ON COHABITATING NETWORKING TECHNOLOGIES WITH COMMON WIRELESS ACCESS FOR HOME AUTOMATION SYSTEM PURPOSES

JORDI MONGAY BATALLA, GEORGE MASTORAKIS, CONSTANDINOS X. MAVROMOUSTAKIS, AND JERZY ŻUREK

ABSTRACT

An increasing number of home automation systems using wireless devices compete for the radio access in the same space and time. Lately, a stressing trend consists of aggregating home automation systems to save energy consumption, while at the same time avoiding wireless interference. This article proposes virtualization, open software deployment, and separation of radio and higher layers as the response to the increasing expandability of home automation systems combined with the increasing number of technologies for connecting wireless devices. A system has been developed, containing three different technologies: ZigBee, Idsecom, and 6LoWPAN simultaneously working over a virtualization platform with access to a common antenna. The evaluation tests performed on the system validate the solution and separately show the performance capacity of virtualization platform, software (ZigBee, Idsecom and 6LoWPAN) nodes, and 802.15.4 wireless antennas.

INTRODUCTION

The atomicity of the Internet of Things (IoT) has arisen to handle energy efficiency and adaptability requirements, as well as interoperability, scalability, and extendibility challenges. In home automation systems (HASs), this atomicity has brought a high number of gateways permanently connected to the electrical current and competing for the radio access. The competition will dramatically rise with the introduction of Thread, which is an incoming technology for HAS applications based on the 6LoWPAN protocol created by outstanding technological companies coordinated by Google Inc. Thread is called to compete with ZigBee and Z-Ware (popular in the United States), but at the same time, they will coexist together in many spaces, creating cross-technology interference. The cross-technology interference in the so-called unlicensed spectrum range is an old research problem, which is only partially solved thanks to, among others, dynamic channel selection. A good survey for understanding the problem of cohabitation of different technologies was presented by Yang, Xu, and Gidlund in [1]. The current trend in IoT centers is associated with integrated scenarios and systems toward creating steady platforms that are capable to synchronize different applications. At the sensor level, integration of existing solutions may avoid interference in the radio, as well as increasing simplicity by reducing the number of sensors fulfilling similar operations. Several solutions have been deployed to integrate different IoT technologies in HASs. Wibutler [2] joins together ZigBee, Enocean, Z-wave, and Bluetooth. The drawback of such a solution is the limitations of the functionalities to be performed by Wibutler, since it operates with an evolving closed source application. Other systems such as Connected Home Hub mydlink Home [3] and Samsung's Comcast Xfinity [4] work with a reduced number of devices (of several technologies), which are well known by the developers of the platforms.

The current integrative platforms present closed solutions with applications hardly extendable (by third parties), which creates utilization boundaries, as also pointed out in [5]. Different investigations try to break the closed approach of HAS solutions at different levels. An interesting approach is RIOT [6], which is an open source operating system that should be the base for all the IoT technologies and applications. A similar approach is UBOS [7], a Linux (open source) "distro" that runs on IoT-specific gateways for running users' personal services, without the necessity for commercial wizards and applications. The major issue of the aforementioned approaches is the necessity of adapting the existing commercial solutions, which seems an unattainable task. In this article, different technologies are proposed to be integrated into a programmable virtualized platform. All the technologies are developed as virtual machines (VMs) in the platform, and they share computing resources as well as radio access (one unique 802.15.4 antenna for all the technologies). This

Jordi Mongay Batalla and Jerzy Żurek are with the National Institute of Telecommunications.

Jordi Mongay Batalla is also with Warsaw University of Technology.

George Mastorakis is with the Technological Educational Institute of Crete.

Constandinos X. Mavromoustakis is with the University of Nicosia.

