IoT Architecture Based on RINA

Maryan Rizinski
Department of Computer Science
Metropolitan College, Boston University
Massachusetts, USA
rizinski@bu.edu

John Day
Department of Computer Science
Metropolitan College, Boston University
Massachusetts, USA
day@bu.edu

Lou Chitkushev
Department of Computer Science
Metropolitan College, Boston University
Massachusetts, USA
ltc@bu.edu

***Abstract*—Interoperability is one of the biggest challenges facing the Internet of Things (IoT). While the emergence of many IoT protocols promises Internet connectivity to a large number of devices, it also leads to an inevitable fragmentation that hinders the IoT adoption. The existing literature focuses on creating device abstractions to deal with the multiplicity of protocols, but little work has been done to eliminate the root cause of fragmentation – the number of protocols itself. This paper proposes a novel approach to simplify IoT by design based on the Recursive InterNetwork Architecture (RINA) and explores RINA's benefits such as reducing protocol complexity, improving standardization, and enhancing security.**

Keywords—Internet of Things (IoT), Recursive InterNetwork Architecture (RINA), Network Architecture, Future Internet.

I. Introduction

One of the top trends impacting the Internet of Things (IoT) is that the number of things, or devices, that connect to the Internet grows at a tremendous rate. According to Gartner, there are 14.2 billion connected things in use in 2019 with an estimated increase of up to 25 billion by 2021 [1]. This opens a remarkable potential to create enormous business value and to reshape entire industries. A report by McKinsey predicts that the IoT can generate a global economic value of up to $11.1 trillion a year by 2025 [2].

With that in mind, it becomes crucial to design an IoT architecture that will ensure that things are connected in a seamless and interoperable way. The current architectures face challenges due to the existence of a plethora of connectivity protocols. Realizing the lucrative opportunities that IoT unfolds, various industry players create and promote their own protocols for connecting and managing devices, thus competing for a market share among each other. The consequences on interoperability are evident: a device that supports one protocol can hardly be connected out-of-the-box into an IoT system that supports another protocol without implementing additional customizations to that system. Even a device that supports one protocol and is used in one IoT system can be difficult to be readily used in another IoT system that supports the same protocol due to potential differences in middleware and proliferation of "standards."

Despite the fact that these protocols share commonalities (e.g. switching on or switching off a dimmer is the same conceptual operation irrespective of whether it is performed via ZigBee or Z-Wave), they are different from each other from an applications and networking perspective. Translating between application protocols can be especially challenging as it requires translating both the semantics of the operations and of the object models. As a result, using different technologies to achieve identical use cases can lead to fragmentation and barriers to entry, factors that can negatively affect interoperability and significantly increase the cost, and even cause erroneous (and potentially catastrophic) behavior thus compromising IoT's promise for an unprecedented growth of connected devices.

Several surveys have been published on the current architectures and enabling technologies for the IoT such as [3]-[5]. For example, [3] identifies middleware as a key technology for the development of IoT applications. The purpose of IoT middleware is to hide the underlying details of the low-level communication protocols. This helps reduce the cost and time for development along with having the capability to bridge across different types of devices. Similarly, [4] emphasizes the need for middleware to hide the details of different technologies and to provide standard interfaces to assist application development. In addition, it states that one of the most important roles of IoT is to connect heterogeneous networks using various communication technologies and protocols. Furthermore, [5] sees architecture standardization as a factor that will foster market competition based on creating quality products, but at the same time claims the architecture requires new protocols between heterogeneous devices – this fragments the market, reducing quality.

The existing literature tends to focus on creating device abstractions for dealing with the multiplicity of protocols. However, little work has been done to eliminate the root cause of fragmentation – the number of protocols itself. In this paper, we propose a novel architecture for IoT based on the pioneering work on the Recursive InterNetwork Architecture (RINA) [6]. Using RINA's single recursive layer, our purpose is to model an architecture of what should be the "ideal" case that will provide interoperability by design and that will inherently reduce protocol complexity. An IoT architecture with RINA is based on a single layer that makes a clear delineation between transmission mechanisms and transmission policies. While transmission mechanisms are fixed, transmission policies can be adjusted.

The rest of this paper is organized as follows. Section II discusses the challenges of IoT in terms of interoperability, fragmentation, increased cost for implementation, as well as security issues. Section III proposes a novel architectural approach to IoT that is based on RINA's single recursive layer. Section IV discusses the benefits of using RINA in IoT. Finally, in Section V we provide a conclusion.

