ISSN 2349-4506
Impact Factor: 2.545

Global Journal of Engineering Science and Research Management

HOME AUTOMATION USING RESBERRYPI & CLOUD

Hemanth P R*
* PG Student, Department of Computer Network Engineering Cambridge Institute of Technology Bangalore, Karnataka, India

DOI: 10.5281/zenodo.55959

KEYWORDS: Internet of Things, Smart Home, Cloud Computing, Information Centric Networking, Respberry Pi.

ABSTRACT
The Cloud Computing provides native multicast support, content-based security, in network caching, and easy data access, which can be especially useful in the Internet of Things(IOT), and Resberry Pi is model to control the real time object.

In this paper, the attention is on the design of an ICN framework tailored to the smart home domain, considered as a major representative of IoT scenarios. The proposed solution encompasses the definition of a flexible and expressive naming scheme that supports data/command exchanges and configuration/ management operations, and also fits the common service models in the smart home domain (i.e., push, pull, multi-party). Use cases are provided to shed light on the system behavior and preliminarily assess its potential and performance.

INTRODUCTION
Home network is getting increasingly complex with the presence of application specific sensors, smart appliances, and smart networking devices such as residential gateway. Home automation today is being driven by several alliances [10] such as DLNA, ZigBee, and Z-Wave all aimed at supporting homenet services such as multimedia sharing, lighting, climate control, and energy management. The lack of inter-operability among these standards results in high cost, in flexibility, and inter-operability issues. The IETF home net (IETF-home) working group focusses on enabling an end-to-end IPv6 based homenet reusing existing protocols such as mDNS, DHCPv6, and OSPF to support features such as auto- configuration of IP interfaces, auto-discovery of services, and policy-based routing, with concerns in several areas including security, mobility, and content distribution.

At a high level, the objective of home networking is to allow efficient flow of information between service producers and consumers, both while inside or outside the home environment.

IoT and showed the main benefits and open issues related to the deployment of a high-level IoT architecture based on named contents. We pointed out that the main ICN. Building blocks (e.g., naming, forwarding, routing), originally Conceived for content dissemination in the Internet core, must be extended and customized to address the IoT traffic patterns and the presence of resource-constrained devices with littleto- none caching capabilities.

In this paper, the smart home has been chosen as a representative IoT use case, where monitoring and control functions need to be supported for a wide range of applications (e.g., user comfort/care, smart energy management) [1]. Almost the totality of protocols designed for smart home environments are based on proprietary solutions, thus sacrificing interoperability. Efforts for open, IP-based, standards are underway to ensure global access to services and information, also in home environments [2]. The 6LoWPAN stack [3] and the Constrained Application Protocol (CoAP) [4] are aimed, respectively, to support IPv6 in low-power and lossy networks (LLNs) and to extend the HyperText Transfer Protocol (HTTP) web service paradigm into LLNs. However, such solutions could inherit some of the drawbacks of IP when dealing with intermittent connectivity and challenged networks [5], [6]. At the same time, ICN holds great promise for building smart home management systems, because it simplifies network configuration, data retrieval, and service access, and also provides inherently security at the network layer [7], [8].

The Resberry pi model through an association by an accomplishment by an controlling the external devices. It runs an small power source though an home applications are maintained and can control and maintain.

In this paper contain how to access and control the smart home domain and its scenarios are given and concluded.

SMART HOME

A smart home is a very heterogeneous environment, characterized by several types of devices, different connectivity technologies, applications, and service patterns.

Considering some of the study,several challenges towards home automation: 1) high cost of ownership which includes installation and maintenance deterring any incremen- tal addition of new services; 2) inflexibility due to variety of standards and lack of inter-operability of devices leading to situation conceptualized in Fig. 2(a), where the services may have to hop between multiple protocols requiring gateways to handle adaptation functions; 3) poor manageability, due to complex realization of the home automation systems; 4) difficulty in achieving security due to lack of granular access policy enabling features offered by current systems; 5) remote home controlling through an Zigbee instrumentation.

A smart home is a very heterogeneous environment, characterized by several types of devices, different connectivity technologies, applications, and service patterns.

Local and global connectivity, security

The smart home may be locally connected but here using through Resberry pi smart home can be communicated locally are by global by connecting through an Cloud (e.g., Amazon E2 type of Service) Data delivered to remote consumer for further processing and decision.

