An Integrated IoT enabled On-demand Grocery Shopping and Delivery Cloud System using MTComm at the Edge

S M Nahian Al Sunny, Xiaoqing "Frank" Liu, Md Rakib Shahriar
Department of Computer Science and Computer Engineering
University of Arkansas
Fayetteville, Arkansas, USA
Email: {smsunny, frankliu, mrshahri}@uark.edu*

***Abstract*—In this paper, we present a novel approach to design and develop an integrated IoT enabled on-demand grocery shopping and delivery cloud (IGSDC) system using Machine Tool Communication (MTComm) method at the edge. MTComm was developed to provide remote monitoring and operation services of manufacturing machines over the Internet. Here we designed an MTComm based edge computing hub (MEH) that connects to heterogeneous physical resources using their own languages and protocols and exchanges information with cloud applications through RESTful services and XML messages. Based on IoT data acquired through MEHs, the IGSDC system tracks grocery product inventories, notifies consumers when they are running low on grocery products, and offers available purchase options from different suppliers. On the other hand, suppliers connect their warehouse robots using MEHs to the cloud by publishing their monitoring and operation capabilities and perform product delivery remotely using autonomous mobile robots. Using MEHs, users can monitor, manage, and operate their resources both locally and remotely. This paper describes different components and processes of the four-layered service-oriented IGSDC system. It also discusses how MTComm is utilized for edge computing in both environments to improve scalability and efficiency. A fully functional prototype system with several IoT devices and robots was implemented. Experiments demonstrated excellent feasibility and effectiveness of MTComm based edge computing to integrate heterogeneous in-home IoT devices with in-warehouse robots through cloud applications and provide enhanced on-demand grocery shopping and delivery services.**

***Keywords*-MTComm; edge computing; cloud based grocery services; IoT devices; delivery robots**

I. INTRODUCTION

A study in 2015 found that 25 percent of U.S. consumers ordered groceries online, and 55 percent were willing to do so [1]. Internet-of-Things (IoT) is making significant contributions to improve the current trends of online grocery services [2]. As grocery purchases are usually done on need-basis, IoT is aiding online grocery services by allowing consumers to keep track of grocery items, set reminders for expiration, and even order directly from built-in interfaces [3]. Moreover, suppliers and retailers are using IoT technologies to improve management and distribution of products and promote better shopping experience. The Internet-of-Shopping is considered as the future of supermarkets where no human intervention is required to manage it [4]. Innovations in robotics, artificial intelligence, and computer vision made product delivery by self-driving robots a reality [5]. Integrating in-home and in-warehouse IoT devices and robots through cloud services has the promise of creating a robust and efficient on-demand grocery shopping and delivery platform that connects consumers and suppliers based on their needs and offered services respectively and allows them to monitor, manage, and even operate their physical resources over the Internet. Heterogeneity of communication protocols used by such resources is a major challenge to develop such a system. IoT devices use various communication protocols, such as MQTT, CoAP, Z-wave, Zigbee etc., data formats, definitions, and system architectures [6]. Moreover, industrial resources use their own communication protocols, e.g. Ethernet/IP, Modbus, PROFINET, EtherCAT etc., and standards [7]. Edge computing can aid by transforming data near end devices to a common format and provide cloud applications a uniform interface to interact with them [8].

In this paper, we present a novel approach to design and develop an integrated IoT enabled On-demand Grocery Shopping and Delivery Cloud (IGSDC) using Machine Tool Communication (MTComm) method at the edge. MTComm enables monitoring and operation of heterogeneous manufacturing machine tools at both machine and component level through RESTful web services and XML messages over the Internet [9]. In IGSDC, we designed an MTComm based Edge computing Hub (MEH) for IoT home appliances and devices to enable service-oriented cloud based on-demand grocery product tracking and ordering capabilities. An MEH communicates with different types of IoT devices and robot systems using their own languages and protocols. The collected data is converted to a common MTComm XML format and sent to cloud services. Based on IoT data, the cloud notifies consumers upon detecting low supply and offers available ordering options from different suppliers. On the other hand, suppliers connect their warehouse IoT devices and delivery robots using MEH to the cloud by publishing their capabilities as RESTful web services and perform product delivery remotely and autonomously upon receiving an order. As all physical resources communicate with the cloud through MEH using MTComm XML messages, IGSDC does not require to deploy separate cloud services and applications for consumers and suppliers, hence provides better scalability and efficiency. As many robot

