Internet of Things Virtual Networks

Bringing Network Virtualization to Resource-constrained Devices

Isam Ishaq, Jeroen Hoebeke, Ingrid Moerman, Piet Demeester
Department of Information Technology (INTEC)
Ghent University – iMinds
Ghent, Belgium
{isam.ishaq, jeroen.hoebeke, ingrid.moerman, piet.demeester}@intec.ugent.be

***Abstract*—Networks of smart resource-constrained objects, such as sensors and actuators, can support a wide range of application domains. In most cases these networks were proprietary and stand-alone. More recently, many efforts have been undertaken to connect these networks to the Internet using standard protocols. Current solutions that integrate smart resource-constrained objects into the Internet are mostly gateway-based. In these solutions, security, firewalling, protocol translations and intelligence are implemented by gateways at the border of the Internet and the resource-constrained networks. In this paper, we introduce a complementary approach to facilitate the realization of what is called the Internet of Things. Our approach focuses on the objects, both resource-constrained and non-constrained, that need to cooperate by integrating them into a secured virtual network, named an Internet of Things Virtual Network or IoT-VN. Inside this IoT-VN full end-to-end communication can take place through the use of protocols that take the limitations of the most resource-constrained devices into account. We describe how this concept maps to several generic use cases and, as such, can constitute a valid alternative approach for supporting selected applications. A first implementation demonstrating the key concepts of this approach is described. It illustrates the feasibility of integrating resource-constrained devices into virtual networks, but also reveals open challenges.**

Keywords-Internet of things; sensors; actuators; virtualization; virtual network; network architecture; end-to-end communication; resource-constrained devices

I. INTRODUCTION

The Internet and its applications are evolving in many directions. In one direction, it is expected that small, embedded devices such as sensors and actuators will become a cornerstone of the Future Internet transforming it into the Internet of Things (IoT). These devices enable us to collect information about the physical world and inject it into the virtual world where it can be used for a plethora of applications, crossing many application domains. To realize this Internet of Things and to be able to make use of the data generated by these resource-constrained devices, these devices need to be integrated into the Internet and to be able to connect to other devices. They have to make their data accessible to interested parties, which can be web services, smart phones, cloud resources, etc.

Making these data available through the Internet is one thing; doing this in a controlled way, not exposing data to the whole world, is another thing. As such, integrating resource-constrained devices into the Internet is more than simply connecting these devices to the Internet in one way or another. The traditional approach to achieve this, is through the use of gateways that reside at the border of the Internet and the sensor networks. They incorporate intelligence and access control, collect the sensor data themselves and expose it to the Internet. This approach has certain advantages, but also limitations. In some cases, interested parties want to be able to access the data directly, requiring direct connectivity to the sensors. Here, the gateways play a less prominent role, primarily dealing with the translation of Internet protocols to sensor protocols and vice versa, implying that more complexity is shifted to the sensors.

In this paper we present a complimentary approach, by integrating all objects that need to cooperate, including both resource-constrained and non-constrained devices, into a secured virtual network, called an Internet of Things Virtual Network or IoT-VN, in which direct end-to-end communication can take place. This approach was named "managed ecosystems of networked objects" (MENO) and was first introduced and discussed in detail in [1]. Such an ecosystem was defined as "*a completely independent, managed, observable, virtual environment of interdependent, networked objects that cooperate in harmony*." In this paper we now lay the foundation of this new concept by presenting a first implementation demonstrating its key concepts. As such, the main contributions of this paper are the following:

- A detailed discussion of the IoT-VN concept and its potential benefits in several use cases.
- A middleware for non-constrained devices to securely exchange raw data over layer 2 or layer 3 virtual links inside a self-organized virtual network.
- Simple extension to resource-constrained devices: using neighbor discovery direct virtual links can be established.
- An evaluation of the feasibility to run proprietary lightweight protocols inside the resulting virtual network.

In the following, we first describe in more detail the current approaches to integrate resource-constrained devices into the Internet and their advantages and limitations (Section II). Next, we briefly describe our approach and compare it with related work (Section III), followed by an illustration of how it can be applied to a number of generic use cases (Section IV). Our implementation and first test deployment are then described in Sections V and VI. Finally we summarize our conclusions in Section VII.

II. Current Approaches and Limitations

It is envisioned that resource-constrained devices, such as sensors and actuators will play an important role in the Future Internet and will enable a whole new set of novel services. These devices are considered resource-constrained because they have less memory, CPU, energy and bandwidth than typical hosts. To preserve energy, these devices often have long sleeping schemas and they typically form networks that are lossy and are a lot less reliable than Ethernet. For these reasons current Internet Protocols are generally considered not suitable to run on resource-constrained devices. Other approaches are required to expose services offered by these resource-constrained devices to the outside world. Most of these approaches can be categorized into two main categories: use of gateways and integration of sensors into the IP-world.

A. *Use of Gateways*

A multitude of specialized control protocols for resource-constrained devices are now used in the industry, e.g. the ZigBee [1] standard or BACnet [2] - a data communication protocol for building automation and control networks. In the absence of widely accepted standard protocols, many vendors were encouraged to develop proprietary protocols to run inside their sensor networks.

Connectivity between the Internet and many of the sensor networks is nowadays achieved through the use of gateways or proxies. These gateways have to translate between protocols used in the Internet and protocols used in the sensor networks.

