A Study for a Name-based Coordination of Autonomic IoT Functions

Godwin Asaamoning, Paulo Mendes
COPELABS, University Lusofona
{godwin.asaamoning, paulo.mendes}@ulusofona.pt

***Abstract*—The Internet-of-Things represent a future trend of the Internet, in which massive numbers of networked objects are assumed to have the capability of sharing data collected through sensing activities. It is envisioned that such objects are not dedicated hardware modules embedded specific machines communicating via proprietary wired or wireless networks to dedicated software applications. On contrary, Internet-of-Things may have a more autonomic deployment, in which objects are autonomous and communicate via existing IP based networks. Due to its distributed nature, there is the need to rely on autonomic networking to improve the efficiency while reducing the operational costs of Internet-of-Things frameworks. This paper aims to provide a study about the best data plane to support Internet-of-Things, and an analysis about the best model to support its distributed autonomic behavior.**

***Index Terms*—Internet-of-Things; Edge Computing; Autonomic Networking; Named Data Networking**

I. Introduction

Internet-of-Things (IoT) is a current trend in which massive numbers of networked objects are assumed to have the capability of sharing data collected through sensing activities. The IoT paradigm provides the required support for the creation of cyber-physical systems, which may be used to improve the daily life experience of citizens as well as to bring social and economic benefits.

In order to ensure the large-scale deployment of IoT systems, there is the need to ensure that applications designed in different vertical domains are able to access any kind of remote sensing devices: this is where Machine-to-Machine (M2M) [1] technologies can play a role supporting the emergence of IoT.

M2M supports cellular connectivity between machines without or little human intervention for data transfer. However, M2M technology is being mostly applied to industrial scenarios, or close networks around customers and suppliers using a subscription based charging model. Hence, to deploy IoT in a large set of vertical markets, there is the need to design a distributed IoT framework able of supporting not only machine-to-machine communications but also machine-to-human and vice-versa.

In what concerns the distribution of IoT computational functions, many of today's networking services are based on cloud computing. However, IoT manufacturers and developers are analyzing the potential benefits of doing more computation and analytics on the devices themselves or at the edge of the network. This edge networking approach helps reduce latency for critical IoT applications, lower dependence on the cloud, and better manage massive amount of data, aiming to support a wide set of IoT services, namely those relying on machine learning for tasks such as object detection, face recognition, and obstacle avoidance.

However, distributing IoT computational functions brings a certain level of complexity to align the operation of a large number of edge network elements. The need to reduce the management complexity, and the workload of the human operators in order to lower operational expenditures and at the same time improve network capacity and service quality led to a strong demand for highly automated IoT solutions.

This paper provides an initial study of suitable approaches to deploy autonomic IoT services based on the distribution of computational functions among a set of edge devices. This paper is organized as follows: section II provides a study about the most suitable solution for data transmission in a distributed IoT framework, based on a comparison of M2M and information-centric networking approaches. In section III it is provided an analysis of current requirements for an autonomic IoT behavior and of the most suitable autonomic networking reference model. Section IV describes two alternative paradigms to devise an autonomic IoT. The final section summarizes the presented analysis and enumerates open issues related to the coordination of autonomic functions.

II. IoT Data Communication Framework

As a first step to identify the most suitable approach for the large scale deployment of IoT, we start by analyzing the most suitable data communication infrastructure. This section provides detailed information about M2M communications and its limitations to support a IoT framework able to support several vertical markets, and points out to the advantages of designing a data-centric IoT networking approach.

Shifting from traditional host-based Internet operation to a data-centric approaches suites the deployment of a pervasive IoT by promising embedded security, seamless mobility, and scalability.

A. M2M as an Integral Part of IoT

M2M communications are triggered by hardware modules embedded on specific machines able of cap-

turing events (e.g. sensing activity), and transmitting the collected data to software applications that are able to convert it into useful information.

M2M communications have minimal human intervention, are short lived (few small packets per connection), and have high energy consumption constraints. M2M communications are therefore very different from human triggered communications, which normally are long lived, with a large amount of data exchanged in each connection, and without energy consumption constraints. Although M2M machines (with sensors, RFID badges, and tracking tags) typically use short-range radio technology like Zigbee, Bluetooth, and WiFi, network operators have mostly focused on cellular networks for M2M applications, assuming that most M2M applications will not require particularly high-data rates. Operators are relying on Long-term evolution (LTE) and LTE-advanced (LTE-A) to be able to support IoT operations through a flexible communication architecture designed to enable communication at a lower cost per bit and to accommodate continuous growth in number of connections. The security and authentication baked into the LTE standard will be particularly important for medical monitoring applications, for instance.

