Confident-based Adaptable Connected objects discovery to HArmonize smart City Applications

Riccardo Petrolo, Valeria Loscrì, Nathalie Mitton

Confident-based Adaptable Connected objects discovery to HArmonize smart City Applications

Riccardo Petrolo, Valeria Loscrì, Nathalie Mitton
Inria Lille - Nord Europe, France. e-mail: {name.lastname@inria.fr}

***Abstract*—With the advent of the Internet of Things (IoT) and its need to boost cooperation between different *objects* in order to improve the quality and the completeness of the produced information, it is important to understand and to evaluate data which comes out and goes into each *thing*. In this paper we propose CACHACA, a ranking mechanism for Sensor Networks that facilitate the discovery of services provided by each network element. By running the proposed algorithm, it is possible to evaluate and classify the neighborhood and the available services for each node. Performances of CACHACA has been first evaluated through extensive simulations and them stressed when facing a realistic environment through experimentations run on the FIT IoT-LAB testbed. Achieved results demonstrate its effectiveness in the discovery of services process with regards to traditional approaches.**

Keywords*—*Internet of Things, Wireless Sensor Network, Service Discovery.

I. Introduction

In the last decades, wireless Sensor Networks (SNs) have been broadly investigated by both academic and industry fields. This tremendous attention gave birth to numerous applications in different domains (e.g., event detection, environment monitoring, etc.). However, those applications are typically purpose-built and therefore based on multiple architectures and standards making then a highly fragmented scenario.

The advent of the Internet of Things (IoT) [1] and its promise to connect *anyone from anyplace and anytime to anythings*, highlighted the need of interoperability between different *things*. As a result, various solutions have been proposed in literature to homogenize interfaces. IETF introduced 6LoWPAN [2], which defines mechanisms to fragment and compress the header of IPv6 datagrams into $IEEE$ 802.15.4 frames, enabling the integration of the physical world with computer networks. In [3], Guinard et al. define the Web of Things (WoT) by combining REST principles into embedded devices and making therefore possible the mash-up of physical and virtual worlds. Thanks to its ability to transform *everything as a service*, also the Cloud Computing is called to play a key role in the IoT revolution; the Sensing-as-a-service model introduced in [4], is at the base of the Cloud of Things (CoT) that we presented in [5], [6]; CoT aims to better use distributed resources, putting them together and thus enabling the horizontal integration of various *things*. On the other hand, CoT and in general Cloud services, have to face latency and intermittent connectivity issues. Recently, fog computing [7] represents an interesting approach to provide low latency, location awareness, and QoS for real-time applications. In all these paradigms and technologies, SNs keep playing a primary role, since they provide the major hardware infrastructure; at the same time, it is obvious that an evolution of the traditional application-specific design of WSN towards shared system design is needed and that the service model could represent a good enabler. In this context, with a huge number of resources, it is important to discover the ones (and only the ones) that are the more suitable for the application subject.

This work is partially supported by CPER DATA and by the European Community in the framework of the FP7 VITAL project.

In this paper, we propose a Confident-based Adaptable Connected objects discovery to HArmonize smart City Applications (CACHACA), a ranking mechanism for Sensor Networks that facilitates the discovery of services provided by each network element. By running CACHACA, it is possible to evaluate and classify the neighborhood and the available services for each node. In order to estimate the pertinence of neighbors and services, we leverage on the flexibility of the fuzzy logic and on its capacity to handle imprecise and incomplete data. CACHACA can be used in order to obtain:

- a **complete** information, by combining different data sources that offer different services;
- an **accurate** information, by combining different data sources that offer the same services.

The main novelties introduced by CACHACA rely on the adoption of the Sensing-as-a-Service model [4], which allows each network element to be seen as a service provider and the possibility to rank those services. Performances of CACHACA has been first evaluated through extensive simulations and them stressed when facing a realistic environment through experimentations run on the FIT IoT-LAB testbed. Achieved results demonstrate its effectiveness in the discovery of services process with regards to traditional approaches.

