# Enhancing Context-Awareness in Autonomous Fog Nodes for IoT Systems

Basil Nikolopoulos, Maria Voreakou, Mara Nikolaidou, Dimosthenis Anagnostopoulos
Department of Informatics and Telematics
Harokopio University of Athens
Athens, Greece
basil@hua.gr; voreakou@hua.gr; mara@hua.gr; dimosthe@hua.gr

*Abstract*— **The main aim of this research is to lay the foundation work for the development of a communication protocol which will enhance the context awareness of autonomous Fog nodes in IoT systems, by enabling them to share information on their status. Analysis is being conducted on numerous basic and current architectures, models and communication patterns used by IoT systems. A new, adjusted, model is proposed as a protocol for communication in IoT systems encompassing autonomous agents. Finally, modelization of the adjusted protocol (ECTORAS) is conducted, with main criteria the challenges of IoT, Fog, and Cloud computing.**

***Keywords: IoT systems, context-aware architectures, Fog nodes, knowledge sharing, communication protocols***

## I. INTRODUCTION

IoT and IoT systems have undoubtedly made their way to everyone's everyday life, improving it in a number of areas, with applications ranging from Smart-Health to Smart-Buildings and many more. As IoT solutions become more mainstream, the more complex the systems become. Numerous efforts have been observed lately with the aim of promoting distributed architectures fit for the complexity of IoT systems, ensuring the scalability and interoperability of the components encompassed in them. [1] [2]

A significant oversight of early IoT architectures was the underestimation of the load that would come hand in hand with large deployments of IoT systems, let alone an interconnected world. To tackle such issues, the architectures of Edge and Fog computing were introduced. [3] [4] [5] Both approaches aim at moving away load from the Cloud computing layer, and moving part of said load, that of data preprocessing, (filtering, aggregation etc.) to new layers. Each propose new entities, the Fog layer [4], and the Edge nodes [5] respectively. Both solutions are immensely helpful as they speed up processes and remove load from the cloud, which has become a bottleneck.

Another major goal yet to be solidified for IoT systems is the characteristic of autonomy. As defined by [6] an autonomic system is defined as "an intelligent system, or system of systems where data acquired by sensing or monitoring capability is utilized in an overall autonomic decision-making process." To that end, authors of [1] have proposed an IoT System architecture with autonomous components. Said components are able to achieve their autonomicity by use of a number of dynamic policies that aim at providing the autonomous components with Self-X properties, like Self-Adaptation and Self-Configuration.

At this moment, research on the system proposed in [1], has yet to provide of a formal model to cater for communication amongst autonomous components, and to cater for the share of information, and at a higher operating level, knowledge amongst them. To achieve this, two new features would have to be introduced in the architecture, a new internal unit for the autonomous Fog nodes, the knowledge diffusion unit, and a communication protocol to enable inter-node communications, both of which will be further discussed in detail in this paper.

The rest of this paper will follow the following format: Section 2 discusses related and relevant work on the subject. Section 3 provides a detail description of the proposed communication protocol, ECTORAS (Efficient Communication Technology Over REST Architecture Systems). Conclusions and future extensions of the proposed endeavors reside in section 4.

## II. RELATED AND RELEVANT WORK

### A. Challenges and Characteristics of Fog computing

In the recent years there has been a lot of research around the topics that are engulfed in the topic of Edge computing. A common discussion topic is the challenges that surround systems with Edge computing capabilities. In [7] the authors describe that the challenges of Fog computing, and as an extension Edge computing, fall under the following categories:

- *Heterogeneity* among Fog nodes
- *Constraint diversity* on available resources
- *Locality and geo-distribution* of nodes that are required to cooperate and provide one single service in a timely manner
- *Multi-tenancy* of services on a single node
- *Scalability*

### B. Communication protocols for the IoT

According to [8], the communication protocols used widely in the IoT are split in two categories, request-reply protocols, and publish-subscribe protocols. The authors further split the communication protocols in two additional categories, those that are primarily used for Fog to Cloud communications and IoT to Fog communications. The summary of this split can be seen in table 1

| | Fog to Cloud | IoT to Fog |
|---|---|---|
| Request - Reply | HTTP | CoAP |
| Publish - Subscribe | DDS, AMQP | MQTT, XMPP |

Table 1. Split of Communication Protocols for the IoT

Of the presented request-reply protocols are considered reliable, each for their own reasons. HTTP uses TCP to communicate its data, which in turn provides the underlying reliability. The HTTP protocol is used as a basis for the implementation of RESTful communications. [9] CoAP, on the other hand, is similar in nature to HTTP, as it is a stripped down version of HTTP working with binary headers to minimize communication overhead. In contrast to HTTP, it uses UDP to maximize communication speed, with extra communication verbs (CON, NON, ACK, RST) providing extra features to provide a layer of reliability. [10] [11] [8]