The IoT space is however very fragmented and diversified, meaning that it makes sense to think about the development of an easier and cheaper way to ensure a large-scale deployment of IoT to any market, allowing vendors to use IoT applications as they wish. When we consider such large-scale scenarios, M2M technology start to show its limitations. M2M applications are typically composed of hardware modules embedded in machines at a customer site that communicate via proprietary cellular or wired networks to a dedicated software application, often at the supplier's network, based on a point-to-point communication model. The success of IoT passes by the support of different vertical markets, which may require the same dedicated devices/assets/machines as M2M applications, but also low-power and passive sensors as well as inexpensive devices that should communicate over different types of access (e.g. wired, 802.15.4, 802.11, 802.11p, UWB, cellular) and transport networks (e.g. TCP/IP, ZigBee). In this aspect, M2M technology such as the Message Queuing Telemetry Transport (MQTT) is a common solution for TCP/IP based solutions, but requires the deployment of a variation of the main protocol, called MQTT-SN, to operate on ZigBee networks. This lack of universal communication limits the support that such technology can give to a large-scale deployment of IoT. Moreover the fact that MQTT relies on message broker brings also scalability limitations.

Moreover, a successful deployment of IoT in several vertical markets should allow not only machine-to-machine communications but also machine-to-human and vice-versa. Although M2M solutions offer remote access to machine data, these data is traditionally targeted at specific solutions in service management applications. On contrary, integration of devices and sensor data with big data, analytic and other enterprise applications is a core concept behind the emerging pervasive IoT. Based on this analysis, we can clearly say that although M2M in an integral part of IoT, it does not seem a suitable solution to support a successful large-scale deployment of IoT in different vertical markets.

B. Data-centric Networking as a Driver of IoT

Within the current hourglass architecture of the Internet, the Internet protocol (IP) is the universal layer for global interconnection. IP was designed for establishing communication between end points by transmitting packets. The unresolved problems of IP based networks such as mobility, security, and awareness of users' expectations are major inspirations for the development of the information-centric networking paradigm. In the specific domain of IoT, the has been a large effort in modifying the TCP/IP protocol stack to fit IoT deployment scenarios. These efforts have resulted in extensions to existing protocols in the TCP/IP protocol suite as well as development of multiple new protocols. Hence, there are evidences that existing IP-based solutions are inefficient to support IoT applications, and that a more effective solution would embrace the Information Centric Network architecture[2].

Information-centric networking evolves the Internet infrastructure away from a host-centric paradigm, to a network architecture based on naming information. With an information-centric networking paradigm, connectivity of a more pervasive IoT may well be intermittent, security may become more flexible by relying on packets and not on channels, and mobility, multi-access, anycast, and multicast are natively supported.

The Named Data Networking (NDN) [3] project instantiates the information-centric networking paradigm by generalizing the role of this protocol stack thin waist, such that packets can name objects other than communication endpoints: networking semantics are shifted from delivering the packet to a given destination address to fetching data identified by a given name. The name in an NDN packet can name anything, from an endpoint, to a command to extract data or control embedded devices (e.g. city lights).

Communication in NDN is driven by data consumers, through the exchange of two types of packets: Interest and Data. Both types of packets carry a name that identifies the requested data. A consumer places the name of the requested data into an Interest packet and sends it to the network. Routers use this name to forward the Interest toward a data producer or a data copy. Once the Interest reaches a node that has the requested data, the node returns a Data packet that contains both the name and the content, together with a signature by the producer's key which binds the two. The NDN design assumes hierarchically structured names, which allows

applications to represent the context and relationships of data elements.

The applications of IoT often imply the usage of information-centric networking, where users or devices consume data and information from the network instead of communicating with specific hosts. In this context, NDN seems to be suitable to support data sharing among pervasively deployed networked objects[4]. However, generating low data rate information, due for instance to monitoring and measurement operations, requires further investigation about applying in-network caching to IoT traffic flows. The main challenge refers to the transient nature of IoT data. New research findings show that IoT applications can also benefit from new caching techniques aware of the small side and potential high frequency of transient data: for instance, Serdar Vural et al, [5] demonstrate that caching transient data is a promising information-centric networking technique that can reduce the distance between content requesters and the location in the network where the content is fetched from. This research work demonstrates the feasibility and benefit of using Internet routers to cache transient data generated by IoT applications.

