Computer Networks 57 (2013) 622–633

Contents lists available at SciVerse ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

A cross-layer communication module for the Internet of Things

Chong Han [a,*], Josep Miquel Jornet [a], Etimad Fadel [b], Ian F. Akyildiz [a,b]

[a] *Broadband Wireless Networking (BWN) Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA*
[b] *Department of Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia*

ARTICLE INFO

Article history:
Received 28 September 2012
Accepted 5 October 2012
Available online 22 October 2012

Keywords:
Internet of Things
Cyber-physical systems
Cross-layer design
Optimization

ABSTRACT

The Internet of Things (IoT) is a novel networking paradigm which allows the communication among all sorts of physical objects over the Internet. The IoT defines a world-wide cyber-physical system with a plethora of applications in the fields of domotics, e-health, goods monitoring and logistics, among others. The use of cross-layer communication schemes to provide adaptive solutions for the IoT is motivated by the high heterogeneity in the hardware capabilities and the communication requirements among things. In this paper, a novel cross-layer module for the IoT is proposed to accurately capture both the high heterogeneity of the IoT and the impact of the Internet as part of the network architecture. The fundamental part of the module is a mathematical framework, which is developed to obtain the optimal routing paths and the communication parameters among things, by exploiting the interrelations among different layer functionalities in the IoT. Moreover, a cross-layer communication protocol is presented to implement this optimization framework in practical scenarios. The results show that the proposed solution can achieve a global communication optimum and outperforms existing layered solutions. The novel cross-layer module is a primary step towards providing efficient and reliable end-to-end communication in the IoT.

1. Introduction

Nowadays, the Internet is used by more than two billion customers around the world to browse content, send and receive emails, access multimedia resources, play online games, and make social networking, among others. Moreover, the Internet is also expected to serve as a global platform to interconnect physical objects or "things", thus, enabling new ways of working, interacting, entertaining, and living [1,2].

Within such perspective, the Internet of Things (IoT) is a novel networking paradigm which allows the communication among all sorts of physical objects over the Internet [3,4]. The IoT is enabled by embedding communication capabilities and, in some cases, identification, sensing and actuation functionalities into daily things and communicating in extended Internet technologies. The IoT defines a truly world-wide cyber-physical system in which every physical object can be connected and controlled remotely. The ensemble of applications and services leveraging such technologies open a plethora of new business and market opportunities in the fields of domotics, e-health, real-time monitoring of industrial processes, and intelligent logistics, among others [5,6].

A major bottleneck in the IoT is posed by the very high heterogeneity in both the *hardware capabilities* and the *communication requirements* among different types of things.

- From the hardware perspective, things can have very different computation, memory, power or communication capabilities. For instance, a cellular phone or a tablet has much better communication and computation capabilities than a single-purpose electronic product such as a heart rate monitor watch.

* Corresponding author. Tel.: +1 404 894 6616.
E-mail addresses: chong.han@ece.gatech.edu (C. Han), jmjornet@ece.gatech.edu (J.M. Jornet), eafadel@kau.edu.sa (E. Fadel), ian@ece.gatech.edu (I.F. Akyildiz).

1389-1286/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comnet.2012.10.003

- Things can have very different Quality of Service (QoS) requirements in terms of delay, energy consumption or reliability. For example, minimizing the energy for communication/computation purposes is a major constraint for the batter-powered devices without efficient energy harvesting techniques. On the contrary, this energy constraint is not critical for the devices with power supply connection.

These two characteristics pose the conflict in designing a unifying framework which can take care of diversity of capabilities and functionalities of things. As a result, the heterogeneity traits of the network motivate the use of adaptive cross-layer communication schemes for the IoT.

There exists several cross-layer protocols for Wireless Sensor Networks (WSNs) [7,8], Wireless Mesh Networks (WMNs) [9] or Ad Hoc Networks (AHNs) [10]. However, they cannot be applied to the IoT due to several reasons. First, the heterogeneity of the IoT incurs that things have largely diverse hardware capabilities, different QoS requirements and individual goals. On the contrary, in WSNs, nodes usually have very similar hardware specifications, common communication requirements and a shared goal. Second, the Internet is involved in the IoT network architecture, from which it inherits a centralized and hierarchical architecture. In comparison, in WSNs, WMNs and AHNs, highly flat network architectures are considered, in which nodes communicate in a multi-hop fashion and also the Internet is not involved.

In this paper, we propose a novel cross-layer optimization framework and communication module for the IoT. Our proposed mathematical framework captures the high device and service heterogeneities of the IoT. In particular, it exploits the interrelations among the device specifications, physical layer, link layer and network layer, to find the optimal routing paths and their corresponding communication parameters, which jointly optimize the end-to-end delay and energy consumption for given QoS requirements. In addition, the impact of the Internet on the achieved QoS is also taken into account. Moreover, we propose a cross-layer communication protocol which can implement the optimization framework in practical scenarios. The results show that the proposed solution outperforms existing layered solutions and the joint-objective cross-layer solution can balance between different design objectives.

The contributions of this paper are summarized as follows:

- We provide an in-depth review of the state of the art in IoT-related initiatives and standardization efforts, to better motivate the need of a unified cross-layer solution.
- We identify the interrelations among the device capabilities, physical layer, link layer and network layer, and explain how these are captured in our solution.
- We define a cross-layer optimization framework with a single weighted joint-objective function according to the service-dependent QoS requirements.
- We propose a cross-layer protocol to implement the cross-layer optimization framework in practical scenarios.
- We provide extensive simulation results which show that our proposed solution outperforms existing layered protocols.

To the best of our knowledge, this is the first unified solution for end-to-end efficient and reliable communications in the IoT. The remainder of this article is organized as follows. In Section 2, we revise the related work in terms of on-going standardization efforts for the IoT. In Section 3, we describe the reference network architecture for the IoT that we consider throughout the paper. In Section 4, we describe our design approach and develop the cross-layer optimization framework which captures the existing relations among the different layers of the protocol stack. In Section 5, we describe the cross-layer protocol operation needed to implement our optimization framework in practical scenarios. In Section 6, we evaluate the performance of the proposed solution by means of simulation. Finally, we conclude the paper in Section 7.

2. Related work

Many research organizations are working towards the deployment and standardization of the IoT. We summarize the most relevant standardization efforts to date as follows:

- *IEEE 802.15.4 Standard* [11]: provides the specifications for the physical layer (frequency spectrum allocation, modulation, data rates, and power control) and the link layer (MAC and error control) for Low-Rate Wireless Personal Area Networks (LR-WPANs). However, it does not specify the implementation of higher-layer functionalities (e.g., routing, end-to-end reliability) or how the communication among things over the Internet is realized.
- *IETF Low power Wireless Personal Area Networks (6LoWPAN) Standard* [12]: defines a set of protocols to integrate low-complexity devices which operate under the IEEE 802.15.4 Standard into IPv6 networks. However, several challenges appear due to, among others, the mismatch between the minimum packet size for IPv6 networks and that of IEEE 802.15.4, or the difficulty to manage routing tables for the expected number of nodes involved in the IoT. Mechanisms to guarantee end-to-end reliability are not provided, either.
- *IETF Routing Over Low power and Lossy networks (ROLL) Working Group (WG)* [13]: focuses on the development of new routing protocols for low power and lossy networks, and appears as a complement to the 6LoWPAN. Unfortunately, there is still a long path for the WG to reach a complete solution. Moreover, its attention is only on efficient routing and does not guarantee any QoS requirement such as end-to-end delay or reliability.
- *ETSI Machine to Machine (M2M) Technical Committee* [14]: concentrates mainly on ad hoc networks among things in which Internet is not part of the sys-

tem. For the time being, no specifications are provided for things' addressing, location, and QoS. Furthermore, different devices use different network protocols to communicate within the same M2M network [15]. This dramatically increases the burden on the gateway, which needs to adapt every transmission among various devices.