Fig. 1 displays two various sensor networks that are connected to the Internet by gateways. Users on the Internet have to connect to the gateways in order to obtain data from the respective sensor network. There are several ways how a gateway can handle such user requests.

- The gateway from vendor1 translates standard Internet protocols into the proprietary sensor protocols and relays the requests to the sensors in its network. It then gets the answers from the relevant sensors by means of the proprietary sensor protocols and sends back the appropriate replies to the user using standard Internet protocols. The gateway offers an API that applications should use in order to create requests that can be understood by the gateway.
- Although it might look the same to the user, the gateway from vendor2 behaves in a different way than the gateway from vendor1. This gateway contains a database with pre-collected sensor data. When it gets a request from a user on the Internet, it replies directly to the requester using the data in the database. In some cases, the gateway is simply running a web server that makes the data available to the outside world. The user usually cannot tell, whether the returned data is coming in real-time from the sensors or whether it is coming from a value that has been previously stored in a database.

The use of gateways has certainly many advantages. Since it provides a single entry to the sensor network, it shields the sensor resources from the Internet and enables a high degree of access control. If needed, the gateway can also process and aggregate data from different sensors and present them in a uniform way to the users.

However there are also disadvantages that accompany the use of gateways. The gateway is the only entity that can talk to the sensors directly. Due to the lack of real end-to-end connectivity or interaction with the resource-constrained devices, flexibility of usage is reduced. Users can query the sensors only in the way that is allowed by the gateway. Adding new sensor resource often requires adaptation on the gateway. Another disadvantage is the vendor lock-in. Gateway and sensors often have to be from the same vendor in order to be compatible.

B. *Integration of sensors into the IP-world*

To address the networking needs of resource-constrained devices the IETF has formed several working groups: IPv6 over Low Power WPAN (6LoWPAN) [2], Routing Over Low Power and Lossy Networks (ROLL) [3], Constrained Restful Environments (CORE) [4], and Constrained Application Protocol (CoAP) [5].

Similar to the previous category of approaches, gateways are still used to translate between protocols used in the Internet and protocols used in the sensor networks, e.g. IPv6 to 6LoWPAN and vice versa (Fig. 2). However the difference here is that through the use of standard protocols, many of the disadvantages from the previous approaches are now taken care of. For example it is now possible to have the gateway and the sensors from different vendors.

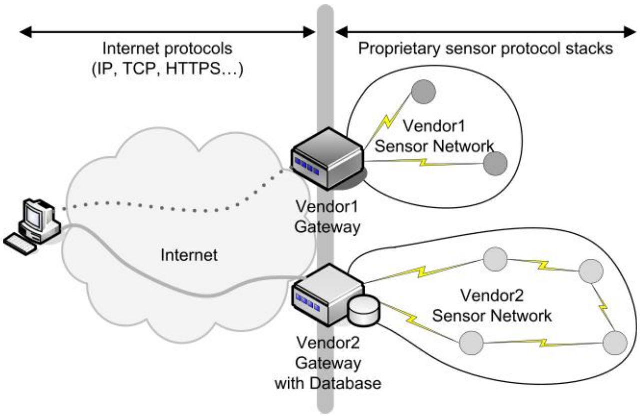


Figure 1. Gateways are often used to interconnect sensor networks to the Internet.

[1] http://www.zigbee.org/

[2] http://www.bacnet.org/

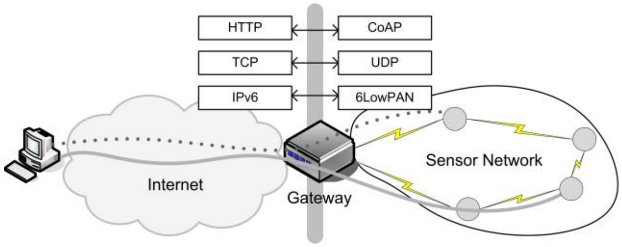


Figure 2. Internet protocols are extended to the sensor networks. The Gateway translates between the two protocol stacks.

Flexibility is also improved by this approach. Users can now query the sensors without the need for the gateway to understand the query and the data itself. The application payload can now travel directly from the client to the sensor, where it is processed and acted upon. The gateway takes care of the translation between standardized protocols. This makes adding and removing sensor resources transparent to the gateway and improves interoperability of devices.

Of course, by allowing direct interaction with the resource-constrained devices, new challenges related to access control, authentication, etc. are introduced, up to the level where the resource-constrained devices themselves need to manage access to the resources they offer. These security aspects are considered a major challenge in the IP-based Internet of Things [6]. In summary, both approaches have their advantages and disadvantages, characterized by the degree of openness in accessing the services on the resource-constrained devices. In the next section, we present a third, novel, complementary approach.

