Defining Quality of Services for Remotely Controlled Robotics Devices under Intermittent Communication Environment

Arata Koike[1] and Yoshiko Sueda[2]
[1] Tokyo Kasei University, Tokyo 173-8602, Japan
[2] Meisei University, Tokyo 191-8506, Japan
Email: koike@ieee.org; yoshiko.sueda@meisei-u.ac.jp

Abstract—Remotely controlled robotics devices require connectivity to a remote server to maintain the stability of control. We implicitly assume that we can utilize such always-on connectivity and availability over the existing mobile Internet. This always-on property will be lost when the device moves out from coverage area of a wireless network, and thus we cannot control the device unless the device returns to the coverage area. Many mobile carriers made a tremendous investment to extend the coverage area and we can now enjoy the mobile network spread across almost anywhere in our nation. As long as we focus on our smartphone application, we do not feel interruption of connectivity during communication. Our question here is that if a remotely controlled robotics device can perceive the same way as human feels for our mobile networks. We investigated the behavior of a robotics device that requires continuous high frequency feedback-based controls between the device and a remote server over the mobile Internet. Our experiments show that robotics devices experience large impact by delay variation by handovers from the viewpoint of robotics control. The delay variation does not have a big influence on our daily smartphone usage so we perceive that our mobile network can provide always-on connectivity. On the other hand, for remote robotics control, the delays effectively create interruptions of connectivity and thus we should handle the underlining network as a network with intermittent connectivity. Based on the observation above, we propose metrics to evaluate such intermittency for our robotics applications.

Index Terms—Intermittent, robotics control, quality metric, simulation, measurement

I. INTRODUCTION

Digital transformation is rapidly spreading in various industries to change their old-fashioned analogue technologies and business styles. Industries relying on analogue processing technologies or business styles required engineers with highly skilled craftmanship to integrate various parts or processes to make them a complete product or a system. It is reported that such integral skills protect the industries and their business models from new business players with a different industrial background [1]. Digital transformation is replacing such integral skills and knowledge, which were acquired by engineers after their long-time experiences, with digital processing system using artificial intelligence. The progress of digital transformation thus reduces bars to challengers from different industrial segments. It also produces a new concept of business which connects industrial domains. Internet of Things (IoT) supports and accelerates the changes above. We can utilize various IoT-based applications and services via the Internet. For example, by collecting data periodically from IoT devices, we can realize personal e-health services for mass consumers. Mobile networks promote this movement too since they cover almost all populated areas across the globe so that we can keep connecting to the Internet and utilizing the data from the IoT devices. Mobile Internet can further assist the construction of IoT-based data collection infrastructures for our society from sensors disseminated broadly; by using data from sensors installed along a road, we can monitor road traffic status and that from sensors in electric meters, we can check power consumption status in a vicinity.

We implicitly assume always-on connectivity to the Internet when we use or think these applications and services. We feel this always-on environment in our daily usage by mobile networks and we seldom encounter out-of-coverage area. Such feelings further populate IoT solutions to tackle various social problems and we further depends on always-on communication environment.

When we look at robotics devices, the number of devices that are controlled over network is gradually increasing. Such robotics devices vary from location fixed robot arms and remote surgery robots to movable robotics devices such as delivery robot, drone and autonomous driving cars. Our focus is such remotely controlled robotics devices. We especially target movable robotics devices. In this paper, we contribute on defining quality metrics for remotely controlling robotics devices that move around. We point out that we could not assume always-on communication environment from the view point of controlling robotics devices. One typical situation is that if we look at Unmanned Aerial Vehicles (UAV) or drones, they will move away from cellular or wireless coverage areas as those antennas are not directed

Manuscript received November 20, 2019; revised March 30, 2020.
Part of this work was supported by JSPS KAKENHI Grant Number JP 18K11279.
Corresponding author email: koike@ieee.org.
doi:10.12720/jcm.15.5.427-432

to the height of 150m where such devices move. The UAVs will be disconnected during flight so that we will not be able to send a message at that duration unless they return to the coverage area. From the viewpoint of the robotics devices, the communication environment is intermittent. We also point out that there are other intermittency for communication environment for remotely controlling robotics devices even if we do not perceive it for our daily usage for smartphone applications. Intermittency will create problems on stability for controlling robotics devices. We discuss such problems and propose new quality metrics to grasp communication environment for robotics devices.

This paper has the following structure. In Section II, we briefly explain related technologies and researches. Section III explains our model of remotely controlled robotics devices and discuss their stabilities. Section IV shows the measurement results of delay stability on a mobile network and we point out the existence of intermittency for such network from the viewpoint of controlling robotics devices due to handover process. In Section V, we propose quality metrics to characterize the intermittency based on the observation on the previous section. We conclude this paper in Section VI.