Smart home applications may rely on highly sensitive information, which requires privacy, integrity and authentication support. The same data could be requested by different consumers which are located in different administrative domains; thus, a strong security support is a must to protect information independently from the channel/connection over which it travels. At the same time, EDs must be protected from external intrusions, which could create several malfunctioning, like denial-of-service (DoS) attacks.

Service models. Smart home applications exhibit different service models with specific quality constraints. Some applications can tolerate variable *delays* up to a few seconds, while others require real-time interactions. Without loss of generality, we can identify two basic service models, namely *pull* and *push*. A pull service captures a wide range of *(i)* control applications, where the execution of actions is required, and *(ii)* monitoring applications, where sensing information is required. Vice versa, a push service encompasses unsolicited

Smart Home Architecture

So far, a few very recent works have been proposed that leverage Cloud clients can give more solid, available and overhauled administrations to their customers thus. Cloud itself comprises of physical gear in the server farms of cloud suppliers. Virtualization is given on top of these physical machines. These virtual machines are given to the cloud clients. E.g. Amazon EC2 empowers the clients to hold low level points of interest where Google App-Engine gives an improvement stage to the designers to develop their applications. So the cloud administrations are partitioned into different sorts like Software as a Service, Platform as a Service or Infrastructure as a Service. These administrations are accessible in abundance of the Internet in the complete world where the cloud goes about as the single purpose of access for serve all clients.

Basic Storage Service (S3) for capacity of information. Clients can procure suitable sum CPU force, stockpiling, and memory with no forthright confirmation. Clients can control the whole programming stack from piece upwards. The engineering has two instruments one is the EC2 for registering purposes and S3 is for capacity purposes This data are made by the RASPBERRY PI by utilizing distinctive module and transmit it to the checking station where it stores the data in record and show it on graphical client interface (GUI) that is easy to understand. This outline give data continuously utilizing µc/OS-II

ISSN 2349-4506
Impact Factor: 2.545

Global Journal of Engineering Science and Research Management

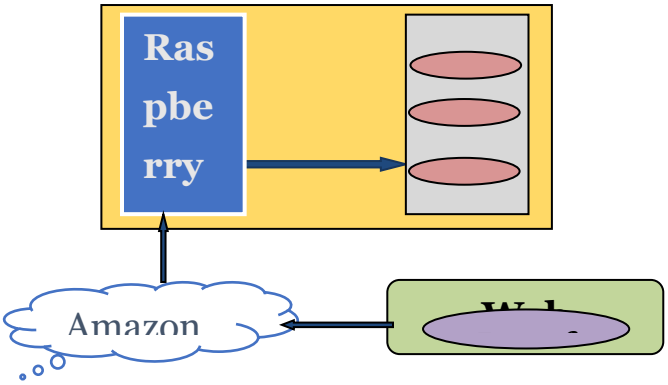

Fig.1 System Architecture

The model has to consume low power and process it faster with in sustain amount of time thus Resberry pi model is effective and low consumption of resources to operate .

The user are communicated in remotely by an inter face by an web browser in an remotely and this inter face is going through an web association of an amazon s3 server, Amazon server is used by an platform as an service through an association to compute the process and the possess data is transformed to an cloud network , the computing is done by an Resberry pi model it is run in an low power through an the module is convert an digital signals and compute with respect an output in realistic word (e.g., when an remote use want to check the states of an his home door by an this method he can check the states).

RESBERRY PI

The Broadcom SoC utilized as a part of the Raspberry Pi is relating to a chip utilized as a part of an old advanced mobile phone (Android or iPhone).While utilizing at 700 MHz as a matter of course, the Raspberry Pi gives a certifiable execution around comparable to the 0.041 GFLOPS On the CPU level the execution is like a 300 MHz Pentium II of 1997-1999.The GPU gives 1 Gpixel/s or 1.5 of representation preparing or 24 GFLOPS of regular reason processing execution. The illustrations capacities of the Raspberry Pi are around equal to the level of execution of the Xbox of 2001. The Raspberry Pi chip, working at 700 MHz as a matter of course, won't transform into sufficiently hot to require a warmth sink or uncommon cooling. The SoC is stacked underneath the RAM chip, so just its edge can be seen. On the prior beta model B loads up, 128 MB was designated by defaulting to the GPU, leaving 128 MB for the CPU. On the initial 256 MB discharge model B (and Model A), three distinct parts conceivable. The default split was 192 MB (CPU RAM), which must be adequate for standalone 1080p video unraveling, or for basic 3D, yet likely not for commonly together. 224 MB was for Linux just, with only a 1080p structure support, and was liable to fizzle for a few video or 3D. 128 MB was for overwhelming 3D, maybe additionally with video disentangling (e.g. XBMC).

