Challenges of Satisfying Multiple Stakeholders: Quality of Service in the Internet of Things

Chien-Liang Fok, Christine Julien
The University of Texas at Austin
Austin, Texas, USA
liangfok@mail.utexas.edu
c.julien@mail.utexas.edu

Gruia-Catalin Roman, Chenyang Lu
Washington University in Saint Louis
Saint Louis, MO, USA
roman@wustl.edu
lu@cse.wustl.edu

ABSTRACT

As wireless sensor networks become increasingly integrated with the Internet, the Internet of Things (IOT) is beginning to emerge. The IOT is a massive cyber-physical system that presents many software engineering challenges. One fundamental challenge is the need for multi-dimensional QoS that can satisfy the individual constraints of the many participants in the system. In this paper, we investigate the challenges of providing such a mechanism via a simple abstraction consisting of a general QoS function provided by each application. This function distills the multi-dimensional QoS specifications from each stakeholder into a single value that is used to determine the best configuration of interactions. We prototype our approach in a real wireless sensor network using a pervasive healthcare fall-detection application and highlight the many challenges it unveils.

Categories and Subject Descriptors

D.2 [**Software Engineering**]: General

General Terms

Design

Keywords

Internet of Things, Multi-Dimensional Quality of Service

1. INTRODUCTION

With the creation of wireless sensor networks (WSNs) and their integration with the Internet, the Internet of Things (IOT) is fast-becoming a reality. The IOT is unique in its cyber-physical properties and scope; it forms a giant pervasive machine that can both sense and affect its environment. WSNs extend the Internet's digital nerve-endings into everyday objects. Applications are many; the most often cited include home automation/assisted living and business logistics. The IOT enables greater efficiency through automation and better decisions based on increased context information.

SESENA '11, May 22, 2011, Waikiki, Honolulu, HI, USA

Consider the IOT applied to pervasive healthcare. Cost constraints restrict doctor-patient contact time, and a patient's health may deteriorate between visits [2]. The IOT can prevent this by embedding networked vital sign sensors into the patient's clothing and the environment to continuously monitor the patient [3]. This enables applications to monitor a patient's health and detect emergencies. On a larger scale, epidemiologists can use the IOT to monitor and contain the outbreak of disease [11].

The IOT is relatively new and presents numerous challenges. Of critical importance is how to determine an interaction's endpoints, given that an application must select one or more resources from within the IOT to achieve its desired semantics. For example, consider a patient fall-detection application that alerts caregivers within a certain amount of time and operates at a certain level of reliability. An IOT resource selected to detect a fall must satisfy the functional requirements (i.e., it must be able to detect a fall); this could be in the form of accelerometers on the person, ceiling cameras, floor sensors, etc. The selected resource must also satisfy non-functional requirements such as accuracy and reliability. These requirements place constraints on the selected resource(s), the application requesting the resource, and the network of devices that connects the two.

This working example demonstrates the primary focus of this paper; we investigate the challenges of satisfying the quality-of-service (QoS) demands of the numerous stakeholders in the IOT. The IOT is inherently a complex and shared system consisting of a plethora of applications, networked components, and resources. Many devices are constrained in terms of computation, communication, and energy. These constraints are dynamic and heterogeneous. Coupled with the critical nature of many IOT applications, this motivates the need to provide QoS across multiple dimensions. The first dimension is the nature of the stakeholders. These include the application, resource providers, and the network that connects them. The second is the nature of competing applications in the IOT, since multiple applications must coexist. Finally, the nature of the constraints must be considered (e.g., network characteristics like latency and bandwidth, device properties like battery power and memory, environment attributes like location and temperature, or application requirements like precision and responsiveness).

In the WSN-portion of the IOT, *every device* is an independent stakeholder since every node can serve as a router and endpoint. Thus the demands of each intermediate node involved in an interaction must be considered in addition to the endpoints. Traditionally, endpoint requirements are considered in the application layer, while intermediate node requirements are considered lower in the network stack. These disconnected optimizations may result in conflicts and highlight a key WSN-specific property of our work: the need to allow intermediate nodes to participate in the decision of the final connection between application endpoints.