The remainder of the paper is organized as follows. Section II reviews the related literature. CACHACA is introduced in Section III and its performances are evaluated in Section V. Section VI concludes the paper and provides hints for future work.

II. Related Work

Within the IoT context, one of the most important challenges is to find appropriate services that satisfy user requirements; the community refers to this challenge as *service discovery* [8]. Generally, IoT services are published into registers that can be queried by users in order to obtain a list of candidate services.

Those registers are typically available as end-points of IoT platforms. In the last years, several solutions have been

proposed in literature in order to manage Sensor Networks. In this context, the use of semantic in IoT is recognized as one of the most important functionality to connect objects together [8]. In [9], authors point out on the need of a semantic representation to understand data which comes out and goes into the *things* interfaces; this "data exchange layer" could influence discovery and routing approaches and it can be crucial to enable scalability.

The benefits of semantic annotation are widely explained in [10]; to summarize the most representative are *(i)* re-use of Machine-to-Machine (M2M) data by many applications; *(ii)* "write-once run-anywhere" applications; *(iii)* easy adaptation in case of failures/changes of the available sensor sets.

In order to constitute semantic information in our context, two main options arise: either to use standardized data types, like the one defined by the IPSO Alliance in [11], or rely on ontologies, like the SSN ontology proposed in [12].

Niu et al. [13] proposed a context-aware service ranking approach by aggregating the user rating and WSN service context but do not consider a single device but rather the whole network. Durmus et al. [14] proposed a discovery protocol based on semantic representation of services; the mechanism operates in the network layer and can directly run SPARQL queries on top of those devices. Anyway, this approach is not suitable for SN context due to the lack of resources. Finally, [15] introduced a ranking strategy by estimating the cost of accessing sensor services using properties of the sensor nodes as well as relevant contextual information extracted from the service access process. In our approach, we do not consider the cost of the service, but instead, we evaluate its quality by considering some physical aspects (e.g., RSSI) related to the service provider node and features of the service itself.

III. Confident-based Adaptable Connected objects discovery to HArmonize smart City Applications

A. Assumptions and metrics

CACHACA distinguishes three different network elements (see Figure 1):
1) a ***node*** that has communication capabilities, and therefore able to communicate within other elements. If a node is equipped with some sensors (circles) we refer to it as **full node**; then it is capable to measure physical events, for example providing the temperature of a room, or the availability of a parking spot;
2) a ***relay*** is a node with communication capabilities;
3) a ***gateway*** that is a node in charge of gathering and managing data produced by sensors, and at the same time, enhanced to act as an end-point for the communication with the Internet or with other local devices. In this work we consider a gateway just as a service provider, like a node.

We assume that a Neighbor Discovery mechanism is running on each node u to allow u to discover other nodes v in communication range. So, at a frequency f, each node receives information about its neighborhood that it stores in a Neighbor Table (NT). Note that the exact format of NT is implementation-specific, but according to [16] it should contain, at least, the following for each neighbor v of u:

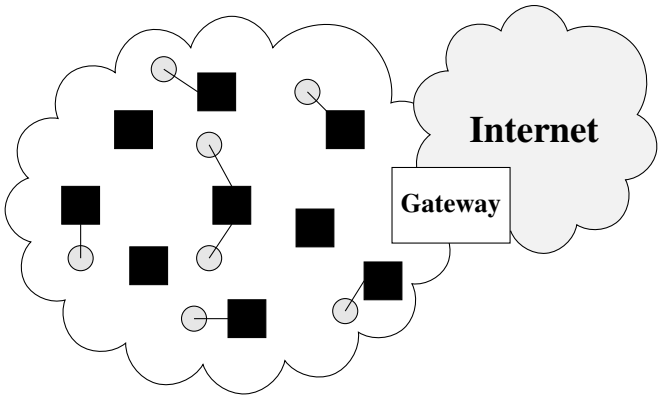

Fig. 1: Network elements: *Node* (square), *sensor* (circle), and *gateway* (rectangle).