III. IoT-VN

In several use cases, there is no need to expose the data generated by resource-constrained devices to the whole world. Only a limited number of devices are involved, both resource-constrained and non-constrained, that need to cooperate in order to achieve a specific goal. Based on this observation, we propose adding a complementary approach to the approaches targeting such use cases. This approach was first introduced and discussed in [1] and named "managed ecosystem of networked objects" (MENO). It aims to realize a secured and confined environment in which all objects that need to cooperate can communicate in an end-to-end manner as shown in Fig. 3. This is achieved by creating a virtual network of all involved devices, including resource-constrained devices. In the remainder of this paper we refer to this virtual network as an Internet of Things Virtual Network or IoT-VN. An IoT-VN is established on top of different types of physical networks and consists of virtual links. A virtual link can be established between two devices connected to the Internet (on top of layer 3 IP connectivity), between two devices in a LAN (on top of layer 2 connectivity) or between two devices in a resource-constrained network such as a sensor network. Inside this virtual network, end-to-end communication between any two devices or groups of devices can take place through the use of existing protocols for communication in resource-constrained networks, through the use of proprietary protocols or through the use of novel protocols specifically designed for the targeted use case. Applications or services running inside the virtual network only see this logical layer. In a transparent way, all protocols running inside the virtual network communicate by transmitting their data over the virtual links. The logic for managing and establishing the virtual network automatically takes care of connecting the different members of the virtual network through the establishment of the virtual links and the secure transmission of data over these links. As such, security is inherently part of the design at the connectivity level already, by only allowing access to devices that are member of the virtual network. For more details about the concept, we refer to [1].

To the best of our knowledge, creating virtual networks among resource-constrained and non-constrained devices in the same setting has not received much attention in the literature so far. Still, it has been addressed a few times in the past years. According to the VITRO/FP7 project [7], Virtual Sensor Networking is an emergent approach which enables the dynamic collaboration of a subset of sensor nodes, not necessarily controlled or owned by the same Administrative Domain, aiming to complete a certain task or computation at a given time.

Reference [8] considers collaboration and resource sharing to be the main idea of Virtual Sensor Networks. To achieve this, nodes can be grouped into different Virtual Sensor Networks based on the phenomenon they track or the task they perform. Virtual Sensor Networks are expected to provide the protocol support for formation, usage, adaptation, and maintenance of subset of sensors collaborating on a specific task. Furthermore, Virtual Sensor Networks should make efficient use of intermediate nodes, networks, or other Virtual Sensor Networks to deliver messages across members of a Virtual Sensor Network. On the other hand, virtual networking is common in the Internet world such as VLAN [9], VPN [10], or VPAN [11]. Although network virtualization is frequently used to achieve secure communication over the unsecure Internet, the integration of resource-constrained devices is largely unexplored. End-to-End secure communication between IP-

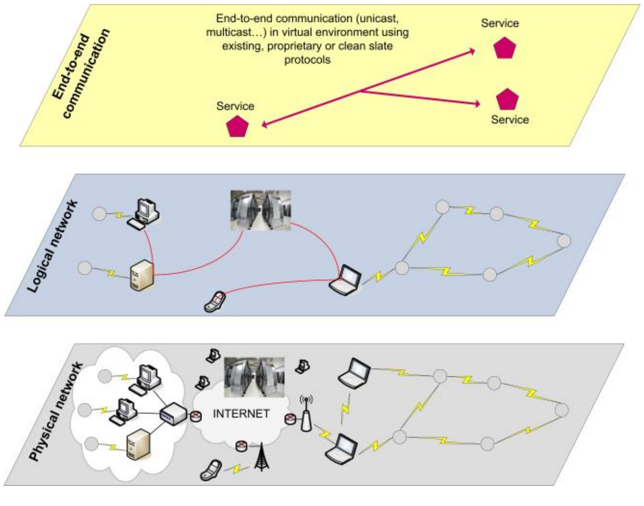

End-to-end communication
End-to-end communication (unicast, multicast...) in virtual environment using existing, proprietary or clean slate protocols
Service
Service
Service
Logical network
Physical network
INTERNET

Figure 3. MENO Concept – IoT-VN approach

enabled resource-constrained devices and the traditional Internet using IPsec was demonstrated in [12]. This approach provides secure communication on the network layer between end points and the technology presented there could be used to establish tunnels with resource-constrained devices, if complemented with appropriate solutions to establish trust between all IoT-VN members. However, the IoT-VN approach also aims to include layer 2 secure communication between neighboring devices and aims to realize a complete virtual network involving resource-constrained and non-constrained devices. Further, we do not limit ourselves to secure IP communication, but explore the possibilities of running protocols adapted to the limitations of resource-constrained devices inside the virtual environment. When considering the realization of trust, IoT-VNs can also benefit from ongoing research on bringing security solutions in reach of resource-constrained devices such as [13] and [14].

To the best of our knowledge, no other related work has taken our approach in providing a flexible virtualized network that contains both resource-constrained and non-constrained devices in the same virtual network.

IV. Example Use Cases

IoT-VNs can be beneficial in several use cases, when compared to the traditional integration approaches described in Section II. In this section we explore several generic use cases of IoT-VNs.

A. Partitioning a sensor network

The simplest application of IoT-VNs is the partitioning of a sensor network into two or more virtual networks. In Fig. 4 a virtual network is shown that only contains a subset of the available sensors in a sensor network. Secure communication is available only between the members of the virtual network. Other sensors of the sensor network may still be used to forward traffic between the IoT-VN members, however these sensors will not be able to interpret the data being forwarded.

This can be used for example in building management where sensor networks belonging to different administrators, owners, departments, etc. are deployed. These networks cooperate to enhance data forwarding, but all internal traffic for control, management and data collection is shielded.