This paper takes initial steps towards demonstrating the need and ability to satisfy the multi-dimensional QoS constraints of the multiple stakeholders in an interaction within the IOT. As our first contribution, we extend an existing middleware for resource discovery in WSNs to account for multi-hop resource utilization and the expression of multi-dimensional QoS constraints. We review this WSN middleware, Servilla [5], in Section 3 and define our novel approach to QoS specifications in IOT interactions in Section 4. Our second contribution demonstrates feasibility through a concrete implementation that allows provision of resources, definition of application-level constraints, and the resolution of those constraints through a resource discovery process. This feasibility study is described in Section 5. We believe that IOT applications require a middleware and infrastructure that supports the expressive definition of application requirements and constraints; this work offers a first step in providing this for the emerging IOT.

2. BACKGROUND AND MOTIVATION

Supporting the IOT requires flexible and expressive programming abstractions for adaptive applications. We support one aspect of such abstractions, namely the ability to understand and express constraints over QoS requirements.

2.1 Problem Definition

The IOT exhibits properties that render application development difficult, including extreme dynamics, device and network heterogeneity, network size, and resource sharing by many applications. Applications view the IOT as the Internet augmented with an enormous number of embedded devices that can sense and actuate upon the environment; WSN nodes can be accessed in the same manner as traditional Internet devices. Unfortunately, WSN middleware do not always transfer to the IOT since, due to resource constraints, they usually only support a fixed and narrow set of policies for connecting application components. IOT applications demand a more robust and expressive set of abstractions that consider multiple dimensions of QoS. Each application will have its own set of priorities. Mission-critical applications like pervasive healthcare may prioritize energy efficiency and low communication latency, while a less critical application like building automation may prioritize monitoring quality and network utilization. A static set of services and constraints cannot satisfy every application. Furthermore, in a network with interacting applications, achieving the desired multi-dimensional QoS often requires active compliance with both sides of an interaction and the network itself. In this work, we investigate the challenges of balancing the demands of numerous parties (applications, resource providers, and the intermediate network) in a highly diverse, dynamic, and unpredictable network like the IOT.

2.2 Solution Overview

We must instrument resource discovery capabilities to account for the requirements of the users, applications, resources, and the networks that connect them. An application in the IOT requires access to one or more distributed resources. Which resources are best depend not only on resources' capabilities but also on the network that connects the application to the resources. We adopt a model inspired by the application sessions preference specifications [10] combined with resource discovery and use abstractions provided by a WSN middleware, Servilla [5]. In this paper, we extend the ability to discover WSN resources to the multi-hop environment of the IOT. We then investigate how these discovery capabilities can tailor interactions based on diverse multi-dimensional QoS constraints.

2.3 Related Work

Existing WSN programming abstractions connect applications to resources in the environment. These middleware provide either best-effort services or optimization along a single QoS metric like energy [21, 24]. Other metrics explored include quality of monitoring [1, 23], network bandwidth utilization [17, 18], query optimization [12, 20], routing efficiency [18], and network coverage [22]. In the IOT, resource discovery must consider all of these requirements simultaneously. We also aim to develop abstractions that can connect (and reconnect) applications to the best set of resources to support their desired semantics in the IOT.

The provision of QoS-sensitive interactions has also been explored in more traditional mobile ad hoc networks. Many projects mediate QoS requirements through object mobility [7, 8], bringing objects close to clients, which is infeasible in the IOT where applications may be both mobile and location-dependent and resources are shared among applications distributed across the network. Network sensitive service selection [9] observed the differences between performing user-side resource selection and provider-side resource selection; this work motivates our inclusion of QoS constraints within resource requests. Other approaches have differentiated QoS parameters into metrics and policies, considering both constraints on the user and constraints on the provider in determining selected interactions [13, 25]. These approaches do not naturally accommodate dynamic QoS requirements or environments and instead focus on optimal creation and placement of resource instances. We instead wish to discover and make use of available shared resources.

