Context-Aware Management for Sensor Networks

Frieder Ganz, Payam Barnaghi, Francois Carrez and Klaus Moessner
Centre for Communication Systems Research
University of Surrey
Guildford, GU2 7XH, UK
{F.Ganz, P.Barnaghi, F.Carrez, K.Moessner}@surrey.ac.uk

ABSTRACT

The wide field of wireless sensor networks requires that hundreds or even thousands of sensor nodes have to be maintained and configured. With the upcoming initatives such as Smart Home and Internet of Things, we need new mechanism to discover and manage this amount of sensors. In this paper, we describe a middleware architecture that uses context information of sensors to supply a plug-and-play gateway and resource management framework for heterogeneous sensor networks. Our main goals are to minimise the effort for network engineers to configure and maintain the network and supply a unified interface to access the underlying heterogeneous network. Based on the context information such as battery status, routing information, location and radio signal strength the gateway will configure and maintain the sensor network. The sensors are associated to nearby base stations using an approach that is adapted from the 802.11 WLAN association and negotiation mechanism to provide registration and connectivity services for the underlying sensor devices. This abstracted connection layer can be used to integrate the underlying sensor networks into high-level services and applications such as IP-based networks and Web services.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*; C.2.2 [**Computer-Communication Networks**]: Network Operations—*Network Management*

General Terms

Management, Design

Keywords

sensor network management, zero-configuration, plug and play, context-awareness, middleware

DOI 10.4108/comsware.2011.9

1. INTRODUCTION

Wireless Sensor Networks (WSN) are often used to measure and control physical objects. These days sensors are more and more integrated into our daily applications and services. Smart homes, smart environments and the Internet of Things (IoT) are some of the current initiatives that leverage the capabilities from sensors to enhance our daily lives.

One effect of the increasing trends in exploiting sensor networks is that on the one hand the amount of nodes raises and on the other hand different standards based on different hardware, software and protocols are introduced. The problem which arises is to manage this huge amount of heterogeneous sensor nodes and their data. Managing a network includes configuration, maintenance, control and provisioning of networked systems [**?**].

In traditional wired networks, the main focus lies on fast response time and data throughput. However, in wireless sensor networks the primary goal is to reduce the use of energy and resources and therefore reducing communication between the network nodes. To connect and manage sensors to the middleware that provides high-level connectivity, a plug-and-play approach is targeted. This will minimise the configuration effort for network engineers and system designers. A simple solution similar to 802.11 WLAN association standard has been developed which allows zero-configuration management of the nodes. Nodes register to nearby base stations from where they are controlled. Each node publishes its context-information such as distance to base station, radio strength, time of (packet) arrival, latency the management strategy is adapted. This will provide intelligent control and access mechanisms to the nodes, for example:

Nodes with weak battery will be queried less (in non critical queries), queries could be handled from the cache to save energy depending on the data freshness preferences specified by the consumer. The sensing data from faulty nodes could be served from the nodes in the same spatial area and similar capabilities.

In sensor networks, often several types of sensors are found. The difference can be in hardware, capabilities and software platform. The middleware presented in this paper provides a generic approach to different platforms, operating systems and protocols which are managed through the same central system. In this paper we assume that the lower layers from physical to session layer are implemented efficiently. Efficient in the sense that optimisations for energy-saving methods and routing over several hops are guaranteed. The

scope of work is to provide a homogeneous connection layer which uses the information from the underlying layers to seamlessly integrate the heterogeneous sensor nodes to high level services and application networks. We provide a connectivity abstraction layer that interacts with different platforms using specific plug-ins developed for each particular platform.