However, while the IoT shares similar properties of other information centric services, pervasive IoT encompasses potentially intermittently connected devices, high rate of new data being generated, and heterogeneous requirements from applications. This means that although NDN is suitable to support data transmission in pervasive IoT, identifying the right approach requires further research (e.g. naming and routing). For instance, in order to be truly pervasive, IoT should be supported by a networking system that allows the exchange of data based on any communication opportunity, independently of the intermittent presence of 3G coverage or open WiFi networks, while being agnostic of the location of devices. This will require working around the forwarding strategies of NDN to support opportunistic routing, which may be based on social-aware approaches, assuming that most of the mobile sensing devices will be associated to people [6], [7], [8].

The naming design of sensing data may also follows an hierarchical syntax structure of NDN. Nevertheless, one major difference towards the name used by content retrieval applications, is that the syntax used by sensing applications should include context information. For instance, the inclusion of geolocation information may be needed since the usefulness of sensing data has a space constraint. This means that the hierarchical naming structure of sensing applications may follow a syntax such as */sensing-app/geolocation/timestamp/tag1/tagn* [9] where: sensing-app identifies the type of application; geolocation allows data to be mapped to a specific geographical area; timestamp allows data to be mapped to a specified time period; tagging indicates the meaning of the sensing data itself.

III. Control of Edge IoT Functions

Although a more pervasive IoT shows up as an emerging technology able of supporting different vertical markets, there are several challenges that need to be tackled before deploying it, such as the distribution of IoT computational functions, which nowadays relies mostly on cloud computing frameworks. However relying only on cloud computing it starting to raise scalability and performance problems, facing an increasing number of sensors and devices that are continuously producing data and exchanging it via complex networks aiming to monitor and control critical infrastructures and applications.

As a strategy to mitigate the problems related to resource congestion and long delays, edge computing has emerged as a new paradigm to solve the mentioned resource congestion and long delays issues. The distribution of IoT computational functions among a set of edge devices may balance network traffic, reduce transmission latency between edge/cloudlet servers and end users, and reduce response times for real-time IoT applications in comparison with traditional cloud services.

However, distributing IoT computational functions brings a certain level of complexity to align the operation of a large number of edge network elements. The need to reduce the management complexity aiming to lower operational expenditures and improve network capacity and service quality led to a strong demand for highly automated IoT solutions. This section starts by supporting this argument, analyzing the operational and architectural requirements for the deployment of autonomic IoT functions. Such requirements are then used to analyze the most suitable reference model for the control of autonomic IoT functions.

A. Requirements for Autonomic IoT Functions

The ultimate goal of autonomic networking is to replace human and automated operations by autonomic functions, so that networks can run independently without depending on a human or network management system for routine.

1) From the network operator point of view: Network operators are currently facing numerous challenges (business and technical) related to market deregulation, converged of network services (e.g. virtualization, clouds), quality-of-service (QoS), and reduction of operational expenses: the configuration, optimization, and healing processes of large-scale mobile networks is very complex, and the capital expenditure (CAPEX), as well as the operational expenditures (OPEX) are very high.

The management of such dynamic networking environment is possible based on the interaction of different autonomic functions able of coordinating their activities to provide augmented network performance based on information collected in different parts of the network [10], [11]. Each autonomic function locally communicates with neighboring autonomic functions to reach global efficiency by improving its own efficiency, while

reducing OPEX and human intervention. To ensure minimal human intervention, the operation of pervasive IoT networks must exhibit self-organization functionalities (self-configuration, self-optimization, self-healing). Self-organization is a process where some form of overall coordination arises out of the local interactions between simple autonomic functions of an initially disordered system [12].

2) From an architectural point of view: The development of a framework for pervasive IoT needs to take into account architectural challenges related to security, efficiency, robustness, scalability, and reliability. In this networking context, the automation of network functions is required to achieve systems efficiency while dealing with potential disturbances (e.g. conflicts among autonomic functions).

In the context of telecommunications, The European Telecommunications Standards Institution (ETSI) provides standards to provide automate the operation of networks, including M2M, while reducing the burden on network provider/operator. Such standards include troubleshooting and recovery mechanisms; discovery and configuration of objects; and information forwarding management.