Recently, middleware supporting multi-dimensional QoS has been developed [14]. It focuses on optimizing the overall QoS provided by a composition of service endpoints. This complements our work, which focuses on incorporating the demands of intermediate nodes that connect the endpoints. In addition, a technique for handling conflicts among multiple QoS demands has recently been developed [15]. This technique was applied to optimizing software architecture. It may be used to enhance our approach when the constraints of intermediate nodes conflict with the endpoints.

We explicitly consider the requirements of each endpoint (i.e., the application and the resource) and the network that connects them. Since the network is a stakeholder, we consider the needs of the interconnecting nodes. This differs from existing WSN services like the Collection Tree Protocol [6], which considers the perceived path quality from each node to a designated sink. It does not consider the unique demands of each node in the path, which may be heterogeneous and dynamic in the IOT.

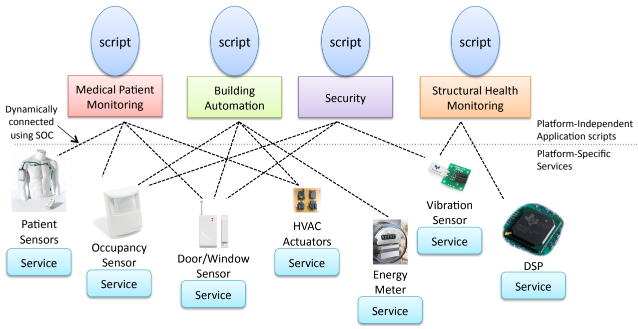

Figure 1: Service-oriented computing (SOC) is typically used in WSNs to enable platform-independent application tasks to be dynamically connected to platform-specific services.

NAME = FallDetection
METHOD = hasFallen
INPUT =
OUTPUT = boolean
ATTRIBUTE version = 5
ATTRIBUTE accuracy = 3
ATTRIBUTE power = 8

Figure 2: The specification of a fall detection service

3. REVIEW OF SERVILLA

Service-oriented computing (SOC) [16] is widely used on the Internet and therefore a logical choice for integrating WSNs with the IOT. SOC structures interacting software components as service providers and consumers, enabling machine-to-machine communication even across different organizations [19]. While this also applies to the IOT, SOC provides another capability that can be of even greater value: decoupling service providers and consumers. This enables dynamic relationships between service consumers and providers. Specifically, service providers can be discovered and connected to service consumers on-demand. The connections between providers and consumers can be created transparently to the consumer. In addition, connections can be modified for other purposes. In the context of WSNs, Servilla tailors connections to enable adaptation to device heterogeneity [5] and to increase energy efficiency [4]. We build upon Servilla to support multi-dimensional and multi-stakeholder QoS in the context of the IOT.

Servilla provides the SOC programming model in WSNs, i.e., it facilitates the interconnections of service consumers and providers in a WSN. Figure 1 shows an overview of Servilla's programming model. It is divided into two levels. The top contains applications that function as service consumers, while the bottom depicts service providers. Applications are platform-independent, simplifying their implementation, though at the expense of some efficiency due to the need for code interpretation. Services are platform-dependent and perform platform-specific operations like sensing. These natively-implemented processes are optimized for efficiency. Servilla dynamically connects applications to services and adjusts these connections in response to network dynamics and to conserve energy.

Servilla enables service specification through a lightweight language whose syntax is shown in Figure 2. This simple syntax is used to conserve memory since existing standards are relatively verbose. Servilla's service specification consists of an interface that captures the functional properties and attributes that describe the non-functional properties. When determining if a service provider can fulfill a consumer's needs, the consumer's request must exactly match the interface of the service specification. To allow flexibility in matching, the attributes support logical operators. All service providers whose specifications match are assumed to be interchangeable from the consumer's perspective.

