Towards an Edge-Cloud Platform for Multirobot-Multihuman Cooperation in Urban IoT Ecosystems

Seng W. Loke, Amin B. Abkenar, Niroshinie Fernando
Centre for Internet of Things ECOsystems Research and Experimentation (CITECORE)
School of Information Technology, Deakin University, Geelong, Australia
seng.loke@deakin.edu.au,a.abkenar@deakin.edu.au,niroshinie.fernando@deakin.edu.au

***Abstract*—We envision robots in cities will be increasingly pervasive - forming new computational nodes, connected to each other and to the Internet, adding to the already proliferating mobile, wearable and fixed ubiquitous computing devices. This paper discusses the notion of cooperation schemes to enable such urban robots to work together with each other, with IoT devices and with humans in different modes, and outlines prototype distributed middleware we are building towards this end.**

***Index Terms*—edge-cloud computing, Internet of Things, Robots, Cooperation Schemes**

I. Introduction

Emerging is the notion of Internet-connected robots in private and public spaces, in particular in cities, from homes, offices, aged care homes, shopping malls, museums, exhibition halls, walkways, streets, to city canals [9], [13], [14], [18]. Such robots would be cloud-connected, but also utilise edge computing infrastructure where desirable. Recent work has seen a number of initiatives going beyond cloud robotics to edge-cloud robotics [7], [17].

This paper first outlines the idea of the multirobot-multihuman IoT ecosystems, where robots and humans occupy shared spaces and work together, cooperating in different ways, in a vision of a smart city. Then, we describe the concept of *cooperation schemes*, representing abstractions of patterns of cooperation among robots, and among robots and people. We then describe our current prototype implementation of three cooperation schemes, and conclude with future work.

II. Multihuman-Multirobot Cooperation Schemes: Concept and Architecture

There can be a number of different types of robots cooperating for different tasks in public spaces. As an illustration, Figure 1 illustrates different collections of robots, some acting as tour guides, some acting as information kiosks and some as security guards.

The idea is that a distributed platform, with components installed on each robot, enables different collections of robots to work together in different ways. Minimally, the platform should enable the robots to communicate with each and exchange information while at the same time connecting to the cloud (as needed) to connect to centralised control, which includes (i) the manufacturer to receive software and configuration updates, (ii) administrators for reconfiguration and commands, and (iii) maintainers who maintain and manage the robots in their day-to-day operation.

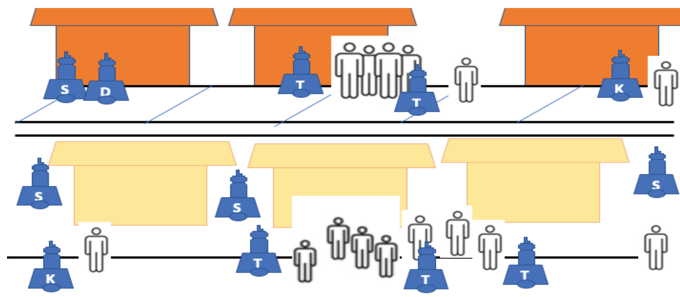

Fig. 1. Illustration of different collections of robots cooperating for different tasks, one could imagine, in a marketplace (shopping mall) or a large art gallery. Robots labelled T are tour guide robots - a group of two T robots are guiding one group and another group of three T robots are guiding another group. The S robots are security robots and work together and distribute themselves to monitor mall at different places, including stores and walkways, and the K robots are mobile kiosks robots containing advertisements and act as information booths which users can interact with via a screen or voice dialogue. Although the robots are depicted as having the same form, it is possible that physically, the robots have different forms as customised for their purposes.

Figure 2 illustrates the distributed architecture of our platform comprising fog nodes providing additional resources to edge nodes, edge nodes (as robots), and the centralised cloud platform to which fog nodes connect to. Humans are in the loop and can control the robots, via the cloud, fog nodes or directly send commands to robots. Each edge node depicted is a robot, containing components to support each cooperation scheme - illustrated are three components on each robot so that three cooperation schemes are supported. The aim is also that robots can, as needed, download (or human managers can push to robots) components for cooperation schemes, enabling new cooperation behaviours for the robots. Humans can also download and install on their mobile devices components for cooperation schemes to work with robots in new ways, as needed.