For the time being, there are many independent solutions for the different layer functionalities in the protocol stack. In the following, we review some of the related work, for instance:

- *Physical layer*: in [16], the authors advocate a physical-layer-driven approach to protocol design for wireless sensor networks with emphasis on the underlying hardware parameters. In [17], the authors demonstrate the importance of the physical layer modeling on the performance evaluation. Nevertheless, neither of them exploits the interrelation among the MAC and other layer functionalities with the physical layer and hence, their joint influence on the end-to-end communication performance.
- *Link layer*: the authors in [18] present a broad overview of the MAC protocols conducted in the field of wireless sensor and ad hoc networks. However, they fail to provide a vivid guidance on the choice of MAC techniques and the associated parameters. By contrast, the authors in [19] propose spatial correlation-based collaborative medium access control (CC-MAC), an energy efficient MAC that exploits spatial correlation in wireless sensor networks on the MAC layer. Both of them admit that the MAC layer plays an important role in the performance of the overall system and affects other layers, while they ignore these effects and their impact on the system performance.
- *Network layer*: the authors in [20] summarize the data routing algorithms and classify the approaches into three categories: data-centric, hierarchical and location based. In addition, in [21], the authors study the design trade-offs between energy and communication overhead savings for the existing routing protocols. However, neither of them is appropriate for IoT since the physical attributes of nodes are not taken into account, which have direct impact on the performance and even the validity of the routing algorithms. Furthermore, they omit the interactions between routing algorithm and other layers.

As can be seen, these layered solutions cannot successfully capture the twofold heterogeneity of the IoT. In our vision, it is particularly challenging to develop a "one-size-fits-all" solution by following a classical approach. Alternatively, more advanced adaptive cross-layer schemes are necessary to guarantee efficient and reliable communication in the IoT. Cross-layer protocols have been successfully developed in other paradigms such as WSNs [7,8], WMNs [9] or AHNs [10], among others. However, these cannot be directly used in the IoT, due to the major aforementioned differences in the paradigms.

3. Reference network architecture

We define next the reference network architecture for the IoT which is considered throughout the paper. We identify the following network components:

- *Things:* i.e., physical objects with very diverse hardware specifications in terms of communication, computation, memory and data storage capacity, or transmission power. Personal electronic devices, home appliances or all sorts of equipment, are examples of things.
- *Access Points (APs):* i.e., more advanced devices which play the role of local network coordinator as well as interface and gateway for the communication over the Internet. We refer to the set of things under the control of a single AP as the *AP domain*.
- *The Internet:* i.e., a fundamental component of the IoT. In our analysis, we treat the Internet as a black box which is characterized by an stochastic queueing delay model and a stochastic packet loss model [22].

Fig. 1 illustrates the network architecture of the IoT, in which several *things* are connected to the *Internet* via a common *AP*. In common scenarios (e.g., at home, in the office), the *AP domain* is composed by a few tens of things. Additional network considerations are summarized as follows:

- *Communication types*: we distinguish two types of communication, namely, intra-AP and inter-AP. *Intra-AP* communication is established among things within the same AP domain. Despite direct transmission among things might be possible, different capabilities among them motivates the use of the AP for coordination. *Inter-AP* communication is established among things in different AP domains. In this case, the AP serves as a gateway as well as the coordinator.
- *Interconnection between things and AP*: the AP is able to directly communicate with all the things in its domain in a single hop. However, not all the things are able to directly communicate with the AP, in which case

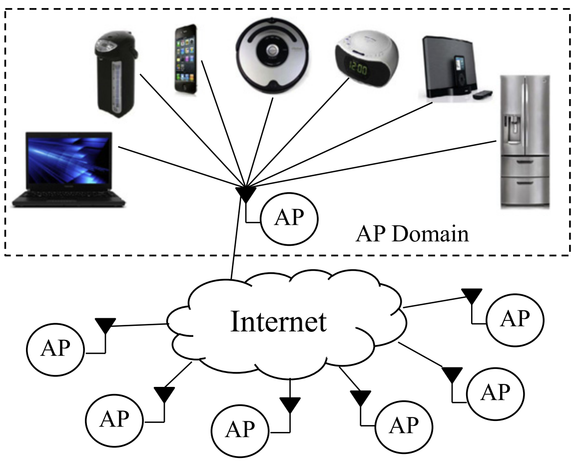


Fig. 1. Abstract network architecture of the IoT.

multi-hop links are required. This asymmetry in the links results from the fact that the transmission power of the AP is generally higher than that of many things.

- *Network knowledge*: the AP is able to collect information about all the things in its domain (e.g., during the network association phase, as we explain in Section 5), such as device type, approximate location, communication or computation capabilities, among others. By contrast, things might only know the AP ID, but have no information about other things. Things are not able to perform complex computation locally.
- *Centralized decision*: the AP is able to run optimization algorithms locally by exploiting its network knowledge. Therefore, the computation complexity is shifted from things as in a distributed manner to the AP in a centralized fashion. As a result, global optimal routes and communication parameter values for the AP domain can be obtained. This is feasible since the size of the AP domain is in the order of a few tens of things and, thus, the resulting complexity is affordable for standard AP hardware capabilities.

4. Cross-layer optimization framework

In this section, we present our new cross-layer optimization framework for the IoT. First, we describe our approach to define and design the optimization framework. Second, we describe the functionalities and interrelations among the physical layer, the link layer and the network layer as well as the things' capabilities. Third, we mathematically derive the framework.

4.1. Cross-layer design approach

We follow a *resource allocation approach* [23] to integrate different communication functionalities into one coherent mathematical optimization model and to provide an adaptive solution for cross-layer design and control. In our framework, we incorporate multiple application-based objectives, with different scaling weights for different situations, into one optimization problem. Additionally, our solution relies in a *centralized optimization model* to jointly control the parameters at the physical layer (channel, modulation), link layer (MAC, error control), and network layer (addressing, routing) in the IoT, to ultimately reach the optimality according to the application-dependent objective function.

4.2. Multi-objective optimization

The IoT should provide differentiated services for applications with different QoS requirements, ranging from error-limited applications or minimum energy consumption applications to highly-delay-sensitive applications or any combination of them. Hence, we consider a multi-objective optimization problem [24] which can simultaneously optimize multiple conflicting objectives subject to certain constraints, as follows:

$$\min\{\mathrm{PER}_{e2e}, E_{e2e}, T_{e2e}\}, \tag{1}$$

where $\mathrm{PER}_{e2e}, E_{e2e}$ and T_{e2e} stand for end-to-end packet error rate, energy consumption and delay, respectively.

