly than the traditional satellite-based detection approach. This paper mainly describes the data collecting and processing in wireless sensor networks for real-time forest fire detection. A neural network method is applied to in-network data processing. We evaluate the performance of our approach by simulations.**

Keywords-wireless sensor networks; neural network

I. INTRODUCTION

Wireless sensor networks have been attracting many research efforts during the past few years. Sensor networks, usually composed of a few sinks and a large quantity of inexpensive and small sensor nodes, have been deployed in a variety of applications [5] such as habitat monitoring [14], forest fire detection, etc.

In this paper we explore the use of wireless sensor network technology in real-time forest fire detection. The forest fire is a fatal threat in the world: it is reported [3] that a total of 77,534 wildfires burned 6,790,692 acres in USA for 2004. Satellite-based monitoring is a popular method to detect forest fire now [13]. But the long scan period and low resolution of satellites [1] restrict the effectiveness of the satellite-based forest fire detection. Moreover, satellites usually cannot forecast forest fires before the fire is spread uncontrollable.

In this paper, we propose a new real-time forest fire detection method by using wireless sensor networks. Our goal is to detect and predict forest fire promptly and accurately in order to minimize the loss of forests, wild animals, and people in the forest fire. In our proposed paradigm, a large number of sensor nodes are densely deployed in a forest. Sensor nodes collect measured data (e.g., temperature, relative humidity) and send to their respective cluster nodes that collaboratively process the data by constructing a neural network. The neural network takes the measured data as input to produce *weather index*, which measures the likelihood for the weather to cause a fire. Cluster headers will send weather indexes to a manager node via the sink. Then the manager node concludes the forest fire danger rate based on received weather indexes and some other factors. In certain emergent situations, sensor nodes may detect smoke or abnormal temperature. They will directly send an emergence report to the manager node.

We apply the neural network method to the data processing in the cluster header in our design. The neural network has been applied to a lot of application [12] in the past two decades, including satellite-based forest fire detection [7]. To our best knowledge, there exists no previous work that has proposed in-network processing using neural network in wireless sensor networks.

We exploit the essence of neural network that complex data processing can be done by performing simple calculations at many organized single nodes. Such an approach fits sensor networks well because individual sensor nodes have limited computing capability. Moreover, the constructed neural network operates on vast raw data and extracts small amount of information useful for the final decision; thus, both communication overhead and energy consumption are significantly reduced.

In addition to real-time forest fire detection, the proposed sensor network approach can forecast potential forest fires, and provide helpful information to extinguish forest fires and investigate the cause of the fires. These features make our proposed paradigm superior to the traditional satellite-based forest fire detection approach.

The remainder of this paper is organized as follows. Section II describes the network model for forest fire detecting. Section III presents how to process the collected data efficiently. The simulation results will be showed in Section IV. In Section V, we discuss related issues. Finally, Section VI concludes this paper.

II. SENSOR NETWORK FOR FOREST FIRE DETECTION

In our forest fire detection method, sensor nodes collect measurement data such as relative humidity, temperature, smoke, and windy speed — all these factors are required for determining the forest fire danger rate. According to the United States National Fire Danger Rating System [4], there exist some other factors including lightning activity level, man-caused risk, slop class, and vegetation class. In this paper, we will not discuss them since they are relatively stable.

0-7803-9335-X/05/$20.00

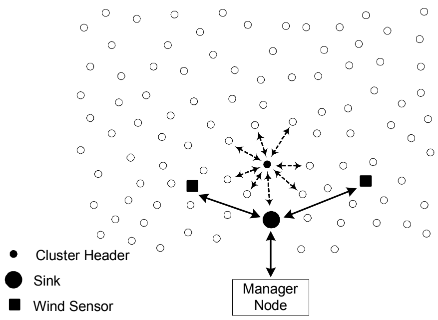

Figure 1. A wireless sensor network for real-time forest fire detection.

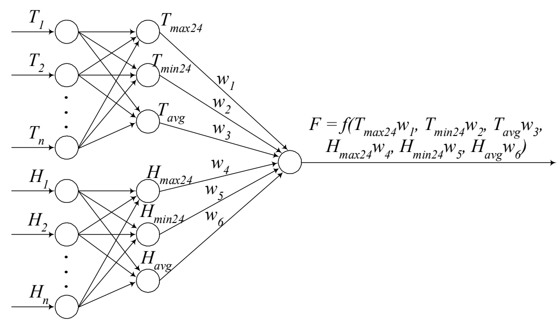

Figure 2. RR packet processing using neural network method.