Services must be discovered before they can be used. In Servilla, service discovery consists of a consumer broadcasting the desired specification and waiting for providers to respond. Each provider compares any received request against its specification. If they match, the provider responds with a set of values that characterize its energy efficiency (since Servilla focuses exclusively on minimizing energy consumption). The service consumer collects responses, ranks potential providers by energy efficiency, and connects to the most efficient provider. While focusing on energy efficiency makes sense in WSNs where energy is among the scarcest of resources, Servilla's approach must be enhanced to support multi-dimensional, multi-application QoS requirements within the more diverse IOT.

Although designed for WSNs, Servilla serves as an excellent platform for evaluating mechanisms by which multi-dimensional and multi-stakeholder QoS can be provided on the IOT. To do this, multi-hop service discovery must be supported since IOT applications will involve devices that span a wider physical area than in WSNs, and additional metrics must be supported to account for a broad range of QoS demands. With the introduction of multi-hop connections, the effects of the intermediate routing nodes on an interaction's overall QoS must be considered. That is, intermediate nodes may limit their support of multi-hop connections that do not directly benefit them. All of these challenges differ from Servilla's current model, which only differentiates providers based on energy efficiency, only considers single hop connections among consumers and providers, and only considers the energy-constraints of the end points. The IOT will involve more applications of greater complexity spanning a larger number of nodes, resulting in a greater diversity of application QoS demands and device constraints.

4. DEFINING QUALITY OF SERVICE

QoS is inherently application specific. Thus, a mechanism is needed that enables specifying application-specific QoS requirements while keeping service-provisioning general. In this paper, we use a *QoS function* that converts multiple input values representing relevant quantifiable constraints of each stakeholder into a single *QoS value* whose magnitude represents the inverse of the total QoS provided. Only quantifiable constraints are considered to simplify our exploration of the problem domain. In the future, we must account for QoS attributes like security that are not easily quantifiable and generalize the single QoS value into a vector [14]. The QoS value incorporates requirements of the consumer, provider, and network participants needed to support the interaction. Example input values include devices' energy efficiencies, energy availabilities, sensor accuracies, network bandwidth, and processor speeds. We assume that the set of potential input values is known, predefined, and satisfies the needs of the QoS functions. Currently, input values are delivered to the initiator node, which computes the QoS value locally; in the future, the QoS function should be distributed to achieve greater scalability.

Our QoS function facilitates multi-dimensional, multi-stakeholder QoS. It is multi-dimensional since each input value represents a different dimension along which QoS can be quantified. Our fall-detection application may place more weight on sensor accuracy and less weight on energy efficiency. If a different application most values system lifetime, it can place more weight on energy efficiency and deemphasize sensing accuracy. Each application submits its own QoS function. For each application, the middleware evaluates its QoS function on demand and configures the most ideal system configuration in terms of maximizing the QoS values of the applications' interactions with service providers.

In the fall-detection application, the stakeholders include the application, the sensing services that detect patient falls, and the intermediate routing nodes that connect the two. While each of these stakeholders have their own constraints, only the application provides a QoS function. This is because QoS must be defined in the context of an application. The constraints of the other stakeholders are accounted for by the input values that they supply to the QoS function. For example, many different services may exist that provide the sensing necessary to detect patient falls. One service may use an accelerometer on the patient, another may rely on pressure sensors in the floor, while another may rely on a network of video cameras in the ceiling. These services differ in energy efficiency and sensing accuracy, which is reflected by different input values for the QoS functions. Their constraints are considered during the service selection process since their semantics are integrated into the resulting QoS value.

Using the same mechanism described above, the network constraints are also considered. They are those of the intermediate nodes that route data between the consumers and providers. Their constraints deal with network-related issues like bandwidth availability, network capacity, and latency. If the constraints of an intermediate node result in suboptimal QoS, the middleware will detect this through a lower QoS value and attempt to select a different route to the service provider or switch to an entirely different provider. This exemplifies one approach to multi-dimensional constraints of each stakeholder in a multi-application system.

