Exploring Semantic Interference in Heterogeneous Sensor Networks

Laura Marie Feeney
Swedish Institute of Computer Science
lmfeeney@sics.se

ABSTRACT

As the use of wireless sensor networks expands, there will be multiple, independent networks operating in the same physical space. These networks will run heterogeneous applications and be managed by different entities. At the same time, the development of commercially available, general purpose sensor platforms makes it likely that co-located networks will be based on the same platform. Such networks will unavoidably interact with each other due their common communication hardware. "Semantic interference" occurs when frames transmitted in one sensor network are successfully decoded and (mis-)interpreted in another. This paper examines modes of conflict, co-existence, and cooperation that obtain in this context.

Categories and Subject Descriptors

C.2 [**Computer-Communication Networks**]: Network Architecture and Design

General Terms

Design

Keywords

sensor network architecture, sensor network deployment, semantic interference

1. INTRODUCTION

It is widely expected that wireless sensor networks will be an important element of future home, work, and urban environments – as an opening sentence, it is a cliché. To the best of our knowledge, however, there has been relatively little discussion of some of the practical implications of multiple, independent wireless sensor networks operating in close proximity to each other.

This paper is intended to highlight and explore the implications of a sensor network deployment model where there is both a very high heterogeneity of applications and ownership and a relatively limited range of sensor platforms and system software. The case of co-located networks that use the same transceiver hardware and system software is particularly interesting, because of the possibility that frames transmitted by nodes in one application network are received and (mis-)interpreted by nodes that are part of a different application network. We call this conflict "semantic interference", to distinguish it from conventional RF interference. More generally, this paper explores various modes of conflict, co-existence, and cooperation between co-located sensor networks.

The contribution of the paper is organized as follows: First, we describe our model and assumptions and show how interactions between independent co-located sensor networks can result in semantic interference. We also compare these conflicts with more conventional attacks on a network. Second, we describe ways that networks can isolate themselves at various layers in the protocol stack or via cryptographic signatures, allowing for co-existence between networks. We also suggest some practical lessons learned for implementing and evaluating systems that can co-exist safely. Third, we argue that it is useful to do better than mere co-existence and speculate on three ways in which networks might effectively cooperate. We conclude by describing some related work and highlighting the prospect for interesting future work in developing architectures that support cooperative interactions between co-located sensor networks.

2. SEMANTIC INTERFERENCE

Sensor networks will be widely deployed for a variety of purposes and therefore multiple, independent sensor networks will operate in the same physical space, as in Figure 1. These networks will be heterogenous with respect to the application services they provide and with respect to the entities that configure and manage them.

Assumption: The common deployment case will involve independent, co-located sensor networks, providing heterogeneous application services.

As the market for sensor network products expands, we expect that many applications will be developed by vendors who act as VARs (value-added resellers) of one of a relatively small number of general purpose sensor networking platforms that use the same radio hardware or radios implementing the same standardized protocols. To obtain basic network functionality in a cost effective manner, vendors will also base their products on commercial or freely available

HeterSANET'08, May 26, 2008, Hong Kong SAR, China.

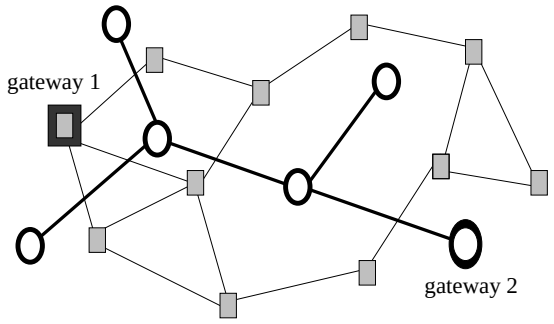

Figure 1: Two sensor networks deployed in the same location.

