Gateways for the Internet of Things
An Old Problem Revisited

Vinton G. Cerf
Google, Inc.
Mountain View, California, USA
Email: vint@google.com

Peter T. Kirstein
University College London
London, United Kingdom
Email: P.Kirstein@cs.ucl.ac.uk

***Abstract*—An early paper in network interconnection outlined the choices between adoption and adaption of protocol structures at different levels of the early Internet. This paper revisits that theme in the light of the many advances of the last three decades, and the emergence of the Internet of Things. We maintain that there are close parallels between the variety of incompatible networks which were in vogue in the early days of the Internet and the current situation with domain-specific sensor and actuator systems in the Internet of Things. We point out that there is now much more agreement on the approach to basic services, and a more universal approach to adaption. While universal adoption of common technologies is still far distant, there has been a broad consensus on the use of the Internet and web services for the access to such systems. Because of the standardization in some of the levels of web services, particularly in the context of IPv6, a consistent architecture is defined and examples of possible implementations given.**

***Keywords*-Internet of Things; IPv6; Gateway; end-to-end; Architecture, DEVNET.**

I. Introduction

Thirty-five years ago these two authors had an argument: what gateways between networks were needed, and what should they contain? The result of the argument was a paper [1]. As far as we know, this was the second paper on gateways for internetworking (the first was about TCP [2] [1]). The authors did not agree with each other. Cerf believed that the whole world would recognize that TCP/IP (which evolved from the original TCP) would be the way to go, and thus a gateway need only have the functionality to transfer data from one network to another at the network level. Kirstein felt that Cerf was unrealistic; from his experience with commercial computer networks, there would continue to be a wide variety of network architectures and technologies. For this reason, gateways would have to translate all the different layers of protocol making them application gateways with a much more complex structure and functionality. For the first 15 years Kirstein was right; for the next 20, Cerf showed his superior foresight! In recent years, there has been an interest in the Internet of Things (IoT) [3]. Here the concept is that there will be countless billions of appliances, sensors and actuators in all walks of life and they will all be networked together and accessible via the public Internet. The concept of IoT is so general, that it covers a large proportion of the activities in modern society. In this paper we consider a particular subset: ones in which to access information from a number of sensors, actuators or appliances through the Internet, and/or to pass information to them. In some sense, even this limited subset allows for very general interpretation. Any appliance that can provide state information and accept some form of advice or control might fall into this subset. A printer, for example, can have state (e.g. How much ink is left? Has the paper supply run low or out? Has there been a jam?) and can accept control (e.g., Print this file now. Switch to a different paper supply.).

It is largely agreed that the wide-area technology that will connect all these devices together will be the Internet possibly with some extended functionality that will emerge as the current Internet continues to evolve. While currently most systems are still using the IPv4 protocols family [4], the newer IPv6 family [5] is now becoming more prevalent. We remain divided, however, as to what extent the networks that link the devices to the Internet will follow the Internet architecture, or will persist in using various proprietary protocols. We seem to have a replay of the thirty-five year old discussion! This paper tries to address this question in the light of the current problems, current environment and available technology. We are also conscious of the growing problems with IPv4 exhaustion and the formal introduction of IPv6 into the operational Internet on June 6, 2012. We expect that IPv6 will become the long-term target of Internet evolution at least at the IP-layer in the protocol architecture.

Fundamental to our considerations is our model of the environment of the Internet of Things for a specific application or domain.

We begin with the presumption that the public Internet or its evolutionary successor is used by service providers to access and control devices that are connected on a possibly

[1] In [2], published in May 1974, one finds this sentence: After a brief introduction to internetwork protocol issues, we describe the function of a GATEWAY as an interface between networks and discuss its role in the protocol.

