

PUBLISH/SUBSCRIBE-ENABLED SOFTWARE DEFINED NETWORKING FOR EFFICIENT AND SCALABLE IoT COMMUNICATIONS

The authors outline the most important issues related to standardization efforts, mobility of objects, networking and gateway access, and QoS support. They describe a novel IoT network architecture that integrates SDN and DDS middleware. The proposed architecture will improve service delivery of IoT systems and will bring flexibility to the network.

Akram Hakiri, Pascal Berthou, Aniruddha Gokhale, and Slim Abdellatif

ABSTRACT

The Internet of Things (IoT) is the result of many different enabling technologies such as embedded systems, wireless sensor networks, cloud computing, big-data, etc., which are used to gather, process, infer, and transmit data. Combining all these technologies requires a research effort to address all the challenges of these technologies, especially for sensing and delivering information from the physical world to cloud-hosted services. In this article we outline the most important issues related to standardization efforts, mobility of objects, networking and gateway access, and QoS support. In particular, we describe a novel IoT network architecture that integrates software defined networking (SDN) and data distribution service (DDS) middleware. The proposed architecture will improve service delivery of IoT systems and will bring flexibility to the network.

INTRODUCTION

The widespread evolution of the Internet of Things (IoT) concept imposes complex requirements on both the underlying networks and communication mechanisms between heterogeneous smart objects that communicate over the Internet. The current estimate for the number of deployed things are on the order of 50 billion connected devices, and by the year 2020 there will be 1000 times more connected mobile devices, all with different requirements, which will have a big impact on how people will interact with the surrounding things. However, this is a very difficult goal to achieve because today's network is limited in its ability to address the requirements of even current IoT deployments. In the future, leveraging the IoT potential will be the key enabler to the creation of different applications areas, including smart cities, e-health, industrial, transportation, retail, safety, and environmental services.

In order to achieve this goal, it is necessary to

provide an efficient network architecture that aids in developing many scalable, interoperable, and predictable IoT applications. This network architecture must be flexible enough to be reprogrammed in accordance with any change in IoT application needs. IoT usually adopts different communication schemes compared to the traditional Internet, stemming primarily from the variety of resource-constrained hardware platforms. In particular, an IoT network is a highly unstructured cloud of wireless and potentially mobile devices, whose states change dynamically (e.g. sleeping and waking up, connected and/or disconnected), and whose locations and speeds change as well. These smart applications should run in resource-constrained devices with limited computation resources, small data-storage capabilities, and low-power consumption. Since these devices sense the environment and communicate, they should be autonomic to respond to different scenarios without human interaction.

Since this large number of smart objects are being provided as services connected to the Internet, the current IPv4 addressing no longer suffices since it has reached its limit, and hence efforts are underway to deploy IPv6 addresses. Additionally, the heterogeneous and dynamic aspects of IoT systems pose major challenges for the underlying network by requiring support for handling heterogeneity, dynamic changes, device discovery, and context-awareness. Contemporary network protocols are designed in isolation to solve a specific problem and are often retrofitted to

address a new requirement. Unfortunately, this approach has limitations for IoT. Moreover, existing protocols often lack the right abstractions that address the requirements of IoT communication. Thus, if network resource utilization is a concern, the network must be flexible enough to be reprogrammed in accordance with any change in IoT application needs. Current network provisioning approaches neither address the dynamicity of IoT applications nor care about resource utilization. Recently, software defined networking (SDN) [1] was introduced to deliver dramatic improvements in network agility and flexibility. The SDN paradigm is thus a promising solution to solve the resource management needs of the IoT environment, but it cannot address the heterogeneous and dynamic needs of IoT applications.

At the distributed systems-level, the Web of things (WoT) [2] concept was introduced to alleviate the heterogeneity issues by allowing smart devices to speak the same language based on open web technologies such as HTTP and REST principles for information sharing. However, to fully explore the potential of WoT, many challenges remain unresolved, such as security and scalability. Moreover, WoT cannot address the need for distributed, peer-to-peer, publish/subscribe semantics, which are a key requirement for IoT applications.

Many solutions at the middleware level were introduced for IoT applications. LinkSmart [3] supports resource discovery, description, and access based on XML and web protocols. Open-IoT [4] realizes on-demand access to cloud-

COMMUNICATIONS STANDARDS

Akram Hakiri is with
ISSAT Mateur, Tunisia.

Pascal Berthou and Slim
Abdellatif are with
LAAS-CNRS, France.

Aniruddha Gokhale is
with Vanderbilt University,
USA.

based IoT services through Internet-connected smart objects. These contributions focus on the upper layer problems such as enabling IP-based radio communication over middleware, but without paying attention to the underlying network. The Object Management Group's data distribution service (DDS) provides real-time, scalable, data-centric publish/subscribe capabilities. However, although DDS has been used to develop many scalable, efficient, and predictable applications at a local area network-scale, its QoS mechanisms are rarely propagated to the network layer, which impedes its use across a wide area scale. Moreover, DDS is not a network-level solution.

To address both the network and distributed systems-level concerns in IoT, we propose combining ideas from SDN with the message-oriented publish/subscribe DDS middleware to define a powerful and simple abstract layer that is independent of the specific networking protocols and technology.