2. CONTEXT-AWARENESS IN SENSOR NETWORKS

The sensor node context is the information which is not related to the real sensing operation and information. This can be the current battery status, location, radio information and its capabilities; in general the context and information that helps to understand more about the sensor and its surrounding [**?**]. This information, for example, can be used to make decisions on how queries from the user to the sensors can be distributed.
Despite the energy-optimisation, nodes tend to be faulty due to internal factors such as low battery and external factors such as environmental conditions and radio-telecommunication issues. Another problem is to find and locate the wireless nodes. If they are deployed in a wide area, routing and discovery is also an issue which has to be considered.
The context information from the sensor nodes can be used to address some of these issues. The energy consumption can be reduced and emerging bottlenecks can be detected. If we discover a sensor node with low energy status or heavy traffic required to query the node due to a great hop distance between node and sink,this can be compensated by introducing caching and replacement strategies wherever applicable. In order to create a uniform base, platform depending wrappers are used to create a generic access interface to the nodes. On the sensor side management modules are introduced which implement the proposed negotiation and association mechanism according to the capabilities of the particular system. On the sensor network edge, each module has a counterpart which establishes the connection to the nodes. Table 1 gives an overview of the requirements and their occurrence in various systems. Since this is a work-in-progress only some of the existing systems are evaluated.

Requirement	SunSpot	TinyOs	Contiki	6LoWPAN
Battery Life Information	X	X	X	
Signal Strength	X	X	X	
Hop Distance	X	X		X
Response Time	X	X	X	X

Table 1: Context Information provided by different approaches

We have created a fundamental set of attributes for context information used by the proposed management system. However additional information can be used to get more detailed information from a node.To design and implement our middleware solution , we focus on several IEEE 802.15.4 based platforms:
1) the Oracle java-based SunSpot [**?**] architecture which supplies special java capable nodes and the therefore required operating system.
2) The TinyOs operating system (OS) [**?**] which is an open source OS supporting several different microcontroller families and radio chips with a large user community.
3) Contiki [**?**] a similar open source (OS) which also supports several different hardware devices.
4) As there is an increasing interest to run IP Stacks on the sensor nodes we also introduce the integration of 6LowWPAN [**?**] protocol into our system. As 6LoWPAN does not supply context information such as battery life signal strength, it depends on the underlying OS, if the data can be provided.
The battery life information will be used to get the information if a specific node should be queried in non critical interactions or not, and if the data value should be supplied by a cache running on the middleware. The querying of a specific node also depends on other factors such as number of hops in between and the priority of the needed freshness from the query.
We use the received signal strength indication (RSSI) which is common for IEEE 802.11 based networks and can be obtained from most of the observed platforms. As different systems have different measurement ranges, we used approximate mappings between them. The signal strength information and other factors such as other supplied location data can be used to identify the position of a node.

Using the information of the amount of hops between a particular node and a gateway and also the RSSI and battery information of each hop node. We can obtain information about the distance to the specific node and the probabilistic value of availability to the node. The hop Information is supplied by the underlying MAC and routing layers.

Time of Arrival (TOA) could also be used to get distance and location information of the sensor nodes. TOA refers to the travel time from a transmitter to a receiver.
All of this data is gathered and processed by the middleware connection layer. The connection layer is used as a base for higher layers. On top of this layer we build the application layer which can simulate IP connectivity or Web Service integration and orchestration. The advantage of using a middleware approach instead of using node based implementations such as 6LoWPAN or executing web services directly on the node is that for each change in the application, software on each node has to be redeployed. In a network of several hundred nodes redeployment is difficult. Either each node has to be deployed manually which requires a lot of time or if deployment over the air is supported heavy network traffic and therefore battery consumption is needed.

3. RELATED WORK

The goal of our work is to create a framework to connect, register and manage sensors in a middleware without the time consuming task of manual configuration. Our focus lies on the higher integration beyond classical connectivity protocols such as MAC and IP based connectivity. The 6LowPan [**?**] standard enables the IP stack on sensor nodes: This requires high capable nodes which are able to implement the required standard. However without the use of further application protocols such as the Simple Network Management Protocol (SNMP) or SNMP for wireless networks, it is not easy to obtain context and status information other then using Internet Control Message Protocol (ICMP) data from the node. ICMP as part of the IP protocol allows to get the routing information as well as the transmission time between nodes. SNMP enables network management