In order to solve this multi-objective optimization problem, an intuitive approach is to construct a single *aggregate objective function* which is defined by the weighted linear combination of each objective. However, one concern that arises in this situation is that the three objectives differ in the units in which they are measured as well as their order of magnitude. To resolve this, we normalize each term and optimize their deviations with respect to a pre-defined threshold, instead to minimize their absolute values. The overall objective function of the framework becomes:

$$\min\left\{w_{\mathrm{PER}}\cdot\left|\frac{\mathrm{PER}_{e2e}}{\mathrm{PER}_{opt}}-1\right|+w_E\cdot\left|\frac{E_{e2e}}{E_{opt}}-1\right|w_{\mathrm{PER}}\cdot\left|\frac{\mathrm{PER}_{e2e}}{\mathrm{PER}_{TH}}-1\right|\right.$$
$$\left.+\,w_T\cdot\left|\frac{T_{e2e}}{T_{opt}}-1\right|\right\}, \tag{2}$$

where $w_{\mathrm{PER}}+w_E+w_T=1$ are the three linear weights for the end-to-end packet error rate, energy consumption and time delay objectives, and $\mathrm{PER}_{opt}, E_{opt}, T_{opt}$ are the end-to-end packet error rate, energy consumption and delay *utopia values* [25] for normalizing purposes, respectively. These utopia values are defined to be the unattainable minimum values, which are used to provide the non-dimensional objective functions and can be computed offline. Therefore, we have a single objective optimization problem to solve. Accordingly to different QoS requirements, we can control the emphasis of each term in the objective function (2) by assigning different weights.

With the objective of either minimizing the packet error rate to improve the network efficiency, minimizing the energy consumption to prolong the network lifetime, reducing the network end-to-end delay to increase the system throughput, or any combination of these, the proposed cross-layer framework jointly selects the best combination of transmit power and modulation (Section 4.3), the appropriate error control mechanism and MAC parameter values (Section 4.4), and the proper addressing and routing algorithms (Section 4.5). We define next the functionalities implemented at each layer, the tunable parameter values and the interrelations among layers, as well as the impact of the things capabilities in each of them.

4.3. Physical layer functionalities

At the physical layer, different things have different maximum transmission power, can select different modulation schemes, and have different data storage capacity (i.e., number of packets that thincs can locally queue).

4.3.1. Frequency allocation and channel model

In our model, we consider that things follow the frequency spectrum allocation defined by the IEEE 802.15.4 standard [11]. Things are able to dynamically select among any of the 5-MHz-wide sub-channels in the 2400–2480 MHz band. Since many of the applications of the IoT are indoor (e.g., home, office), we consider the ITU channel model for indoor propagation [26]. The total path loss L in dB is given by

$$L(f,d) = 20\log_{10}(f) + N\log_{10}(d) + L_f(n) - 28, \quad (3)$$

where f is the carrier frequency in MHz, d is the transmission distance in meters, N is the distance attenuation coefficient (i.e., $N = 20$ in our simulations), and L_f is the floor penetration loss factor and n is the number of floors between the transmitter and the receiver (we consider only one floor in our current scenario, i.e., $n = 1, L_f(1) = 0$).

4.3.2. Transmission power, modulation and bit error rate

The transmission power and the modulation have a direct impact on the Bit Error Rate (BER). Over link i, The BER, BER^i_{link}, is determined by the Signal-to-Noise Ratio (SNR), SNR^i_{link}, and the modulation mod^i, as follows

$$\mathrm{BER}^i_{link}(f,d) = \Psi\left(\mathrm{SNR}^i_{link}(f,d), mod^i\right), \quad (4)$$

where Ψ returns the BER for a given modulation and SNR, and it is well-known for standard modulations. The SNR of link i, SNR^i_{link} is given in dB by

$$\mathrm{SNR}^i_{link}(f,d) = P^i_{Tx} - L(f,d)^i_{link} - P_{noise}, \quad (5)$$

where P^i_{Tx} is the transmission power in dB over link i and P_{noise} refers to the total noise power at the receiver in dB.

We consider three standard modulations, namely, BPSK, QPSK and 16-QAM, but any other modulation can be easily included in the framework. These modulations are distinguishable in terms of achievable BER for a given SNR, i.e., Ψ in (4), and spectral efficiency, i.e., theoretical achievable data bit-rate for a given transmission bandwidth. A higher complexity modulation exhibits higher bandwidth efficiency, which results in a higher transmission data rate or a shorter transmission time, T^i_{data}. However, these come with the cost of an increase in the energy consumption at the transmitter and the receiver due to the increase in the computation complexity, as well as, potentially, also in the processing time. Furthermore, more complex modulations require a higher SNR^i_{link}, thus, higher P^i_{Tx}, to achieve the same BER^i_{link}. At the same time, though, the transmission time, T^i_{data}, is shorter, which also affects the link energy consumption. These interrelations are properly captured in the framework.

4.3.3. Data storage capacity and packet dropout probability

The data storage capacity mem of the things affects the packet dropout rate, i.e., the probability of discarding a packet at link i, $P^i_{packet\text{-}dropout}$, due to the fact that it cannot be queued at the transmitter or at the receiver. This is given by

$$P^i_{packet\text{-}dropout} = \Gamma\left(mem^i, R^i_{traffic}\right), \quad (6)$$

where Γ is a function that relates the maximum number of packets mem^i that can be queued at the transmitter or the receiver and the total local traffic (own traffic and relayed traffic), $R^i_{traffic}$. For example, in the simplest case, Poisson traffic can be assumed, and transmitter and receiver can be modeled as a single server queue with a buffer of size mem^i.

4.4. Link layer

At the link layer, we analyze the impact of the error control mechanism and the MAC protocol on the network performance, as well as, their interrelations with other layers and the limitations imposed by the things capabilities.

4.4.1. Error control and packet error rate

Among the options for error control, Forward Error Correction (FEC) codes are used to fix erroneous bits introduced by the channel, the noise and the interference, and ultimately to reduce the Packet Error Rate of link i, PER^i_{link}. Specifically, if we consider the use of block codes, for a (n, k, t) code with block size n and k bits of information, t errors per block can be successfully corrected at most. In our framework, we use the following notations for $n = N_{bits}$ and $k = N^{data}_{bits}$. Despite FEC codes improve the PER, the addition of N^{FEC}_{bits} redundant bits reduces the effective transmission data rate by N^{data}_{bits}/N_{bits}.

Another common error control mechanism is Automatic Repeat reQuest (ARQ), which is based on the retransmission of the whole packet in case of erroneous reception. This scheme is spectrally efficient only when the channel conditions are favorable, i.e., SNR^i_{link} is high, or BER^i_{link} is low, since no redundancy bits are introduced in the packet. If the BER^i_{link} is relatively high, the number of retransmissions increases, and so do both the energy consumption and delay.

To exploit the benefits from FEC codes for poor quality channel conditions, i.e., when the SNR^i_{link} is low, as well as the merits of ARQ when the channel conditions are good, i.e., when the SNR^i_{link} is high, we advocate for the use of a Hybrid ARQ scheme [27]. Hybrid ARQ schemes result from the combination of the two approaches. Initially, an uncoded or lightly coded packet is transmitted. If the received packet has more errors than those that can be corrected by the chosen FEC code, a more robust FEC code is chosen.