- *numTx*: number of transmitted packets to v;
- *numTxAck*: number of packets acknowledged by v;
- *numRx*: number of packets received from v;
- *Timestamp* of the last frame received from v;
- *Connectivity statistics* (e.g., RSSI, LQI), which can be used to determine the quality of the link.

At the same time, we suppose that each node uses a standardized format (e.g., IPSO [11]) for describing its services (i.e., temperature, light, humidity, etc.). Each service is combined with other complementary information such as:
- *freshness* of the information; can be real-time or temporized;
- *provider*: to specify whether the service is directly provided by the node itself or by a neighbor.
The above parameters can be used in order to define relationship of a node with the neighborhood. In this sense, in this work we use some of the above parameters in order to introduce two additional functions, the PHYSICAL CONFIDENCE (φ) - based on the *RSSI* and *Timestamp* - and the SERVICE CONFIDENCE (ω) that is computed based on the service information.

RSSI (Received Signal Strength Indicator) represents the measured power of a received radio signal; it is widely used in different standards (e.g., IEEE 802.11). According to [17], the RSSI is reported as an integer ranging from $-100\ dBm$ to $0\ dBm$; in this work we normalize it as a value from 0 to 100.

B. Fuzzy logic

In order to compute the physical confidence, we use a rule-based fuzzy inference system [18]. A fuzzy logic system can be developed in three steps:
- **Definition of fuzzy sets (fuzzification).** In this first round non-fuzzy inputs (i.e., numbers) are converted into fuzzy sets by using membership functions (e.g., triangular, trapezoid, singleton, bell, or some other type of function).
- **Definition of fuzzy rules.** Expressed as statements like "IF … THEN …", the fuzzy rules summarize the relationship between the fuzzy sets and the output variable.
- **Defuzzification.** This last stage is used to convert the fuzzy output back into a value that can be later used to make decisions.

C. Physical confidence computation

The physical confidence is computed based on fuzzy logic rules applied to RSSI and timestamp collected by the neighbor discovery protocol in a local and distributed way by each node u for each of its neighbors v. We thus consider three fuzzy sets based on the RSSI values : BAD, GOOD, and EXCELLENT as displayed by Figure 2 shows the diagrammatic representation of the RSSI that is computed using a trapezoidal membership function.

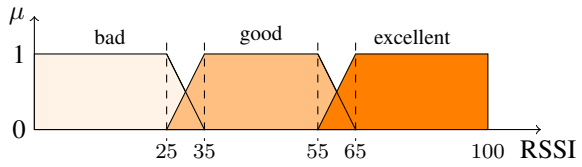

Fig. 2: Diagrammatic representation of RSSI.

The other parameter used for the estimation of φ is the Timestamp; in this case, we consider the difference Δt (Eq. 1) between the instant at which the computation process is executed (t_{now}) and the Timestamp $t_{timestamp}$ stored into the NT.

$$\Delta t = t_{now} - t_{timestamp} \quad (1)$$

Once Δt is obtained, we consider again three fuzzy sets: BAD, GOOD, and EXCELLENT. Since we supposed that the application is time-constrained, we favor small values of Δt (Fig. 3); therefore a node that provides services in real-time will be highly compared to one with higher values of Δt.

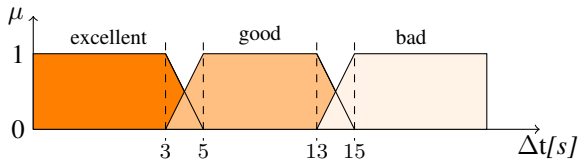

Fig. 3: Diagrammatic representation of Δt.

