

SEARCH IN SENSOR NETWORKS: CHALLENGES, TECHNIQUES, AND APPLICATIONS

Jennifer L. Wong, Miodrag Potkonjak

University of California, Los Angeles, Computer Science Department, Los Angeles, CA 90095-1596

ABSTRACT

Recently embedded wireless sensor networks (EWSN) have emerged as a solution to bridge the gap between the physical (and chemical and biological) world and the information world of the Internet. Deployment and effective use of EWSN poses numerous conceptual and technological challenges. In particular, there is a genuine need for the coordinated use of techniques from a variety of scientific and engineering fields, including DSP, computer science, information theory, communication, and applied mathematics.

We address the problem of search in sensor networks using techniques from DSP and computer science. The search problem encompasses tasks such as indexing, stream searching, information filtering, semantic and syntax compression, profiling and recommending. The development of search tools will greatly facilitate efforts in many applications of sensor networks, including event monitoring, low power operation, and physical space modelling. EWSN also poses unique challenges. For example, in the case of power and storage limitations, the key requirement is distributed space localized processing. The technical highlights of the paper include the first statistical stream search technique, the first technique for simultaneous search of correlated streams and the first technique for effective acquisition of correlated information under bandwidth and power constraints.

1. INTRODUCTION

Spectacular architecture and technology progress in computers, storage, and mainly optical communication enabled the development of the Internet. The Internet can be viewed as a global world of information because of its increasing percentage of data, information and knowledge which is available to everyone. The development of the Internet has greatly facilitated not only the business and entertainment worlds, but also the storage and dissemination of scientific information.

In the last decade, sensors and wireless communication started their exponential growth in performances with exponential reduction in cost and energy. This development will enable the development and deployment of wireless sensor networks.

In order to realize the full potential of sensor networks numerous exceptionally challenging problems have to be solved. Research and development communities will need

to design new node and network architectures, new operating systems and protocols, new DSP, and continuous and discrete optimization algorithms. One of the especially interesting and important tasks is to develop new information collection and dissemination mechanisms, information architectures, algorithms and software. We believe that, in particular, the development of new search algorithms is crucial.

The area of searching in EWSN includes tasks such as indexing, stream searching, information filtering, semantic and syntax compression, profiling, and recommending. The development of search tools will greatly facilitate the efforts in many applications of sensor networks, including event monitoring, low power EWSN operation, and physical space modelling.

Searching on the Internet and in EWSN share some common points, however there are also a number of sharp difference. Technological constraints on sensor networks will have a high impact on how various tasks are realized. For example, energy, bandwidth, and storage requirements in an unreliable communication environment will impose a number of technical challenges not found in the Internet. The data in sensor networks also has intrinsically different nature than data on the Internet. In EWSN, we have streams of noisy data often with crucially important space and time attributes.

In the remainder of the paper, we will present a global framework for addressing search in EWSN and formulate and solve three specific search related problems: efficient stream search, distributed search of correlated streams, and distributed data collection. Related work includes efforts in distributed systems[5], WSN[3], and Internet Search[2, 4].

2. SEARCH IN SENSOR NETWORKS: INFORMATION ARCHITECTURE

A wireless sensor network (WSN) is a distributed system which consists of a number of sensor nodes. Each node has one or more sensors such as a thermometer, manometer, seismic sensor, light detector, microphone, or chemical and biological sensors.

The data rate can vary significantly between types of sensors. For example, the temperature sensor works at a frequency significantly less than 1 Hz and generates less than 1 byte of information at one time, while a high resolution camera can generate 10's of megabits of information per

Beginning from right-to-left we check the pattern against the string by comparing the W against the T in the string. The characters do not match, and T appears nowhere in the pattern. As a result the skip value will be four, and we compare the W to the fourth character after the T. This time, the W doesn't match the B, but B does appear in the pattern. Since B appears in the pattern, the skip value will be 3. As a result the B in Boy and the B of our pattern now line up. We compare the W in the pattern and see that it doesn't match because of the space in the string. The skip value is 4, and the pattern now lines up with the word BLUE in the string. We continue by comparing again the W, which doesn't match with E and E is not in the string. We skip forward four characters, to compare the pattern with the O in BLOW of the string. As a result of the mismatch we skip forward only one character and then compare each of the letters from the right to the left until they all match.