II. Related Technologies and Researches

In this section we provide brief explanations and reviews for Intermittent Communication, Publish/Subscribe (Pub/Sum) messaging, and Robot Operation System (ROS).

A. Intermittent Communication

Communication in an environment where constant connectivity is not guaranteed is called intermittent communication. In the days when computer hosts were regularly connected by dial-up modems, many people used intermittent communication to get the access to the rest of the Internet and to deliver or receive email. In this era, end-to-end information distribution was achieved using the Unix-to-Unix Copy Protocol (UUCP) [2] to deliver e-mail and news.

It is necessary to reduce battery consumption in a network of low power wireless environments where IoT sensors use. There, IoT devices periodically change to a low power consumption mode, and their communication functions are suspended during that period. In recent years, research on communication between devices or between devices and servers has been conducted as intermittent communication in this situation [3].

In this paper, we discuss the intermittent communication caused by the movement of robotics devices. At the best of our knowledge, there are no studies on the intermittent communication from this viewpoint.

B. Publish/Subscribe(Pub/Sub)

The Publish / Subscribe (Pub/Sub) system is a method for transmitting information by message exchange between the sending side and the receiving side in a distributed system [5]. In this method, it is possible to link the information sender (Publisher) and the receiver (Subscriber) in a loosely coupled form that can maintain independence from each other. Specifically, in the topic type Pub/Sub method, the Subscriber side registers in advance (Subscribe) the information (Topic) provided by the Publisher. Every time the Publisher side generates information, the information is delivered to the Subscriber side asynchronously. Unlike the request/response type communication, the PUB/SUB system does not need to wait for a response and does not depend on the other peer's state. Therefore, it is used in various cases as an efficient message exchange method in a large-scale distributed system.

The PUB/SUB type is a system in the application layer. Therefore, there are no specific requirements for lower-layer networks. The advantage of this is that it can be applied to a stored-forward type architecture such as an overlay network. In other words, the Publisher can publish information at its own timing without being aware of the Subscriber's status, and the Subscriber can also receive that information at the timing when the information arrives without being aware of the Publisher's status. In [4], PUB/SUB is used for Delay Tolerant Network (DTN) environment in combination with Information Centric Networking (ICN) technology. Also, in [5], a PUB/SUB system is used for application to information transmission using data mules in disaster-stricken areas as an application of ICN.

C. Robot Operating System (ROS)

Developers are accepting Robot Operating System (ROS) [6] as their middleware framework for their implementation. ROS adopts Pub/Sub architecture for its method for communication among functions consisting the software. ROS-based functions form an entire software a robotics device. These functions are loosely coupled each other. Here, “loosely coupled” means that each function notifies its status by sending messages to its peers while it works independently from other peers. Request/Response type transaction, such as Remote Procedure Call (RPC), a process that sends a Request message might keep waiting for its processing to end until it will receive a corresponding Response message. This may suspend the movement of a robotics device since Request/Response type communications couple functions tightly each other. Pub/Sub type messaging communication, on the other hand, can avoid such Head of Line (HOL) blocking as their nature of loose coupling a robotics device can keep working while a part of its function is suspended.

Robotics devices embed their functions within their body and ROS based functions exchange PUB/SUB messages within their chassis to control the movement of the robotics device. We proposed and demonstrated an architecture called “Cog Network” to split parts of functions out from the body of the robotics device using virtualization technology and place then at edge or remote cloud servers [7]. In this paper, we make the same assumption that is based on the concept of Cog Network architecture, part of functions consisting a robotics device are placed on edge cloud servers. Since ROS-based systems use PUB/SUB messaging, we can send such messages across the Internet. For example, a camera function on a robotics device publishes taken images; a processing function on a cloud subscribed the images; then a commanding function on the cloud publishes directional messages based on the received image information; finally, a mobility management function on the robotics device subscribes the message to change its movement.

D. Architecture of Remotely Controlled Robotics Devices

Robotics devices are usually equipped with actuators and sensors. These sensors and actuators constitute an internal feedback function in cooperation with each other. This feedback function is realized by frequently generating commands for operating the actuator based on feedback generated by the robotics device based on information measured by the sensor. This mechanism is also working in autonomous driving vehicles and flying drones, ensuring the safety of automobiles and correcting their flight routes. For example, robotics devices monitor their surroundings with high-resolution videos; thus, their video images are the sensed information and their operation is performed based on the image information from the camera during autonomous moving. They always check the surrounding environment and immediately determine how to change movement to avoid danger using the information. In video cameras, information is generated at a high frequency, for example, 30 frames per second (30 Hz), that is, one frame every 33 milliseconds. The internal processor generates command information for operation at almost the same frequency by processing image information. The intelligent information processing robot that performs such high-frequency feedback is the target of this paper. That is, the behavior of the application that operates our robotics devices is determined by the control commands generated by the remote-control server on an edge cloud. The robotics devices also send the video image stream to the remote server to convey the feedback results of the control based on the command.

