Opportunistic Direct Interconnection Between Co-Located Wireless Sensor Networks

Teng Jiang, Geoff V. Merrett, Nick R. Harris
Electronics and Computer Science, University of Southampton, UK
Email: {tj2g11, gvm, nrh}@ecs.soton.ac.uk

***Abstract*— Wireless sensor networks are usually designed to avoid interaction with other networks. To share information, they are usually connected via a backbone network (e.g. the Internet) using gateways. The realization of visions for pervasive computing depends upon effective interconnection between individual networks. As the number of deployed sensor networks increases, the chance of any network having multiple neighbors also increases. In this paper, we argue that a paradigm shift towards 'opportunistic direct interconnection' is required. This enables one network to share information or resources with neighboring networks that it was unaware of at design-time. We present OI-MAC, which supports automatic neighbor discovery and cross-boundary data exchange without sacrificing the independence of each network. The effects of discovery and cross-boundary data injection are evaluated using both analytical models and network simulation. Initial results indicate that neighbor discovery has little effect on latency, while energy consumption increases insignificantly compared to ordinary operations of each node. If network traffic is doubled by packets 'injected' from a neighboring network, latency increases by around 7% while average power consumption increases by 20%.**

Keywords—Wireless sensor network; co-located networks; direct interconnection

I. Introduction

Sensor-based monitoring solutions are widely used in our daily lives, for example in wearable health monitors, intelligent buildings, and environmental monitoring systems [1]. In most cases, each wireless sensor network (WSNs) is designed and configured for a specific application and deployment [2]. To avoid interference and ensure that a WSN's operation is not disrupted, each is designed to enforce a virtual 'wall' around its perimeter which deliberately renders interaction with neighboring networks impossible. While this approach is suitable for many traditional applications, some require cooperation between WSNs. For example, a flood prediction system requires information to be obtained from and potentially shared between spatially-separated WSNs [3]. Currently, the adopted method for supporting cross-network data exchange is to add a gateway node to each WSN (effectively making a small hole in the virtual wall) which allows them to connect with a shared backbone network, usually the Internet. As we move to realize visions for pervasive computing, smart homes and smart cities [4], such forms of interconnection become ever increasingly important. We propose that the unbreakable boundaries around individual WSNs will act as a barrier to growth in pervasive computing; instead we argue that, for many applications, a paradigm shift towards '*opportunistic direct interconnection*' is required.

As the number of deployed WSNs increases, the chance of any one network having multiple neighbors also increases. By *opportunistic interconnect*, we refer to the ability of a WSN to share, and potentially trade, information and resources (such as computation, energy or packet routing etc.) with neighboring networks. Such sharing must be opportunistic as it cannot be conceived at design time; rather it must be identified after deployment. Neighboring networks will also appear and disappear throughout the life of the network as other pervasive services are added or removed within the WSN's vicinity.

To enable opportunistic interconnect, the use of a backbone network accessed via gateways is not suitable, as 1) the backbone may be unavailable or unreliable in many remote or hazardous locations, 2) the backbone is usually accessed through a single gateway in each WSN, reducing flexibility and robustness, and 3) while it can support data sharing between networks, resource sharing cannot be effectively implemented. Therefore, to enable our paradigm shift we instead propose the use of *direct interconnection* between neighboring WSNs. In direct interconnection, nodes are able to dynamically identify and communicate directly with nodes in neighboring networks. In this way, we can remove the virtual wall around a network, instead allowing inter-network traffic to occur at network boundaries rather than via a dedicated backbone.