The presented publish-subscribe protocols work in essence by allowing users to subscribe to a topic and then communicate with each other based on the topics they are interested in. Two are the major characteristics that segregate these protocols: the existence of a message broker, responsible for distributing messages to the topic subscribers, and the existence of QoS levels for communication. All of AMQP, MQTT, and XMPP require a broker to function, which leads to a centralized mean of communication. [12] [13] [14] On the other hand, the DDS standard defines a publish-subscribe communication model that does not rely on brokers to distribute messages, but rather allows for peer to peer communication. [15] [16]

## III. ECTORAS PROTOCOL

ECTORAS Protocol is built from the bottom up, with the restrictions of IoT systems in mind. It is in essence a protocol based on pre-existing communication designs, specifically DDS and REST. The main aim of the protocol is twofold:

- Distribute knowledge freely within the IoT systems
- Enable communications among Fog Nodes

Knowledge distribution and diffusion within an IoT system can help accelerate processes, as in the migration policy's example, from [1], of picking a suitable candidate to transfer a resource to, and even go as far as to state that Fog Nodes become smarter and can achieve certain level of evolutionary traits, given the right policies. This knowledge diffusion is in essence what can lead to Context-Aware Fog Nodes and further boost the property of autonomy for each Fog Node of the system.

Communication among Fog Nodes was available in the initial implementation in [1], by means of the MUJA communication design, and we wanted to make sure that we also incorporate this in our protocol.

### *A. The DDS Design Protocol*

As discussed in chapter II, DDS is a publish-subscribe protocol without the need for a broker, promoting peer to peer communications. We believe that this kind of communication design is important for IoT systems, as the existence of a broker signifies a more centralized architecture, impossible to be achieved in constantly changing created by Fog and Edge components moving in space.

By accepting the system architecture of [1], we can store topic information on the Registry Unit, proposed by the authors as a unit to which autonomous components register their services and can look up available services by other autonomous components.

A topic for ECTORAS is described by the following information:

- A name
- A human readable description
- A machine parsable description
- A list of participants

Topics can be dynamically created by Fog Nodes for any reason, and similarly, Fog Nodes are able to poll the registry for any topic, and ask to be included in the participants' list. Topics can fall under any number of categories, some of which we propose are:

- Topics based on Geo-Location (Sensors of Room with name Kitchen)
- Topics in regards to sensor networks they are able to serve (Fog nodes able to communicate with ZigBee sensors for instance)
- Topics based on an aggregated service to be provided by a number of Fog Nodes that need to cooperate (All Fog Nodes that are responsible for maintaining the temperature of Room with name Living Room)
- Topics based on enstated polices (Aggregators implementing the QoS preservation policy [1] could be discussing whether they are open to accept transfers of resources)

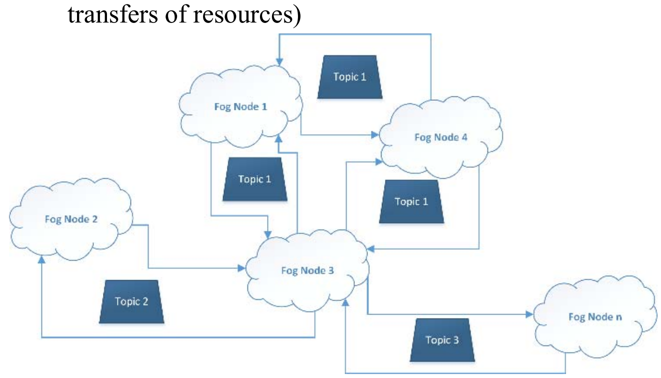


Figure 1. Resulting Peer to Peer communications based on topic subscriptions

A Fog Node is able to ask the Registry Unit for any piece of information regarding a topic, and also to request to be subscribed or unsubscribed to a topic. After subscribing in a topic, when the Fog Node wants to send a status update relevant to the topic's format and discussion, the Fog Node is required to poll the Registry Unit for the list of participants in the topic, and then communicate in peer to peer fashion with them. In Figure 1 we provide a visual example of the topics Fog Nodes are subscribed to and provide the peer to peer communications that would be the outcome of the subscriptions.

Discussions on topics should be limited to the information described by the machine parsable description of the topic. This could be a set of variables or state changes, and topic updates should happen whenever a Fog Node has something to state that has changed. For instance, a Fog Node that was open to accepting resources as part of a transfer that has now become overloaded due to its own load should send an update to the topic stating that it is no longer accepting transfers for resources. Other Fog Nodes, given this new information would then not try to contact the newly overloaded Fog Node as it would probably turn the request down, and secondly would add to the already existing load of the overloaded Fog Node.

It is also important to mention that each Fog Node is responsible for keeping track of changes in context, including possible participant changes, creation of new topics, and relevant information that has been received. Each Fog Node should be on its own autonomous and keep a copy of all relevant information locally. This reduces the need for centralized units.