We describe the proposed sensor network paradigm in Fig. 1. A large number of sensor nodes are densely deployed in the forest. These sensor nodes are organized into clusters so that each node has a corresponding cluster header. Sensor nodes can measure environment temperature, relative humidity and smoke. They are also assumed to know their location information by equipments such as GPS. Every sensor node sends measurement data, as well as the location information, to the corresponding cluster head. The cluster header calculates the weather index using a neural network method, then it further sends the weather index to the manager node via the sink. The sink is connected to a manager node via a wired network. A few wind sensor nodes are manually deployed over the forest and connected to the sink via wired networks to detect wind speed.

The manager node provides two types of information to users: (1) emergence report for abnormal event (e.g. smoke or extremely high temperature is detected); (2) real-time forest fire danger rate for each cluster based on the weather indexes from the cluster header and other forest fire factors. Users can also query the current temperature and humidity data in particular cluster area. We don't discuss the detail related to wind sensor nodes and the manager node in this paper.

III. Design

In this section, we present details of our design, including clustering, data collection, and data processing.

A. Clustering

We choose a clustering algorithm [15] as the routing protocol in the wireless sensor network. Clustering routing also meets the demand of the neural network method. More specifically, we can treat sensor nodes, cluster headers as input layer nodes, hidden layer nodes in the neural network respectively. In each cluster, the weather index, which is generated by the neural network method, will be sent to the manager node via the sink. The cluster header is selected dynamically to balance the energy consumption of all sensor nodes. We will discuss this in more details in Section III-D.

B. Data Collection

Each node can generate three classes of data packets: (1) Regular Report (RR); (2) Query Response (QR); (3) Emergence Report (ER). A tag is inserted into each packet to identify the packet class.

Each node periodically collects sensing data and encapsulates them into a RR packet whose destination is the respective cluster head. The QR packet is only sent to the sink by part of nodes immediately after getting a query packet from the sink. A node that detected an abnormal event, e.g. smoke is detected, will immediately generate and send the sink an ER packet containing the information related to the abnormal event.

C. Data Processing

Once a cluster header receives a packet from other nodes, it processes the packet according to the packet type. In the rest of this section, we illustrate how each type of packets is processed.

When a cluster header receives an ER packet that contains an abnormal event, it will forward the ER packet to the sink as rapidly as possible. The ER packet has the highest priority to be forwarded by intermediate nodes towards the sink, that is, after an intermediate node receives an ER packet, the node will insert the new ER packet ahead of any other types of packets in the outing packet queue.

When a cluster header receives a QR packet, the header can apply the aggregation algorithms (e.g. [11]) to process the QR packet.

The processing of RR packets in cluster headers is more involved. A constructed neural network will take all the received RR packets as input and generate a metric called *weather index*. Such a weather index will be encapsulated into a processed report (PR) packet and be sent to the sink.

We use Fig. 2 to further illustrate the neural network method. There is a cluster header with n cluster member nodes. Each cluster member node N_i ($1 \le i \le n$) periodically sends the cluster header a RR packet that contains data related to temperature or relative humidity, denoted by T_i and H_i respectively. The cluster header periodically computes T_{max24}, T_{min24}, T_{avg}, H_{max24}, H_{min24}, and H_{avg} once per Δt minutes. T_{max24} and T_{min24} denote the highest temperature and the lowest temperature in the past 24 hours respectively. H_{max24} and H_{min24} denote the highest relative humidity and the lowest relative humidity in the past 24 hours respectively. T_{avg} and H_{avg} denote current average temperature and the average relative humidity respectively. As a medium output of the neural network, the cluster header can get a weather index F which equals

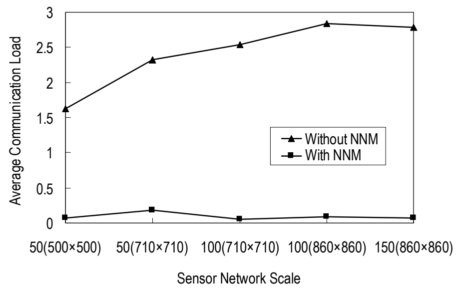

Figure 3. Average communication load with different network scale.

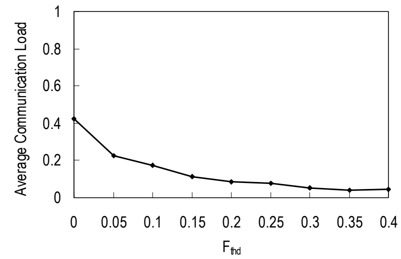

Figure 4. Average communication load with different threshold.

$f(T_{max24}w_1,\ T_{min24}w_2,\ T_{avg}w_3,\ H_{max24}w_4,\ H_{min24}w_5,\ H_{avg}w_6) = T_{max24}w_1 + T_{min24}w_2 + T_{avg}w_3 + H_{max24}w_4 + H_{min24}w_5 + H_{avg}w_6$.