In our framework, things make use of Bose–Chaudhuri–Hocquenghem (BCH) codes [28]. BCH codes are used instead of more complex codes such as Convolutional Codes, because BCH codes have higher energy efficiency and lower computation complexity [29]. These two factors are critical for things with very limited energy storage and computation capability. In transmission, the initial packet is either not coded or coded with a (128, 106, 3) BCH code to reduce PER^i_{link} without drastically sacrificing the transmission data rate. If the first transmission fails, i.e., the number of errors is larger than that can be corrected, a more robust FEC code is used for the retransmitted packet (e.g., (128, 78, 7)) until the packet is successfully decoded or the maximum number of transmissions is expired. Using this Hybrid ARQ error control scheme, the overall packet error rate over link i is given by

$$\mathrm{PER}^i_{link\text{-}RTR} = \Upsilon(\mathrm{PER}^i_{link\text{-}uncoded}, N^i_{link\text{-}max}, N^{FEC}_{bits}), \quad (7)$$

where Υ is a function that relates the PER of link i after Hybrid ARQ error control, $\mathrm{PER}^i_{link\text{-}RTR}$, with the uncoded link PER $\mathrm{PER}^i_{link\text{-}uncoded}$. In the above equation, N^{FEC}_{bits} is the FEC redundancy length and $N^{i}_{link\text{-}max}$ is the maximum number of transmissions including retransmissions, which are both adjustable in our cross-layer framework.

As can be seen, larger N^{FEC}_{bits} and larger $N^{i}_{link\text{-}max}$ lead to (i) a lower transmission data rate or larger link delay, (ii) a lower PER, and (iii) a higher total energy consumption. The impact of the error control parameters interplays with those made by adjusting the transmission power and the modulation scheme at the physical layer. Hence, it is beneficial to explore the interactions among the physical layer and link layer functionalities and jointly determine the transmission power, the modulation scheme, and the error control parameters.

4.4.2. Medium access control

We consider a variation of Sleep MAC (SMAC) [30] and Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), for mainly two reasons. First, idle listening is a major source of energy consumption in low traffic applications such as those expected in the IoT, and many of the physical objects in the IoT may have very limited energy storage. Therefore, we adopt the idea of SMAC in which things periodically listen and sleep [30]. For example, almost, 90% of the total energy consumed during idle listening can be saved when things use an awake/sleep duty cycle of 10%. However, by decreasing the duty cycle, the chances for things to be connected decrease too and the link delay increases. Second, the interference among the things in one AP domain or in close AP domains affects the SNR and thus degrades the BER and the PER. In the CSMA/CA, a node attempts to reserve the channel after it sees the medium idle for an Inter-Frame Space (IFS) amount of time. If the node fails to reserve the medium, it switches to sleep mode to save energy and waits for the next listening cycle. This medium access method can eliminate the interference drastically if the carrier sensing is properly performed. As a result, the hybrid of SMAC and CSMA/CA medium access protocol can save the energy as well as reduce the interference among the things.

In our framework, the duration of the listen and sleep cycles ($T_{listen}, T_{sleep} = 9 \times T_{listen}$ for a 10% duty cycle) are adaptive to the QoS requirements and they are set the same for all nodes in one AP domain. The longer the sleep duration is, the lower the idle energy consumption, but the longer end-to-end delay. This duration parameter in the MAC protocol is taken into account in our cross-layer framework to interplay with the physical layer parameters as well as Hybrid ARQ parameters.

4.5. Network layer

At the network layer, we look at the things addressing and information routing in the IoT. In addition, we discuss the impact of the packet size on the different layers.

4.5.1. Addressing of things

The IoT is expected to have an incredibly high number of things and each of them should be retrievable with a unique address. Consistently with 6LoWPAN, we advocate for the use of IPv6 addressing for the IoT. IPv6 addresses are expressed by 128 bits, which allow the definition of 10^{38} unique addresses (these are expectedly enough for the time being). However, IPv6 addresses are only used for inter-AP communications, while much shorter local addresses are used in intra-AP communications. The AP replaces the thing local address by the thing IPv6 full address for communication with other AP domains.

4.5.2. Routing

We consider a destination-based routing mechanism [21] where the AP selects the end-to-end route and the configuration parameters for each link in a centralized manner, due to the following reasons. First, it is consistent with the Internet hierarchical architecture composed of the Internet, gateway, router and end users from top to bottom. Second, only the AP has information of the things in its operation region and is capable to perform optimization computations. Third, the complexity of this centralized routing is comparable to that of existing distributed cross-layer solutions. By contrast, our framework achieves the global optimal solution, contrary to the distributed case, in which the end-to-end optimal problem is solved by doing only local optimization link by link.

4.5.3. Impact of the packet length

In our framework, a fixed packet size N_{bits} is selected and used for all the links throughout a given path. A larger packet size results in a reduced end-to-end delay by saving the handshake time $T^{i}_{handshake}$, the acknowledgement time T^{i}_{ack}, and the queuing time $T^{i}_{queueing}$ that are required for each packet. Additionally, the reduction of the total number of packets to be sent has an impact on the total energy consumption, while at the same time, transmitting more bits of information in a packet affects the PER. All these interrelations are incorporated in our cross-layer framework.

So far, we have explored the interrelations among the parameters at the physical layer (Section 4.3), the link layer (Section 4.4) and the network layer 4.5. These parameters including the transmission power, modulation type, the FEC length, the number of retransmissions, the listen duration, the packet size and their interactions, are all captured in our cross-layer framework, as described in the following section.

4.6. Mathematical framework

By starting from the knowledge of the things in its domain, the AP runs the optimization algorithm locally, finding the optimal domain path and the corresponding communication parameters, which include transmission power at link i, P^{i}_{link}, modulation type, mod^{i}, packet size, N_{bits}, additional bits due to FEC codes N^{FEC}_{bits}, the maximum number of transmissions including retransmission $N^{i}_{link\text{-}max}$, and listen time duration T_{listen}. The cross-layer optimization problem is mathematically defined as follows:

$$\text{Given (offline)}: \ \mathrm{PER}_{opt}, E_{opt}, T_{opt}, \mathrm{PER}_{TH}, E_{TH}, E^{k}_{TH}, T_{TH}, R_{TH}, N^{header}_{bits}, \mathrm{PER}_{Internet}, T_{Internet}, \tag{8}$$

$$\Gamma(\cdot), \Upsilon(\cdot), \Psi(\cdot), \gamma, T^i_{handshake}, T^i_{data}, T^i_{timeout}, T^i_{ack}, T^i_{queueing}, mem^i. \tag{9}$$

$$\text{Compute (offline)}: \ w_{\text{PER}}, w_E, w_T, N^i_{link}, R^i_{traffic}. \tag{10}$$

$$\text{Find}: \ P^i_{Tx}, mod^i, N_{bits}, N^{FEC}_{bits}, T_{listen}, N^i_{link-max}. \tag{11}$$

$$\text{Minimize}: \ w_{\text{PER}} \cdot \left|\frac{\text{PER}_{e2e}}{\text{PER}_{opt}} - 1\right| + w_E \cdot \left|\frac{E_{e2e}}{E_{opt}} - 1\right| + w_T \cdot \left|\frac{T_{e2e}}{T_{opt}} - 1\right| \tag{12}$$

$$\text{Subject to}: \ w_{\text{PER}} + w_E + w_T = 1, \tag{13}$$

$$\text{PER}_{e2e} = 1 - (1 - \text{PER}_{multi-hops})(1 - \text{PER}_{Internet}) \leq \text{PER}_{TH}, \tag{14}$$

$$N_{bits} = N^{header}_{bits} + N^{FEC}_{bits} + N^{data}_{bits}, \tag{15}$$

$$\text{PER}^i_{link-RTR} = \Upsilon(\text{PER}^i_{link-uncoded}, N^i_{link}, N^{FEC}_{bits}), \tag{16}$$