Each cooperation scheme specifies and implements a pattern of interaction among the edge nodes, or a type of cooperation possible among robots, e.g., robots can work together as a "supercomputer" to process sensor data (workload sharing co-

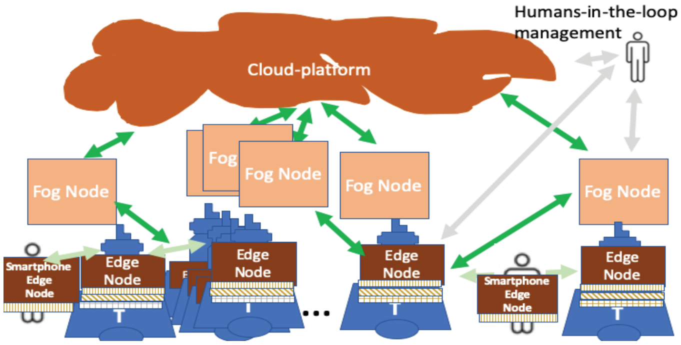

Fig. 2. Illustration of our edge-cloud platform. Each robot is viewed as an edge node - the illustration shows three shaded rectangles on each edge node (diagonal lines, vertical lines, and squares) showing three different (types of) components installed on each robot, each for a different cooperation scheme. Humans can also cooperate, via their smartphone, in a particular cooperation scheme with robots (illustrated via the component depicted by the rectangle with vertical lines on smartphones and robots). For certain applications, the remote cloud platform need not even be involved - just fog/edge nodes.

operation scheme), robots can send/receive messages to/from each other and humans (messaging cooperation scheme), and robots can recognize collective activities of multiple humans (a group activity recognition cooperation scheme) and react to this. The three example schemes mentioned here are discussed in detail in the next section.

III. Prototype of Cooperation Schemes

A. *Multirobot-Multihuman Messaging*

Messaging mechanism in any cooperative Multirobot-Multihuman Ecosystems would be an essential components of such systems. In the proposed architecture, both p2p and centralized messaging can be used depending on the availability of communication technology on the edge/fog devices. We have used WiFi-Direct as p2p communication technology. Mahmud *et al.* [11] categorizes nodal collaboration in fog computing into cluster, p2p and Master-Slave collaboration. In context of fog/edge nodes messaging, we have used Firebase Cloud Messaging (FCM)[1] which can be used in both p2p and cluster messaging. Although there have been messaging tools [19] and protocols [8], still most of the non-ROS (Robot Operating System) robots suffers from lack of standardized messaging protocols. Firebase broadcasts messages in real-time to all the registered devices or to a custom group. We have used JSON format in which the receiver parses the JSON message and responds to it accordingly. The body of the message might be a command to the robot or a context data received from other robots/humans.

B. *Working Together Opportunistically*

Localised work offloading, where a group of locally available devices form ‘device clouds’ and share resources to collectively complete a task, has been shown to be a viable option to complement remote cloud servers [2], [4], [10], [12], [15]. Such device clouds are needed in scenarios where the remote cloud is not accessible due to connectivity issues, not practical due to latency issues, or not feasible due to the particular task requiring specialised capabilities such as sensing. In previous work, Fernando *et al.* [4] proposed an opportunistic offloading model for mobile edge crowds, called ‘Honeybee’, for forming device clouds on the fly. In the Honeybee model, mobile devices can share work using p2p connectivity, and extensive experiments showed considerable performance gains and energy savings. Honeybee uses an adaptation of work stealing to load balance a set of independent jobs among heterogeneous mobile nodes, without any information about the resourcefulness of the participating nodes a priori. In Honeybee, the ‘delegator’ device is the originating node of the task to be completed, which is decomposed to a pool of independent jobs. The delegator shares the processing of these jobs with willing ‘worker’ devices in the vicinity, while also doing a portion of the jobs by itself. The Honeybee model also contains fault-tolerance methods to handle unforeseen disconnections, as well as mechanisms to exploit random resource node encounters, and uses Wi-Fi Direct for p2p communication among the participating nodes. In this work, we extended the existing Honeybee model and API[2] to accommodate robot-machine opportunistic collaboration, and to support a dynamic job pool where the delegator is adding new jobs in parallel with the job stealing and processing, as illustrated in Figure 3. Here, the robot has the role of the delegator, and enlists mobile devices in the vicinity to share some of its workload.