978-1-7281-2708-8/19/$31.00 ©2019 IEEE
DOI 10.1109/EDGE.2019.00024

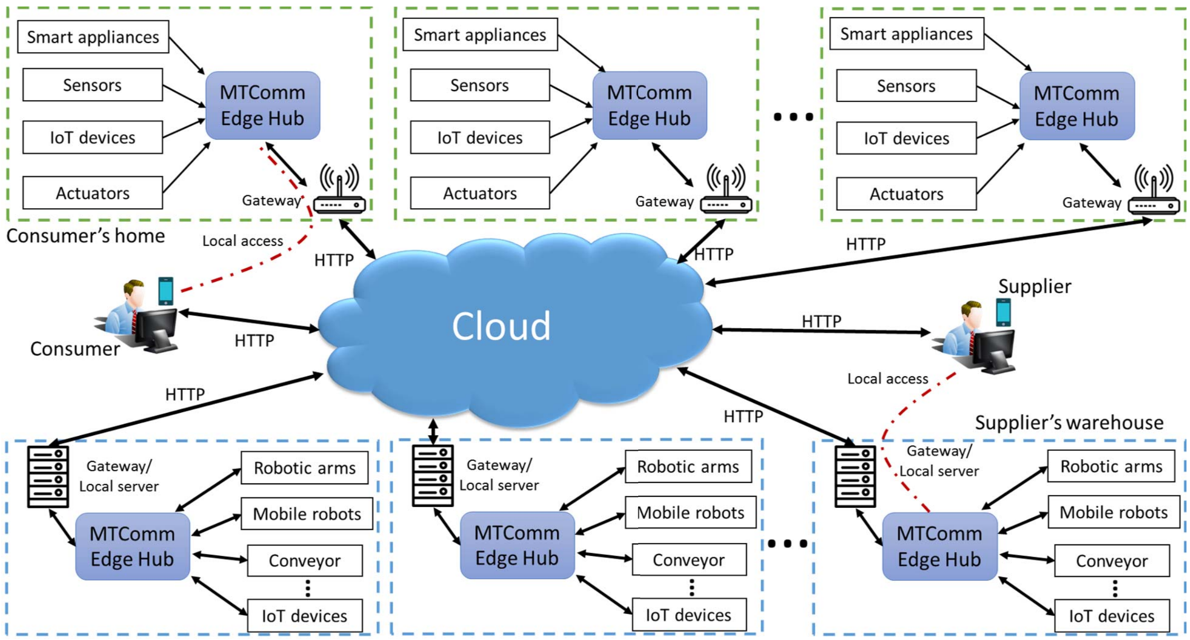

Figure 1: Conceptual framework of the IGSDC system

systems used in manufacturing and warehouse environments share similar operation principles, MTComm can be easily applied to in-warehouse robot systems. However, IoT devices are not readily compliant with MTComm. Therefore we extended MTComm method to incorporate IoT resources. A fully functional prototype system with several MEHs, IoT devices, and robots was implemented. Experiments demonstrated excellent feasibility and effectiveness of MTComm based edge computing to integrate heterogeneous resources with cloud services and provide on-demand grocery shopping and delivery services.

Section II explains the conceptual framework and architecture of IGSDC. Utilization of MTComm for edge computing are discussed in Section III. Section IV briefly describes the service execution process of providing on-demand grocery service. Details of implementation of an IGSDC prototype system are discussed in Section V. Conclusions are given in Section VI.

II. Architecture and Components of IGSDC

Fig. 1 shows a conceptual framework of IGSDC. Users (consumers and suppliers) interact with the cloud via smartphone or web browser applications using HTTP. All physical resources, including in-home IoT appliances devices and in-warehouse machines, are connected with cloud services using MEHs. An MEH is an embedded system, like Raspberry Pi, ASIC, FPGA etc., with sufficient computational and network capabilities to establish bi-way communication between different types of resources and cloud services using MTComm. MEHs communicate with the cloud using HTTP via gateways. Users can also interact with an MEH locally via a smartphone app or built-in display interface.