In what concerns the Internet, several protocols already exhibits many aspects of self-configuration (e.g. IP Address Management and DNS) and self-optimization (e.g. routing). However, many network operations are still heavily dependent on human intervention, or on centralized top-down network management systems [13]: operations such as network management, security setup and forecasting are the targets of autonomic networking technologies.

B. Reference Models for Autonomic Networking

This subsection analyzes the reference models being developed by Internet Engineering Task Force (IETF) and ETSI, in terms of their advantages and limitations to support pervasive IoT for control and management.

1) ANIMA reference model: The ANIMA reference model for autonomic networking, being standardized in the IETF, provides a high level architectural view of the network [14]. An autonomic network consists of a number of autonomic nodes, which interact directly with each other. Such nodes provide a common set of capabilities across the network, called the Autonomic Networking Infrastructure (ANI). The ANI provides functions like naming, addressing, negotiation, synchronization, discovery and messaging. Autonomic functions may span several nodes in the network, implementing one atomic entity in each node. These atomic entities are called Autonomic Service Agents (ASA) [15].

Autonomic nodes must communicate with each other, for example to synchronize a set of network parameters, that were previously negotiated. In this context ANIMA is standardizing a signaling protocol to be used between autonomic agents, called Generic Autonomic Signaling Protocol (GRASP) [16], which may be be used for discovery, negotiation and synchronization. The GRASP protocol presents various unresolved design questions, which need to be tackle in the context of a pervasive IoT, namely: every participant needs to have an NTP-synchronized clock; DoS Attack Protection is not handled; the usage of UDP/TCP needs clarification; current built-in security mechanism requires expensive asymmetric cryptographic calculations for every message.

2) ETSI reference model: For designing a restful architecture of heterogeneous M2M communications, ETSI released a set of specifications that enables seamless service provisioning [17]. The ETSI M2M standard (oneM2M follows similar design) is an illustration of connectivity with a functional architecture that consists of a set of generic capabilities, in which each machine represents an instance of a service capability layer (SCL). The ETSI M2M is an abstract model that consists in the definition of a SCL for devices (DSCL), gateways (GSCL), and networks (NSCL) since each of them has a corresponding entity in the service layer.

As of now the ETSI M2M architecture is a centralized approach: The network SCL is responsible for mutual authentications, data exchange between massive number of machines, resource discovery and subscription. In other words, the current ETSI M2M architecture suffers from scalability and fault tolerance. Since it has to support millions of machines, the NSCL has to take care of subscriptions and data exchange between a large set of end points. Moreover, the implementation of NSCL in a distributed way is not enough to mitigate the scalability and reliability limitations, because of high costs and scaling issues. Due to these reasons, recent studies [18] aim to provide extensions for M2M standards aiming to provide scalable and distributed SCL layers. However such proposals present several limitations to handle distributed caching and service discovery.

3) Comparison of reference models: The reference model for autonomic networking being developed in the IETF ANIMA working group is fully distributed, being designed based on the current Internet architecture. This means that it could be easily adapt to control pervasive IoT where data transmission is done based on the NDN framework. The ETSI reference model is not distributed in nature and is being designed towards a more closed network architecture (e.g., pay for use with subscription).

ANIMA supports direct communication between autonomic functions through a generic signaling protocol, while when it comes to ETSI there is no standard signaling and control protocol. In ANIMA there is a coordination reference model that supports discovery, negotiation, and synchronization, while in ETSI there is a need to work out on coordination model scenarios. Some operation parameters, such as energy efficiency, scalability, and reliability, have high priority in ANIMA but not in ETSI.

Based on the general characteristics of both models, the ANIMA reference model seems more suitable for the control of autonomic functions aiming to support the deployment of large-scale IoT solutions for different vertical markets.

IV. Framework for an Autonomic IoT

The study described on sections II and III shows that a promising solution to support a successful deployment of IoT in different vertical markets, passes by using the ANIMA reference models to control autonomic IoT functions deployed at the edges of an NDN network.