The benefits of direct interconnection have been previously proposed by Nagata *et al*. [5], who showed that cross-boundary routing can extend the lifetime of co-located WSNs by balancing traffic load. However, solutions to enable direct interconnect have not been investigated, and many challenges are still to be overcome. Most existing techniques for establishing connections between networks – such as Sensor Web [6], Service Oriented Architecture (SOA) [7], Virtual Private Networks (VPN) [8] and 6LoWPAN [9] – assume that the links are already established and focus on higher network layers. In order to build link-layer interconnect, Poorter *et al.* [10] proposed IDRA, which separates protocol logic from packet formation by requiring each node to contain packet descriptors for every communication standard. Considering the number of different WSN protocols in use, this will inevitably lead to resource explosion if implemented uniformly. Research has also been conducted into integrating different link-layer standards, e.g. in IEEE 802.21 [11] and Inter-MAC [12], by defining handover mechanisms. However these assume that nodes know, at design time, which protocols the nodes will be using; clearly this is not possible in opportunistic interconnection.

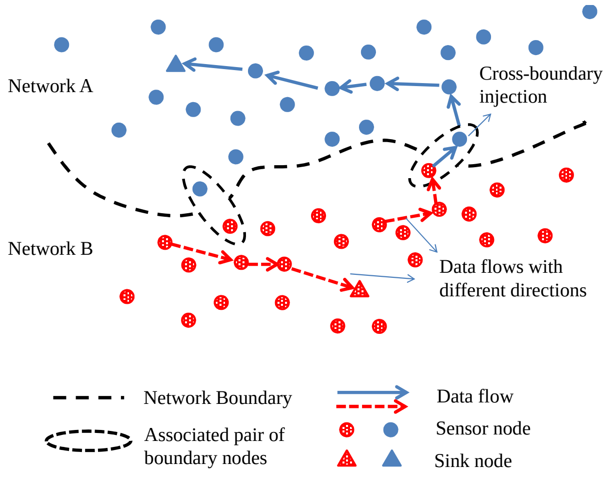


Figure 1: The concept of OI-MAC, showing direct opportunistic interconnection between co-located WSNs.

In this paper we propose OI-MAC, which supports opportunistic discovery and direct interconnect between neighboring WSNs. This is the first paper to present a mechanism for enabling this, which has the potential to redefine how designers consider interconnect in WSNs. The effect of discovery and cross-boundary data injection on packet latency and power consumption are evaluated. Results show that discovery has little effect on packet latency and increased energy cost is insignificant compared with that in normal operations. When network traffic is doubled due to packets injected from neighboring networks, packet delay increases by around 7% while average power consumption increases by 20%. While OI-MAC is built upon the state-of-the-art RI-MAC protocol [13], we believe that the fundamental concepts presented can also be used to extend other protocols.

II. Enabling Direct Interconnection

The primary challenge in achieving direct interconnect is to overcome protocol heterogeneity in neighboring networks. The first solution for overcoming this problem is for each node to be able to interpret every potential protocol (IDRA [10], described in the previous section, is an example of this). However, to be practically useful we argue that each node would have to interpret hundreds of protocols. Clearly this is infeasible in the resource constrained hardware of most WSNs.

Secondly, a shared-protocol could be used by nodes communicating across the boundary. This allows WSNs to use their own, often propriety, protocols for intra-network communication, but use a standardized common-protocol for discovery and communication between networks (in a similar fashion to most gateway-based interconnect schemes). Because neighboring networks are discovered opportunistically, the boundaries are not known at design time; hence, all nodes in the network will need to support two different protocols. Clearly, the resource implications of this are again prohibitive.

The third solution and that adopted by OI-MAC, is to standardize the link-layer protocol used by each node, and integrate both discovery and interconnection functions into the MAC protocol. Taking this to an extreme, one could consider a scenario whereby every node in the world is physically part of a single WSN, but which is virtually segregated into smaller sub-networks (the concept of a virtual sensor network, VSN [14]). However, we believe that the approach taken by OI-MAC is more achievable, as stakeholders can retain real ownership of their own network, and interconnection across a real boundary can be moderated by future higher-layer trading protocols.

