Applied Mechanics and Materials
ISSN: 1662-7482, Vols. 433-435, pp 1618-1624
doi:10.4028/www.scientific.net/AMM.433-435.1618

Online: 2013-10-15

Distributed Customer Service Management for Internet of Things

Shuren ZHU

Information Science school, Guangdong University of Finance and Economics, Guangzhou, P. R. China

zhusr@gdcc.edu.cn

Keywords: Service Management, logical network, EPCIS, Internet of Things.

Abstract. It extends the network management mode of CSM, and offers a service-oriented management interface between customers and service providers. Extends some provider's service management towards the customers and enables customers to monitor and control information about service-specific QoS parameters. Logical network, a hierarchical presentation of abstract network is presented; the definition, attribute, structure and computing method of the logic network is discussed. The presentation and access of CSM-MIB which is implemented using distributed objects is completed; by taking advantage of logical network, the abstract customized views of physical network is built and the management functionalities of agreement, assignment, performance monitoring and reporting of CSM are implemented. Practice in implementing CSM system indicates that it is a reasonable method to construct CSM system using distributed object technology. Service Management is a developing new network management paradigm. At current stage, the applied CSM target concentrates on IP connection service.

Introduction

The Internet of Things is a kind of expansion for Ubiquitous computing. It is described as a self-configuring wireless network of sensors whose purpose would be to interconnect all things [1]. Integrated management solutions based on standardized network management architectures aim to support the network and service provider efforts in maintaining a high degree of quality and availability of their services while decreasing the cost of running the information technology infrastructure [2]. This implies some new management requirements. On the one hand, nowadays, an enterprise is not only free to choice among numerous IT service providers and network carriers but also needs to verify the Quality of Service (QoS) of the subscribed services. On the other hand, Internet Protocol and its standardized layered services and protocols have been accepted widely in enterprise inter-network. This leads to a layered hierarchy among service producers/consumers where a service provider is at the same time the customer of another provider. Thus, the problem of hierarchical service management must be settled. The problem domain dealing with the fulfillment of service-level agreements (SLA) [3] between service providers and their users is usually known as Customer Service Management (CSM) and is one of the research hotspots currently [4,5].

Although a large number of management implementations can be chosen, the traditional "flat" management methods are not competent well. In this paper, we present a novel virtual network based the Internet of Things service management solution in which Common Object Request Broker Architecture (CORBA) [6] is used as an underlying supporting environment, the interoperability of heterogeneous systems is achieved by CORBA; the presentation, management and visit of CSM is implemented through CORBA common service and facilities, such as Naming, Lifecycle, and Events services [7]. At the high level, through the method of defining, establishing and safeguarding etc, this architecture allows users to reassemble physical network elements to define and maintain a hierarchical "logical network", exhibiting physical networks in different views at different levels and facilitating the implementation of management functionalities of customer services such as agreements, assignments, and performance monitoring and reporting.

Requirement analysis of Customer Service Management

RFID (Radio Frequency Identification) tag technology is the “Internet of things” key Technology. Electronic tags can be attached to objects to be identified while readers can read or read/write, which depends on the memory structure and technology. Main modules are integrated into a single chip, complete communication with readers. When reader receives the RFID data, decoding and error checking are carried out to determine the validity of data, then, to transfer data wirelessly to a computer network. When the tag enters the magnetic field, it begins to accept the RF signal emitted from readers, and send out the product information stored in the chip by the energy obtained from induced current. Reader reads the information and decodes, then sent to the relevant central data processing information systems.

In a traditional service management model, customer and provider negotiate about QoS parameters, which reflect the characteristic of the service. Based on these negotiations, a SLA is signed, which defines rights, obligations and responsibilities for both parties. By this, both the provider and the customer usually use the same network and system management facilities (such as management platforms, trouble ticket systems and other various tools), thus there is no exchange of management information between customers and providers. Customers cannot monitor or control the QoS parameters of their subscribed services; no integrated fault management can be established, and an automated configuration and change management is not feasible at all.

