Censor Networks: A Critique of "Sensor Networks" from a Systems Perspective

Bhaskaran Raman
Department of Computer Science and Engineering
Indian Institute of Technology Bombay
Mumbai, India 400076
br@cse.iitb.ac.in

Kameswari Chebrolu
Department of Computer Science and Engineering
Indian Institute of Technology Bombay
Mumbai, India 400076
chebrolu@cse.iitb.ac.in

ABSTRACT

This writeup presents a critique of the field of "Wireless Sensor Networks (WSNs)". Literature in this domain falls into two main, distinct categories: (1) algorithms or protocols, and (2) application-centric system design. A striking observation is that references across these two categories are minimal, and superficial at best. We argue that this is not accidental, and is the result of three main flaws in the former category of work. Going forward, an application-driven, bottom-up approach is required for meaningful articulation and subsequent solution of any networking issues in WSNs.

Categories and Subject Descriptors: C.2.1 [Network Architecture and Design]: Wireless communication

General Terms: Design

Keywords: Wireless sensor networks, application specific system design

1. INTRODUCTION

The original vision of "Wireless Sensor Networks (WSNs)" was articulated by researchers about a decade back [1, 2, 3]. Since then, several hundreds of research papers have been written on various algorithms and protocols specific to WSNs. Over a dozen or so WSN applications have been prototyped and several of these deployed too. Most networking conferences have this field included in their call-for-papers: this domain is a *hot topic* today.

We classify literature in this field in two main classes: (1) algorithms or protocols, and (2) application-centric system design. The former consists of work which proposes protocols for WSNs at various layers of the network (MAC, routing, transport) as well as other functionalities such as time synchronization, localization, security, data-aggregation, transmit power based topology control, etc. The second class of work has designed, prototyped, and in many cases also deployed specific applications of sensor networks. These include *habitat monitoring* [4, 5], *Redwood tree* micro-climate study [6], *industrial motor monitoring* [7], *structural monitoring* [8, 9], *bridge monitoring* [10, 11], *volcano monitoring* [12], *forest fire prediction* [13], *vehicle tracking* [14], *zebra tracking* [15], *sheep tracking* [16], *land-slide prediction* [17] etc.

The application-centric work is far less voluminous as compared to the protocol work. This is understandable since application prototyping requires more time and money, as well as collaboration with experts from various other fields (biology, structural engineering, seismology, etc. depending on the application). What is striking however, is the lack of in-depth references or citations across these two categories of work. We claim that this is not accidental: most protocol work in this domain have three basic methodological flaws. (1) The application-scenario description in protocol work lacks depth, with an implicit and incorrect assumption of generality. (2) A rich set of simplifying design choices are implicitly ignored, without adequate consideration. (3) The choice of parameter set in any evaluation of protocols lacks justification.

Furthermore, the application-centric work have shown that simple protocols work in practice. Thus there is no articulation in the literature today, of why the more complex approaches in the protocol work are even necessary. Going forward, such an articulation is possible only if the application-centric work drives research in this domain.

Methodology: The critique in this writeup is based upon reviews of papers in the domain of WSNs in various conferences and journals: ACM SenSys, ACM MobiCom, EWSN, ACM MobiSys, ACM SIGCOMM, IEEE INFOCOM. These are some of the leading venues of networking research, in our (subjective) opinion as well as that of colleagues we interact with. While we cite specific application-centric papers, we do not cite specific protocol papers, since they are far too voluminous. Clearly, given the volume of the literature, we cannot claim to have considered all of it. However, our critique consisting of the three flaws applies more or less *uniformly* to the protocol literature we have read. And since we have considered some of the leading publication venues, we feel confident with respect to our critique.

2. CRITIQUE OF "SENSOR NETWORKS"

We now present our three main critiques of WSN protocol work.