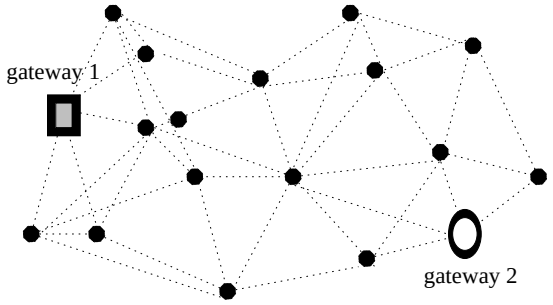

Figure 2: Two sensor networks deployed in the same location. If the two networks share a common underlying platform, frames may be intelligible in both networks.

software platforms, rather than reinventing the underlying operating system and network protocols. This approach allows vendors to focus on developing application specific sensing hardware and/or data processing software and on providing system management and other services. (Note that this assumption is not intended to imply that sensor networks are necessarily internally homogeneous, simply that general purpose platforms will frequently be used as a component of these networks.)

Assumption: Sensor networks will often be based on sensor platforms that use the same radio and system software.

It will be difficult to predict in advance what sensor networks will operate in a given physical space, especially because such networks may be highly mobile. For example, a future train station environment might include sensor networks for police security, for monitoring the track infrastructure, for ensuring food safety in restaurants in the station, for monitoring temperature and ventilation, and so forth. Passing through this environment are the sensor networks present on the trains themselves: including networks for monitoring the train equipment, for passenger wagons, and for monitoring cargo, as well as any personal networks carried by people moving about the station.

Assumption: No single entity will have knowledge or control over all of the networks operating in a location. Due to network mobility, the set of co-located networks may be highly dynamic.

We have assumed both the widespread deployment of sensor networks and a business model where at least some vendors base their products on one of a fairly small number of general purpose sensor platforms. It is therefore likely that there will be situations where independent sensor networks based on the same transceiver hardware will be co-located.

As a result, frames transmitted in one network may be successfully decoded by nodes in another network, creating an integrated network as in figure 2. Unless receivers have some way to identify and filter frames that are intended for another network, there is a risk that they will be incorrectly treated as legitimate data in the network. We refer to this interaction as "semantic interference", to distinguish it from RF interference.

In some cases, the foreign data frame will be diagnosed as ill-formed when it is processed by a higher layer protocol. For example, the frame length may be incorrect, or the bits that are (mis-)interpreted as a message-type may be invalid. However, the receiver will have spent processing and battery resources, only to eventually discard the frame.

In other cases, the frame may be mis-interpreted in a way that suggests that there is a potential problem within the network. For example, the bits that are (mis-)interpreted as a sensor data value may suggest that a sensor is obtaining inconsistent measurements. The network may spend significant resources attempting to cope with an apparently dysfunctional sensor (e.g. by initiating re-calibration).

Finally, because frame formats used in sensor networks are generally very simple, there is a non-trivial risk that the foreign frame may appear to plausible to the receiver. Misinterpreting foreign data frames this way is a significant risk because the sensor network will treat foreign frames as carrying legitimate sensor information. If high layer protocols and mechanisms such as data aggregation are used the source of the problem may be difficult to detect. If the network also includes some form of actuator that responds to sensor observations, the consequences could be serious. This case is described in more detail in later sections.

Assumption: Frames transmitted in one sensor network may be successfully decoded by nodes in another network. Semantic interference *occurs when such frames are misinterpreted.*

Sensor networks are often deployed in public environments, where they are vulnerable to a variety of attacks because it is difficult or impossible to prevent physical access to devices in the network. There is however an expectation that the attacker faces some limitation with respect to physical access or communication and energy resources. For this reason, many security mechanisms for sensor networks are intended to limit the impact that an attacker has on overall network functionality. Such mechanisms are also intended to protect the network in the case where some subset of the nodes or keys have been compromised and to ensure that as much valid information as possible is obtained.

This work focuses on the effects of interactions between legitimate networks that are operating as designed and do not intentionally disrupt the operation of other networks. As a result, the situation is a little different, because the two networks may have the same coverage area (as in Figure 1), while the larger network may also have much greater total energy capacity. There is naturally substantial overlap between these problems.