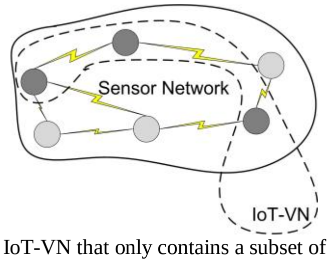


Figure 4. An IoT-VN that only contains a subset of the available sensors in a sensor network. Secure communication is available only between members of the virtual network.

B. Agregating multiple sensor networks

It can be required to connect two or more geographically separate sensor networks. In Fig. 5 an IoT-VN that combines multiple sensor networks into a single big virtual sensor network is shown. The two networks can be combined for example by creating a secure Layer-3 tunnel over the Internet between the two Edge Routers (ER) of the sensor networks. Secure communication is available between all the members of the virtual network, regardless of their physical location and connection to the network. Typical sensor network protocols running inside the IoT-VN only see a single large sensor network.

This can be used for example in case sensors or actuators in one sensor network should directly act upon measurements collected in another, not directly connected, sensor network.

C. Extending a sensor network with non-constrained devices

In many cases, it is required to have one or more non-constrained devices as part of the sensor network. These devices are used for example for storage or in order to perform calculations that are beyond the capabilities of the sensors. Fig. 6 shows an IoT-VN that is extended to include remote non-constrained devices.

For example, a server in the cloud can be used to directly gather information from all sensor nodes in the IoT-VN, aggregate that information, and present it to the Internet in a controlled and secure manner. The cloud server transparently acts as the sink of the network and can run the corresponding protocols.

D. A hybrid of sensor networks

Of course, all of the above scenarios can be combined together as needed to fit the needs of the network owner. In Fig. 7 an IoT-VN is shown, that is a hybrid of the previous scenarios. This IoT-VN is created by combining partitions from two separate sensor networks and extending them with remote non-constrained devices. It is clear that the number of possible configurations is almost unlimited and will be strongly dependent on the use case that needs to be realized.

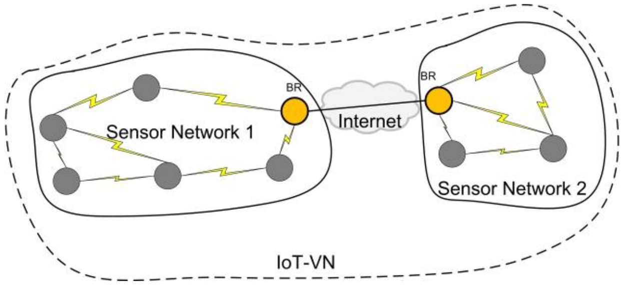


Figure 5. An IoT-VN that combines several sensor networks into one big virtual sensor network, by establishing a virtual link between the two border routers (BR). Secure communication is available between all members of the IoT-VN, regardless of their physical location.

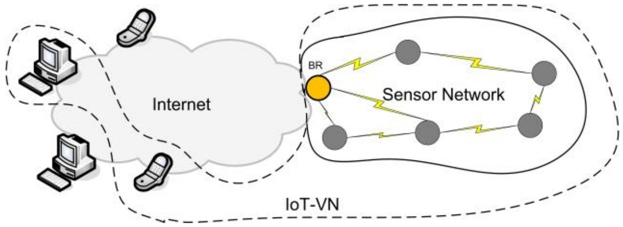


Figure 6. An IoT-VN that is extended to include non-constrained devices.

V. Implementation

The goal of our implementation is the establishment of an IoT-VN, a secure self-organizing virtual network environment. The virtual network is a collection of virtual links on top of which data can be exchanged and new protocols can be designed from scratch. It should be possible that devices of various capabilities are able to be part of the IoT-VN. To that end, two separate but interoperable implementations were developed. The first implementation is designed for non-constrained devices running typical operating systems such as Linux, Windows, OS-X, embedded Linux, etc. The second implementation targets resource-constrained devices such as sensors that run specific operating systems.

A. *Non-constrained implementation*

The non-constrained implementation has been realized in Click Router, a C++ based modular framework that can be used to realize any network packet processing functionality [15]. It consists of several modules that perform a specific task. The modules are combined together using a configuration file to obtain the overall functionality of the system.

As already mentioned, nodes in the IoT-VN can be thought of as being connected by virtual or logical links. These virtual links correspond to a path in the underlying network, perhaps through multiple physical links. For non-constrained devices these virtual link are established either over available layer 3 IP connectivity (e.g. over the Internet) or directly over available layer 2 LAN connectivity (when in the same broadcast domain). Since not every device is allowed to participate in the IoT-VN, a mechanism is needed to identify the members of the IoT-VN. Therefore all devices participating in the IoT-VN share a common cryptographic trust relationship consisting of a public and private key pair, signed by a common IoT-VN private key maintained by a certification authority (implemented using OpenSSL).

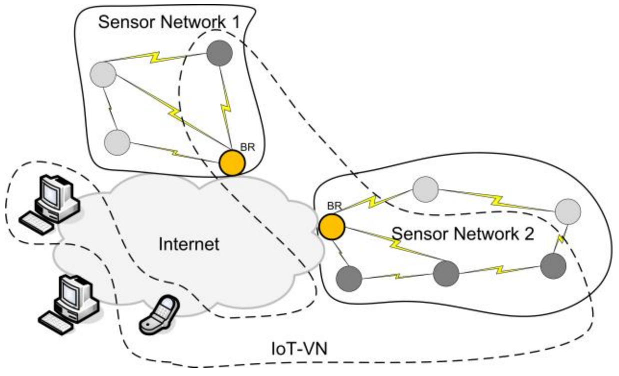


Figure 7. A hybrid IoT-VN.

