A Resource Mobility Scheme for Service-Continuity in the Internet of Things

Frieder Ganz*, Ruidong Li†, Payam Barnaghi*, Hiroaki Harai†
*Centre for Communication Systems Research, University of Surrey
GU2 7XH Guildford, United Kingdom
{F.Ganz, P.Barnaghi}@surrey.ac.uk
†National Institute of Information and Communications Technology
Koganei-shi, 184-8795 Japan
{lrd, harai}@nict.go.jp

***Abstract*—In the Internet of Things (IoT) a large number of devices enable data communication and interaction between physical objects and the cyber world. An important feature of IoT is the possibility of having mobile objects equipped with sensing devices. In service-enabled IoT platforms, where data and interacting are provisioned as services, access and utilisation of these services are affected by the mobility of the resources that provide the data and services. In a reliable and dependable environment, service continuity is supported in scenarios where the IoT resources are mobile or can become unavailable due to handover delays, network disconnection or power outage. In this paper, we propose a resource mobility scheme with two operating modes - caching and tunnelling. We use these methods to enable applications to access the sensory data when the resources become temporarily unavailable. We have implemented a prototype for the proposed scheme in a mobile scenario. The evaluation results show a reduction of service loss in mobility scenarios by 30%.**

I. Introduction

The Internet of Things (IoT) enables the integration of real world objects into the virtual world, where sensors, actuators and other devices interact and communicate data with each other and also with human users and software agents on the Internet. One approach to make the IoT data available to the users is to use service-oriented technologies. In particular using web service technologies can provide direct integration of the IoT data and functionalities into the Web. There are existing solutions that support this type of scenarios e.g the IETF CoAp protocol [4] that supports communication with resource constrained devices via RESTful services.

In our previous work we developed a gateway component for resource constrained IoT devices and proposed a mechanism for node-to-gateway associations [5]. A limitation of our previous work was lack of support for mobility scenarios when nodes move from one gateway to another. In this paper, we address this issue and discuss challenges in supporting continuous data and service access while the resources are ubiquitous.

Commonly used approaches such as WS-* [1] and RESTful [2] services are used on the web to offer data and services via common interfaces. These specifications, however, have never been designed to meet the requirements of mobile and low-processing IoT environments. The emerging solutions, such as 6LOWPan [3] and CoAP, enable IP and HTTP-based communications and interactions in the constrained environments; however they do not directly address the mobility requirements for ubiquitous resources in those environments.

II. Related Work

The service-oriented solutions are commonly used in business environments where typically powerful application servers run the services and enable interacting with the end-user or other applications. In IoT, due to constraints and mobility of resources, new lightweight solutions are required. Currently there are two different ways how to handle the mobility issue for service-oriented solutions in IoT. Either the mobility is addressed at the network level where transport and lower layer protocols are used to handle the mobility tasks or it is supported at the service level where mechanisms are introduced to support mobile devices.

At the network level, several existing work have investigated mobile scenarios (a detailed survey is provided in [6]), however they do not directly address the issue of mobility and service provisioning.

In that case that an IoT device has enough processing and connection capabilities, Mobile IPv6 [9] can be used to access the device and its data. Mobile IPv6 introduces mechanisms for mobility; however the implementation requires high processing capabilities. The service rendering for nodes with high processing power is shown in Fig. 1 (left side), where the integration of the sensor node into a service environment is handled directly.

6LoWPAN [3] is an implementation of the IP stack for constrained devices based on IPv6 6LoWPAN includes compressed headers and other adaptations that make it lightweight. However, 6LoWPAN has not been implemented with any mechanisms to support mobility and another drawback is that it needs an intermediate ”translator” node to transform the 6LoWPAN protocol into IPv6. In the latter case, when the nodes are not enabled to connect directly to a service provider, an intermediate node is needed as shown in Fig. 1 (right side).

On the service level, new protocols are investigated to integrate constrained devices into existing service architectures. The Restful architecture is designed to use existing web

technologies (i.e. HTTP); however REST is still heavyweight for implementation on constrained mobile devices due its heavy interaction requirements.