functionality but its focus lies more in the status of the network device itself.
However, these protocols are not suitable for Wireless Sensor Networks due to high overhead and also WSN often use different protocol stacks. The approach from Schor et al. [?] introduces a zero configuration mechanism for IP based sensor networks. Advertisement messages are used to give the nodes the information on how to get management information. When devices are connected, they offer their capabilities to the network by sending broadcast messages. In this approach, the multicast Domain Name Service (mDNS) [?] also known as Bonjour is used. mDNS uses the multicast address of the internet protocol to distribute the available hosts or nodes. In the case that a node wants to get access to a specific node it makes a multicast request. Each node stores its own DNS entry and can therefore respond to the query. However those standards are only for IP based networks they allow only the integration of IP enabled sensor nodes. Shaman [?] is a scalable service gateway which uses proxies to integrate sensor devices into homogeneous platforms. One important aspect of Shaman is providing services without manual configuration. The identification of sensors abilities are provided by the sensor itself. A sensor gets a specific lease time and if the lease time is expired and the sensor did not request a new lease (like in DHCP Protocol), the service associated to the sensor is closed. The Shaman service gateway uses a boot protocol running on each node. The protocol tries to establish a connection to the nearest gateway. However establishing the connection is made by the node. The node sends a request and the first responding gateway is chosen by the nodes. We extended this negotiation approach by not simply allowing the node to randomly decide which gateway it will be use, we take the nodes context and deliver this information to the gateway. The gateway then decides how long it will manage this particular node, or if another nearby gateway should register and manage the node.

The Global Sensor Network (GSN) [?] is a middleware that, abstracts from the underlying, heterogeneous sensor network technologies. GSN-specific wrappers are used to connect different types of sensors. Each sensor has to be defined in 1) a wrapper which abstracts the hardware 2) a virtual Sensor which either represent a sensor in the overall GSN framework or combines several sensors and actuators. There is configuration work to be done to get a seamless integration of the sensors. The automatic configuration and deployment capabilities are very limited as each sensor (or virtual sensor) has to be defined in a configuration file. The approach can be used as data source for the GSN platform to provide low level connectivity. The Open Geospatial Consortium (OGC) [?] introduces middleware standards to develop a sensor observation framework. A part of the standard is SensorML a language which describes detectors, actuators, filters, and operators as process models. In our approach the proposed middleware tries to operate without configuration files but in future work SensorML configuration files can also be automatically generated by the framework to integrate into the OGC standards. The SENSEI project [?] offers a framework to integrate heterogeneous sensor nodes into the overall architecture. It uses mainly IP-based nodes but also offers mechanisms to find non IP-based ones and tries to integrate them. Those mechanisms are mainly based on Restful Web services and low-level sensor connectivity. The aforementioned approaches either focus on a limited platform based systems and therefore do not allow heterogeneous integration or need high configuration effort such as manual creation and change of files that limits their applicability in large scale deployments.

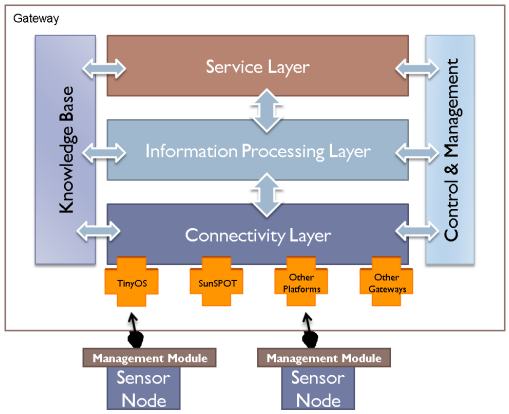

Figure 1: Architecture overview

4. SENSOR DISCOVERY AND REGISTRATION

The architecture is divided into three main layers. We focus on the connection and knowledge layer but in our further research we will build upon this layers to include the sensors context-awareness in higher layers. The layers and main interactions are shown in Figure 1. The connectivity layer establishes the connection with the sensor networks. Connector Modules for each sensor/protocol platform are connected to a sink node. External nodes connect directly to these sink nodes or via multi-hop connections. It provides a common interface which high level applications and services can access the underlying sensor network and its capabilities. When a new node is activated, the information about the node and the association process is stored in the device registry designed in the middleware component. The device registry contains information about the sensor type and capabilities, association time, lease duration, and the context information mentioned before. This information are later used by other applications. The registry is part of the knowledge layer. Despite the device information also semantic-notated information is stored in this layer. Each sensor type has a semantic description template which will be instantiated when a new node is associated. The goal of this annotation is to get a semantic description that other processes can exploit to infer the status and capablities of each node. This concept is clarified in Section 5. The Information processing layer uses the data from the connectivity layer and from the knowledge layer. In this layer intelligent data analysis is done. Different algorithms and mechanisms can be deployed in this layer to discover different patterns and/or events from sensing data or correlate information. This information will be made available through the service provision layer which allows Publish/Subscribe service where users can be informed about the network. Despite the Publish/Subscribe approach other interfaces are introduced to get access to the underlying layers. To establish a reliable connection between nodes and gateway, a similar approach