II. Challenges in IoT

While IoT promises to create an enormous business value in near future, it does not come without challenges. This section discusses the main issues faced by IoT on its way to realize its full potential.

978-1-7281-5127-4/20/$31.00 ©2020 IEEE

A. Interoperability

As explained in [3], the first challenge in IoT is to build a middleware that is capable of supporting all kinds of devices and IoT applications. However, creating such a unified middleware that will be used across the industry has proven to be an impossible task so far due to a combination of factors such as market interests and lack of standardization. For example, industry players tend to create their own IoT platforms with their own technology stacks and abstractions. While the underlying protocols are the same, the implementation details can be different. This not only creates interoperability problems but also limits the choice in making comparisons and deciding which platform to use.

B. Fragmentation

The primary source of fragmentation in IoT is the use of different data transfer protocols. While the rationale for them is the special requirements of the media and IoT, research has shown that this can be accommodated by simply changing policies [7]. Thus, its primary purpose becomes creating barriers to entry and locking in customers. Standards for these protocols merely serve to entrench this effect. One mitigating factor is that translation between data transfer protocols is not necessary in the architecture, since all translation occurs at the application layer. However, it does increase the "parts-count" and increases the necessary inventory of devices and the complexity of management. Fragmentation using different application protocols contributes even more to creating barriers and locking in customers as well. This creates a far more subtle and difficult translation problem. Translating between application protocols requires translating the semantics of the operations and of the object models. It can be very difficult to capture (and preserve) all of the nuances of the semantics, which are never well-specified. Here a standard application protocol can have a major effect by eliminating the need for translation, which would also eliminate unforeseen effects when the translation failed to capture some nuance of the semantics.

These subtle translation problems would make the efforts toward achieving interoperability using a common IoT application layer middleware considerably more complex and more difficult to achieve. It may not be possible to emulate the necessary semantics to the users of the middleware. Common broadly accepted standards for the operators of the application protocol and the object models, at least within industries, is crucial. Not only must the syntax be translated but also the semantics of the commands and the quantities being measured. This can be very subtle and error-prone owing to developers exploiting undocumented and often unintended properties of the implementation. The appropriate (and generally accepted) strategy is to keep the operations as simple as possible, and to put the focus on the object models alone, i.e. the real application. There is a trend among industry players to create and promote their own protocols with the intention of establishing a market presence. This leads to an even greater fragmentation. Not only the resulting IoT platforms lack homogeneity but also the innovation is impeded. Instead of having a single IoT architecture on top of which industry players will compete by building high quality services, the industry is more oriented toward barriers to entry.

C. Cost

From the perspective of an IoT platform provider, connecting new device types that speak different protocols takes implementation time and efforts. This is true for standard protocols and even more for proprietary ones. In order to integrate a new IoT protocol, one would have to deal with two major steps. First of all, it is needed to implement the low-level communication details of the protocol and, secondly, the devices have to be represented in a unified way. The unified representation should be common for different types of protocols. It is typically implemented in the form of a device abstraction that extract the main protocol functionalities so that devices can be managed in the same way irrespective of the underlying protocols they support. All of these activities come at a certain cost for development.

Another aspect of the cost is to ensure support for the newer versions of the underlying protocols. What is more, there are numerous devices by different manufacturers that may introduce behavior that is not fully compliant with the protocol specifications. In addition, if a protocol becomes popular temporarily, but eventually its use does not become widespread in the long term, the efforts for implementing its support into the IoT platform may become completely wasted. At the same time, the IoT platform provider may still be tempted to provide support for it – even if only a relatively small number of companies are using it – expecting that it will eventually become popular again.

From the perspective of an IoT solution provider that builds solutions on top of an IoT platform, the cost is associated with increased complexity and integration effort. Developing a solution to be used in a production system involves integrating various components with different technologies. All of this comes in addition to having to develop the solution itself. In terms of development, even a common middleware does not guarantee entirely smooth operations. For example, if an IoT platform provider upgrades its device abstraction functionality to a new version, the associated changes may lead to substantial efforts on the part of the IoT solution provider to upgrade its service or application to support that new version.

D. Security

One of the most important conclusions made in [3] is that the existing approaches for supporting security, privacy and trust in IoT middleware systems tend to adapt available solutions for existing middleware only as an after-thought. It further states that in order to satisfy the needs of ever-growing IoT applications, IoT middleware needs to be secured by design rather than patching the consequences.

