Interconnecting Wireless Sensor and Wireless Mesh Networks: Challenges and Strategies

Stefan Bouckaert, Eli De Poorter, Pieter De Mil, Ingrid Moerman, Piet DEMEESTER
Ghent University – IBBT – Department of Information Technology (INTEC) – IBCN
Gaston Crommenlaan 8/201, B-9050 Gent, Belgium, stefan.bouckaert@intec.ugent.be

Abstract—Wireless sensor networks consist of several hundredths of simple sensing devices, equipped with a radio. They are typically used for monitoring and automation purposes of large areas. Due to their simplicity, these networks quickly run out of energy, and often have problems regarding scalability and available bandwidth. To solve these issues, current research is mostly limited to the addition of extra sinks to the network, or the use of gateways to request sensor data over the Internet. In this paper, we explore how wireless sensor networks can be combined with wireless mesh networks to obtain a more optimized solution. The mesh network can be used to connect separate sensor networks, to connect sensor nodes with a monitoring platform, or as a scalable backbone for sensor to sensor communication. Additionally, we give an overview of the advantages and disadvantages of existing interconnection techniques between wireless and mesh networks, and propose several new interconnection strategies. Finally, we identify remaining challenges, upon which future research can be based.

I. INTRODUCTION

Since wireless network interfaces became cheap and widely available at the beginning of the millennium, there has been an increased research interest in various types of wireless networks and wireless network applications. Within this research area, the focus initially was on pure ad hoc networks, providing solutions at various OSI layers to enable the interconnection of (mobile) wireless nodes without the need for any form of central infrastructure. Because of the wide applicability of wireless networks, several network sub-types emerged that essentially re-use the idea of the self-forming, self-configuring, infrastructureless ad hoc networks, but operate in specific scenarios using specialized types of hardware. Among several sub-types, wireless mesh networks and wireless sensor networks evolved into independent research topics.

In wireless mesh networks (WMN) [1], the network nodes are considered to be part of the infrastructure and are dedicated to the routing task. The mobility of the mesh nodes is limited or zero, and their processing, memory and bandwidth capacities generally exceed those of traditional ad hoc network nodes. Additionally, the power consumption requirements are often less stringent than those of wireless ad hoc networks.

Conversely, wireless sensor networks (WSN) [2] are formed between sensor devices. These devices are characterized by their small size and low cost, but are burdened with relatively low processing and memory capabilities, a limited power supply, and relatively low link bandwidths. While some sensor devices are immobile, other sensor devices are attached to moving objects and can be highly mobile. Research efforts are mainly located in the field of power efficiency and low-processing routing [3]. Various types of WSNs exist, ranging from simple single hop data collection mechanisms to intelligent multi-hop sensor networks. The latter type of networks allows large scale deployments, and enables the use of advanced mechanisms providing sensor values on demand, or adjusting packet routing according to an application’s need.

In the remainder of the paper, the WSN term will refer to intelligent multi-hop sensor networks.

In traditional WSNs, sensor data is often collected by a single central device called a sink. This central device is connected to an external management server and/or a database, which authorized users can use to look up the interpreted or raw data. However, by introducing actuator nodes, multi hop WSNs can also exist in isolation from other networks, e.g. a temperature sensor can directly steer a fire alarm, or an actuator node acting as an on/off switch for an HVAC unit.

Emerging WSN applications such as Wireless Building Automation [4], Urban Monitoring [5] or Intelligent Transportation Systems push the demand for large scale WSNs consisting out of thousands of nodes, dramatically increasing the hop count towards a sink. This issue can be resolved by adding sink nodes to the networks, thereby increasing scalability. By interconnecting the sink nodes, the data is directed towards the management server. Because of its auto-deployment characteristics, a WMN is a convenient way to interconnect the sink nodes.
However, a traditional two-tiered data collecting mechanism (sensor nodes to the sink, sink to the backbone network) might result in suboptimal routing, especially when sensor node to sensor node applications are deployed in the network. Consider the building automation example of Figure 1, in which several sensor nodes with IEEE802.15.4 communication interfaces are connected to an IEEE802.11 based WMN via gateway nodes. These gateway nodes are equipped with both a sensor and a mesh interface and can act as sinks. It can be seen on the figure that the temperature reading of a single sensor node is used to steer solar screens and a HVAC unit. If the temperature in a room rises above a first threshold on a sunny day, solar screens are lowered to decrease passive solar gain. If, despite this action, temperature keeps rising, crossing a second threshold, the central cooling installation is activated. While the solar screens are located in the immediate environment of the temperature sensor node, the HVAC unit is located in a room on the other side of the building. If the temperature sensor node is sending data to the sensor node controlling the solar screen via its sink (path (ii)), more transmissions are needed than when using the direct path labeled (ii). On the other hand, if the temperature sensor uses the 7-hop path (2) to reach the HVAC unit actuator, the efficiency in terms of number of hops using the sensor technology is reduced compared to taking path (1) over the wireless mesh backbone.