After completing the fuzzification process, we apply the fuzzy rules to obtain the physical confidence. Table I shows the definition of the rules in CACHACA. An example can be observed as: "**IF** RSSI is *Excellent* **AND** Δt is *Excellent* **THEN** φ is *Excellent*". It is worth noting that we give more importance to the time parameter. Indeed, when the RSSI is *Good* and Δt is *Excellent*, neighbors are still noted as *Excellent*; this is because a communication can be completed even with lower RSSI, on the other end, if the neighbor is not often active, it is important to classify it as *Bad*.

D. Service confidence computation

The service confidence (ω) is computed by each node considering one of its mono-modal services per time (e.g., temperature); in this case we use just the *Freshness* feature. As shown in Table II, ω is considered *Excellent* when it is possible to access in real-time to the values of the services.

TABLE I: Rule based fuzzy inference.

$RSSI$	Δt	φ
Excellent	Excellent	Excellent
Good	Excellent	Excellent
Excellent	Good	Good
Good	Good	Good
Bad	Excellent	Good
Excellent	Bad	Bad
Good	Bad	Bad
Bad	Good	Bad
Bad	Bad	Bad

TABLE II: Service confidence computation for a Full node.

$Provider$	$Freshness$	ω
sensor	real-time	Excellent
sensor	temporized	Bad

IV. CACHACA

Yet, a node is now able to characterize the different confidence values for each of its neighbors periodically, for each packet received. Algo. 1 describes how the physical confidence is updated by u upon reception of a new packet from v. u checks whether v is already stored into its NT, if so, it updates its NT with the *RSSI* and the *Timestamp* and then it computes φ for each node present in its NT. If not, a new entry will be added, with the ID of v, the *RSSI* and the *Timestamp*; at this point u computes φ for each neighbor by applying the fuzzy logic rules above presented.

Algorithm 1 Physical confidence update - Run on node u upon reception of packet from node v.

1: **if** $v \in$ NT **then**
2: update RSSI and Timestamp values for v in NT;
3: **else**
4: add v in NT with associated RSSI and Timestamp;
5: **end if**
6: $\forall\ w$ in NT, update $\varphi(w)$ following Table I.

In order to discover efficiently the different services available, nodes advertise their services and the associated confidence periodically and can relay the information about a service offered by a neighbor. The format of the frame is shown in Fig. 4; the $Service$ uses 10 bytes, while the confidence can be transmitted by using only 1 byte. Considering that the length of the MAC frame of IEEE 802.15.4 can be maximum 127 bytes, and subtracting 31 bytes of header and 2 bytes of footer, in one message we could advertise up to 7 services.

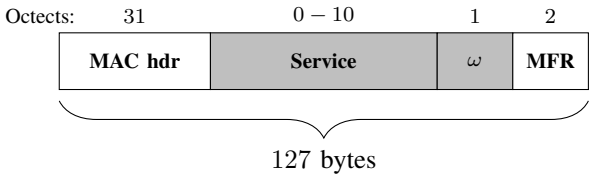

Fig. 4: Frame format with *Service* and *Service confidence*.

Upon reception of such a message from a neighbor v, a node u can thus upgrade entry of v in its NT with these values as shown in Table III; for each neighbor, the information stored will be: the offered *Service*, the *ID*, the two *confidences* (physical and service), the *RSSI*, and the *Timestamp*. This Neighbor Table is used to evaluate the neighborhood and to rank the neighbors. Furthermore, the NT can be cleaned by removing deprecated entries for space saving purposes.

TABLE III: Neighbor Table.

ID	$Service$	ω	φ	$RSSI$	$Timestamp$
1	temp	excellent	good	80	1431108000
30	light	good	good	50	1431108008
2	temp	excellent	excellent	90	1431108007
...					

When the information about a service is relayed, the ω confidence is also function of the physical confidence. Table IV shows how ω transmitted by the relay node is influenced by φ and ω of the Neighbor that provides the service; the best value that we can obtain is *Good* and it is verified when the $\omega_{Neighbor}$ is *Excellent* and $\varphi_{Neighbor}$ is *Excellent* or, at at least, *Good*; while, even when the $\omega_{Neighbor}$ is *Excellent*, ω will be *Bad* if the $\varphi_{Neighbor}$ is *Bad*. Algo. 2 shows the process of advertisement of a service by a relay node; this can be done just when the ω is *Good*.