MAC protocols designed for WSNs can be classified into two categories, namely single-channel and multi-channel protocols. Single-channel protocols assume that nodes in one network will all communicate on the same radio channel, and a major challenge is therefore regulating media access. Multi-channel MAC protocols enable communication across multiple radio channels, and hence permit simultaneous communication on different channels. OI-MAC is designed to operate using a multi-channel MAC, where adjacent networks communicate using different channels to avoid interference (maintaining the virtual wall), while a common channel is reserved for neighbor-network discovery. The concept behind OI-MAC is shown in Fig. 1. Initially, both networks A and B operate using different channels. During *discovery*, nodes close to neighboring networks (referred to as *boundary nodes*) communicate and become *associated* (shown as an ‘associated pair’ in Fig. 1). Subsequently, nodes in Network A can transmit packets into B using the receiver’s radio channel (and vice-versa). The routing tables in each network are updated such that nodes become aware of their closest boundary node, as a mechanism for *injecting* packets into a neighboring network.

III. OI-MAC Design

To implement OI-MAC, we have extended an existing state-of-the-art MAC protocol, namely RI-MAC. However, we believe that the proposed techniques could be applied to a wide range of different popular MAC protocols.

A. Overview of RI-MAC

RI-MAC [13] is a receiver-initiated MAC protocol, designed to be particularly power-efficient. Transmission is initiated by receivers, and each node periodically announces its wake-up using a beacon (shown in Fig. 2). A node which has a packet to transmit will wake up and wait for the beacon from the destination receiver. Once it receives the beacon, the node transmits its data packet. The receiver responds with an ACK if the packet is correctly received, which is also used to represent the start of a new transmission. Hence, after transmission of an ACK, the receiver stays awake for a period of time (referred to as the dwell time) to ensure that there are no other potential transmitters. If collisions occur at the receiver, it transmits a back-off to ask the potential transmitters to wait before retransmitting. During the back-off period, the receiver remains awake to ensure packet reception. We refer readers to Sun’s description [13] for further information on the operation of RI-MAC.

In order to extend RI-MAC to support opportunistic direct interconnect, OI-MAC adds support for cross-boundary discovery and data transmission with minimal modifications. Key design criteria are to minimize the effect on network performance (e.g. power consumption, packet latency).

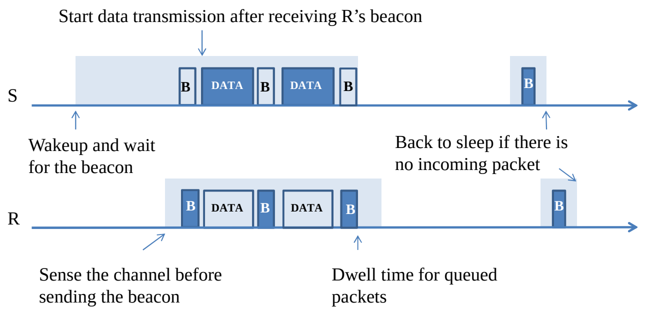

Start data transmission after receiving R's beacon
S
B
DATA
B
DATA
B
B
Wakeup and wait for the beacon
Back to sleep if there is no incoming packet
R
B
DATA
B
DATA
B
B
Sense the channel before sending the beacon
Dwell time for queued packets

Figure 2: Overview of the RI-MAC protocol [13].

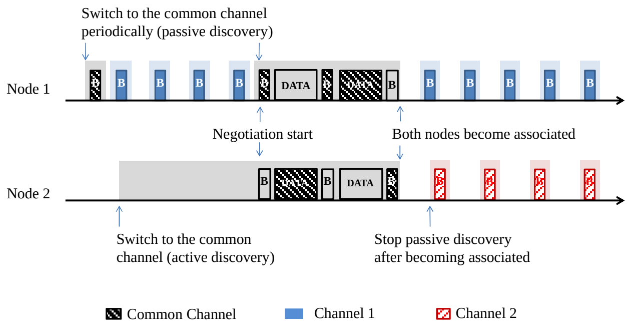

Switch to the common channel periodically (passive discovery)
Node 1
B
DATA
B
DATA
B
Negotiation start
Both nodes become associated
Node 2
B
DATA
B
DATA
B
Switch to the common channel (active discovery)
Stop passive discovery after becoming associated
▨ Common Channel
Channel 1
Channel 2

Figure 3: The discovery mechanism used in OI-MAC.