The remainder of this article is organized as follows. We briefly introduce the concepts of SDN and DDS. We introduce the architecture of IoT and provide a glimpse into the most important open issues related to the deployment of IoT networks. We describe the architecture of our SDN solution for efficient use of IoT and discuss the role of the proposed solution in solving the above mentioned issues. Finally, we provide concluding remarks describing potential future directions and open research problems in this realm.

BACKGROUND

SOFTWARE DEFINED NETWORKING (SDN) AND NETWORK FUNCTION VIRTUALIZATION (NFV)

SDN has emerged as a new intelligent architecture for network programmability. It moves the control plane outside the switches to enable external centralized control of data through a logical software entity called the controller. The controller offers northbound interfaces to network applications that provide higher-level abstractions to program various network-level services and applications. It also uses southbound interfaces to communicate with network devices. OpenFlow is an example of a southbound protocol. OpenFlow's behavior is simple but it can allow complex configurations. The hardware processing pipeline from legacy switches is replaced by a software pipeline based on flow tables. These flow tables are composed of simple rules to process packets, forward them to another table, and finally send them to an output queue or port.

One complementary technology to SDN called network function virtualization (NFV) has the potential to dramatically impact future networking by providing techniques to refactor the architecture of legacy networks by virtualizing as many network functions as possible. NFV advocates the virtualization of network functions as software modules running on standardized IT infrastructure (like commercial off-the-shelf servers), which can be assembled and/or chained to create services.

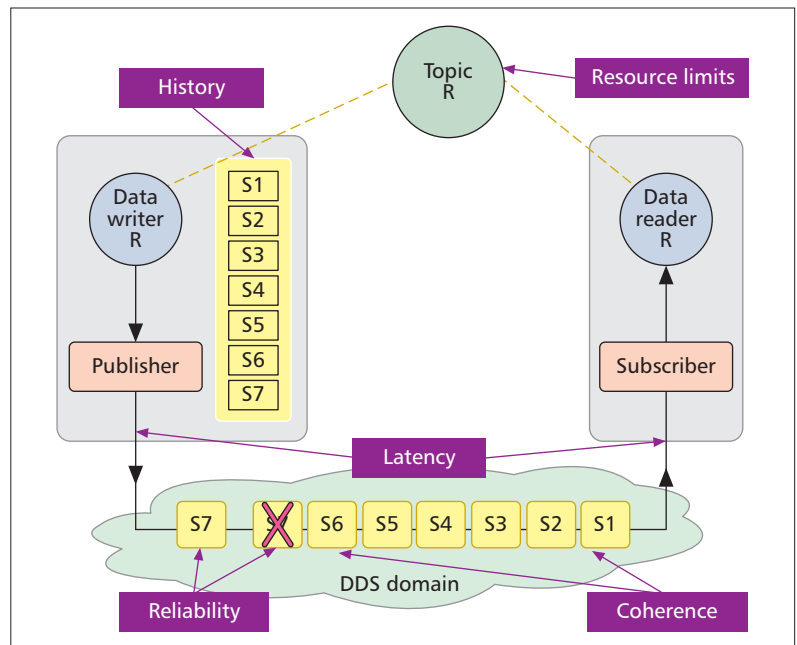


Figure 1. A view of the data distribution service.

OMG DATA DISTRIBUTION SERVICE (DDS)

The Object Management Group has standardized the DDS [5] middleware as a protocol for the IoT to enable network interoperability between connected machines, enterprise systems, and mobile devices. DDS can be deployed in platforms ranging from low-footprint devices to the cloud, and supports efficient bandwidth usage as well as agile orchestration of system components. DDS provides a flexible and modular structure by decoupling:

- Location, via anonymous publish/subscribe.
- Redundancy, by allowing any number of readers and writers.
- Time, by providing asynchronous, time-independent data distribution.
- Message flow, by providing message-based data-centric connection management.
- Platform, by supporting a platform-independent model that can be mapped to different platform-specific models, such as C++ running on VxWorks or Java running on Real-Time Linux.

Figure 1 shows the relation between DDS entities: domains, topics, publishers, data writers, subscribers, and data readers. A DDS domain represents a virtual global data-space; information provided in the domain are accessible by the applications registered to that domain. Publishers manage one or more data writers, and subscribers manage one or more data readers. Also, DDS recognizes the importance of discovery and of meta-data, two areas addressed by IoT systems. The OMG DDS standard has evolved over time. Initially, it provided only the platform-independent and language-independent mechanisms to build distributed publish/subscribe systems with QoS capabilities. Subsequently, to promote interoperability, the standard provided a discovery mechanism via the Real-Time Publish-Subscribe (RTPS) protocol [6].