The Constrained Application Protocol (CoAp) is a standardisation effort to introduce HTTP service based solutions for the constrained environments, such as IoT [4]. CoAp uses 6LoWPAN to enable HTTP services similar to Rest on constrained devices. CoAp allows integration of devices into service environments, but it is limited in mobility support.

Elsaleh *et al.* [11] introduce mobility for sensor devices in the IoT domain at the gateway level. Services provided by sensors can migrate from one gateway to another gateway. However it does not support service availability during the handover and in the cases where a node is not connected to any gateway due to the lack of coverage. The service provisioning is stopped during the movement and resumed after migration and if the migration takes too long or never finalises, the service remains unavailable.

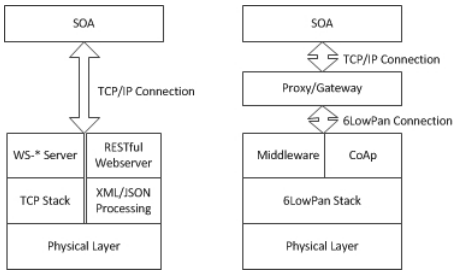

Fig. 1: Service rendering on node vs intermediate rendering

III. Problem Identification and Requirements

In this section we describe the mobility issues during mobility scenarios when a number of sensor nodes are connected to interact via several gateways. The mobility is considered as one sensor node moving from one gateway to another. Several challenges arise during the movement of the sensor nodes in this scenario. Fig. 2 shows a mobility scenario where a user queries a Service Provider (SP) for web services. The SP finds the nodes that can provide data related to the query. SP obtains the information of the available devices via the sensor gateways (SGW) and uses this information to process the sensor description, find, select and forward the query to relevant resources to respond to the user queries.

On the service level, services can become unavailable because of the loss of access to the sensor for several reasons. One reason is that the handover has not been communicated between the new and old SGW and therefore SP is not informed about the new access path to retrieve the sensor data via the new gateway. This is depicted in Fig. 2 (handover delay) between the movement from SGW_0 to SGW_1. Another reason for service unavailability is the loss of signal coverage during the movement process. This is depicted in the Fig. 2 (coverage loss) during the handover between SGW_1 and SGW_2. Another challenge on the service level is the inflexibility of common service providers. In our sample scenario, the service provider

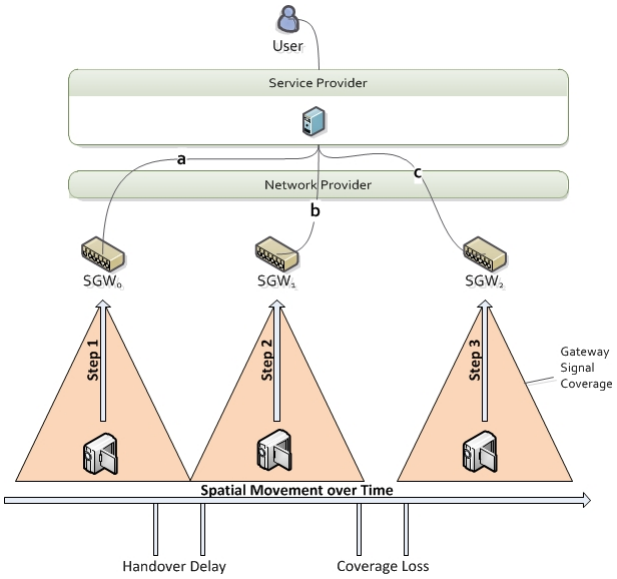

Fig. 2: IoT Scenario

uses SOA that is not designed for mobile scenarios and can not automatically update the access paths regarding to topology changes as they occur in mobile scenarios.

Especially in scenarios with different stakeholder, security and trust is a challenge contemplating the seamless service continuity. In our work, the security issues are addressed in a secure sensor sharing framework [12] and a trust evaluation mechanism [13], which can be extended to construct a security/trust platform in the generic IoT scenario. We do not discuss the details of the security related issues and will focus on seamless service continuity for the mobility scenarios.