5. IMPLEMENTATION AND EVALUATION

In this section, we demonstrate the feasibility of the proposed multi-dimensional multi-application QoS mechanism and how it impacts the functional behavior of the network. Our implementation builds upon Servilla. Its architecture is shown in Figure 3 with modifications highlighted in red.

5.1 Implementation Details

Our extensions to Servilla provide an implementation of the QoS function. We do not perform distributed evaluation of this function which is likely to be essential in a full-scale implementation. In this section, we describe our first effort and its use to demonstrate the impact and import of muti-dimensional, multi-stakeholder QoS in the IOT; in Section 6 we elaborate on ways in which this initial framework enables future research.

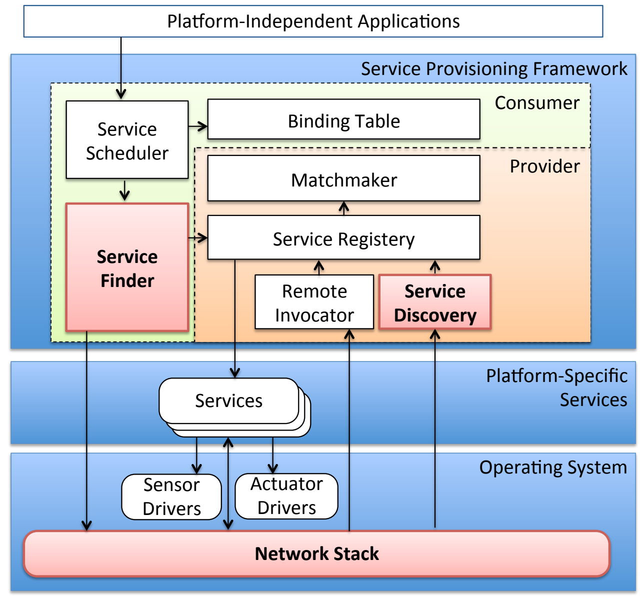

Figure 3: Multi-dimensional, multi-application QoS in Servilla.

The service finder searches for matching providers and ultimately selects which provider to use. We extended it to take as input a QoS function. This function consists of an array of n integers that characterize the cost of n QoS dimensions. That is, the number of integers in the QoS function is equal to the number of dimensions of QoS supported. The service finder also records the n corresponding QoS properties for each potential provider and intermediate node in a *constraint table*. Finally, the service finder includes a *network topology table* that records the network topology. Using these tables the service finder can calculate the QoS value of each potential provider while considering the QoS demands of the intermediate network by calculating the sum of the products between the values in the QoS function and the constraints in the constraint table; lower QoS values are better.

The service discovery component responds to messages sent by the service finder. We modified it to report the local node's QoS constraints and list of immediate neighbors in addition to whether it provides the service. This allows the service finder to construct its constraint and network topology tables. The service discovery component responds even when it does not provide the required service since it may still serve as a router.

Finally, we modified the network stack to support multi-hop routing. This is necessary since IOT applications are expected to span greater distances. We used source routing for multi-hop unicast operations and hop-limited flooding for multicast operations. These are not sophisticated approaches, but their simplicity makes them well suited for demonstrating the feasibility and utility of supporting multi-dimensional QoS within the IOT.