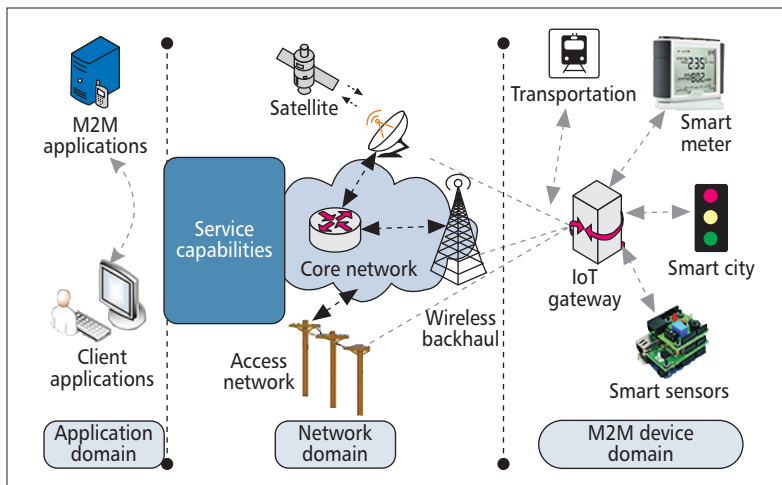


Figure 2. High-level IoT architecture.

During the discovery process each domain participant maintains a local database about all the active data writers and data readers that are in the same DDS domain. Several additional enhancements to the standard are being discussed, including the extended and dynamic types for topics, remote method invocation, integration with Web protocols, security extensions, and integration with SDN at the northbound APIs.

NETWORK COMMUNICATION CHALLENGES IN IOT SYSTEMS

In this section we first describe the high-level architecture of IoT systems. Then we describe their most challenging networking open issues.

IoT ARCHITECTURE

Figure 2 shows a high-level architecture of IoT systems, which is composed of three domains: the device domain, the network domain, and the application domain. In the device domain, the device provides direct connectivity to the network domain via access networks, which may include limited range PAN technologies such as Bluetooth, ZigBee, etc. or via a gateway that acts as a network proxy for the network domain.

Such a gateway must be flexible enough to efficiently manage available resources, QoS, security, and multimedia data exchange. These gateway concepts are prevalent in home ADSL models and WiFi access points found in cyber cafes and wireless hotspots. Because IoT systems integrate heterogeneous smart objects, the design of the gateway is quite different because it must not require each IoT subnetwork to have its own gateway. Thus, a convergent architecture toward a unique solution that integrates traffic incoming from heterogeneous smart devices should be designed. Furthermore, as smart objects are resource-constrained and energy-constrained, the gateway should be aware of the context of each process being managed. It should also employ intelligent routing protocols and caching techniques to route the traffic across the less constrained paths.

The network domain includes different access networks, which provide connectivity through

diverse technologies, such as xSDL, satellite, etc., to devices and/or gateways. It also provides connectivity to the core network, which includes heterogeneous and multi-technology connectivity, such as 3GPP, TISPAN, and LTE-A. Finally, the application domain includes the IoT applications and server/cloud infrastructures. The latter have to share their content, possibly back them up to other devices, analytic programs, and/or people who need to monitor real-time responses to events. They also include service capabilities, which provide functions shared between different applications through open, high-level abstractions and interfaces that hide the specifics of the underlying networks.

NETWORK-LEVEL CHALLENGES FOR IoT

We now describe five key network-level challenges, active research areas, and standardization efforts for IoT.

Current Standardization Efforts: Several initiatives around the world taken by academic organizations, industries, standardization bodies, and governments have emerged during the past few years to enable IoT deployment in everyday life. Among them, the 6LoWPAN protocol was introduced to reside between the IPv6 and MAC protocol layers to make the IPv6 protocol compatible with low-capacity devices. The routing over low-power and lossy networks (ROLL) effort focuses on the routing issues for lower-energy consumption smart object networks. The Constrained Application Protocol (CoAP) is a specialized web transfer protocol for constrained smart objects and networks. Similarly, machine-to-machine (M2M) promotes the development and the maintenance of an end-to-end architecture for M2M, including sensor network integration, naming, addressing, location, QoS, security, etc. These efforts remain isolated, however.

Mobility Management: A large number of IoT devices are not fixed but are found in diverse mobile scenarios. Therefore, supporting and managing efficient mobility for those smart objects is of primary interest to the IoT community. Mobility management in IPv6 networks may be based on either a home agent such as in mobility IP or home location registrar/visitor location registers. The former does not require the use of a central server, which is critical in terms of scalability and flexibility. The latter is widely used in cellular networks, but is not suited for IoT communication since it requires additional and dedicated infrastructure to manage mobility. Both approaches generate a huge amount of signaling traffic to discover devices and maintain their up to date positions/locations, which degrades the network performance. Given the frequent mobility of IoT devices, it is challenging for a SDN controller to have a network view about the mobility of IoT objects to manage their spatial-temporal requests, collaborate with other controllers for adaptive handover, and dynamic flow scheduling in IoT networks

Recurring Distributed Systems Issues: Middleware is required in the IoT environment to provide reusable solutions to frequently encountered problems like heterogeneity, interoperability, security, and

dependability. The Message Queuing Telemetry Transport (MQTT) protocol [7] was introduced as a reliable and lightweight messaging protocol for low-bandwidth, high-latency smart devices. MQTT enables device-to-device communication through a centralized broker. However, the centralized broker presents a single point of failure. Moreover, maintaining a TCP connection between the client and the broker at all times becomes problematic for environments where packet loss is high and computation resources are scarce.