Overall Information Network Architecture. An advanced intelligent scheduling platform is consisting of information system, database, users at all levels and machines for machinery cross-regional working. The information system and database will be constructed and monitored uniformly by information central with the aid of cloud computing technology and constructing the private cloud of machinery that will control all computing resource of information system and storage resource of database. The system users consist of mechanization administrative departments from information central to local plant, service organizations and machine users. The mechanization administrative departments at all levels and service organizations of machinery can supervise the machinery and implements in their areas and realize the intelligent scheduling of cross-regional working by using the information system uniformly. The machine users can get related scheduling information and all kinds of service information with the system. The trends, static state and other information factors related to machinery production including produce time in different areas, machines distribution and state, current product plan will be stored in the database of machinery cloud, and the database can get information related to machinery production by network connecting to the databases of national in time. With Internet of Things technology, machines and implements in all areas can communicate with datacenter of information service platform in wireless mode by configuration of monitoring front-end equipments machine carrying, and upload all kinds of information such as running state and geographical position automatically. The information platform can be simplified to three components including information service system, communication line and monitoring front-end equipment machine carrying [8, 9, 10].

EPC information service module. The EPCIS (electronic product code information service) module is the core module in the Internet of things, in the whole architecture of Internet of things, it is different from The ALE and RFID readers, the EPCIS is in the upper position, it is based on building with ALE and RFID Read device management middleware and so on[11]. The module defines four event types, as following: Object Event: represents an event that happened to one or more entities denoted by EPCIS. Aggregation Event: represents an event that happened to one or more entities denoted by EPCIS that are physically aggregated together (physically constrained to be in the same place at the same time, as when cases are aggregated to a pallet). Count Event: represents an event statistics some label of item's inventory capacity. Transaction Event: represents an event in which one or more entities denoted by EPCIS become associated or disassociated with one or more identified business transactions. Secondly, the function of EPCIS is to define the range of services in the Internet of things, it is very important that the services for building one Standard of the Internet of things system, the service mainly include two kinds: Capture service and Query service. The role of the capture service is to capture the underlying ALE events, so as to achieve knowledge of label Read status indirectly listening. The Query service is divided into the following kinds. Core search service: The characteristics of the service is the electronic label real-time query status of the corresponding, User convenient instant query or one kind of electronic label status, thus understands the current shape of the item State. Query subscription services: The characteristics of the service is to be able to provide a series of subscription rules, including the label Sign types, event types, subscribe to the effective time, subscription period, etc. For users to subscribe to the user in the submission Subscription request, the service can be based on user subscription rules periodically generate query report And sent to the client, it is convenient to customer learn information of interest.

Performance analysis of CSM. CSM offers a service-oriented management interface between customers and service providers. Service management is extended towards the customer side and enables customer to monitor and control the up-to-date and meaningful information about service-specific QoS parameters.

The service management is distributed and customizable. Customers should only view the management functionalities and information that belongs to them. The execution distribution and customization of management functionalities can not only relieve the management burden and provide better performance of service, but also facilitate load balancing and system reliabilities.

Management functionalities and information can be abstracted. A primary flaw of traditional management methods lies in that the load of network management information increases rapidly when network scales and the management information focuses mainly on the level of network elements. The information is in the lack of necessary refinement and abstraction. CSM will transfer selectable functionalities and refined information to customers, improving management capacity at the client side and leading to management security and lower network load.

CSM is easily used and its platform is independent. Different from traditional dedicated management platforms with which customers are not convenient to join in a management work, CSM can provide customers with the platform independent interfaces that are easy to use.

Main functionality analysis of CSM. *Configuration management*: used to access service configuration, with the ability to change the configuration (and the associated QoS parameters) and to handle the subscription of services, including subscribing/ unsubscribing services or sub-services.

Fault and performance management: used to enable monitor and control QoS parameters and SLA violations, and to notify the customers automatically that problems occur when running a service (by the provider), or to report problems detected by the customer. In addition, it must offer the capability to access historical statistic data about QoS parameters.