Assumption: We do not assume that disruptive interactions are the result of an intentional attack on a network.

The problem of semantic interference must be addressed within the constraints of limited resources available in sensor devices. There has recently been some movement toward more capable devices such as iMote2 [5] and Sun SPOT [16] and toward hierarchical sensor networks which include both more and less capable devices. Nevertheless, sensor platforms will generally continue to have fairly limited processing, memory, communication, and energy capacity.

These limitations are especially important when several networks share the same wireless channel. In this case, most of the frames a node receives will be identified as foreign, so the cost of filtering and discarding this traffic will be significant.

Another unexpected interaction that can occur is in power saving mechanisms that are based on wakeup signaling. With wakeup radio [2], the primary radio sleeps and a simple very low-power receiver listens on a separate channel for a busy-tone. With preamble sampling[3], the primary radio periodically listens to the channel to determine whether it is being paged via an extended preamble transmission. In either case, a receiver will always have to check whether it is the target of a given wakeup signal. If multiple networks use the same wakeup signaling mechanism, the number of signals that each node will have to check increases significantly. In the worst case, the wakeup signals may be incompatible in such a way that receiver spends much time awake.

Assumption: Sensor networks will continue to include devices that have limited memory, processing, communication and energy capacities.

From these assumptions, we conclude that semantic interference can occur when independent networks based on common radio hardware are deployed in the same geographical area. Semantic interference leads to increased resource consumption, as receivers attempt to process foreign frames. There is also potential for incorrect operation of the network.

3. EXAMPLE

To make these ideas more concrete, consider the example of an office building of the near future. The building manager has purchased a "building security solution" from one vendor and a "green building (HVAC) solution" from another vendor. The former includes door/window sensors to track people in the building, while the latter uses motion and temperature sensors to optimize the use of heating, cooling, and lighting. Although the building manager "owns" both networks, the systems are configured and managed by their respective vendors. The vendors do not develop most of the hardware and software themselves; they purchase an existing platform and provide application specific sensor/actuator interfaces, application specific software, and management and maintenance services.

In our example, both vendors' solutions happen to be based on the same sensor platform. Obviously, we cannot know what platforms will be commercially popular in the future. In order to have a specific example, we assume that both vendors use the Contiki operating system [7] running on the ESB platform [14].

Because both networks use the same transceiver (the RFM TR1001 in our example) and frame format, packets transmitted in one network are successfully decoded by nodes in the other. In figure 3, we assume that both vendors use the X-MAC [3] low power MAC layer provided in the current Contiki release. The MAC layer header contains short sender and receiver addresses, but these are not necessarily unique across large numbers of nodes, nor do they necessarily allow receivers to distinguish traffic from foreign networks.

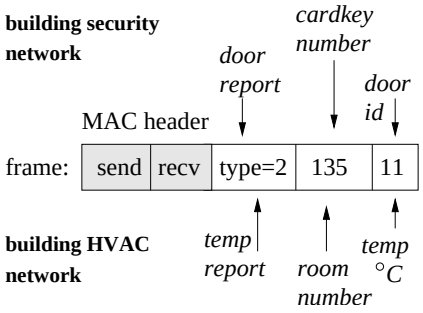

Figure 3: How is this frame interpreted?

If the two sensor networks both use a simple data frame format such as Figure 3, a receiver may not be able to determine in which network a frame was generated. As a result, frames will be successfully decoded and (mis-)interpreted by nodes in both the security sensor network and the building HVAC sensor network. In the worst case, every time the system administrator (badge number 135) opens the door to the machine room (door number 11), the HVAC system turns up the heating in room 135 because the temperature is reported to be only 11°C.

Clearly, this example is artificial, at least with respect to the risk of overheating a room in an office building. It is far more likely that misinterpreting foreign frames in a network will simply result in problems such as unexpected frame or buffer sizes and nonsense data fields. Even in the best case, processing such frames consumes energy resources or the sensor software may fail, if it is not sufficiently resilient to unexpected input data. Nevertheless, this risk of semantic interference resulting in incorrect operation of sensor-actor network highlights the importance of applying mechanisms for avoiding it.