Of particular interest is the pub/sub data-centric middleware because it offers common services to help deliver events from source nodes to interested destinations in an asynchronous way. This kind of middleware integrates diverse types of communication patterns expected for IoT systems. The key challenge is how to cope with congestion control that arises in many M2M scenarios. A novel solution should be investigated to take into account the requirements of IoT devices such as short packet length and short-lived.

Communication Protocols: The major goals of widely used transport protocols such as TCP are to guarantee end-to-end reliability as well as congestion control. However, TCP is unsuitable for IoT scenarios since it incurs substantial overhead during the connection setup and severe performance degradation over wireless channels during congestion control. It also requires data buffering at both sources and sinks for packet retransmission and/or in-order delivery. Conversely, UDP has been used for CoAP due to its low overhead and connectionless properties. With gateways acting as repeaters and/or protocol translators, devices are expected to contribute to QoS management by optimizing the resource utilization of IoT networks. Thus, significant research efforts must be devoted to address the particular issues related to reliable transport, routing, and QoS provisioning that arise in many IoT scenarios.

Security and Privacy: The current landscape of IoT uses diverse wireless and/or cellular communication technologies, which makes eavesdropping and vulnerability of channels extremely simple. Also, IoT devices have limited power and computation resources, making complex security schemes infeasible. In particular, various properties, such as confidentiality, integrity, and privacy, must be ensured. Security policies and mechanisms should be specified to guarantee privacy. They should be configurable to suit individual ways to control which of their personal data is being collected, who is collecting those data, and which operations will be performed on such a data. However, in IoT there is no such infrastructure or servers to manage authentication that achieve the appropriate security policies. Thus, none of the existing solutions that have been proposed for wireless sensor networks can be applied to IoT.

SDN-DDS ARCHITECTURE FOR IOT

This work considers the typical architecture of an IoT system described in Fig. 2, where sensors and actuators are connected to local processing and to the Internet. Internet access is provided through a core router connected to a service

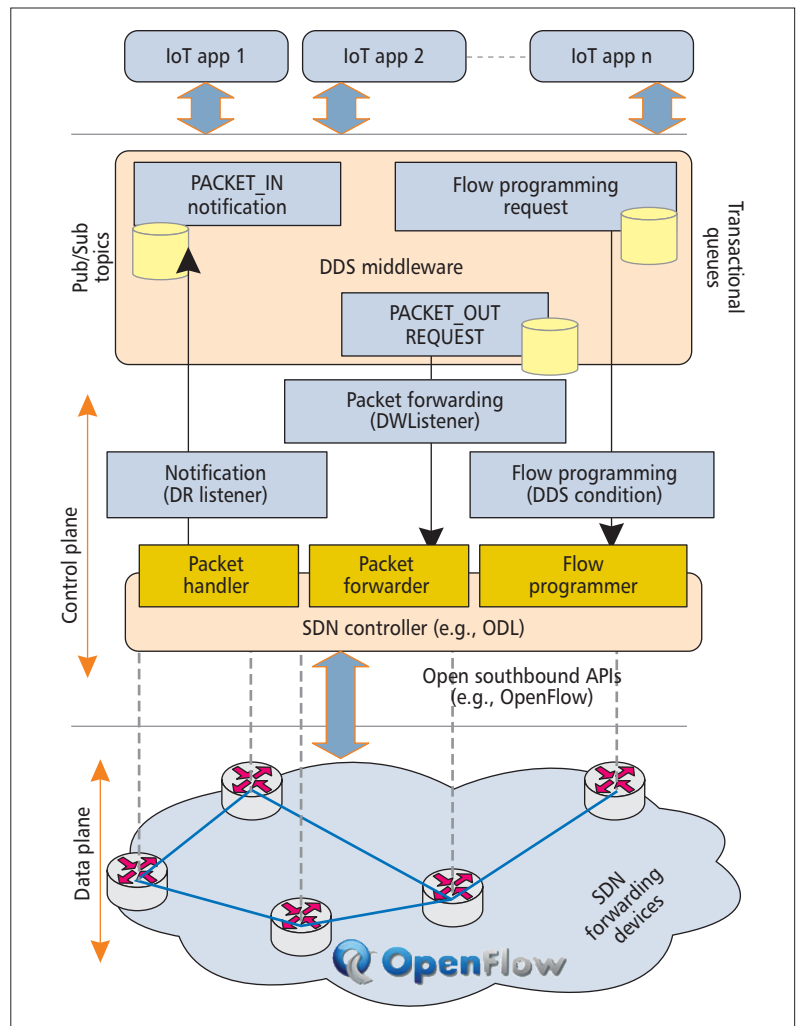


Figure 3. A view of the SDN controller with the DDS layer.

provider through terrestrial or mobile access (i.e. 3G/4G, ADSL, etc.). The router communicates with IoT gateways to enable connectivity over a wide range of communication technologies to support Wi-Fi, Ethernet, VPN, USB, Serial, Zig-Bee, etc. Both smart objects and gateways use DDS middleware to publish/subscribe data. To allow IoT systems to support SDN, we added a DDS northbound interface to the SDN controller that exposes all the necessary functionalities of IoT network applications to the controller to build network-agnostic support for IoT systems.