B. *Broadcast support*

To support route and neighbor discovery, broadcast packets are required, although this is not supported natively by RI-MAC as it focuses on optimizing unicast traffic. Therefore, OI-MAC adds a broadcasting scheme, which is also receiver-initiated as the transmitter remains awake for long enough to hear at least one beacon from a neighbor. On receipt of a beacon, the node will respond by transmitting a unicast packet back. As duplicate beacons may be received from the same node, duplication checks are added to ensure a broadcast packet is received by a node only once. Considering that each node will keep waiting and stop transmitting the beacon itself, this may mean that some broadcast packets cannot be received during this process. To solve this problem, OI-MAC requires that each node transmits a beacon before starting broadcasting.

C. *Discovery scheme*

OI-MAC provides two discovery modes: *active discovery* and *passive discovery* (shown in Fig. 3). In passive discovery, sensor nodes switch to the common channel periodically (defined by the passive discovery period), and broadcast a discovery beacon after performing a clear channel assessment (CCA). After the discovery beacon is sent, the node stays in the common channel for an additional period (the dwell time) to wait for a response from neighboring networks (active discovery, see below). If a response is received, the node enters the handshake process to decide whether to accept the connection (the detail of this process is beyond the discussion scope of this paper, and we assume all queries are accepted). If the query is accepted, both transmitter and receiver become associated. In active discovery, a sensor node switches to the common channel and keeps sensing the channel for the maximum passive discovery period. If a discovery beacon is received, it responds and enters the handshake process. To reduce overhead, we assume that a node can only associate with a single node from another network; hence, if a node knows that a neighbor has become associated with another network, it will stop active discovery.

D. *Cross-boundary data exchange*

Cross-boundary data exchange can be performed after one or more nodes become associated. Once a node becomes associated, it informs the other nodes in its network about the discovery, either through the existing route discovery process or by 'piggybacking' information onto existing beacon packets.

When nodes in the network have packets that they wish to transmit into neighboring networks, they select the best boundary node to route data via. In our implementation, the boundary node is selected based upon the lowest hop count. However, this selection could potentially take other factors into consideration, such as node energy and traffic load. When the boundary node receives a packet destined for the neighbor network, it switches to the corresponding channel, waits for the beacon, and transmits (or 'injects') the packet.

As boundary nodes can be considered as additional data sinks, a network will exhibit packet flows in multiple directions. As a result, there is potential for two nodes wanting to transmit to each other at the same time. In these cases, RI-MAC can enter a deadlock loop, where both nodes that wish to transmit packets to each other wait for a beacon, resulting in neither node receiving the beacon and hence starting transmission. OI-MAC adds a timer to detect this condition, causing it to broadcast a beacon when deadlock is entered.