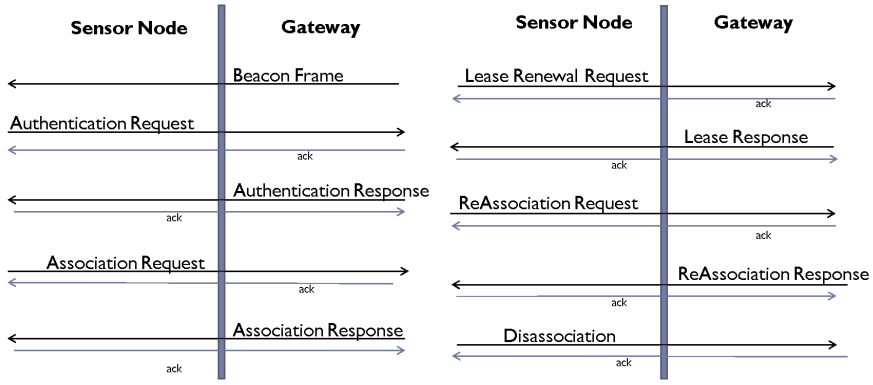

Figure 2: Proposed Association and Negotiation Protocol

as the association and negotiation protocol from the IEEE 802.11 Standard has been adopted. Before a device is able to exchange information it has to be registered in at least one gateway. The gateway will send a beacon signal every few second which the devices can use to register themselves to the gateway. First, after receiving a beacon signal the node has to verify if it is allowed to connect to a specific gateway. This is done by sending an authentication request to the gateway. The request includes the ID of the device and the encryption setting if available. The ID is needed to identify if the device is allowed to connect. Encryption setting can be also used to establish a secured connection between device and gateway. Messages send by a device or by the gateway must be acknowledged, if a request or response does not get acknowledged in a certain amount of time (depending on the distance between device and gateway) the request or response will be send again. The response time can be also used to approximate the distance between a node and the gateway.
The authentication response tells the node if it is allowed to proceed in the negotiation process or not. If it is allowed, the device has to request association with an association request. The request includes: battery status, signal strength, hop-distance (if possible) and the capabilities of the device and their current state . The middleware will send a response, which includes the lease time of the association. During the lease time period, the device must renew its lease to indicate the middleware that it is still available. The lease renewal process requires that the renewal request contains the same information as an association request. The response contains the new lease period. The information about the capabilities and the current sensing information is used to build a cache of information. In the case that the sensor becomes unavailable or battery life is low and an approximate measurement is needed, the cache can be exploited. If accurate data is needed, the accuracy and freshness information will be provided via the processing layer. If the lease expires without renewal, the device is deleted from the device registry. The lease duration time depends on the battery life and the distance to the node.

User queries will be routed to find the right sensor island/sensor node amongst different gateways which can satisfy the request by the user. In a first approach the gateways are linked together in a peer-to-peer way. Each gateway knows its neighbors. If a gateway can not satisfy a request by the user the query is send to the connected gateways. This simple flooding mechanism will be later improved by hierarchical device indexes and or intelligent knowledge distribution mechanisms. However, data/service discovery mechanisms require more comprehensive and efficient solutions, that are considered in our ongoing work.

5. KNOWLEDGE MANAGEMENT

The information of the nodes is stored in the knowledge layer. There are three main repositories for the information which is gathered by the gateway.
The device registry stores the data needed to get information about which devices are registered, when they are associated, when the lease expires and what type of sensors are connected.
The data storage saves the information about the sensors capabilities. During association and lease renewal the sensor delivers its current sensing information. Each transaction is saved to build a session which can later be used to compensate missing nodes or save transmission activity in the network.
The semantic data storage saves the information of the nodes in a machine-interpretable representation. As an upper ontology we use the Semantic Sensor ontology[1] from the W3C Semantic Sensor Network Incubator Group [**?**] and extended it to represent more context information in the model[2].