TABLE IV: Service confidence computation for a Relay node.

$\varphi_{Neighbor}$	$\omega_{Neighbor}$	ω
Excellent	Excellent	Good
Good	Excellent	Good
Good	Bad	Bad
Bad	Bad	Bad
Bad	Excellent	Bad

Algorithm 2 Service confidence computation for a Node. - Run at node u upon reception of a packet from v

```
if v ∈ NT then
    update NT(RSSI, Timestamp) for NT.ID;
else
    store v in NT with associated RSSI and Timestamp;
end if
∀w ∈ NT do update φ(w) with Table I
if ((φ(w)) = (Excellent)) || ((φ(w)) = (Good)) then
    compute ω(w);
    if (ω = (Good)) then
        broadcast (Service, ω);
    end if
end if
```

V. Performance Evaluation

To evaluate the performance of CACHACA, we use Contiki-OS[1] and its simulation tool Cooja; Table V summarizes the principal parameters. Among the others (e.g., TinyOS, Riot[2]) we choose Contiki because its good assessment by the community, its completeness and re-usability; with Contiki indeed, it is possible to run simulations and then re-use the code to flash real devices. We consider an area of 200x200 m^2, in which M network elements are randomly positioned; M is the sum of R relays and N full nodes equipped with 1 sensor. Values of M, R and N and what they stand for depend of the scenario under evaluation as detailed later.

TABLE V: Simulator parameters.

$Parameter$	$Value$
Nodes radio chip	CC 2420
Nodes flash memory	1 MB
Simulation seed	random
Simulation runs ∀ scenario	10

We use the following metrics to assess the performances of CACHACA:

- $service_{avg}$ represents the average number of services discovered by each node;
- $neighbor_{avg}$ is the average number of neighbors discovered by each node;
- $packets_{avg}$ is the average number of packets transmitted by each node;
- ω_{avg} is the average value of the service confidence ω computed by each node;
- φ_{avg} is the average value of the physical confidence φ computed by each node.

We performed the simulations in five different scenarios (Table VI). In all scenarios, N is set to 5; each node advertises its own service periodically. In the first Scenario, we have just the 5 nodes running, while in the second and third scenarios we introduce some relay nodes. In the last 2 Scenarios, we consider that relay nodes can move inside the area with an average speed of 1 m/s. This set of Scenarios can be used to describe a generic smart city use case (e.g., smart building). A number of different sensors is available in distinct rooms; those sensors can offer services like temperature, luminosity, and so on; other devices (attached for instance to the smart-phones of employed) act as relay for the sensors' services. We chose to use only 5 Full nodes and evaluate the number of relay necessary to discover all the potential services. Moreover, we vary the number of relay between 10 and 15 because we want to study the behavior when the network is not highly dense and therefore avoiding to compare CACHACA with the *Broadcast scenario* that suffers from crowded cases.

TABLE VI: Simulator scenarios.

	N	$Services$	R_{fix}	R_{mobile}	M
Scenario1	5	5	0	0	5
Scenario2	5	5	5	0	10
Scenario3	5	5	10	0	15
Scenario4	5	5	0	5	10
Scenario5	5	5	0	10	15

[1]http://www.contiki-os.org

[2]http://tinyos.net, http://www.riot-os.org

A. Simulation results

Fig. 5 shows the average number of packets sent per node. We can observe that the $Packet_{avg}$ is higher in the first Scenario; while this number decreases with relay nodes. This is because relay nodes broadcast packets only if the service offered by the neighbor has ω = *Excellent* and the φ of the neighbor is at least *Good*. Introducing mobility makes $Packets_{avg}$ increase because of the higher possibility to meet nodes and therefore for relay to broadcast services. For the sake of equity and fairness, we have also considered that relay nodes broadcast a service immediately when it is discovered, without taking account of the quality (dashed lines in the Figure). In this case, we can observe that the number of messages increases intensely and so also the quality of the channel and the energy consumption will be negatively affected.