1536-1284/16/$25.00 © 2016 IEEE

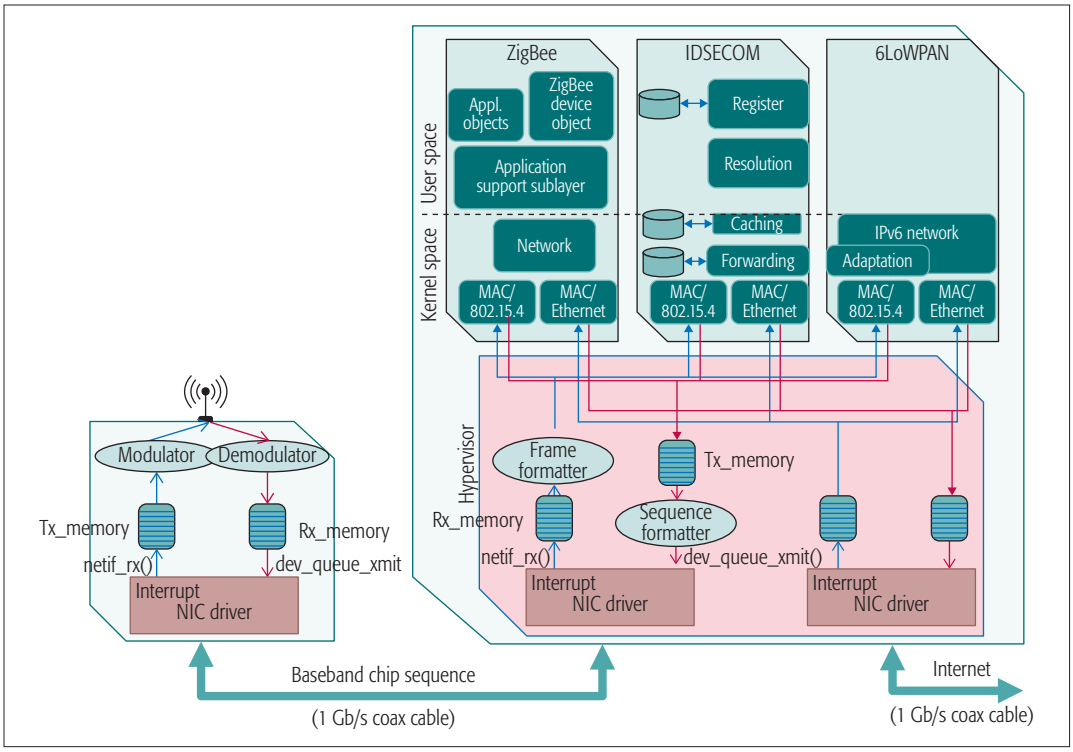

Figure 1. Integrated platform: block diagram.

way, many small devices used in HASs for communication with the sensors may be supplanted by one unique device with one antenna. The proposed system provides the following main benefits:

- It is fully programmable and may easily be extendable to new technologies, thanks to virtualization.
- It has reduced costs thanks to the simplicity of the hardware and "softwarization" of most of the functionalities.
- It avoids interference between different technologies' sensors.

Architecture of the Software-Defined IoT Platform

The integrated platform for the HAS should consider the continuous apparition of new technologies, which are often scenario-specific, as well as the limits of available radio spectrum. Therefore, we propose to clearly separate higher-layer functionalities (performing from routing to IoT gateway functions) from radio access (which will be common for all technologies). The desegregation of technologies suggests the virtualization of technology-dependent functionalities, while offering openness to the IoT development. Instead, the radio access is shared between all the technologies, avoiding rivalry based on power consumption. The proposed solution significantly increases energy saving, by reducing the amount of hardware being connected, especially by reducing the number of different gateways (which, in general, are continuously connected to the electric current) operating within the same location. Figure 1 shows the block diagram of the platform integrating three different IoT technologies: ZigBee, Idsecom, and 6LoWPAN. Each technology is developed in a separate VM, gaining flexibility to add new technologies to the platform. Moreover, new functionalities can easily be added to each technology, integrating any of the layers (from routing/forwarding of packets to advanced IoT-specific operations, e.g., registration and discovery) to adapt to specific use case scenarios.

The virtualization platform has an important influence on the efficiency of the system, understood as the capability of forwarding messages and completing IoT operations. The election of the virtualization platform and its influence on forwarding performance are analyzed in the next section.

The ZigBee node (ZigBee open source stack) integrated in our platform is a coordinator node, with full functions at the medium access control (MAC) level for constructing the topology, capable of building one wireless personal area network (WPAN, with its own PAN ID) and selecting the best channel (in 802.15.4 radio) to transmit. These functionalities are performed in the MAC layer and are common for the three nodes implemented in the platform (MAC/802.15.4 module in Fig. 1). By implementing MAC functionalities inside the nodes, these may decide the setup of a WPAN. The ZigBee node may contain a maximum of 240 application objects, containing a number of functionalities pertaining to one or more clusters, as required by the ZigBee standard. In the case when one scenario in a HAS needs more than 240 objects, it is assumed that two different ZigBee nodes (two VMs) will be implemented in the platform. The Idsecom technology is a novel approach specially used in business and home buildings [8] driven by forwarding the messages based on the hierarchized IoT identifiers. The system is capa-

Each technology is developed in a separate virtual machine, gaining flexibility to add new technologies to the platform. Moreover, new functionalities can be easily added to each technology, integrating any of the layers (from routing/forwarding of packets until advanced IoT-specific operations as, for example, registration and discovery) to adapt to specific use-case scenarios.

