EURASIP Journal on Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH Open Access

Enabling direct connectivity between heterogeneous objects in the internet of things through a network-service-oriented architecture

Eli De Poorter*, Ingrid Moerman and Piet Demeester

Abstract

In a future internet of things, an increasing number of everyday objects are connected with each other. These objects can be very diverse in terms of the used network protocols and communication technologies, which leads to a wild growth of co-located networking technologies. Unfortunately, current consumer items are not designed to communicate with co-located devices that use different communication technologies. In addition, commercially available internet of things devices, such as sensor nodes, often use vendor-specific propriety network solutions. As a result, communication between these devices is only possible through the use of gateway nodes, resulting in inefficient use of the wireless medium. To remedy this situation, this paper discusses which features are required to integrate such a diverse number of heterogeneous objects into a single internet of things. In addition, the paper introduces the IDRA architecture, which is designed specifically to enable connectivity between heterogeneous resource-constrained objects. The IDRA architecture has the following advantages. (1) IDRA can connect co-located objects directly, without the need for complex translation gateways. (2) The architecture is clean slate, but supports backward compatibility with existing deployments. (3) Due to its low memory footprint, the architecture can be used in resource-constrained objects. Finally, the paper evaluates the performance of the IDRA architecture and discusses the feasibility of introducing IDRA in existing networks.

Keywords: Internet of things, Network architecture, Clean slate, Heterogeneity

I Introduction

New communication technologies are introduced and deployed on a regular basis. Even, common everyday objects nowadays come equipped with (wireless) communication possibilities. As a result, many sources have described an 'internet of things' view of the future, in which every object is connected with every other object [1] (Figure 1). By connecting these different objects, intelligent next-generation applications such as wireless building automation applications [2] or e-health applications [3,4] become possible.

However, as the number of communicating objects increases, so does the number of co-located communication technologies. When multiple networks operate in the same geographical environment, co-located networks overhear transmissions from multiple networks. Most often, overhearing these transmissions results in harmful interference and performance degradation, since the overheard transmissions cannot be interpreted by devices that are not part of the originating network. This is especially a problem in 'last mile' home and office networks. A typical example is the interference in the free license ISM band, which is used by a variety of communication technologies such as IEEE 802.1 (WiFi), car alarms, baby monitors, IEEE 802.15.1 (bluetooth), cordless DECT phones and IEEE 802.15.4 (zigbee) personal body area networks.

Even when co-located devices use the same radio technology, direct communication between devices is not always supported. For example, existing sensor and actuator networks often use propriety network technologies that are incompatible with technologies from other vendors, even though the devices use the same radio chip. To enable communication between networks from different vendors, or between devices that use different

* Correspondence: eli.depoorter@intec.ugent.be
Department of Information Technology (INTEC), Ghent University - IBBT, Gaston Crommenlaan 8, Bus 201, 9050 Ghent, Belgium

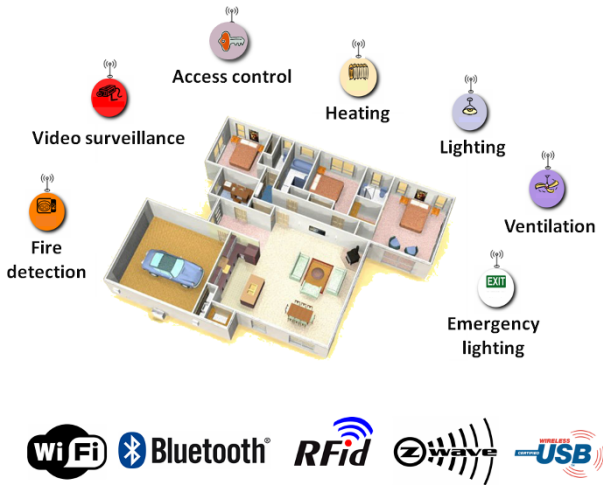

Figure 1 In the vision of the internet of things, everyday objects will all become interconnected using a variety of communication technologies. These objects can be used in intelligent applications such as wireless building automation or e-health scenarios.

network protocols, each network is connected to a different vendor-specific translation gateway. This translation gateway terminates the connection from one network and sets up a new connection to a second network. However, translation gateways break the end-to-end communication paradigm and are inherently complex to design, manage and deploy [5,6]. In addition, forcing all communication through the gateway results in additional packet overhead, which in turn leads to increased interference, lower throughput and a lower network lifetime for battery-powered devices. To remedy this situation, this paper describes how the IDRA architecture, which was previously implemented in [7], can be used to enable efficient direct connectivity between heterogeneous wired and wireless devices using different communication technologies.

The remainder of the paper is structured as follows. Section I gave an introduction on the vision of the internet of things and argued that current devices are typically not able to efficiently connect with co-located devices that use different communication technologies. Section II discusses this topic further by giving a thorough overview of the requirements that should be solved to realize a more efficient internet of things. Related work is given in Section III where existing architectures are listed and the advantages and disadvantages of each of these approaches are discussed. Section IV gives a high-level overview of the proposed IDRA architecture and discusses how this architecture fits in the vision of an internet of things. Afterward, Section V describes how IDRA can be used to support two typical internet of things use cases: (1) supporting backward compatibility with legacy networks and (2) bridging networks using different communication technologies. In addition, the performance of the architecture is evaluated and the economic viability of introducing IDRA in existing networks is discussed. Finally, Section VI concludes the paper.

II Requirements of a future internet of things

Several sources already listed a large amount of challenges that must be overcome to support an all-encompassing connectivity between objects [8-10]. Amongst the listed internet of things requirements, the following four requirements can typically be found: *providing network connectivity*, *supplying content*, *easily managing the network* and *being extensible*.

A Network connectivity

The first and foremost requirement of the internet of things is to provide connectivity between any type of object: from machine to machine, from person to machine or from machine to person. The involved objects differ in terms of both *communication technologies* and *capabilities*.

- Co-located devices that wish to exchange information often use *different communication technologies*. Any architecture suitable for an internet of things must be able to efficiently support communication between devices, even if they use different protocol stacks, different radio frequencies, different communication technologies and different packet types. In addition, many objects will be equipped with *multiple communication interfaces* such as a IEEE 802.15.1 bluetooth interface and a IEEE 802.11 WiFi interface.
- In addition, the devices will have *different hardware and software capabilities*. Internet of things devices range from high-end PC devices to low-end battery-powered embedded devices. Even networks that use only a single communication technology can consist of heterogeneous nodes. For example, networks used for wireless building automation or industry monitoring used both resource-constrained embedded devices and high-end controller PCs. Using the traditional OSI reference stack, this heterogeneity is difficult to support: each communicating device requires exactly the same protocol stack. However, the communication stack of the powerful devices should not be limited by the capabilities of the most restrictive objects.

B Content and context

The internet of things will also become increasingly content oriented [11]. Users expect to be able to retrieve

any content, from any device. This includes content that is part of the public domain (dictionaries, transportation information, etc), but also private content such as e-mails, personal media and home information such as the current home temperature. Some of the challenges that need to be overcome are the following.

