A REVIEW OF IoT ENABLED NETWORKS' ARCHITECTURE AND ACCESS CONTROL

Lebogang Bopape
Durban University of Technology/South Africa / 21029737@dut4life.ac.za

Bakhe Nleya
Durban University of Technology/South Africa / bakhen@dut.ac.za

Richard Chidzonga
Mangosuthu University of Technology/South Africa / rcfoya@mut.ac.za

Abstract
Current IP and other networks such as Power Smart Grids are fast evolving, thus resulting in diverse connectivity challenges. This has led to the emergence of "the Internet of Things" (IoT) network transforming the legacy networks towards Device-to-Device (D-2-D) communication standards. By 2022, IoT will seamlessly interconnect the globe via intelligent devices and sensors of varying types and in the process large volumes of data will be generated. The resultant network structure will benefit mankind by helping in making complex decisions. In this paper, we overview both IoT enabled network architecture as well as and access and control framework for various services and applications. We commence with a review of a generalized IoT enabled network's security architecture as well as how the various elements constituting them interact. The paper then describes an access control framework applicable to the various would be applications and services.

Keywords: IP Network, Encryption, Federated Clouds, Information Security, Internet of Things, Smart Grid, Smart Objects.

Introduction

Currently, we are seeing a gradual shift in our conception of the traditional IP and related networks towards a universal and global network of "smart objects", now referred to as the Internet of Things (IoT). The continued acceleration of this paradigm shift is expected to peak by the year 2022 [1]. Key enabler to this acceleration is the significant fall in hardware costs, rapid advancement and development of enabling communication technologies as well as the current Internet's technological maturity. Ultimately, the goal is to interconnect all available physical objects and devices, thus enabling mobile and widespread access. An IoT networking concept can be broadly defined as facilitating networking as well as communication among various types of physical objects across the IP network. Humanity areas that stand to benefit include healthcare, agriculture, environmental monitoring, disaster areas, supply chain management, transport systems, smart homes and cities. For example, as at 2018, in excess of 2 billion people were connected to the IP network and thus can access various kinds of resources, e.g., content browsing, online gaming, exchange emails, as well as social networking. On the implementation side, the IoT capability is enabled by extending and blending ICT technologies and capabilities into common daily things and facilitating connectivity in extended Internet technologies. This has created a global cyber-physical system interconnecting all objects and enabling them to be controlled remotely.

The diverse heterogeneity in both the communication requirements as well as the hardware capabilities among the various types of devices will severely constrain transmission resource capabilities. At hardware level perspective, various objects have differing resource requirements, e.g. memory, power, computation, or communication capabilities. The various objects will also generally have varying Quality of Service (QoS) requirements in terms of resilience, reliability, data losses, latency or energy consumption constraints. As an example, it is not so critical that devices with power supply connection minimize the energy for computation/communication purposes, whereas that is a significant impacting constraint for battery-powered devices that do not have efficient energy replenishing or harvesting techniques [2]. These two contrasting characteristics intricate a universal network design that can satisfy both the general diversity of functionalities of things as well as capabilities. It is for this reason that adaptive cross-layer communication schemes are being pursued instead.

Whereas there exist quite many cross-layer protocols for various wireless networks such as sensor(WSNs), mesh (WMNs), and Ad-Hoc (AHNs) [3], [4], these however cannot be directly integrated or applied to the envisaged IoT enabled networks for various reasons such as:-

- Typical IoT enabled networks comprise both centralized as well as hierarchical architectures which they inherit from IP networks, whereas on the other hand AHNs, WSNs and WMN networks have rather flat network architectures, in which devices link and communicate in a hopping manner without the involvement of core Internet.
- In WSNs, nodes normally will have a shared goal, thus similar hardware specifications and common communication protocols whereas IoT enabled networks' devices and things are highly heterogeneous in terms of QoS requirements, hardware capabilities, functionalities as well as individual goals.

Addressing privacy as well as security challenges in IoT enabled networks is also of paramount importance. However, it is a challenge to do so as the vast numbers as well as diversity of “emerging things", heterogeneity, and dynamic changes in IoT environments complicates that task. Conventional security controls normally can only be confined to a domain i.e., monitoring a specified infrastructure unit and safeguarding a particular service, such as access control. However, the IoT enabled networks accommodate lots of resource-constrained things e.g. body sensors and thus from a design point of view, it may not be practical to directly implement current security and access control measures. These controls are also generally platform-specific and would not be cost effective or generally feasible to implement them in a multi-vendor/multi-domain heterogeneous space such as the IoT networks.

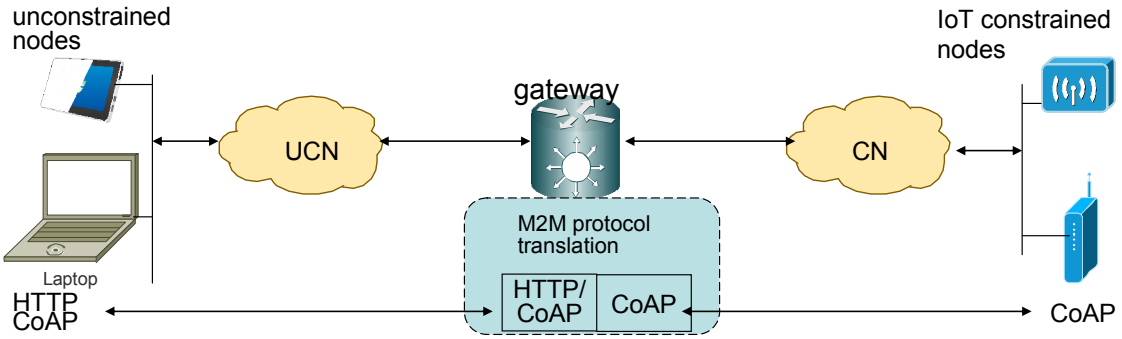


Fig. 1: IoT heterogeneous space [5]

Overall effective security and privacy in IoT enabled networks should generally satisfy basic criteria such as confidentiality, authorization, authentication, availability as well as integrity.

With regards to security concerns, the current focus is on developing D2D networking protocols rather than applying the existing M2M/IP communication security ones as this complicates characteristics and deployment environments.

- *Front-end Sensors and Equipment:* This acquires data directly from smart sensors before relaying to authorized central processing systems via D2D modules. The current set up compromises their security as they are not monitored.
- *Network Denial of Service (DoS)*: IoT enabled networks directly facilitate overall D2D communication coordination/management and renderable QoS and may often be susceptible to DoS attacks. A DoS will occur when several systems flood the resources (e.g. bandwidth) of a targeted system. Such an attack is often the result of multiple security-compromised systems concurrently flooding a targeted system (such as a webserver) with traffic.
- *Back-end*: Back-end is an integral part of an IoT enabled network system that provisions the required security/as well as efficient sensor management functionalities and data analysis to facilitate expeditious data processing capabilities. A key IoT security framework comprises seven elements namely: communication security, access control, user authentication, privacy protection, data integrity as well as data confidentiality and its availability whenever needed.

Currently, setting up secure channels within an unconstrained network (UCN) domain is possible using existing security protocols such as IPsec [5]. These however cannot connect directly with the constrained network (CN) nodes due to the mismatches in processing resources requirements. One possibility is to offload computationally intensive tasks and instead delegate them to a trusted neighbouring UCN. Intermediary IoT Gateways will adapt the communications between the UCN and CN domains (Fig. 1).

Ensuring privacy is also quite important. In general, in a distributed network of sensor devices, the acquired data must be relayed via fixed or mobile communication. It is imperative that privacy be preserved throughout a wireless communication scenario, i.e. while the data is in the sensor device, during communication in storage, as well as during actual processing.

- *Privacy in Device*: The unsecured device may have vulnerabilities such that confidential information it has acquired can be siphoned out and diverted elsewhere. It is therefore important that sensors (devices) that gather sensitive data be reliable and robust in this regard.
- *Privacy during Communication*: Utilizing a set of secured communication protocols would ensure privacy during communication.
- *Privacy in Storage*: Information stored in storage devices such as hard drives ought to be secured. Enforcing pseudonymization and anonymization in accessing this data may be an appropriate approach.
- *Privacy at Processing*: The data must be treated with privacy at this stage. To prevent the data from being leaked to unauthorized, Digital Right Management (DRM) techniques can be considered in which illegal use and re-distribution can be avoided.