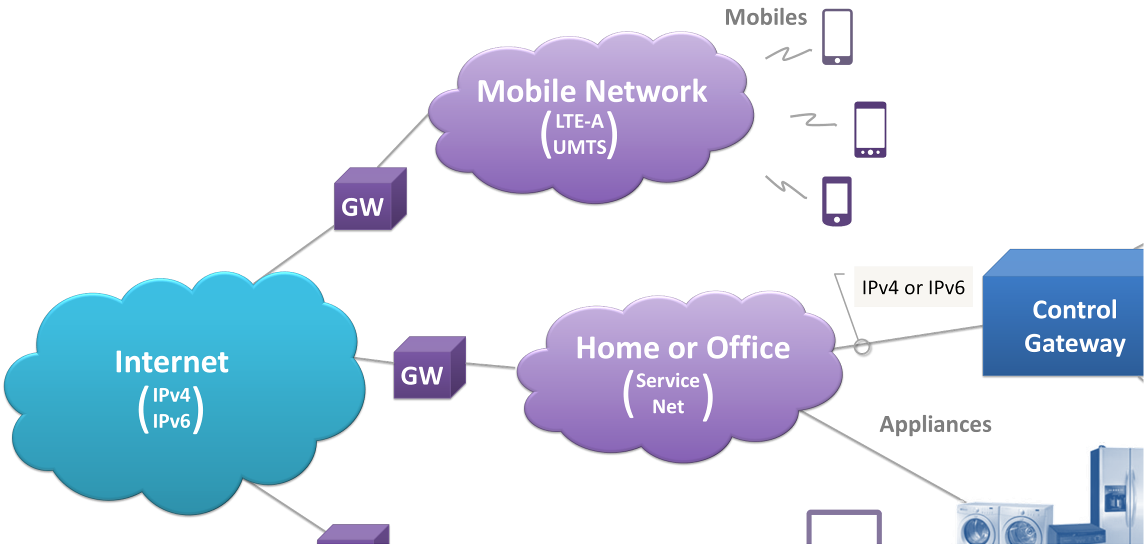

Figure 1. A general architecture of the Internet of Things.

proprietary device network in a residence or office building or industrial plant. This device network we will refer to generically as DEVNET, keeping in mind there may be several of them and they may use multiple proprietary protocols or perhaps Internet protocols such as IPv4 or IPv6. The services require data from the devices and may operate on and send data to the devices (appliances, sensors, actuators, and so on). We imagine that the DEVNET(s) are connected to a network that is local to the residence of office building. We will refer to this as the SERVICENET. It seems likely that the SERVICENET network will be IPv4 or IPv6 capable, and probably both in time. While eventually we hope, and expect, that all sensors will in some way be part of an IPv6-enabled device network, this is not the case today and may not be for quite a long time. Certainly currently many sensors are not even IP-enabled; though many popular types are starting at least to have IPv4 or IPv6 interfaces. While we cannot, for the purposes of this paper, specify the technology used for the devices, we do assume that at least an IPv4 interface is provided to the DEVNET, possibly by way of an application layer gateway (ALG).

Between each of the networks of Figure 1 (ie. Internet, SERVICENET and DEVNET) there will be some form of gateway. The subject of this paper is to consider the functionality of these gateways, and whether some of that functionality may be held elsewhere than at the boundary between the networks. In this paper we will first review in Section II the salient points of the gateways in [2]. Here we will stress the simplifications that arise if there is a wider adoption of the same protocols at various levels. Then, in Section III, we will highlight the impact of directories and Name Servers, which came in to the picture in the '80s. In Section IV, we introduce the concept of scope, that became essential as the networks grew larger and more complex. We have already highlighted the importance of the Internet of things (IoT). What we mean by this is discussed in Section V, and its functions in Section VI . One of the critical items in the acceptability of the IoT is that networks and the objects they touch can be made secure in the sense that only authorized parties can interact with the devices on the DEVNET(s). While this was not a serious consideration at the time of [1], it became significant soon after. There is a discussion in Section 6 on how it can be provided in a fairly general way. Finally, some conclusions are drawn in Section VII.