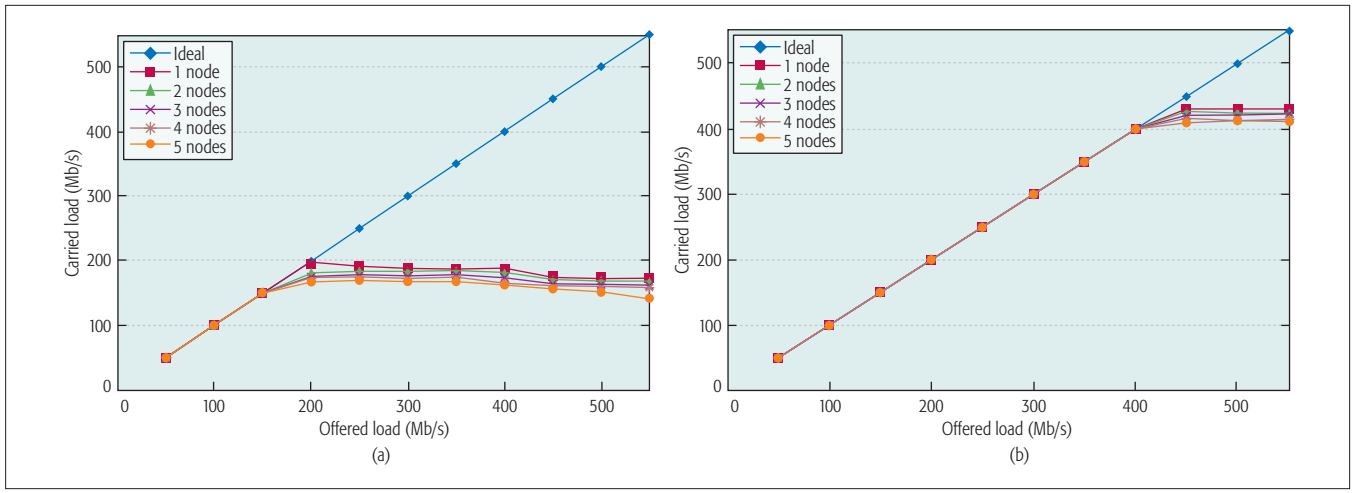

Figure 2. Carried load vs. offered load (127 bytes) for 6LoWPAN node deployed over a) XEN; b) LXC tools.

ble of integrating the addressing of objects and services, providing identifier/locator separation with one unique addressing structure, distributing and facilitating registration and publication of objects/services, and caching IoT information in the network for energy saving. The functioning of this node is shown in the following sections together with analysis of the forwarding performance. At last, the 6LoWPAN node has the functionalities of a 6LoWPAN edge router. Unlike the other nodes, this is application-agnostic, that is, it does not perform any operation above the network layer. The adaptation layer performs IPv6-to-6LoWPAN address conversion for UDP datagrams.

Radio access is common to all the technologies and is based on the IEEE 802.15.4 standard. The idea is to bring all the complexity of the radio access to the platform, while keeping the antenna as simple as possible. The communication between the platform (containing the nodes) and the antenna uses low-bandwidth coax cable. The platform is in charge of preparing the baseband chip sequence to be modulated. For this scope, the sequence formatter maps each frame to the correspondent baseband chip sequence, and the frame formatter performs the opposite operation. The antenna uniquely shapes the sequence by the half sine wave and propagates into the air. This approach makes the cohabitation of all the technologies in the same space possible, sharing the unlicensed spectrum.

The performance of the system is limited by the virtualization platform, the functioning of the three nodes, and the capacity of the antenna to propagate the frames. In the following sections, we provide details on the virtualization tool, nodes, and radio access, discussing implementation issues related to them and showing measurements of the performance of these three parts of the system.

Virtualized Platform

The virtualization environment is necessary for scalability purposes where new technologies may be added seamlessly onto the platform. The main inconvenience of virtualization is the limitations of performance (e.g., diminution of throughput or increase of packet losses in highly loaded systems) due to the overhead introduced by the virtualization tool. This overhead is considered necessary for efficiently managing the available resources between the network devices that are sharing the same node (real hardware). The limitations consist of a major barrier, especially while operations requiring very short timescales are taking place (i.e., the operations performed in the data plane [forwarding of packets/ frames]). To ensure openness and software-defined capabilities, we used a software open source virtualization environment. The software virtualization tool should allow the physical resources of the platform to be shared in such a way that each node may make use of the interface with the radio access, as well as the interface to the Internet. Two main techniques are used in software virtualization: hardware level and operating system (OS) level. The difference between them lies in the placement of the virtualization layer within the device: on top of the hardware layer or above the host OS. In hardware-level virtualization, each virtual router uses its own operating system kernel, offering advanced isolation to the VM. The nonvirtualizable instructions of the guest kernel are saved by calling the hypervisor (in the case of paravirtualization, e.g., XEN platform) or extending hardware functionalities for intercepting such nonvirtualizable instructions (in the case of full virtualization).

For this purpose, OS-level virtualization methods use the common kernel of the host OS, which has its system calls modified to allow multiple isolated user spaces to run multiple application instances. The host kernel is responsible for ensuring isolation (where this is feasible) between the application instances. The instances use the normal system call interface, which results in a reduction of the necessary overhead for managing virtualization. The virtualization method has a great influence on the I/O devices' performance. At the hardware level, access to the hardware is limited to the host kernel, which receives requests from the virtual network interface controller (NIC) driver in the guest kernel. Furthermore, data must be copied between guest and host kernels, increasing the operational time of forwarding the packet. In the opposite case, OS-level virtualization moves virtual interfac-

es into the VM. Then the kernel is responsible for keeping track of the owner of each interface, which results in no additional data copying requirements. In conclusion, OS-level methods save overhead in terms of both use of resources and time of operation, but lose flexibility as all the VMs must communicate with the same kernel, which often means the installation of the same OS in all the VMs. Flexibility is a crucial implementation parameter, since it limits the likelihood of adding other nodes, which have been independently developed, to the platform. In our platform, access to a NIC connected to a media antenna plays an important role since the baseband sequence chip stream sent by this interface has a high bit rate. Therefore, we investigated the effect of the virtualization methods on the performance of our platform.