RASPIAN OS: The Raspberry Pi above all else utilizes Linux piece based working frameworks. The ARM11 chip by the heart of the Pi depends on adaptation 6 of the ARM. The present arrivals of a few mainstream forms of Linux, including Ubuntu. Won't keep running on the ARM11. It is not prone to run Windows on the Raspberry Pi. Raspbian (suggested) – Maintained autonomously of the association; in light of the ARM hard-drift (armhf) Debian 7 "Wheezy" engineering port at first intended for ARMv7 and later processors (with Jazelle RCT/ThumbEE,VFPv3, and NEON SIMD expansions), ordered for the more constrained ARMv6 direction set of the Raspberry Pi. A minimum size of 4 GB SD card is required. There is a Pi Store for trading programs. Raspbian is a for nothing out of pocket working framework taking into account Debian streamlined for the Raspberry Pi equipment. A working framework is the arrangement of crucial projects and utilities that make your Raspberry Pi run. Then again, Raspbian gives more than an immaculate OS: it drops by method for more than 35,000 bundles, pre-accumulated programming packaged in a fine configuration for basic establishment on your The underlying form of more than 35,000 Raspbian bundles, upgraded for finest execution on the Raspberry Pi, was finished in June of 2012. Nonetheless, Raspbian is still in dynamic improvement with a significance on

enhancing the steadiness and execution of however many Debian bundles as could be expected under the circumstances.

HOME SCENARIOS FOR NAMING

The NDOMUS framework in Fig. 1 is based on the system architecture that we introduced in [9]. In this paper, the NDN-IoT functionalities, originally abstracted at a high level, are specified for the smart home domain.

The reference home scenario is depicted in Fig. 2, where the HS (for the sake of simplicity and without loss of generality, co-located with the HG) communicates with a set of EDs. All nodes implement the NDN building blocks designed for the conceived *Data* and *Management/Control* planes, but with different restrictions that depend on their role and resource availability. The Data plane is responsible for handling the individual packets, both Interest and Data, and operations on top of them (e.g., naming, caching, forwarding, retransmissions, security). while the Management/Control plane accounts for configuration and control functions onto the Data plane.

In the following, we focus on three main NDOMUS features: naming scheme (a *cross-plane* functionality strictly related to configuration and security operations), service model, and strategy for multi-party communications.

Naming scheme

In NDOMUS, application-specific and human readable names support both sensing/action and management/configuration operations in the house. For this purpose, we identify two sub-namespace classes: *(i) Configuration and management* namespace, identified by the prefix */conf*, used for home network initialization, configuration updates and management operations; and *(ii) Task* namespace, identified by the prefix */task*, used to identify and enable all the control and monitoring operations.

With such a classification in mind, we build a *name tree* to represent a highly flexible and extensible global namespace, which is also meant to easily support multi-party communications. The root node is a logical name component that uniquely identifies the house referred in the following as to */homeID*. It can be related to the geographic location of the house and/or to a owner identification number. According to the NDN routing protocol, the HS may announce the */homeID* prefix in the core network to advertise the availability of its smart home services and allow global reachability. The */homeID* prefix is followed by two main branches for */conf* and */task* sub-namespaces, under which different names can be composed, as explained below.

Configuration and management sub-namespace. Set-up operations in the home network, including EDs discovery and configuration, are identified by the prefix "*/homeID/conf/*". Similarly to the solution in [10], we assume that a configuration and authorization manager is in charge of registering the EDs by assigning the namespaces under which they can operate, together with the related keys and other information for security management (e.g., access control list, encryption mechanism). We assume that the manager is co-located in the HS, which maintains a list of active EDs and their parameters. EDs periodically broadcast *keep-alive* messages in Interest packets with a refresh rate of the order of some minutes or more. The HS removes a device from its EDs list if it does not hear from it for a certain interval and, eventually, notifies the owner. Any application (including the owner's one) that wants to access the home network to collect secure sensitive data or to trigger an action, must also interact with the HS to receive configuration instructions.

Task sub-namespace. Task names describe the type of operations to be carried out by EDs, such as measurement reporting or action execution. These names are used in both Interest and Data packets.