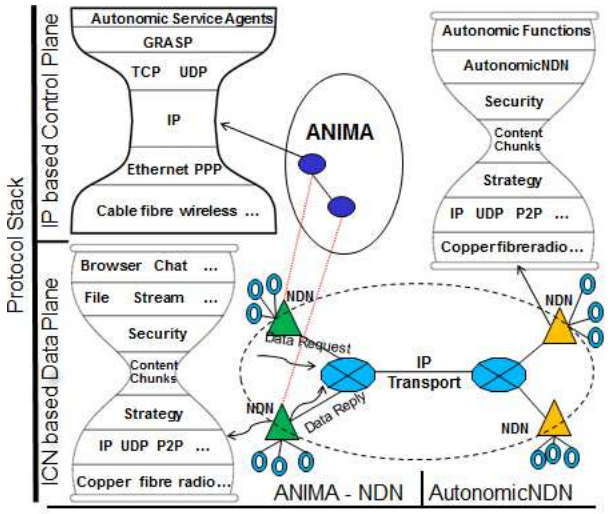

Figure 1. Approaches to control autonomic IoT functions at network edges

In this section we discuss two alternative approaches for the control of autonomic IoT functions in such networking scenario, illustrated in Figure 1: an hybrid ANIMA-NDN approach (left side of the figure) where the Generic Autonomic Signaling Protocol (GRASP) is used on top of a NDN network, interacting with NDN nodes at the edge of a network; a full information-centric approach based only on NDN, AutonomicNDN (right side of the figure), that relies on Chronosync to support the basic operation of autonomic functions.

A. Hybrid ANIMA-NDN

As shown in figure 1 each NDN node also operates as an autonomic node, as described by ANIMA, implementing several autonomic functions and ASAs, c.f. section III-B1. Therefore, every node implements two protocol stacks: an NDN stack to support data transfer, and an IP stack used by GRASP. Although it will cause problems with large messages, we assume that GRASP uses UDP in a pervasive IoT due to the presence of small packets.

Autonomic nodes use GRASP for direct communication in a secured environment. Hence, a trust infrastructure such as a PKI infrastructure must be in place based on a trust anchor for each domain [14], which may bring scalability challenges in a pervasive IoT scenario.

In this hybrid approach the signaling among autonomic functions is done through the three stages supported by GRASP: discovery, negotiation, and synchronization. The discovery process is a two hand-shake protocol that is trigger by having an initiator sending a message (with a discover objective) via UDP to a link-local multicast address that includes all GRASP neighbors. Every node that supports GRASP always listens to a well-known UDP port to capture the discovery messages. If no discovery response is received within a reasonable timeout, the initiator may send another discovery message, after an exponential back-off. After a node successfully discovers a set of peers, it must cache this information, which will be used by the negotiation and synchronization procedures.

This request-response hand-shake is also used in the subsequent two phases for the negotiation of the network parameters that will be monitored among the discovered peers, and after that, for the synchronization of the state of such parameters among the selected peers. During the negotiation phase, a bidirectional procedure may be used to reach a compromise between two ASAs, if the receiver replies to the initiator with a proposed alternative configuration (e.g. a configuration that uses fewer resources). The negotiation procedure is ended when one of the negotiation peers sends a negotiation Ending message.

In terms of synchronization, the message exchange is unicast and concerns only one synchronization objective. This to say that to handle large groups of nodes in a pervasive IoT scenario, synchronization flooding may be an issue.

In a first analysis, it is clear that a hybrid ANIMA-NDN approach may not scale in a pervasive IoT environment, since for each new peer added to the network the GRASP signaling takes n^3 transactions where n is the number of transactions between peers for synchronization of data. Moreover, robustness problems raise, since the system relies on a single certification authority. Protocol wise, there is a clear signaling overhead inherent to the implementation of the discovery, negotiation and synchronization phases by GRASP and the usage of two protocol stacks. A clear solution passes by merging the three signaling stages, as suggested by the GRASP rapid mode, as well as avoiding the implementation of two protocol stacks.

B. AutonomicNDN

The AutonomicNDN framework is a proposal to mitigate the limitations of the ANIMA-NDN hybrid approach. With the AutonomicNDN framework, the signaling for the control of autonomic functions is done by using the same packets used to exchange data in an NDN network.

This section describes how the control of autonomic functions on a more pervasive IoT can be achieved by using only NDN. The major goal is to follow the sugges-

tion provided by the GRASP rapid mode and simplify the three phase signaling process while avoiding the robustness issues of the hybrid ANIMA-NDN approach. The latter is done in NDN by requiring data producers to cryptographically sign every data packet.