II. The Gateways in 1978-1983

The Gateway paper [1] was written in 1978. This was when the only networks deployed were either based on X.25 [6] from the carriers, or based on proprietary protocols, or protocols from the DARPA programmers. The latter were based either on the old Arpanet NCP [7] or were based on the newer Internet protocol TCP/IP [2] between networks like Packet Radio, Packet satellite and bits of the ARPANET. In fact it was only those based on the newer Internetwork protocol that we were envisaging for connecting networks. At the time of [1], we had a fairly sophisticated notion of protocol layering but there was more to come (Figure 3). We knew that each network would need a list of its hosts, but the concept of a distributed name server, like the DNS [9] had not been invented. To ensure the relevance of this paper to the current situation, we will adopt a network and interconnection model that would have been formulated only about five years later. We will adopt the notation of Figure 2. Here the left side shows the somewhat crude model where protocol levels are ignored. The right side shows the model

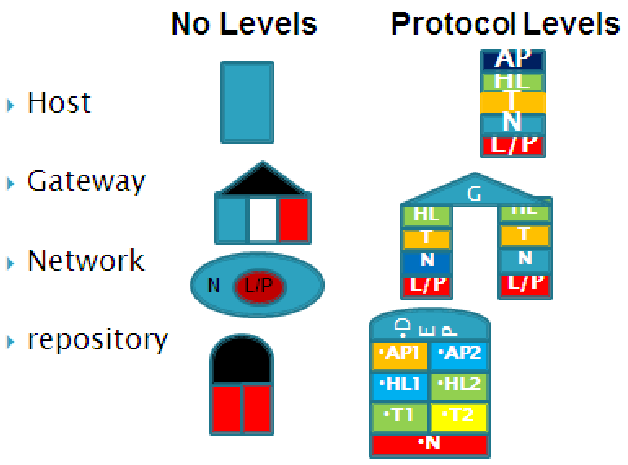

Figure 2. Notation for servers, networks, gateways and protocol servers.

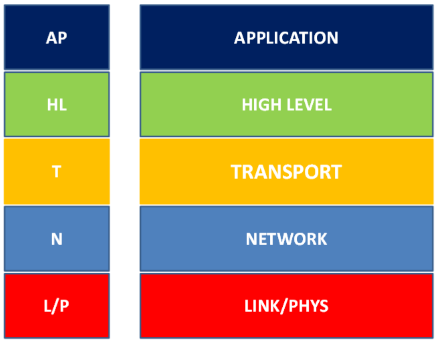

Figure 3. Protocol Levels.

when protocol levels are included.

In Figure 2, we must expand the individual components to describe the protocol levels. This we do in Figure 3.

The simplest connection of two networks is as shown in Figure 4.

The concept of application layer gateway (ALG) has been around for some time [28]. In principle, all layers of protocol below the application are terminated at the gateway that connects one network to another. Application layer information is re-encapsulated in the new lower layers and sent into the new network to the next gateway or destination. The application layer information or possibly configuration information in the gateway is used to correctly exercise the protocol layers below the application in the next network.

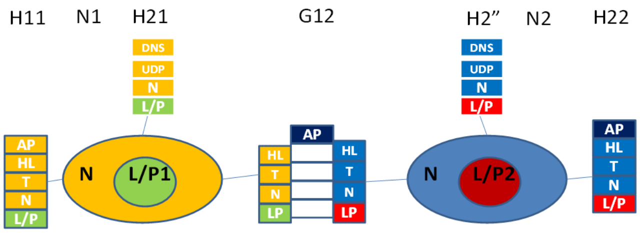

Figure 4. Adaptation between protocol layers in a gateway made up of two half-gateways.

When the same protocols are adopted in different networks at a particular level, the functionality of the gateway is simplified greatly. In the Internet, for example, the networks may use different physical and link-level technologies, but adopt the same network and higher levels, in that case an ALG is not needed, and the gateway becomes a router. In another scenario, the lower level protocols may be identical; just the application level is different. An example in the '90s was in the electronic mail application. Here the Internet suite used SMTP [10] while another community used X/400 [11]. Here the adaption could be carried out in a server in one of the networks. Here there were sometimes differences in the facilities supported. For example one might support the automatic sending of a "receipt" message when the recipient opened a message and the sender had so stipulated. Two forms of adaptation could then be used. One was just not to exercise that functionality; the second was to extend the application to include it. Both these techniques were used and are relevant to the IoT.