For this purpose, we performed forwarding performance tests by installing two virtualization tools: Xen (hardware-level virtualization) and Linux LXC (OS-level virtualization), and over them we implemented from one to five identical 6LoWPAN software nodes with two MAC/ Ethernet interfaces. The tester (Spirent TestCenter equipped with CM-1G-D4 card) and the platform were connected by two 1 Gb/s Ethernet links in ring topology. The tester generated IPv6 127-byte packets (UDP datagrams). The selection of 127-byte packet length comes from the fact that 802.15.4 radio maximum transmission unit (MTU) is 127 bytes, so the nodes will work with this range of packet size. The generated packets were tagged with one of five different VLAN tags, which are used in the platform to distinguish the node that should process the packet (at the MAC/ Ethernet layer). We set up the nodes to forward IPv6 traffic arriving from one interface to the second after querying the routing table and adapting the IPv6 address to the 802.15.4 address format. The platform was developed over Wanboard with a Freescale i.MX6 Quad processor and attached hard disk drives — SATA II, supplementary Ethernet card IEEE 1588, and extended memory. The Xen version was 4.1.2 with host kernel 2.6.56. This version is a little older than current ones; however, it is lighter than current versions, which is suitable for our hardware. The Linux LXC installed was the stable 1.0 version. Figure 2 presents the traffic forwarded back to the tester (carried load) for increasing load of offered traffic. The results present the scenarios when 1–5 6LoWPAN nodes are implemented within the platform. The throughput of the platform can be obtained from the figure as the highest load value where carried load and offered load coincide in each test. The tests were performed 8 times in order to obtain confidence intervals, which resulted in lower than 20 percent of the mean values at the 95 percent confidence level in all the tests. For clarity purposes, we do not show the confidence intervals in the figure.

The results in Fig. 2 confirm that the OS-level virtualization method (Linux LXC) is more effective than hardware-level virtualization. For the studied packet length, LXC overcomes almost three times as many XEN forwarding functionalities. Moreover, it is important to observe that the performance decreases for a higher number of software nodes competing for the resources. This means that the platform may take in a maximum number of nodes due to the increase of overhead into the virtualization system. The number of nodes that can operate within the platform depends on the hardware selected and the virtualization tool. However, the throughput of 5 nodes is higher than 150 Mb/s, which is several orders of magnitude higher than normal IoT traffic load. Even if XEN allows for lower forwarding performance in the nodes, we select this platform for deploying a number of different technologies (ZigBee, Idsecom, 6LoWPAN), since XEN allows implementation of the own kernel space in each node, which offers a high deployment flexibility, as indicated above. In general, the modules over the network layer (i.e., application layer) have been implemented in the user space since they do not require a very fast reaction, whereas the network and MAC have been deployed in the kernel because the operations performed in these layers require very short timescales (Fig. 1).

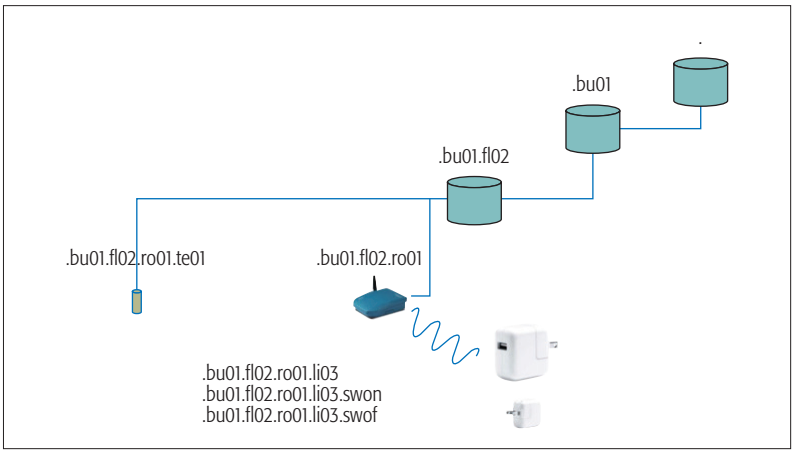

Figure 3. Exemplary network topology of an ID-layer-capable system.

Idsecom Node: Exemplary Clean Slate Technology

In our implementation, we developed a novel node (called the Idsecom[1] node) that is capable of forwarding Idsecom frames based on IoT identifiers instead of network addresses. The implementation of such a forwarder is an example of the possibilities of our platform when integrating new solutions and technologies. An IoT identifier is a label describing the location and name of an object together with the name of the required service, as presented in Fig. 3. An extended explanation of this concept was provided in [8]. The Idsecom node fits the intelligent buildings topology well, where sensors/actuators are located in areas of the building following an established hierarchy. The node uses this topology for addressing the frames, so service composition has no need to translate identifiers to network addresses. Other characteristics of the presented solution is that the identifier of the service is integrated into the network address, avoiding the necessity of an extra layer (in ZigBee this functionality is performed by the application objects layer) and the possibility of introducing caching of IoT information into the nodes, which increas-

[1] Idsecom is for Identifier-Based Secure Communication.

es energy saving in the objects.

The address of each network node, object, or service is formed as the concatenation of all labels beginning from the root node separated by a full stop character (Fig. 3). All nodes form a tree topology with eight levels, including end nodes and services offered by them (enough in HAS). Each level is addressed by four ASCII characters. For example, in Fig. 3, one light service has addresses .bu01.fl02.ro01.li03.swon corresponding to root_building01_floor02_room01_light03_switch on. The symbol "*" is reserved and used for broadcast addressing.