Devices, Elements and Architecture

ITU's ITU-T Y.2060 Recommendation does provide an overview of the IoT's concept and scope, identifies its key fundamental characteristics and high-level requirements, as well as describes the IoT reference model. It defines an IoT network as a global infrastructure for the information society enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies [6]. As defined in [6], a fully-fledged IoT is envisaged to be a "dynamic as well as universal network enabling interoperable networking protocols where both virtual and physical objects can communicate.
Fig. 2 summarizes IoT devices and components.

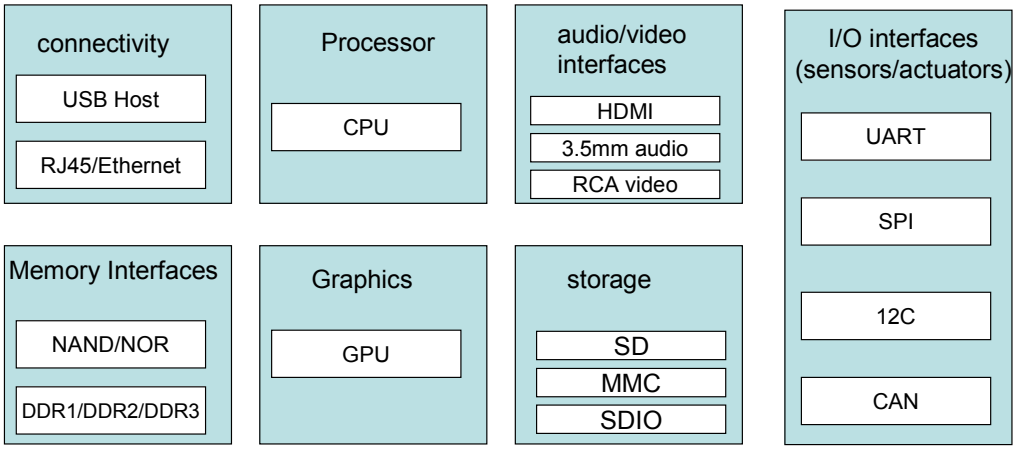


Fig. 2: IoT devices and components

Various utilities are facilitated by several functional architectural elements comprising the system. Example utilities include: -

- *Remote sensing*, in which devices are responsible for the acquisitioning of data which is in turn processed and extracted useful information used to guide on future actions locally or remotely.
- *Self-adapting utilities*: The interconnected elements within the IoT domain are provisioned with capabilities of dynamically adapting to changing contexts as well as expected to realign their actions based on the changed/varied operating conditions, sensed environment or user's context.
- *Self-configuring*: Self-configuring capability is enabled among the devices thus allowing a group of IoT devices to operate harmoniously to provide certain desired functionalities (such as climatic conditions, or weather monitoring). The same devices should be able to self-configure and perform any necessary software upgrades.
- *Interoperable protocols*: IoT objects and entities facilitate as well as leverage a diverse set of interoperable networking protocols and thus be able to communicate with other devices as well as the existing infrastructure.
- *Self-Integrating*: This is to enable IoT devices to integrate themselves automatically with other devices onto a particular information network such that they harmoniously network with other systems as well as devices. Normally, they are discoverable by peers, after which they provide self-description to their new peer devices or user applications. An example is when an individual weather forecasting device describes its

capabilities to neighbouring connected nodes hence collectively and collaboratively, they can provide "smarter" weather predictions.

- *Unique identity*: Every physical or virtual device is assigned a unique identity as well as an identifier. These elements may also be coupled/provisioned with context -adaptable interfaces that also facilitate remote querying, monitoring and control of associated devices.

Key functional architectural elements include communication, services, security, management as well as application.

- *Communication*: This block facilitates communication among the various devices and components.
- *Services*: An IoT system provides a diverse set of functions such as services for the purpose of facilitating device control and modelling data analysis and publishing as well as device discovery.
- *Management*: This is key to ensuring various functions to govern an IoT.
- *Security*: This will generally provide security to the IoT enabled system by provisioning security related services and functions, e.g authorization, authentication, message integrity, privacy, data security as well as content integrity.
- *Application:* This interface directly with users, thus provisioning the necessary modules for monitoring and controlling of various aspects of the IoT system(s).

Currently no standardized architecture for IoT enabled works as well as the number of layer functionalities. However, most of the proposed models commonly define the following layers: -.*Physical* (*perception) layer:* This layer comprises sensor devices and objects for acquiring information about the vicinity environment.

- *Network layer:* This layer facilitates interconnecting other smart things, network devices, and servers within the IoT.
- *Transport layer:* This layer ensures process to delivery of data.
- *Application layer:* This layer defines the various services and as well delivers application specific services to end users or systems.
- *Processing layer*. It is responsible for processing data after the transport layer.
- *Enterprise /Business layer:* This layer generally regulates the entire IoT operations, including business and profit models as well as security.

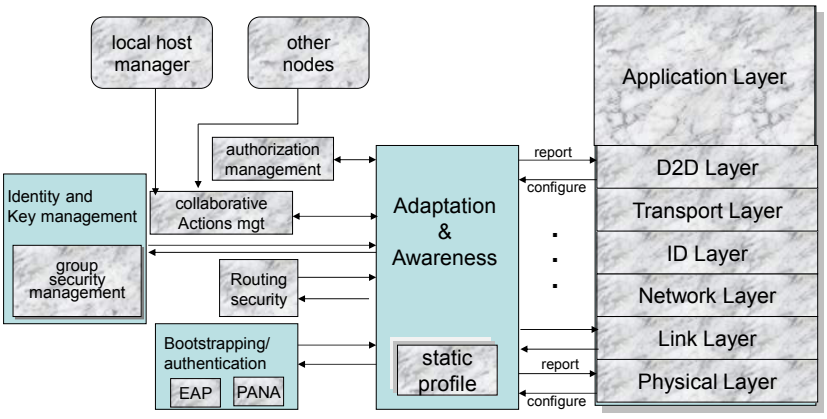


Fig. 3: IoT Generalized Secured Communications Architecture

An example architecture is illustrated in Fig. 3, [5], [6]. It basically incorporates security blocks on the left-hand side. A D2D service layer just below the applications layer facilitates communication between different network elements thus overcoming lack of interoperability of legacy and current machine-to-machine (M2M) technologies [7],[8].

Transport Layer's functions are dependent on required QoS, but generally provides end-to-end delivery as well as performance guarantees between communicating endpoints.

Identification (ID) layer's main function is to carry out resource's identification. Decoupling this functionality from the network layer (traditional IP network) assists in enhancing security by making it possible to implement authentication service based on the node ID. A protocol such as the Host Identity Protocol (HIP) can be applied at this layer.

Network (NET) layer has the IPng layer as its main routing protocol that takes care of node to node addressing as well as packet routing.

MAC layer governs usage of channel resources. In so doing it minimizes contentions, that otherwise might ultimately degrade performance at this access layer [9].

Physical Layer (PHY) addresses the physical specifications of the data associated signals, e.g. it deals with channel coding, modulation/ demodulation as well as transmission over a specified medium.

Security is addressed by the security blocks such as:

Bootstrapping and Authentication regulates the addition of new nodes to the network. The Authentication service is utilized by each node, when joining a new network, typically after mobility. It relies on access protocols such the Extensible Authentication Protocol (EAP) and the Protocol for carrying Authentication for Network Access (PANA)[10], [11]. The latter is also utilized to ensure improved interoperability.

Static Profile shares its own specifications with each endpoint, e.g, its power source size, storage capacity, processing power, desired security profile/preference. Typically, they mutually agree on a cryptographic suite during the negotiation phase.

Collaborative Actions Management renders assistance to **a** resource constrained IoT node that suddenly cannot cope up with certain tasks, e.g. computations, hence seeking assistance from a *trusted entity* within the neighbourhood's *constrained* network topology to recommend possible assisting peers.

Identity and Key Management block guarantees object or device privacy by choosing a unique ID for data exchange sessions as well as ensuring and provisioning total privacy during the communications session through the use of robust encryption .

Adaptation and Awareness block gathers information about an IoT node, as well as configuring the necessary protocol(s).

Group Security Management provisions and enforces multicast-related privacy at the Network layer.

Routing Security block guards against possible classical routing attacks. It does that in conjunction with the Local Trust Manager and as well as with the Bootstrapping and Authentication modules.

Authorization Management (AuthZ Mgt.) regulates access to resources and other related services. It will liaise with relevant Authorization infrastructure to retrieve trust certificates for accessing any resources as well as verifications on whether access can be granted without certificates.