4. CO-EXISTENCE VIA ISOLATION

One way to avoid semantic interference and allow heterogenous networks to co-exist without operational conflict is to ensure that they are isolated from each other, such that foreign frames are not received or are quickly identified as foreign and discarded without being further interpreted.

In this section, we present isolation mechanisms that can be applied at different layers of the protocol stack. Isolation at the PHY layer is always effective, but it is not possible to ensure that enough independent channels are available. Isolation at the MAC (and higher layers) is effective only in the case where a receiver can determine that is has correctly identified the MAC (or higher layer) header. Only cryptographic signatures can ensure, with high probability, that a receiver can determine whether a frame was generated by a sender who has used a common key, which acts as a network identifiers in this context.

4.1 Isolating networks at the PHY layer

In this context, we say that networks are isolated at the physical layer if frames transmitted in one network cannot be successfully decoded by devices in a co-located network.

Transceivers operating in ISM bands may use a wide variety of channel allocation, modulation and media access schemes. If transceivers in two networks transmit on interfering channels or use incompatible modulation and coding schemes while transmitting on the same channel, the result will be RF interference. Receivers cannot decode the interfering transmitter's signal and cannot misinterpret its payload. The RF interference issue, though important, is generally out of scope of this work. Regulatory agencies such as the FCC and ETSI define rules for transmitters communicating in the ISM bands, which are intended to minimize such conflict.

If transceivers using the same channel also use the same modulation and coding scheme, either because they use the same radio hardware or because they follow the same standard, frames will be intelligible both networks. In this case, nodes must transmit on different channels to isolate the two networks at the PHY layer.

Given our assumptions about the lack of centralized control, or even knowledge, of all co-located sensor networks, manually configuring and maintaining a non-overlapping channel allocation is not feasible. Dynamic channel selection and adaptation is feasible for some radio technologies and is widely studied as a technique for mitigating RF interference and channel contention. Dynamic channel selection is much more difficult, and may become impractical, if the set of networks operating in the same area is dynamic. Moreover, it is unlikely that there is sufficient channel capacity to ensure that all co-located networks can be fully isolated, particularly in crowded bands like the 2.4GHz ISM band. PHY layer isolation may also be infeasible in the case of simple single-channel transceivers.

4.2 Isolating networks at the MAC layer

Networks are isolated at the MAC layer if frames are mutually intelligible between the two networks, but there is some mechanism for identifying and discarding foreign frames based on information in the MAC header. Naturally, isolation at the MAC layer is only possible if a receiver are able to reliably determine whether the sender is using the same MAC protocol.

The popular tMote Sky, iMote2, and SunSPOT platforms are based on a packetizing radio that implements the standardized IEEE 802.15.4 PHY and MAC, using the Chipcon CC2420 transceiver. The IEEE 802.15.4 MAC header includes a 16-bit network ID, which can be used to isolate networks by allowing receivers to identify and discard foreign frames. Assuming that the network ID's are randomly generated and distributed to all nodes intended to participate in a common network, isolation can be achieved with some probability. However, the effectiveness of this mechanism is limited in large, long-lived, dynamic environments – like the train station scenario mentioned earlier – where a very large number of networks will come into contact.

Although IEEE 802.15.4 is a popular communication layer for wireless sensor networks, other commercially available platforms, such as Scatterweb are based on simple, inexpensive byte-level transceivers, such as the RFM TR1001 transceiver. Some proprietary and semi-custom platforms are also based on such transceivers. Using byte-level transceivers means that the processor has precise control over the transmission, which may be useful for specialized power saving or transmission scheduling mechanisms, for example. These platforms can easily support a wide variety of non-standardized and specialized MAC protocols[9]. Many of these MACs have extremely minimal headers and are therefore very susceptible to semantic interference.

4.3 Isolating networks at higher layers

Under the assumption that all transmissions in the network can be reliably identified as belonging to a common communication framework, the isolation can occur at the higher layers.