2.1 Lack of depth in application description

WSN is an applied research domain, and the value of abstract ideas are to be measured in terms of their applicability in realistic settings. In other words, while coming up with a protocol, it is important to state a specific application scenario, or a set of applications, in which it is useful. So our first and foremost critique is that in most WSN protocol literature, there is no mention, or only a passing mention of the application scenario in which the idea or protocol is applicable. Phrases such as "military and civilian applications" or "surveillance and monitoring" are used as catch-all in several papers. The implicit assumption here is that the design of a given protocol or algorithm is somehow independent of the nature of the WSN application, and that there is little or no variability across these applications. Such an assumption is clearly false; WSN applications vary widely on several counts, as described below.

What is the nature of the expected data traffic?

The choices in protocol design depend on factors such as how often to sense, how often to send the data to the sink, the quantity of data involved, and the data fidelity requirement. These vary widely depending on the application, as outlined below.

How often to sense? This can vary widely. For instance, in habitat monitoring [4], sensing is required about once in five minutes. In industrial monitoring [7], or bridge monitoring [10], sensing is required at a frequency of 50-100 Hz, but this needs to be done only once in a long while (several hours to several days). In volcano monitoring [12], continuous sensing is required during seismic activity.

How often to send the data to a sink? This too can vary widely. In deployments such as habitat monitoring and volcano monitoring, online data collection (every few minutes or hours) is useful. In ZebraNet [15] and BriMon [11], data is collected for analysis as and when a vehicle comes within range (of a Zebra or the bridge, respectively). In air or river pollution monitoring, it is imaginable that there is little use to collecting data very frequently.

What is the quantity of data? In habitat monitoring, just a few bytes of data are collected in several minutes. City pollution monitoring may involve just a few bytes in a day. On the other hand, the volcano monitoring application involves a few mega-bytes of data per volcanic activity.

What is the nature of data fidelity requirement? In environmental monitoring applications such as the redwood tree deployment or in pollution monitoring, it is imaginable that it is alright to lose a few samples. But applications such as bridge monitoring or volcano monitoring require all data samples for structural/seismic analysis.

Nature of topology

How many nodes? This too has a strong dependence on the application. Deployments such as volcano monitoring, industrial monitoring, habitat monitoring involve about few tens of nodes. Monitoring of a large bridge may involve a few hundred sensors. Clearly, protocol design for a few tens of nodes need not be as complex as that for a network with hundreds or thousands of nodes (as envisioned in most WSN protocol literature).

Over what region is the deployment? The deployment environment can be widely varying: indoors (e.g. industrial monitoring), outdoor with different levels of foliage (e.g. Redwood tree deployment), many line-of-sight links (e.g. volcano monitoring), etc. Such details impact network behaviour right from the link layer, and hence affect protocol design.

What to sense?

Mode of deployment: What is the physical phenomenon which is to be sensed, using what type of sensor? This determines an important detail: the mode of deployment. In the habitat monitoring deployment [4], the sensor motes are placed in birds' burrows. In the Redwood tree study [6], the motes are placed approximately 2 m apart from one another along the length of a tall tree. In volcano monitoring [12], the accelerometers are embedded into the ground, and in SenSlide [17], the strain gauges need to be inserted into rocks using a small drill. In industrial monitoring [7], the sensors are to be attached to induction motors. In fact, if we pay attention to the application-centric work, the assumption (sometimes implicit) of "sensors air-dropped randomly in a region" made in most WSN protocol literature appears rather wishful. The mode of deployment is important since it determines what design choices are available in protocol design.

Sensor power consumption: The other important factor determined by the sensor is the power consumption of the sensor, and as a corollary, how it compares with that of the radio. The sensor power consumption can be widely varying. For instance, a humidity sensor can consume a few orders of magnitude more power than a thermistor [4]. The power consumption of accelerometer sensors can vary widely, depending on the frequency of measurement and the mass of any oscillating unit. Similar comments hold true for other sensors such as pollution sensors too. Sensor power consumption is important information, since it determines how useful fine-grained protocol optimizations are, relative to overall system power consumption.