IV. A Resource Mobility Scheme for the Service-Enabled Internet of Things

To address the disruption issue, we propose a resource mobility scheme for the service-enabled IoT scenarios. The scheme introduces two modes: a caching and a tunnelling mode. The caching method is used to address the delay issues: *Handover Delay* and *Coverage Loss*. In the caching mode, the gateway caches the last reading from the sensor every time it is queried and also during the initial sensor association. The tunnelling mode addresses the *Inflexibility* of SP to handle changes in the underlying topology or the case that SP has not been updated about the new location of a particular sensor node. In Fig. 3 and 4, the scheme for the caching and tunnelling modes is depicted. The association phase is the same for both, tunnelling mode and caching mode and described as follows.

In the association phase between sensor and SGW the sensor receives a beacon signal from SGW_1 sent as $Send\_Beacon()$ and thereupon requests authentication with $Start\_Authentication(nodeId)$. If the sensor is allowed to associate to the node, permission is granted $Grant\_Permission()$. The node requests a session from SGW_1 $Request\_Session\_Id()$ and a session ID is retrieved

via $Retreive\_Session\_Id()$. The session ID contains the name or address of the node's current associated gateway.

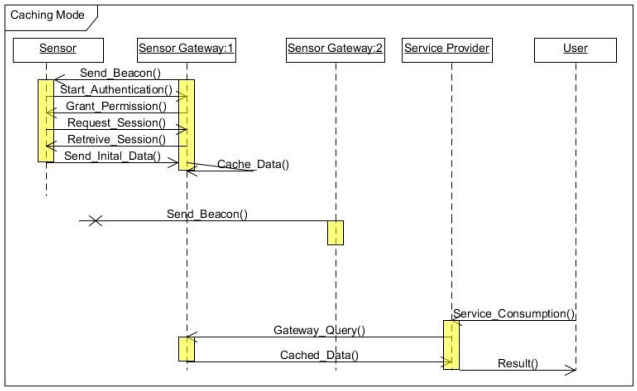

Fig. 3: Caching Mode

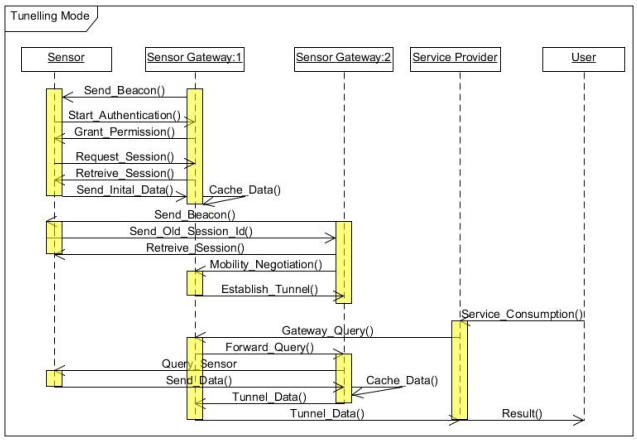

Fig. 4: Tunnelling Mode

1) Caching:

The proposed scheme introduces a cache, for each SGW to store the latest data of each connected node cached on a gateway. This data is used during the handover delay to respond to the queries. The service remains available even during long sensor disconnection (i.e moving in an area with no coverage to another SGW). If a user queries the SP and SP tries to access the sensor data via a SGW which cannot access the requested sensor, cached data is used to serve the query.

After the association phase in Fig. 3, the node sends its current data to SGW by $Send\_Initial\_Data()$ and SGW caches this data. The cache is also updated during every successful sensor query with the latest sensor data. In the case that the sensor node starts to move and can not receive a new beacon signal due to coverage issues, the caching mode will be applied. During the movement of the node, a user uses SP services by sending $Service\_Consumption()$ request. SP forwards the queries to the responsible SGW_1, which is not able to contact the particular node. SGW_1 will use its latest cached reading to respond to the SP $Cached\_Data()$ which will be also used to serve the request of the user via SP.

The trade-off for this approach is the data freshness. In some critical scenarios where the latest data needs to be accessed, such as medical or surveillance scenarios this approach is not applicable.