For example, TinyOS [10] messages carry a group ID, which can be used to distinguish logical networks. This mechanism has the same limitations and risks that are associated with the IEEE 802.15.4 network ID, namely the lack of any central authority for assigning ID's to all networks operating in a given physical location and the potential risk of collision in dynamic environments where many networks are present.

The TinyOS networking abstractions also support multiple network protocols, managed via a Network Service Manager. The isolation achieved by this mechanism requires not only that all interacting networks are TinyOS based, but also that all networks present in a location are using a common TinyOS revision with a non-conflicting set of protocol ID's. This may be impractical to achieve given both an open software environment, where developers from many different organizations can modify and extend the software, and the shared access to wireless media.

The Internet, of course, supports the assumption that all traffic is based on IP and follows IANA assigned numbers as protocol identifiers. Thus, traffic associated with many different application endpoints is forwarded through the network. In principle, routers forward packets as opaque data, based on the destination IP address. The application endpoint at the destination is specified via the TCP or UDP port number; a portion of the port number space is devoted to registered ports to provide applications with predetermined rendezvous points and the rest to dynamic/private ports.

It has been shown that TCP/IP is an effective transport protocol, even for very limited sensor platforms[6], while the 6LoWPAN IETF WG has specified an 802.15.4-specific IPv6 [11] to support both mesh forwarding and header compression. As with isolation at the MAC layer, if a receiver can be sure that it is parsing a valid TCP/IP packet, it can use this information for isolation.

Because TCP (and UDP) registered port numbers are unique and vendor/application specific, a receiver can easily determine whether it will be able to correctly process a frame. The ability to identify application endpoints is useful because, unlike the Internet, many sensor networks are based on in-network data aggregation. If a node does not have a listener on the specified port, the frame may be assumed to be associated with a foreign sensor network and discarded.

Isolating networks at higher layers has the significant advantage of allowing a node to potentially support multiple network endpoints. However, administrative issues associated with centrally assigned identifiers are exacerbated in

decentralized wireless sensor networks. Even in the Internet context, although several thousand registered port numbers are currently unassigned, it may be difficult to ensure that all vendors use proper IANA-assigned port numbers.

4.4 Isolating networks cryptographically

The open nature of sensor networks makes them vulnerable to a variety of attacks, including injecting false sensor data, preventing valid sensor data from reaching a gateway, and draining nodes' batteries by forcing nodes to communicate unnecessarily. An attacker's ability to physically capture nodes also means that keying material may be compromised. Protection from such attacks is necessary for practical deployments and security mechanisms that take into account the resource limitations of sensor networks have been widely studied [12].

A network can isolate itself cryptographically, by applying a signature or message integrity code (MIC) to all frames transmitted in the network. The MIC allows a receiver to confirm that that the sender has authenticated the message with a known shared key. If a receiver determines that a frame was not authenticated with the correct key, the frame is assumed to be associated with a foreign network.

Cryptographic isolation requires processing, memory, bandwidth and energy resources to compute and append the MIC to each frame. This overhead may be significant in smaller devices [13], but it should be pointed out that here the goal is to prevent accidental collisions, rather than to provide strong protection against deliberate tampering. Therefore it may be reasonable to use less expensive checksum operations. The goal is simply to ensure that the probability that the bits that a receiver interprets as the MIC happens to randomly correspond to the checksum computed with the shared key is extremely low.

The IEEE 802.15.4 specification requires that devices support AES-based CCM* encryption and authentication, including the CBC-MAC. This support makes it relatively straightforward to provide cryptographic isolation in IEEE 802.15.4 based networks.

Despite the costs, only cryptographic isolation is likely to be highly effective in situations where a large number of networks may interact and there is a significant risk of semantic interference. For MAC layer and higher layer isolation, the receiver must be able to determine whether the received frame has a known format, e.g. a TCP/IP frame. By contrast, cryptographic isolation is effective regardless of the frame format.