THE CONTROLLER ARCHITECTURE

The controller integrates a DDS messaging layer that in turn acts as a mediator between the IoT system and the network. It supports both the proactive and reactive flow programming without the need for complex RESTful interfaces nor OSGI services. In contrast to RESTful interfaces that use synchronous client-server point-to-point communication, which may hinder scalability, DDS pub/sub messaging allows anonymous, asynchronous, and many-to-many communication semantics.

As depicted in Fig. 3, three services are implemented by the mediation layer: the packet Handler uses the DDS DataReader (DR) listen-

er to receive the `PACKET_IN` events captured by the controller after being forwarded by the SDN data plane. The packet forwarder service enables forwarding packets received through `PACKET_IN` events or new packets created by IoT applications. The latter are encapsulated in DDS topics, then sent to the packet forwarder service through the publisher and its data writer (DW) listener. Finally, the flow programming service is used to define the flow programming rules on the OpenFlow switches.

The mediation layer transports messages between the network control application and SDN services in the controller. That is, the DDS middleware natively fulfills the requirements of the future proactive IoT in SDN communication, which makes it an attractive technology to satisfy the features they need, such as scalability, reliability, flexibility, security and real-time data delivery.

ADDRESSING OPEN RESEARCH CHALLENGES

The proposed architecture addresses the research questions discussed earlier as follows.

Standardization and Open Innovation: The proposed architecture is based on SDN and DDS, which are open and standardized, which will minimize or

even prevent the lock-in effect. As the middleware choice in our architecture, DDS can interoperate with existing standardized protocols such as CoAP, which can be easily plugged in as an extensions to the middleware without adding any complexity to the underlying network. Moreover, network virtualization enabled SDN as exemplified by OpenFlow will allow experimenting with new ideas. Using a modular design with some inexpensive general hardware platform and customized switch control software, it will be possible to improve the performance of the network.

Addressing and Mobility: Both DDS and SDN help enable a programmable wireless data plane to allow mobility management, dynamic channel configuration, and rapid client association. First, DDS enables many-to-many, broker-less IoT communication so that smart objects can benefit from self-addressable and self-routable data. It also enables the bounded use of resources over potentially intermittent links, and makes it possible to replay messages upon reconnection. Second, SDN techniques are being applied to wireless sensor networks to improve resource and mobility management [8].

We propose to virtualize IoT gateways to maintain active session continuity for IoT devices. Gateway virtualization helps in improving IoT multi-homing by enabling dynamic attachment of sensors to multiple networks. Figure 4 illustrates an example of handover migration of mobile smart devices from an SDN-enabled gateway (GW1) to another gateway (GW2) in a wide area network. If at any time a smart devices moves between different networks, then the centralized SDN controller modifies flow-tables in OpenFlow switches to take into account the mobility of the device.

Efficient Middleware Integration: Using a single middleware that supports multiple communication patterns is a very cost-effective way to develop and maintain large, distributed IoT systems. DDS provides message-oriented publish/subscribe semantics that implement various communication patterns, including transactional queues for request/response interaction (proactive flow programming), guaranteed delivery, and publish/subscribe topics for event-based interaction (reactive flow programming). Thus, we can program flows through a request-response interface implemented by message queues and react to packet-in events by subscribing to events through message topics. Moreover, DDS supports traffic differentiation and prioritization along with selective dawn-sampling.

Service Discovery and Interoperability: DDS discovery helps different smart devices find each other. The IoT gateway maintains a string formatted local objects list to build a complete picture about all its surrounding devices. The list includes the location and the description of each IoT object in the network, e.g. IP addresses, CoAP URI, and multicast address. However, in some cases multicast is not available in IoT networks. So if discovery frames sent with multicast addresses cannot be handled by the gateway, they will automatically be redirected to the controller.