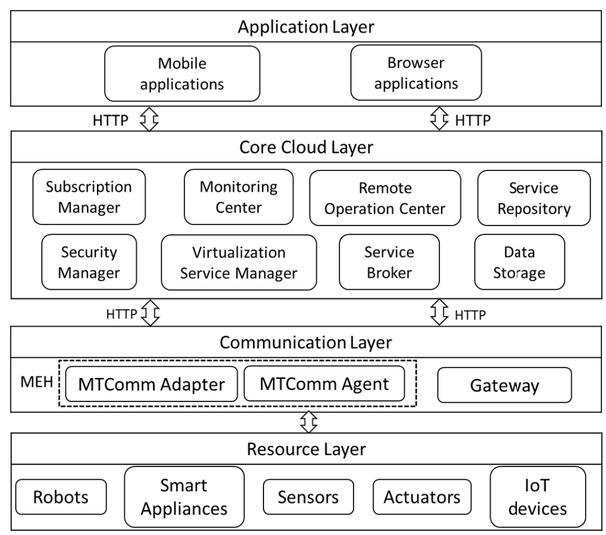

Figure 2: Four layered architecture of IGSDC

Fig. 2 presents a scalable service-oriented four-layered architecture of IGSDC. The architecture presents a hierarchical view of system components and associated services. The resource layer contains different types of physical end devices, such as sensors, actuators, IoT devices, smart home appliances, robotic arms, mobile delivery robots, computing resources, conveyors etc. Communication layer establishes communication between resource layer and core cloud layer using MTComm. Core cloud layer refers to the cloud infrastructure that hosts necessary cloud services to connect consumers with suppliers and manage web services of their subscribed physical resources. It includes subscription services (SS) for managing user and MEH subscriptions, virtualization service manager which use MTComm *probe* to acquire structural and configuration data of MEHs and create virtual copies, security manager for ensuring security and privacy, monitoring center (MC) which uses MTComm *current* to acquire most recent status of resources, remore operation center (ROC) for executing operations of resources remotely through MTComm *operation* service, service broker for running recommendation algorithms to match potential customers and service providers, service repository, and data storage. Application layer manages multiplatform applications for users to interact with the IGSDC cloud.

III. Utilization of MTComm in Edge Computing

A. MTComm Edge Hub (MEH)

MTComm Edge Hub is an embedded device positioned at the edge of an environment that uses MTComm method to integrate web and cloud services with physical resources. As depicted in Fig. 3, an MEH primarily consists of MTComm Adapter and MTComm Agent. An adapter module in MEH has five components - 'Data Collector' that acquires data from resources periodically, 'Operation Driver' performing requested operations by sending commands or executing a program, 'Resource Manager' responsible for administering connected resources, 'Dictionary Handler' which handles both incoming (operational information) and outgoing (collected data) key-value pair based dictionary, and 'Communicator' that establishes communication with resources and the Agent module's 'Communicator'. The Agent contains an 'XML Handler' to generate, translate, and parse XML messages. 'Data processor' is used to perform computation and analysis with received information (both monitoring and operational) including identifying critical events, generating alerts, verifying incoming operation requests etc. The Agent also includes an 'HTTP Server' for communication over the Internet with a 'Service Manager' that manages MTComm RESTful web services and maps them with corresponding physical resources. An MEH also contains a 'Security Manager' module for authentication and encryption, a 'Configuration module' through which users can add, remove, and configure their resources, and a 'Storage' unit that is shared by all other components. An MEH includes multiple interface modules, such as USB, Ethernet, Bluetooth, Wi-Fi etc., and support for different communication protocols to connect to heterogeneous resources. As these protocols vary from one environment to another, MEH Adapters are required to be custom-made for different environments.