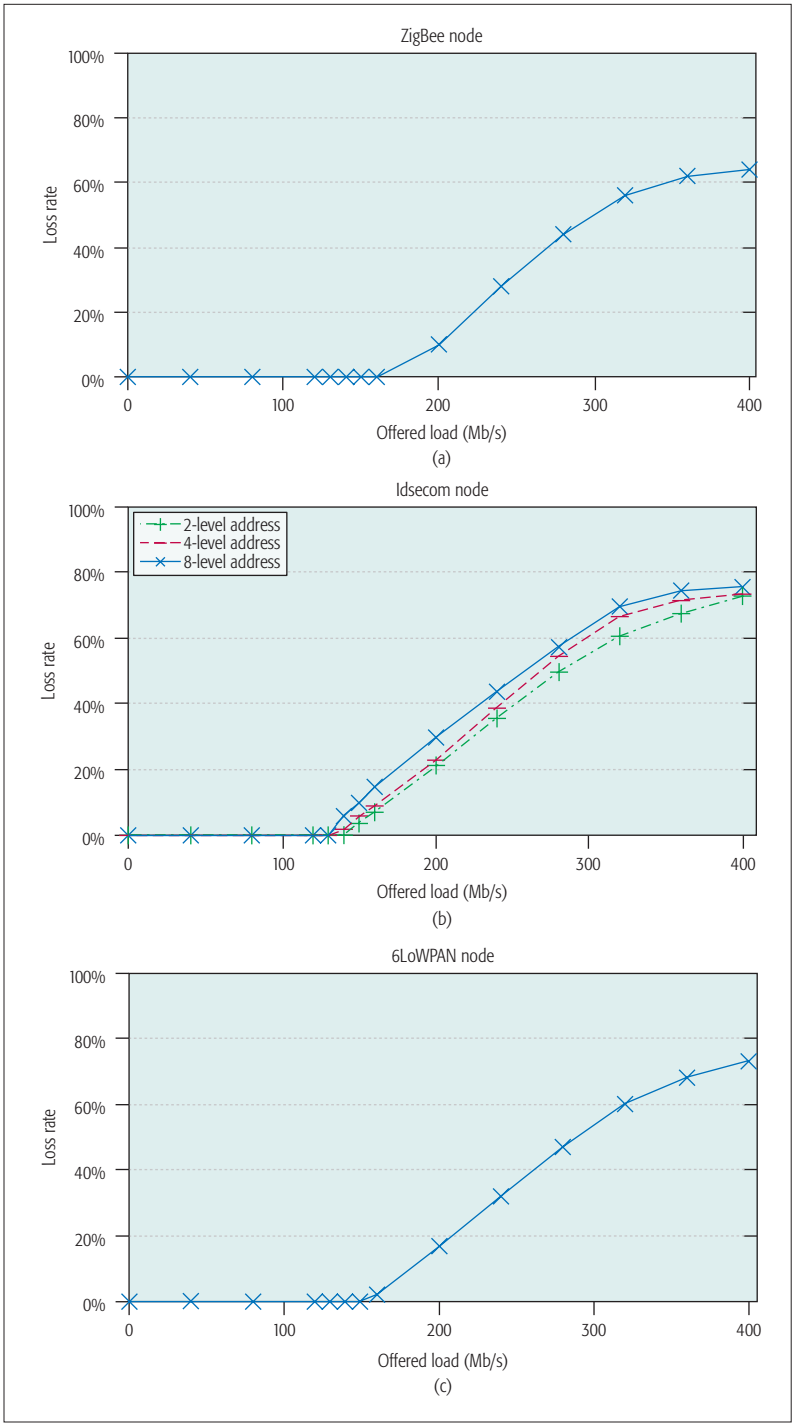

Figure 4. Exemplary network topology of ID layer capable system; frame loss ratio vs. offered load for a) ZigBee; b Idsecom; c) 6LoWPAN nodes.

Idsecom nodes perform a number of IoT-specific operations.

Registration of New Objects and Services: The registration is performed only in the Idsecom node attached to the new object/service and will not be propagated to the whole network. Since the address of the new object/service will contain identification of the location, the applications searching the new object/service will query the Idsecom node of that location.

Resolution of Objects and Services Attached to Given Idsecom Node: The nodes develop a communication process for making the information about objects and services accessible whenever any application requires it.

Publication of Services: The Idsecom node maintains information about objects/services registered and may interoperate with third party discovery platforms (e.g., Xively™).

The definition of the procedures for IoT operations together with the frame format for these procedures may be found in [8]. One of the most interesting features of Idsecom nodes is the possibility of caching the information of the IoT services at the network layer of the Idsecom nodes, which is possible since the address of the frames contains the name of the service. A parameter indicating the validity time of the information is also attached to the cached information.

In the wireless network (802.15.4) the Idsecom frames are propagated by using a dedicated PAN ID address, as with any other technology. The forwarding layer of an Idsecom node keeps track of the Idsecom sensors' (802.15.4) addresses and maps Idsecom frame addresses to their own PAN addresses.

The aforementioned functionalities have been implemented as modules (and databases) in user and kernel space of Linux OS (Fig. 1). The operations implemented in kernel space are related to forwarding and caching, which use databases for comparing addresses of the child nodes and for caching IoT information, respectively. IoT operations (registration, discovery) are deployed in user space.