When members are in the same broadcast domain (e.g. Ethernet, Wi-Fi), they can discover each other by periodically sending beacon packets. Upon the reception of such a beacon, a challenge-response mechanism is initiated to authenticate each other and negotiate a symmetric session key resulting in a secure virtual link between both nodes.

When members are connected to the Internet (via cable, Wi-Fi, UMTS, GPRS, etc.) they can register with a trusted agent. This information is then exchanged with already registered members and subsequently used to establish tunnels between the different members. The implementation includes NA(P)T detection, hole punching and relaying via the trusted agent in order to deal with NA(P)T boxes.

Using these neighbor detection and tunneling mechanisms, layer 2 and layer 3 virtual links can be established as illustrated in Fig. 8. A Virtual Link Manager module manages all virtual links. Together with a Virtual Link Forwarder module, it is possible to send any data over a single virtual link or over multiple virtual links. Also, upon reception of data over a virtual link, the virtual link over which the data arrived is identified. Using this basic functionality it is possible to deploy any protocol on top of the established virtual links.

Further some modules are present to maintain and manage the IoT-VN and to deal with all used network interfaces and changes in their properties. Part of this implementation is based on the VPAN implementation described in [11], but was thoroughly modified in order to become IP agnostic.

B. *Extension of the IoT-VN concept to resource-constrained devices*

The extension of IoT-VNs to sensors consists of three parts, which will be further described in this section: an extension of the non-constrained implementation to use a physical network interface to the sensor network (802.15.4 network interface), an implementation of the IoT-VN concept on resource-constrained devices and new modules for the non-constrained implementation for establishing virtual links with a resource-constrained devices over the 802.15.4 interface.

Although the non-constrained implementation mentioned in section A runs well on devices down to the level of a

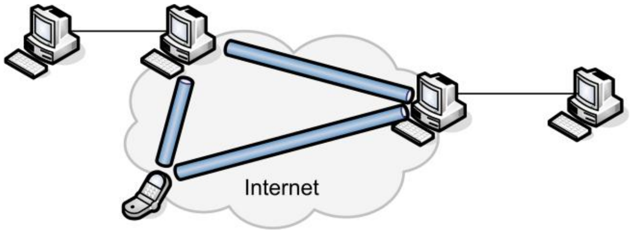


Figure 8. An IoT-VN that contains layer 2 (same broadcast domain) and layer 3 (IP connectivity over Internet) virtual links.

smart objects with embedded Linux, this implementation has a footprint that is way beyond the capabilities of typical resource-constrained devices (e.g., 48K bytes of ROM, 8K bytes of RAM). In order to have an implementation that would run on such devices, customized operating systems and development environment should be used. These environments try to optimize the footprint of the generated code as much as possible, in order for it to fit on the limited resource of today's typical resource-constrained devices.

Our resource-constrained devices implementation of the IoT-VN has been realized using the IDRA framework [16]. IDRA is a network architecture and application platform written in nesC and developed for TinyOS [17] - an event-driven operating system designed for sensor network nodes that have very limited resources.

This implementation uses a similar design and the same packet format (e.g. beacon format) as the non-constrained implementation and is thus compatible with it. However, this implementation does not include all the features that are supported in non constrained implementation (e.g. data encryption).

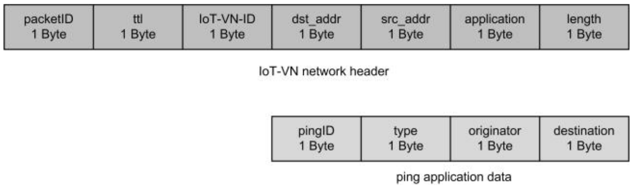
In addition to the previous described two types of virtual links, a third type has been introduced. This type is used whenever at least one node of the virtual link nodes is a resource-constrained device. In this way it becomes possible for a non-constrained device to adopt its communication to the limitations imposed be the resource-constrained device. For example it would be possible to use a more light-weight (but maybe a weaker) encryption algorithm in the communication between a resource-constrained and a non-constrained device.

C. Protocols inside the IoT-VN

Inside the IoT-VN it is possible to run either standard sensor protocols or to design and run completely new protocols that fit the needs of the particular IoT-VN. In order to demonstrate both cases we have implemented two simple protocols: Ad hoc On-Demand Distance Vector (AODV) Routing [18] and a PING application.

1) AODV

To enable multi-hop communication between nodes, we implemented the basic functionality of the existing AODV protocol. To achieve this, every member of the IoT-VN is assigned a unique 1-byte address. Also a proprietary network header as shown in the top part of Fig. 9 is defined. For AODV routing messages, the AODV header is appended to this network header. When one member, the source, wants to transmit data to another member, the destination, and no route has been established yet, a route request is broadcasted inside the IoT-VN. The source sends the route request over all established virtual links using the virtual link broadcasting functionality. All receiving nodes repeat this process until the destination is reached. This establishes a reverse path to the source node, specifying the next hop 1-byte address and the virtual link over which the data needs to be sent to reach the next hop towards the source. Upon reception of the route request by the destination, the destination will send back a route reply using the backward path to the source. While the reply travels from the destination to the source, a forward path is established. This completes the establishment of the route.



packetID 1 Byte	ttl 1 Byte	IoT-VN-ID 1 Byte	dst_addr 1 Byte	src_addr 1 Byte	application 1 Byte	length 1 Byte

IoT-VN network header

pingID 1 Byte	type 1 Byte	originator 1 Byte	destination 1 Byte

ping application data

Figure 9. Top: IoT-VN Network header format. Bottom: ping application data format.