NDOMUS adopts the following basic name structure: */homeID/task/type/subtype/location/*, where the *type* component specifies the task type (*sensing* or *action*); the *subtype* component describes which specific sensing or action task must be performed (e.g., temperature sensing); and the *location* component identifies the physical position of the ED in the house. The proposed name structure can be easily extended by including new sub-task and locations.

Fig. 3 shows an easy example of task name tree, where light

ISSN 2349-4506
Impact Factor: 2.545

Global Journal of Engineering Science and Research Management

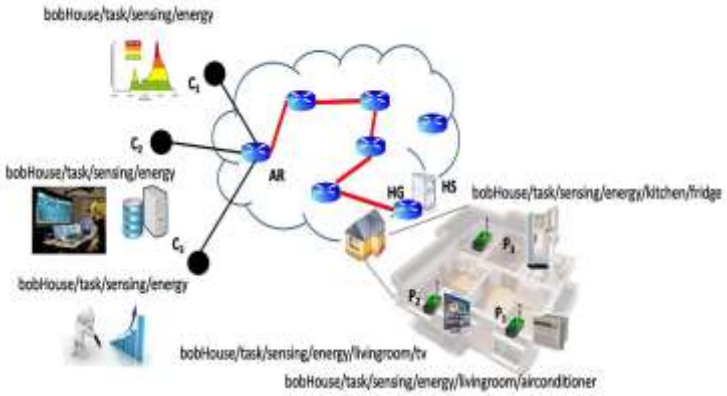

Fig. 2. Reference smart home scenario with several consumers and producers.

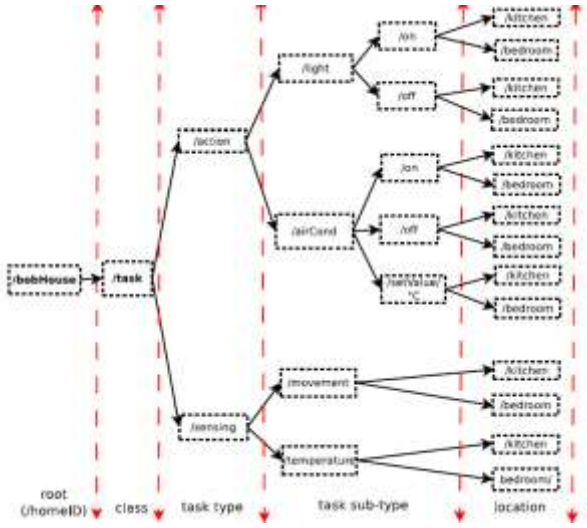

Fig. 3. Name tree example for task sub-namespace.

and air conditioning are associated as action subtype, while movement and temperature are considered as sensing subtype. An Interest with name */bobHouse/task/action/light/on/kitchen* is issued to require the kitchen light fixture to turn on. After the task execution, the ED can send a Data packet with the same name while the payload includes a Boolean value as a result of the operation. In case of failure, the payload can also include a specific explanation code, e.g., the bulb burnt out.

Although NDN potentially imposes no restrictions on the name structure, it is worth noticing that some IoT access technologies (e.g., IEEE 802.15.4) support small payloads. Thus, names should be maintained *thin* and preferably not exceed the length of the required content also to avoid packet fragmentation. To this aim, when HS and EDs agree upon the namespace in the initial configuration stage, they can also share a dictionary storing the mapping between long names and shorter name versions to be used locally.

Service model

From the service model perspective, smart home applications can be classified in three main categories: pull, *periodic* push, and *event-triggered* push. NDOMUS supports all the three models and provides Interest retransmission routines to guarantee reliable delivery, as defined by the strategy function.

Pull. This service model is naturally supported by NDN: the consumer application sends an Interest to ask for a measurement or an action, and waits for a Data packet from the ED containing the requested parameter value or the result of the triggered action. For security purposes, both Interests and Data are *authenticated* with proper

encryption mechanisms. For a guaranteed deliver, in case no Data are received within the Interest timeout expiration, the Interest is retransmitted by the consumer. In NDOMUS, the HS is in charge of autonomously retransmitting unsatisfied Interests which are originated by a remote application. The Data is processed by the HS and, eventually, cached and forwarded to the remote consumer, which shall use longer Interest retransmission timeouts.