III. The Rise of Directories and Name Servers

The sizes of the networks in 1978-83 were still small, though the number of users was already very significant. It became clear soon that various directories were needed as the networks grew. By 1975, there were already at least two areas where directories were needed: Hosts [12] and people [13]. Various Directories for people were developed in the commercial systems, the ARPANET and later other National Research Networks (NRENs). These indicated which machines to access, and how the network was to organize its internal routing. Initially there were local directories only for people at each site. However, as collaboration services like Chat and e-mail grew, it was clear that centralized services spanning the whole network were needed for these. For instance, ARPANET introduced the WHOIS system [13]. As one went over to the Internet protocol suite, the number of hosts and users grew. For these, a centralized system could not cope. This led to the Domain Name System (DNS) [9]; this is a globally distributed system. Around the same time, the British NREN developed their National Registration Scheme (NRS) [12]. While both used a similar hierarchic system e.g. host.stanford.edu for the DNS, they used opposite hierarchies; thus the NRS would have used edu.stanford.host. When communicating between the two networks, both refused to adopt the others' naming, so the gateway had to adapt.

For directories relating to people, the result, in the late '80s was largely to adopt the distributed X.500 system [14], which could hold many more attributes and be searched in more complex fashions. In both cases, it was soon realized that the establishment of a globally distributed service was a major effort, and attempts were made to increase the range of objects that could be stored, and the functions that the repositories could perform.

For the DNS these extensions were resisted strongly, because of the need to ensure that the servers had very high performance. Recently this has had to be relaxed in two areas. First it has become clear that a secure DNS is required; hence a secured version is now being deployed [14]. Second, with the inevitable move to IPv6, the applicability of the DNS has had to be extended to include that protocol.

While directories of people remain important, the function of their Repositories has grown enormously. Some are now based on locality, others on organizations. Many have now been integrated with other services such as organizational function, telephone number and e-mail address. Partially as a result, there is little homogeneity in such repositories. In the early '90s the World Wide Web [16] appeared on the scene. This now included far richer information. Web servers became one of the principal repositories of information. Web searching became a huge and powerful industry, and methods for processing the objects in the web servers became one of the principal forms of information service. Thanks to the World Wide Web Consortium (WWWC) [17] and the Internet Engineering Task Force (IETF) [18]. Web services [19] became standardized and very widely adopted. Thus two sets of repositories have been adopted: The Domain Name System (DNS) [9] for resolving name/address mapping, and web services for other data. Mechanisms for securing both forms of repository have been standardized.

IV. The Concept of Scope

Very early in the development of directory technology, it became clear that scope was a very important concept. While some quantities are needed to be known very broadly, others are needed in a much narrower domain. Of course many directories have to be compiled locally; in some cases the data must be accessible widely, in others only locally. In the DNS, this process was formalized in the delegation of zone management and the hierarchic naming structures. Each Domain had to be registered at a relevant level of the naming tree. Each globally accessible end-point entity had to be registered in a local domain name zone, which itself was hierarchically bound in the DNS.

IP address management followed a parallel track. The notion of Autonomous System (AS) emerged as a network of routers using a common routing algorithm. Several routing protocols were developed, all of them members of the class called "Interior Gateway Protocols" (IGPs). The Autonomous Systems were interconnected through specialized routers that execute a Border Gateway Protocol (BGP) that is now in its fourth version. Each AS had to have more than one route to it in order to be registered in the global routing tables created by the Border Gateway Protocol - otherwise it would be announced to the Internet only by the AS from which it obtained its Internet access.

Some users of the IPv4 address space found that they needed to attach more devices than the address space available. To accommodate this need, the notion of private address space was developed [29]. These required a local (and often rapidly changing) cache of private addresses that were mapped at need to globally routable addresses. The technique was called Network Address Translation (NATs) [30]. It is expected that the much larger address space of IPv6 over IPv4 would obviate the need for NATs, and its complex system of cache management and mapping.