2) Ping

A simple ping application has been implemented, consisting of a ping request and a ping reply. The header of this proprietary ping protocol is shown at the bottom part of Fig. 9. This header is appended to the network header, with the application field set to a value that is assigned to the ping protocol. The ping packets are forwarded using the routes established by AODV.

Using AODV and Ping it is possible to demonstrate communication between any two members inside the IoT-VN. A well-known protocol, AODV, is used, but complemented with other things such as the 1-byte address, network header, etc. which are only known inside this IoT-VN. Together it realizes end-to-end communication capabilities inside the IoT-VN, capable of also running on resource-constrained devices.

VI. Results

In order to test our resource-constrained and non-constrained implementations and the interoperability between these two implementations, we built a small network that consists of two sensors and two PCs as shown in Fig. 10. Each of the sensors was connected to one PC wirelessly using IEEE 802.15.4. The two PCs were connected together via an Ethernet link. All four devices were manually configured to be part of the same IoT-VN – identified by ID AA.

When the applications run on the devices, neighboring devices discover each other and automatically create virtual links between them. In this simple case the three created virtual links mapped directly to the respective physical links between the devices. The link between the two PCs was negotiated to be a secure link with 128-bit AES encryption. However the links between the PCs and the sensors were not encrypted, since the sensor implementation does not support encryption yet.

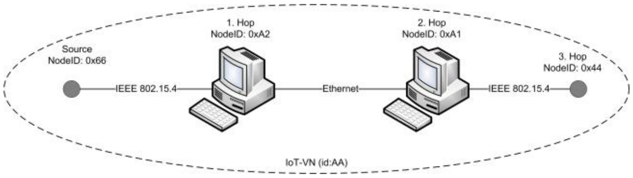


Figure 10. The network that was used to test the IoT-VN implementation on PCs running Linux and on sensors.

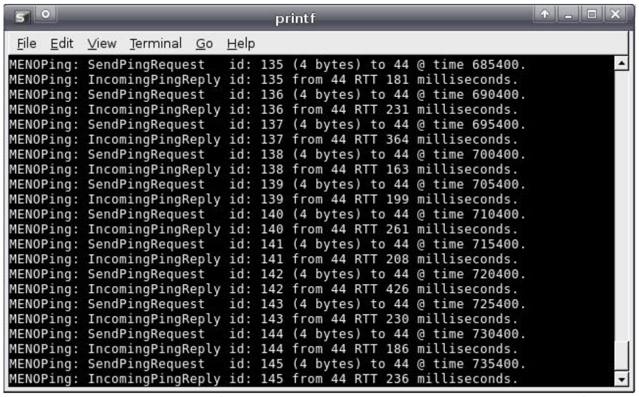
Several ping tests were conducted between the devices to verify the connectivity between the devices. Fig. 11 shows a screenshot of the output of the ping application on the USB interface of the sensor that started the ping test. As one can see in this screen shot, the addresses used by the ping application is the virtual node addresses (nodeID). In fact, the ping application, does not know, whether the destination is another resource-constrained or a non-constrained device. All that is needed for the application is the nodeID of the destination.

In the following table the average round trip times are shown along with the standard deviation for a sample of 100 pings between a sensor at one end of the network and the other devices in the network. It is worth noticing, that the largest time in the path is the time to reach the first hop and get the reply from it (125ms). This is due to the fact, that the sender is a resource-constrained device and it had a few debug options turned on in order to see what is going on in the test. The next part along the path added 29ms to the round trip time. This time is less than one fourth of the time for the first part, although the packet in this part needed more processing power since it was encrypted and decrypted. The reason for this is that this part is between two PCs, which of course have more processing power than the sensors. The third part of the path added 62ms to the round trip time. Although this part of the network is very similar to the first part, the round trip time is about half the time of the first part along the path. The main reason for this reduced time is that this second sensor had all debug options turned off in contrast to the first sensor.

Table 1. Ping round trip times in millisecond.

Hop Count	1	2	3
Round Trip Time (millisecond)	125	154	216
Standard Deviation	75	92	98