- Location awareness is increasingly important. This includes awareness of the personal surroundings, the tracking and positioning of objects, as well as support for user and object mobility. For example, in applications that require vehicle-to-vehicle communication, all networked objects are mobile.
- The context associated with information is also increasingly important. Future devices require mechanisms to easily associate metadata with content, such as the originating location, information about the producer of the content and the content description. This metadata should also be included at the network level. For example, by associating metadata with packets, the location of the packet destination can be added to packets to facilitate geographic routing.
- Media, security and emergency content often has strict real-time requirements. As such, mechanisms are required that provide quality-of-service solutions that span several networks.
- Finally, mechanisms that control access to information, and that provide privacy, security and anonymity should be supported over several network layers and physical network boundaries.

C Network management

To be commercially viable, a future internet of things should be easy to set up and use even for non-network experts [9].

- Self-configuration [12,13] solutions are required that automatically set up and configure devices. This includes solutions for automatic address allocation and automatic detection of configuration inconsistencies.
- The internet of thing can be fully autonomous: in the absence of human intervention the network should be able to take its own decisions by detecting potential (network) problems and proposing solutions based on artificial intelligence algorithms.
- Finally, to ease network management, underlying network solutions should become more 'invisible'. To be able to reuse network solutions in different contexts, underlying communication interfaces should be presented in an abstract and ubiquitous way. However, this abstraction should not hinder the collection of detailed metadata (such as the radio frequency) that is associated with the used technology.

D Network extensibility

Finally, a sustainable internet of things architecture should not only be robust, but needs to cope with continuously changing application requirements and changing hardware capabilities.

- It should be easy to install new software so that new applications can be deployed on previously installed devices.
- Networks should become more 'service-like', where network services can be added and reconfigured according to the applications needs.
- In addition, to support ongoing innovation, it should be possible to change any protocol characteristics such as the addressing schemes (for example from IPv4 to IPv6), the used packet types, the communication technology or the security mechanisms without making any changes to the network protocols themselves. Ideally, these changes should be possible at runtime, without breaking the active communication between devices.

III Related architectures

There is a need for new protocol architectures that inherently support these requirements, such as reconfigurability and support for heterogeneity, over all network layers [14]. This related work section gives a non-exhaustive overview of architectures that are designed to support *direct network connectivity* between *heterogeneous networks*. Two main approaches are discussed: (1) incremental 'evolutionary' architectures and (2) clean slate 'revolutionary' architectures.

A Evolutionary internet of things approaches

Advocates of an evolutionary approach to a future internet of things create new architectures by reusing as many components as possible from existing networking solutions. In their vision, the current internet should 'evolve' into an architecture that is more suited for an internet of things. A first approach is to gradually improve the existing communication stacks, replacing one function at a time, whenever the need arises. A typical example is the introduction of IPv6 addresses to replace current IPv4 addresses. For this approach to be successful, architectures should be easily extensible. Otherwise, this approach results in a difficult adoption of new technologies, as shown by the problematic transition into IPv6 we are witnessing at the moment.

An alternative evolutionary approach is the use of virtualized network components. Network virtualization [15] is used to present underlying network layers in a uniform way toward a high-level application. Different devices are connected by forming virtual networks on top of existing networks: logical links are created between distributed systems using native internet routing and standard IP addresses. Well-known examples are Virtual Private Networks (VPN) [16] or peer-to-peer applications [17]. The FP7 4WARD project [18] considers virtual networks to be a fundamental part of the design of future internet devices. The project includes virtualized network solutions for in-network management, generic connectivity and content-centric information objects [19]. Similarly, the MAGNET project [20,21] offers network virtualization at both layer 2 and layer 3, whereas the ITEA2 usenet project [22] focuses on network virtualization for machine-to-machine communication. One well-developed solution is VPAN [23], in which self-organizing and self-maintaining overlay networks are created that provide a shielded and trusted environment for networked applications that share a common context.

In the context of an internet of things, network virtualization can be viewed in two ways [24]. First, these techniques can be used as a tool for evaluating new disruptive architectures on a large scale using existing networks. Secondly, virtualization can be regarded as a fundamental part of next-generation architectures, whereby multiple 'overlay' networks coexist by creating different logical networks for communication purposes [25]. However, for directly connecting heterogeneous networks (such as described in our vision of the internet of things), the use of virtualization techniques has the following disadvantages. (1) Network virtualization is not yet proven to be highly scalable, since setting up an overlay network is often difficult and time-consuming. (2) Virtualization techniques are often too complex and inefficient to be implemented on resource-constrained embedded devices. And (3) virtualization techniques are often too high in the protocol stack to efficiently bridge networks that use different communication technologies.

B Revolutionary internet of things approaches

Opponents of the evolutionary approach emphasize the need for a redesigned, clean slate architecture that inherently copes with next-generation network challenges [14,26], sometimes even abandoning IP-based addressing in favor of different addressing schemes. Clean slate initiatives are not always meant to be used directly in new devices, but can be used to sketch a revolutionary new perspective, which can then be brought into existing networks. Several approaches have been proposed.

(i) *Database centric architectures* hide the heterogeneity of underlying networks by only allowing access to network information using database operations. For example, the SENSEI project [27] solves the inaccessibility of low-resource end devices by collecting all data from the end devices and making it available in a centrally accessible database. Unfortunately, this approach often results in significant network overhead.

(ii) *Content-centric architectures* focus on describing the information that is exchanged between networks. For example, the SemsorGrid4Env project [28] focuses on the development of a semantic middleware layer. At the network layer, network protocols are implemented semantically using a 'descriptive language' [29] that focuses on functionality rather than implementation. Unfortunately, support for directly connecting different networks at the lower network layers is still lacking.

(iii) *Cloud computing approaches* try to offload resource intensive tasks to more capable nodes. Typically, cloud computing can offer infrastructure, platforms or software as a service to less-capable devices [30,31]. Since cloud computing is regarded as a high-layer service, this approach does not solve connectivity challenges.

(iv) *Service-oriented architectures* (SOAs) use loosely coupled software entities that implement a single software function. These software services are dynamically combined to form ad hoc applications. In regards to the internet of things, SOAs have two main disadvantages [32]: (1) SOAs focus mainly on higher layers rather than solving network issues and (2) the technologies used to realize service-oriented architectures, such as ML, SOAP, Web Services or BPEL, are often not suited for use in resource-constrained devices.

(v) *Modular approaches* have also been proposed, whereby the protocol stack is divided into different modules which can be combined to create new network protocols with different functionalities. As such, these approaches can easily integrate new network technologies by updating a single module. Modular frameworks, such as SNA [33], can be used to design new network layers. However, most existing modular frameworks compile these distinct modules into a static network layer. In addition, current modular approaches do not focus on supporting connectivity in heterogeneous environments. Thus, although promising, existing modular approaches offer no additional run-time flexibility when compared to traditional layered approaches. In contrast, the NewArch project [34] discusses how a flexible internet architecture can be created whereby different roles can dynamically be combined at run-time to form 'heaps' [35] which can be adapted to the needs of the network. Unfortunately, the project did not result in a practical proof-of-concept implementation.

C The need for new architectures

As shown in the previous sections, several research projects are currently involved with the design of new network architectures. However, an economically viable solution might still be a long way off:

- Though several research projects are currently involved with future inter-net research, most of these efforts focus on the design of a (high-speed) future internet backbones. These solutions are not suitable for use in resource-constrained environments.
- As cited in [36] 'too many future internet proposals are just extensions of existing protocols or architectures'. As such, these proposals lack the innovation to cope with specific internet of things challenges.
- Finally, too many proposals remain 'paperware': there is a definite lack of implemented prototypes [14,37].