While the device registry and the data storage are updated on every transaction, it needs some time to change the Semantic Storage and its semantic interdependence on every update. This is why only meta information is stored in this storage. A semantic context model based on the W3C SSN ontology is developed. The SSN ontology allows to represent the information about sensor instances, each time a new sensor registers, a template for the sensor type is loaded and filled with the information from the sensor. In this case that there is no existing template for a specific sensor node, the assumption is that the middleware plug-in developer for the particular sensor type will provide the template or the sensor node will send semantically annotated data according to our model. The SSN ontology allows to give more information to a measurement capability of a sensor node, but does not provide information about the sensors context itself except for location and energy status. We introduce a new Entity "SensorContext" which can be used to describe the

[1]http://purl.oclc.org/NET/ssnx/ssn
[2]http://personal.ee.surrey.ac.uk/Personal/F.Ganz/ssnExtended.owl

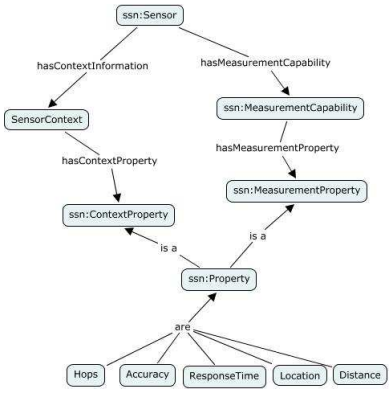

Figure 3: Extended W3C Ontology

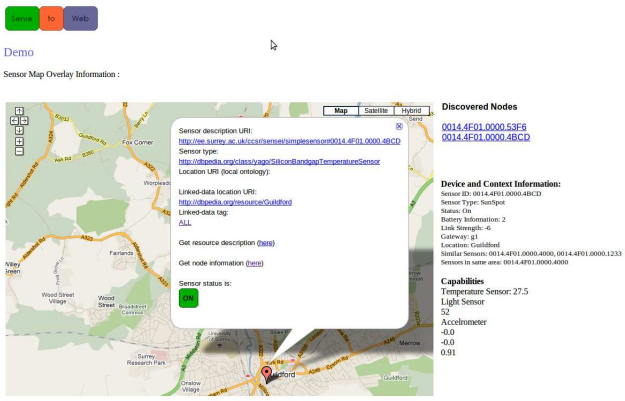

Figure 4: Sense2Web application

sensors context with the same properties used to represent a measurement capability see Figure. 3. This allows to obtain more information about the sensor like network information as signal strength and hop distance to the nearest gateway.

6. PROTOTYPE IMPLEMENTATION

To illustrate how the association and management of sensor networks throughout our middleware works, we have developed a demonstrator showing connected and associated nodes, the capabilities and the context data. As a base we use the linked sensor data platform developed at the University of Surrey which is called Sense2Web [**?**] and integrated our connectivity module with it. Sense2Web is a linked-data platform to publish sensor data and link them to existing resources on the semantic web. As our connectivity module produces individuals of the extended W3C SSN ontology it is less complex to connect both platforms.

In this prototype we use Oracle SunSpot nodes with the deployed management module to connect to the gateway. The gateway has a SunSpot compatible sink which send a beacon signal every three seconds. The SunSpot can receive the signal and start to authenticate against the gateway. In our current prototype we use any authentication and each node can request association by sending its type information, and the context information described earlier. The gateway will then create an association response. When the node associates to the gateway, a template for SunSpot nodes is loaded and respectively individuals are created from the instances. In Listing 1. a snippet from the ontology is shown where an individual of a node with its context information is created.