The ping tests demonstrated, that it is possible to communicate between two sensors, that belong to different sensor networks, as long as the sensors belong to the same IoT-VN. Furthermore it demonstrated that end-to-end communication between all devices inside an IoT-VN, regardless whether resource-constrained or not, is carried out without any protocol translation.



```
MENOPing: SendPingRequest   id: 135 (4 bytes) to 44 @ time 685400.
MENOPing: IncomingPingReply id: 135 from 44 RTT 181 milliseconds.
MENOPing: SendPingRequest   id: 136 (4 bytes) to 44 @ time 690400.
MENOPing: IncomingPingReply id: 136 from 44 RTT 231 milliseconds.
MENOPing: SendPingRequest   id: 137 (4 bytes) to 44 @ time 695400.
MENOPing: IncomingPingReply id: 137 from 44 RTT 364 milliseconds.
MENOPing: SendPingRequest   id: 138 (4 bytes) to 44 @ time 700400.
MENOPing: IncomingPingReply id: 138 from 44 RTT 163 milliseconds.
MENOPing: SendPingRequest   id: 139 (4 bytes) to 44 @ time 705400.
MENOPing: IncomingPingReply id: 139 from 44 RTT 199 milliseconds.
MENOPing: SendPingRequest   id: 140 (4 bytes) to 44 @ time 710400.
MENOPing: IncomingPingReply id: 140 from 44 RTT 261 milliseconds.
MENOPing: SendPingRequest   id: 141 (4 bytes) to 44 @ time 715400.
MENOPing: IncomingPingReply id: 141 from 44 RTT 208 milliseconds.
MENOPing: SendPingRequest   id: 142 (4 bytes) to 44 @ time 720400.
MENOPing: IncomingPingReply id: 142 from 44 RTT 426 milliseconds.
MENOPing: SendPingRequest   id: 143 (4 bytes) to 44 @ time 725400.
MENOPing: IncomingPingReply id: 143 from 44 RTT 230 milliseconds.
MENOPing: SendPingRequest   id: 144 (4 bytes) to 44 @ time 730400.
MENOPing: IncomingPingReply id: 144 from 44 RTT 186 milliseconds.
MENOPing: SendPingRequest   id: 145 (4 bytes) to 44 @ time 735400.
MENOPing: IncomingPingReply id: 145 from 44 RTT 236 milliseconds.
```

Figure 11. Screenshot from the ping application showing successful ping between two sensor that belong to the same IoT-VN. The address of the destination is the nodeID in the virtual network. The output was taken by connecting to the USB interface of the sending sensor.

Fig. 12 illustrates the packet flow of the ping packet as it travels its path from the source of the ping to its destination over the virtual network. The ping packet is prepended with a IoT-VN network header to facilitate routing of the packet in the virtual network. The sending sensor also prepends the 802.15.4 header to enable the packet to travel over the sensor networks. When the first pc receives the packet it strips the 802.15.4 header and sends it to the IoT-VN application to decide on the next hop in the virtual network. Since in this case the next hop is the second PC, the data should be sent encrypted over the link. The PC calculates the encrypted data and adds the necessary security headers and trailers and prepends them with the Ethernet header in order to send the packet over Ethernet. The packet continues its way along its path following the same rules.

The implemented sensor application (including MAC, AODV routing, IoT-VN functionality and ping) has a footprint of 35508 bytes in ROM and 4981 bytes in RAM. This footprint fits even on very resource-constrained devices.

The test results and the small footprint of the implementation demonstrate that it is feasible to realize our approach in including resource-constrained and non-

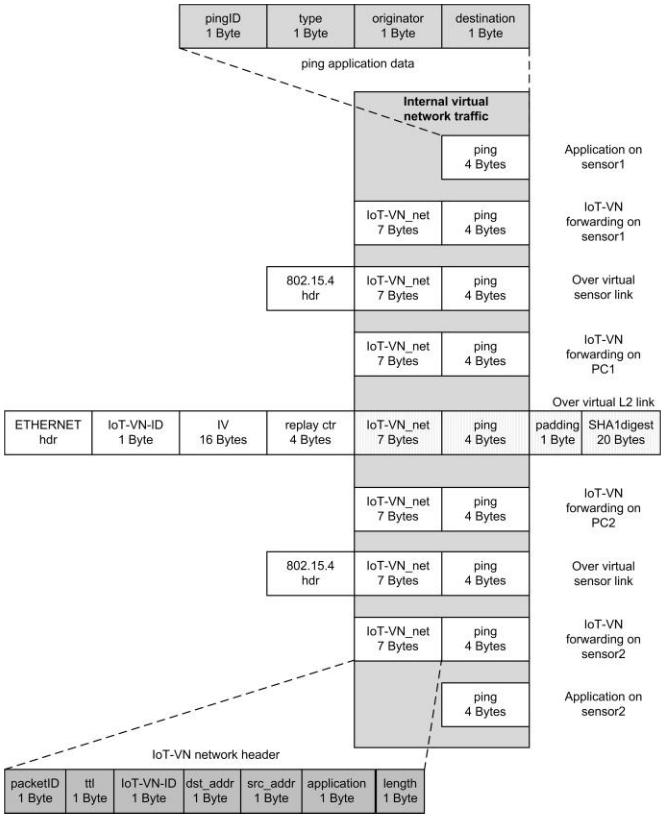


Figure 12. A ping packet traveling over a virtual network. The ping packet is prepended with a IoT-VN network header to facilitate routing of the packet in the virtual network. When traveling over sensor networks the 802.15.4 header is prepended. In addition to the Ethernet header security headers and trailers are added before sending the packet over Ethernet.

constrained devices in one virtual network. However, the implementation is still at an early stage, mainly demonstrating the concept, its feasibility and possible applications. In order to really allow an in-depth evaluation of the IoT-VN concept, several hurdles need to be overcome. For example, the current solution requires a non-constrained device at the border of the network and does not foresee end-to-end tunneling as used in [11]. At this stage, security mechanisms have not been applied yet and will definitely increase the footprint. The addition of security in the near future, based on available research works, will enable us to tackle other related problems such as key distribution, trust establishment and overlay management and would allow more thorough conclusions about the realization of the IoT-VN concept. Another limitation is that the evaluation is based on a very small test bed. This only allows to demonstrate the feasibility, not however to evaluate the performance of the approach in real settings. For performance evaluation a bigger testbed has to be used.