This example shows that adding a mesh network to a sensor network allows decreasing the load that is imposed on the sensor networks. Interconnecting WSNs with WMNs is especially relevant in case (i) a sensor network deployment has grown to large proportions and can no longer provide the needed QoS guarantees such as bandwidth or delay on its own, or (ii) an area that is already equipped with a mesh network is put under the control of sensor networks. However, because of the heterogeneity in terms of transmission technology, neighbor discovery, routing protocols, performance metrics and energy requirements, combining sensor and mesh links is a non trivial task.

Currently, very little research focuses on the interconnection of wireless sensor networks and wireless mesh networks. Although partial solutions to some interconnection issues are available in literature, a systematic overview of interconnection techniques is missing.

Therefore, the goal of this paper is to reveal challenges in interconnecting WSNs and WMNs (Section II), and to explore different strategies that can be used to realize the connection (Section III). The strategies are further discussed in Section IV. In the same section, we identify key issues that still need to be solved before fully optimized interconnection strategies can be deployed. Finally, Section V concludes the paper.

Some of the interconnection techniques, or aspects thereof, have already extensively been described in literature, while other techniques such as the translation gateway (Section III-D) or stack virtualization (Section III-E) have rarely or even never been reported upon yet in the context of interconnecting WSNs and WMNs. The goal of this paper is not to pronounce a single interconnection strategy as the ultimate novel solution for any combined WSN/WMN solution, but rather to provide the reader with a wide selection of interconnection strategies, each with their own positive and negative aspects.

II. CHALLENGES INTERCONNECTING WSNs AND WMNs

When interconnecting WSNs and WMNs, several challenges need to be tackled.

First, in order to bridge the differences in radio technology, gateway nodes are needed that are equipped with both a sensor and a mesh technology interface. In spite of having different radio technologies, the WSN and WMN might operate in the same frequency range, causing mutual radio interference. The success of combining the two types of networks will largely depend on the ability to avoid or cope with radio disturbances [6], [7].

Second, if a mesh network is added to an existing sensor network, or vice versa, backwards compatibility becomes a major issue. If a WMN is added to a WSN in a fully transparent way, no adjustments are needed to the sensor nodes. Analogously, if an existing WMN backbone can be reused to interconnect sensor networks, deployment costs will be lower.

Third, if a sensor node is to decide independently whether to route the traffic over sensor hops only, or if it can increase efficiency by using the supporting WMN backbone, the sensor node should somehow be informed about the availability and service quality of alternative routes that go through the WMN. If the WMN is used transparently, a meaningful translation of WMN to WSN metrics, should be injected into the sensor network, masking the alternative WMN path as yet another sensor path.

Fourth, addressing schemes can be different in WSN and WMN networks. For example, while an existing company WMN will probably use IPv4, a WSN network added at a later time might use 6LoWPAN addresses. When the WSN need to communicate with the WMN (and vice-versa), an unique mapping between the address types is required.

Fifth, even when WSN and WMN are using the same IP version, the pursued routing strategies might be completely different.

Sixth, WSN often implement sleeping schemes in order to increase power efficiency. If a packet is injected from the WMN to the WSN, the gateway node needs to be aware of the synchronization strategies in place.

For all of the interconnection strategies defined in the next paragraph, and, depending on the specific WSN and WMN technology to be combined, a subset of the above challenges will need to be tackled.

III. INTERCONNECTION STRATEGIES

As stated in Section I, routing WSN traffic over a mesh backbone has several advantages. In general, the mesh network has more bandwidth and energy available. If, in order to minimize energy use, the transmission power in the sensor network is kept to a minimum, a smaller amount of intermediate hops between source and destination can be achieved by going
through the WMN, reducing the end-to-end delay. The end-to-end delay is reduced even more in case sleeping schemes are used inside the WSN.

In this section, we present and discuss several approaches for routing the WSN traffic over a mesh backbone to improve the performance of the WSN.