B. Adopting MTComm for IoT resources

MTComm uses a semantic ontology to represent a machine's organization and characteristics [9]. The ontology describes a machine and its components in a hierarchical fashion. A machine is considered as an MTComm *device*. A *device* is composed of one or more *components*. A *device* or *component* has one or more *dataitem* - a piece of status data

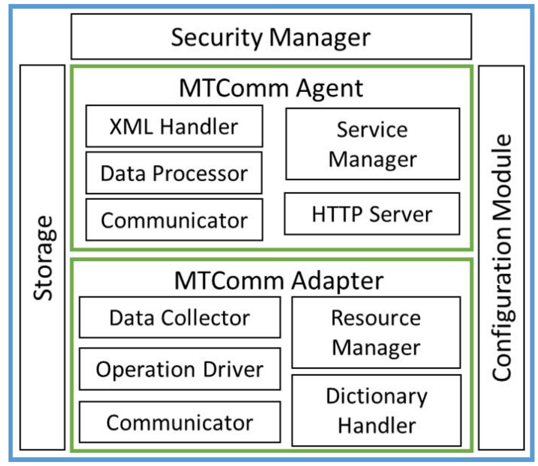

Figure 3: Architecture of MTComm Edge Hub

(numeric/alphabetic) and *operation* - a task or activity. An *operation* can have several *parameters* if required. All nodes of this ontology have their own attributes. *Dataitems* and *operations* are grouped based on their logical and functional organization, instead of physical organization.

Unlike manufacturing machine tools, IoT devices typically have simpler data structure with fewer or even no components. Many IoT sensors generate only a single type of data values. Hence presenting a single IoT device using MTComm ontology is not very efficient. So we considered multiple IoT resources together as an MTComm *device* based on their functionality, location, or user choice. Each individual IoT resource is a component of the device entity. If an IoT resource consists of multiple elements, those are considered as lower level *components*. *Dataitems* refer to sensor values and *operations* refer to actuation activities or tasks. Characteristic information or metadata are described by attributes. A home may have one or more *device*. After a user confirms the organization of a *device*, the agent creates a *probe* document accordingly. Any IoT resource can be added to or removed from an existing *device* as a *component*. A *current* message includes most recent values of all *dataitems* of a *device*. Data streams are timestamped and sequenced.

To explain the mechanism more, let us consider a user whose home has one smart refrigerator (Rf), one IoT light bulb (Lb), and one IoT humidity sensor (Hmd). Rf consists of two parts - freezer and refrigerator, each with a temperature sensor and a temperature changing activity. Rf also shows overall operation mode. Lb provides current status and performs switching action. Hmd gives current humidity level of the room. The user can create a single MTComm *device* for his entire home, as shown in Fig. 4.

C. MTComm for in-warehouse resources

MTComm is used for in-warehouse resources in similar fashion as described above, with one key difference. As most in-warehouse resources are complex machines with many

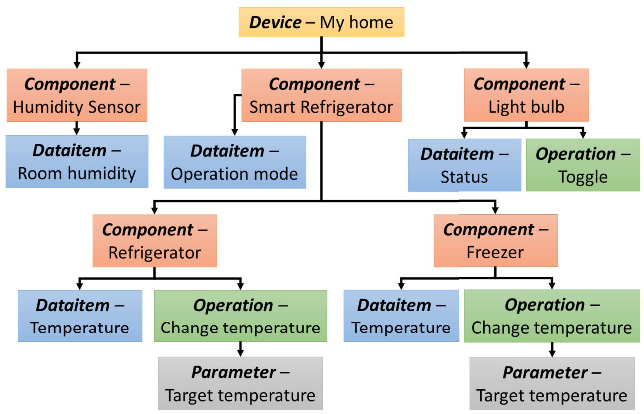

Figure 4: Example of an MTComm *device* hierarchy

data elements and operations, each resource and its related modules including controllers (hardware and software), external sensors, and actuators, is considered as one MTComm *device*. One MEH can support one or multiple *devices*.

D. Advantages of Using MTComm at the Edge

1) Naming: MEH employs a two-step naming scheme. 'Resource Manager' in MEH Adapter assigns unique (locally) identifier and network address for all connected resources. These information alongside their metadata (if any) are stored in a relational table in the MEH storage. 'Service Manager' of MEH agent assigns user and service friendly unique (locally) name (either assigned by users or auto-generated) based on the MTComm ontology using dot convention and maps them with corresponding RESTful services. For instance, considering previous IoT home example, the *dataitem* of Lb is addressed as `"my_home.bulb.status"`. This naming mechanism makes service and resource management easier for both users and cloud/web services. Each MEH is given a unique (globally) identifier by the cloud/web server after registration, so each name assigned by an MEH becomes unique globally as well (e.g., `"meh01b_my_home.bulb.status"`). The format of the web service URLs is simple `"address_of_MEH/device_id/service_name"`. MTComm also allows the use of attributes as path parameters in the service URL to perform element specific queries and actions. Service URL to Lb is - `http://10.0.5.5:7777/my_home/operation?path=//bulb//Operation[@id="toggle"]`