A relevant aspect that would have been worth discussing in [3] is that many of the architectural and security problems about IoT, and the Internet as a whole, are actually related to the TCP/IP model. As known, the protocols of the TCP/IP suite were developed without security considerations in mind. Therefore, securing the TCP/IP itself has been largely an after-thought rather than providing solutions from within the TCP/IP architecture. IoT naturally inherits those issues as the current IoT architectures are based on TCP/IP.

Another important observation in [3] is that it is difficult to achieve end-to-end security in the IoT environment. As information travels through different hops in a network, protocol translations at intermediary nodes might not work with the existing security solutions for two reasons: first, the information relevant for translation is sent in encrypted form and might not be available to the nodes, and second, the changes made at the gateways might invalidate the end-to-end data integrity protection [3].

III. RINA-Based Approach to IoT

This section presents the fundamentals of RINA, gives an overview of the main RINA elements, and proposes a novel RINA-based approach for designing the architecture of IoT.

A. *RINA: A Different Kind of Networking*

RINA is proposed in the pioneering work [6] as a network architecture that is based on the principle that networking is inter-process communication (IPC). As such, RINA takes an alternative approach to networking compared to the mainstream TCP/IP model. Instead of assuming a different set of dedicated layered functions as is the case of TCP/IP, it builds on the premise that there is a single layer responsible for IPC that recurses over different scopes. Unlike TCP/IP, the number of layers in RINA is not fixed, and layers can be stacked on top of each other depending on the application or networking needs. The traditional network architecture attempts in one layer to effectively provide and manage traffic over multiple orders of magnitude. This is an unreasonable expectation. RINA uses layers of repeating functionality configured to provide and manage a given range of operation (e.g. capacity, QoS and scale). By using divide and conquer the great range is more effectively addressed.

B. *Main Concepts in RINA*

As a prerequisite for designing an IoT architecture based on RINA, the following RINA concepts will be introduced: Distributed Application Facility (DAF), Distributed Application Process (DAP), Application Process (AP), Distributed IPC Facility (DIF), IPC Process (IPCP), Error and Flow Control Protocol (EFCP), and Common Distributed Application Protocol (CDAP). Their definitions as well as a full list of RINA-related concepts and terminology can be found in [8].

A DAF consists of individual processes, called DAPs, cooperating across one or more processing systems, that exchange information using IPC and maintain a shared state. Each DAP has tasks that manage distributed resource allocation, storage, and IPC within the DAF as well as the tasks specific to the distributed application.

A DIF is a specialization of a DAF. It is a collection of two or more APs that cooperate to provide IPC services to applications or upper layers. The elements unique to a DIF are delimiting, EFCP, and the flow allocation. An AP is a program instantiation that is intended to accomplish a specific purpose, while an IPCP is a special AP within a DIF that delivers IPC services.

A distinguishing feature of RINA is that a DIF provides a clear separation between mechanisms and policies. While all DIFs are independent of each other, each DIF provides the same set of fixed mechanisms such as data transfer, routing, congestion control, etc. What is different is that these mechanisms can be configured differently for different use cases, i.e. using different policies. In RINA, if an application makes a request to allocate networking resources, then this common RINA layer determines the mechanisms and policies that will satisfy the application's request.

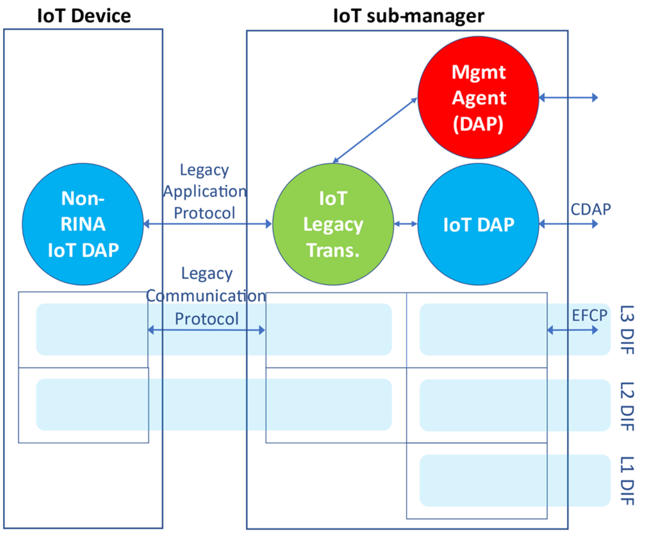

Fig. 1. A RINA-based representation of an IoT device and an IoT sub-manager.