The tests performed on the platform compare the forwarding capabilities of an Idsecom node with ZigBee and 6LoWPAN nodes, all three deployed on the XEN virtualization tool. Once again, the platform and the Spirent TestCenter are connected in a ring topology, so the two interfaces of the platform are 1 Gb/s Ethernet (the radio access is not tested here). The three nodes forward technology-specific frames from one interface to the other after performing all the operations at the MAC and network layers. At the MAC/Ethernet layers, the nodes identify the packets by VLAN tags. In the network layer, the ZigBee node only forwards packets, whereas the Idsecom node analyzes the IoT identifier and maps the identifier to an 802.15.4 address after querying the routing table. At last, 6LoWPAN node adapts the IPv6 address to the 802.15.4 address and forwards the frame to the outgoing interface. The frame loss ratios for increasing traffic load offered to each node are presented in Fig. 4. All the nodes have similar loss patterns: the loss ratio rises linearly through higher values of offered load (around 350 kb/s). The linear increase of loss is due to the fact that the load

carried by the node remains constant when the offered traffic increases.

Idsecom node results show the cases when the IoT identifiers have two, four, and six levels (in the Idsecom hierarchy). Longer IoT identifiers cause longer routing operations due to more complex queries to the tables. However, the performance of an Idsecom node does not differ much from other nodes, which validates the Idsecom implementation from the forwarding efficiency point of view. The three forwarding nodes (BigBee, Idsecom, and 6LoWPAN) host around 140 Mb/s throughput, which is much higher than the necessary throughput for communicating with devices by radio access (which is shown below). This undoubtedly leads us to conclude that the presented virtualization solution hosting the different technologies' nodes is suitable for communicating with the devices in HAS.

Access to Radio

The access to the radio space is shared by all the nodes of the platform. The idea is to keep the antenna hardware as simple as possible, and in terms of cost as cheap as possible, while the whole complexity rests with the programmable platform. The platform containing the virtual nodes composes a sequence of ones and zeros, which is the result of the mapping of raw data into chip sequence and is generally called baseband chip sequence. This mapping results in a high overhead of the information sent to the antenna, so the raw data rate equal to 250 kb/s required in 802.15.4 transmissions needs several megabits per second connection between platform and antenna (note that 4 bits correspond to 32 chip values without counting baseband overhead). To this end, it is important to remark that such a bit rate may be served by coax cable without the necessity of using fiber, which reduces the cost of the solution. The baseband chip sequence also contains the physical layer protocol data unit (PPDU) for synchronization at the receptor's side. The PPDU, in turn, contains information about the length of the data included in the frame. The antenna receives the baseband chip sequence and shapes it by the half-sine pulse factor and spreads the signal into the medium (modulation and amplification of signal power are the functionalities of the antenna). The modulation in the antenna follows the 802.15.4 standard for the 2.4 GHz unlicensed frequency range. Concretely, the frames are modulated with offset quadrature phase shift keying (O-QPSK). The main objective of the common antenna is to reduce interference between different systems using the same frequency range. In our platform most of the technologies may be developed sharing the radio medium in a seamless way. Thus, we avoid the complications of different small devices creating private networks in the same place by using the unlicensed radio spectrum (especially when many of the channels are interfered by WiFi transmission).

The modules (located in the platform) that send and receive the baseband chip sequence and control the access to the antenna have been developed as Linux Fedora modules on the XEN dom0 (hypervisor). From the sensors, the baseband chip sequence is queued into the kernel stack from the NIC interrupt handler (Fig. 1). If there is not enough space in the queue, the sequence gets dropped. In the next step, the whole sequence is caught by the frame formatter module, which obtains the associated frame (the formatter uses skb function to modify the sequence). The frame is copied to the MAC/802.15.4 modules (in each VM). It is the responsibility of the MAC and network layers inside the nodes to discard the frames that are not directed to the specific node.

In the opposite direction (to the sensors), the MAC/802.15.4 of the node queues the frame ready to be sent to the radio interface. The sequence formatter creates the corresponding baseband chip sequence, which is sent to the NIC. Let us remark that the MTU in the card has been increased in order to permit sending the baseband chip sequence instead of Ethernet frames. Since the distance between platform and antenna is always very short (some meters as a maximum), there is no risk that network cards in antenna and platform lose synchronization even for long sequences of bits. The tests performed with the antenna did not show any issue with hardware synchronization.

The tests presented below show the limitations of the antenna to propagate information from the nodes into the air (downlink direction). The tests are not intended to understand the effect of different wireless devices trying to transmit to the antenna (uplink) and the collisions produced in this case. This is solved, in part, by carrier sense multiple access with collision avoidance (CSMA-CA). The test scenario consisted of the platform connected to the Spirent TestCenter by a 1 Gb/s Ethernet link and to the antenna by a 1 Gb/s link. The tester generates three types of frames: ZigBee frames, Idsecom frames, and IPv6 packets directed to the three nodes and tagged with different VLAN identifiers for correct distribution to the nodes. All the frames are 127 bytes long. The rate of the flows generated in the tester increases from one test to another. The nodes process the frames and forward them to the interface connected to the antenna. The frames are converted into a baseband chip sequence and sent to the NIC, which sends it to the antenna. The antenna catches and modulates the sequence and propagates the signal into the air. The antenna uses different channels for each node's traffic by modulating with a different carrier, which is the result of the channel scan performed by MAC modules. At last, a receptor located near the antenna (5 m) converts the signal and stores the sequences that have arrived. Analysis of the sequences that have arrived makes it possible to understand which frames (from which node) properly arrived at the receptor. The proximity of the receptor together with the isolation of the air space ensure no losses in the signal propagation.