2) Tunnelling:

The tunnelling approach can be used during the handover while the sensor node is already connected to the new SGW but the SP has not been updated with this movement information. This could be the case for static SPs which do not support the changes in the underlying topology. While SP is not updated with the new SGW information, the SGW can tunnel the request to the new SGW in the tunnelling mode of the proposed resource mobility scheme. This is possible because the sensor node can submit ID of the new SGW to the old one during the re-association phase. After the first association to SGW_1 , the sensor starts moving and will receive a stronger beacon signal from SGW_2, the sensor sends its current session with $Send\_Old\_Session(session)$ to the new SGW_2. After the authentication the sensor node receives a new session. SGW_2 starts the negotiation with SGW_1 with $Mobility\_Negotiation()$ and SGW_1 establishes a tunnel connection to SGW_2 via $Establish\_Tunnel()$.

In the case that SP has or can not be updated with the underlying change the old SGW can reroute queries to the new one as shown in Fig. 4. The user wants to subscribe a service at the SP which will query the old SGW_1 via $Gateway\_Query(\text{SGW}_1, sensorId)$ to retrieve the sensor data. The SGW_1, however, is not in control of the sensor device, but is aware of the new location which has been exchanged while the $Mobility\_Negotiation()$. The old SGW will forward the query from the SP to the new SGW via the established tunnel. SGW_2 can query the node via $Query\_Sensor(sensorId)$ and send this data back via the tunnel to the SGW_1. In case that the node at the new SGW is not available any more, cached data (as in the caching mode) can be used. The old SGW will reply to the SP with the tunnelled data and the user can be served.

However, the trade-off in this mechanism is the longer response time for service consumption as the query will get rerouted from one SGW to the other which increases the messaging.

V. Implementation and Evaluation

To evaluate the resource mobility scheme we simulated the scenario by including a service provider, several SGWs and mobile sensor nodes in our simulator that is discussed in [10].

To evaluate the performance of the proposed schemes, we use: average response time (avg), minimum response time (min) and maximum response time (max) metrics in microseconds as shown in Table I. The error rate is defined as the ratio between successful and failed queries.

We assume a slow mobility scenario and a fast mobility scenario with high frequency of node movements between different gateways. We evaluate the two different modes in

TABLE I: Comparison between different modes and direct access

	mode	avg	min	max	Error Rate
Fast Mobility Pattern					
	Tunnel Mode	10100	2090	10149	0.0
	Cache Mode	9421	2018	10106	0.0
	Direct	9635	2023	10107	0.58
Slow Mobility Pattern					
	Tunnel Mode	9981	2024	10112	0.0
	Cache Mode	9625	2020	10011	0.0
	Direct	9632	2012	10100	0.34

terms of response time and error rate (=service unavailability) Another parameter taken into account is the frequency of connectivity loss between sensor and any gateway. In the slow mobility pattern, we assume that a sensor node changes its location from one gateway to another gateway every 2 seconds. In the fast mobility scenario, the frequency is set to a higher value. (Every 0.2 seconds)

The experiment setup consists of 10 nodes which can connect to two different gateways. We simulate 10 requests that query a service from SP. The requests each send 10 queries to SP, resulting in a total of 100 queries. We take the averages of 50 runs. The evaluation results of this experiment are shown in Table I. The Apache JMeter Benchmark tool[1] is used to simulate simultaneous HTTP client requests to submit queries to the service provider.

1) slow mobility scenario - no interrupt through loss of coverage

In this case, we assume that the node does not lose any connection to a gateway. Referring to Table I, we can see that in both modes the service availability is higher than the direct mode which runs without any mobility mechanisms and has an error rate of 34%. The response times are not significantly different in all three modes. Nevertheless caching mode is faster (but less accurate) than the direct mode, and the tunnelling mode has the highest communication steps (but it is more accurate).

2) fast mobility scenario - no interrupt through loss of coverage

The results of the fast mobility scenario, shown in Table I, are similar to the slow mobility scenario in terms of response time. The significant change is the higher loss of service in the direct mode (58%). This is due to the fact that a higher number of node mobility between gateways leads to a higher handover messaging and therefore longer delays in establishing a connection.