V. The Internet of Things

When the *Internet of Things* (IoT) [3] became of wide interest, it became accepted that information from and about the *things* should be stored in repositories, and accessed through web services. To that extent, there was *adoption*. However it has proved very difficult to standardize these repositories. This is mainly because the IoT has such a broad area of applicability, that there is a range of requirements that are often contradictory. For example in some cases the integrity and authentication are so vital that a major effort must be gone through even to store the object in the repository; examples are copyrighted documents and patents. In some access to the objects must be protected strongly; examples are confidential documents, medical records and many personal details. Sometimes information can be, and must remain, a local phenomenon. In others wide area access is essential with relevant restriction on access. All that one can say with confidence is that often there needs to be constraint on access, authentication of and access control to objects stored and a uniform mechanism, via web services [19], to the information stored. Thus at the level of access to the repositories, web services are the norm and *adoption* has taken place. As regards the semantics of the attributes, there is no such agreement and *adaptation* must be performed if different systems are to cooperate. For example two such systems are the Handle system [20] developed initially for copyrighted documents, and the Electronic Product Code Information System (EPCIS) and Smart Things Information System (STIS) [21] developed for RFID objects. Under a recent project [6], both are used to store information about objects, and interfaces between them have been constructed.

The basic concept of scope carries over from the Internet to the IoT. It may well be necessary to obtain information about objects on the IoT and even access them, through the Internet. It will often be unnecessary, and even undesirable, for different security reasons, to provide detailed information about them outside a particular domain. Thus, for example, one may require information about whether sensors in a building reach alarm temperatures. The request for such information may be made to a repository associated with the building, which stores detailed configuration information

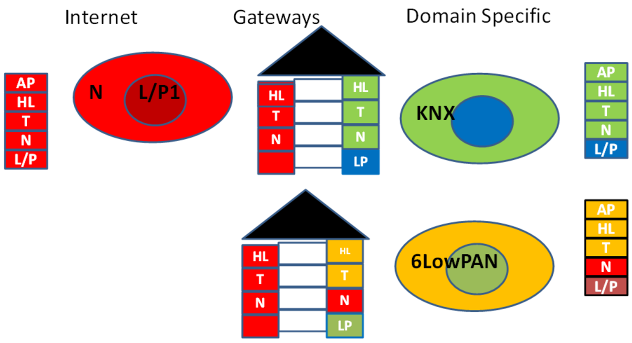

Figure 5. Access to sensor data through the Internet and a sensor net.

in its database. It is not necessary that there be any global awareness of this configuration information. The repository, not the sensors, would be the end-point of the interaction and that site that must be globally accessible. The repository, however, will need direct access to the sensors to gather data or to set configuration information. In some designs, the repository is contained within the gateway in the form of a proxy that acts both as a kind of application layer gateway and a device controller. There are no paths from the outside world to the device, only to the proxy. In other designs, the repository is located somewhere on the Internet and must be able to communicate securely with the devices it manages or controls. All others needing information about the devices will interface only with the repository. Once again the repository serves as a proxy for the devices.

Similar considerations about adoption and adaptation apply to gateways, repositories and sensors. Figure 5 shows a typical example of a monitoring process in a control machine on a network Let us consider the more detailed schematic of Figure 6.

Either a person or a monitoring process wishes to access the sensor data on a number of sensors on a sensor net N2. N1 and N2 are connected by a gateway G12. It is an application gateway containing an application gathering the data from the sensors. The upper network is a completely foreign technology it is shown as KNX [22], though there are many more in use. The gateway is shown as a complete application layer system analogous to the situation in Figure 4. The lower network shown uses 6LowPAN [23]. This is a technology based on the Internet, which normally supports only UDP [25] as the Transport protocol. Because of this amount of adoption, this second case already has a much simpler gateway than the upper one. Another available integration of IPv6 for constrained devices is GLoWBAL IPv6 [24].

In fact, we believe that in many use cases for IoT, there will be much more adoption of a whole system architecture and we hope but cannot be certain that this will occur soon. While there will be many variants, we think that many of the use cases can be described by the configuration of Figure 6.