The tasks unique to a DIF are Flow Allocation, Delimiting, Relaying, and EFCP. EFCP deals with maintaining an IPC instance within a DIF. EFCP is a data transfer protocol that ensures reliability, order, and flow control. On the other hand, CDAP enables distributed applications to deal with communications at an object level, rather than forcing applications to explicitly deal with serialization and input/output operations. It helps sharing data over a network without having to create specialized protocols and thus can be used to construct arbitrary distributed applications.

C. *Design of IoT Architecture Using RINA*

The short answer to what IoT architecture is, is that it is the same as the architecture of network management. Put another way, network management is one kind of IoT. In general, IoT consists of devices of all kinds such as sensors and actuators that are connected to the Internet using different protocols. Irrespective of the device type or supported protocol, the function of an IoT device is to provide an analog-to-digital (A-to-D) interface between the analog and digital worlds that is accessed by an IoT Agent (IoTA). An IoTA is an application in IoT. The A-to-D interface resides in the device's operating system.

An IoT device may have one or more A-to-D interfaces that are accessed by one or more IoTAs, and one Network Management Agent (NMA), used to manage the DIFs in the IoT device. The IoTAs belong to an IoT DAF that is managed by an IoT Application Manager.

An IoT network may consist of different systems, and every system has its own NMA. The NMAs belong to the Network Management DAF (NMA DAF). These agents report to a Distributed Management System (DMS) that aggregates and processes sensed data to make it available to the users of the IoT network. The DMS is the IoT Management Application. DMS is distributed across IoT managers that process DMS data for different regions of devices. DMS also includes IoT sub-managers that have the task to translate legacy application protocols to a common protocol as well as preprocess DMS data for sub-regions of devices. Translation is needed as IoT devices use different application and communication protocols.

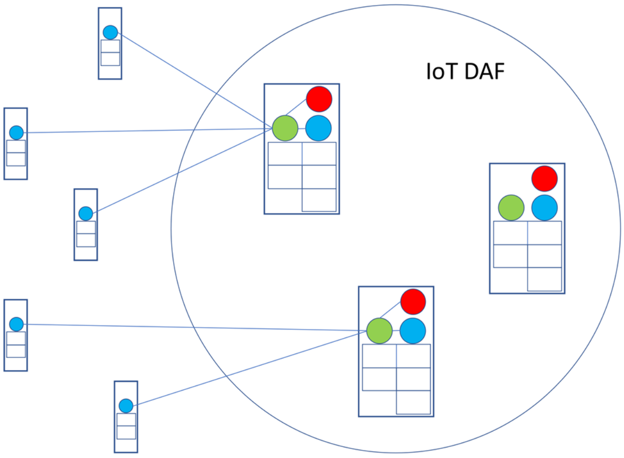

Fig. 2. IoT DAF.

Fig. 1 gives the main constituent element of a RINA-based architecture for IoT. It includes an IoT device and an IoT sub-manager. An IoT device has two DAPs: one DAP is part of the IoT DAF, and the other DAP that is part of the NMA DAF. An IoT device may employ two or more types of DIFs: one type of DIF interfaces to each instance of the physical medium (different media may require a differently configured media DIF). The other DIF which is over the media DIFs, is concerned with managing resource allocation across the different instances of physical media.

The main purpose of the IoT sub-manager is to translate a legacy (non-RINA) IoT protocol to a common RINA protocol – CDAP. CDAP achieves commonality by taking advantage of what has been known in the networking industry since the late 1980s – namely, that application protocols modify state external to the protocol, and the only operations that can be performed remotely are read/write, create/delete, and start/stop on object models. (CDAP is based on the experience with CMIP, HEMS, and even SNMP). As noted earlier, these simple operators ensure that the focus is on the definition of the object models, i.e. the application, where it belongs. Clearly, there can be a wide range of object models, which are not part of the protocol. Hence the number of application protocols is actually quite small [6]. This approach is a major advantage to development and shortening time to market. It is much easier to add new object models than to modify an existing protocol. Given that CDAP is independent of the object models, arbitrary application protocols (e.g. Z-Wave or ZigBee) can be translated using an identical underlying methodology since they are mostly concerned with operations such as read/write, create/delete, and start/stop.