$$N^i_{link} = (1 - \text{PER}^i_{link-uncoded})^{-1}, \tag{17}$$

$$E^k = N_{bits} \cdot E^k_b \leq E^k_{TH}, \tag{18}$$

$$E_{e2e} = \sum_{k=1}^{N_{hop}} E^k \leq E_{TH}, \tag{19}$$

$$T^i \leq (T^i_{handshake} + T^i_{data} + T^i_{timeout}) \cdot (N^i_{link} - 1) + (T^i_{handshake} + T^i_{data} + T^i_{ack}) + T_{sleep} + T^i_{DSP}, \tag{20}$$

$$T_{e2e} = \sum_{i=1}^{N_{hop}} (T^i_{queueing} + T^i) + T_{Internet}, \tag{21}$$

$$P(T_{e2e} \leq T_{TH}) \geq \gamma, \tag{22}$$

$$P(T_{e2e} \geq T_{TH}) \leq \frac{\text{var}(T_{queueing+Internet})}{\text{var}(T_{queueing+Internet}) + \left(T_{TH} - \sum_{i=1}^{N_{hop}} T^i - \overline{T_{queueing+Internet}}\right)^2} \leq 1 - \gamma, \tag{23}$$

$$R_{e2e} = \frac{N^{data}_{bits}}{T_{e2e}} \geq R_{TH}. \tag{24}$$

The notation in the framework is as follows:

- Restricted by the threshold $\text{PER}_{TH}, \text{PER}_{e2e}$ is the end-to-end packet error rate, which is dependent on the PER of the multi-hop transmission, $\text{PER}_{multi\text{-}hops}$, and the PER of the Internet, $\text{PER}_{Internet}$.
- $\text{PER}_{multi\text{-}hops}$ is a function of the PER over link i with Hybrid ARQ error control, $\text{PER}^i_{link\text{-}RTR}$, given by

$$\text{PER}_{multi\text{-}hops} = 1 - \prod_{i=1}^{N_{hop}} \left(1 - \text{PER}^i_{link\text{-}RTR}\right). \tag{25}$$

- $P^i_{packet\text{-}dropout}$ is the packet dropout rate over link i, and BER^i_{link} is the bit error rate over link i, which is a function of signal to noise ratio over the link SNR^i_{link} and the modulation type mod^i (see (4)). These parameters determine the uncoded packet error rate over link i as follows:

$$\text{PER}^i_{link\text{-}uncoded} = \left(1 - P^i_{packet\text{-}dropout}\right) \cdot \left[1 - \left(1 - \text{BER}^i_{link}\right)^{N_{bits}}\right]. \tag{26}$$

- N_{bits} is the packet size, which contains the header length, data length and the FEC redundancy length.
- N^i_{link} is the upper-bound for the number of transmissions of a packet with correctly decoding, over link i.
- E^k is the energy on kth node, which is the product of the packet size N_{bits} and the energy required for one bit, E^k_b.
- E^k_b is calculated as

$$E^k_b = 2E^{elec}_b + \frac{P^k_{Tx}}{R^k_{traffic}}, \tag{27}$$

where $E^{elec}_b = E^{elec}_{b-Tx} = E^{elec}_{b-Rx}$ in Joule/bit is the distance-independent energy to transmit one bit. E^{elec}_{b-Tx} is the energy per bit needed by the transmitter electronics (PLLs, VCOs, AMPs, DSP, etc.) and E^{elec}_{b-Rx} is the energy per bit utilized by the receiver electronics.

- E_{e2e} is the overall energy consumption over the entire path, with the constraint E_{TH}.
- T^i is the delay at link i excluding the queueing delay. It is composed of the time for handshake $T^i_{handshake}$, time for data transmission T^i_{data}, timeout delay $T^i_{timeout}$, time for acknowledgement T^i_{ack}, sleep time T_{sleep}, and signal processing time T^i_{DSP}.
- Restricted by the constraint T_{TH}, T_{e2e} is the end-to-end time duration including the Internet delay $T_{Internet}$ for inter-AP communications, and the link queueing delay $T^i_{queueing}$. Both of the queueing delay and the Internet duration are determined by many factors like current traffic, other nodes behavior or hardware status, among others. By Central Limit Theorem, we can model the overall end-to-to delay T_{e2e} as a Gaussian distributed random variable with mean $\left(\sum_{i=1}^{N_{hop}} T^i + \overline{T_{queueing+Internet}}\right)$ and variance $(\text{var}(T_{queueing+Internet}))$, where $T_{queueing+Internet} = \sum_{i=1}^{N_{hop}} T^i_{queueing} + T_{Internet}$.
- For a target probability γ, we use the Chebyshev's inequality [31] to decompose the end-to-end delay constraint into $P(T_{e2e} \geqslant T_{TH}) \leqslant 1 - \gamma$, by satisfying

$$T_{TH} - \sum_{i=1}^{N_{hop}} T^i - \overline{T_{queueing+Internet}} > 0. \tag{28}$$

- R_{e2e} is the end-to-end throughput, which is a inversely proportional to the end-to-end delay T_{e2e}.
- Table 1 summarizes the parameters that consist of the optimization solution.

Table 1
Adaptive parameters in the cross-layer framework.

Layer	Parameters
Physical layer	Transmission power P^i_{Tx} Modulation scheme mod^i
Link layer	FEC coding scheme Coding length N^{FEC}_{bits} Maximum number of transmissions $N^i_{link-max}$ Listening period length T_{listen}
Network layer	Routing path Packet length N_{bits}

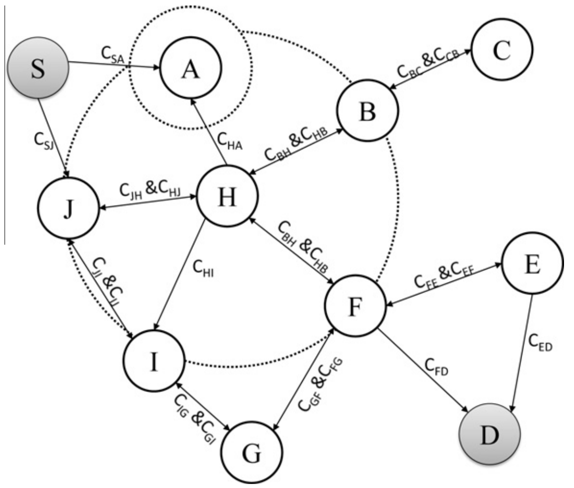


Fig. 2. Illustration of the optimization problem in a directed acyclic graph.

The proposed optimization problem can be solved in two steps.

- *Formulation of the directed acyclic graph*. According to the objective function, the constraints, and the hardware capabilities of things, the optimization problem can be expressed as a corresponding shortest-path problem in a directed acyclic graph, e.g., Fig. 2 for an intra-AP communication scenario. Node *S* is the source, node *D* is the destination, and *A, B, C, E, F, G, H, I, J* are intermediate nodes. Since the nodes have distinct communication abilities, the coverage range, denoted by the dashed circle (for a clear view, only the coverage range circles for node *A* and *H* are shown), varies for each node. The costs associated to the directional link from node i to j, C_{ij}, are derived accordingly to be consistent with the objective function. Without lose of generality, this graph can be extended for the inter-AP communication, by changing node *D* to be the AP for the Internet connection. In addition, a bilateral Internet link between the two APs and the topology for the destination AP range need to be added.
- *Solving the shortest path problem*. So far, the optimization problem has been transformed into a shortest path problem in the directed acyclic graph, with nodes, directional edges, different coverage, and associated costs. The goal of this shortest path problem is to select the path with the smallest cost from *S* to *D*. Many algorithms can solve this problem, e.g., Dijkstra's algorithm [32], which functions by constructing a shortest-path tree from the source node to every other node in the graph, and stops once the shortest path to the destination node has been determined. The complexity of Dijkstra's algorithm is bounded by $\mathcal{O}(V^2)$, where *V* is the number of nodes involved.