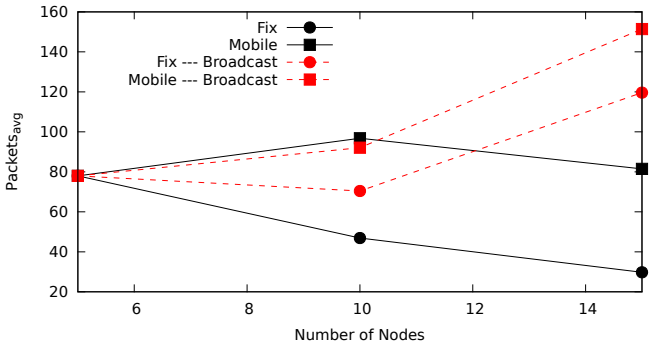

Fig. 5: Number of Elements vs. $Packets_{avg}$ sent.

Fig. 6 indicates the average number of Neighbors and Services discovered by each element network in function of the Number of Nodes. In the first Scenario, and in general when there is no mobility, the performance are bad, this because nodes and relay are randomly deployed on the field and therefore it is possible that they are not in communication range. With mobility (Scenario5), the performance improves; each node is capable to discover about 40% of the available services and more than 50% of neighbors.

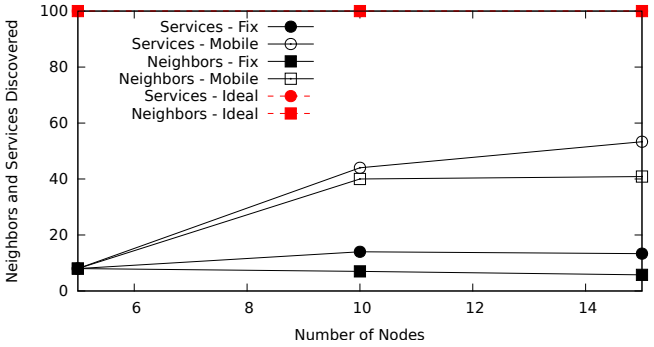

Fig. 6: Number of Nodes vs. Services and Neighbors Discovered.

In the last investigation (Figure 7), we consider the behavior of φ and ω in function of the Number of Nodes. We can observe that the average ω computed by each node for the discovered services is *Good*; this means that each node can discover more than 40% of the services provided with *Good* confidence. Regarding the Physical Confidence, we can note that it increases when mobility comes in play, but at the same time, φ decreases with a higher number of nodes, because of possible interference. Anyway, it is important to highlight that even when φ is *Bad*, a communication can happen.

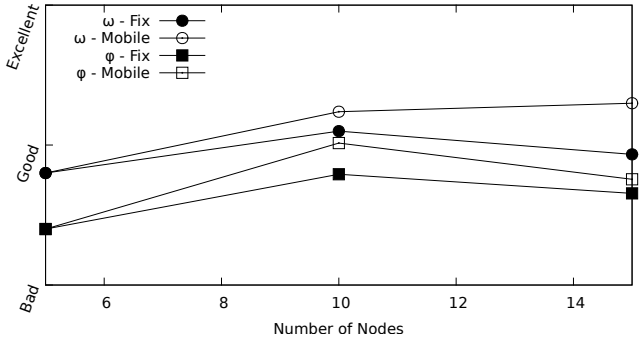

Fig. 7: Number of Nodes vs. Physical and Service Confidence.

B. Experimentation results

In order to face CACHACA to a realistic environment, we ran experimentation on FIT (Future Internet of Things) IoT-lab[3]; a very large scale infrastructure facility suitable for testing small wireless sensor devices and heterogeneous communicating objects over large scale. We used the Rennes site, and we performed experimentation (parameters available in Table VII) using Scenarios $1, 2$ and 3 (no mobility).