IV. Evaluation

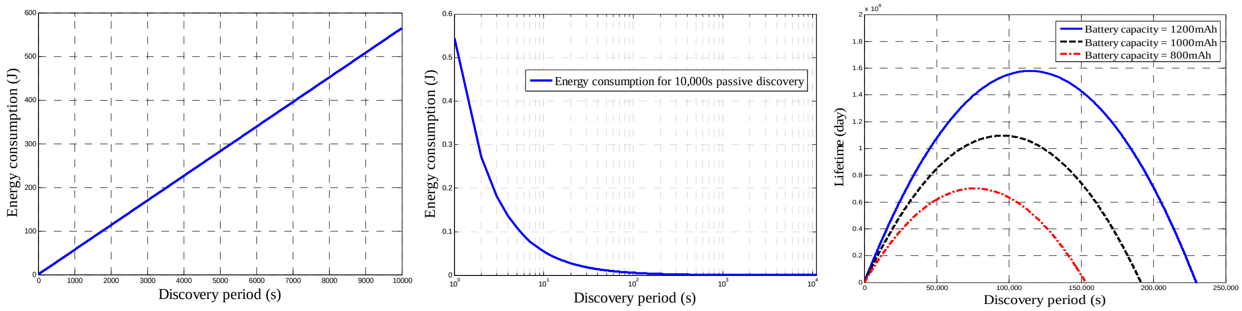
To evaluate OI-MAC, we consider its effect on packet latency and mean power consumption. OI-MAC was simulated in OMNeT++ [15], representing the scenario shown in Fig. 1. The two networks A and B each contain 25 nodes, with a single data sink in each network. The network topology is randomized for each simulation, and the results shown are the averages following 100 simulation runs. The routing protocol used is a simple loop-free routing protocol [16] and the channel model used is the IEEE 802.15.4 path loss model with log-normal shadowing [17]. The OI-MAC parameters used in the simulations are summarized in Table I, while the radio parameters represent those of the Texas Instruments CC2420 transceiver (Table II).

TABLE I: Simulation MAC Parameters

Retry limit	3	Backoff window	0-127
CCA (T_{cca})	128 µs	Size of Beacon	16 byte
Dwell time	Variable	Node wakeup period	2s

TABLE II: Simulation Radio Parameters

Frequency	2.4GHz	Idle listen (C_{listen})	18 mA
Data rate	250 kbps	Sleep current (C_{sleep})	0.02 mA
Tx current (C_{Tx})	17.4 mA	Rx current (C_{Rx})	18.8 mA
Tx range	257 m	Rx-to-Tx time (T_{RxTx})	192 µs
Voltage (V)	3 V	Rx-Tx current (C_{RxTx})	17 mA
Slot time	320 µs	SIFS (T_{SIFS})	192 µs



(a) (b) (c)

Figure 4: The effects of discovery period on energy consumption; (a) for active discovery, (b) for 10,000s passive discovery, and (c) estimated lifetime with different battery capacities. As defined by the OI-MAC protocol, nodes perform active discovery once and then remain in passive discovery mode.

Two experiments were conducted to evaluate the effect of discovery and cross-boundary packet injection. In the first experiment, each sensor node generates and transmits a data packet to the sink periodically. Network performance is compared with and without discovery enabled. In the second experiment, Network B is deployed later than A, to simulate opportunistic appearance. To evaluate the effect of cross-boundary packet injection, Network B injects packets to A through the boundary nodes (once the discovery and handshake processes are complete). The number of injected packets are controlled by the *injection ratio*, α. Whenever a node in Network B generates a packet, it has a possibility α of its destination being to its own sink (i.e. no packets are injected into A when $\alpha = 0$, while all packets will be injected when $\alpha = 1$).

First the effects of the discovery on both average packet latency and the energy consumption have been analyzed. Simualtion results indicate that there is no observable effect on packet latency after enabling discovery. This is expected as the time for each discovery action (including channel switching, beacon broadcasting and dwelling) is around 544μs and hence can be ignored when compared to the wakeup period of 2s. Figs. 4-a and 4-b illustrate the energy consumed by active discovery and passive discovery. As passive discovery is a discrete process, we calculate the average energy consumption during a period of 10^4 s:

$$E_a(p) = C_{listen} \cdot V \cdot p \qquad (1)$$

$$E_p(p) = (C_{listen}T_{cca} + C_{RxTx}T_{RxTx} + C_{Tx}T_{Tx} + C_{listen}T_{SIFS}) \cdot V \cdot \frac{10^4}{p} \qquad (2)$$

where E_a [J] and E_p [J] are the energy consumption for active (1) and passive (2) discovery respectively. The discovery period is denoted by p [s]. T_{Tx} [s] is the transmission period of a beacon, which is calculated based on the beacon length and data rate. The meanings and values of other variables are listed in Table I and Table II. It can be seen that, by increasing the discovery period, the energy consumed by active discovery increases due to the longer sensing time. For passive discovery, however, the energy consumption decreases because the frequency of channel switching and beacon broadcasting reduces. Fig. 4-c shows the lifetime of each node (if only performing discovery), calculated using (3):