Other application-dependent details

There are many other application dependent details and requirements too, which demand a scenario description before considering any protocol issue. Is time synchronization required? At what granularity? Some applications may require time synchronization of a few ms (e.g. bridge monitoring), while others (e.g environment monitoring) may be able to tolerate a few seconds or more error, and still others may require no synchronization (e.g. industrial motor monitoring).

Is cost a constraint in the application? To what extent? Is location information required? If so, at what granularity? Is security required? Does data aggregation make sense in the application? Answers to these clearly depend on the application. For instance, few of the deployed applications we have cited (which form a large fraction of the concrete WSN applications thus far) have mentioned the need for automated location information, security, or data aggregation. Whether or not such functionalities are needed become apparent only on describing an application scenario in reasonable detail. Such a description is required before protocols are designed to provide these functionalities.

Summary of first critique

In sum, our first critique is that although WSN protocol design has a deep dependence on the application scenario, and although application scenarios vary widely, most protocol literature does not cite or use any specific application in its design.

2.2 Narrow design choices

There are a rich set of *simplifying* design choices available for WSNs. Most of these are overlooked in WSN protocol literature, whereas many of them have been used as an obvious choice in the application-centric work. The design choices listed below have varying degrees of dependence on the application characteristics (which adds to the set of reasons why any protocol design should describe the intended application in reasonable depth).

Is global coordination necessary?

A very significant design choice is the question of whether sensors can operate independently or if global coordination is required. That is, do all the sensor nodes need to form a single large network?

Not all applications require global coordination! For instance, in a WSN deployed for pollution monitoring, each node may just collect data independently and store it, to be retrieved at leisure later. Similarly, in ZebraNet [15], each node operated more or less independently to collect and store the data. In BriMon too, there is no need to have one large network. In a bridge, only data within a span is correlated, and data across spans is not [11]. So we only need to have several small networks, rather than one large network.

In abstract, sensor nodes may need to form one single network for one of two possible reasons: (a) time synchronization across all nodes is needed to interpret the data, or (b) to relay the collected data to a single point. But in practice, applications (e.g. bridge monitoring) require tight time synchronization only within small clusters. And the issue of relaying data can be addressed simply by having multiple independent gateways (as in habitat monitoring), or by using a mobile vehicle to collect the data (as in ZebraNet or BriMon).

If the application requirements are such that only small clusters of sensors need to communicate with one another, or if individual nodes can simply collect data for retrieval at leisure, we have essentially side-stepped the issue of *scalable* operation. It is worth noting that most WSN protocol literature considers "hundreds or thousands of densely deployed sensors forming [one large] network" as the main design challenge!

Centralized protocols/algorithms

The next significant simplifying design choice is that of centralized design, as opposed to using a distributed approach. Traditionally, centralized design is looked down upon for two main reasons: (a) the lack of fault tolerance (single point of failure), and (b) lack of scalability. But in WSNs, neither of these reasons apply very strongly. Most WSN deployments anyway have a sink node, which is a single point of failure. So there are no additional fault-tolerance issues when centralizing any other protocol functionality.

Similarly, the issue of scaling too is not a big concern since typical deployments anyway have sink nodes with far greater CPU and memory capacity. It is very well imaginable that such sink nodes can handle centralized protocols in networks of a few hundred nodes, if not more. In any case, even if a centralized approach were to be rejected finally, it does need careful consideration before discarding.

High gain omni/sector/directional antennas

A natural design option in many application scenarios is the use external, high-gain antennas. Such antennas may be of large form factor, reasonably heavy (a few kg) and may require some careful deployment using appropriate fixtures. But all the same several deployments such as the habitat monitoring [4] and volcano monitoring [12] have used this option. Such use is feasible in cases where the form factor is not a hindrance, which is the case in many applications[1].

The use of high gain antennas increases the range significantly. This can greatly reduce the number of hops to/from the sink node. Intuitively, protocol design for a network which is just 5-6 hops is far easier than for one with 50-60 hops.