TABLE VII: Experimentation parameters.

Parameter	*Value*
Nodes type	WSN 430
Nodes radio chip	TI CC 2420 @ 2.4 GHz
Nodes flash memory	1 MB

Figure 8 shows the Services and the Neighbors discovered. We can observe that in this case, CACHACA has performance similar to the ideal scenario (dashed lines); when the Number of Nodes is 10, each network element discovers almost all the Services and about 70% of the neighbors; the same trend is maintained when the Number of Nodes is 15. Those results are in line with the ones obtained running simulation with Cooja; therefore we can conclude that when we increase the Number of Nodes the efficiency of our proposal is higher.

Regarding the Physical and Service confidences (Fig. 9), we can observe that both parameters have better performance when the network is sparse; this is because, the services are directly provided by the direct neighbor, without the intervention of relay; φ decreases with the Number of Nodes, because of more interference. The trend obtained in this analysis is once again complementary to the one obtained with the simulator.

[3]https://www.iot-lab.info

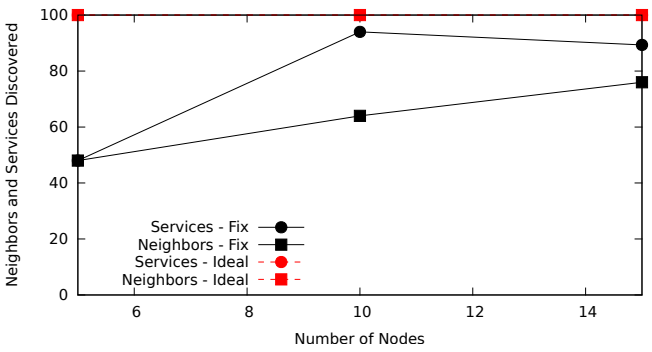

Fig. 8: Number of Nodes vs. Service and Neighbor Discovered (FIT IoT-lab).

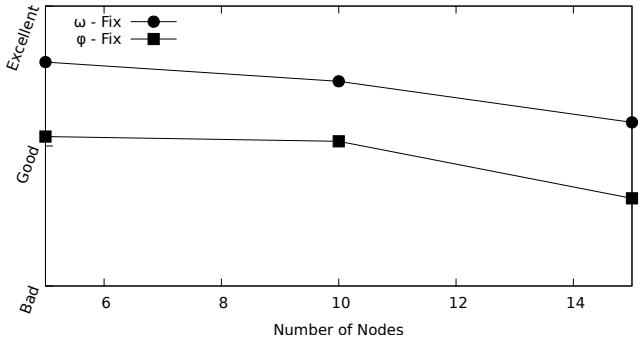

Fig. 9: Number of Nodes vs. Physical and Service Confidence (FIT IoT-lab).

VI. Conclusion

With the massive number of deployed *things* and the need of interoperability between those devices, it is important to understand which "service" each node can offer and to evaluate it. In this paper we have proposed CACHACA, a ranking mechanism for Sensor Networks that facilitates the discovery of services provided by each network element. CACHACA could run on different Smart City use cases.

Smart Building which different nodes are installed. Those nodes are equipped with various sensors in order to measure physical events, e.g., environmental conditions, room occupancy, building structure. Thanks to CACHACA, every node will be aware about the services available in its neighborhood; therefore they will be able to cooperate in order to achieve common application tasks, e.g., room environmental conditions, office occupancy.

Smart Street in which bus stops, cars, buses, and parking spots of a city are equipped with nodes running CACHACA. Thanks to the service discovery and to the ranking mechanism, a car can recognize the best node which offers parking facility and use it. Moreover, a bus could advertise the availability of free places and so on.

We evaluated performance under different settings both with simulation and experimentation. Results show that CACHACA performs better in terms of Packets sent while obtaining good results compared to the ideal scenario in terms of Services and Neighbors discovered.