Fundamentally a Host H1 wishes to gather sensor data from DEVNET NS. There are now three networks. N1 is the normal Internet. Because of the sheer number of IoT objects, and some of the advantages of newer IPv6 over the traditional IPv4, there are considerable advantages in using IPv6 at some stages of the system. In fact there are advantages in using IPv6 for the N1, but we realize it may be some time before the whole Internet moves over to IPv6, and it is important to use the general Internet in the architecture of Figure 6. N3 is a device network. Currently in many environments it is uses quite a different network technology than the Internet in-line with the network interconnection system 25 years ago with the Internet and legacy networks. N2 is a domain-specific SERVICENET; in that portion of the system we can insist on IPv6 in order to capitalize on its many advantages. Because this domain is normally used for specific domains or application areas, there is little loss of generality, and many advantages, in specifying that it be IPv6.

There has, indeed, already been a fair amount of adoption in the interface between N2 and N3. Many of the legacy networks have added Internet interfaces albeit most IPv4. However it is straightforward state-of-the art to add an IPv4-IPv6 gateway. In which the IPv6 address can be tunneled over the IPv4 Internet. This is at most what is needed in G12, and internally in G23 of Figure 6. There has been a general move to use web services for this sort of applications. That implies using HTTP/TCP in Host H1, and in the repository. At each level there may be different degrees of adaptation and adoption in Figure 6 following the argument used earlier after Figure 4. Because H1, REP-2 and G12 are all on the Internet, they must all *adopt* the Internet access protocols. In fact, these protocols have many variants, and there may be technologies contained in N1 providing only that they obey the Internet protocols. To the level of detail shown, it would be possible, for example, for the machine H1 to use IPv4 while the Repository uses IPv6 provided there is carried within the payload of the IPv4 packet. There may be, by the same token, different network access technologies like that of the network access of H11 is WIFI, while that of REP-2 is Ethernet. Again this merely requires processing inside N1. In the same way, the G12 is considered an *Applications Gateway*.

It uses whatever protocols N1 requires when communicating with REP-2, and that N2 requires when communicating with the sensors. There may be different amounts of processing in AP1 and AP2; for example, there may be processing of the raw data before it is passed to the Repository. There has been further adoption on the DEVNET interface. We have stated already that web services are largely accepted as a way of accessing sensor controllers. When the DEVNET is running IPv6, there has been further standardization of a simpler form of web service interface called Constrained

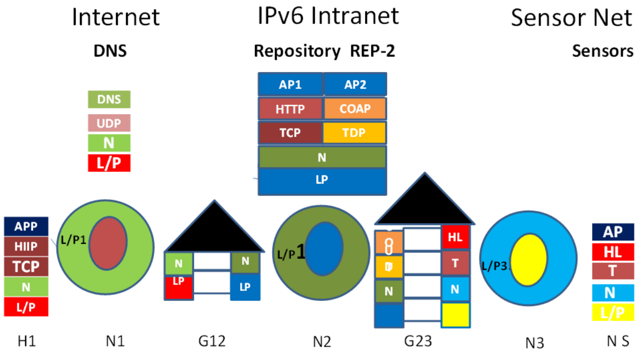

Figure 6. Reference Architecture for the Internet of Things.

Application Protocol (COAP) [25].

This last is a further important aspect. Where *adaptation* is needed, there may be different locations where it is done, and optimization procedures to determine the most suitable. To give a simple example, AP1 may request the reading of a whole group of sensors, which may not be supported in N2. If the relative adaptation operations are carried out in G13, it may minimize network traffic in N1 but at the cost of making the operations needed in that gateway much more complex. It may be much more efficient to carry out the group translation operations in REP-2, and carry multiple operations between REP-2 and G23. Although the network traffic in N1 is higher, the greater simplicity in G13 may make this very worthwhile. This is particularly relevant if G13 is itself a constrained device because of size or power constraints. This is the situation envisaged in Figure 6. Here we are assuming that Web services are used between a monitoring process and data storage of the sensor data collected. This is the reason HTTP/TCP is shown in Figure 6 as the transport and High Level protocols between the repository and the monitoring Host, On the other hand, the protocols in the half-gateway of Rep-2 will support COAP/UDP. This is the reason for the split structure in Figure 6.