Multiple channels, multiple radios

A design option which has received some, but in our opinion, not adequate attention is that of using multiple radios per node, and multiple channels of operation. Miniature radios are quite inexpensive; for instance the 802.15.4 enabled CC2420 radio costs under U.S.$10. And the 802.15.4 standard has a total of 26 channels, and 16 channels just in the 2.4 GHz band.

Clever use of the available channels can easily side-step MAC related issues. For instance, the bridge monitoring [11], using different channels for the nodes in the different spans of a bridge means that we only have to deal with MAC issues for 6-12 node networks. This is of course far easier problem than that of dealing with MAC issues for 100-200 nodes operating in the same channel.

Other design options

A few other simplifying design options which have not received adequate attention are as follows.

Use of GPS at some nodes: The issue of time-synchronization (if needed by the application) can be alleviated significantly by the use of GPS-enabled nodes. This is a viable option for outdoor deployments, unless most nodes are expected to be covered by dense foliage. Even if we have a small fraction of the nodes GPS enabled, we would not have to deal with time synchronization beyond a few hops. And once we have time synchronization, MAC related issues can also be addressed more easily.

Alleviating power issues: Most WSN protocol literature is centered around reducing power consumption. For applications where form factor is not an issue (many outdoor deployments), just using a larger battery is a viable option. In indoor deployments, just using a power outlet, at least at a fraction of the nodes may be a viable option. For instance, for an application such as structural monitoring [9] where the sensors have to be fitted to the building anyway, access to power outlet may be viable. Similarly, for monitoring industrial motors, where the motors anyway run off power supply, using this for the sensor nodes too may be possible. Having a few nodes in the network which have much greater power availability significantly eases the issue of power-efficient protocol design.

Combination of design options

For a given application scenario, several of the above simplifying options will make sense. This further simplifies the problem space. For instance, suppose we divide up the collection of nodes into small independent clusters, each with just few tens of nodes. Each cluster has one node which is more powerful than the rest (e.g. a StarGate node), and each cluster is allocated a separate channel of operation. We can use centralized protocols for time synchronization, MAC, routing, transport, etc., with the central coordination being done by the StarGate node. We may even use external antennas to reduce the number of hops in each cluster, and GPS-enable the StarGate nodes if we want to correlate the data from different clusters.

With the unbridled use of all possible simplifying design options as above, it is rather difficult to imagine non-trivial networking problems. (This is substantiated further in Sec. 3).

Summary of second critique

Our second critique is thus: most WSN protocol literature does not consider the various simplifying design choices, which leaves the complexity and depth in the literature unjustified.

2.3 Unjustified parameter space

Our final critique of WSN protocol literature is with respect to an important aspect of protocol evaluation. Evaluation clearly depends on a host of parameters including: (1) the number of nodes, (2) the density of nodes, (3) the pattern of node deployment, (4) what kind of antennas are in use and what is the resultant expected number of hops in the network, (5) the traffic pattern, (6) the data generation rate, etc.

However, in most WSN protocol literature we have come across, the choice of parameter space for evaluation has *no* justification. More specifically, we are not aware of any work in WSN protocol literature which has taken parameter values from (a) any of the deployed applications, or (b) some explicitly specified futuristic extensions of these applications.

In a related vein, [18] has questioned the repeatability and rigour of simulation studies in MANETs (mobile ad-hoc networks). Our claim on WSN protocol evaluations raises a more fundamental issue: in a research domain which is about a decade old, and has several deployments, there is little reason to explore an unjustified parameter space. In retrospect, the lack of justification for evalua-

9 [1] It is worth noting that the use of high gain antennas does not affect the radio power consumption in any way; nor does it affect the symmetry of the links.

tion parameters is not surprising. Since most WSN protocol literature does not cite or describe in detail any specific application, there is no basis upon which they can offer justification for the evaluation parameters.

3. SIMPLE SOLUTIONS IN PRACTICE

We now observe that *simple* protocols have indeed worked in practice. In making this point, we have examined [5, 6, 7, 8, 10, 12, 13, 14, 15, 16], which represent a large fraction of the WSN application deployments so far.