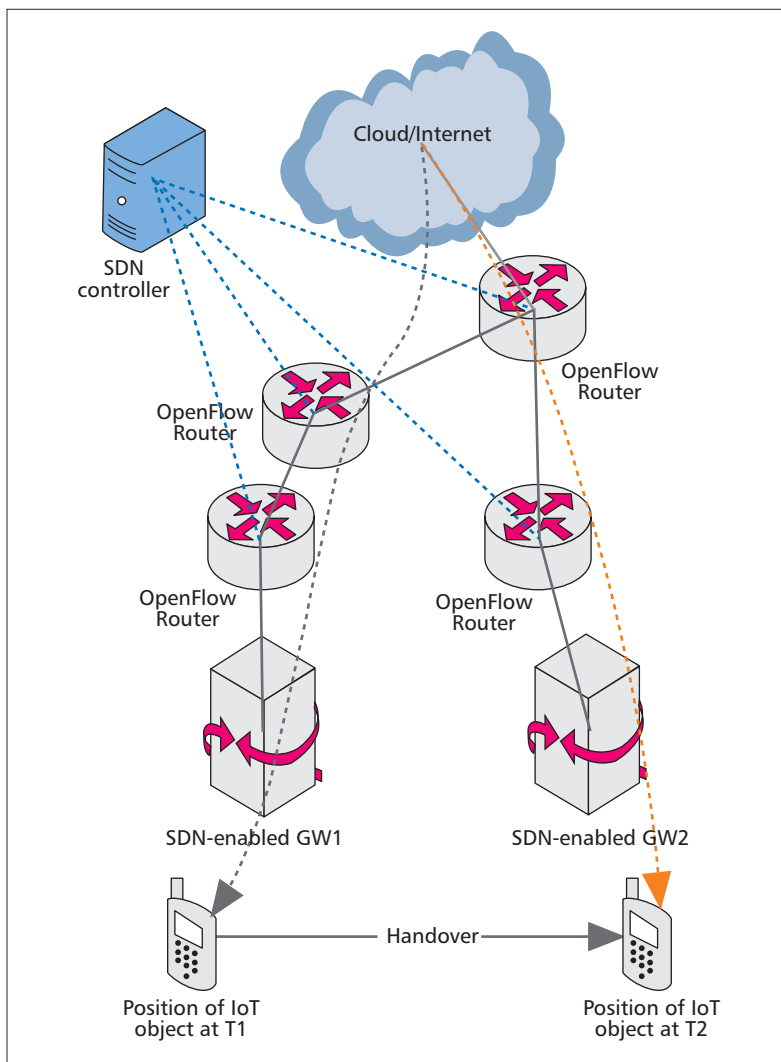


Figure 4. Use of SDN controller to improve mobility management.

The later will install a correct rule in the gateway to simulate a broadcast network so the next frame will be handled without being sent to the controller. This concept is extended to improve the interoperability of incompatible protocols. For instance, when a non DDS compliant IoT gateway is connected to the wireless AP, the gateway frames are redirected to the controller, which also redirects it to an SDN application that encapsulates data into DDS frames according to a predefined scheme.

Scalability Handling: A usual way to connect IoT devices consists of bridging devices with a gateway that collects data. This simple architecture, although frequently used, is not scalable when multiple gateways are used. The main problem stems from the bottleneck when several nodes have to send data at the same time. Usually, data filtering and data fusion are used to cope with this problem. Data filtering aims at dropping unnecessary packets according to a given policy. Data fusion consists of combining data to reduce the number of forwarded packets while keeping almost the same information quality.

DDS offers fusion and filtering mechanisms to reduce useless traffic. The former, called batching mode, enables collecting multiple small data samples and sending them in a single network packet to increase the bandwidth effectiveness. The latter is called content filtered topic, which makes it possible for a smart device to subscribe to a given topic and at the same time specify that it is interested in a subset of this topic data. For example, suppose a topic that contains temperature sensor samples is published with values from 0° to 100°, but the subscribing device needs only values that exceed some threshold. The content filtered topic mode can be used to limit the number of data samples the subscriber has to process and reduce the amount of data sent by the network.

As shown in Fig. 5, OpenFlow completes these mechanisms with an efficient monitoring system. The controller can collect traffic statistics at any granularity and is configured with flow entries downloaded to the switches. Byte or packet counters are associated with every OpenFlow entry. It is then easy to be informed of network flooding and react accordingly.

Context Aware Network Management: IoT, and especially embedded smart sensors, tend to generate huge volumes of data that must be transmitted over the network. It is not always easy to configure the transmission scheme to select exactly the relevant sensor data. DDS offers a useful mechanism called multi-channel.

With multichannel data writers, network traffic can be reduced by subdividing a data flow into a set of streams to be sent over multiple channels. Each channel maintains a set of multicast addresses, defined by an application-specified predicate called the filter expression. It is then possible to associate different multicast addresses with filter expressions, so that only the topics data that match the expression are delivered to the subscriber.

However, multichannel should be configured inside each sending device and cannot be easily