Accounting management: used to access all relevant accounting information of subscribed services, for example, resource usage, cost incurred current pricing scheme and electronic invoices.

CSM should provide sensible, instantaneous and adequate information so as to reflect the SLAs and QoS parameters negotiated. The information would be useful especially in a situation where problems and/or failures tamper with the quality of the service. CSM information is usually presented using visualization techniques, and possibly being integrated into the management platform of the customer. CSM information and functionality should be organized freely into different views that could be configured by providers according to the customer’s and service demand. This ensures that only relevant information in a suitable granularity can be passed on, and only appropriate actions can be taken by the customer.

A CSM Management Model

Logical Network. In order to present and implement the CSM management model, we introduce a hierarchical model, called as logical network. A logical network (LN) is a multiple way tree. Each node in the tree is a logical network element. A leaf node is associated with a physical network element in the specified management domain such as SNMP and includes attributes which reflect the management information of the physical network element, while an internal node is abstract one whose attribute values are determined by its children nodes. The root is a special node which stands for the LN. According to the recursive definition of the tree, each sub-tree of the LN is called as a sub-LN.

Each node has two kinds of attribute. One for management information and another for constrain rules. The management information attribute of each node is related to the managed object, either physically or logically. Its constitution and acquiring method are specified when LNs being defined. The leaf node usually corresponds to a physical managed object. Their management information attributes are tightly related to a specific management information protocol such as SNMP or CMIP. The internal node corresponds to a logical managed object. Their management information is computed from the children nodes in term of the node constrain rules.

Each LN tree represents a customized management task which consists of physical network elements in traditional management models and abstract logical network elements so as to exhibit a comprehensive and hierarchical management view. All LNs totally form an application-oriented and LN based management information base, called as LN-MIB such as CSM-MIB.

In definition stage, LN has confirmed the restraint rules of structure, MIB forming and getting management information. In running stage, the content of MIB changes with time. It can be described with a four element group LMIB＝MIB(A, C, S, T), among them, A is a management information attribute set of all LN node, C is a set for restraining the attribute, S is a node set in order confirmed by LN, and T is the specific time value. Besides this, LN has corresponding management interface, in order to finish the maintenance and inquiry of MIB. The interface has defined typical management to operate, such as get, set, walk, etc.

Formally, a logical network T is defined as below: T is a multiway tree and, T is either empty, or, T has a root r which can be defined as a 5-tuple.

$$T = T(I, M, C, n, Ti) \tag{1}$$

Where I is the identifier of T, M represents management information, C represents constrain and resolution rules, n is the number of children of root r, Ti is a sub-tree of T and $0 < i < n$.

Each sub-tree Ti has a root ri. If ri has at least a child, it must be an internal node of T and defined with the same tuple of r.

$$Ti = Ti(I, M, C, n, Tk) \tag{2}$$

Where the definitions of I, M, C, n are same with T's above and $0 < k < n$.

If ri has no child, then it is a leaf node of T and is defined as

$$Ti = Ti(I, M, C) \tag{3}$$

In the description above, I, M, C are all with complex data type, typically the object. I consists of the name, type and description of management tasks. On the leaf node, M represents management information of specific physical device, C denotes management operations against physical objects. For example, operations for get, set walk via SNMP interface. On the internal node, M represents the abstractive management information, C denotes constrain and resolution rules.

Each a sub-tree of a LN tree can be computed individually. Assume that there is a logical network management environment that is composed of a LN manager Ψ, m LN executors Θ_k, $1 < k < m$. The initial LN-MIB is Ω, managing period is Δ. Now we can describe the computing of task dispatching and management information collecting as below.

$$\Omega \xrightarrow[\Psi]{T_i} \Theta_{ki} \tag{4}$$

It describes that the manager is dispatching a task sub-tree Ti to the executor Θ_k and running Ti on it.

$$\Omega_{ik} \xleftarrow[\Delta,\Psi]{T_i} \Theta_{ki} \tag{5}$$

It describes that the executor Θ_k returns a partial LN-MIB Ω_{ik} to the manager after it completed managing operations specified by task sub-tree Ti and the sampling period Δ. The resulting LN-MIB after the sampling period Δ and completion of all task sub-trees is the union set of all sub-MIBs

$$\Omega = \coprod_{i=1,k=1}^{i=n,k=m} \Omega_{ik} \tag{6}$$

It describes that n is the number of sub-trees dispatched to the executor; m is the number of executors.