With respect to the MAC, deployments such as Habitat monitoring [5, 6] have used low-power listening [19], which is one of the early and simple MAC protocols suggested for sensor networks. Most other deployments mentioned above do not quite mention the MAC protocol in use; they likely use simple CSMA/CA.

A look at the routing protocols used in practice is revealing. Habitat monitoring [5], the Redwood study [6], WISDEN [8], and Bridge monitoring [10] have used MintRoute [20]. Volcano monitoring [12] has used a variation of MintRoute, with a modified metric more suited to the CC2420 radio. The industrial monitoring deployment [7] reports to have used single-destination DSDV [21]. Both DSDV and MintRoute are among the early routing solutions for multi-hop wireless networks. Significantly, neither consider any power optimization related mechanisms, which form the crux of the majority of WSN routing literature.

Furthermore, some deployments have found it easy enough to design *simple* routing solutions from scratch: e.g. FireWxNet [13], vehicle tracking [14]. In a similar vein, WISDEN [8] and volcano monitoring [12] have described their simple custom transport protocol[2].

Coming to time-synchronization protocols, bridge monitoring [10] and volcano monitoring [12] have used FTSP [22]. Although FTSP itself is quite simple, several deployments have explicitly rejected even the drift-correction feature (of FTSP) as unnecessary complexity. WISDEN [8], FireWxNet [13], and the vehicle tracking [14] applications design their respective (even simpler) time-synchronization mechanisms[2].

The above applications have scarcely mentioned the need for functionalities such as localization, security, data aggregation, or transmit-power based topology control. ZebraNet [15] and electronic shepherd [16] are unique in that they make scant mention of any networking protocol related issue.

So if simple protocol solutions have sufficed in practice, the burden of motivating the need for complex solutions is even greater. Any WSN protocol proposed must justify its need with respect to these applications, or with respect to well-articulated futuristic extensions of these applications.

4. CONCLUSION

There are three critical unanswered questions in WSN protocol literature. (1) What are the specific applications which need to be deployed using a *single* large (hundreds or thousands of nodes) network, instead of just several independent small networks? (2) Why exactly do the various simplifying design choices we have listed not make the various networking or protocol issues trivial in practice? Many of the design choices have in fact been obvious choices in WSN deployments. (3) And finally, what is the basis for evaluating newer ideas or protocols, when the evaluation parameter space is not justified against specific applications?

Networking issues in the deployed applications have had quite simple solutions thus far. If there are indeed complex issues, they require in-depth articulation and study in the context of specific applications (and not in the abstract).

As WSN protocol literature stands today, the importance given to *networking issues* is disproportionately large. This we believe is due to (an unjustified) desire for generic solutions to begin with. In system design, a *bottom-up* approach is more fruitful, where we design protocols for one specific scenario after another; and after a few specific solutions, look for generality across the specifics.

Acknowledgment: We thank Srinivasan Keshav and Purushottam Kulkarni for their feedback and insight on earlier versions of this writeup.

9 [2] In fact, the protocol descriptions are less than half a page each.