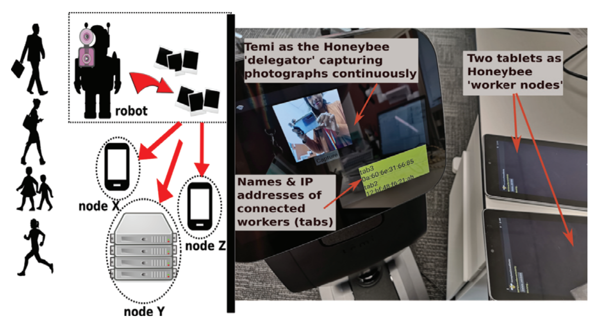

Fig. 3. On the left: A robot continuously taking photographs of people passing by, and sharing the workload of processing the photographs with nearby devices of X, Y and Z. On the right: Proof of concept prototype with the Temi robot continuously taking photographs and collaboratively processing them with two tablets

For example, consider a scenario where a child has been reported missing in a busy shopping mall, and robots are assisting with the search. As shown in the leftmost illustration in Figure 3, a robot is standing near to where the child was last seen, and continuously taking photographs of people passing by. The photographs are then run through image processing algorithms to check if the missing child appears in any one of them. To ease the robot’s workload, and to speedup the image detection, it enlists three other worker nodes in the vicinity;

[1]https://firebase.google.com/

[2]https://github.com/niroshini/honeybee/tree/robot

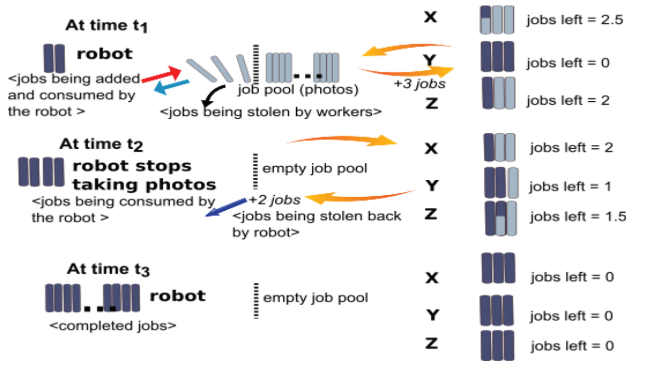

Fig. 4. Load balancing work among the robot and unknown heterogeneous worker nodes. Work stealing enables efficient sharing of workload among the collaborating devices, even if the current capacity of the devices is heterogeneous, subject to change, and unknown a priori.