The weather index F will be encapsulated into a PR packet and sent to the manager node. The administrator can set a threshold F_{thd} to decide whether the cluster header should send the new weather index F to the sink or not. More specifically, suppose F' is the last weather index that has been sent to the sink, and F is the new weather index. If $\left|F' - F\right| \geq F_{thd}$, the cluster header will send the new weather index F to the sink, otherwise it will not.

Parameters w_i $(1 \leq i \leq 6)$ are the key factors for processing data via neural network method. They may be modified through a particular learning algorithm. The learning procedure will be processed in the manager node. When there is a new set of parameters w_i, the sink will propagate them to all cluster headers.

D. Cluster header Selection and Parameters Handover

In order to balance the energy consumption of all sensor nodes, we use a cluster routing algorithm [15] to dynamically select cluster headers based on the residual energy and a few parameters such as node degree.

Because the cluster headers perform most of the in-network processing tasks, they are easily exhausted. Therefore it is necessary to shift the workload from an overloaded cluster header to a newly selected one.

The new selected cluster header will send an announcement message before replacing the overloaded header. The overloaded header should send the newly selected one a handover packet, which includes all parameters for the in-network processing after receiving the announcement message. The handover packet should contain the following parameters: location information, w_i $(1 \leq i \leq 6)$, T_{max24}, T_{min24}, H_{max24}, H_{min24}, and F_{thd}. If a newly selected header received more than one handover packets from different overloaded headers, it chooses one set of parameters from the cluster header who has the minimum Euclidean distance away from the newly selected header.

IV. Performance Evaluation

We evaluate the performance of the in-network processing using neural network method through simulations in this section. We implemented the in-network processing using neural network method in *ns-2.26* [2]. Each sensor node generates one RR packet per two seconds. Each simulation run lasts 200 seconds, and each result is averaged over three random network topologies.

We first study the efficiency of our approach by comparing it with the scenario without the neural network method. Five simulation scenarios with different number of nodes and different field size are chosen as (50, 500m×500m), (50, 710m×710m), (100, 710m×710m), (100, 860m×860m), and (150, 860m×860m). The threshold F_{thd} is 0.2 in neural network method. Fig. 3 shows the average communication load in five different scenarios. The value of communication load is equal to N_t/N_{RR}, where N_t denotes the number of one-hop transmissions from cluster headers to the sink and N_{RR} denotes the number of RR packets sent by all sensor nodes. The value of communication load denotes the communication cost for send the information contained in a RR packet to the sink. Lower average communication load means less energy consumed for forwarding RR packets. We make two observations. First, The average communication load is very significant saved with the neural network method. The ratio between the average communication load with neural network method and the one without neural network method varies from 2.5% (the result of the scenario with 100 sensor nodes distributed in a 710m×710m field) to 8% (the result of the scenario with 50 sensor nodes distributed in a 710m×710m field). Second, as the sensor network scale increases, the average communication load varies unclearly in the scenarios with neural network method, while it increases clearly in the scenarios without neural network method.

We next evaluate the average communication load impacted by different threshold parameter F_{thd}. We vary the threshold F_{thd} from 0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35 to 0.4 that are between a safe scales to keep the accuracy of the processing result using the neural network method in our design. In this scenario, 200 sensor nodes are distributed in a 1000m×1000m field. Fig. 4 plots the average communication load, which tends to decrease when the threshold F_{thd} increases. If the threshold F_{thd} is equal to zero, the cluster header will send all the processing result, i.e. the weather index, to the sink. The simulation result all show that threshold F_{thd} should not be kept in a reasonable scale because the big threshold F_{thd} will reduce the accuracy of collected data but could not give more significant energy saving. According to the simulation result, 0.2 is a good choice for F_{thd} in our design.

V. Related Work

We categorize existing related work into two classes: in-networking processing and forest fire detecting.

A. In-networking Processing

In-networking processing [6][9][11] can reduce the communication overhead and save energy consumption in wireless sensor networks. In the past few years, a variety of in-network processing aspects have been explored. S. Madden *et al.* [11] proposed an in-network aggregation approach that is driven by a generally-purposed SQL-style interface. Their approach can execute queries over sensor data and provide opportunities for optimization. Hong and Prasanna [9] focused on in-network processing from an algorithm perspective. They devise a decentralized adaptive algorithm to maximize the throughput of a class of in-network processing applications. Arici *et al.* proposed PINCO [6], an in-network data compression scheme for data collection in wireless sensor networks. PINCO combines sensor data into groups of data through a pipelined compression scheme; it reduces redundancy in the collected data.

Our proposed neural network method is a new type of in-network processing operation. Such an operation extracts high-level information from the vast raw data. Thus it is more efficient than other in-network processing approaches in certain scenarios.