VI. Conclusion

This paper analyses the challenges in providing service continuity mobility scenarios involving a number of mobile nodes and multiple gateways. We assume that the nodes are connected through gateways. Service interruption happens when mobile nodes lose their connection while moving from one gateway to another or the service provider can not be updated after moving a node has associated to a new gateway.

We propose a resource mobility scheme with two modes: caching and tunnelling to support the service continuity. We have evaluated our solution in terms of service availability and response time by changing the frequency of mobility of nodes and gateway coverage. We compare and discuss the trade-offs of the two proposed mechanisms and discuss data freshness and higher response time issues for different scenarios.

[1] http://jmeter.apache.org/

Acknowledgment

Frieder Ganz and Payam Barnaghi would like to thank the ICT project ICT-257521, IoT.est, which is partly funded by the European Union. The authors would like to acknowledge the contributions of their colleagues, although the views expressed are those of the authors and do not necessarily represent the project. The authors would also like to acknowledge support from the *Japanese Society for the Promotion of Science* (JSPS).

References

[1] S. Weerawarana, F. Curbera, F. Leymann, T. Storey, D. Ferguson, "Web Services Platform Architecture: Soap, Wsdl, Ws-Policy, Ws-Addressing, Ws-Bpel, Ws-Reliable Messaging and More", Prentice Hall PTR Upper Saddle River, NJ, USA 2005.

[2] R. Fielding,T. Taylor, N. Richard, "Principled Design of the Modern Web Architecture", *ACM Transactions on Internet Technology (TOIT)*, Volume 2 Issue 2, pp. 115 - 150, 2002.

[3] G. Mulligan, "The 6LoWPAN architecture", *ACM Proceedings of the 4th workshop on Embedded networked sensors)*, pp. 78 - 82, 2007.

[4] Z. Shelby, K. Hartke, C. Bormann, B. Frank, "Constrained Application Protocol (CoAP)", *Internet Engineering Task Force Draft, http://tools.ietf.org/html/draft-ietf-core-coap-09,* 2009.

[5] F. Ganz, P. Barnaghi, F. Carrez and K. Moessner, "Context-Aware Management of Sensor Networks", *The Fifth International Conference on COMmunication System softWAre and middlewaRE (COMSWARE11)*, Article No. 6, 2011.

[6] L. Junhai, Y. Danxia, X. Liu, F. Mingyu, "A survey of multicast routing protocols for mobile Ad-Hoc networks", *IEEE Communications Surveys & Tutorials*, pp. 78 - 91, 2009.

[7] F. Thiesse, M. Kohler, "An Analysis of Usage-Based Pricing Policies for Smart Products", *Electronic Market*, pp. 232 - 241, 2008.

[8] D. Guinard, V. Trifa , S. Karnouskos, P. Spiess, D. Savio, "Interacting with the SOA-Based Internet of Things: Discovery, Query, Selection, and On-Demand Provisioning of Web Services", *IEEE Transactions on Services Computing*, pp. 223 - 235, 2010.

[9] D. Johnson, C. Perkins, J. Arkko, "Mobility Support in IPv6", *Internet Engineering Task Force: RFC3775*, 2004.

[10] F. Ganz, P. Barnaghi, F. Carrez, and K. Moessner, "A Mediated Gossiping Mechanism for Large-scale Sensor Networks", *The International Workshop on Machine-to-Machine Communications (IWM2M), GLOBECOM*, pp. 405 - 409, 2011.

[11] T. Elsaleh, A. Gluhak, K. Moessner, "Service Continuity for Subscribers of the Mobile Real World Internet", *IEEE International Conference on Communications Workshops (ICC)*, pp. 1 - 5, 2011.

[12] J. Li, R. Li, and J. Kato, "Future Trust Management Framework for Mobile Ad Hoc Networks", *IEEE Communication Magazine*, pp. 108 - 114, 2008.

[13] R. Li, and M. Inoue, "Secure Sensor Sharing Framework for Mobile and Sensor Access Platform Network", *IEICE Transactions On Communications*, pp. 1565 - 1576, 2011.