5.2 Feasibility Demonstration

We demonstrate our implementation using a fall-detection application and a testbed of five TelosB nodes. This demonstration is not a performance evaluation but instead demon-

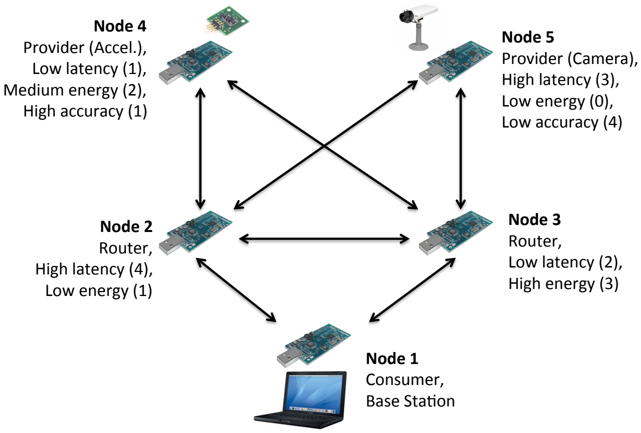

Figure 4: Network topology for demonstrating multi-dimensional QoS for fall detection.

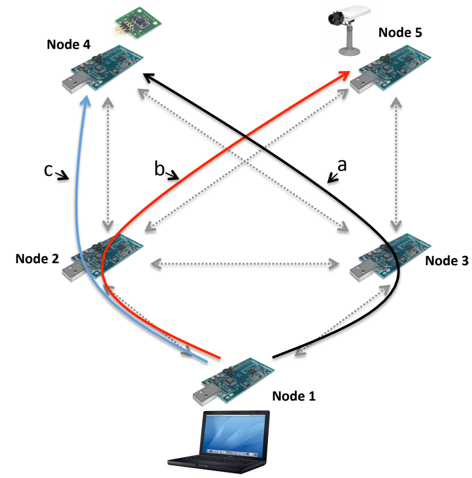

Figure 5: Three network configurations for different QoS functions: (a) low latency and medium accuracy, (b) energy efficiency, and (c) energy efficiency and high accuracy.

strates the potential impact of a framework for supporting multi-dimensional QoS constraints submitted by multiple stakeholders in the IOT. We aim to illustrate how considering the constraints of all stakeholders results in functional differences in network behavior with tangible benefits.

We use three QoS dimensions: energy, latency, and accuracy. This is simply for demonstration purposes; more dimensions will be used in a real-world deployment. The semantic of each dimension is lower values denote higher QoS. Energy characterizes how much energy a node consumes for the interaction relative to the amount of energy available. Latency is the amount of delay introduced by the node when serving as a router or provider. Finally, accuracy quantifies the provider's ability to provide the service. In this case, this is how reliably the service can detect a fall. Accuracy is only defined by nodes that provide the service; other nodes have an accuracy of zero, which negates its impact on the resulting QoS value.

Consider the wireless sensor network shown in Figure 4 that consists of five nodes deployed in a house. Node 1 is the base station and has a QoS specification of [latency=0, energy=0, accuracy=0]. The application requires a fall detection service whose specification is shown in Figure 2. Nodes 4 and 5 provide the service but differ in QoS. Node 4 uses an accelerometer physically mounted on the patient, while node 5 uses cameras mounted throughout the home. Node 4 is more accurate and faster than node 5. However, node 5 is powered by the grid and is not energy-constrained. Thus, the QoS specifications of nodes 4 and 5 are [latency=1, energy=2, accuracy=1], and [latency=3, energy=0, accuracy=4], respectively. The two remaining nodes serve as potential routers. They also differ in QoS. Specifically, the QoS specifications of nodes 2 and 3 are [latency=4, energy=1, accuracy=0] and [latency=2, energy=3, accuracy=0], respectively.

In the current implementation, all QoS specifications and network topology information are delivered to the consumer during service discovery. This node uses this information to analyze every possible path between it and available providers. For each potential path, the cost is calculated using the QoS function to generate a QoS value for each node along a path and summing these values to create a single QoS value for the path. Suppose the application values low latency and medium accuracy by using the following QoS function: [latency sig.=2, energy sig.=0, accuracy sig.=1].[1] The cost of the connection $1 \rightarrow 3 \rightarrow 4$ is $(2 \cdot 0 + 0 \cdot 0 + 1 \cdot 0) + (2 \cdot 2 + 0 \cdot 3 + 1 \cdot 0) + (2 \cdot 1 + 0 \cdot 2 + 1 \cdot 1) = 7$, which is the lowest QoS value (i.e., best QoS) among all the possible configurations, given this particular QoS function. This is shown in Figure 5, configuration (a).