Since NDN is host agnostic there is no need to discover neighbor nodes with whom to synchronize network parameters. The selection of nodes to participate in the coordination of an agreed set of network parameters is done implicitly by the used naming scheme.

The proposed AutonomicNDN approach is designed based on the ChronoSync library of NDN [19]. AutonomicNDN should be seen as a general-purpose autonomic signaling service, where network nodes share information about different autonomic functions and their parameters. Each shared function is composed of a set of parameters, which were locally created or previously discovered.

An AutonomicNDN domain is the basic security entity. Each domain owns a specific namespace and a public/private key pair, which has been previously certified via an appropriate trust model. For example, if a traffic-domain owns /lisbon/traffic namespace, and the city of Lisbon authorizes this ownership, then the traffic-domain (can be a neighbor) can use the key to authorize local nodes (e.g. parking spots, buses) to share certain autonomic functions by issuing node-function specific certificates. This allows each domain to participate in the coordination of multiple different autonomic functions. Nodes are holders of shared autonomic functions authorized by the AutonomicNDN domain. The current design of AutonomicNDN assumes that each node can be mobile, having only intermittent Internet connectivity, getting disconnected any time, or being completely shut down at some point.

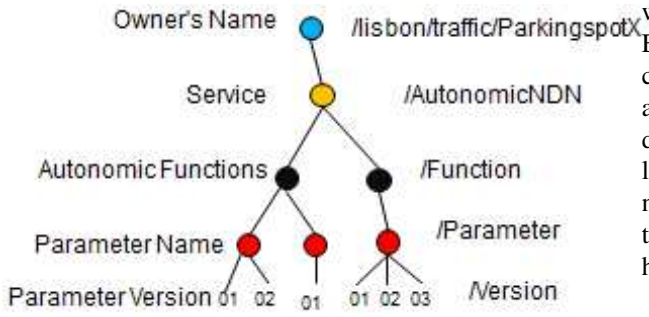

Figure 2. Structure of AutonomicNDN names

The NDN name of an autonomic parameter consists of five components as illustrated in Figure 2: owner's prefix, service name, function name, parameter name, parameter version. The prefix guides NDN Interest packets towards the specified node (e.g. /lisbon/traffic/parkingspotX). The next component identify the service (AutonomicNDN) on the node, with a role similar to port number in IP. The remaining fields are used internally by the service to identify a specific autonomic function, the required parameter and its version.

Based on the proposed naming scheme, AutonomicNDN defines names used to synchronize autonomic functions among a set of nodes belonging to the same domain avoiding the discovery and negotiation phases on GRASP. Such sync names consist of owner's prefix, service name, function name, parameter name and action type (e.g. /lisbon/traffic/parkingspotX/autonomicAN/space-mng/update). Based on sync names each node can show their interest in synchronizing data related to a set of autonomic functions under some namespace. Based on the NDN hierarchical naming scheme, the synchronization may target all autonomic functions, all parameter of a specific function, or only a specific parameter under a certain domain (e.g. /lisbon/traffic or /lisbon/traffic/parkingspotX). With the usage of sync names autonomicNDN does not need to implement an explicit negotiation phase as in GRASP.

In order to secure the sharing of information about autonomic functions, AutonomicNDN relies on several aspects of the NDN framework. First of all, unauthorized parties cannot jeopardize communications, since all NDN packets are uniquely named and signed. Moreover, NDN provides per-packet state and two-way symmetric Interest/Data flows to effectively mitigate interest flooding attacks. Based on the NDN framework some autonomic functions can also be made private by using shared key or group based encryption. In this case, the owner can authorize new nodes to join in and share the private autonomic functions.

V. Summary

To deploy IoT in a large set of vertical markets, there is the need to design a distributed framework, which currently is mostly supported by cloud computing. Edge networking may help to mitigate some problems of cloud computing solutions, by balancing network traffic, and reducing transmission latency for instance. However, distributing IoT computational functions brings a certain level of complexity to align the operation of a large number of edge network elements. The need to reduce the management complexity led to a strong demand for highly automated IoT solutions.

In an attempt to identify a suitable framework to control autonomic IoT functions two alternative approaches were analyzed, based on the requirement that data communication should be supported by the Named-Data Networking framework. This requirements comes from the study that show the evidences that existing IP-based solutions are inefficient to support IoT applications.