A. Repeater backbone

A simple solution is the design of a ‘repeater’ backbone (Figure 2a), where all gateway nodes are set to promiscuous (always on) mode. Whenever a gateway node receives a sensor packet from the WSN, this packet is sent over the mesh backbone to all other gateway nodes, using the mesh routing protocol or pre-set tunnels. The packet is not processed by the mesh backbone: the original packet is sent without any changes through the sensor interfaces of all gateways. In effect, the mesh backbone acts as an invisible repeater towards the WSN.

The repeater solution is simple to implement, and can be added to an existing sensor network in a transparent way. However, several disadvantages are associated with this technique. (i) Captured packets at one gateway are sent to every other gateway node, multiplying the number of sensor packets sent in the presence of \( n \) WMN nodes by \( n - 1 \), increasing radio interference and decreasing scalability of the network. Moreover, (ii) repeaters should be able to detect packets from other repeaters, in order to avoid sending the same packets over and over again. Furthermore, (iii) since the repeating gateways do not process the packet, sleeping schemes of sensor nodes residing at different edges of the network are not synchronized. Finally, (iv) if multi-channel sensor protocols are used, the gateway node needs to add a sensor interface for every channel to be supported.

B. Supernode

Some of the drawbacks of the repeater backbone are caused by the fact that the sensor nodes are completely unaware of the fact that packets are forwarded through a supporting WMN. In addition, the forwarding backbone is delivering the packets unintelligently because the sensor interface of the gateway nodes is simply used as a sniffer/packet transmitter.

By combining an intelligent sensor interface with the forwarding behavior of the repeater backbone solution, these issues can partly be resolved. Figure 2b shows how the sensor interfaces of all gateway nodes announce themselves to the sensor network as being sensor node \( E \). If a packet is received by any of the gateway nodes, it is processed and added to a common state of the virtual node \( E \), maintained by the WMN gateway nodes. A common state can be implemented in many ways. The easiest solution is to first let the receiving gateway process the packet, and then forward the same packet, or the relevant information that results in a state update, to the other gateway nodes.

If a sensor packet is re-injected from the WMN to the WSN, the packet is transmitted by every gateway. From the perspective of the sensor nodes, the WMN thus behaves as a sensor node \( E \) with extended communication range and large energy capacity. As such, this technique can be added transparently to an existing sensor network.

C. Termination gateway

The techniques presented above enable the WSN to transparently route traffic over the mesh backbone, but as the WMN infrastructure is used as an unintelligent forwarding backbone, many ‘useless’ packet transmissions take place. In addition, the above mechanisms only support communication between sensor nodes that were already part of the network. By adding intelligence to the WMN and configuring the gateway nodes as an explicit sink for the WSN, a two tier architecture supporting intelligent routing in the WMN is constructed. We call the sink nodes termination gateways, since, from the point of view of the sensor nodes, the network paths are always terminated at the sinks. Packet forwarding in the mesh network is performed using a traditional mesh protocol stack.

In the example of Figure 2c, the termination gateways broadcast their presence over the WSN, so that sensor nodes can add a default path to their routing tables. Based on a metric indicating the quality of the path towards a specific gateway, the sensor nodes register with a gateway of their choice. Node \( A \) can then easily contact node \( B \) by routing packets to the gateway. The choice for a specific gateway can vary over time. In [8], a registration method is specified which allows the sensor nodes to choose a different gateway without the complexities of terminating an existing registration.

When communication between sensor nodes is required, sensor nodes must choose whether to route the packet over the WSN, or to send the packet to the gateway node. This choice can be based on many criteria, for example by querying the gateway for a performance metric for contacting a specific node through the WMN, and comparing this with the potential other routes to the same node in the local database. This way, transmission between node \( C \) and \( D \) will happen directly, rather than using the path through the gateway nodes.

If sensors are organized in a subnet per gateway, communication between subnets necessarily needs to pass through the termination gateways. Subnet based networks reduce the amount of routes in the sensor routing tables and allow for a straightforward implementation of spectral multiplexing. This increases scalability of the sensor network and simplifies the routing task of the WMN, as no routes to individual nodes need to be stored. However, Figure 2c illustrates how node \( C \) and node \( D \), now belonging to a different subnet, can no longer directly communicate to each other. This results in suboptimal routing. Additionally, in case some sensor nodes are mobile, a mobility solution needs to be implemented to support roaming in subnets.