A similar technique can be used to compare streaming data in WSN. Instead of comparing character strings, we are comparing signal profiles. Using the same techniques, the intensity of the streaming data is compared to the intensity of the signal profile. With signal profiles, consistently comparing the end of the signal may not be an efficient approach. Consider the signal profile with a single peak. The beginning and the end of the signal profile are more likely to occur than the peak value. Therefore searching according to a value close to the peak will trigger less processing than either end of the signal. A statistical skip value should also be considered. Statistical analysis can be done on the likelihood of signal changes at a particular value. Given this information, an informed skip value can be calculated while comparing the signal profile.

In the string searching the match is absolute, however due to noise, this is not true for data collected in WSN. We assume that each signal profile is completely specified. One possible method to deal with noisy signals is to add every k^{th} point together. If the value exceeds a set threshold, then a more in-depth search can be done in the area to see if the signal profile can be found.

4. DISTRIBUTED SEARCH OF CORRELATED STREAMS

Using the Publish-Subscribe model, users can specify events in the WSN which are of interest to them via the Internet. It is necessary for data aggregation to occur in the network to determine, all conditions of the event are satisfied before informing the user. Recall the example used in Section 2. With the subscription $(A, 0.8, \geq)$, $(B, 0.6, \geq)$, $(C, 0.4, \geq)$ and the event $(A, 0.87)$, $(B, 0.91)$, $(C, 0.41)$.

In order to detect this event in the WSN the data must be aggregated from all three sensor to a single sensor to do the processing and analysis. The single sensor doing the analysis may be one of the three sensors, A, B, or C, or may be a separate sensor altogether. Because of the energy

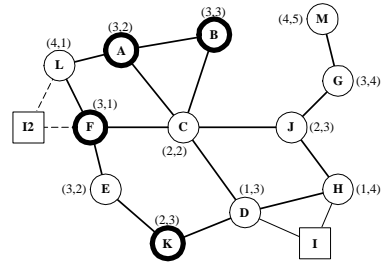


Fig. 3. Sensor Network: Communication links.

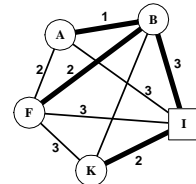


Fig. 4. MST for selected node communication

constraints, the aggregation of data should be done using the smallest amount of communication necessary. One way to accomplish this is for the sensor with the constraint that is least likely to happen, in the example this would be sensor A, waits until its event is satisfied, and then it send its data to the sensor with the next least likely event, and so on. In this situation B will only pass on its data to C if it detects a value > 0.6 and it has A's information.

Another approach takes into account the distance the sensor is from the Internet connection and therefore the user. Information can be passed from sensors farther away from the Internet connection to sensors which reside closer. In the event that all subscription information is satisfied, the node can easily pass the it on to the user. These two approaches are simply applied when the example is small, but what is a more interesting problem is how to handle a large number of different subscriptions in a large WSN. Each sensor must know who will aggregate the data and who to pass its data to. The problem can be defined formally as follows:

Given: N sensor nodes, k subscriptions.

Find: Location to aggregate data such that communication is minimized.

We solve this problem using Simulated-Annealing based search[1].

5. DISTRIBUTED DATA COLLECTION

When retrieving information from the sensor network, two types of situations can occur. The first is the case that we wish to receive information from all the sensors in the network. The second case involves only a selected set of nodes in the network communicating with the user. In both cases, the user may want to minimize power consumption or there may be bandwidth issues in the network that need to be considered.