Some similar considerations concern the repositories themselves. If all the repositories use a similar technology (adoption), it may be possible to have part of them work as a distributed repository. An example of this is the Handle System [20], which has the capability of being a distributed system. If the repositories use different technologies, then an adaptation layer between them is required which may make the distribution much more tricky.

VI. Security in Gateways

Even in the original environments of [1], some primitive security measures were considered. At that time simple password protection was included, for example, in gateway allowing traffic to pass between the UK and US in [26]. In the IoT, security considerations are paramount. Without adequate security provision, many application domains are not feasible. The requirement that COAP be in the end-points, or the gateways to end-point networks, do take this requirement into consideration. COAP has full provision for security, using the DTLS protocol [31] for this purpose. It is hard to go further in the general case, since many of the legacy sensor networks have not taken this problem seriously relying to the local nature of many of the deployments. In a system like Handle, all Internet access is controlled by using the full infrastructure of a Public Key system, with the capability of checking fine-grained authorisation, integrity and confidentiality. Only authorised operations would then be passed through to the subsequent technology translation servers running COAP with a cryptographically secured token. Thereafter the COAP DTLS would be adequate for secured operation of the actual gateways.

When security has been taken into account in these systems, then the same consideration of adoption and adaptation will occur for security. There is, however, one more twist. The DTLS has some four modes for the security infrastructure. It would be quite in order to choose the mode that is supported by the legacy system easing the task of adaptation.

VII. Conclusions

We have shown that many of the considerations that applied to the Internetworking scene in 1980 apply again to the interconnection of legacy IoT systems into the Internet. We have tried to ease both the analysis and the implementation problems of the IoT by segmenting the general system into three domains: the Internet, the SERVICENET and the DEVNET. The first and the third have very broad scope; homogenization across all the relevant applications domains is very hard. We are able to concentrate on the SERVICENET, where is much more under the control of one or more targeted domains. By introducing the concept of half-gateways, we have reduced further the variables in the individual gateways. Although we have not developed the argument further in the text, the repositories will adopt a similar approach. In one direction they will be accessed and controlled through the Internet probably using web services. Facing the sensor networks, there will be relatively simple requests and responses between the repositories and the sensor networks. By careful decomposition of the transactions into different levels, it should be possible to localize the areas in which adaptation is necessary. Many examples of this are given in the text. By adopting common standards at some levels, the amount of adaptation needed can be greatly decreased.

In the IoT6 project we have carried through the implementation of the interconnection of some particular legacy sensor systems into the Internet [27]. While we have carried out the implementation there in half-gateways, there has not yet been any attempt to decompose the gateway to optimize the location of the adaptation functions. While we hope this paper has made a start on simplifying the connection of foreign device and sensor networks into the Internet, it is only a beginning. The complexity of the task, the range of foreign systems, and the variety of the application areas are far more daunting than the network interconnection of the '80s. In 1978, we under-estimated the degree of uniformity that would be achieved in only 15 years from all adopting the Internet approach. At least one of us felt that large-scale adaptation would be needed for many decades. We hope that common adoption occurs much more universally and faster than the authors now expect. Thus the problems of adaptation would be reduced dramatically.

Acknowledgment

We would like to acknowledge the invaluable help of Antonio J. Jara of the University of Murcia in preparing this paper. This work has been sponsored by the the STREP European Project "Universal Integration of the Internet of Things through an IPv6-based Service Oriented Architecture enabling heterogeneous components interoperability (IoT6)".

References

[1] Cerf, Vinton G., and Peter T. Kirstein. "Issues in packet-network interconnection." Proceedings of the IEEE 66.11 (1978): 1386-1408.

[2] Cerf, Vinton G., and Robert E. Icahn. "A protocol for packet network intercommunication." ACM SIGCOMM Computer Communication Review 35.2 (2005): 71-82.