In such an application, we need a stable always-on connectivity. If a large delay suddenly occurs, this greatly affects the control stability of the robotics devices. It is likely that the devices will malfunction. Therefore, in Section III.B, we first consider by simulation how the robot arm, as an example of robotics devices, is controlled when a large delay occurs.

E. Evaluation of Stability of Robotics Device

We conducted simulation using NS-3 [8] for the evaluation of movement of robotics device. Fig. 1 shows the configuration of a robotics device for evaluation; there is a robot arm that operates as an actuator, a video camera as a sensor, a network and a server in a cloud. The robotics device moves the arm toward the target position based on the command by feedback control. The robotics device then sends the video images to the control function on the cloud side. In response to the result, the control function also sends a new command to the robot arm to adjust its position. Fig. 2 shows this model. We use the TCP protocol implemented in NS-3 as the transport layer protocol (This corresponds to the ROS implementation). The video frame size was 200 Kbytes, and the TCP MTU size was 1.5 Kbytes. It is assumed that TCP segments carry 200 bytes of messages containing control commands. Robotics devices publish video frames every 20ms. The link speed of the network was 1Gbps. The delay between the device and the network is 3ms, and the delay between the network servers is 3ms. In addition, as a background load, 30 TCP connections are applied to the access links.

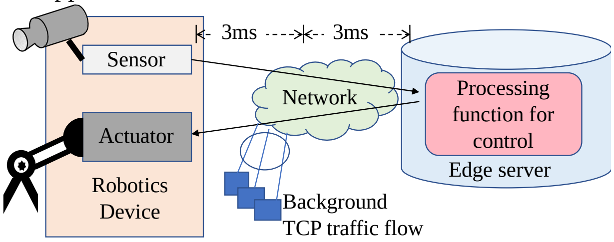

Fig. 1. System configuration.

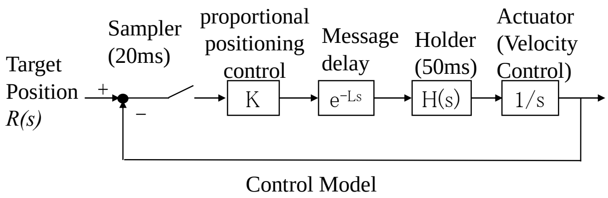

Fig. 2. Control model.

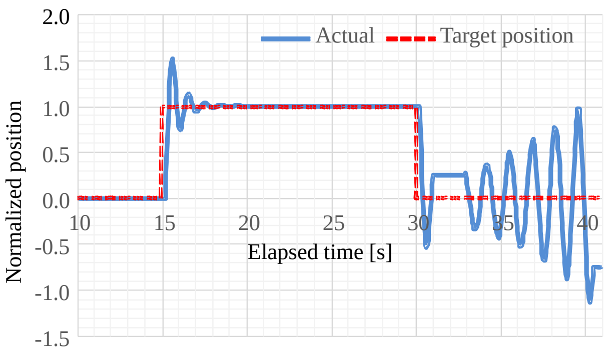

Fig. 3. Result of position control.

Figs. 3 and 4 show the simulation results. Fig. 3 shows how the arm follows the control commands to move to the target position. Fig. 4 shows the response delay, cumulative received message count, and lost message count. The unit of the horizontal axis is seconds. Response delay occurred immediately after startup (around 15 seconds) due to the influence of TCP slow start, but it can be seen from Fig. 4 that the response delay characteristics have stabilized to about 100ms for a while after that. Also, looking at the corresponding time part in Fig. 3, we can see that the arm (y (t)), solid line, has also quickly converged to the target position (r (t)), dashed line, because of the control. However, a sudden response delay of about 2,000ms occurred suddenly after 30 seconds. As a result, it was found that the position control also lost control and did not converge, and the position of the arm diverged although the response delay itself stabilized around 250ms thereafter.

From these results, we can find that the momentary increase in response delay has a significant effect on remote control of robotics devices.