CSM based on LN. The logical network based CSM model is organized using distributed objects. The LN is an object which encapsulates management operations and CSM-MIB. By taking advantage of distributed objects, the structure, management attributes and constrain conditions of the logical network can be defined flexibly and dispatched to LN executor in different places by the LN's manager on demand. To implement a CSM system, steps would be taken below: Map CSM management tasks (eg. monitoring QoS performance of IP connections) into LNs, i.e. defining CSM tasks in the LN management environment (manager platform). The CSM tasks include account information, SLA of the service, CSM service priority, generating strategies of QoS monitoring and reporting. According to task requirements, define attributes and constrain rules of internal nodes in each level of LNs. Dispatch CSM management tasks represented in LN to distributed LN executor and run them. Clients query and verify QoS parameters, browse historical data, performance and fee reports using common browser. For instance, in the management of IP connection, monitor bandwidth, throughput and utilization of a specific connection.

Implementation of Distributed CSM based on LN

An LN is a customized, hierarchical logical view of network which lies between underlying physical network and network applications and is independent of the structure of the physical network. The major characteristics of logical network include: abstraction, scalability and customized capability, and hierarchy. Firstly, LN is service-oriented and customer-oriented, disregarding concrete properties of physical devices or end systems, providing customers with valuable and refined information. Secondly, by taking advantage of logical network, a providers or any third party entity can define or change flexibly the topology of network views on customers demand and configure service QoS parameters freely for monitoring and controlling. Thirdly, with a tree-like virtual network, LN is suitable for describing hierarchical administrative structure of an enterprise or organization and facilitates to implement security management on customer's demand.

Architecture. The implementation of a CSM system architecture based on CORBA consists of three entities: CSM backend system which provides CSM network management functionalities and can be distributed on the proper position to support clients to participate in customized management work; CSM clients which communicate with backend system, monitoring and controlling customized management tasks being delegated to run on backend systems; CORBA/SNMP gateway which bridges between CORBA and SNMP management domains and gives CSM system in CORBA domain a capability to manage the logical devices in SNMP domain in a transparent way. All entities are implemented using distributed objects and interoperate each other in the CORBA domain. CSM backend system is implemented in two layers. The lower layer is composed of CORBA ORB and CORBA common core services such as Lifecycle, Naming, Property, Persistence, and Security Event services. This middleware layer is used to manage distributed objects in a common style which will be used in the CSM higher layer.

Being set up above the middleware layer, the higher layer of CSM backend system forms a logical network overlay layer which consists of two kinds of distributed sub-system: one is Logical Network Management Environment (LNME); another is Logical Network Runtime Environment (LNRE). LNME is the manager of logical network which is provided and maintained by service provider. The functionalities of LNME as follows: This service executes physical network discovering, and creates physical element information base.

LNRE registering and loads balancing. This service accepts registering request and operating loads information from LNRE, delegates and dispatches management tasks of logical network to the LNRE with low operating loads by means of LNRE loads balancing algorithm. This service accepts logging requests from clients, authenticates customers and sends logical networks management and running status to legal customers. Logical network defining, maintaining and operating and pausing, These service generate customized management views of service SLAs and QoS parameters on the basis of customers according to physical network and CSM requirements.

LNRE is a distributed and cross-platform agent of CSM management system. The functionalities of LNRE as follows: Logical network executive management. This service receives logical network information dispatched and delegated by LNME, interacts with other LNRE to exchange information of operating loads and transfers management tasks among LNREs on the occasion of some LNREs being shutdown or in failure.