Listing 1: Instance of template

```
<rdf:Description rdf:about="#SUNSPOT-0000.8B54">
<hasSubSystem rdf:resource="#SunTemperatureSensor"/>
<hasConnectionInfo rdf:resource="#responseTime"/>
<hasRadioInformation rdf:resource="#SignalStrength"/>
<hasRoutingInformation rdf:resource="#hop_Distance"/>
<hasEnergyInformation rdf:resource="#energyLevel"/>
<rdf:type rdf:resource="#System"/>
</rdf:Description>

<rdf:Description rdf:about="#SensorContext">
<rdfs:subClassOf rdf:resource="#System"/>
<rdf:type rdf:resource="#Class"/>
</rdf:Description>
```

Sense2Web links the context information to other data sources. The geographical information such as the location of the gateway and the location of the nodes is shown on a modified google map[3] as it can be seen in Figure 4. Despite the geo location other context information as described in Section 2 is shown (See Figure 4).
In an inital evaluation the software was installed on 8 SunSpot nodes. They were activated one after another. The time period for associatons for this nodes was around 4 minutes. 145 messages sent including the beacon signal. The overall number of messages exchanged was 177 with a size of 4017 bytes and an average load of 67 bytes per node during the transaction process.

7. CONCLUSIONS AND FUTURE WORK

This paper presents an association and registration mechanism for sensor nodes. We use context information from each node to provide the status of the whole node. The approach reduces the configuration of discovery and registration of sensor nodes by introducing platform depended management modules in a platform independent middleware solution that automatically establish connection to the nearest gateway. The information about the nodes and the overall network is stored in a semantic context model which is used in the knowledge layer to support requests and interactions from high level services and applications. This work can be seen as a network enabler which abstracts from the technical underlying sensor network layer and provided easy access to it. Higher Layers such as information processing and service provision layers can seamlessly connect to middleware and request the network information from a more powerful and managed component. The management and the automatic integration of sensor nodes can support several large-scale networks and therefore create a framework where other software can leverage the power of the homogeneous access interfaces. In future work we will focus how information from the network can be provided to other applications and services. This includes service provision aspects such as providing web service abstraction from the sensor network to seamlessly integrate it into existing business solutions.

[3]http://maps.google.com

8. ACKNOWLEDGMENTS

This work has been performed in the framework of the ICT project ICT-5-258512 EXALTED, 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. This information reflects the consortiums view, the Community is not liable for any use that may be made of any of the information contained therein.

9. REFERENCES

[1] K. Aberer, P. Cudre-Mauroux, A. Datta, Z. Despotovic, M. Hauswirth, M. Punceva, and R. Schmidt. Infrastructure for data processing in large-scale interconnected sensor networks. *ACM SIGMOD Record*, 32(3):29–33, 2003.

[2] P. Barnaghi, M. Presser, and K. Moessner. Publishing linked sensor data. *Proceedings of the 3rd International Workshop on Semantic Sensor Networks (SSN)*, 2010.

[3] S. Cheshire and M. Krochmal. Multicast DNS. IETF draft, 2006.

[4] A. Clemm. *Network Management Fundamentals*. Fundamentals. Cisco Press, 1st edition, Nov. 2006.

[5] A. Dunkels, B. Gronvall, and T. Voigt. Contiki - a lightweight and flexible operating system for tiny networked sensors. pages 455–462, 2004.

[6] W3C semantic sensor network incubator group. http://www.w3.org/2005/Incubator/ssn/

[7] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, et al. Tinyos: An operating system for sensor networks. *Ambient Intelligence*, 35, 2005.

[8] G. Mulligan. The 6LoWPAN architecture. In *Proceedings of the 4th workshop on Embedded networked sensors*, pages 78–82, 2007.

[9] C. Reed, M. Botts, J. Davidson, and G. Percivall. Ogc sensor web enablement: overview and high level achhitecture. In *Autotestcon, 2007 IEEE*, pages 372–380, 2007.

[10] A. Schmidt, M. Beigl, and H. W. Gellersen. There is more to context than location. *Computers & Graphics*, 23(6):893–901, 1999.

[11] L. Schor, P. Sommer, and R. Wattenhofer. Towards a zero-configuration wireless sensor network architecture for smart buildings. In *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, pages 31–36, 2009.

[12] P. Schramm, E. Naroska, P. Resch, et al. A service gateway for networked sensor systems. *IEEE Pervasive computing*, pages 66–74, 2004.

[13] R. Smith. SPOTWorld and the sun SPOT. . *6th International Symposium on Information Processing in Sensor Networks*, 2007.

[14] V. Tsiatsis, A. Gluhak, T. Bauge, F. Montagut, J. Bernat, M. Bauer, C. Villalonga, P. Barnaghi, S. Krco The SENSEI real world internet architecture. In *Towards the future internet : emerging trends from European research*, 2010.