The relation between the load offered by the nodes and the load properly carried to the radio receptor is presented in Fig. 5a. The figure shows the carried traffic belonging to each one of the nodes and the aggregate (the sum of all the nodes). The rates shown in the figure refer to raw data traffic, that is, we did not consider in the analysis the first setting frames (e.g., energy

The access to the radio space is shared by all the nodes of the platform. The idea is to keep the antenna hardware as simple as possible, and in terms of cost as cheap as possible, while the whole complexity rests with the programmable platform.

™Xlively is a trademark of LogMeIn.

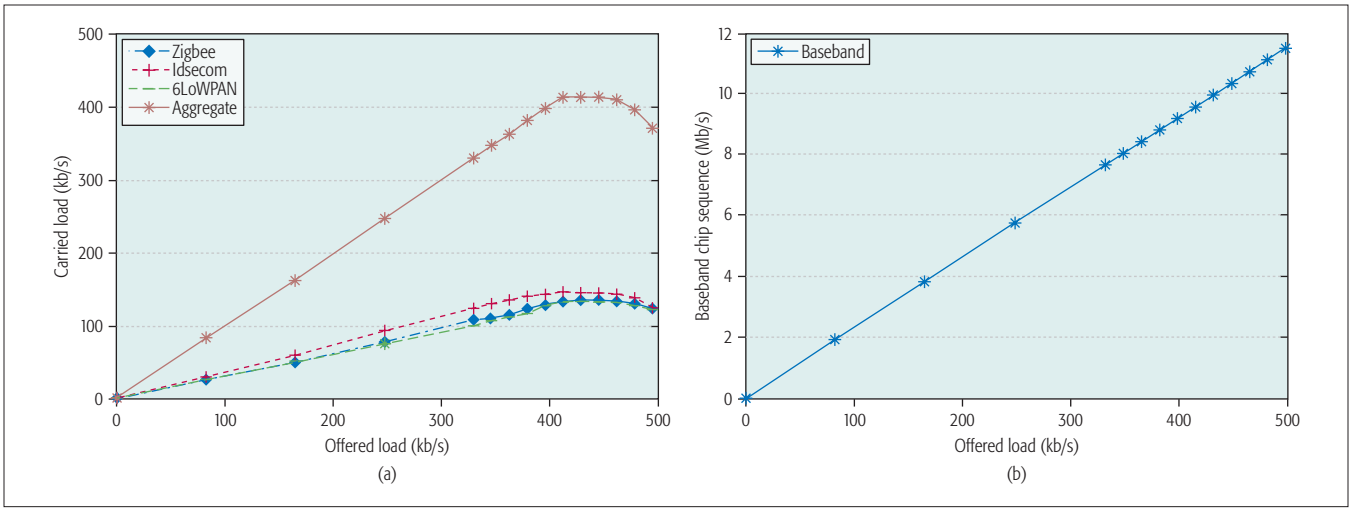

Figure 5. Results of a) carried load; b) baseband chip sequence load vs. offered load.

scan) or other management operations during the transmission. As we may observe, there are no losses until 417 kb/s offered load (offered to the three nodes in all). Each node is able to carry around 135–145 kb/s. The difference with the 250 kb/s transmission rate proposed in the standard is mainly due to the fact that we present only raw data and there is other management and control traffic effectively sent by the antenna (MAC operations). The values of carried traffic considering management and control are around 238 kb/s. The Idsecom node carries more traffic than the other nodes. The reason is that after forwarding operations, the Idsecom frame sent to the antenna is much shorter than the 6LoWPAN and ZigBee frames. As a result, the relation of data frames to management frames increases for the Idsecom case.

Figure 5b shows the load of the baseband chip sequence experienced in the link between antenna and platform. The baseband chip sequence does not suffer from losses for high values of offered load (the chip sequence rate rises linearly with the offered load). Besides this, no losses are observed in the nodes during processing of frames (the nodes are able to forward higher rates as shown in the previous section). This means that all the losses occur in the queue of the antenna and are provoked by the scarce radio spectrum resources. For 417 kb/s offered load (three full channels), the chip sequence is around 8.8 Mb/s. Thus, when the platform hosts 16 different nodes, filling the whole radio spectrum, the chip sequence rate would be less than 50 Mb/s. We may conclude that a 100 Mb/s link is enough to connect the antenna and platform, which may reduce the cost of the system.

Conclusion

This article presents a novel approach for the development of HAS based on the integration of different technologies' nodes into one virtualization platform with common access to the antenna, which is physically separated from the platform. The interface communication with the antenna is performed by sending baseband chip sequence.

The results show that the virtualization of software nodes is suitable for HAS since the overhead introduced by both software operations and virtualization allows for the normal functioning of the nodes, which is mostly limited by the access to the wireless space. The deployed platform makes it possible to integrate a high number of IoT gateways controlling wireless devices and also fulfilling application-layer operations. In addition, the software-oriented and virtualization features of the platform make feasible easy integration of any new technology for IoT deployment in HASs, such as a Thread.

Acknowledgment

This work was undertaken under the Pollux II IDSECOM project supported by the National Research Fund Luxembourg and the National Centre for Research and Development in Poland.