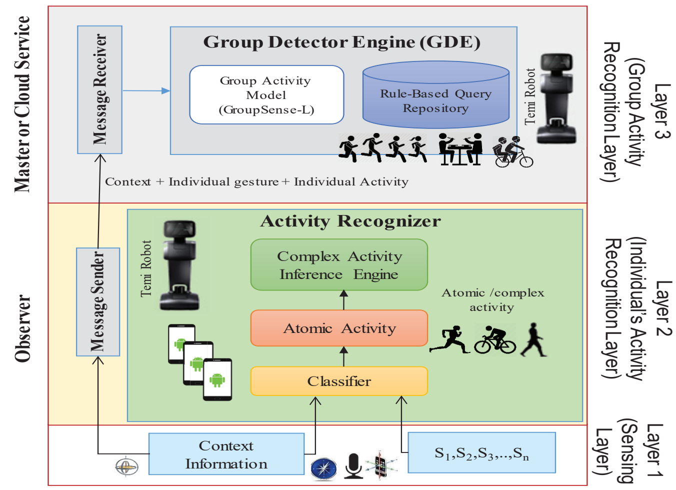

Fig. 5. GroupSense Architecture

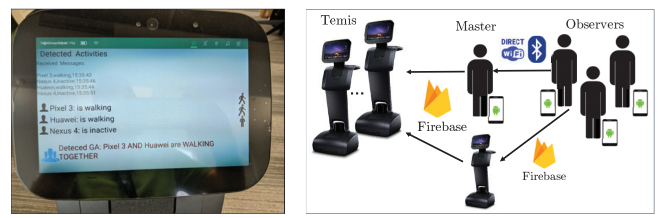

Fig. 6. Running GroupSense on Temi Robot as Master Device

nodes X and Z, which are two smart phones belonging to two people having a coffee in the food court nearby, and node Y, which is a fog computing server installed by the shopping mall. The rightmost image in Figure 3 shows a photograph of our extended Honeybee proof-of-concept application being executed on a Temi robot and two Android tablets. The application was implemented using the extended Honeybee API, where the task is to take photographs and to detect human faces in each photograph taken. Here, the Temi robot as the delegator, is continuously creating 'jobs' that are added to the job pool by taking photographs until a condition is met (e.g. keep taking photographs every 1 second for 10 minutes, or until 100 photographs are taken, or until a user notifies the application to stop). At the same time, the robot is also continuously looking for available worker devices. In the photograph showing our proof of concept prototype in Figure 3, there are two Android tablets who take the role of workers. The two tablets connect to the robot and start 'stealing' jobs from the robot's job pool, while at the same time, the robot is also consuming jobs from the same pool. This process is further illustrated in Figure 4 where a robot and three workers X, Y and Z are working together. Here, at time t_1, the robot, and the workers X and Z are busy consuming their local job lists, while the robot is also adding more jobs (e.g., photographs of passers-by) to the pool. However, Y being the strongest node, has exhausted its stolen jobs. Instead of idling, Y steals jobs from the robot's job pool. At time t_2, the robot has stopped taking photographs, and has also exhausted its job pool. Therefore, the robot then steals jobs from X, as X still has jobs to spare. Finally, at time t_3, all jobs have been completed and returned to the originating robot. In this way, load-balancing is automatic across heterogeneous devices, and the idle time of the participating nodes can be minimized despite the robot (delegator) not knowing the capabilities of the workers.

C. Group Activity Recognition by Robots

Recognizing human activities has been one of the challenging topics in context-aware computing. Scaling up from individuals to groups, that is, Group Activity Recognition (GAR), has attracted significant attention recently. In previous work, Abkenar *et al.* [1] proposed a framework named, *GroupSense* for recognizing physical group activities using multi-device embedded sensing and IoT devices. In this work we extended GroupSense to be able to interact with Temi robots using Firebase messaging. As can be seen in Figure 5, GroupSense consists of three layers. Sensing layer collects sensor data and context information using mobile IoT devices which they are called *observers*. Sensing layer transfers the data to the individual activity recognition layer in order to recognize individual activities using pre-trained models. Then the group activity recognition layer which consists of a set of rule-based queries and a rule-based reasoning engine on the *master device*, infers a group activity. *Master Device* receives the detected individual's activities from the observers and context information to be analyzed.

Figure 6 illustrates our prototype implementation of recognizing group activities by the Temi robot using *GroupSense*. In this implementation, the master device can be an Android smart device or a Temi robot which receives data messages from observers via Firebase. The Rule-based engine on Temi detects the group activities that have taken place, and can activate a service to that group accordingly. In the proposed solution the *GroupSense* can be employed to brings more context awareness to edge-fog nodes.

Listing 1. A Multihuman-Multirobot Message Protocol Examle

```
{
 "msgBuilder": "pixel3",
 "msgBuildTime": "19/11/2019 17:45:16",
 "msgSender": "pixel3",
 "msfSentTime": "",
 "msgReceiver": "",
 "msgExiryTime": "",
 "deviceType": "",
 "location: "",
 "content":[
  {
   "type": "command",
   "body": "goToKitchen"
  },
  {
   "type": "contextData",
   "body":[
     { "name": "light", "value": "30"} ,
         { "name": "temprature", "value": "22"}
    ]
   },
 ]
}
```

Tour Guide Scenario Using Robots. In section III-C, we demonstrated our prototype in which a robot can communicate with other context-aware frameworks (*GroupSense*). Let us assume a group of tourists who are visiting a museum; as part of their tour program, it is required to obtain more details about the tourists' activities. Robots (one or more) can play a tour guide role, locate that group and provide more information. Also, the robots can take the group to different places of museum; e.g., two or three robots could cooperatively "herd" the people through the different parts of the museum while tracking their collective movements and progress (via GroupSense). Moreover, if a person leaves a group, it can be detected by the nearby robot and can be tracked by another robot in the museum. The extended HoneyBee (section III-B) can be exploited to support opportunistic offloading in our Multirobot-Multihuman cooperative environment. Social and privacy concerns always have been a serious challenge in those IoT systems in which sensor data are collected from individuals. However, it is not our focus in this research work.

IV. Conclusion

Cloud robotics suffer from technical challenges such as computation, communication, and security [6]. Singh *et al.* [16] discussed the major challenges of decentralized multi-agent systems in the context of IoT. Gudi *et al.* [5] showed the effectiveness of using Fog computing in Human-Robot interaction by comparing latency. But in this work-in-progress, we envision IoT devices, wearables and robots cooperating and working together in an IoT ecosystem. We have described prototype implementations of three cooperation schemes - which are based on integrating and customizing existing edge-cloud platforms for messaging, work sharing and group activity recognition. Future work will integrate more schemes and provide a generalized distributed middleware to support a wide range of such multirobot-multihuman cooperation and explore their applications. Also, an ontology can be employed to further improve our context information sharing [3].

Acknowledgments

This research was supported partially by the Australian Government through the Australian Research Council's Linkage Infrastructure, Equipment and Facilities Grant funding scheme (project LE180100158). The views expressed herein are those of the authors and are not necessarily those of the Australian Government or Australian Research Council.