The Legacy Application Protocol communicates with the IoT Legacy Translation DAP using the Legacy Communication Protocol. The Legacy Communication Protocol carries the data from the IoT device to the top layer (3^{rd} level DIF) of the IoT sub-manager, which in turn delivers the data to the IoT Legacy Translation DAP. The IoT Legacy Translation DAP passes the result to the IoT DAP which uses EFCP as a common communication protocol to transmit data to other parts of the IoT network. While undoubtedly a necessary evil initially, the advantage to the customer (the owner of an IoT installation) to eliminate these translations cannot be overstressed. They are not only unnecessary overhead but also hold the potential for unforeseen (and potentially catastrophic) behavior.

Every IoT network needs to have at least two DAFs: one DAF to manage the communications and another DAF for the IoT applications. The purpose of the first DAF is to manage the IoT network; there are three cases that can be listed in this context: a) IoT devices may be purely end systems connected to media level routers (bridges), b) forward data to other IoT devices in its network, or c) be directly connected in a star configuration to an IoT sub-manager as shown on Fig. 1. On the other hand, the purpose of the second DAF is to manage the IoT application. Depending on the problem domain, more than one IoT device type may be managed by one IoT management application. The commonality RINA brings to these DAFs is again another major advantage.

Fig. 2 represents the IoT DAF. The IoT DAF only includes IoT sub-managers because they are operating within the RINA model. While the IoT devices do not belong to the IoT DAF, they can become part of it by introducing a RINA DIF that will replace the need for protocol translation. After moving to commonality with such a DIF, the entire network would be within the same DAF and primary IoT DAPs could have direct access to the sensors. The IoT DAF has its own DIFs with policies that support its communication requirements. By distinguishing the IoT manager and network manager, it is possible for a single network manager to support multiple IoT applications, or for the two to have different entities (organizations) responsible for them.

IV. Benefits of RINA in IoT

The section discusses why RINA leads to a better architecture for the IoT and what are the possible advantages for the industry.

A. Reduced Protocol Complexity

As mentioned above, little work has been done to reduce the root cause of fragmentation – the number of protocols itself. Existing IoT middleware solutions, including those sponsored corporately (e.g. IoTivity [9]), can be regarded as client-server application layer frameworks without support for distributed applications (or in RINA terms, without the DAF infrastructure). These solutions are narrowly focused and do not support features such as mobility, multicasting and multihoming, which are inherent (no additional overhead) in RINA.

RINA has the potential to simplify IoT – it can reduce the number of protocols and the complexity of network management. As indicated in Fig. 1, the IoT sub-manager deals with translation between various legacy IoT protocols and the IoT DAP. Translation can be problematic in itself as it is hard to implement and may create indeterminate delays. However, translation can be used as a first step to achieve commonality as the main goal is to eliminate the IoT sub-managers. A complete move to commonality would be to introduce a RINA DIF in the architecture, then push the DIFs down to the physical layer, and at the appropriate time replace the legacy IoT protocols. This would also allow moving to common (standard) object models.

Commonality across IoT is a major advantage for IoT customers who should demand it. It is not in the self-interest of the vendors. It cannot be stressed enough that while IoT is very important to the vendor of IoT devices, IoT is not important to the customer. The customers have a use of IoT that benefits their business. They are not in the IoT business.

They just want it to work and do not only what they need but also what they need to do next. Maximizing commonality is of primary importance to the customer. Commonality (plug and play, interchangeable parts) keeps the customers' options open and avoids their capture by a vendor. As a comprehensive theory of networks, operating systems, and distributed applications, RINA would provide commonality beyond the narrow focus of IoT, which would also be the best basis for driving the commonality. Furthermore, the architecture in Fig. 1 proposes how RINA can be used to solve current problems in the industry, thus challenging the misconception that RINA requires "throwing everything away".