How study shows that a hybrid ANIMA-NDN approach may not scale in a pervasive IoT environment, due to the scalability and communication overhead problems. Hence a solution is proposed, called AutonomicNDN, which passes by merging the three signaling

stages, as suggested by the GRASP rapid mode, as well as avoiding the implementation of two protocol stacks. The AutonomicNDN approach is based on the ChronoSync and a set of names used to synchronize autonomic functions among a set of nodes belonging to the same domain independently of their topological location and without the need for negotiation of autonomic functions (avoiding the discovery and negotiation phases of GRASP).

REFERENCES

[1] G. Wu, S. Talwar, K. Johnsson, N. Himayat, and K. D. Johnson, "M2m: From mobile to embedded internet," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 36–43, 2011.

[2] W. Shang, Y. Yu, R. Droms, and L. Zhang, "Challenges in iot networking via tcp/ip architecture," NDN, Technical Report NDN-0038, Tech. Rep., February 2016.

[3] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, kc claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named data networking," *ACM SIGCOMM Computer Communication Review*, 2014.

[4] D. Saxena, V. Raychoudhury, and N. SriMahathi, "Smarthealth-ndnot: Named data network of things for healthcare services," in *Proc of ACM Workshop on Pervasive Wireless Healthcare*. Hangzhou, China, June, 2015, pp. 45–50.

[5] S. Vural, N. Wang, P. Navaratnam, and R. Tafazolli, "Caching transient data in internet content routers," *IEEE/ACM Transactions on Networking*, vol. 25, April 2017.

[6] W. Moreira and P. Mendes, "Pervasive data sharing as an enabler for mobile citizen sensing systems," *IEEE Communications Magazine*, vol. 53, no. 10, pp. 164–170, 2015.

[7] S. Dynerowicz and P. Mendes, "Named-data networking in opportunistic networks," in *Proc. of ACM ICN*. Berlin, Germany, September 2017.

[8] P. Mendes, R. Sofia, V. Tsaoussidis, S. Diamantopoulos, and C. Sarros, "Information-centric routing for opportunistic wireless networks," IETF Internet Draft - draft-mendes-icnrg-dabber-01, Tech. Rep., August 2018.

[9] P. Mendes, "Combining data naming and context awareness for pervasive networks," *Elsevier Journal of Network and Computer Applications*, vol. 50, pp. 114–125, 2015.

[10] P. Mendes, C. Prehofer, and Q. Wei, "Context management with programmable mobile networks," in *Proc of IEEE Computer Communications Workshop*, Oct, 2003 Dana point, CA, USA, pp. 217–223.

[11] Q. Wei, K. Farkas, C. Prehofer, P. Mendes, and B. Plattner, "Context-aware handover using active network technology," *Elsevier Computer Networks*, vol. 50, no. 15, pp. 2855–2872, 2006.

[12] C. Prehofer and C. Bettstetter, "Self-organization in communication networks: principles and design paradigms," *Communications Magazine, IEEE*, vol. 43, no. 7, pp. 78–85, 2005.

[13] S. Jiang, B. Carpenter, and M. Behringer, "General gap analysis for autonomic networking," *Internet RFC 7576*, June, 2015.

[14] M. Behringer, B. Carpenter, T. Eckert, L. Ciavaglia, B. Liu, J. Nobre, and J. Strassner, "A reference model for autonomic networking," *IETF Internet Draft - draft-ietf-anima-reference-model-07*, August, 2018.

[15] M. Behringer, M. Pritikin, S. Bjarnason, A. Clemm, B. Carpenter, S. Jiang, and L. Ciavaglia, "Autonomic networking: Definitions and design goals," *Internet, RFC7575*, June, 2015.

[16] C. Bormann, B. Carpenter, and B. Liu, "A generic autonomic signaling protocol (grasp)," *IETF Internet Draft - draft-ietf-anima-grasp-15*, July 2017.

[17] D. Boswarthick, O. Elloumi, and O. Hersent, *M2m communications: a systems approach*. John Wiley & Sons, 2012.

[18] L. A. Grieco, M. Ben Alaya, T. Monteil, and K. Drira, "Architecting information centric etsi-m2m systems," in *Proc of IEEE PERCOM Workshops on*. Budapest, Hungary, March, 2014, pp. 211–214.

[19] Z. Zhu and A. Afanasyev, "Let's chronosync: Decentralized dataset state synchronization in named data networking," in *Proc of IEEE International Conference on Network Protocols*. Goettingen, Germany, Oct, 2013, pp. 1–10.