The ideal configuration depends on the QoS function. Suppose the application was only interested in conserving energy and provides a QoS function of [latency sig.=0, energy sig.=3, accuracy sig.=0]. In this case, the most ideal configuration is $1 \rightarrow 2 \rightarrow 5$, which has a QoS value of 3 (Figure 5, configuration (b)). As another example, suppose the application wanted energy efficiency *and* high accuracy. It may use a QoS function of [latency sig.=0, energy sig.=3, accuracy sig.=3]. This results in Figure 5, configuration (c).

We deployed this scenario, and, using the QoS functions given above, successfully discovered the "best" services. The diversity of results given different QoS functions demonstrates the importance of considering multiple QoS dimensions from every stakeholder. From the consumer, the QoS function specifies application demands. From the router and provider nodes, the QoS specifications quantify properties of these nodes, allowing these nodes' desires to be considered despite the fact that the consumer node makes the ultimate decision on which provider to use and how the connection to the provider is formed. For example, if a routing node determines that it would not like to provide a routing service, it can specify high values for its latency and energy specifications and thus decrease the likelihood that the consumer selects a path containing it.

6. FUTURE DIRECTIONS

We identified the need to consider the multi-dimensional QoS constraints of each stakeholder within the IOT and presented first steps towards realizing this goal. This ini-

[1]The "sig." notation stands for "significance," which quantifies the metric's importance to the application.

tial investigation opens numerous future research directions. Explicit support of constraints set forth by *all* stakeholders must be investigated as opposed to the implicit support provided by our prototype. This is necessary to allow intermediate nodes to refuse service or initiate changes in the configuration of interactions. One way to achieve this may be to generalize the single QoS value into a multi-value tuple, and to consider different configurations that are, for example, Pareto optimal. Another challenge is to enable explicit support for qualitative QoS attributes. Generalized semantics in consumer-provider relationships must also be investigated. There is no reason to restrict interactions to be between a single consumer and provider. Instead, a consumer may be connected to multiple providers but only use one based on the current system state. This is similar to the clustering approach used to handle network dynamics [14]. Continuous QoS must be provided by dynamically adjusting interactions in the IOT in response to network dynamics. It is necessary to consider not only the application's requirements of an interaction but also the requirements of resource providers and the nodes that support communication necessary for the interaction. Support for this multi-dimensional QoS will enable a fluid, expressive, and adaptive framework for supporting multi-hop, opportunistic interactions in the emerging Internet of Things.

7. REFERENCES

[1] S. Bhattacharya, A. Saifullah, C. Lu, and G.-C. Roman. Multi-application deployment in shared sensor networks based on quality of monitoring. In *Proc. of RTAS*, pages 259–268, 2010.

[2] T. A. Brennan, L. L. Leape, N. M. Laird, L. Hebert, A. R. Localio, A. G. Lawthers, J. P. Newhouse, P. C. Weiler, and H. H. Hiatt. Incidence of adverse events and negligence in hospitalized patients. *New England Journal of Medicine*, 324(6):370–376, January 2010.

[3] O. Chipara, C. Lu, T. C. Bailey, and G.-C. Roman. Reliable clinical monitoring using wireless sensor networks. In *Proc. of SenSys*, New York, NY, USA, 2010. ACM.

[4] C.-L. Fok, G.-C. Roman, and C. Lu. Coordinating resource usage through adaptive service provisioning in wireless sensor networks. In *Proc. of Coordination*, pages 107–121, 2010.

[5] C.-L. Fok, G.-C. Roman, and C. Lu. Servilla: A flexible service provisioning middleware for heterogeneous sensor networks. *Science of Computer Programming*, 2010.