VII. Conclusions and outlook

In this paper we have introduced our approach to complement existing methods of integrating sensor networks into the Internet. Our approach focuses on the objects that need to cooperate by integrating them into a secured virtual network. Inside this virtual network full end-to-end communication can take place between the networked objects regardless whether they are resource-constrained or not. This is achieved through the use of protocols that take into account the limitations of the most resource-constrained devices. We described how this concept can constitute a valid alternative approach for realizing certain real-life scenarios by providing some several generic use cases. Finally we described our first implementation demonstrating the key concepts of this approach. It proved the feasibility of our approach, but also revealed remaining challenges.

Security is an essential part of our approach. While an acceptable level of security was achieved in the communication between non-constrained nodes in our virtual network, secure communication between resource-constrained devices and between resource-constrained and non-constrained devices remains a challenge. In the future, we plan to implement secure communication between sensors and between sensors and non-constrained devices by using encryption.

The scalability of our approach has not been tested yet. We are planning to test it in a larger-scale, real-life environment by using a wireless sensor testbed, such as the w-iLab.t [19].

Additionally, we plan to investigate to what extend any manual configuration that is still required to create and manage the virtual network can be avoided.

Acknowledgment

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n°258885 (SPITFIRE project), from the IBBT ICON project GreenWeCan, and a VLIR PhD scholarship to Isam Ishaq.

References

[1] J. Hoebeke, E. De Poorter, St. Bouckaert, I. Moerman, and P. Demeester, “Managed Ecosystems of Networked Objects,” Wireless Personal Communications 58 (1), pp. 125-143, May 2011.

[2] N. Kushalnagar, G. Montenegro, and C. Schumacher, “IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals,” IETF RFC 4919, August 2007.

[3] Routing Over Low power and Lossy networks (roll), http://datatracker.ietf.org/wg/roll/

[4] Z. Shelby, “Constrained RESTful Environments (CoRE) Link Format,” IETF RFC 6690, August 2012.

[5] Z. Shelby, K. Hartke, C. Bormann, and B. Frank, “Constrained Application Protocol (CoAP),” draft-ietf-core-coap-08, work in progress, http://tools.ietf.org/html/draft-ietf-core-coap-08, November 2011.

[6] O. Garcia-Morchon, S. Keoh, S. Kumar, R. Hummen, and R. Struik, “Security Considerations in the IP-based Internet of Things,” draft-garcia-core-security-04, work in progress, http://tools.ietf.org/html/draft-garcia-core-security-04, March 2012.

[7] The VITRO/FP7 project, Web http://www.vitro-fp7.eu/, April 2012.

[8] A. P. Jayasumana, Q. Han, and T. Illangasekare, "Virtual sensor networks – a resource efficient approach for concurrent applications," Proc. International Conference on Information Technology (ITNG 2007), pp. 111 – 115, Las Vegas, Apr. 2007.

[9] J. Freeman and D. Passmore, “The Virtual LAN Technology Report”, Decisys, Inc., Sterling, VA, 1996.

[10] S. Khanvilkar and A. Khokhar, “Virtual Private Networks: An Overview with Performance Evaluation”, IEEE Communication magazine, Vol. 42(10), pp. 146-154, 2004.

[11] J. Hoebeke, “Adaptive Ad Hoc Routing and Its Application to Virtual Private Ad Hoc Networks”, Phd, ISBN 978-90-8578-172-1, NUR 986, 2007.

[12] S. Raza, S. Duquennoy, T. Chung, D. Yazar, T. Voigt and U. Roedig, “Securing Communication in 6LoWPAN with Compressed IPsec,” Proc. 7th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS '11), pp. 1-8, Barcelona, Spain, 27-29 June 2011.

[13] A. Liu, P. Ning, “TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks,” Proc. 7th International Conference on Information Processing in Sensor Networks (IPSN 2008), SPOTS Track, pages 245-256, April 2008.

[14] “Security of TinySec”, Web http://www.cl.cam.ac.uk/research/security/sensornets/tinysec/, Sep. 2012.

[15] E. Kohler, R. Morris, B. Chen, J. Jannotti and M. F. Kaashoek. “The Click modular router,” ACM Transactions on Computer Systems 18(3), pp. 263-297, 2000.

[16] E. De Poorter, E. Troubleyn, I. Moerman and P. Demeester, “IDRA: a Flexible System Architecture for Next-Generation Wireless Sensor Networks,” Wireless Networks, 17(6), pp. 1423-1440, 2011.

[17] TinyOS, Web http://www.tinyos.net/, April 2012.

[18] C. Perkins, E. Belding-Royer, and S. Das, Ad hoc On-Demand Distance Vector (AODV) Routing, IETF RFC 3561, July 2003.

[19] S. Bouckaert, W. Vandenberghe, B. Jooris, I. Moerman, and P. Demeester, “The w-iLab.t testbed,” Proc. of the International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom '10), pp. 145–154, Berlin, Germany, May 2010.