As such, more practical implementations are needed before a decision can be made regarding the feasibility of an all-encompassing internet of things solution. Especially for resource-constrained devices, there is still room for several improvements. More specifically there exists not yet a simple architecture that

- enables optimized communication **at a network and link level** between co-located heterogeneous networks without the use of complex translation gateways;
- has been implemented and evaluated as a prototype in a large-scale experimental setting;
- is compact enough to fit even on low-resource embedded devices;
- is fully clean slate, but is also backward compatible with legacy networks;

In the following section, we will discuss how our IDRA architecture fills this gap.

IV The IDRA architecture

IDRA [7] was originally developed to support next-generation applications on wireless sensor networks. Wireless sensor networks (WSNs) consist of resource-constrained embedded devices which are wirelessly connected to each other in an ad hoc fashion. Next-generation applications for WSN include topics such as wireless building automation, distributed wireless health monitoring, industrial process monitoring and disaster intervention. Each year, the WSN research community develops new and optimized hardware devices, communication technologies, network protocols and applications. To cope with such a varying environment, IDRA has built-in solutions to support heterogeneous devices (in terms of hardware and communication technologies) and to support evolving services and applications.

There are many similarities between the requirements of WSNs and those of a more general internet of things. This section gives a general overview of the IDRA architecture and discusses which design choices can also be useful in the context of the internet of things.

A Network protocols as services

The OSI reference architecture [38] uses a layered protocol stack whereby all network functionality is assigned to a specific protocol layer. For example, the routing layer includes functionality for providing reliability, for duplicate detection, for retransmissions, etc. In contrast, in IDRA it is possible to implement these different network functions (such as addressing, naming, CRC calculation, routing, etc.) each in a simple, standardized, technology-independent component called a 'network service'. Network services can implement either a full network protocol (such as routing) or a simple function (such as duplicate detection). Each component implements the same interface and functions independent, without direct interaction with other components. New services can be added whenever there is a need for them. For example, a localization service can be added when an application is run that requires location information. This way, more advanced network services can be composed by combining elementary network services according to the needs of the network (Figure 2c). A default 'call sequence' is included that indicates the order in which the network services should be executed. By adding new network services or by changing the call sequence, the network behavior can be changed. Some examples of network services are:

- a localization service inspects the RSSI of received packets so that it can provide the network with accurate location information;
- a MAC service is responsible for controlling the timing and sending of packets;
- a topology service decides from which neighboring devices packets can be received;
- a synchronization service delivers a network-wide reference clock;
- a reliability service is responsible for the retransmission of packets;
- a management service collects and makes available network statistics such as the background noise level and the number of failed transmissions.

Initially, such a network service-oriented approach is compatible with a layered approach: fully implemented

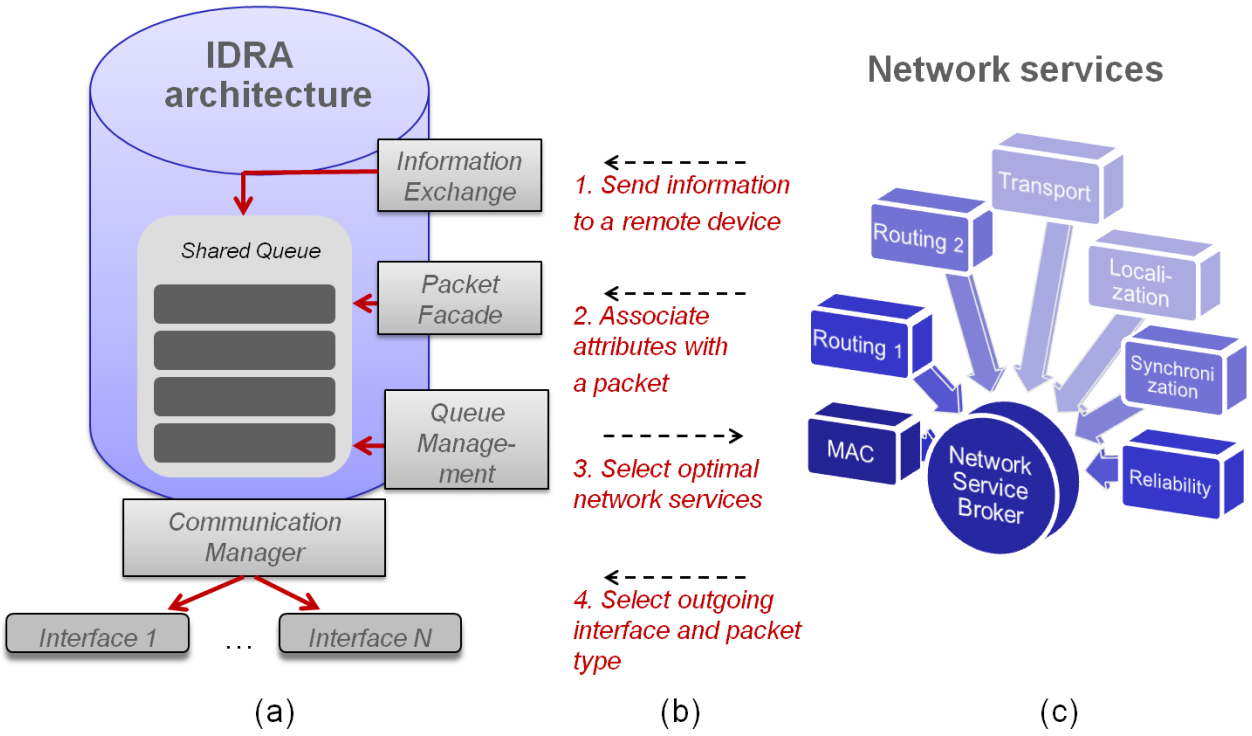

Figure 2 Conceptual presentation of the IDRA architecture. **a** The IDRA system is responsible for packet creation, packet storing and packet interactions. **b** Interactions between the network services and IDRA are mainly descriptive in nature. **c** Network services can be added dynamically according to the needs of the device.

network layers can register themselves as network services. A transition toward a network-service-oriented architecture can occur gradually by standardizing a further decomposition of the protocol layer into multiple well-defined network services.

In the context of the internet of things, a network-service-oriented network architecture has the following advantages.

Pervasiveness

Separating network functionality into different services results in a lower memory footprint, since (1) network services are only added on a per-need base and (2) functionality is not duplicated in several protocol layers. As such, this approach is well suited for networks that contain resource-constrained devices.

Extensibility and maintenance

A second advantage of a service-oriented network architecture is the ease with which the network copes with future developments. New applications are supported at a network level by plugging in the appropriate network services (for example: an advanced encryption service can be added to process all packets from an online banking application).

Transparency

Rather than directly interacting with a multitude of different communication technologies, such as IEEE 802.15.4 (Zigbee), IEEE 802.15.1 (Bluetooth), IEEE 802.11 (Wi-Fi), LAN or UMTS, the communication manager from Figure 2a translates the communication capabilities of the underlying communication technology in terms of the services they can provide, such as average reliability, energy cost, etc.