In the future, we plan to study more sophisticated criteria for Service and Physical Confidences and to create a real testbed that validates the Smart Building scenario.

References

[1] K. Ashton, "That 'Internet of Things' Thing," *RFiD Journal*, 2009.

[2] N. Kushalnagar, G. Montenegro, D. E. Culler, and J. W. Hui, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks, RFC 4944 (Proposed Standard)," IETF, Tech. Rep., 2007.

[3] D. Guinard, V. Trifa, T. Pham, and O. Liechti, "Towards physical mashups in the Web of Things," in *Proceedings of the 6th International Conference on Networked Sensing Systems (INSS)*, Pittsburgh, Pennsylvania, US, Jun. 2009.

[4] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Sensing as a service model for smart cities supported by Internet of Things," *European Trans. on Telecom.*, vol. 25, no. 1, pp. 81–93, 2014.

[5] R. Petrolo, V. Loscri, and N. Mitton, "Towards a Smart City based on Cloud of Things, a survey on the smart city vision and paradigms," *Transactions on Emerging Telecom. Technologies*, pp. n/a–n/a, 2015.

[6] ——, "Towards a Smart City based on Cloud of Things," in *Proc. of the Int. ACM MobiHoc Workshop on Wireless and Mobile Technologies for Smart Cities (WiMobCity)*, Philadelphia, Pennsylvania, US, Aug. 2014.

[7] I. Stojmenovic, "Fog computing: A cloud to the ground support for smart things and machine-to-machine networks," in *Proceedings of the Australasian Telecommunication Networks and Applications Conference (ATNAC)*, Southbank, Victoria, Australia, 2014.

[8] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context Aware Computing for The Internet of Things: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 414–454, 2014.

[9] M. Hauswirth, P. Dennis, and S. Decker, "Making Internet-Connected Objects readily useful," in *Proceedings of the Interconnecting Smart Objects with the Internet Workshop*, Prague, Czech Republic, Mar. 2011.

[10] ETSI, "Machine-to-Machine Communications (M2M): Study on Semantic support for M2M Data," Tech. Rep., 2013. [Online]. Available: http://www.etsi.org/deliver/etsi_tr/101500_101599/101584/02.01.01_60/tr_101584v020101p.pdf

[11] Z. Shelby and C. Chauvenet, "The IPSO Application Framework draft-ipso-app-framework-04," Tech. Rep., 2012. [Online]. Available: http://www.ipso-alliance.org/wp-content/media/draft-ipso-app-framework-04.pdf

[12] M. Compton and et al., "The SSN ontology of the W3C semantic sensor network incubator group," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 17, pp. 25–32, 2012.

[13] W. Niu, J. Lei, E. Tong, G. Li, L. Chang, Z. Shi, and S. Ci, "Context-Aware Service Ranking in Wireless Sensor Networks," *Journal of Network and Systems Management*, vol. 22, no. 1, pp. 50–74, 2013.

[14] Y. Durmus and E. Onur, "Service Knowledge Discovery in Smart Machine Networks," *Wireless Personal Communications*, vol. 81, no. 4, pp. 1455–1480, Mar. 2015.

[15] W. Wang, F. Yao, S. De, K. Moessner, and Z. Sun, "A ranking method for sensor services based on estimation of service access cost," *Information Sciences*, vol. 319, pp. 1–17, 2015.

[16] X. Vilajosana and K. Pister, "Minimal 6TiSCH Configuration," 2015. [Online]. Available: https://tools.ietf.org/html/draft-ietf-6tisch-minimal-11

[17] Veris, "Veris Aerospond Wireless Sensors: Received Signal Strength Indicator (RSSI)," Tech. Rep., 2013. [Online]. Available: http://www.veris.com/docs/whitePaper/vwp18_RSSI_RevA.pdf

[18] L. A. Zadeh, "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 1, pp. 28–44, 1973.