B. Forest Fire Detection

Most existing forest fire detection systems rely on the satellite imagery [1][8][10][13]. The first Moderate-resolution Imaging Spectroradiometer (MODIS) instrument [1], was launched in December 1999. MODIS provides global daily forest fire products based on the satellite imagery. MODIS views the entire surface of the Earth every one to two days. And its detectors acquire data at three spatial resolutions: 2 bands in 250m, 5 bands in 500m, and 29 bands in 1,000m. It is far away from providing real-time forest fire detection. In [8][13], the images from MODIS and AVHRR are used to evaluate forest fire risk and detect forest fire in China and Canada respectively. T. J. Lynham et al. [10] reviewed potential requirements for space-based observations in fire management including fuel mapping, risk assessment, detection, monitoring, mapping, burned area recovery, and smoke management. Weather condition (e.g. clouds) will seriously decrease the accuracy of satellite-based forest fire detection as the limitations led by the long scanning period and low resolution of satellites.

Our proposed forest fire detection system consists of a vast amount of inexpensive and small sensor nodes. Compared with the satellite imagery based approach, our design can detect forest fire more promptly and forecast the forest fire danger rate accurately.

VI. Conclusion

In this paper, we present a wireless sensor network paradigm for real-time forest fire detection. We use neural networks to prolong the lifetime of the sensor network. The simulation results show that our in-network processing approach is efficient to reduce communications between sensor nodes. We believe our neural network based in-network processing approach can be applied to other monitoring and detecting sensor networks.

Acknowledgments

We would like to thank all group members of Wireless Networking Group (WiNG) at UCLA for their help during the development of this project. Special thank to Starsky Wong for his help on the ns-2 simulation.

Liyang Yu appreciates that Dr. Songwu Lu sponsored Yu's study at UCLA as a visiting scholar and gave Yu a lot of helpful suggestion on research.

References

[1] http://modis.gsfc.nasa.gov/, MODIS Web.

[2] http://www.isi.edu/nsnam/ns/, ns-2 Network Simulator.

[3] http://www.nifc.gov/fireinfo/2004/index.html, "Wildland Fire Season 2004 Statistics and Summaries," National Interagency Coordination Center.

[4] http://www.wrh.noaa.gov/sew/fire/olm/nfdrs.htm, "National Fire Danger Rating System."

[5] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," Computer Networks, vol. 38, no. 4, pp. 393-422, March, 2002.

[6] T. Arici, B. Gedik, Y. Altunbasak and Ling Liu, "PINCO: a Pipelined In-Network COmpression Scheme for Data Collection in Wireless Sensor Networks," in Proc. IEEE Int. Conf. Computer Communications and Networks, 2003.

[7] A. Fernandes, A. Utkin, A. Lavrov, and R. Vilar, "Development of Neural Network Committee Machines for Forest Fire Detection Using Lidar," Pattern Recognition, vol. 37, no. 10, pp. 2039-2047, 2004.

[8] G. Guo and M. Zhou, "Using MODIS Land Surface Temperature to Evaluate Forest Fire Risk of Northeast China," IEEE Geoscience and Remote Sensing Letters, vol. 1, no. 2, April 2004.

[9] B. Hong and V. K. Prasanna, "Optimizing a Class of In-network Processing Applications in Networked Sensor Systems," in Proceedings of IEEE International Conference on Mobile Ad-hoc and Sensor Systems, October 2004.

[10] T. J. Lynham, C. W. Dull, and A. Singh, "Requirements for space-based observations in fire management: a report by the Wildland Fire Hazard Team, Committee on Earth Observation Satellites (CEOS) Disaster Management Support Group (DMSG)," in IEEE International Geoscience and Remote Sensing Symposium, vol. 2, pp. 762-764, June 2002.

[11] S. Madden, R. Szewczyk, M. Franklin, and D. Culler. "Supporting aggregate queries over ad-hoc wireless sensor networks," in IEEE Workshop on Mobile Computing Systems and Applications, 2002.

[12] K. Mehrotra, C. K. Mohan, and S. Ranka, Elements of Artificial Neural Networks, the MIT Press, 1997.

[13] Z. Li, S. Nadon, J. Cihlar, "Satellite detection of Canadian boreal forest fires: development and application of the algorithm," International Journal of Remote Sensing, vol. 21, no. 16, pp. 3057-3069, 2000.

[14] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. M. Mainwaring, and D. Estrin, "Habitat monitoring with sensor networks," Communications of the ACM, vol. 47, no.6, pp. 34-40, June 2004.

[15] O. Younis and S. Fahmy, "Distributed Clustering in Ad-hoc Sensor Networks: A Hybrid, Energy-Efficient Approach," in Proceedings of IEEE INFOCOM, March 2004.