Implementing a termination gateway solution differs greatly from using a Supernode connection strategy, and has consequences for both the WSN and WMN.

In the WSN, the routing is no longer flat, because of the introduction of an explicit gateway/sink node. Adding the sink node results in an additional overhead for signaling:
unless a sensor node decides blindly on whether to take a full sensor node path or to route the traffic through its termination gateway, registration and information queries between sensor node and gateway are to be supported. This also means that backwards compatibility is not supported by default: the WSN needs to be designed specifically for use with the termination gateway nodes. In [9], a similar solution is described. The authors point out the risk of a termination gateway, there called Application Level Gateway, becoming a single point of failure in the network.

In the WMN, backwards compatibility is supported to some extent: by adding termination gateway nodes to an existing WMN, the existing network can be reused. The termination gateways are responsible to keep track of the sensor nodes in their own and other clusters, and tunnel the sensor packets over the existing network. Because packets are only forwarded to the gateway in which a destination sensor node resides, the number of transmissions in the mesh network is severely reduced compared to the previous solutions.

Note that, apart from being able to contact a termination gateway, the WSN essentially has no idea about the internal network architecture of the WMN. As the WSN is a client network of the WMN, it now fully relies on the correct operation of the WMN to retrieve some quality metric for contacting another node through its parent network.

D. Translation gateway

In order to avoid frequent signalization between sensor nodes and gateways, and to enable direct WSN to WMN communication, a more sophisticated approach is required. In the previous strategy, a gateway node was responsible for tunneling the sensor packets over the mesh network, towards the correct destination gateway. The task of the gateway nodes can be relaxed by fully translating sensor packets to mesh packets and vice versa. These gateway nodes are called translation gateways. In Figure 3a, it is shown how sensor node B sends a packet to gateway U as if it were another sensor node. Within each network, the packets are routed using the local protocols. The gateways thus should provide (i) protocol translation, (ii) metric translation, (iii) address translation, and (iv) packet translation. Using protocol translation, protocol specific packet headers of the WSN (WMN) protocols are translated into the corresponding packet headers of the WMN (WSN) protocols. It is the responsibility of metric translation to ensure that routing decisions are based on correctly interpreted metrics; one possible application would be for a gateway node to announce a 3-hop path in the WMN as a 1-hop path in to a sensor node, because the protocol developer was able to show that an extra 3 hops in the mesh network is to be preferred above a single hop path within the WSN. Address translation provides a one-to-one mapping between the address in WMN and WSN. As an example, NAT-TP [10] specifies a mechanism that supports communications between IPv4 and IPv6 terminals by translating the header of the IP packets. Packet translation is used to convert packets between different representations, for instance, conversion from little endian to big endian, or the segmentation of packets.

As a result, it is impossible for intermediate nodes to know whether the packet originates from a sensor or mesh node. For the WMN, each sensor node is indistinguishable from mesh nodes, and vice versa. As such, this approach is fully transparent for the WMN and WSN.

A clear disadvantage is the difficulty or, at times, impossibility of providing metric, protocol and address translation. For instance, translating from a reactive routing protocol in the WSN to a proactive routing protocol in the WMN is very impractical. For the case of NAT-TP, [11] discussed a number of significant issues. The major issue is the increased complexity introduced by translation gateways and the fact that some protocol headers are not translatable.

The translation gateway technique thus can only be applied when the network protocol used in the WMN and WSN show many similarities. In addition, because all network nodes will appear to be in the same network, the translation gateway solution scales worse compared to the termination gateway solution. Since all nodes should have a (virtual) address in the WSN and WMN, small subnets cannot be used. Note that, if the same network would be constructed, keeping the WSN but also replacing all mesh nodes with sensor nodes, the solution might even scale worse: the amount of nodes in the network...
would still be the same, thus the routing table remains large. However, the network would no longer be able to use the better performing mesh nodes, most likely decreasing scalability.

On the plus side, if it is possible to connect a WSN and WMN using a translation gateway, the sensor nodes will inherently route packets over the best possible path, and, direct communication between any two nodes in any network becomes possible. This allows application designers to develop advanced services which can fully exploit the diversity of the nodes, without having to worry about the underlying network topology. Because the translation task is the responsibility of the translation gateways only, this solution provides full backwards compatibility with existing sensor and mesh nodes.