5. LESSONS LEARNED

In this section, we summarize four practical lessons that can be learned from our discussion of semantic interference and ways of avoiding it.

Tradeoff between isolation and overhead Many current sensor network applications, especially those that do not use IEEE 802.15.4 platforms, are based on extremely minimal (or absent) L2/L3 headers and transport essentially bare sensor data. This approach substantially reduces memory and energy consumption, but does not provide protection from semantic interference. Performance results based on such minimalist protocols must be considered carefully in the context of the need for such protection.

There is a tradeoff between the cost of resources and complexity required to support more informative headers and cryptographic isolation and the safety benefits of avoiding problems of semantic interference. However, the previous discussion has made the importance of such protection clear.

Defensive programming In order to minimize code size, the application code running in small sensors is often highly simplified, eliminating the code used for sanity checking input data, for example. Particularly in the absence of extremely strong isolation mechanisms, sensor network applications need to be programmed defensively in the same way that conventional programs are. This means that programs may need to effectively protect themselves from mal-formed frames and implausible payload data, despite the memory and computational cost of the additional code.

Detecting foreign traffic If networks are cryptographically isolated, then detecting a significant number of non-authenticated frames is not necessarily indicative of an attack. It may be the normal case that traffic from two or three or more co-located networks may be discarded this way; validated frames may be in the minority. It may be relatively difficult to distinguish an 'attack' from normal case.

Dimensioning sensor networks Dimensioning resources for sensor networks is complex because of the impact of foreign traffic. The cost of filtering and discarding foreign traffic may be non-trivial, especially in the case of cryptographically isolated networks. In addition, power saving schemes that rely on some form of wakeup-radio [2] or preamble sampling [3] may be affected by wakeup signals from the foreign network.

More generally, energy and communication budget calculations that only take into account the activity of its own network (e.g. "each sensor will receive n and transmit m frames per minute) are too simplistic.

6. COOPERATIVE NETWORKS

In the previous sections, we have described ways that heterogeneous sensor networks can safely co-exist by ensuring that they are isolated from each other, particularly via cryptographic isolation. In this section, we sketch a case for developing architectures that allow such networks to cooperate. Such architectures do not preclude the use of network isolation mechanisms such as cryptographic isolation: It will still be necessary to filter out frames that are not compatible with the implemented architecture. In many cases, it will also be important to distinguish among frames belonging to different, but compatible networks.

There are two possible forms of cooperation between networks: the first is cooperative data forwarding, where nodes cooperate to provide better forwarding coverage but do not share sensor data; the second provides cooperative sensing and manipulating sensor data itself. The discussion below focuses on the former case, which seems more feasible in an open deployment environment.

The argument in favor of supporting shared forwarding capability between networks is that it will allow for more efficient use of available resources. Sharing forwarding elements among networks may also reduce communication cost,

by allowing greater path diversity and reducing the problem of high energy consumption hotspots near gateways. Although most sensor networks are intended to have quite low duty cycles and traffic levels, many such networks are intended to work in already crowded ISM bands, particularly the 2.4GHz band, while others are based on simple single-channel transceivers. Moreover, many sensor network applications are intended to detect and report unusual events, so peak traffic during such events may be high. In emergency management networks, shared forwarding capability may also help to prioritize traffic.

In this section we speculate briefly about three possible models for cooperation between independent networks, though certainly there are many more to be invented:

Internet model The earlier discussion of isolating networks presented the use TCP/IP registered port numbers as a way to determine whether a frame was associated with a foreign network. We also suggested that application- or protocol-specific registered port numbers were an effective way of allowing multiple application endpoints to co-exist on a single node. Among those endpoints might be standardized instances of well-known sensor data management protocols. Such endpoints would allow the network to cooperate in sharing forwarding and communication capabilities, while avoiding much harder problem of sharing sensor data. In this context, cryptographic isolation might be used to identify foreign traffic that is nevertheless permitted to use a node's resources, possibly on a quid pro quo basis, as has been proposed for ensuring cooperation in ad hoc networks, e.g. [4].