5. Protocol operation

In this section, we propose a cross-layer protocol which implements our cross-layer optimization framework in practical scenarios. We divide the protocol operation in four phases:

- *Phase I: Service discovery* & *Network association.* The AP periodically broadcasts its ID and supported communication parameters within its domain. This is feasible since we consider that the AP has sufficiently large power to directly communicate with every thing in its domain (Section 3). Correspondingly, the things in the AP domain register themselves to the AP by means of sending a Network Association (NAS) packet. This registration can be done either with direct communication or in a multi-hop manner, depending on the location and the transmission power of each thing. We cannot assume that things can have a notion of their location. However, the AP can cyclically modify its transmission power and define virtual regions. Only those things in a closer virtual region than the current transmitter are eligible to forward the network NAS packet towards the AP. By following this procedure, the AP collects information about all the things in its domain, e.g., device type, communication capabilities and logical location. Things can update their network association at any time by sending a new NAS packet. NAS packets are transmitted by following the normal MAC operation described in Section 4.4.
- *Phase II: Transmission initiation.* When a node has data to transmit, it first checks if it has a valid route for the destination (some things may be able to keep a local routing table, some others might not). If it has a timely route, it proceeds with the transmission by following the MAC normal operation described in Section 4.4. Otherwise, it generates a Route Request (RR) packet containing the destination thing ID. The RR packet is transmitted towards the AP by following the normal MAC operation. When an intermediate node receives a RR packet, it checks its current available resources and its logical location. If it has sufficient resources and it is closer to the AP, it forwards the RR packet after appending its updated information to it. Otherwise, it ignores the packet and continues with its normal operation.
- *Phase III: Route definition.* The AP receives the RR packet over several paths, whose intermediate nodes are listed as priority candidates for data transmission (the RR packet contains the most updated information about

some of the things in the domain). Then the AP formulates the optimization framework given the potential path and QoS requirements to find the optimal path and the associated communication parameters, as explained in Section 4.6. The decision depends on the QoS requirements, the location of the source and destination things, and whether it is an intra-AP (in which the AP may or may not participate in the transmission) or inter-AP transmission (in this case, the AP broadcast the optimal route from the source thing to itself). After finding the optimal path and communication parameters, the AP directly informs the chosen nodes with a single broadcast transmission.

- *Phase IV: Message Transmission.* Upon the reception of the route information directly from the AP, the destination thing sends a Route Acknowledge (RA) packet to the previous hop in the route in order to acknowledge its availability. The process is repeated hop-by-hop until the RA packet reaches the source. After this, the data is transmitted by following the optimal route with the chosen communication parameters and according to the MAC normal operation described in Section 4.4. If the source thing does not receive the RA packet after a given time-out, it sends a new RR packet to the AP, with a flag to mark that the previous route was not valid. The AP can decide to compute a new route or force Phase I.

In this AP-oriented architecture, the computation complexity is shifted from things to the AP. Furthermore, the AP coordination can effectively diminish the impact of having the multiple-flow problems, and provide a globally optimal solution.

6. Performance evaluation

In this section we compare the performance achieved by our cross-layer solution against that achieved by traditional layered solutions, in which individual communication functionalities operate in isolation. We compare the results when the QoS is focused on either end-to-end delay minimization, energy consumption minimization, or a lineal combination of both, while the PER is constraint to be below $\mathrm{PER}_{TH} = 10^{-4}$.

In our simulations, the total amount of data to transmit per transmission is 10^5 bits and the possible packet sizes N_{bits} are 200, 500 or 2000 bits. For each transmission, the thing randomly selects its destination. On average, 50% of the links are inter-AP and 50% of the links are intra-AP. Things have diverse hardware capabilities, in terms of computing, memory, energy storage, power and communication. Specifically, the link rate of things ranges among [250, 10^3, 10^4] kbps and the power of things varies among [10, 30, 50, 80, 100] milliwatts. Besides these deterministic parameter values, we summarize in Table 2 the random variables that are considered in our simulations. The error bars in the figures represent the uncertainty interval at the 95% confidence level. The four layered solutions that are plotted for comparison differ in the modulation scheme and the packet size, and make use of the shortest path routing.

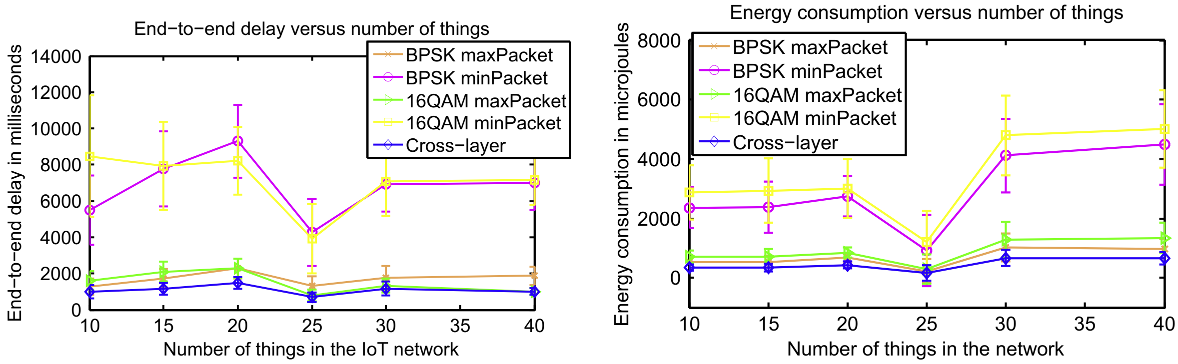
Table 2
Random Variables (RVs) for the parameters in the cross-layer framework.

RV	Distribution
Packet error rate of the Internet	$\mathrm{PER}_{Internet} \sim \mathcal{U}\left(0, 10^{-4}\right)$
Packet dropout rate over link i	$P^{i}_{packet-dropout} \sim \mathcal{U}(0, 0.1)$
Internet delay	$T_{Internet} \sim \mathcal{N}\left(10^2, 10^4\right)$ ms
Things queuing delay at each link	$T^{i}_{queueing} \sim \mathcal{N}\left(10, 10^4\right)$ ms
Noise at each link	$noise^{i} \sim \mathcal{N}(0, P_{noise})$, where $10\log_{10}P_{noise} = -86$ dB

Fig. 3a and b show the end-to-end delay in milliseconds and the energy consumption in microjoules versus the number of things in the IoT network. The distance between the transmitter and the receiver is 40 m. Our cross-layer solution has at least 10% gain over other layered solutions. In addition, we can observe that neither the end-to-end delay nor the energy consumption increases as the number of things increases. This can be explained that the higher node density essentially create more options of paths for transmission, while the end-to-end consumption has no proportional relationship with it. The 95% confidence interval imply that our cross-layer solutions have better stability than other layered solutions, and the end-to-end performance does not necessarily change although the computation complexity at the AP increases.