$$L = \frac{C - E_a(p)}{E_p(p)} \cdot 10^4 \qquad (3)$$

where C is the energy storage of battery in J. For an 800mAh battery, the lifetime rises while p<76600s (as the energy saved by reducing the discovery frequency is greater than the energy consumed by the increased active discovery period). However, for periods longer than 76600s, the lifetime is reduced as the cost of active discovery outweighs that of passive discovery. Hence, using an 800mAh battery, the lifetime of one node can reach more than 219 years if only performing discovery (clearly this is not realistic in practice); considerably longer than the normal battery life of a sensor network (<10 years). Therefore, the energy cost for discovery can be considered insignificant compared to that of a sensor node's normal operation.

Fig. 5 shows the effect of cross-boundary data injection on original packet latency. The latency slowly increases but the increasing ratio is around 7% even if the traffic in original network has been doubled (α = 1). This is because receiver-initiated protocol can increase idle medium time compared with transmitter-initiated protocols, and allow increased data transmission in one single cycle. The error bar length increases because the probability of collision increases. Fig. 6 shows the effect on mean power consumption after the cross-boundary injection happens. The average power consumption of sensor nodes has suffered around a 20% increase when the intra-network traffic is doubled. This is because data packets can be sent in sequence within one single cycle and therefore power consumption will not increase at a prohibitive rate. By testing the power changes of individual nodes in Network A, we find the injection process mainly increases the power consumption of associated boundary nodes as well as the original relay nodes (nodes for packet collection and forward, which are set during the routing discovery process). By comparing relay nodes and boundary nodes power consumption, we find the relay nodes always have higher power consumption and therefore the relay nodes are likely to be the limiting factor when whole network lifetime is considered.

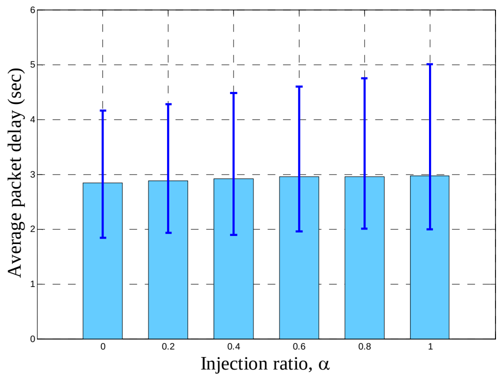


Figure 5: Packet latency when adopting different injection ratios. The median value is used because the data is not following a normal distribution and the error bar represents the data ranging between 5% and 95%. This variance is caused by topology difference and packet collision probability.

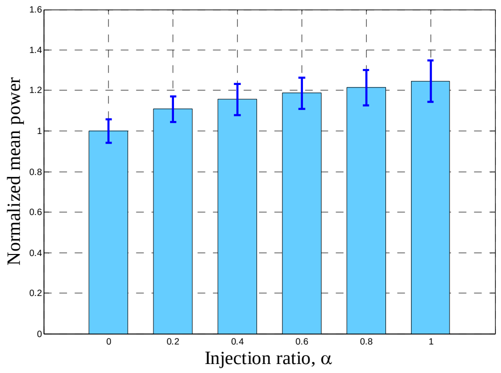


Figure 6: Mean power when adopting different injection ratios. The error bar represents the interquartile range (25%~75%), which is mainly caused by topology differences in the random networks used in the trials.