Thus MTComm allows to create service URLs using the ontology based on user requests. Therefore, the cloud only requires storing MEH id and web address and can dynamically generate necessary service URLs, instead of storing individual service URLs.

2) Data abstraction: By transforming and preprocessing information in MEH, MTComm provides efficient data abstraction at the edge and prevents external access to raw sensitive information like hardware details, network addresses, protocol types and versions etc.

3) Resource management: A user can easily add a new resource to an MEH by establishing the in-between connection and defining its appropriate position in the existing MTComm *device* structure. The process of removing an existing resource is just the opposite. If a resource fails or becomes unavailable for some reason, it does not crashes the whole system. Instead the MEH marks that resource as faulty or unavailable and shows probable causes (if available).

4) Reduced latency and bandwidth usage: Each HTTP packet contains certain header information and metadata. Hence sending data from various sources to the cloud as a single XML message reduces latency and bandwidth usage.

IV. Service Execution Process in IGSDC

Both consumers and suppliers of IGSDC has to subscribe as users, create their user profiles, connect their resources, and configure corresponding MTComm device(s) structure. A user subscribes an MEH to the cloud by providing its web address. SS assigns a unique id for the MEH, requests its *probe* service, parses response XML, creates a virtual copy of the MEH and its resources using extracted data, and stores it in storage. An MEH collects status data from its connected resources and stores them in MTComm XML format at regular intervals. MC in the cloud sends periodic *current* or *sample* requests to MEH, receives XML responses, extracts data, and stores them in storage. When a user requests to view resource status, the MC responds with the most recent data available. When low supply is detected, the service broker searches for potential supplier services using recommendation algorithms and provides consumer with resulting service options. For executing a resource operation, the ROC generates an *operation* request using the ontology with necessary instructions and parameters. Then the ROC collects corresponding MEHs address from service repository, creates specific *operation* URL, and sends the message to the MEH. After verification and validation, MEH extracts operational information and sends necessary commands to corresponding resource to complete the operation.

V. Implementation of IGSDC Prototype

As a proof-of-concept, we implemented a fully functional prototype IGSDC system (Fig. 5). We developed an Android application to interact with the cloud. This demo app had the capability to display current status of MTComm *devices*, view available product ordering options, and place orders. Raspberry Pi (RPi) were used as MEHs. We created two zones in our lab - home zone and warehouse zone, fifteen meters apart. Each zone had one RPi as an MEH running MTComm adapter and agent programs developed in Python. In home zone, we had a GE smart refrigerator and two Arduino boards – one was connected to a weight scale built using a square force-sensitive resistor (FSR) and published

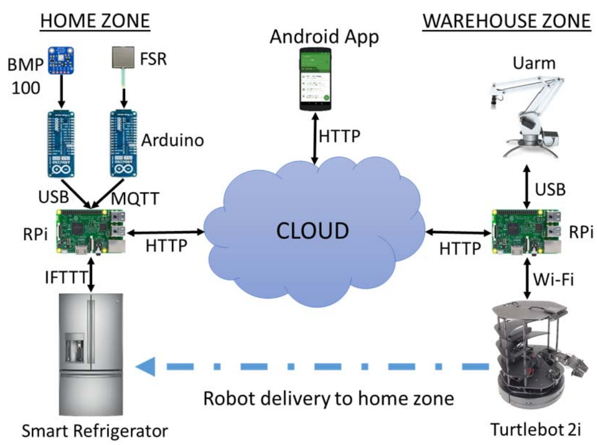

Figure 5: IGSDC prototype

its data via MQTT, and the other was connected to a BMP180 barometric pressure sensor and transferred data to its Rpi via USB. We used IFTTT webhooks to collect data from the refrigerator. The warehouse zone consisted of two robots - one Uarm robotic arm and one Interbotix Turtlebot 2i mobile robot. The warehouse RPi was connected with Uarm using USB connection and with Turtlebot over Wi-Fi. The cloud services were deployed in a virtual machine.