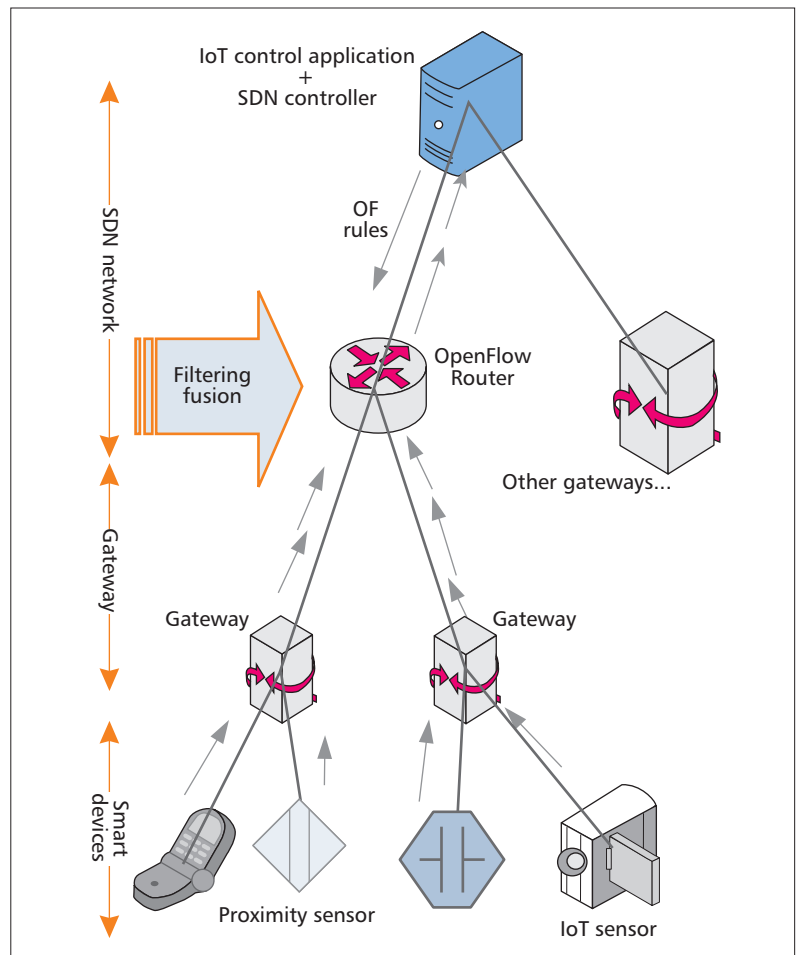


Figure 5. Scalability handling.

modified on-the-fly. OpenFlow could be used to offer a similar service. In Fig. 6, the OpenFlow match-action (Openflow rule) treatment chain could be set to filter data according a keyword contained inside the sent data as close as possible to the source. The decision can be taken at the highest application level, i.e. the application inside the controller. Owing to the OpenFlow monitoring system, the relevant information can be set.

Solving Security Issues: Security issues can be handled by both DDS and SDN. At the middleware level, smart devices can leverage the DDS security model (SM) [9], which offers simple and interoperable security policies without compromising the flexibility, scalability, performance, and QoS-awareness offered by DDS. Thus, we can build a fine-grained secure system that grants permissions to DDS domains, topics, or even data object instances within the topics. Also, DDS “partition” QoS provides another way to create isolated sub-domains to defend IoT networks against attacks. Similarly, with SDN the IoT infrastructure can be sliced into multiple virtual partitions, so that if any attack occurs in a given partition, other IoT partitions remain isolated and secure. Along with network slicing, the controller can program the IoT gateways to conduct fine-grained packet inspection on traffic passing through IoT devices [10]. These statistics collected periodically offer a

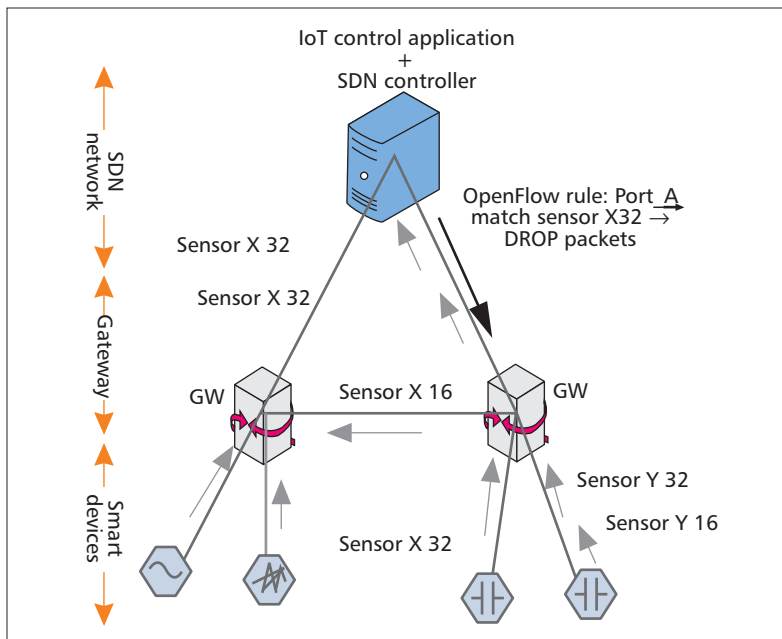


Figure 6. Multi-channel and data filtering.

real-time view of the network state. If any vulnerability occurs in the IoT system, a security SDN application can be deployed on-the-fly at the controller for detecting and driving mitigation of malware and DDoS attacks.

CONCLUSION

The Internet of Things (IoT) promises to have a big impact by adding a new dimension in the way people will interact with surrounding things, and to form a virtual continuum of interconnected smart objects in a worldwide dynamic network. In this article we have surveyed the most important challenges that need to be tackled for efficient support of IoT systems. Then we introduced a data-centric architecture based on a symbiotic relationship between DDS and SDN that enables agile and flexible network orchestration.

As IoT covers a huge range of industries and scales of applications, IoT solutions and architectures are undergoing an evolution with a convergence of technologies, and new innovative solutions such as SDN will be considered, particularly with 5G networks on the horizon (2020 and beyond). As part of our ongoing work on DDS/SDN architectures, we believe that this study can shed light on how we can integrate DDS middleware and SDN to improve IoT communication and promote the adoption of SDN for future IoT networks.

ACKNOWLEDGMENT

This work was partially funded by the Fulbright Visiting Scholars Program and the French National Research Agency (ANR), the French Defense Agency (DGA) under the project ANR DGA ADN (ANR-13-ASTR-0024), and the French Space Agency (CNES). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the

author(s) and do not necessarily reflect the views of the funding parties.