V. CONCLUSION AND FUTURE WORK

This paper has proposed a solution for opportunistic discovery and direct interconnection between co-located wireless sensor networks. No additional facilities are needed because both discovery and interconnection are embedded into the MAC protocol. To demonstrate feasibility, we have proposed OI-MAC, which can achieve automatic network discovery, handshaking and cross-boundary data exchange without sacrificing a network's independence. Based on our analytical models and simulation results, we have shown:

- Discovery has little effect on packet latency, and the increase in energy consumption is insignificant compared to a node's normal operation;
- Cross-boundary data injection has a small effect on original packet latency (a 7% increase when network traffic is doubled), and the mean power consumption of each node increases with increases traffic.

In addition to designing a specific protocol, this research also demonstrates that direct interconnection can be achieved by modifying an existing MAC protocol. Using similar concepts, many existing and established MAC protocols can also be modified to achieve opportunistic interconnect functionality such as network discovery and cross-boundary data exchange. In this way, node hardware and the upper protocols do not need to be modified.

Our future work will investigate the effects of packet exportation (the opposite of injection), and the effects of direct interconnection on individual sensor nodes – especially those responsible for cross-boundary data transmission and reception. Our target is to minimize the effect of direct interconnection on connected networks and hence ensure feasibility. Once a link-layer interconnection can be established without effecting original network performance, further research is necessary in order to realize our vision, primarily into the management of information trading and cross-boundary resource sharing between co-located sensor networks.

REFERENCES

[1] J. Yick et al., "Wireless sensor network survey," Computer Networks, vol. 52, pp. 2292-2330, 2008.

[2] E. De Poorter et al., "Exploring a Boundary-Less Cooperation Approach for Heterogeneous Co-Located Networks," 2011 IEEE International Conference on Communication, 2011, pp. 1-6.

[3] E. A. Basha et al., "Model-based monitoring for early warning flood detection," the Conf Embedded Network Sensor Systems, Raleigh, NC, USA, 2008.

[4] J. Gubbi et al., "Internet of things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems 2013*.

[5] J. Nagata et al., "A Routing Method for Cooperative Forwarding in Multiple Wireless Sensor Networks," the Conf on Networking and Service 2012, pp. 43-46.

[6] A. Broring et al., "New Generation Sensor Web Enablement," *Sensors*, vol. 11, pp. 2652-2699, 2011.

[7] E. Avilés-López et al., "TinySOA: a service-oriented architecture for wireless sensor networks," *Service Oriented Computing and Applications*, vol. 3, pp. 99-108, 2009.

[8] S. Khanvilkar et al., "Virtual private networks: an overview with performance evaluation," *IEEE Communications Magazine*, vol. 42, pp. 146-154, 2004.

[9] G. Mulligan et al., "The 6LoWPAN architecture," Workshop on Embedded Networked Sensors, Cork, Ireland, 2007.

[10] E. De Poorter et al., "IDRA: A flexible system architecture for next generation wireless sensor networks," *Wireless Networks*, vol. 17, pp. 1423-1440, 2011.

[11] A. De La Olivaet al., "An overview of IEEE 802.21: media-independent handover services," *IEEE Wireless Communications*, vol. 15, pp. 96-103, 2008.

[12] M. Brzozowski et al., "Inter-MAC—From vision to demonstration: Enabling heterogeneous meshed home area networks," *ITG Conf Electronic Media Technology (CEMT)* , 2011, pp. 1-6.

[13] Y. Sun et al., "RI-MAC: a receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks," Conf Embedded Network Sensor Systems, Raleigh, NC, USA, 2008.

[14] M. M. Islam et al., "A Survey on Virtualization of Wireless Sensor Networks," *Sensors*, vol. 12, pp. 2175-2207, 2012.

[15] OMNeT++. (2012). Available: http://www.omnetpp.org/

[16] MiXiM. (2013). Simple Module WiseRoute. Available: http://mixim.sourceforge.net/doc/MiXiM/doc/neddoc/index.html?p=org.mixim.modules.netw.WiseRoute.html

[17] A. Köpke, "Simulating wireless and mobile networks in OMNeT++ the MiXiM vision," Int'l Conf Simulation Tools and Techniques for Communications, Networks and Systems, Marseille, France, 2008