References

[1] Amin B Abkenar, Seng W Loke, Arkady Zaslavsky, and Wenny Rahayu. Groupsense: Recognizing and understanding group physical activities using multi-device embedded sensing. *ACM Transactions on Embedded Computing Systems (TECS)*, 17(6):98, 2019.

[2] Vinay Chamola, Chen-Khong Tham, and GS S Chalapathi. Latency aware mobile task assignment and load balancing for edge cloudlets. In *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 587–592. IEEE, 2017.

[3] Everton de Matos, Ramão Tiago Tiburski, Carlos Roberto Moratelli, Sergio Johann Filho, Leonardo Albernaz Amaral, Gowri Ramachandran, Bhaskar Krishnamachari, and Fabiano Hessel. Context information sharing for the internet of things: A survey. *Computer Networks*, 166:106988, 2020.

[4] Niroshinie Fernando, Seng W Loke, and Wenny Rahayu. Computing with nearby mobile devices: a work sharing algorithm for mobile edge-clouds. *IEEE Transactions on Cloud Computing*, 2016.

[5] Siva Leela Krishna Chand Gudi, Suman Ojha, Benjamin Johnston, Jesse Clark, and Mary-Anne Williams. Fog robotics for efficient, fluent and robust human-robot interaction. In *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)*, pages 1–5. IEEE, 2018.

[6] Guoqiang Hu, Wee Peng Tay, and Yonggang Wen. Cloud robotics: architecture, challenges and applications. *IEEE network*, 26(3):21–28, 2012.

[7] Ben Kehoe, Sachin Patil, Pieter Abbeel, and Kenneth Goldberg. A survey of research on cloud robotics and automation. *IEEE Transactions on Automation Science and Engineering*, 12:1–12, 04 2015.

[8] Anis Koubaa, Maram Alajlan, and Basit Qureshi. *ROSLink: Bridging ROS with the Internet-of-Things for Cloud Robotics*, pages 265–283. Springer International Publishing, Cham, 2017.

[9] Seng W. Loke. Robot-friendly cities, 2019.

[10] Seng W Loke, Keegan Napier, Abdulaziz Alali, Niroshinie Fernando, and Wenny Rahayu. Mobile computations with surrounding devices: Proximity sensing and multilayered work stealing. *ACM Transactions on Embedded Computing Systems (TECS)*, 14(2):22, 2015.

[11] Redowan Mahmud, Ramamohanarao Kotagiri, and Rajkumar Buyya. Fog computing: A taxonomy, survey and future directions. In *Internet of everything*, pages 103–130. Springer, 2018.

[12] Radu-Corneliu Marin, Alexandru Gherghina-Pestrea, Alexandru Florin Robert Timisica, Radu-Ioan Ciobanu, and Ciprian Dobre. Device to device collaboration for mobile clouds in drop computing. In *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 298–303. IEEE, 2019.

[13] Michael Nagenborg. Urban robotics and responsible urban innovation. *Ethics and Information Technology*, Jan 2018.

[14] Pericle Salvini. Urban robotics: Towards responsible innovations for our cities. *Robotics and Autonomous Systems*, 100:278 – 286, 2018.

[15] Mahadev Satyanarayanan, Victor Bahl, Ramón Caceres, and Nigel Davies. The case for vm-based cloudlets in mobile computing. *IEEE pervasive Computing*, 2009.

[16] Munindar P Singh and Amit K Chopra. The internet of things and multiagent systems: Decentralized intelligence in distributed computing. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pages 1738–1747. IEEE, 2017.

[17] Ajay Kumar Tanwani, Nitesh Mor, John Kubiatowicz, Joseph E. Gonzalez, and Ken Goldberg. A fog robotics approach to deep robot learning: Application to object recognition and grasp planning in surface decluttering. In *Proc. of IEEE Intl Conf. on Robotics and Automation (ICRA)*, pages 1–8, 2019.

[18] Ilaria Tiddi, Emanuele Bastianelli, Enrico Daga, Mathieu d'Aquin, and Enrico Motta. Robot–city interaction: Mapping the research landscape—a survey of the interactions between robots and modern cities. *International Journal of Social Robotics*, Mar 2019.

[19] R. Toris, J. Kammerl, D. V. Lu, J. Lee, O. C. Jenkins, S. Osentoski, M. Wills, and S. Chernova. Robot web tools: Efficient messaging for cloud robotics. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4530–4537, Sep. 2015.