The Internet model is inherently interesting here not only for its ability to support co-existence between protocols, but also its effective standardization processes.

Shared runtime environment Another way to ensure cooperation among networks is to require that all cooperating networks are based on a common runtime environment. In this case, all sensor protocols would be implemented in terms of basic functionality provided by the runtime environment, allowing co-existence between independently implemented protocols and applications, allowing cooperative operation of multiple networks. The runtime environment would be responsible for ensuring that only valid network operations were permitted to run in the combined network infrastructure (e.g. via per-application keys) and that traffic associated with different logical networks was managed appropriately.

One disadvantage of such an approach is that both consumers and application developers would be locked into a single runtime environment, in which the application functionality would be limited to that which could implemented from functionality supported by the underlying environment. The overhead of such a runtime environment might also be significant.

In an extreme case of the earlier example of future building management, some or all of the wireless sensors available in a given location could be shared infrastructure, which would reduce installation and maintenance costs. To extend the earlier example, the building manager would install some sensor network infrastructure in the building. She would then purchase, from various vendors, applications that were compatible with the sensor platform that was used in the building's infrastructure. Each application would be installed and explicitly authorized to operate in the runtime environment.

Virtualization Virtualization is frequently used to support divergent functionality in shared hardware and networks. The resource limitations associated with many sensor devices suggests that this may be difficult in all but the most capable sensor platforms.

An example of such a platform is the Sun SPOT platform and Squawk JVM[15], which runs directly on SunSPOT hardware. In the most interesting case, a virtual machine architecture might allow sensor networks to deploy application specific data processing code along with the necessary frame identification and dispatch methods at optimal locations in the network. Under this model, it is particularly important to be able to manage resources between virtual machines in order share computing and communication resources fairly among logical networks.

7. RELATED WORK

To the best of our knowledge, there has been little discussion of issues arising from interaction between heterogenous sensor networks.

There is a considerable body of work on sensor networks that are heterogenous in the sense of being composed of sensors with different capabilities. In this context, however, the sensors are assumed to be operating together within a common logical network environment. This kind of device heterogeneity is orthogonal to the problem of semantic interference. Although this discussion has treated interacting networks as homogeneous, the ideas are equally relevant to networks that are internally heterogeneous as well. In fact, such networks are likely to include the more simpler and more limited devices that use the kinds of minimal protocols that are particularly susceptible to semantic interference.

The problem of co-existence between networks at lower layers has been widely studied at both the PHY layer and MAC layer. In this case, the goal is to avoid RF interference and MAC layer contention between transmitters. Network coding and other cooperative transmission techniques have also been proposed. This work is also largely orthogonal to the problem of semantic interference.

There has also been considerable work in the area of architecture for sensor networks, such as the Berkeley modular architecture[17] and Chameleon[8]. These architectures are intended to address the important question: How is the underlying system best structured to make it easy to implement a breadth of both protocols and applications over various underlying radio communication technologies? The focus of this work is therefore on creating good abstractions, primitive operations and models for cross layer interaction.

Although both of these architectures seem to support multiple protocol stacks running on a single node, neither one fully addresses the problem of semantic interference between networks, even in the case where all interacting networks are implemented using the same architecture.

Cooperative sensing has also been an active area of research, often described in the context the "World Wide Sen-

sor Web", e.g. [1]. Here the sensor networks themselves are generally modeled as abstract, independent sources of sensor data. Correlating, coordinating, and searching the data is mostly done in a fully-resourced networked computing environment.

8. CONCLUSION

In this paper, we introduced the notion of "semantic interference", which can occur when frames are mutually intelligible between independent co-located wireless sensor networks. We describe some mechanisms for that allow receivers to identify and discard foreign traffic. Of these, only a cryptographic MIC is likely to be very safe. Based on this discussion, we derive some practical lessons learned about mechanisms for developing sensor networks that can operate effectively an area where there are multiple independent sensor networks. We speculate on three possible mechanisms that can allow for cooperation between networks – creating constructive interference between networks. It is clear that this is an interesting and potentially fruitful topic for further research.