Clouds of Things

This is a platform for rapidly provisioning a set of pooled configurable computing resources by means of an enabling, on-demand network access in IoT enabled networks, [12], [13]. This is illustrated in Fig. 4.

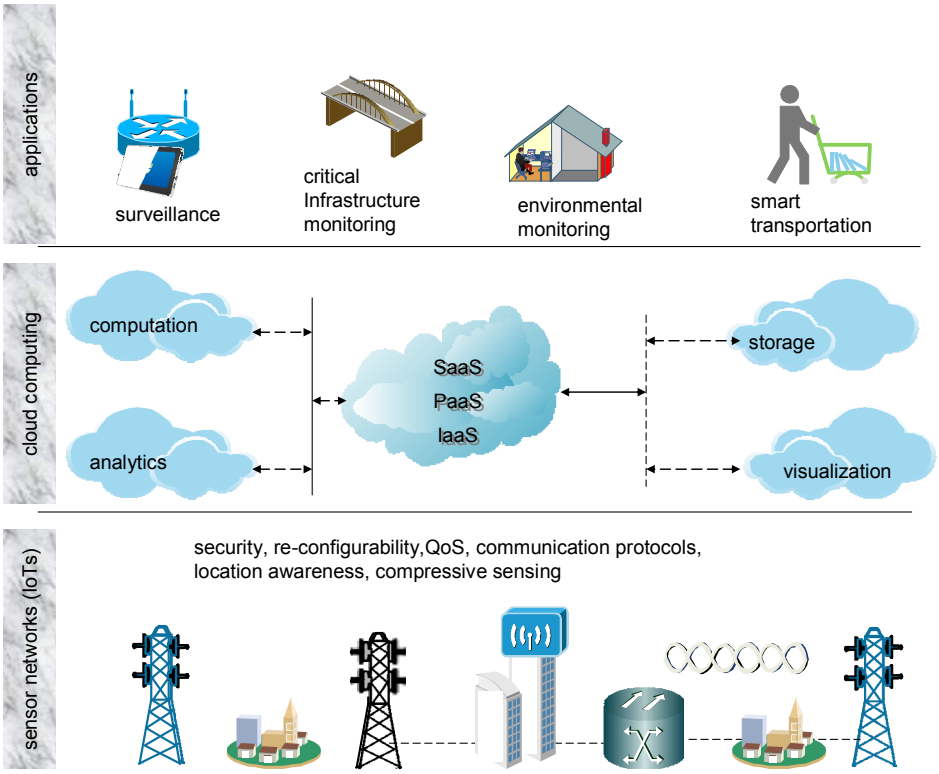


Fig. 4: Cloud as a middleware in IoT paradigm

Typical cloud computing characteristics are: -

On-Demand self-service: i.e ,the ability to render users instantaneous access, to computing resources requirements (e.g. CPU time, storage space, network access etc.) without requiring any human interaction with the provider of those resources.

Network Access: Such requested resources are deliverable through the IoT enabled network and accessible to several clients as well as client applications with diverse platforms requiring standard protocols and mechanisms to access them.

Resource Pooling: The available resources are pooled together to serve many customers concurrently utilizing various dynamical assigned physical and virtual resources so as to satisfy customers' QoS expectations. This "multitenancy" model relies on the use of virtualization and in that way, IT resources can be dynamically assigned and reassigned, according to demands.

Rapid Elasticity: The service provisioned by cloud provider elastically deployed, assigned, released or scaled as per demand.

Measured Service: The ability of the cloud service to monitor and measure actual individual usage and charge fairly.

In terms of infrastructural deployment within the IoT context, four models exist, and these are [14]:

Private Clouds: This infrastructure is provisioned to an individual organization so that it restricts access and usage of the services it avails employees.
Community Cloud: This is an infrastructure to a community who share a common goal *Public Cloud*: Such an infrastructure's services are provisioned for open use on a pay-per-use model.

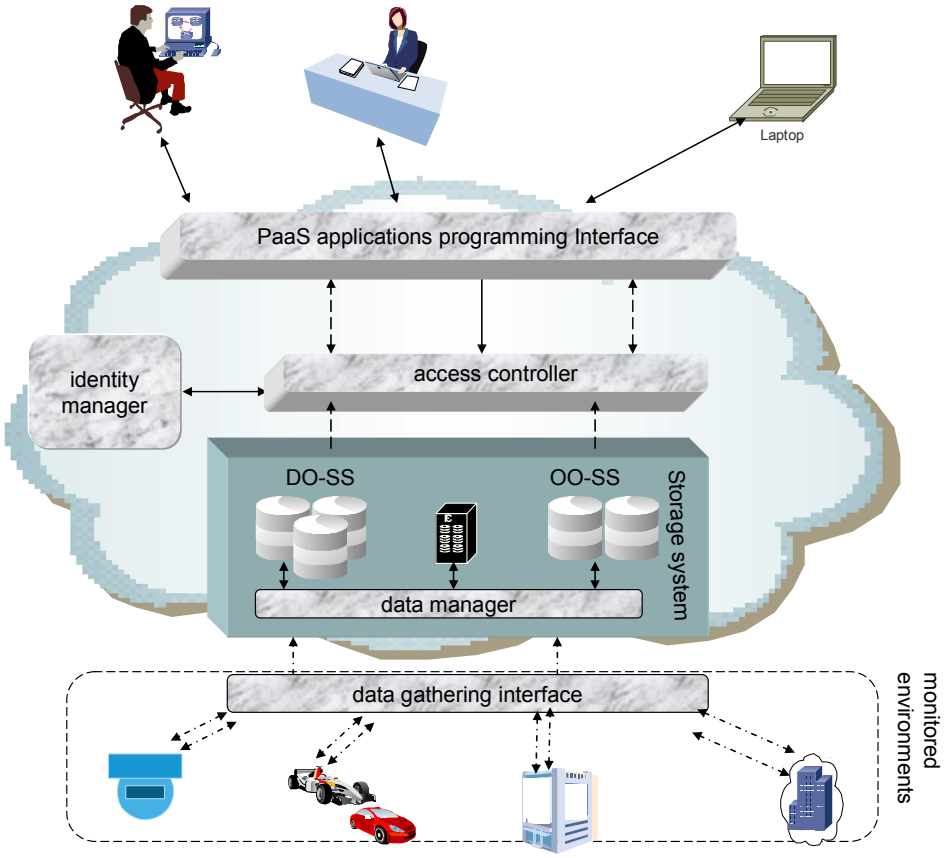


Fig. 5: Cloud Storage System

Hybrid Cloud: In this case the infrastructure blends two or more distinct infrastructure deployment models.
Inter-Clouds (Cloud Federations): This is a relatively newer cloud provisioning model that offers more flexibility, as well as improved reliability and a geographic distribution.
Depending on cloud services that are renderable by cloud providers, three service models are specified. These differ on control granted to requested resources by a user as well as, the general functionalities and the architectural layer offered.

Software-as-a-Service (SaaS): In this case the users *rent out their applications via a service provider.*
Platform-as-a-Service (PaaS): This is primarily a development platform that is provisioned to customers to develop their proper applications or services.
Infrastructure-as-a-Service (IaaS): The users are allowed direct usage of the IoT infrastructure. This include processing, storage and network resources. In practice, this is implementable through virtualization techniques.

The convergence between Cloud Computing and IoT has led to the "Cloud of Things" or CloudIoT.
In the advent of IoT, storing data locally and temporarily will not be feasible anymore as more storage space would be required. In any case, most of the data would require processing externally (in the Clouds) where there are better, efficient and more capable computing resources.
Primarily, IoT services are provided as isolated vertical solution in which a given application and related components are tightly coupled to the specific context of application. Coagulating and rendering IoT services via the Cloud will ease the delivery and the deployment of them by leveraging all the flexibility of Cloud models. In this regard, the Cloud computing facilitates applications development and makes possible an abstract vision of the IoT systems.
IoT can also provide a platform for the Smart Cities services that are envisaged in the next 5-10 years.