[6] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis. Collection tree protocol. In *Proc. of SenSys*, 2009.

[7] R. Grimm, J. Davis, E. Lemar, A. MacBeth, S. Swanson, T. Anderson, B. Bershad, G. Borriello, S. Gribble, and D. Wetherall. System support for pervasive applications. *ACM Trans. on Computer Systems*, 22(4):421–486, 2004.

[8] O. Holder, I. Ben-Shaul, and H. Gazit. Dynamic layout of distributed applications in FarGo. In *Proc. of ICSE*, pages 163–173, 1999.

[9] A.-C. Huang and P. Steenkiste. Network-sensitive service discovery. *Journal of Grid Comput.*, 1(3):309–326, 2003.

[10] C. Julien. Adaptive preference specifications for application sessions. In *Proc. of ICSOC*, volume 4294 of *Lecture Notes in Computer Science*, pages 78–89. Springer Berlin / Heidelberg, 2006.

[11] M. Kazandjieva, J. W. Lee, M. Salathe, M. W. Feldman, J. H. Jones, and P. Levis. Experiences in measuring a human contact network for epidemiology research. In *Proc. of HotEmNets*, 2010.

[12] Y. W. Lee, K. Y. Lee, and M. H. Kim. Energy-efficient multiple query optimization for wireless sensor networks. In *Proc. of SENSORCOMM*, pages 531–538, 2009.

[13] J. Liu and V. Issarny. QoS-aware service location in mobile ad hoc networks. In *Proc. of MDM*, pages 224–235, 2004.

[14] N. B. Mabrouk, S. Beauche, E. Kuznetsova, N. Georgantas, and V. Issarny. Qos-aware service composition in dynamic service oriented environments. In *Proceedings of Middleware*, Middleware '09, pages 7:1–7:20, 2009.

[15] A. Martens, D. Ardagna, H. Koziolek, R. Mirandola, and R. Reussner. A hybrid approach for multi-attribute qos optimisation in component based software systems. In *LNCS*, volume 6093, pages 84–101. Springer Berlin / Heidelberg, 2010.

[16] M. P. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann. Service-oriented computing: State of the art and research challenges. *Computer*, 40(11):38–45, 2007.

[17] M. Rabbat and R. Nowak. Distributed optimization in sensor networks. In *Proc. of IPSN*, pages 20–27, 2004.

[18] P. Skraba, H. Aghajan, and A. Bahai. Cross-layer optimization for high density sensor networks: Distributed passive routing decisions. In *Ad-Hoc, Mobile, and Wireless Networks*, volume 3158, pages 630–630. Lecture Notes in Computer Science, 2004.

[19] J.-P. Vasseur and A. Dunkels. *Interconnecting Smart Objects with IP: The Next Internet*. Elsevier, 2010.

[20] S. Xiang, H. B. Lim, K.-L. Tan, and Y. Zhou. Two-tier multiple query optimization for sensor networks. In *Proceedings of the 27th International Conference on Distributed Computing Systems*, page 39, 2007.

[21] G. Xing, C. Lu, Y. Zhang, Q. Huang, and R. Pless. Minimum power configuration for wireless communication in sensor networks. *ACM Trans. Sen. Netw.*, 3(2), 2007.

[22] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated coverage and connectivity configuration for energy conservation in sensor networks. *ACM Trans. Sen. Netw.*, 1(1):36–72, 2005.

[23] Y. Xu, A. Saifullah, Y. Chen, C. Lu, and S. Bhattacharya. Near optimal multi-application allocation in shared sensor networks. In *Proc. of MobiCom*, 2010.

[24] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient mac protocol for wireless sensor networks. In *Proc. of Infocom*, pages 1567–1576, June 2002.

[25] T. Yu and K.-J. Lin. Service selection algorithms for composing complex services with multiple QoS constraints. In *Proc. of ICSOC*, pages 130–143, 2005.