REFERENCES

- [1] A. Hakiri *et al.*, "Software-Defined Networking: Challenges and Research Opportunities for Future Internet," *Computer Networks*, vol. 75, part A, Dec. 2014, pp. 453-71.
- [2] C. Benoit *et al.*, "The Web of Things Vision: Things as a Service and Interaction Patterns," *Bell Labs Technical J.*, vol. 16, no. 1, June 2011, pp. 55-61.
- [3] P. Kostelnik, M. Sarnovsky, and K. Furdik, "The Semantic Middleware for Networked Embedded Systems Applied in the Internet of Things and Services Domain," *Scalable Computing: Practice and Experience*, vol. 12, no. 3, Aug. 31, 2011.
- [4] J. Kim and J. W. Lee, "OpenIoT: An Open Service Framework for the Internet of Things," *2014 IEEE World Forum on Internet of Things (WF-IoT)*, Mar 2014, pp. 89-93.
- [5] Object Management Group, "Data-Distribution Service for Real-Time Systems," OMG, version 1.4. Sept. 2014.
- [6] Object Management Group, "The Real-Time Publish-Subscribe Wire Protocol DDS Interoperability Wire Protocol Specification," OMG, Version 2.2. Sept. 2014.
- [7] A. Stanford-Clark and H. L. Truong, "MQTT for Sensor Networks (MQTT-SN) Protocol Specification," Version 1.2. IBM Zurich Res. Lab., Zurich, Nov. 2013.
- [8] Y. Li *et al.*, "Software Defined Networking for Distributed Mobility Management," *2013 IEEE Globecom Wkshps. (GC Wkshps.)*, 9-13 Dec. 2013, pp. 885-89.
- [9] Object Management Group, "DDS Security Specification," OMG, version 1.0 — Beta 1, June 2014.
- [10] C. Yoon *et al.*, "Enabling Security Functions with SDN: A Feasibility Study," *Computer Networks*, vol. 85, July 2015, pp. 19-35.

BIOGRAPHIES

AKRAM HAKIRI (akram.hakiri@laas.fr) is an associate professor of electrical engineering and computer science at ISSAT Mateur, and a research scientist at SYSCOM Labs' ENIT, both in Tunisia. He is also a senior scientist and researcher at LAAS-CNRS, Toulouse, France, and a visiting research scientist at the Institute for Software Integrated Systems (ISIS) at Vanderbilt University, Nashville, TN, USA. His current research focuses on developing novel solutions to emerging challenges in network virtualization, software defined networking, QoS and resource management in wireless networks, and DRE middleware running over data networks and embedded system interconnects. He obtained his engineer's degree (computer engineering) from the National Institute of Applied Science and Technology (INSAT) in Tunisia in 2007, the M.S. (computer science) from Paul Sabatier University in Toulouse in 2008, and the D.Sc. (computer science) from Paul Sabatier University in Toulouse in 2012.

PASCAL BERTHOU (pascal.berthou@laas.fr) received a Ph.D. in computer science and telecommunication from the National Polytechnic Institute of Toulouse in 2001. He is an associate professor at the University Paul Sabatier. He joined the Laboratory for Analysis and Architecture of Systems of the French National Centre for Scientific Research (LAAS-CNRS) in 1998 as a research staff member, where he works in the area of high-speed networks and protocols and multimedia communications. Since then he has covered two major areas of activity. The first area deals with satellite communication systems. The second research area is sensor networks, particularly communication systems and their application, which in recent years has been directed toward the design of WSN for instrumentation networks. Within this domain he has focused on hardware/software network interface issues and cross layering interactions to reduce the energy consumption in WSN. He recently started research contributions in the area of SDN.

ANIRUDDHA S. GOKHALE (a.gokhale@vanderbilt.edu) is an associate professor in the Department of Electrical Engineering and Computer Science, and a senior research scientist at the Institute for Software Integrated Systems (ISIS), both at Vanderbilt University, Nashville, TN, USA. He has more than 175 technical articles to his credit focusing on topics pertaining to model-driven engineering (MDE), middleware solutions involving design patterns for quality of service (QoS) assurance, and correct-by-construction design and development of distributed real-time and embedded systems. His current research focuses on developing novel solutions to emerging challenges in mobile cloud computing, real-time stream processing, publish/subscribe systems, and cyber physical systems. He is also working on using cloud computing technologies for STEM education. Dr. Gokhale obtained his B.E. (computer engineering) from the University of Pune, India, in 1989, the M.S. (computer science) from Arizona State University in 1992, and the D.Sc (computer science) from Washington University in St. Louis in 1998. Prior to joining Vanderbilt, he was a member of technical staff at Lucent Bell Laboratories in NJ. He is a senior member of both IEEE and ACM, and a member of ASEE. His research has been funded in the past by DARPA, DoD, and NSF, including a NSF CAREER award.

SLIM ABDELLATIF (slim@laas.fr) received the M.S. (1998) and the Ph.D. (2002) in computer science, both from the University of Toulouse, France. He is currently an assistant professor at the National Institute of Applied Science (INSA) of Toulouse, and a research scientist at the French National Centre for Scientific Research (LAAS-CNRS). His current research interests include network virtualization, software defined networking, QoS, and resource management in wireless networks.