Related Alliances, Organisations and Standards

A: Key IoT Related Organisations

Key Organizations related to IoT development and deployment activities include [15]:

- *The European Telecommunications Standards Institute (ETSI)* focusing on connecting "Things" as well as clustering them.
- *The Internet Engineering Task Force (IETF)*: This is the current Internet's leading standards setting body that has since set up an additional IoT Directorate Group that is spearheading and coordinating related efforts in reviewing specifications for consistency, and monitoring IoT-related matters.
- *The Institute of Electrical and Electronics Engineers (IEEE)* focuses on IoT related innovations as well as specifications.
- *Object Management Group (OMG)* focuses on Data Distribution Service Portal;
- *The Organization for the Advancement of Structured Information Standards (OASIS*) whose MQTT Technical Committee spearhead IoT related issues;
- *Open Geospatial Consortium (OGC)* focusing on Sensor Web for IoT Standards Working Group;
- *The European Lighthouse Integrated Project addressing the IoT Architecture (IoT-A*) which focuses on the formulation of a standardized protocol/architectural reference model for the IoT.
- *One_M2M*, which proposes a single or one M2M and hence are also focusing on developing technical specifications for a universally standardized M-2-M Service Layer whose compatibility with various hardware and software enables reliable interconnection of all devices with M2M application servers globally.
- *Open Standards IoT (OSIoT*,) whose focus is on developing and promoting free open source standards.
- *Eclipse Paho Project*: This is an organization that focuses on the overall integration of D-2-D/M2M applications.
- *OpenWSN*: This is a platform as well as repository for open-source implementations of protocol stacks based on IoT standards.
- *CASAGRAS*: An initiative by Europe, the USA, China, Japan and Korea that addresses universal standards, concerning RFID and its overall role in realizing an IoT.

B: Alliances

These include [14], [15]:

- *The AllSeen Alliance*: which is focusing towards enabling and spearheading universal adoption of IoT related devices, systems and products through an open, universal development framework. The AllSeen Alliance is in the process of merging with the Open Connectivity Foundation (OCF) and the merged consortium will retain the OCF name. Overall the merged Alliance will focus on a code base of diverse and various modular applications and services that facilitate critical activities such as pairing and discovery of neighboring objects and devices, message routing, and security. The cross-platform nature of the open source codebase facilitates interoperability among diverse as well as basic objects and systems.
- *IP for Smart Objects Alliance* (IPSO) – The IPSO Alliance is an open, forum comprising several organizations and individuals that promote the value of using the Internet Protocol for the networking of Smart Objects.

Its R&D efforts are geared towards achieving IoT interoperability by facilitating data metadata exchanges effortlessly, i.e. this is an approach that eradicates the need for translators. The new approach universally defines all objects and devices , so that each no longer requires predefining nor preregistering. Overall, it emphasizes as well as advocates for IP networked devices in healthcare, energy, consumer and industrial applications.

- *Wi-SUN Alliance*: It promotes the use of IEEE's 802.15.4g based interoperability protocol standard to advance seamless connectivity. Primarily, the Wi-SUN Alliance promotes open industry standards for: 1. Wireless Smart Ubiquitous Networks and related applications. 2. Advancement, standardization as well as interoperability of wireless Smart Ubiquitous Networks globally. 3. Other activities include user education, industry outreach and other support programs as well as lobbying regional regulatory bodies for spectrum allocation for smart grid services.

C: Protocols

Broadly, IoT candidate protocols can be categorized as: Infrastructural, Identification, Communications & Transport) Service Discovery, Data Protocols, Device Management and Semantic (security).

Infrastructure Protocols

- *IPv6 over Low Power Wireless Personal Area Networks (6LoWPAN).* It is an adaptation layer protocol for IPv6 over IEEE802.15.4 links.
- *Nano Internet Protocol (NanoIP):* This is a concept that seeks to bring IP-like networking services to embed with sensor devices, by secluding the TCP/IP overheads.

Discovery Protocols

- *Multicast Domain Name System (mDNS*) - Can resolve and map device names to global IP addresses.
- *Universal Plug and Play (UPnP)* - This category of protocols facilitates self-discovery and interaction capabilities by networked sensors and devices.

Data Protocols

- *MQTT For Sensor Networks (MQTT-SN)*: An open protocol designed specifically for mobile and M2M/D-2-Dapplications.
- *Constrained Application Protocol (CoAP):* An application layer protocol for WSN nodes.

Communication / Transport layer

- *IEEE 802.15.4*: This is a standard which specifies the physical layer and media access control for low-rate wireless personal area networks (LR-WPANs).
- *ANT*: A wireless sensor network technology- designed for collection and transfer of sensor data and the integration of remote control systems such as controlling indoor lighting or a television set.
- *LoRaWAN:* Network protocol intended for wireless battery-operated devices.

Semantic

- *SensorML*: It is an approved Open Geospatial Consortium standard. That primarily rovides standard models and an XML encoding for describing sensors and measurement processes.
- *Media Types for Sensor Markup Language (SENML)*: A simple sensor, such as a temperature sensor, could use this media type in protocols such as HTTP or CoAP to transport the measurements of the sensor or to be configured.

Security

- *Open Trust Protocol (OTrP*) - This protocol essentially is designed to enhance and manage security configurations in Trusted Execution Environments (TEEs). It aims at creating an open universal protocol defining how objects and devices trust each other in a networked environment. It uses the Public Key Infrastructure architecture (PKI), and certificate authorities, as its basic underlying system..
- *X.509* - Standard for managing digital certificates and public-key encryption.

Access Control in Multi-domain Federated Clouds

In this section, we describe a possible access control in Federated IoT Clouds. A federated cloud system is illustrated as shown in Fig. 6.

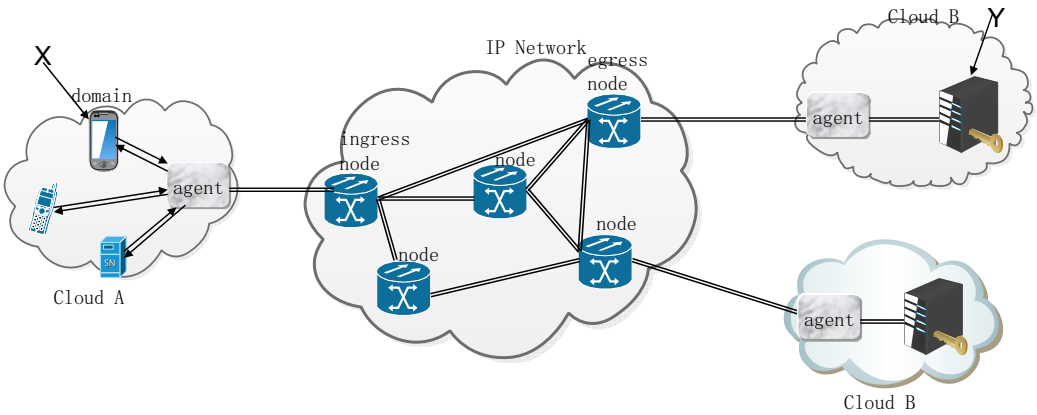


Fig. 6: Federated Cloud Access Control architecture.

Usually it suffices to have an individual user authenticated in a single domain. It is recalled that IoT enabled networks in general will be characterized by relatively dynamic nodes connectivity as well as network topologies. Because wireless channels are dynamic in nature, there is a need to accordingly incorporate a suitable flexible as well as dynamic access control system that is suitable for the federated Cloud IoT environment.

A: Access Control Architecture

We propose an access control architecture as illustrated in Fig. 7 and was partly modified from a proposal in [17]. Each domain has an Agent Unit (AU) to which all devices and components are connected. The domain is also connected to the IP backbone network. Features characterizing the architecture include authentication for each user's access request (s) as well as a QoS secure path selection.
The authentication network is decentralized and hence each domain handles authentication requests from all its devices and components. High bandwidth end-to-end authentication channels are logically separated from encrypted and QoS ensured data channels.