B. Improved Standardization and Reduced Time-to-Market

Moving to a common RINA DIF is a move to real standardization as opposed to erecting barriers. Instead of supporting a wide range of IoT protocols, only a single DIF layer is needed together with an IoT DAP. This follows the findings presented in [7] that the IoT protocols are degenerate cases of EFCP. Such a single layer can achieve interoperability between different IoT platform and solution providers. The RINA-based standardization will ensure a common interoperable IoT architecture in which the IoT providers would compete on factors such as cost, functionality, usability and quality of service rather than creating barriers to entry. This will effectively reduce or completely eliminate today's problem with fragmentation in IoT. In addition, instead of building their own IoT platforms, industry companies will focus on what brings value – services and applications. Without commonality in the architecture, it would be expensive to create innovative services and applications, while at the same time having to build the underlying architecture (this is similar to having to build a proprietary WAN network ignoring the fact that leasing circuits from common carriers is incomparably cheaper). RINA can also help reduce the time-to-market by eliminating a significant part of the cost needed to build a common architecture, and instead help focus the efforts on developing services and applications that bring the business value in IoT. It is also worth mentioning that constrained IoT devices would benefit from a lower header overhead (a comparison between the RINA and TCP/IP header sizes is available in [10]). A lower header overhead eliminates the need for header compression, which is one of the features that makes the implementation larger in the case of IP-based IoT devices.

C. Improved Security

RINA comes with inherent security advantages over the mainstream TCP/IP model. Due to RINA's architecture specifics, security is not merely an after-thought. Security is inherent to the structure. The recursion of the single layer provides security isolation in itself and end-to-end security is provided from within the architecture. It has been shown that RINA can resist a number of security attacks faced by TCP/IP even without using cryptographic techniques [11]. For example, as documented in [11], some of the security features in RINA are that it requires explicit enrollment of IPCPs when joining DIFs, it has addresses that are internal to a DIF (unlike TCP/IP's global addressing space which allows systems to freely connect to each other), and the applications in RINA are accessed by their application name (while in TCP/IP they listen to a well-known port). IoT can readily leverage these features that RINA possesses inherently in its networking model.

V. Conclusion

IoT is expected to create enormous business value, but it faces interoperability and fragmentation issues that prevent it from realizing its full potential. While the existing literature focuses on the need for middleware to hide the underlying implementation details, little work has been done to eliminate the root cause of fragmentation – the number of protocols itself. This paper presents a novel RINA-based architecture for the IoT that reduces the protocol complexity, improves standardization as well as enhances security. We propose IoT architecture with RINA from the perspective of network management. We define the elements of an IoT network and discuss that it consists of at least two DAPs: one DAP is part of the IoT DAF, and the other DAP that is part of the NMA DAF. Based on the premise that networking is inter-process communication (IPC), RINA uses a single layer called DIF that recurses over different scopes. We discuss how commonality in IoT can be achieved by introducing a RINA DIF. Commonality simplifies IoT as it eliminates the need for various IoT protocols, improves standardization, and reduces the cost to build services and applications.

References

[1] G. Omale, "Gartner Identifies Top 10 Strategic IoT Technologies and Trends," Gartner. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2018-11-07-gartner-identifies-top-10-strategic-iot-technologies-and-trends. Accessed November 2019.

[2] J. Manyika et al., "The Internet of Things: Mapping the Value Beyond the Hype," McKinsey Global Institute, June 2015. [Online]. Available: https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/the-internet-of-things-the-value-of-digitizing-the-physical-world. Accessed November 2019.

[3] A. H. Ngu et al., "IoT Middleware: A Survey on Issues and Enabling Technologies," IEEE Internet of Things Journal, Vol. 4, No. 1, pp. 1-20, February 2017.

[4] J. Lin et al., "A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy and Applications," IEEE Internet of Things Journal, Vol. 4, No. 5, pp. 1125-1142, October 2017.

[5] A. Al-Fuqaha et al., "Internet of Things: A Survey of Enabling Technologies, Protocols, and Applications," IEEE Communication Surveys & Tutorials, Vol. 17, No. 4, pp. 2347-2376, June 2015.

[6] J. Day, "Patterns in Network Architecture: A Return to Fundamentals," Prentice Hall, 2008.

[7] S. Chawla et al., "IoT or Coping with the Tribble Syndrome," Pouzin Society. [Online]. Available: http://pouzinsociety.org/iotsyndrome. Accessed January 2020.

[8] Pouzin Society, RINA Education, Terminology. [Online]. Available: http://pouzinsociety.org/education/terminology. Accessed November 2019.

[9] IoTivity. [Online]. Available: https://iotivity.org. Accessed January 2020.

[10] J. Day, E. Grasa, "About Layers, More or Less," Pouzin Society, January 2016.

[11] G. Boddapati, J. Day, I. Matta, L. Chitkushev, "Assessing the Security of a Clean-Slate Internet Architecture," 20th IEEE International Conference on Network Protocols (ICNP), October 2012.