The end-to-end delay and the energy consumption curves are shown in Fig. 4a and 4b as a function of the distance between source and destination. Since many of the applications of IoT are indoor (e.g., home, office), we consider the distance less than 40 m. In our simulation, the distance increases from 10 m to 40 m and totally 10 things are involved. When the transmission distance is increased, the propagation delay increases and the number of nodes in the path may increase. We can neglect the influence of the propagation delay over this order of distance. However, increasing the number of hops in the path implies the rise of the end-to-end delay since there are additional handshake, processing and queuing delay introduced into the transmission. We can observe the trend of the increase in T_{e2e} as the distance grows. Similarly, the energy consumed for the longer distance and by the additional nodes increase the overall E_{e2e}. As shown in Fig. 4, the performance gain of our cross-layer solution over the best mod/N_{bits} combination increases with the distance, which further proves the advantage of our cross-layer solution over the layered solutions. Among the four layered solutions considered, the one with the simplest modulation scheme and the largest packet size performs the best, although it has around 10% performance degradation compared to our cross-layer solution.

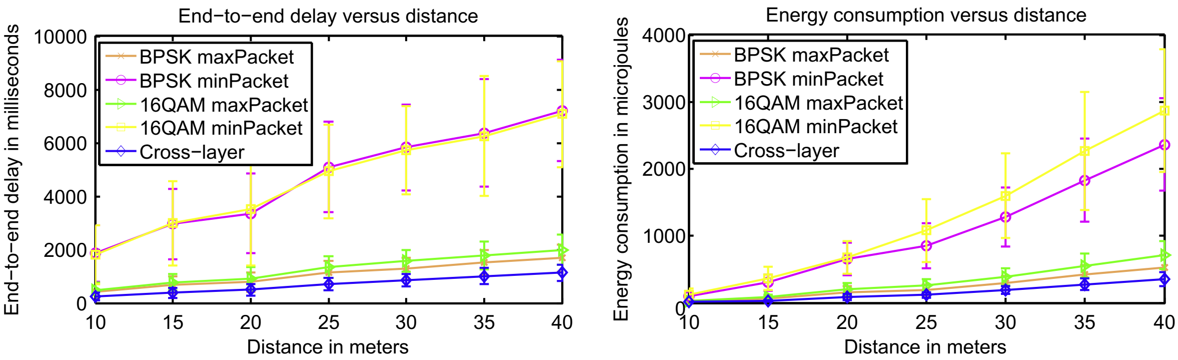
Moreover, we investigate the performance difference between the cross-layer solution with single objective and that with joint objectives, in Fig. 5a and 5b. In Fig. 5a, we compare the T_{e2e} performance among cross-layer solutions with a single-objective (minimizing either



(a) Delay vs. number of things

(b) Energy consumption vs. number of things

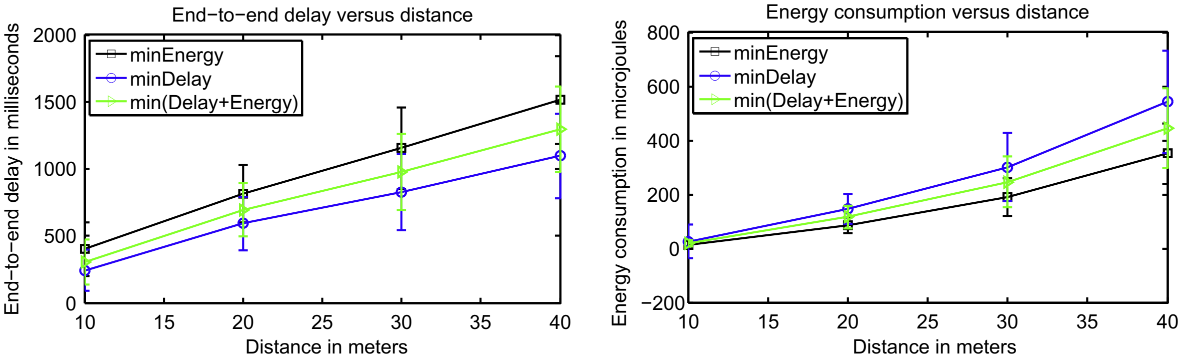
Fig. 3. Comparison between layered solutions and cross-layer solutions with respect to the number of things.



(a) Delay vs. distance

(b) Energy consumption vs. distance

Fig. 4. Comparison between layered solutions and cross-layer solutions with respect to the distance.



(a) Delay vs. distance

(b) Energy consumption vs. distance

Fig. 5. Comparison between single-objective cross-layer solutions and joint-objective cross-layer solution with respect to the distance.

T_{e2e} or E_{e2e}) and cross-layer solution with a joint-objective (minimizing $\{0.5 \cdot T_{e2e} + 0.5 \cdot E_{e2e}\}$). The cross-layer solution with single objective of minimizing T_{e2e} has the smallest end-to-end delay with an increasing performance gain over the other two solutions. Similar to the energy consumption curves shown in Fig. 5b, the single objective solution outperforms for its focused criterion, while it exhibits degradation for the other. For example, the solution with the objective of minimizing energy consumption consumes the least amount of energy, but it introduces the largest end-to-end delay. On the contrary, the cross-layer solution with joint objectives of minimizing T_{e2e} as well as E_{e2e} achieves satisfactory performance in both end-to-end delay and energy consumption.

7. Conclusion

The IoT is the enabling technology for a plethora of long-awaited applications and business opportunities in the fields of domotics, e-health, real-time monitoring and logistics, among others, by allowing the seamless communication among all sorts of physical devices. There are many on-going world-wide research initiatives and standardization efforts, which aim at making the IoT a reality. However, the very high heterogeneity in hardware capabilities of things and QoS requirements for different applications limits the performance of classical layered protocol solutions and the existing cross-layer solutions for wireless sensor networks or ad hoc networks.

In this paper, we have explored the interaction among functionalities across the different layers in the protocol stack and developed a novel cross-layer optimization framework for the IoT. This framework captures these interrelations among different layers, the twofold heterogeneity of things and the impact of the Internet in the network performance. Moreover, we have proposed a cross-layer protocol to practically implement the proposed optimization framework. The results show that the proposed solutions outperforms existing layered solutions by exploiting the interactions among layers and the implicitly centralized network architecture of the IoT.

Acknowledgment

This work was supported by la Fundación Caja Madrid.

References

[1] D. Miorandi, S. Sicari, F. Pellegrini, I. Chlamtac, Internet of things: vision applications & research challenges, Ad Hoc Networks (Elsevier) Journal (2012).

[2] M. Weiser, The computer for the 21st century, Scientific American 265 (3) (1991) 94–104.

[3] ITU Internet Reports, The Internet of Things, International Telecommunication Union, Tech. Rep., 2005.

[4] L. Atzori, A. Iera, G. Morabito, The internet of things: a survey, Computer Networks (Elsevier) Journal 54 (2010) 2787–2805.

[5] S. Gusmeroli, S. Haller, M. Harrison, K. Kalaboukas, M. Tomasella, O. Vermesan, H. Vogt, K. Wouters, Vision and challenges for realising the internet of things, in: P. Friess, P. Guillemin, H. Sundmaeker, S. Woelfflé, (Eds.), European Commission, 2010.

[6] I.F. Akyildiz, J.M. Jornet, The internet of nano-things, IEEE Wireless Communications Magazine 17 (6) (2010) 58–63.

[7] M.C. Vuran, I.F. Akyildiz, Xlp: a cross-layer protocol for efficient communication in wireless sensor networks, IEEE Transactions on Mobile Computing 9 (11) (2010) 1578–1591.

[8] D. Pompili, I. Akyildiz, A multimedia cross-layer protocol for underwater acoustic sensor networks, IEEE Transactions on Wireless Communications 9 (9) (2010) 2924–2933.