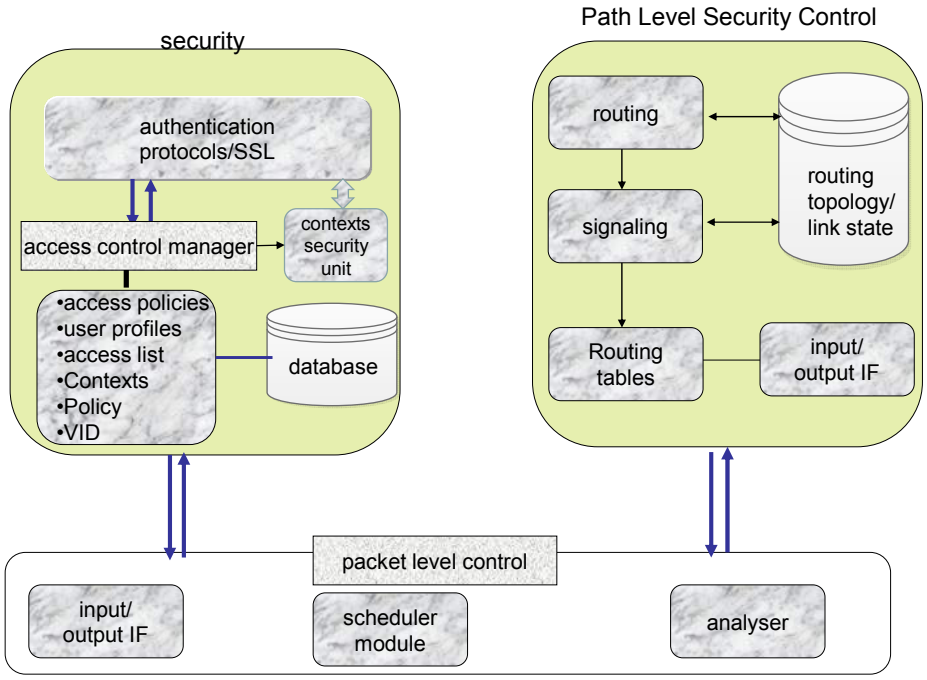


Fig. 7: Proposed Access Control Architecture and Agent Unit functionalities.

The packet level control consists of an input/output interface. Upon reception of a packet, the analyzer unit differentiates *access request*, *data* or *control* packets by studying the service identifier field (ID). Any received authentication packet is passed on to the Security Block (Fig. 7).

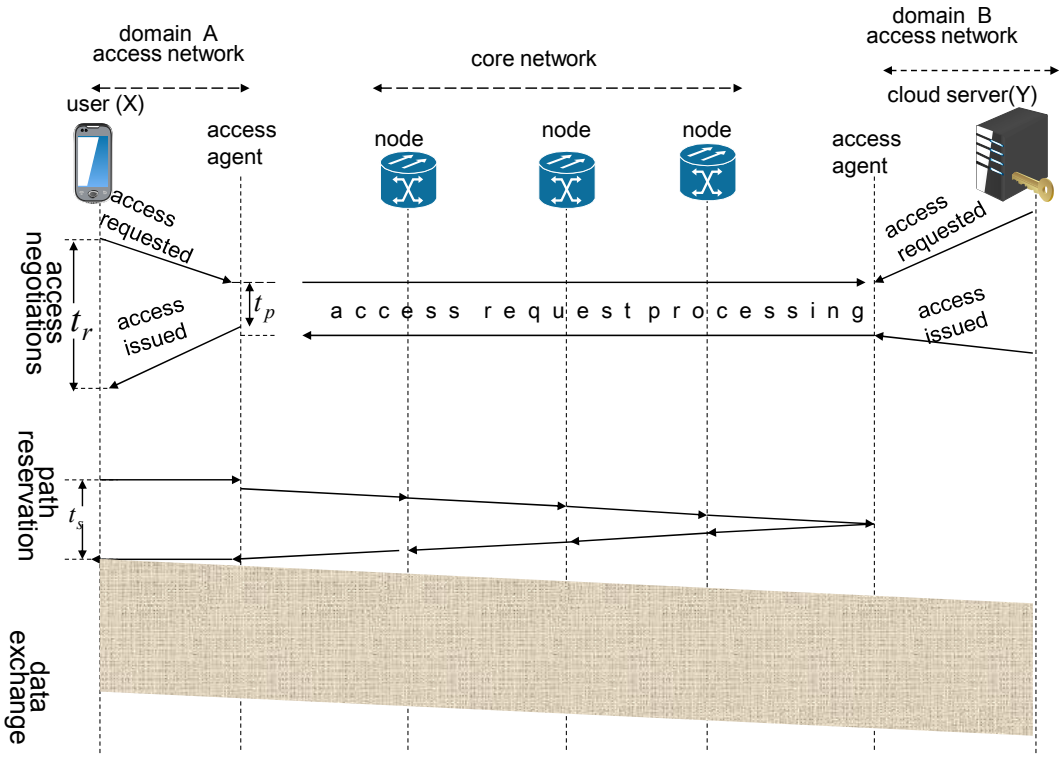


Fig. 8: Message exchanges in authentication, channel reservation and data exchange processes.

Within this block, the Access Control manager (ACM) together with a set of authentication/SSL protocols [18] will negotiate for the desired access to a requested resource under the coordination of the contexts security unit (CSU). After the access is granted, an authentication notification in the form of a ticket is issued to the user. In a way, the CSU is a central point for security decisions. The control packet will also be utilized by the Path Level Security Control block in setting up an encrypted path between the ingress and egress nodes. In so doing it uses the routing topology/link state database. The path selection is based on random routing shortest path first. The explicit route information i.e. the set of nodes to be traversed as well as required resources is now incorporated in the signal from the ingress node to the egress node over a secure and dedicated control/signalling channel and ultimately in the process reserving the requested secure path between the Agents. A summary message exchanges in authentication, channel reservation and data exchange processes is illustrated in Fig. 8.

B: *Contexts Access Control Details*

In the Agent generalized architecture illustrated in Fig. 7, the Security Block receives *the access request* and ultimately decides to accede to it or not [18]. The ACM basically provides several sets of primitives such as:
Policies: This is a repository comprising a set of various access policies for accessing available objects or resources. We distinguish; (1) user-based policies, which primarily comprise sets of user profiles ^{P} and rules versus (2) subject-based policies each comprising a set of objects (^{O}) and rules. Note that an object can be defined as a lone tag ID or as a sequence applicable to a set of tags e.g.;

$$O = \{VO_{i,1,}...VO_{i,n}\} \quad (1)$$

where, VO denotes value of an object as we assume that access control data structure in this regard will generally be based upon element (E) → *Attribute (A)* → *Value(V)* ternary relationships.

$Policy \in \{P, C, AR\}$, where P is the user profile, C is the context and AR is the type of access rights, such as save, read, write, copy, etc.

User Profile: This is defined here, as an attributes-based set of user profiles. The attributes are specified by the administrator. If P_i defines the profile of user i, then each P_i comprises several profile attributes (AP):

$$AP_i = \{AP_{i,1},...AP_{i,n}\}. \quad (2)$$

Access List; As proposed in [19], [20], [21], an access list supports access request when access request-based authority delegation is requested and executed.

Contexts; Scalability issues due to the dynamic nature of IoTs as well as large numbers of devices and users, it is further necessary to further enhance access control by way of incorporating contexts (C) that define virtual identity (VID) and a set of contexts (C_{Set}) with different types (C_{Type}). Each C_{Type} is assigned a constraint (C_{const}) hence:

$$C_{Type} \in \{trust\_level, auth\_level, location,, time,...\} \quad (3)$$

$$C_{Set} = \{C_{Type(1),}C_{Type(2)},...,C_{Type(n)}\} \quad (4)$$

$$C_{const} = \langle C_{Type} \rangle \langle OP \rangle \langle value \rangle \quad (5)$$

where $value$ is a specified value of C_{Type} and OP a logical operator, thus giving:

$$C = \{C_{const(1)}, C_{const(2),,...,}C_{const(n)}\} \quad (6)$$

VID: This is a set comprising, the user ID, contexts, subject policies and a set of disclosure policies;

$$VID = \{ID, P, C, Policies\} \quad (7)$$

C: Access Request Initiation

An access request to an Object (O) generally specified by profile set P, and an Electronic Product Code Information System (EPCIS) set [21], event types ET each with its own attributes e.g. (AE_i) and context C. In order to validate a request, its profile, (P^{REQ}) and requested object policy's profile should match.