5. REFERENCES

[1] Gregory J. Pottie. Wireless Sensor Networks. In *Information Theory Workshop*, Jun 1998.
[2] Deborah Estrin, Ramesh Govindan, John Heidemann, and Satish Kumar. Next Century Challenges: Scalable Coordination in Sensor Networks. In *MobiCom*, Aug 1999.
[3] Deborah Estrin, Lewis Girod, Greg Pottie, and Mani Srivastava. Instrumenting the World with Wireless Sensor Networks. In *ICASSP*, May 2001.
[4] Alan Mainwaring, Joseph Polastre, Robert Szewczyk, David Culler, and John Anderson. Wireless Sensor Networks for Habitat Monitoring. In *ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, 2002.
[5] Robert Szewczyk, Alan Mainwaring, Joseph Polastre, John Anderson, and David Culler. An Analysis of a Large Scale Habitat Monitoring Application. In *SenSys*, Nov 2004.
[6] Gilman Tolle, Joseph Polastre, Robert Szewczyk, David Culler, Neil Turner, Kevin Tu, Stephen Burgess, Todd Dawson, Phil Buonadonna, David Gay, and Wei Hong. A Macroscope in the Redwoods. In *SenSys*, Nov 2005.
[7] Lakshman Krishnamurthy, Robert Adler, Phil Buonadonna, Jasmeet Chhabra, Mick Flanigan, Nandakishore Kushalnagar, Lama Nachman, and Mark Yarvis. Design and Deployment of Industrial Sensor Networks: Experiences from a Semiconductor Plant and the North Sea. In *SenSys*, Nov 2005.
[8] Ning Xu, Sumit Rangwala, Krishna Kant Chintalapudi, Deepak Ganesan, Alan Broad, Ramesh Govindan, and Deborah Estrin. A Wireless Sensor Network For Structural Monitoring. In *SenSys*, Nov 2004.
[9] Jeongyeup Paek, Krishna Chintalapudi, John Cafferey, Ramesh Govindan, and Sami Masri. A Wireless Sensor Network for Structural Health Monitoring: Performance and Experience. In *EmNetS-II*, May 2005.
[10] Sukun Kim, Shamim Pakzad, David Culler, James Demmel, Gregory Fenves, Steven Glaser, and Martin Turon. Health Monitoring of Civil Infrastructures Using Wireless Sensor Networks. In *IPSN*, Apr 2007.
[11] Kameswari Chebrolu, Bhaskaran Raman, Nilesh Mishra, Phani Kumar Valiveti, and Raj Kumar. BriMon: A Sensor Network System for Railway Bridge Monitoring. In *MobiSys*, Jun 2008.
[12] Geoff Werner-Allen, Konrad Lorincz, Jeff Johnson, Jonathan Lees, and Matt Welsh. Fidelity and Yield in a Volcano Monitoring Sensor Network. In *OSDI*, 2006.
[13] Carl Hartung, Richard Han, Carl Seielstad, and Saxon Holbrook. FireWxNet: A MultiTiered Portable Wireless System for Monitoring Weather Conditions in Wildland Fire Environments. In *MobiSys*, Jun 2006.
[14] Tian He, Sudha Krishnamurthy, John A. Stankovic, Tarek Abdelzaher, Liqian Luo, Radu Stoleru, Ting Yan, and Lin Gu. Energy-Efficient Surveillance System Using Wireless Sensor Networks. In *MobiSys*, Jun 2004.
[15] Pei Zhang, Christopher M. Sadler, Stephen A. Lyon, and Margaret Martonosi. Hardware Design Experiences in ZebraNet. In *SenSys*, Nov 2004.
[16] Bjorn Thorstensen, Tore Syversen, Trond-Are Bjornvold, and Tron Walseth. Electronic Shepherd - A Low-Cost, Low-Bandwidth, Wireless Network System. In *MobiSys*, Jun 2004.
[17] A. Sheth, K. Tejaswi, P. Mehta, C. Parekh, R. Bansal, S. Merchant, T. Singh, Thekkath C. A, and K. Toyama. SenSlide - A Sensor Network Based Landslide Prediction System. In *SenSys*, 2005. Poster.
[18] Stuart Kurkowski, Tracy Camp, and Michael Colagrosso. MANET simulation studies: the incredibles. *ACM SIGMOBILE Mobile Computing and Communications Review*, 9(4):50–61, Oct 2005.
[19] Joseph Polastre, Jason Hill, and David Culler. Versatile Low Power Media Access for Wireless Sensor Networks. In *SenSys*, Nov 2004.
[20] Alec Woo, Terence Tong, and David Culler. Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks. In *SenSys*, Nov 2003.
[21] Charles E. Perkins and Pravin Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In *SIGCOMM*, Sep 1994.
[22] Miklos Maroti, Branislav Kusy, Gyula Simon, and Akos Ledeczi. The Flooding Time Synchronization Protocol. In *SenSys*, Nov 2004.