[9] I. Akyildiz, X. Wang, Cross-layer design in wireless mesh networks, IEEE Transactions on Vehicular Technology 57 (2) (2008) 1061–1076.

[10] M. Conti, G. Maselli, G. Turi, S. Giordano, Cross-layering in mobile ad hoc network design, Computer 37 (2) (2004) 48–51.

[11] IEEE 802.15 Wireless Personal Area Networks Task Group 4. <http://www.ieee802.org/15/pub/TG4.html>.

[12] N. Kushalnagar, C. Schumacher, G. Montenegro, IPv6 Over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals, IETF, 2007.

[13] IETF Routing Over Low power and Lossy networks (ROLL). < https://datatracker.ietf.org/doc/charter-ietf-roll/>.

[14] Z. Shelby, ETSI M2M Standardization, 2009.

[15] Z. Fadlullah, M. Fouda, N. Kato, A. Takeuchi, N. Iwasaki, Y. Nozaki, Toward intelligent machine-to-machine communications in smart grid, IEEE Communications Magazine 49 (4) (2011) 60–65.

[16] E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, A. Chandrakasan, Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks, in: Proceedings of the 7th ACM Annual International Conference on Mobile Computing and Networking, 2001, pp. 272–287.

[17] M. Takai, J. Martin, R. Bagrodia, Effects of wireless physical layer modeling in mobile ad hoc networks, in: Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing, 2001, pp. 87–94.

[18] S. Kumar, V. Raghavan, J. Deng, Medium access control protocols for ad hoc wireless networks: a survey, Ad Hoc Networks (Elsevier) Journal 4 (3) (2006) 326–358.

[19] M. Vuran, I. Akyildiz, Spatial correlation-based collaborative medium access control in wireless sensor networks, IEEE/ACM Transactions on Networking 14 (2) (2006) 316–329.

[20] K. Akkaya, M. Younis, A survey on routing protocols for wireless sensor networks, Ad Hoc Networks (Elsevier) Journal 3 (3) (2005) 325–349.

[21] J. Al-Karaki, A. Kamal, Routing techniques in wireless sensor networks: a survey, IEEE Transactions on Wireless Communications 11 (6) (2004) 6–28.

[22] W. Jiang, H. Schulzrinne, Modeling of packet loss and delay and their effect on real-time multimedia service quality, in: Proceedings of NOSSDAV, Citeseer, 2000.

[23] X. Lin, N. Shroff, R. Srikant, A tutorial on cross-layer optimization in wireless networks, IEEE Journal on Selected Areas in Communications 24 (8) (2006) 1452–1463.

[24] K. Deb, Multi-objective optimization, Search Methodologies (2005) 273–316.

[25] R. Marler, J. Arora, Survey of Multi-Objective Optimization Methods for Engineering, vol. 26, Springer, 2004 (no. 6).

[26] J. Seybold, Introduction to RF Propagation, Wiley Online, Library, 2005.

[27] M. Vuran, I. Akyildiz, Error control in wireless sensor networks: a cross layer analysis, IEEE/ACM Transactions on Networking 17 (4) (2009) 1186–1199.

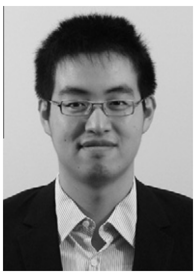
[28] S. Wicker, Error Control Systems for Digital Communication and Storage, Prentice-Hall Inc., 1994.

[29] Y. Sankarasubramaniam, I. Akyildiz, S. McLaughlin, Energy efficiency based packet size optimization in wireless sensor networks, in: Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications, 2003, pp. 1–8.

[30] W. Ye, J. Heidemann, D. Estrin, Medium access control with coordinated adaptive sleeping for wireless sensor networks, IEEE/ACM Transactions on Networking 12 (3) (2004) 493–506.

[31] P. Huber, The behavior of maximum likelihood estimates under nonstandard conditions, in: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1(1), 1967, pp. 221–233.

[32] D. Johnson, A note on Dijkstra's shortest path algorithm, Journal of the ACM (JACM) 20 (3) (1973) 385–388.



Chong Han received the Bachelor of Engineering in Electrical Engineering and Telecommunications from The University of New South Wales, Sydney, Australia, in 2011. Currently, he is a graduate research assistant in the Broadband Wireless Networking Laboratory (BWN Lab), School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA. He is pursuing his Ph.D. degree under the supervision of Prof. Ian F. Akyildiz. He is a student member of the IEEE. His current research interests are in the Internet of Things, Internet of Nano-Things, Electromagnetic Nanonetworks, Graphene-enabled Wireless Communication and Terahertz Band Communication Networks.

Josep Miquel Jornet received the Engineering Degree in Telecommunication Engineering and the Master of Science in Information and Communication Technologies from the School of Electrical Engineering, Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, in 2008. From September 2007 to December 2008, he was a visiting researcher at the Massachusetts Institute of Technology, Cambridge, under the MIT Sea Grant program. Currently, he is pursuing his Ph.D. degree in the Broadband Wireless Networking Laboratory (BWN Lab), School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, with a fellowship from "la Caixa" (2009–2010) and Fundación Caja Madrid (2011–2012). He is a student member of the IEEE and the ACM. His current research interests are in electromagnetic nanonetworks, graphene-enabled wireless communication, Terahertz Band communication networks and the Internet of Nano-Things.

Etimad A. Fadel received the Bachelors degree in Computer Science at King Abdul Aziz University with Senior Project title ATARES: Arabic Character Analysis and Recognition in 1994. She was awarded the Mphil/PhD degree in computer science at De Montfort University (DMU) with Thesis title Distributed Systems Management Service in 2007. Currently she is working as Assistant Professor at the Computer Science Department at KAU. Her main research interest is Distributed Systems, which are developed based on middleware technology. Currently she is looking into and working on Wireless Networks, Internet of Things and Internet of Nano-Things.

Ian F. Akyildiz received the B.S., M.S., and Ph.D. degrees in Computer Engineering from the University of Erlangen-Nürnberg, Germany, in 1978, 1981 and 1984, respectively. Currently, he is the Ken Byers Chair Professor in Telecommunications with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, the Director of the Broadband Wireless Networking Laboratory and Chair of the Telecommunication Group at Georgia Tech. Dr. Akyildiz is an honorary professor with the School of Electrical Engineering at Universitat Politècnica de Catalunya (UPC) in Barcelona, Catalunya, Spain and the founder of N3Cat (NaNoNetworking Center in Catalunya). He is also an honorary professor with the Department of Electrical, Electronic and Computer Engineering at the University of Pretoria, South Africa and the founder of the Advanced Sensor Networks Lab. Since 2011, he is a Consulting Chair Professor at the Department of Information Technology, King Abdulaziz University (KAU) in Jeddah, Saudi Arabia. Since September 2012, Dr. Akyildiz is also a FiDiPro Professor (Finland Distinguished Professor Program (FiDiPro) supported by the Academy of Finland) at Tampere University of Technology, Department of Communications Engineering, Finland. He is the Editor-in-Chief of Computer Networks (Elsevier) Journal, and the founding Editor-in-Chief of the Ad Hoc Networks (Elsevier) Journal, the Physical Communication (Elsevier) Journal and the Nano Communication Networks (Elsevier) Journal. He is an IEEE Fellow (1996) and an ACM Fellow (1997). He received numerous awards from IEEE and ACM. His current research interests are in nanonetworks, Long Term Evolution (LTE) advanced networks, cognitive radio networks and wireless sensor networks.