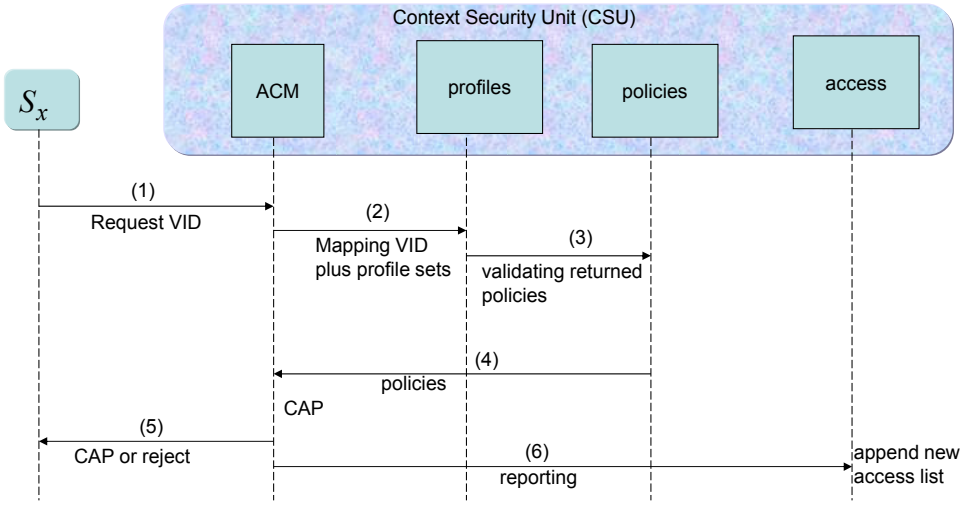


Fig. 9: Access Control initiation/creation.

If the number of matching access rules is p , then;

$$C^{REQ} \cap \left(C_1 \cup .... \cup C_p\right) \tag{8}$$

where;

$$C_i = \left\{AE_{i,j}^S \wedge ... \wedge AE_{i,n}^S\right\} \tag{9}$$

Specifies the ith matched access rule conditions.

$$R^{REQ} = \left\{AE_k^P, ..., AE_m^P\right\} \tag{10}$$

The process commences by device S_x sending its virtual identifiers ($VIDs$) together with those of the desired object (VID_o) to be accessed to the AGENT. This is illustrated in Fig. 9.

1. The AGENT will respond by requesting the ACM to map as well as furnish back the profiles of S_x based on the supplied VIDs. The ACM will in turn furnishing back the requested values, i.e, profile P as well as context, C of S_x .
2. Both P and C are passed on to the Policies Repository, for validation against relevant policies of the corresponding object O, (based on its (VID_o) .It will in turn pass them on to the ACM.
3. The ACM processes and validates the received P and C before creating a new capability of the object (CAP) and sends to S_x
4. The ACM notifies the Access Control Servers (ACU s) about the successful creation of an Object for subject S_x , it also starts creating a new propagation tree.

D: Access Provisioning

Access provisioning relies on $_{ext}CAP$ validation as well as evaluation of its contained $contexts$, C and this is carried out as follows:

1. Upon receipt of the $_{ext}CAP_x$ as well as VID_s from the device, S_x requesting access, the ACU checks their validity/authenticity. It does so by scripting a one-way function $f(S_x, O, AR, \Re nd_o)$ and comparing the output result with $\Re nd_i$ in $_{ext}CAP_x$.

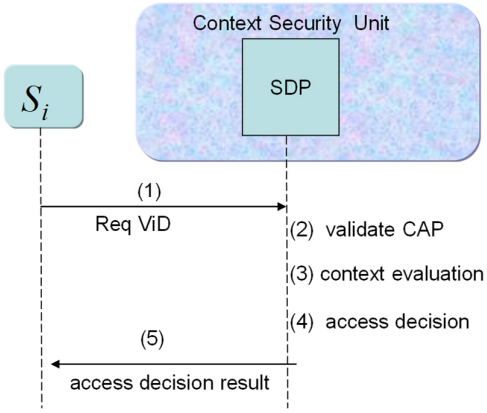


Fig. 10: Access control

2. If $_{ext}CAP_x$ is successfully validated, the ACU shall now validate the contexts C_{const} contained in C and if the result is $true$, the access request response is acknowledged to S_x.

The various steps are summarized in Fig. 10.

E: External Access delegations

For a federation network in IoT, trust relationships among the multiple domains are established prior to implementing authority delegation. This will allow all federation domains (members) to mutually authenticate each other.

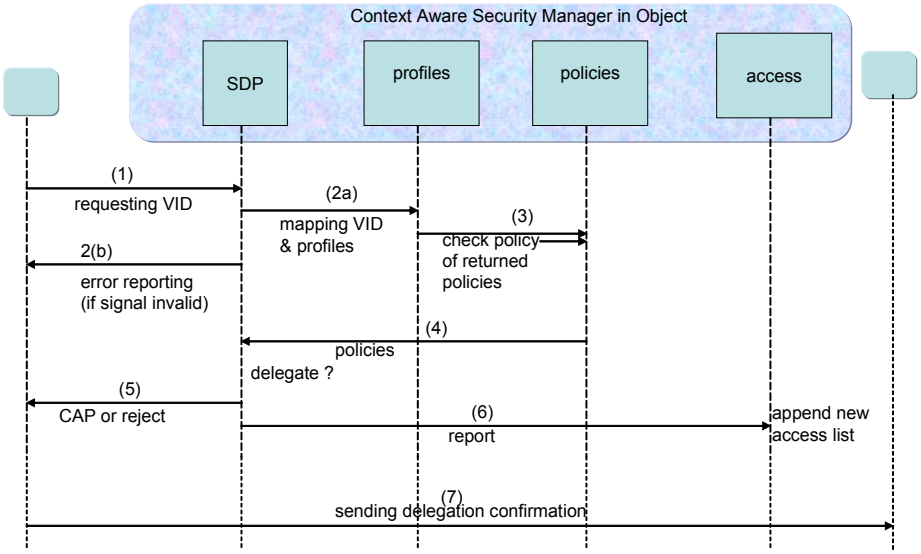


Fig. 11: External authority delegation.

In practice, a device that already has access rights can delegate the Object (O) to access the required resource(s). Summarily O upon receiving a delegation request from S_x and validates it before responding back to S_x with an external capability ${}_{ext}CAP_D$ together with **D**'s identity. In turn, S_x now dispatches the ${}_{ext}CAP_D$ concurrently with a public key known to D**.**

The sequence of events are as follows:

1. S_x sends a delegation request to the ACM within O. The request message that includes a VID_D is signed with a Federated IoT certificate which O uses to verify that the request is indeed from S_x.
2. The ACM validates the message's signature and if successful, it requests VID_D mapping from the $VID - \text{Pr}ofiles$ mapping. It will return with **D**'s profile (P, C, VID_D).
3. **D**'s profile is sent to the Policies' repository for validation checks against policies of the corresponding Object.
4. The policies Repository passes on the relevant VID_O policies to the ACM.
5. The ACM (subject to approving the received policies) creates a new capability for D (${}_{ext}CAP_D$) and sends it to S_x.
6. The propagation tree is updated.
7. Finally, S_x sends out the authority delegation statement in the form of ${}_{ext}CAP_D$.

System Model and Performance evaluation

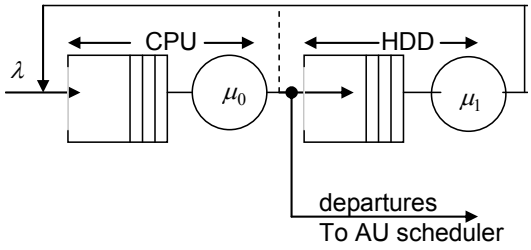
In this section analyze the proposed system framework model in which each end user requests a specific QoS as well as security (confidentiality). The Path Level Security Control plane chooses and sets up a path that satisfies the requested QoS and Security level constraints before any data exchanges can take place.. Once the request is accepted, it is also expected that the two constraints will be maintained throughout. The Jackson's queuing network theory model is utilized because whereas in general, imposes limits on the processing time distribution of each queue (i.e. that must be exponentially distributed), it however, produces quick and simple results.

Firstly, we will analyse the processing time delays at each AU ,followed by effects of increasing the number of AUs on overall response times. A proposed two-stage tandem network model depicting overall core functionality of an AU in negotiating required resources and security levels is shown in Fig. 12.

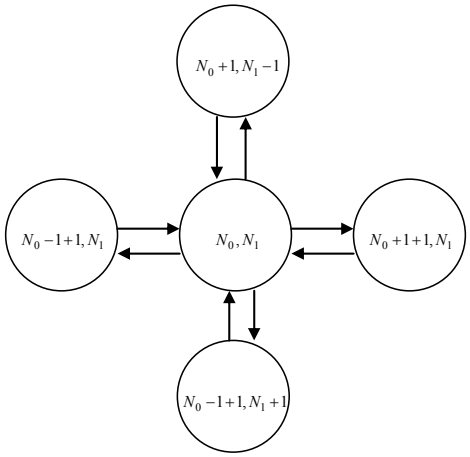
A few assumptions are made as follows:

- We assume each user is authenticated by the ingress AU on behalf of the Federation before. relaying the required security level together with desired QoS to the rest of the Federated network.
  - We assume a network with K classes of users. Each user class $k \in K$ has a fixed routing through the network in which the desired QoS together with security level can be guaranteed.
  - Arrival and service processes are not known apriori, but means and standard deviations of inter-arrival times and service times are known.
  - m_{si} -is the average processing time at an AU, for $i = 1,2,3,...$
  - s_{si} - is the standard deviation of processing time(s) at a given AU.