In the first case, the user requests information from every sensor in the network. For this situation, we propose the use of Breadth-first search to determine the shortest path

back to the Internet, and the user. Consider the sensor network shown in Figure 3, excluding the Internet connection I2. For this case, we apply Breadth-first search from the Internet connection I, calculating the shortest-hop path to I. Each sensor n hops from the user, will then pass its information back to the user via its sensor neighbor with $n-1$ hops. The shortest-hop count for this situation is shown in the first values of the pair at each node in Figure 3. If we have multiple Internet connections in the network, Breadth-first search can be calculated from each connection. Communication is then done between each sensor and the Internet connection with the shortest path. Figure 3 shows the shortest-hop calculations for I2 in the second value of the pair. For example, sensor M would communicate with I, due to the shorter path. However, sensor C could communicate with either I or I2 without any extra cost.

We assume that the communication cost from sensor a to sensor b and from b to a are of equal value. This assumption is reasonable in the sense that the distance in either direction is the same, and therefore the transmit power need to communicate will be the same. In this situation Dijkstra’s algorithm can be used to calculate the Minimal Spanning Tree from the Internet connection to each sensor. Again, in the case of multiple Internet connections, the Minimal Spanning Tree for each connection can be calculated, and each sensor will send information to the closest connection.

When information is to only be collected from a subset of the sensors in the network, the minimal spanning tree should be calculated whether minimizing the number of sensors used or minimizing power consumption. When minimizing the number of sensors used to communicate a separate graph containing only the selected sensors and the Internet connection is created. The graph is fully connected, and the weight of each edge is determined by the shortest-hop path between the nodes. Consider the bold sensors, A, B, F, K and connection I in Figure 3. We create the graph shown in Figure 4 to calculate the minimal spanning tree using Kruskal’s or Prim’s algorithm. Once the minimal spanning tree has been calculated as shown by the bold links in Figure 4, the sensors used as hops on the chosen connections must be included. In the example, sensors C and D must be added to the minimal spanning tree to complete the tree in the sensor network.

6. EXPERIMENTAL RESULTS

We tested our approach for stream search as well as our data aggregation approach. We used measurements taken from vibrations sensors to test the stream searching approach. The results are shown in Table 1(a). We selected three patterns and two data streams for testing purposes. For each pattern and stream we found the runtime improvement over the brute-force Boyer-Moore approach. Note that 41% of this improvement is due to the statistical modifications of the Boyer-Moore approach.

Pattern	Stream	Runtime	Subs.	Topo.	Power
1	1	17.2	1	1	4.01
1	2	18.9	1	2	6.22
2	1	31.4	2	1	5.21
2	2	27.3	2	2	6.25
3	1	28.9	3	1	4.72
3	2	33.2	3	2	7.08

(a)

(b)

Table 1. (a) Speed-up using statistical Boyer-Moore algorithm for stream search. (b) Power consumption reduction using aggregation. For explanation see Section 6.

For the aggregation approach, we created 3 subscriber models each containing 10 sensors. We also created two different sensor network topologies, one with 10 and the other with 20 randomly placed sensors. We compared the amount of power saved in the network compared to the naive aggregation approach. The results, in Table 1(b), show that the approach had an improvement of 4-7 times over the naive approach.

7. CONCLUSION

We studied search problems in wireless sensor networks. We outlined the main issues and identified the key tasks. We have developed an information architecture for sensor networks that enables the development of many search applications. We addressed three specific search related problems: efficient stream search, distributed search of correlated streams, and distributed data collection. The experimental result demonstrate that the developed algorithms are efficient and can result in very significant power reductions and speed improvements.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. ANI-0085773. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).

8. REFERENCES

- [1] E. Aarts, J. Korst. "Simulated Annealing and Boltzmann Machines: a stochastic approach to combinatorial optimization and neural computing." New York: Wiley, 1989.
- [2] S. Brin and L. Page. "The anatomy of a large-scale hypertextual Web search engine". International World Wide Web Conference, pp. 107-117, 1998.
- [3] D. Estrin, R. Govindan, and J. Heidemann. "Embedding the Internet: introduction.". Communications of the ACM, Volume 43, No. 5, pp.38-42, 2000.
- [4] J. Kleinberg. "Authoritative sources in a hyperlinked environment." ACM Symposium on Discrete Algorithms, pp.668-677, 1998.
- [5] A. Tanenbaum. *Distributed operating systems*. Englewood Cliffs, N.J. : Prentice Hall, 1995.