Experiments were conducted emulating a hypothetical grocery shopping and delivery scenario. The refrigerator and two sensors were considered as components of a single MTComm *device* with seven *dataitems*. The home RPi collected data from these sources at 10 Hz and created corresponding XML messages. A full half-gallon milk carton was placed on the weight scale at the beginning. When a nearly empty carton was placed, the cloud detected low supply and notified the user along with available ordering options through android app. Once an order was placed, the cloud generated an *operation* request containing instructions for both robots to their RPi. First, the Uarm executed a pre-programmed co-ordinate file in `.csv` format to emulate carrying a new carton to the Turtlebot. Once done, the MEH agent extracted destination co-ordinates (position of the refrigerator) from the *operation* message and sent them with necessary commands to the Turtlebot which then traveled to the home zone navigating autonomously. When it reached the refrigerator, the cloud detected the event through location matching and notify user through app. The Turtlebot then emulated dropping the product near the refrigerator and traveled back to its original position. Several test runs were successfully completed without any human intervention.

VI. Conclusion

In summary, this paper presents a service-oriented cloud for grocery shopping and delivery through integration of different types of IoT device and delivery robots. MTComm based edge computing is the key technology of IGSDC as it allows cloud services to interact with users and their resources in a uniform manner. Using IGSDC, consumers can track their grocery items through IoT devices and suppliers can monitor and operate their warehouse machines remotely. IGSDC provides data driven matching of consumers and suppliers based on their needs and capabilities. MTComm enables efficient XML based bi-way communication between cloud and IoT resources through an edge device. The prototype system successfully demonstrated IGSDC's ability to provide unique on-demand IoT enabled grocery shopping and delivery capabilities. To realize its full potential, more researches are required including developing better security and privacy mechanisms, using data analysis and machine learning algorithms, applying resource management etc.

Acknowledgment

This was supported by NSF Grant CMMI 1551448 entitled "EAGER/Cybermanufacturing: Architecture and Protocols for Scalable Cyber-Physical Manufacturing Systems".

References

[1] Statista, "U.s. consumers: Online grocery shopping - statistics & facts," 2015 (accessed April 16, 2019). [Online]. Available: https://www.statista.com/topics/1915/us-consumers-online-grocery-shopping/

[2] A. Fagerstrøm, N. Eriksson, and V. Sigurosson, "Whats the thing in internet of things in grocery shopping? a customer approach," *Procedia computer science*, vol. 121, pp. 384–388, 2017.

[3] H. Desai, D. Guruvayurappan, M. Merchant, S. Somaiya, and H. Mundra, "Iot based grocery monitoring system," in *2017 14th Int. Conf. on Wireless and Optical Communications Networks (WOCN)*, 2017, pp. 1–4.

[4] Y. Vilner, "The internet of shopping (ios) is a thing, and its growing," 2016. [Online]. Available: https://www.forbes.com/sites/yoavvilner/2016/03/23/the-internet-of-shopping-ios-is-a-thing-and-its-growing/

[5] S. Cho, D. Lee, Y. Jung, U. Lee, and D. H. Shim, "Development of a cooperative heterogeneous unmanned system for delivery services," *J. of Inst. of Control, Robotics and Sys.*, vol. 20, no. 12, pp. 1181–1188, 2014.

[6] A. Sill, "Standards at the edge of the cloud," *IEEE Cloud Computing*, vol. 4, no. 2, pp. 63–67, 2017.

[7] S. M. N. A. Sunny, X. F. Liu, and M. R. Shahriar, "Communication method for manufacturing services in a cyber–physical manufacturing cloud," *Int. J. of Comp. Integrated Manuf.*, vol. 31, no. 7, pp. 636–652, 2018.

[8] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[9] S. M. N. A. Sunny, X. F. Liu, and M. R. Shahriar, "Mtcomm: A semantic ontology based internet scale communication method of manufacturing services in a cyber-physical manufacturing cloud," in *2017 IEEE Int. Cong. on Internet of Things (ICIOT)*, 2017, pp. 121–128.