In a way each AU can be viewed as a *GI*/*G*/1 queue, hence, we used a Jackson network model as well as theorem to approximate it.



(a)



(b)

Fig. 12: (a). AU traffic model and (b) internal state transitions.

The waiting time at each AU is calculated from:

$$W_q(GI/G/1) = \frac{\left[C_a^2 + C_s^2\right]}{2} W_q(M/M/1) \tag{11}$$

At each AU the total waiting time is;

$$W = W_q + 1/\mu \tag{12}$$

Fig. 13 shows the average processing time at a single AU as a function of CPU utilization. Setting $C_a^2 = 2$ brings about increases in processing time thus indicating that variations in traffic arrival rates significantly affects the processing times.

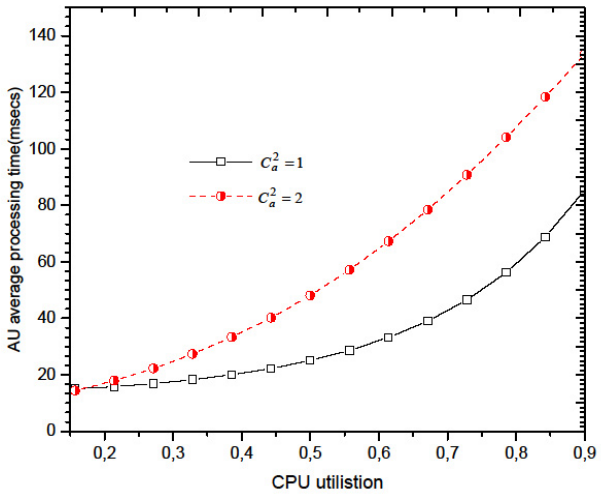


Fig. 13: AU Processing time as a function of CPU‘s utilization**.**

Shown in Fig. 14 is the fractional processing time(s) as a function of the number of AUs in the distributed architecture. For both $C_a^2 = 1$ and $C_a^2 = 2$ the processing times exponentially decays with increasing numbers of AUs.

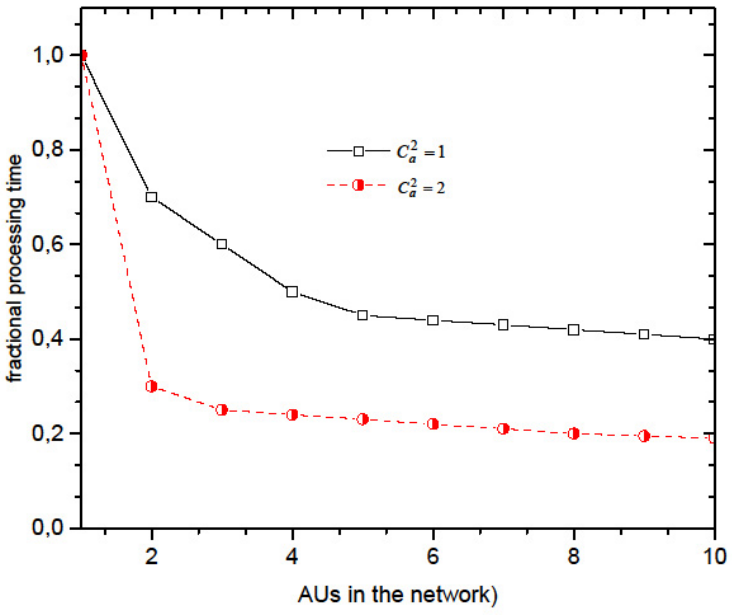


Fig. 14: Number of AUs versus processing times

This infers a distributed architecture will significantly bring about a reduction in processing times.

We further extend our performance analysis of the proposed authentication framework model by comparing three request negotiation algorithms (protocols) all of which relate to the manner in which end-to-end resources are negotiated by the ingress AU. These are:
Algorithm I: The Sequential Resources Request negotiation Protocol(S-RRNP) in which the ingress AU negotiates the required end-to-end resources in a sequential manner.
Algorithm II: Parallel Resources Request negotiation Protocol (P-RRNP), in which case the ingress AU identifies a candidate path before sending a resources request message to all associated transit AUs simultaneously.
Algorithm III: Centralised Requests negotiation Protocol (C-RRNP): The resources negotiation within the entire federation are carried out in a centralised manner. As such the ingress AU always requests the required resources and security via a designated central AU.
In our simulation, we compare the performance of the various requests negotiation protocols. In order to carry out the simulation we make further assumptions as follows:

- that the AU receives a Resources Request from users and maintains a state in memory for each of such Requests (e.g. representing the processing state of this resources request).
- two queues are necessary: one introduces the AU's Resources request processing time, while the other introduces the waiting time for the response.
- that the waiting time for the response from a remote AU equals its processing time of the Resources |Request message as well as generation of the Response.

Each ingress AU searches for a suitable end-to-end channel by querying with all associated transit AUs on the desired path. When it fails to find a-channel, it may either discard, or loop it back . The looped back Requests are queued once more with new arrivals thus may cause bottlenecks. We define three probabilities as follows:

- q - the probability that no channel was found on the first attempt hence, hence the request is looped back (looping probability).
- p - is the probability of discarding the Request on as resources do not exist. In this case a *FAIL* message is relayed back to the user.
- m -is the probability of finding a suitable channel on the first attempt.

Note that $p+q+m=1$.
As cited earlier, the looping back of requests traffic causes a bottleneck at the ingress AU. As such the maximum load at the ingress AU is approximated by:

$$\rho_{ko} = \frac{\lambda \cdot \mu_{ko}^{-1}}{1-(q+m)} \tag{13}$$

where k_o – is the loop back queue, ρ_{ko-} is the internal load.
Since memory requirement is directly linked to the number of request in the system, by using Little's formula, the mean number of Resources requests is:

$$\overline{N}_{RR} = \sum_{k \in Queues} \frac{\rho_k}{1-\rho_k} = \lambda T_d \tag{14}$$

The mean end-to-end delay can be calculated from the preceding formula taking into account blocking in the rest of the cascaded AUs on the chosen path. A simulation model was built in MATLAB[22],. The main units of each cascaded model built includes, an Entity Generator block (for generating requests), a Simulink Function uniformArrivalTime() block that

determines the interarrival times for the generated entities, Entity Queue block for storing entities in a FIFO order) and an Entity Server block modelling a server.

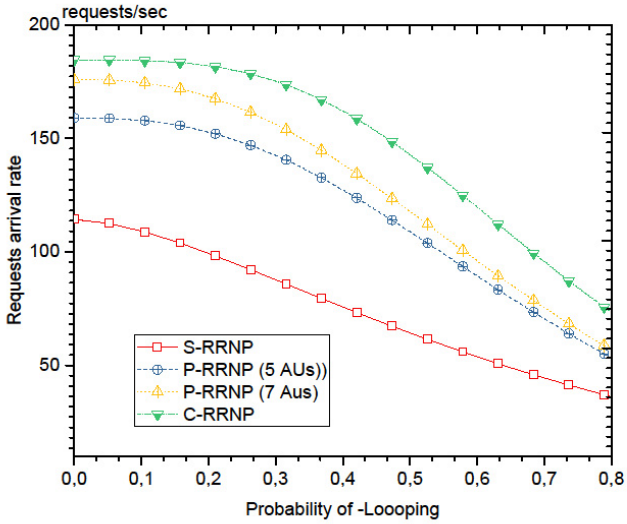


Fig. 15. Requests Arrivals as a function of Looping

In our simulation all three protocols have the probabilities p , q , and m fixed. In Fig. 15 we plot the arrival rate at a given ingress Au as a function of the probability of looping. The looped requests are also queued at the input and hence are a bottleneck hence the same queue is limiting the maximum possible arrival rate.