B Information driven network services

To limit the dependence of network services on specific technologies, network services are technology independent. Rather than creating technology-dependent packets to exchange information, network services hand over to the IDRA system any information they wish to send (Figure 2b, interaction 1). Example information exchanges are a route request, a web page request or a packet acknowledgment. IDRA creates the actual packet, encapsulates the information in the payload and stores the resulting packet in a system-wide queue. IDRA can be configured to create or interpret any packet type (see Section VI-C). Thus, instead of packet-based sending, IDRA offers *information-based communication*.

Similarly, all interactions with the communication interfaces are descriptive. For example, a MAC protocol can describe when and how each packet is allowed to be transmitted (e.g., the maximum tolerated background noise, the scheduled sending time, the radio frequency, etc) and how the communication interfaces should be configured

(listening frequency, power state, etc.). A conflict resolution scheme is used to detect conflicting settings and inform the MAC service of undesired behavior. As a result, multiple MAC services can reside on the same node, each with one or more associated communication interfaces.

Delegating all technology-related functions, such as packet creation, packet manipulation, packet sending and buffer provisioning, to the IDRA architecture has the following advantages.

Hardware heterogeneity

Tasks which are typically duplicated in several network layers (ie: packet creation, packet manipulation, packet sending and buffer provisioning) are delegated to the system, thus avoiding code redundancy. As a result, the overall code size is reduced, making it possible to support a large number of services even on resource-constrained devices.

Ease of use

Since network services need to consider only 'information exchanges', the development of network services is simplified.

Transparency

The same information exchange mechanism is used for all network services, independent of which packet type will be created and which communication interface will be used. The complexity of low-level operations (buffer management, packet construction, etc) are thus hidden from the network services. This way, network services are not technology dependent, which promotes reuse of network services in different contexts.

C Decoupling of protocol logic and packet representation

Since the network services do not create the packets, they have no knowledge about the header structure of received packets. To retrieve information about created or received packets, network services interact with packets through a 'packet facade' (Figure 2b, interaction 2). Through this packet facade, standardized *packet attributes* (metadata) can be added, updated or requested. This metadata can represent header fields such as 'destination', 'quality-of-service' or 'time-to-live', but can also be used to describe extra context information such as the packet origin or destination location, the packet owner or additional descriptive information about the packet.

To interpret the structure of created or received packets, the packet facade uses one or more 'packet descriptors'. These packet descriptors describe at which header offset each packet attributes should be stored (Figure 3). This way, any packet type can be generated or interpreted, as long as the correct packet descriptor is available. Packet attributes that do not have a fixed location in the packet header are stored sequentially in the payload in the form of type-length-value (TLV) elements. Since there is no direct interaction between network services and packet descriptors, the packet type can be changed without any changes to the used network services. Decoupling the protocol logic from the packet structure has several advantages.

Packet heterogeneity

Since any packet type can be interpreted as long as the correct packet descriptor is available, it is possible for multiple packet types to reside on a single node, transparent for the network services.

Legacy support

By providing the packet descriptors of legacy devices, IDRA services can interpret packets from legacy networks and interact with legacy packet types.

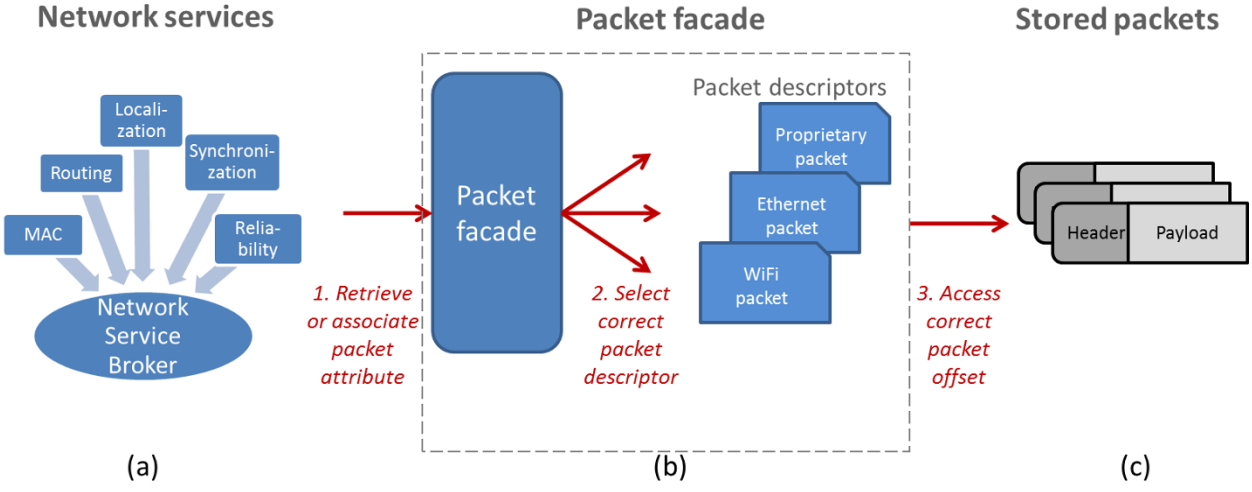

Figure 3 Network services can transparently interact with any packet type. **a** Network services can associate metadata with, or retrieve metadata from, stored packets using the packet facade. **b** Only the packet facade requires knowledge about the packet format. As long as the correct packet descriptor is available, the packet facade knows how and where metadata is stored. **c** Finally, the packet facade accesses the correct header offset or the packet payload.

Context awareness

Any type of context information can be associated with a packet in the form of a packet attribute. This promotes the development of new network services which rely on advanced packet information such as ownership or visibility rights. Metadata can also be used to facilitate mobility solutions [39].

Hardware heterogeneity

Using a packet facade, network services do not need to strip the protocol headers from received packets to interact with packets. Thus, when non-essential network services are omitted from devices with low resources, the remaining network services can still interpret the received packets. In addition, packet attributes remain associated with a packet even if network protocols are omitted at intermediate nodes. Thus, when lightweight nodes are provided with simpler versions of the network services, these simpler services can inspect the packet attributes that were added by more advanced network services.

Future proof

Network services can be implemented independently from the representation of the packets. As a result, reuse of network services is promoted. To change the packet structure or support new packet structures, only the packet descriptor needs to be updated. All other network services remain unchanged.

D Queue management

Depending on the required network performance, different queuing systems can be used in IDRA. To reduce the memory footprint of the queues, the current implementation of IDRA uses a single, system-wide queue for storing packets. Arriving packets are stored once in the shared queue and remain there until processing is finished. This approach limits the number of copy actions of the packets. Network protocols can interact with any of the packets from the shared queue using the packet facade. Since network services are not responsible for queue management, the complexity and memory footprint of the network services are reduced. The advantages of using a single, system-wide queue are the following.

Simplicity

In layered architectures, each network layer requires overprovisioning of its provided buffers to ensure that all packets can be stored. Using a shared queue approach, this overprovisioning is required only once. As a result, network services are simpler and have a small memory footprint.

Network management

Using a single queue ensures that the system can monitor all available packets. This eases the gathering of real-time network statistics.

QoS management

The shared queue has an associated QoS module. This QoS module has a global view on the number of packets, their current processing state and their expected delay. The QoS module can drop packets and selects which packets are processed or transmitted first. Since the QoS module only interacts with the queue, it can provide basic, protocol-independent QoS which can transparently be combined with any IDRA network service.