Given the versatility the topics definition provides, it could be feasible for policies and services to define communication topics for collaboration, and even propose evolutionary policies that given topic discussion information could improve the operation of a Fog Node as it collects information about its surrounding context. The definition of such policies is not part of the scope of this research and is as a future research proposition.

One issue of the software architecture proposed in [1] is that there is no software block in the Fog nodes to enable surrounding context-awareness. As such it is important for the protocol to function that such a software entity is created. To that end we propose a new software unit to be added, the Knowledge Diffusion Unit. This unit's main goal will be to enable the Fog Unit to communicate with other neighboring Fog Units and diffuse knowledge amongst them. In addition, it is responsible for storing this information in a proper manner, either in the Fog Node's internal storage, or on the Shared Memory unit, for the Fog Node's Decision Making Unit to use. This unit will be the one using ECTORAS protocol in the overall System Architecture.

## IV. Future Work and Conclusion

ECTORAS protocol is still being researched and worked on. We currently rely on the Registry Unit proposed in [1] for storage of topics and keeping updated information on topics. One future endeavor on ECTORAS protocol would be the substitution of the Registry Unit by a Super-Topic that all Fog Nodes are required to subscribe to at start time. This would be a broadcasting bus on which topic updates, and service updates can be made public, and the registry unit can then become part of the Fog Nodes.

For this extension to be viable, coordination and cooperation among the Fog Nodes is vital. While the broadcast bus is there to announce certain events, and for Fog Nodes to publicly request information on topics, it should not be over crowded. In that sense the bus should be only used to declare new Topics and their participants, and new services and Fog Node that provides the new services. Each Fog Node then needs to contact each of the participants of a certain topic in order to announce their request to subscribe the topic, and for them to have an updated participants' list. After tackling the lack of the registry unit by substituting its need by a universal broadcasting bus, we can say that the system is truly distributed, and not bound by a centralized entity, which would be the Registry Unit in this example.

## References


[1] B. Nikolopoulos, A. C. Dimopoulos, M. Nikolaidou, G. Dimitrakopoulos and D. Anagnostopoulos, "The role of autonomous aggregators in IoT multi-core systems," in *Proceedings of the Seventh International Conference on the Internet of Things*, Linz, 2017.

[2] L. Atzori, A. Iera and G. Morabito, "The internet of things: A survey," *Computer networks,* vol. 54, no. 15, pp. 2787-2805, 2010.

[3] OpenFog Consortium, "OpenFog Reference Architecture," OpenFog Consortium, 2017.

[4] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet of Things Journal,* vol. 3, no. 6, pp. 854 - 864, 2016.

[5] Y. Ai, M. Peng and K. Zhang, "Edge computing technologies for Internet of Things: a primer," *Digital Communications and Networks,* vol. 4, no. 2, pp. 77 - 86, 2018.

[6] Q. M. Ashraf and M. H. Habaebi, "Introducing autonomy in internet of things," in *14th International Conference on Applied Computer and Applied Computational Science (ACACOS'15)*, Kuala Lumpur, 2015.

[7] Y. Xia, X. Etchevers, L. Letondeur, T. Coupaye and F. Desprez, "Combining hardware nodes and software components ordering-based heuristics for optimizing the placement of distributed IoT applications in the fog," in *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, Pau, 2018.

[8] J. Dizdarević, F. Carpio, A. Jukan and X. Masip-Bruin, "A Survey of Communication Protocols for Internet of Things and Related Challenges of Fog and Cloud Computing Integration," *ACM Computing Surveys (CSUR),* vol. 51, no. 6, p. 116, 2019.

[9] L. Richardson and S. Ruby, RESTful web services, O'Reilly Media, Inc., 2008.

[10] Z. Shelby, K. Hartke and C. Bormann, "The constrained application protocol (CoAP)," 2014.

[11] C. Bormann, A. P. Castellani and Z. Shelby, "Coap: An application protocol for billions of tiny internet nodes," *IEEE Internet Computing,* no. 2, pp. 62 - 67, 2012.

[12] S. Vinoski, "Advanced message queuing protocol," *IEEE Internet Computing,* no. 6, pp. 87 - 89, 2006.

[13] P. Saint-Andre, "Extensible messaging and presence protocol (XMPP): Core," 2011.

[14] U. Hunkeler, H. L. Truong and A. Stanford-Clark, "MQTT-S—A publish/subscribe protocol for Wireless Sensor Networks," in *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08)*, 2008.

[15] G. Pardo-Castellote, "Omg data-distribution service: Architectural overview," in *23rd International Conference on Distributed Computing Systems Workshops, 2003. Proceedings.*, 2003.

[16] G. Pardo-Castellote, "Omg data distribution service: Real-time publish/subscribe becomes a standard," *RTC Magazine,* no. 14, 2005.