This service collects running information of real physical elements defined in the logical networks, recalculates or refines the information acquired and saves it in a meaningful way into the CSM-MIB for querying of client. At the same time, all information is also stored in an external database, typically a RDBMS, as history information for statistics. The LN executive will draw support from a CORBA/SNMP gateway, sending get/set requests in CORBA domain to SNMP domain or response trap events from SNMP domain to complete inter-domain management. Client service which accepts querying requests from authenticated clients and sends out management information instantaneously to clients.

CSM clients equips a visual user interface through which customers can send login requests to LNME and get the customer views of logical networks related to them after being successful in authentication. A customer can monitor or control service's SLAs or QoS dynamically from the client by operating against the logical networks in GUI. Clients will connect to a LNRE automatically to get up-to-date and meaningful information about LNs that customer need and display service's performance or status information (such as status of network element, topology, and time curve of QoS parameters) in CSM GUI. Historical or statistics information of LNs can also be obtained by clients. Using these information, CSM clients provides a comprehensive report for customers about the services subscribed by charting or reporting visual methods.

CSM Prototype Implementation. In our prototype implementation, we used the X/Open Joint Inter-Domain Management translation algorithm, translated the relevant SNMP MIBs (Management Information Bases) into CORBA IDL. By building a simple CORBA/SNMP gateway, we converted CORBA method invocations into SNMP PDUs (Protocol Data Units) and vice-versa. We also made use of an SNMP toolkit that is available as freeware from CMU.

A UML class diagram of CSM prototype implementation includes the implement of CSM-MIB class, LNME and LNRE which finish LN management, task Dispatching and execution. LNs are organized as visual hierarchical views in which each node stands for a specific GSM management task. LN can carry out on a LNRE i also can be running on different LNRE. According to environment, each sub LN responsible updating for the sub LN which it represents. LNME can carry on task dispatching and switching over automatically, according to the load or failure situation that is monitored. LNRE stores information of CSM-MIB to backend database, it can be saved as historical datum or create more abstract report datum.

Conclusion

Customer Service management is an under developing new network management paradigm. It offers a service-oriented management interface between customers and service providers and extends the provider's service management towards the customers and enables customers to monitor and control up-to-date and meaningful information about service-specific QoS parameters. Practice in implementing CSM system indicates that it is a reasonable method to construct CSM system using distributed object technology. At current stage, the applied target of CSM concentrates on IP connection service. It will try to extend it to more services such as ATM-PVC.

Acknowledgements

The research work was supported by the Natural Science Foundation of Guangdong Province under Grant No. 10151032001000001.

References

[1] Conner M, Sensors empower the "Internet of Things", E-Publishing Inc., New York, 2010.

[2] HeinzGerd H, Sebastian A, Data Communications and Networks, Vol.40 (2008), pp491-492.

[3] Dinesh V, Supporting Service Level Agreements on IP Networks, Macmillan, New York, 2009.

[4] Langer M, Loidl S, Nerb M, Customer Service Management: A more transparent view to your subscribed services. In: Proceedings of the 8th IFIP/IEEE International Workshop on Distributed Systems: Operations & Management, Newark, USA, 2010.

[5] Goldszmidt G., Yemini Y., IEEE Communications Magazine, Vol.36 (2008), pp66-70.

[6] Andrew J., SNMP agent toolkit, Pittsburgh, USA: Carnegie-Mellon University, 2004.

[7] Object Management Group, Unified Modeling Language™ (UML) version 1.5. Needham, USA: Object Management Group, Int. 2003.

[8] Ding W., Liang C., Xia, M., China Computer & Communication, Vol.7 (2009), pp:36-37. (in Chinese)

[9] Zhang Q. z., Liu B. w., Li J.t., Logistics Technology, Vol.206 (2009), pp200-202. (in Chinese)

[10] Lang M.-x., Journal of Trans. Systems Engineering and Information Tech., Vol.10 (2006), pp65–68.

[11] EPC Global, EPC Information Service Version 1.0.1 Specification, 2007.