E Network service broker

Network services can be dynamically added, removed or updated according to the needs of the network. Rather than having a strict execution order (such as in layered networks), network services are activated only when they are needed. Currently, IDRA implements a simple service broker. Each network service registers itself using 'filters' which describe the function of the network service (e.g., routing, localization, etc). In addition, the filters specify for which types of packets the network services can be used (Figure 2b, interaction 3). For example, a QoS aware routing service can register itself for routing high-priority packets ('QoS attribute higher than 5'), a georouting service can be used for routing when location information is available ('location attribute is available') and a multicast routing service registers itself for routing packets to multicast addresses.

In high-end devices, intelligent service discovery mechanisms can be used to detect the capabilities of the network services, and to automatically select and configure the appropriate network services depending on the application requirements. For example, to support a 'fire alarm' or 'emergency reporting' service, the network broker can disable energy-efficient routing in favor of an optimized low-delay routing service. Or a key-distribution service can be activated before a device is allowed to join an existing network. As such, a dynamic network service broker promotes a more flexible internet of things.

Self-configuration

Devices can change their own behavior by plugging in new network services when required. Intelligent self-configuring networks can use the service broker for self-adaptation and for automatic network upgrades.

Hardware heterogeneity

Devices with less capabilities can be configured to use simplified execution sequences which contain less network services. Alternatively, they can negotiate with neighboring nodes to use simplified versions of the required network services.

Legacy support

When interacting with legacy neighboring devices, the service broker can be configured to execute legacy

network services in a typical layered order (e.g., MAC, routing, transport, application). This way, network service-oriented devices can coexist with traditional layered devices.

F System-wide aggregation

Many information exchanges (such as status updates, low-priority monitoring information or delay-tolerant measurements) between different devices do not need to be transmitted immediately. In the IDRA system, network services can include information about the maximum tolerated delay when handing over information exchanges to the IDRA system. Time sensitive parameters are immediately encapsulated in a packet, but all other parameters are temporarily stored in a central repository. Whenever a packet is relayed through the node, all information parameters with the same 'next hop' or 'destination' attribute are added to the packet. Delay-tolerant parameters can remain in the waiting space for up to a per-parameter predefined period of time. If no data have been relayed within the allowed waiting time, the system generates a new packet which combines all parameters that have the same destination. As a result, information exchanges from all network services using IDRA can be combined. To avoid high end-to-end delays, the current implementation only delays the parameters in the waiting space of the initial node: packets are not further delayed in intermediate nodes. Since the aggregation is part of the IDRA architecture, the aggregation approach can be changed or optimized depending on the network requirements, without any changes to the network services. The advantages of aggregation at an architectural level are the following.

Reduced interference

By limiting the number of transmissions, the overall wireless interference decreases, resulting in more optimal use of wireless bandwidth.

Increased throughput

It has been shown that the use of small packets has a negative influence on the maximum throughput of transmission systems, in particular for wireless networks. By combining multiple information exchanges in a larger packet when possible, the overall throughput increases.

Increased energy efficiency

Finally, for devices powered by small batteries or by energy scavenging, the use of a radio for transmitting packets results in a serious decrease in network lifetime. By limiting the number of transmissions, the time before battery replacement increases.

V Advanced IDRA use cases

The previous section gave a high-level overview of different IDRA concepts and discussed their relevance in the context of the internet of things. This next section describes in more detail on how IDRA can support two important internet of things use cases: (1) supporting backward compatibility with existing networks and (2) transparently bridging a diverse number of (co-located) wired and wireless communication technologies.

Backward compatibility

Before an all-encompassing internet of things exists, there will be a need for a transitional period, whereby internet of things devices transparently coexist with existing (legacy) networks. As an example, consider a scenario in which an existing corporate Wi-Fi mesh network is extended with new internet of things Wi-Fi devices that use a next-generation protocol stack. Most clean slate solutions solve this by setting up a new network that is fully separated from the existing legacy network (Figure 4). Communication between the legacy nodes and the state-of-the-art devices typically requires the development of a complex gateway device, in which all network protocols are translated. In contrast, when IDRA is used on the new devices, nodes can communicate directly with any existing Wi-Fi node resulting in an optimized network performance.

Heterogeneous networks

In the future, the internet of things will be accessible using a large number of communication technologies. As an example, consider an industry building where the following communication technologies are used: a wireless entrance and security system is installed, wireless internet access is provided through Wi-Fi access points, the wireless company phone network uses DECT, a company LAN network is used for high-speed connections and finally an expensive UMTS connection is used to provide connectivity to remote parts of the industry terrain. When a resource-constrained Body Area Network (BAN) is introduced to monitor the health of the employees, the BAN nodes should be able to connect directly to all existing co-located networks, without the use of a remote gateway (Figure 5). When using IDRA, it is not necessary to install a full protocol stack for each of these diverse communication technologies. As such, IDRA can implement 'always best connected' (ABC) solutions even on resource-constrained internet of things devices.

To realize these use cases, IDRA has built-in features that are able to cope with the following network challenges: (1) heterogeneous (legacy) devices can use different packet types, (2) heterogeneous (legacy) devices can use conflicting medium access mechanisms and (3) heterogeneous (legacy) devices can use different higher-layer network protocols