E. Stack virtualization

Protocol or metric translation requires a thorough knowledge of the protocols, and complex translation software. An alternative approach is to provide every mesh node with a ‘virtual WSN stack’. This stack contains all relevant WSN network protocols. Whenever a WSN packet arrives at a mesh node, this stack processes the packet as if the packet would have been processed by a sensor node. This approach is similar to the virtualization approach used in the Akari project[12]. In this project, among other goals, the researchers add virtual stacks to the edge and core components in the Internet, in order to enable clean-slate Internet protocols to be tested on an operational network. By adding a virtual stack instead of replacing an existing stack, backwards compatibility is guaranteed.

Between two mesh nodes, an encapsulated sensor packet is sent over a mesh link. Before and after encapsulation, the packet is processed by the virtual sensor stack. Except at the gateway, the virtual sensor stack of all mesh nodes can be tweaked as to avoid any sleeping schemes that might be present. For communication between WMN and WSN, the gateway uses the sensor stack implemented on its sensor interface. Thus, as illustrated in Figure 3b, the sensor nodes use the mesh nodes as if they were a part of the WSN.

This virtual stack solution is closely related to the translation gateway solution in terms of scalability and need for metric translation. However, protocol translation, packet translation and address translation are supported natively by the virtual stack. The main drawback of this solution is that the solution is not compatible with existing mesh networks, since in general, it will not be possible to implement a virtual stack on already deployed mesh networks.

F. Identical stack

As stated in the introduction, WSNs and WMNs are characterized by their specific challenges. Nevertheless, these two type of networks show a lot of similarities: in essence, both WMNs and WSNs transmit data over a shared medium, and need to do this as efficient as possible, given their specific processing, memory, transmission and power constraints. A solution that was originally developed for WMNs can often be adapted in order to be used in WSNs, and vice versa. While sensor nodes with better specifications will be released in the future, mesh nodes will be developed that are cheaper and more energy efficient.

This observation leads us to expect that WSNs and WMNs will eventually converge into a single type of network, built on top of heterogeneous nodes. A first step towards this convergence was taken with the development of 6LoWPAN [13], a frame format for transmission of IPv6 packets on IEEE 802.15.4 networks.

Considering these facts, another option to interconnect WSNs and WMNs is to use an identical routing and addressing scheme in both networks (Figure 3c). Each multiple interfaced node uses a hardware abstraction layer, such as described in [14], to mask the heterogeneity of the underlying interfaces, while the upper layers are identical. This is, in effect, a special case of the translation gateway, in which the upper layer protocols are identical.

The main risk of using an identical upper layer stack is the inefficient use of the different node types. For this reason, adaptive network protocols need to be developed, that are able to adjust their operation characteristics to the individual node needs, while guaranteeing interoperability. Note that WSNs are already becoming increasingly heterogeneous [15], [3], confirming the need for adaptive protocols, which take the capabilities of the intermediate nodes into account.
Finally, with the development of cognitive radio solutions [16], it will be possible to deploy networks consisting of identically manufactured nodes, that do not only adapt their upper layer parameters to match the lower layer settings, but also adapt their radio configuration to meet the connectivity requirements.

IV. ADDITIONAL DISCUSSION AND FUTURE DIRECTIONS

In the previous sections, we discussed and proposed several possible interconnection strategies. Which strategy is optimal depends strongly on the considered use case.

An overview of the different techniques is given in Table I. The techniques are sorted according to their implementation complexity. The following characteristics of each technique are given:

- **WSN→WSN**: can the technique be used for WSN to WSN communication over a WMN backbone?
- **WSN→WMN**: can the technique be used for WSN to WMN communication (and vice-versa)?
- **Transparent for WSN**: can the technique be used without adaptations to the WSN?
- **Transparent for WMN**: can the technique be used without adaptations to the WMN (gateway nodes excluded)?
- **WMN behavior**: from the viewpoint of the WSN, what does the WMN backbone look like?
- **Scalable**: does the solution scale for large networks?
- **Synchronization**: does the technique support synchronized sleeping schemes?
- **Metric translation**: is metric translation required?
- **Complexity**: how complex is the solution to implement?