References

[1] D. Yang, Y. Xu, and M. Gidlund, "Wireless Coexistence between IEEE 802.11-and IEEE 802.15.4-Based Networks: A Survey," *Int'l. J. Distributed Sensor Networks*, vol. 2011, 2011.
[2] Wibutler centerpiece (last accessed Oct. 14, 2015): https://www.wibutler.com/en_GB/wibutler.
[3] Connected Home Hub mydlink Home (last accessed Oct. 14, 2015): http://www.dlink.com/pl/pl/home-solutions/mydlink-home/smart-plugs/dch-g020-mydlink-connected-home-hub
[4] Samsung's Comcast Xfinity (last accessed Oct. 14, 2015): http://www.slashgear.com/samsungs-comcast-xfinity-box-speaks-smart-home-31323120/.
[5] O. Vermesan and P. Friess, *Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems* River Publishers, 2013.
[6] E. Baccelli *et al.*, "RIOT OS: Towards an OS for the Internet of Things," *Proc. 32nd IEEE INFOCOM*, Turin, Italy, Apr. 2013.
[7] Indie Box Project (last accessed Oct. 14, 2015): ubos.net
[8] J. Mongay Batalla and P. Krawiec, "Conception of ID Layer Performance at the Network Level for Internet of Things," *Springer J. Personal and Ubiquitous Computing*, vol.18, Issue 2, 2014 , pp. 465-480.

Biographies

Jordi Mongay Batalla received his M.Sc. degree from Universitat Politecnica de Valencia (2000) and Ph.D. degree from Warsaw University of Technology (2010), where he still works as an assistant professor. In the past he worked at Telcordia Poland (Ericsson R&D), and he is now with the National Institute of Telecommunications, where, since 2010, he is head of the Internet Architectures and Applications Department. He has taken part (coordination and/

or participation) in a dozen international ICT projects. His research interest focuses mainly on quality of service in IPv4/v6, MPLS, SDN networks, future Internet architectures (content-aware networks, information-centric networks), as well as applications for the future Internet (Internet of Things, smart cities, IPTV). He is an editor of several books and an author of more than 100 international journal and conference papers.

GEORGE MASTORAKIS received his B.Eng. (Hons) in electronic engineering from the University of Manchester Institute of Science and Technology, United Kingdom, in 2000, his M.Sc. in telecommunications from University College London, United Kingdom, in 2001, and his Ph.D. in telecommunications from the University of the Aegean, Greece, in 2008. He is serving as an associate professor at the Technological Educational Institute of Crete and as a research associate at the Research & Development of Telecommunications Systems Laboratory at the Centre for Technological Research of Crete, Greece. His research interests include cognitive radio networks, networking traffic analysis, radio resource management, and energy-efficient networks. He has more than 120 publications in various international conference proceedings, workshops, scientific journals, and book chapters.

CONSTANDINOS X. MAVROMOUSTAKIS is currently an associate professor at the Department of Computer Science at the University of Nicosia, Cyprus. He received a five-year dipl. Eng (B.Sc., B.Eng., M.Eng.) in electronic and computer engineering from the Technical University of Crete, an M.Sc. in telecommunications from University College London, and his Ph.D. from the Department of Informatics at Aristotle University of Thessaloniki, Greece. He is leading the Mobile Systems Lab (MOSys Lab, http://www.mosys.unic.ac.cy/) in the Department of Computer Science at the University of Nicosia, dealing with design and implementation of hybrid wireless testbed environments and MP2P systems, IoT configurations and smart applications, as well as high performance cloud and mobile cloud computing (MCC) systems, modeling and simulation of mobile computing environments, and protocol development and deployment for large-scale heterogeneous networks and new green mobility-based protocols. He is an active member (Vice-Chair) of the IEEE/ R8 regional Cyprus Section since January 2016, and since May 2009 he has served as the Chair of the C16 Computer Society Chapter of the Cyprus IEEE Section. He has a great deal of research output in distributed mobile systems and spatio-temporal scheduling, consisting of numerous refereed publications including several books (IDEA/IGI, Springer, and Elsevier). He has served as a consultant to many industrial bodies (e.g., member of the Technical Experts for Internet of Things-IoT competition at Intel Corporation LLC [www.intel.com] for ChallengeMe), is a management member of the IEEE Communications Society (ComSoc) Radio Communications Committee (RCC) and a Board member the IEEE-SA Standards IEEE SCC42 WG2040, and has served as Track Chair and Co-Chair of various IEEE international conferences (including AINA, IWCMC, ICC, GLOBECOM, IEEE Internet of Things etc).

JERZY ŻUREK received his M.Sc. degree from Wysza Szkoa Morska, Gdynia, Poland, and his Ph.D. degree from Politechnika Gdaska, Gdask, Poland, in 2005. He is now with the National Institute of Telecommunications, where, since 2014, he is the director of the Institute. Since 2005, he has cooperated with the Volpe Research Center of the U.S. Department of Transportation in Boston Massachusetts, as well as with the University of New Hampshire, concretely with CIDLab (Critical Infrastructure Dependability Laboratory). He has participated in several European projects, among them INTERREG IV EfficienSEA, where he coordinated the work Akademia Morska partner. His research interest centers on distributive frequency range and frequency hopping systems (both terrestrial and satellite), mobile systems (3G, 4G, and 5G), software defined radio, security in radio systems, embedded systems, wireless networking, ad hoc wireless sensor networks, and cognitive radio. He is the author or co-author of more than 100 publications published in national and international journals and conference proceedings.