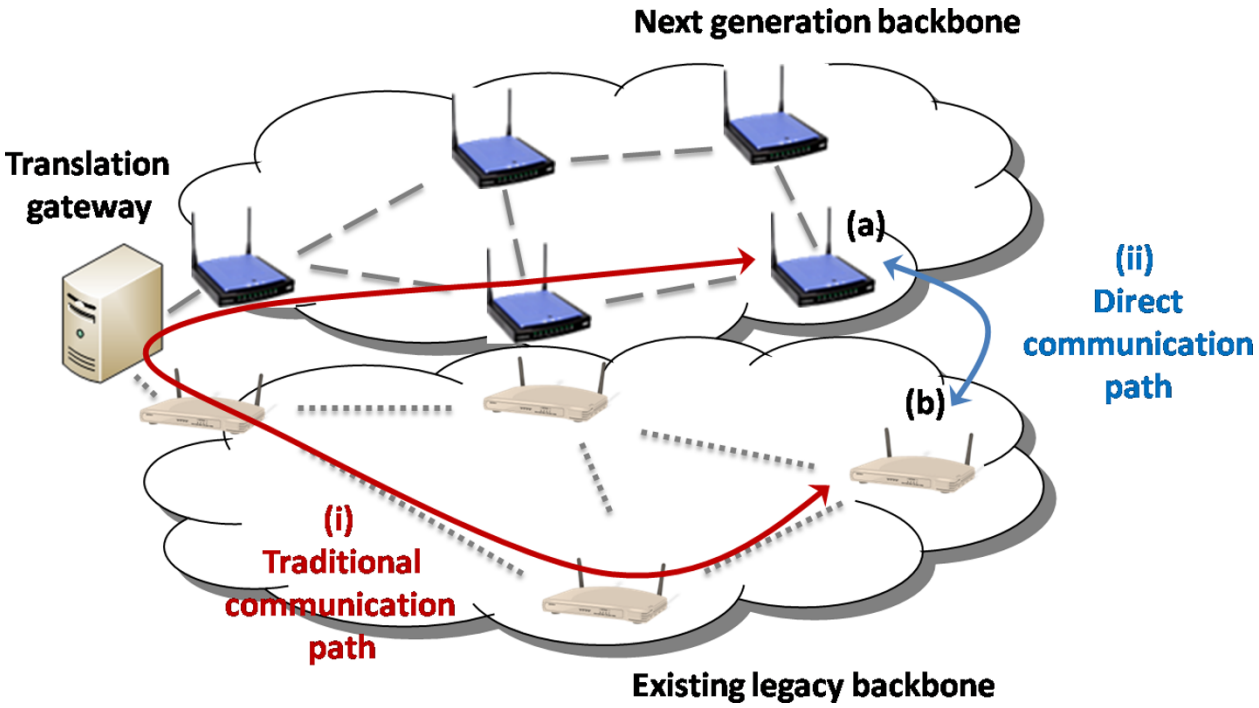

Figure 4 The coverage of an existing legacy network is expanded by installing an additional next-generation backbone. (*i*) Using existing technology, all communication must pass through a translation gateway, resulting in suboptimal use of the network. (*ii*) A next-generation IDRA network can converge with existing networks and use direct communication paths, thus prolonging the operational lifetime of legacy networks.

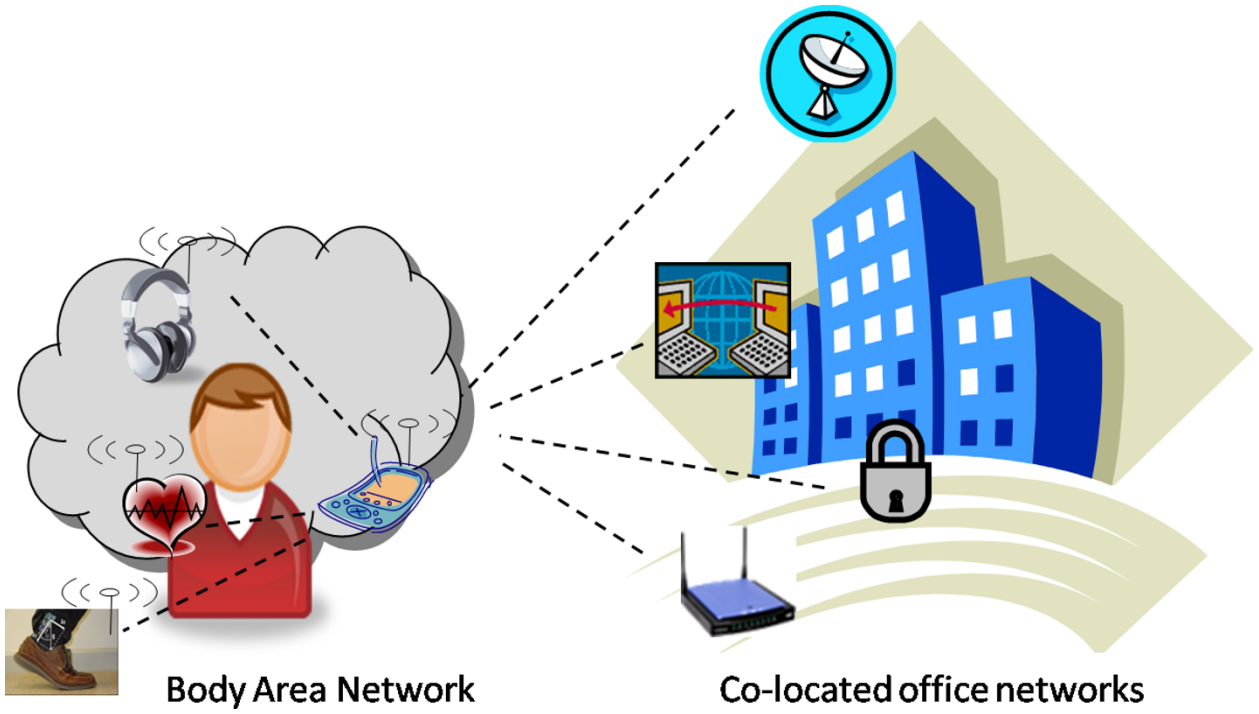

Figure 5 A resource-constrained personal Body Area Network (*BAN*) monitors the health of an employee. For efficient communication, the BAN should be able to communicate directly with all co-located network technology such as wireless entrance and security control, UMTS, Wi-Fi and DECT.

A Connecting devices that use different packet types

In a heterogeneous environment, multiple packet types can transparently reside on the same node at the same time. The IDRA packet facade is able to interpret any incoming packet type, as long as the correct packet part descriptors are available. To this end, IDRA includes a packet identification service that indicates which packet descriptors should be used to interpret incoming packets. To identify *incoming packets*, one of the following identification services can be used.

- Networks that utilize multiple communication technologies can associate a packet type with each interface (e.g., an 802.11 packet type for the Wi-Fi interface, an 6LoWPAN packet type for the IEEE 802.15.4 interface, etc).
- Alternatively, in case multiple packet types can arrive on the same interface, a publicly available, standardized packet type can be added as a unique packet identification field to each outgoing packet.
- If the radio offers hardware address recognition features, the address of the sending node can be identified. In this case, the neighbor table is used to describe the expected packet type of each neighboring node. This approach is not possible for networks that use non radio-compliant MAC headers or networks that include address-free communication interfaces (such as USB interfaces).
- Finally, a last option is to compare incoming packets with the descriptors of existing packet descriptors using bitmap operations.

This wide range of identification methods ensures that network designers can always choose the most optimal method for identifying incoming packets. The IDRA system automatically drops all packets that are not recognized by any of these packet identification services.

To select the correct *outgoing* packet type, a configurable shared neighbor table is used. For each of its neighbors, an entry is available in the shared neighbor table that indicates the preferred packet type, routing protocol and MAC protocol. IDRA will automatically select the correct MAC protocol and sent the packet over the correct radio interface. This shared neighbor table can be configured at run-time or at compile-time.

In heterogeneous networks, the outgoing packet type might be different from the incoming packet type. *Packet conversion* occurs when an outgoing packet must be transmitted to a neighbor that is associated with a different packet type. When packet conversion is required, the packet facade is used to create a new packet of the correct type. Next, the packet facade extracts all packet attributes from the original packet (thus dismantling the original packet). Finally, the packet facade is used to add all extracted packet attributes to the newly created packet. The conversion process is fully transparent for the network protocols: the network protocols cannot distinguish the new packet from the original packet.

B Connecting devices with different MAC protocols

IDRA can also support communication with (legacy) devices that use different MAC protocols. To this end, both the legacy and the new MAC service can be registered to manage the same network interface. Using the shared neighbor table, the IDRA service broker will automatically use the correct MAC service when sending packets to the legacy nodes. Also, using packet filters incoming packets can be directed toward the correct MAC protocol based on their packet type or other packet attributes. Finally, IDRA includes several simple algorithms for resolving MAC conflicts that occur when multiple MAC protocols want to manage the same radio interface. For example, the radio will only be disabled when all MAC protocols have requested a low power radio state.

C Connecting devices which use different routing protocols

To cope with different routing protocols, the legacy routing protocol can be installed on the IDRA device as an additional routing service. The dynamic network service broker can be configured to use this routing protocol for any packet that goes to (or comes from) a legacy device. In addition, by providing the correct packet descriptor, IDRA nodes can interpret legacy headers to retrieve the source and destination of each (legacy) packet. As such, IDRA devices can route legacy packets to their destination using a new state-of-the-art routing protocol, without changing the packet structure. This way, next-generation IDRA devices can be used to transparently route packets from legacy nodes.