<table>
<thead>
<tr>
<th>Technique</th>
<th>Complexity</th>
<th>Scalable</th>
<th>Synchronization</th>
<th>Metric translation</th>
<th>WMN behavior</th>
<th>Transparent for WSN</th>
<th>Transparent for WMN</th>
<th>WMN behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSN→WSN</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>WSN→WMN</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Transparent for WSN</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transparent for WMN</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WMN behavior</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Synchronization</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Metric translation</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Complexity</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

There are several caveats associated with using a mesh backbone to optimize the performance of a WSN. The intelligent use of a mesh backbone has a positive effect on the reliability of transmissions, increases the total possible throughput and decreases the average energy consumption of the WSN network. On the other hand, utilizing a WMN with the single intention of increasing the range of a WSN makes little sense: provided enough energy is available, most sensor nodes can transmit as far as mesh node. Additionally, using a mesh backbone does not necessarily result in less delay: if no sleeping schemes are used, the processing time of a packet on a sensor node is about the same as the processing time of a packet on a mesh node.

Because of the wide range of transmission technologies, network topologies and applications, there is no single answer to the question of which of the different interconnection strategies provides the best solution. As such, whenever a specific network is to be rolled out, a study is needed to investigate if it is advised to use a mesh backbone, or if the additional complexity is not worth the effort.

To optimize routing over the WMN, packet that cross multiple networks require metric translation. As more and more diverse radio technologies have to cooperate, there will be an increasing need for universal, technology independent cost functions associated with each path. Multiple cost functions can be associated with a single path: available bandwidth, total delay, reliability, etc. Choosing between paths with different costs should be based on the application requirements.

Finally, as more independent wireless networks become co-located, the scarce bandwidth has to be shared. To do so efficiently, the networks need to cooperate across their boundaries. They will need to exchange information about their routing protocols, spectral requirements and capabilities. There is a need for communal optimization protocols between independent networks in order to better utilize the scarce network resources [17].

V. CONCLUSION

Both wireless sensor networks and wireless mesh networks are self-organizing and easy to deploy. There are many reasons to combine these different types of networks. Most strikingly, the detailed sensing capabilities of sensor networks can be enhanced with increased throughput, reliability and energy provisions in the form of mesh networks. However, currently, very little research focuses on interconnecting both types of networks. Additionally, a systematic overview of existing interconnection techniques is lacking.

Therefore, in this paper, we gave an overview of existing techniques, and proposed several new interconnecting strategies for WSNs and WMNs. Each technique has its own advantages and disadvantages, as discussed in the respective sections. Based on these discussions, interested developers can easily get an idea of the characteristics of possible interconnections approaches, and choose the most appropriate solution that fits their needs.

Unfortunately, due to the complexity of the problem, many challenges remain unsolved. These challenges vary from the definition of global routing metrics, to the design of translation protocols.

We feel that the differences between sensor and mesh networks will ultimately disappear. Thus, in time, WSNs and WMNs will converge, requiring the development of adaptive network protocols suitable for networks with strongly heterogeneous nodes.

The combination of wireless sensor networks and wireless mesh networks is a promising approach. As such, we hope that more researchers will focus on these interesting challenges.

ACKNOWLEDGMENT

S. Bouckaert and E. De Poorter thank the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen) for funding this research through a grant. This research is partly funded through the IBBT-DEUS and FWO-CLAWS projects.

REFERENCES

TABLE I
COMPARISON OF THE DIFFERENT INTERCONNECTION STRATEGIES. [N.A.: NOT APPLICABLE. (*) : CF. DISCUSSION SECTION III.]

<table>
<thead>
<tr>
<th></th>
<th>WSN/WSN</th>
<th>WSN/WMN</th>
<th>Transparent for WSN</th>
<th>Transparent for WMN</th>
<th>WMN behavior</th>
<th>Scalable</th>
<th>Synchronization</th>
<th>Metric transl.</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repeater</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>no-hop</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>low</td>
</tr>
<tr>
<td>Supernode</td>
<td>yes</td>
<td>only gateway</td>
<td>no</td>
<td>yes</td>
<td>single-hop</td>
<td>no</td>
<td>yes (*)</td>
<td>no</td>
<td>medium</td>
</tr>
<tr>
<td>Termination</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>sink</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>medium</td>
</tr>
<tr>
<td>Translation</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>multi-hop</td>
<td>varies (*)</td>
<td>yes</td>
<td>yes</td>
<td>high</td>
</tr>
<tr>
<td>Virtualization</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>multi-hop</td>
<td>no</td>
<td>yes</td>
<td>high</td>
<td></td>
</tr>
<tr>
<td>Identical stack</td>
<td>yes</td>
<td>yes</td>
<td>n.a.</td>
<td>n.a.</td>
<td>multi-hop</td>
<td>n.a.</td>
<td>yes</td>
<td>n.a.</td>
<td>high</td>
</tr>
</tbody>
</table>