[3] Atzori, Luigi, Antonio Iera, and Giacomo Morabito. "The internet of things: A survey." Computer Networks 54.15 (2010): 2787-2805.

[4] Postel, Jon. "RFC-791 Internet Protocol DARPA Internet Program Protocol Specification." Information Sciences Institute 2 (1981).

[5] Deering, Stephen E. "Internet protocol, version 6 (IPv6) specification." (1998).

[6] IOT6 European Project. "Universal Integration of the Internet of Things through an IPv6-based Service Oriented Architecture enabling heterogenous components interoperability" (2012).

[7] Deasington, Richard J. "X. 25 explained: protocols for packet switching networks". Halsted Press, (1986).

[8] Carr, C. Stephen, Stephen D. Crocker, and Vinton G. Cerf. "HOST-HOST communication protocol in the ARPA network." Proceedings of the May 5-7, 1970, spring joint computer conference. ACM, (1970).

[9] Atkins, Derek, and Rob Austein. "Threat analysis of the domain name system (DNS)." (2004).

[10] Postel, Jon. "Simple mail transfer protocol." Information Sciences (1982).

[11] Kane, John R. "X.400: Electronic mail message delivery system." U.S. Patent No. 5,487,100. 23 Jan. (1996).

[12] Wells, Mike. "NRS and JANET-the United Kingdom Joint Academic Network." Serials: The Journal for the Serials Community 1.3 (1988): 28-36.

[13] Daigle, Leslie. "WHOIS protocol specification." (2004).

[14] Chadwick, David. "Understanding X. 500: The Directory". Chapman and Hall, Ltd., (1994).

[15] Kolkman, Olaf M. "DNSSEC operational practices." (2006).

[16] Berners-Lee, Tim J. "The world-wide web." Computer Networks and ISDN Systems 25.4 (1992): 454-459.

[17] Booth, D. "WWWC: Web Service Architecture", W3C note, WWWC. W3. TR/wsarch (2004).

[18] IETF. "The Internet engineering task", www.ietf.org (2012).

[19] Alonso, G., Casati, F., Kuno, H., Machiraju, V. "Web services", Springer Berlin Heidelberg (2004).

[20] Sun, Sam, Larry Lannom, and Brian Boesch. "Handle system overview". RFC 3650, November, (2003).

[21] Harrison, Mark. "EPC information service (EPCIS)." Auto-ID Labs Research Workshop. (2004).

[22] Bushby, Steven T. "BACnet: a standard communication infrastructure for intelligent buildings." Automation in Construction 6.5 (1997): 529-540.

[23] Shelby, Zach, and Carsten Bormann. "6LoWPAN: The wireless embedded Internet". Vol. 43. Wiley, (2011).

[24] Jara, Antonio J., Miguel A. Zamora, and Antonio Skarmeta. "GLoWBAL IP: An adaptive and transparent IPv6 integration in the Internet of Things." Mobile Information Systems 8.3 (2012): 177-197.

[25] Shelby, Zach, Klaus Hartke, and Carsten Bormann. "Constrained application protocol (CoAP)" (2013).

[26] Kirstein, Peter T. "Early experiences with the Arpanet and Internet in the United Kingdom." Annals of the History of Computing, IEEE 21.1 (1999): 38-44.

[27] Jara, Antonio J., Socrates Varakliotis2, Antonio F. Skarmeta1 and Peter Kirstein. "Extending the Internet of Things to the Future Internet through IPv6 support." Mobile Information Systems (2013).

[28] Wikipedia. "Application Layer Gateway". http://en.wikipedia.org/wiki/Application_Layer_Gateway (2013).

[29] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G. J. and Lear. E. "RFC 1918: Address Allocation for Private Internet's", (1996).

[30] Egevang, Kjeld, and Paul Francis. "The IP network address translator (NAT)". RFC 1631, (1994).

[31] Rescorla, Eric, and Nagendra Modadugu. "Datagram Transport Layer Security Version 1.2." (2012).