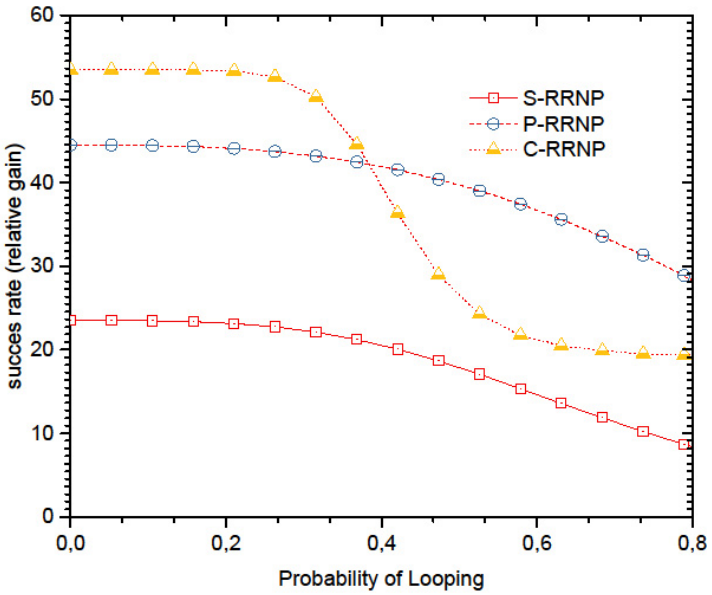


Fig. 16. Relative Gain comparisons

Overall the S- RRNP's performance as expected is the worst, whilst the C-RRNP outperforms by supporting a maximum arrival rate of 76 Requests per second in comparison to 30 Requests for the S-RRNP. We also compare the relative gain on the arrival rates for the three protocols. Whereas as expected the C-RRNP outperforms the other two, however when the looping probability increases beyond 0.45 its performance steeply degrades. This is because in this case, the frequency of signalling messages, as well as looping queues increase hence creating a further bottleneck on the designated central Au itself thus the steep drop in performance.

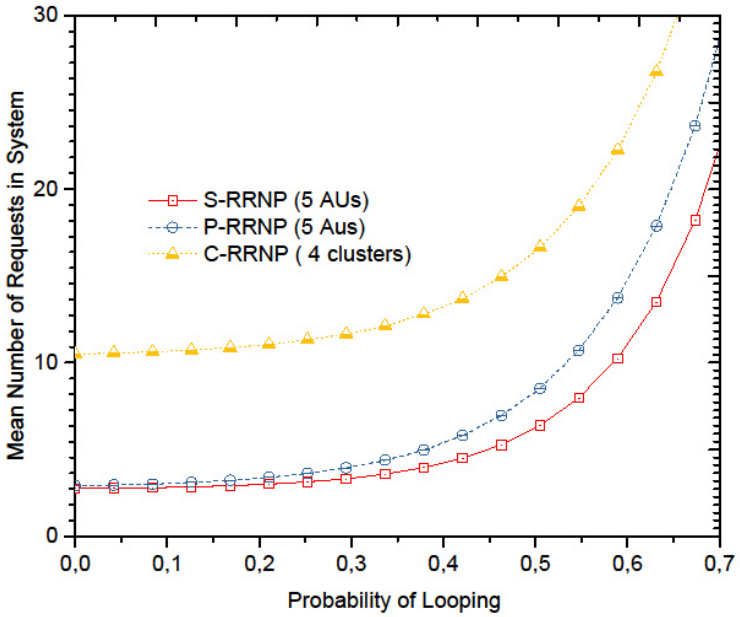


Fig. 17. Average number of requests at the ingress AU's scheduler.

The average number of requests in the AU's scheduler (number in system) is explored for the three different protocols when the arrival rate is fixed to 30 requests per second. From Fig. 17, we deduce that q greatly influences the number of requests in the system. Above a certain threshold value of q, the system can become unstable. It is therefore necessary to limit the looping probability so as to ensure acceptable QoS especially with regards to processing delays.

Conclusion

In this paper, we overview both IoT enabled network architecture as well as and access and control framework for various services and applications. A review of a generalized IoT enabled network's security architecture, as well as standards and protocols are also carried out. The paper also describes an access control framework that is applicable to the various would be applications and services. A distributed access control architecture is also analysed Overall it is deduced that a distributed architecture will significantly bring about a reduction in processing times.

References

[1] P. Fremantle, B. Azi, J. Kopeck and P.Scot, "Federated Identity and Access Management for the Internet of Things", SIOTP Proceedings, September 2014.

[2] M. Rita, E. Palattella, N. Accettura, X. Vilajosana, .T. Watteyne, L. A Grieco, I. G. Boggia, and I Dohler, " Standardized Protocol Stack for the Internet of (Important) Things", IEEE Communications Surveys & Tutorials, Volume 15, Number 3. Quarter, 2013.

[3] A. Messia, et.al, " A New Internet of Things Architecture with Cross-Layer Communication", Proceedings of the Seventh International Conference on Emerging Networks and Systems Intelligence, EMERGING 2015.

[4] T. Markmann, et. al, "Federated End-to-End Authentication for the Constrained Internet of Things Using IBC and ECC SIGCOMM '15 August 17-21, 2015, London, United Kingdom.

[5] R. Bonetto, N.Bui, V. Lakkundi, A. Olivereau, A Serbanati and M Ross, "Secure Communication for Smart IoT Objects: Protocol Stacks, Global Journal of Computer Science and Technology, Volume 16 Issue 7, 2016.

[6] ITU. ITU-T Y.2060 . "Intro to Internet of Things" https://www.itu.int/rec/T-REC-Y.2060-201206-I/en, June 2012.

[7] 802.15.4e-2012: IEEE Standard for Local and Metropolitan Area Networks - Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC Sublayer, Institute of Electrical and Electronics Engineers Std., 16 April 2012.

[8] G. Piro, G. Boggia and L. A. Grieco, "A standard compliant security framework for Low-power and Lossy Networks, draft-piro-6tisch-security-issues-01 (work in progress)", IETF 6TiSCH WG, December 14, 2013.

[9] S. M.Sajjad and, M. Yousaf,"Security Analysis of IEEE 802.15.4 MAC in the context of Internet of Things (IoT)", 2014 IEEE Conference on Information Assurance and Cyber Security (CIACS). Military College of Signals, Rawalpindi.

[10] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz, "RFC 3748: Extensible Authentication Protocol (EAP)," IETF Request For Comments, ttp://www.ietf.org/ rfc/rfc3748.txt, Jun. 2004.

[11] D. Forsberg, Y. Ohba, B. Patil, H. Tschofenig, and A. Yegin, "RFC5191: Protocol for Carrying Authentication for Network Access (PANA)," IETF Request For Comments, http://tools. ietf.org/rfc/rfc5191.txt, May 2008.

[12] Multi-cloud Secure Applications (MUSA) Project. Call: H2020-ICT-2014-1: http://www.musa-project.eu.

[13] Cloud-of-Things - (ClouT) Project. Call: FP7-ICT-2013- EU-Japan. http://clout-project.eu.

[14] In-network programmability for next-generation personal cloud service support (INPUT) Project. Call: H2020-ICT-2014-1,:http://input-project.eu.

[15] D. E. Culler, "The Internet of Everything - steps toward sustainability", University of California, Berkeley, CWSN Keynote, Sept. 26, 2011.

[16] S. Kraijak and P. Tuwanut, " A Survey on Internet of Things Architecture, Protocols, Possible Applications, Security, Privacy, Real-World Implementation and Future Trends. Proceedings of ICCT20 15.

[17] K. Sagara, K. Nishiki and Minoru Koizumi, " A Distributed Authentication Platform Architecture for Peer-to-Peer Applications, IEICE Transactions on Communications, Volume E88, Number 3, March, 2005.

[18] T. Elgamul. The Secure Sockets Layer Protocol.(SSL). http://www.ietf.org/proceedings/95Apr/sec/cat.elgamal.slides.html, April, 1995.

[19] G. Bai, L. Yan, L. Gu, Y. Guo and X. Chen, "Context-aware usage control for web of things", Security and Communication Networks, 2012.

[20] K. Hasebe and M. Mabuchi. Capability-role-based delegation in workflow systems. In Embedded and Ubiquitous Computing (EUC), 2010 IEEE/IFIP. 8th International Conference on, pages 711 –717, Dec. 2010.

[21] D.. Kulkarni and A.Tripathi, "Context-aware role-based access control in pervasive computing systems", In Proceedings of the 13thACM symposium on Access control models and technologies, SACMAT'08, pages 113–122, New York, NY, USA, 2008. ACM.

[22] https://www.mathworks.com/simevents//g1-g-1-queuing-system-and-little-s-law.html