D IDRA performance

According to [14,40], the development of actual prototypes is crucial to prove the merit of future protocol architectures. The concepts above have all been implemented in a proof-of-concept architecture that is available at http://idraproject.net. The following performance metrics are important in the context of the internet of things.

Memory overhead

The memory overhead of the overall IDRA architecture (including queue management, the network service broker, the packet facade, a IEEE802.15.4 packet descriptor and an IPv6 packet descriptor) is less than 30 kB ROM and 4 kB RAM. As such, the concepts can be used even on resource-constrained devices.

Processing overhead

The processing overhead of packet identification is negligible when using fixed packet identifiers that are added to each packet. Packet conversion is more complex, but should only be supported in devices that border two different networks. On a resource-constrained TMoteSky device [41] using a 8 MHz processor, the current packet conversion implementation requires a processing time of less than 1 ms.

Network services

By delegating common tasks, such as buffer management, packet creation, QoS management and aggregation, IDRA network services require up to a factor 2-10 less memory than traditional network protocols. Table 1 gives an overview of the memory requirement of several network services that are currently available [42]. Due to the extremely low memory requirements of IDRA network services, it is feasible to combine a multitude of network services, even on resource-constrained nodes.

Feasibility

Finally, the architecture has been evaluated in the DEUS project [43] using a large-scale testbed of 200 TMoteSky nodes and two real-life network deployments (the arts center 'Vooruit' and a home for the elderly). Devices were installed that use different routing protocols (DYMO, HYDRO or AODV) [44]. Depending on their neighbors and the packet meta-data, the nodes automatically selected the appropriate routing protocol, thus enabling direct connectivity between these different devices. As a result, even on such a large scale, devices running IDRA are capable of efficient direct communication by using packet conversion and a dynamic network service broker on each intermediate hop.

Converting existing layered network protocols to IDRA services typically requires two changes to existing implementations. (1) Whenever a legacy network protocol generates a packet, the protocol should hand over an information exchange to the IDRA architecture instead. (2) Whenever a network protocol interacts with a packet, this interaction should take the form of retrieving or updating a packet attribute through the packet facade. IDRA currently includes several network services such as routing, MAC, topology control, duplicate detection, packet identification, packet ownership, quality-of-service and localization services, which can all be combined as required by the network. In addition, network negotiation, network detection and network management services are under development. IDRA is freely available online as an open-source project at http://idraproject.net.

Table 1 Memory requirements (in bytes) of the currently available IDRA network services

Network service	ROM	RAM
Link level duplicate detection	460	26
Label management	294	4
AODV routing protocol [45]	2,664	240
HYDRO routing protocol [46]	1,924	692(+28 per route)
Positioning algorithm [47]	1,584	1,832
Low-power listening MAC protocol [48]	822	176
Synchronized MAC protocol [49]	1,126	184

E Business aspects

Apart from technical aspects, one of the main drivers behind innovation are marketable results. In regards to IDRA, we identified the following economic advantages.

- Due to its low memory and processing requirements, IDRA can reduce the hardware costs of involved devices.
- Low-cost end devices can be included in IP networks since IP packets can be generated and processed even without a full protocol stack.
- By activating the built-in aggregation service, increased wireless throughput can be provided and battery-powered devices have a longer functional lifetime.
- IDRA transparently supports an 'always best connected' strategy between different technologies at all network levels. Network cooperation can save the consumer money. For example: when watching videos on a cell phone, rather than using an expensive 3 G network, the cell phone can connect with a body area network (BAN) using the bluetooth connection. The BAN, in turn, can make a connection with a nearby Wi-Fi gateway to provide cheap internet access.
- The architecture supports the concept of dynamically plugging in new network services whenever required. This paves the road for pay-per update services and stimulates the development of companies supporting network services.
- Finally, IDRA can be used to deploy next-generation networks while still supporting existing legacy networks.

These advantages can be exploited by business innovators even before the commercialization and roll-out of a large-scale internet of things.

VI Conclusions

Our everyday environment is equipped with an increasing number of communication technologies, from high-speed internet backbones to 'last mile' access and cable replacement technologies such as WiFi, bluetooth or UMTS. This trend will likely not change, prompting the need for internet of things architectures that inherently

cope with this diversity. This paper gave an overview of existing promising architectural approaches. However, at the moment, none of these offers a solution to enable efficient direct communication between co-located resource-constrained devices that use different communication technologies.

This paper argues that this gap can be filled by the IDRA architecture. To motivate this, a broad overview of the IDRA architecture was given, and it was motivated how the discussed concepts fit within the vision of the internet of things. The following contributions of IDRA toward a future internet architecture were discussed in greater detail. (1) IDRA is a clean slate architecture, but is backward compatible with legacy networks. (2) The complexity of IDRA is low enough to implement even on resource-constrained devices. (3) IDRA enables direct communication between co-located networks without the use of complex translation gateways. In addition, (5) IDRA copes with changing network and application requirements by introducing a dynamic service broker responsible for selecting the most optimal network services. And finally, (4) IDRA has fully been implemented and evaluated, which proves the feasibility of the proposed concepts. As such, the IDRA architecture is a promising candidate to connect heterogeneous next-generation networks in a straightforward way, while supporting many of the requirements of the internet of things at an architectural level.

VII Acknowledgments

This research is funded by the FWO G.0298.08 and G.0291.09 grants, by the IBBT MoCo project, the FP7 SPITFIRE project and by the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen) through the SymbioNets project.Several concepts from this publication are part of patent with application no. US 12/817,722 (patent pending, filing date June 17 2010).

Competing interests

The authors declare that they have no competing interests.