9. ACKNOWLEDGMENTS

Parts of this study were carried out within the VINN Excellence Center WISENET, partially funded by VINNOVA, the Swedish Governmental Agency for Innovation Systems. The author would also like to acknowledge the helpful comments of her colleagues, particularly Dr. Bengt Ahlgren, SICS.

10. REFERENCES

[1] M. Balazinska, A. Deshpande, M. J. Franklin, P. B. Gibbons, J. Gray, M. Hansen, M. Liebhold, S. Nath, A. Szalay, and V. Tao. Data management in the worldwide sensor web. *IEEE Pervasive Computing*, 6(2):30–40, 2007.

[2] J. Brown, J. Finney, C. Efstratiou, B. Green, N. Davies, M. Lowton, and G. Kortuem. Network interrupts: supporting delay sensitive applications in low power wireless control networks. In *CHANTS '07: Proceedings of the second workshop on Challenged networks CHANTS*, pages 51–58, 2007.

[3] M. Buettner, G. V. Yee, E. Anderson, and R. Han. X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks. In *SenSys*, pages 307–320, 2006.

[4] L. Buttyán and J.-P. Hubaux. Stimulating cooperation in self-organizing mobile ad hoc networks. *Mob. Netw. Appl.*, 8(5):579–592, 2003.

[5] Crossbow Technology, Inc. http://www.xbow.com.

[6] A. Dunkels, J. Alonso, T. Voigt, H. Ritter, and J. H. Schiller. Connecting wireless sensornets with tcp/ip networks. In *WWIC*, pages 143–152, 2004.

[7] A. Dunkels, B. Grönvall, and T. Voigt. Contiki - a lightweight and flexible operating system for tiny networked sensors. In *Proceedings of the First IEEE Workshop on Embedded Networked Sensors (Emnets-I)*, Tampa, Florida, USA, Nov. 2004.

[8] A. Dunkels, F. Österlind, and Z. He. An adaptive communication architecture for wireless sensor networks. In *Proceedings of the Fifth ACM Conference on Networked Embedded Sensor Systems (SenSys 2007)*, Nov. 2007.

[9] K. Langendoen. Medium access control in wireless sensor networks. In H. Wu and Y. Pan, editors, *Medium Access Control in Wireless Networks, Volume 2: Practice and Standards*. Nova Science Publishers, 2007. Preprint.

[10] P. Levis, S. Madden, D. Gay, J. Polastre, R. Szewczyk, A. Woo, E. Brewer, and D. Culler. The emergence of networking abstractions and techniques in tinyos. In *NSDI'04: Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation*, pages 1–1, Berkeley, CA, USA, 2004. USENIX Association.

[11] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler. Transmission of IPv6 Packets over IEEE 802.15.4 Networks. RFC 4944 (Proposed Standard), Sept. 2007.

[12] A. Perrig, J. Stankovic, and D. Wagner. Security in wireless sensor networks. *Commun. ACM*, 47(6):53–57, 2004.

[13] R. Roman, C. Alcaraz, and J. Lopez. A survey of cryptographic primitives and implementations for hardware-constrained sensor network nodes. *Mob. Netw. Appl.*, 12(4):231–244, 2007.

[14] Scatterweb, GmbH. http://www.scatterweb.com.

[15] D. Simon, C. Cifuentes, D. Cleal, J. Daniels, and D. White. Java(tm) on the bare metal of wireless sensor devices: the squawk java virtual machine. In *VEE '06: Proceedings of the 2nd international conference on Virtual execution environments*, pages 78–88, 2006.

[16] Sun Microsystems, Inc. http://www.sunspotworld.com.

[17] A. Tavakoli, P. Dutta, J. Jeong, S. Kim, J. Ortiz, D. Culler, P. Levis, and S. Shenker. A modular sensornet architecture: past, present, and future directions. *SIGBED Rev.*, 4(3):49–54, 2007.