Periodic pushing. Many home applications require measurements at fixed intervals, typically minutes (e.g., 15 minutes for energy consumption values). Data may be stored for longterm statistics and optimizations. This service can be modeled in two ways: *(i)* the HS periodically pulls the ED with an Interest; *(ii)* the ED autonomously sends an unsolicited packet at regular intervals. While the first strategy works according to the vanilla NDN pull logic, the second one apparently violates the NDN primitives because information is sent without any Interest solicitation. However, the second solution can be implemented by embedding data in the Interest itself; i.e., the ED sends the generated information, which is typically very short, as the last component of the global name. For this reason, the packet is re-named as *Interest Notification* [4]. After sending the notification, the ED waits for a dummy Data packet that is sent by the HS as an acknowledgement, otherwise it retransmits the notification.
3. Event-triggered pushing. The occurrence of some events, like the detection of the owner entering the house, can originate the unsolicited transmission of Data, e.g., for turning on the lights. If the event is an alarm, it must be reported with the highest priority and timeliness. A periodic pulling from the consumer application is therefore unfeasible in this context, since it could lead to intolerable delays and would waste resources. Vice versa, an *Interest Notification* is again the viable strategy.

Control Plane
Service routing: Name-based service routing can be established by publishing the services in a distributed routing control plane or resolution based on a centralized directory lookup mechanism. Considering the richness of ICN's forwarding plane, particularly CCN [11], the function of the routing control plane can extend beyond achieving reach ability. Routing can be extended to distribute service announcements network wide with policy restrictions: for e.g. ICN router proxying a smart grid subnet may choose to advertise its service only to authorized neighbor(s); or the HGw/IR could impose restricted service announcements in designated guest areas, allow site-scope announcement, or share it with PGw for Internet access.

Data Plane
Request/Response forwarding: ICN leverages both computing and storage resources in the routers to conduct intelligent information dissemination. Constructs such as embedding security in content PDU enables the feature of producing information once and consuming several times. Hierarchical naming can be leveraged to conduct efficient consumer request exploration of many sources, and content dissemination through multicast techniques at the same time. Though ICN professes PULL mode, PUSH mode can also be realized to support cases such as LLNs, where it is more efficient to notify sensor events rather than being polled periodically.

CONCLUSION

In this paper home applications.Through using an Raspberry bi module an home controlling using an Internet is achieved an effectively controlling the home appliances, The effective utilization of Cloud s3 resource and Respberry pi and Information centric networking framework that lays its foundations into the Named Data Networking instantiation, and is adequately conceived to enable typical operations and traffic patterns required in the smart home domain. Through this an home controlling is successfully leveraged.

REFERENCES

[1] M. A. Zamora-Izquierdo, J. Santa, and A. F. G´omez-Skarmeta, "An integral and networked home automation solution for indoor ambient intelligence," *Pervasive Computing, IEEE*, vol. 9, no. 4, pp. 66–77, 2010.
[2] O. Bergmann, K. T. Hillmann, and S. Gerdes, "A CoAP-Gateway for Smart Homes," in *IEEE ICNC* 2012, pp. 446–450.
[3] Z. Shelby and C. Bormann, *6LoWPAN: The Wireless Embedded Internet*. Wiley, 2009.

[4] Z. Shelby, K. Hartke, C, Bormann, and B. Frank, "Constrained Application Protocol (CoAP), draft," *RFC 7252, The Internet Engineering Task Force–IETF*, 2014.
[5] Z. Sheng, *et al.*,"A Survey on the IETF Protocol Suite for the Internet of Things: Standards, Challenges, and Opportunities," *Wireless Communications, IEEE*, vol. 20, no. 6, pp. 91–98, 2013.
[6] E. Baccelli, *et al.*, "Information Centric Networking in the IoT: Experiments with NDN in the Wild," in *ACM ICN*, 2014.E. H. Miller, "A note on reflector arrays," *IEEE Trans. Antennas Propagat*., to be published.
[7] W. Shang, *et al.*, "Securing Building Management Systems Using Named Data Networking," *IEEE Network*, 2014.
[8] R. Ravindran, *et al.*, "Information-Centric Networking based Homenet," in *IFIP/IEEE ManFI Workshop*, 2013.
[9] M. Amadeo, C. Campolo, A. Iera, and A. Molinaro, "Named Data Networking for IoT: an Architectural Perspective," in *IFIP EuCNC*, 2014.
[10] W. Shang, *et al.*, "Securing Instrumented Environments over Content Centric Networking: the Case of Lighting Control and NDN," in *IEEE* Infocom NOMEN Workshop, 2013.
[11] V. Jacobson et al. networking named content. In *Proceedings CoNEXT*, 2009.