Received: 9 November 2010 Accepted: 15 August 2011
Published: 15 August 2011

References

1. The internet of things. ITU Internet Reports (2005)
2. P De Mil, T Allemeersch, I Moerman, P Demeester, W De Kimpe, Scalable low-power wsan solution for large-scale building automation, in *A Communications, 2008. ICC '08. IEEE International Conference on*, pp. 3130–3135 (2008)
3. G Virone, A Wood, L Selavo, Q Cao, L Fang, T Doan, Z He, R Stoleru, S Lin, JA Stankovic, An advanced wireless sensor network for health monitoring, in *Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare (D2H2)*, (Arlington, 2006)
4. H Yan, Y Xu, M Gidlund, Experimental e-health applications in wireless sensor networks. **1**, 563–567 (2009)
5. A Dunkels, JP Vasseur, ip for smart objects. White paper (2008)
6. K Mayerm, W Fritsche, Ip-enabled wireless sensor networks and their integration into the internet, in *Proceedings of the First International Conference on Integrated Internet Ad Hoc and Sensor Networks. InterSense '06*, vol. 138 (Nice, 2006)
7. E De Poorter, I Moerman, P Demeester, An information driven sensornet architecture, in *The Third International Conference on Sensor Technologies and Applications (sensorcomm 2009)*, (Athens, 2009)
8. P Stuckmann, R Zimmermann, European research on future internet design. IEEE Wirel Commun Mag (2009)
9. S-S Kim, M-J Choi, H-T Ju, M Ejiri, JW-K Hong, Towards management requirements of future internet, in *Lecture Notes in Computer Science: Challenges for Next Generation Network Operations and Service Management*. **5297**, 156–166 (2008). doi:10.1007/978-3-540-88623-5_16
10. F-U Andersen, H Berndt, H Abramowicz, R Tafazolli, Future internet: From mobile and wireless requirements perspective. EMobil Technol Platf Whitepap (2007)
11. K Cho, J Choi, DD Ko, T Kwon, Y Choi, Content-oriented networking as a future internet infrastructure: Concepts, strengths, and application scenarios, in *Proceedings of International Conference on Future Internet Technologies (CFI) 2008*, (Seoul, 2008)
12. FP7 Self-NET project Self-Management of Cognitive Future InterNET Elements, https://www.ictselfnet.eu/
13. R Boutaba, S Omari, APS Virk, Selfcon: An architecture for self-configuration of networks. J Commun Netw. **3**(4), 317–323 (2001). ISSN 1229-2370
14. A Feldmann, Internet clean-slate design: What and why? SIGCOMM Comput Commun Rev. **37**(3), 59–64 (2007). doi:10.1145/1273445.1273453
15. NMMK Chowdhury, R Boutaba, A survey of network virtualization. Comput Netw. **54**(5), 862–876 (2010). doi:10.1016/j.comnet.2009.10.017
16. S Khanvilkar, A Khokhar, Virtual private networks: An overview with performance evaluation. IEEE Commun Mag. **42**(10), 146–154 (2004). doi:10.1109/MCOM.2004.1341273
17. K Lua, J Crowcroft, M Pias, R Sharma, S Lim, A survey and comparison of peer-to-peer overlay network schemes. IEEE Commun Surv Tutor, 72–93 (2005)
18. The FP7 IST 4WARD project, *Architecture and Design for the Future Internet*, http://www.4ward-project.eu/
19. FP7 IST 4WARD project, D2.1 Technical Requirements.
20. The FP6 MAGNET and MAGNET beyond project, http://magnet.aau.dk/
21. R Prasad, ed, *My personal Adaptive Global NET (MAGNET), ISBN: 978-90-481-3436-6*, 1st edn. (Springer, Signals and Communication Technology). (XXXIV, Hardcover, 2009)
22. The itea2 usenet project, https://usenet.erve.vtt.fi/
23. J Hoebeke, G Holderbeke, I Moerman, B Dhoedt, P Demeester, Virtual private ad hoc networking. Wirel Pers Commun. **38**, 125–141 (2006). doi:10.1007/s11277-006-9021-1
24. NMMK Chowdhury, R Boutaba, Network virtualization: State of the art and research challenges. IEEE Commun Mag IEEE Commun Soc. **47**(7), 20–26 (2009)
25. N Niebert, I El Khayat, S Baucke, R Keller, R Rembarz, J Sachs, Network virtualization: A viable path towards the future internet. Wirel Pers Commun. **45**(4), 511–520 (2008). doi:10.1007/s11277-008-9481-6
26. J Roberts, The clean-slate approach to future internet design: a survey of research initiatives. Ann Telecommun. **64**, 5–6 (2009). doi:10.1007/s12243-008-0068-8
27. FP7 Sensei project, http://www.ict-sensei.org/
28. FP7 SemSorGrid4Env project, http://www.semsorgrid4env.eu/
29. D Chu, JM Hellerstein, T-T Lai, Optimizing declarative sensornets, in *SenSys '08: Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems* (ACM, New York, 2008), pp. 403–404
30. MD Dikaiakos, D Katsaros, G Pallis, A Vakali, P Mehra, Guest editors introduction: Cloud computing. IEEE Internet Comput. **12**(5) (2009)
31. LM Vaquero, *et al.* A break in the clouds: Toward a cloud definition. ACM SIGCOMM Comput Commun Rev. **39**(1), 50–55 (2009). ISSN:0146-4833
32. M Pistore, P Traverso, M Paolucci, M Wagner, From software services to a future internet of services, in *Towards the Future Internet: A European Research Perspective* (IOS Press, Amsterdam, 2009), pp. 183–192
33. A Tavakoli, P Dutta, J Jeong, S Kim, J Ortiz, D Culler, P Levis, S Shenker, A modular sensornet architecture: Past, present, and future directions. SIGBED Rev. **4**(3), 49–54 (2007). doi:10.1145/1317103.1317112
34. D Clark, *et al. NewArch Project: Future-Generation Internet Architecture* http://www.isi.edu/newarch/ (2003)
35. R Braden, T Faber, M Handley, From protocol stack to protocol heap: Role-based architecture. SIGCOMM Comput Commun Rev. **33**(1), 17–22 (2003). doi:10.1145/774763.774765

36. NSF FIND (Future Internet Design) research program, in *FIND Informational Meeting: Lessons from FIND 2006,* http://www.nets-find.net/Meetings/FirstPIMeeting/FirstInfoMeeting.ppt (7 Nov2006)
37. NSF FIND research program, Observer Panel Report 2009. Cerf V, Davie B, Greenberg A, Landau S, Sincoskie D. http://www.nets-find.net/FIND_report_final.pdf (9 April 2009)
38. H Zimmermann, OSI reference model–the ISO model of architecture for open systems interconnection. IEEE Trans Commun. **28**(4), 425–432 (1980). doi:10.1109/TCOM.1980.1094702
39. D Clark, R Braden, A Falk, V Pingali, Fara: Reorganizing the addressing architecture, in *Proceedings ACM SIGCOMM FDNA Workshop*, (Karlruhe, 2003)
40. NSF GENI project–Global Environment for Network Innovations http://www.geni.net/
41. http://www.snm.ethz.ch/Projects/TmoteSky TMoteSky Datasheet
42. An overview of the currently available network services in the IDRA architecture, http://idraproject.net/protocols-and-applications
43. The deus project, Deployment and easy use of wireless services http://www.ibbt.be/project/deus
44. Deus project leaflets, Wireless sensor network https://projects.ibbt.be/deus
45. Ad hoc on-demand distance vector (aodv) routing. networking group request for comments (rfc): 3561 http://tools.ietf.org/html/rfc3561 (July 2003)
46. A Tavakoli, D Culler, HYDRO: A Hybrid Routing Protocol for Lossy and Low Power Networks. IETF Internet Draft, http://tools.ietf.org/html/draft-tavakoli-hydro-01 (10 Sept 2009)
47. P Becue, J Rossey, P De Mil, I Moerman, Generic architectural framework for hybrid positioning (poster teaser), in *2010 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, (Zurich, 2010)
48. J Hill, D Culler, Mica: A wireless platform for deeply embedded networks. IEEE Micro. **22**(6), 12–24 (2002). doi:10.1109/MM.2002.1134340
49. W Ye, J Heidemann, D Estrin, An energy-efficient MAC protocol for wireless sensor networks, in *21st Conference of the IEEE Computer and Communications Societies (INFOCOM)*, **3**, 1567–1576 (2002)

doi:10.1186/1687-1499-2011-61
Cite this article as: De Poorter *et al.*: **Enabling direct connectivity between heterogeneous objects in the internet of things through a network-service-oriented architecture.** *EURASIP Journal on Wireless Communications and Networking* 2011 **2011**:61.