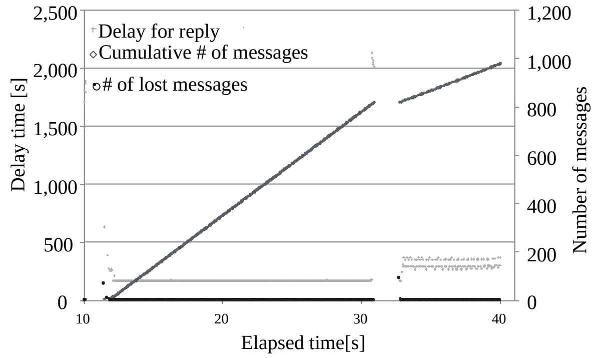

Fig. 4. Reception Status at robot arm.

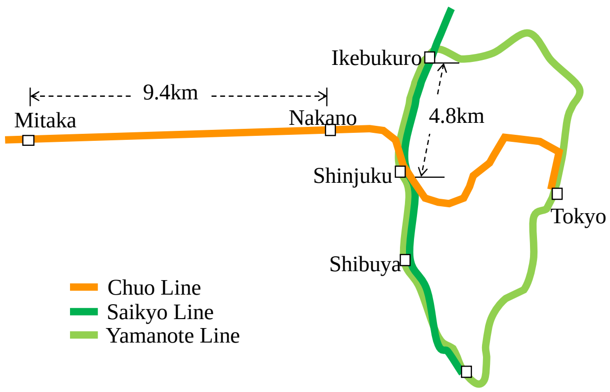

Fig. 5. Note how the caption is centered in the column.

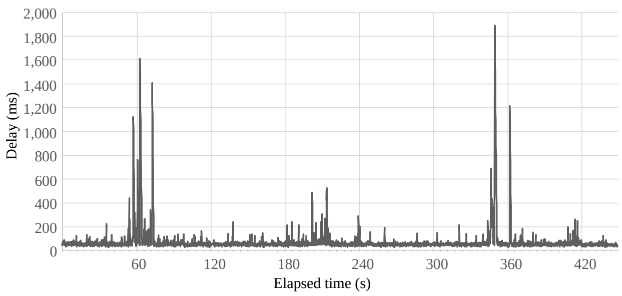

Fig. 6. Ping measurement on train measurement (Nakano-Mitaka).

III. Measuring Mobile Network Stability

As a use case, we assumed a situation in which a robotics device moves at a constant speed in an urban area. In this scenario, we investigate the stability of the mobile network by generating a ping command every 0.1 seconds while the train was running. For ping target, we targeted the web server www.tokyo-kasei.ac.jp (hosted by Akamai). Fig. 5 shows the two train segments we performed our measurements on. These two segments are one of busiest segments in the Tokyo Metropolitan area. Chuo Line connects suburbs of Tokyo to the downtown area. Ikebukuro and Shinjuku are two major terminal stations in downtown Tokyo, which is connected by the Saikyo Line. Fig. 6 shows the results of running on the Chuo Line Special Rapid Train between Nakano station and Mitaka station (non-stop 9.4km segment), and Fig. 7 shows the results of running on the Saikyo Line between Ikebukuro and Shinjuku (4.8km non-stop segment). In these figures, the round-trip delay time is almost stably around 50ms. However, we can observe that there are some delays up to about 50 times the normal number (2 in Fig. 6 and 1 in Fig. 7). We inferred that these large delays are the delays caused by traversing the management boundary of Mobility Management Entity (MME) because of handover. Besides these large delays, low delay peaks exist almost periodically. We thought that these may be due to handover accompanying switching of eNodeB in the same MME. From these results, we can see that there is a delay several tens of times larger than usual, especially when it crossed the management boundary of MME. From the viewpoint of remote control of robotics devices, we can identify that the always-on connectivity that required to maintain control is lost in this duration. Namely, when handover is involved in a mobile network, we can see that communication has intermittency in terms of control of robotics devices. It is a delay that has almost no effect when a human operates a smartphone, etc., and the connection is always maintained for human operation. On the contrary, we can see that applications such as remotely controlled robotics devices are clearly in different situations even with the same communication characteristics.

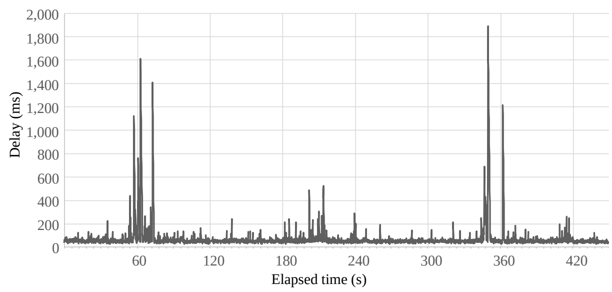

Fig. 7. Ping measurement on train measurement (Ikebukuro-Shinjuku).

IV. Proposal of Quality Measure for Evaluation of Intermittency Associated with Handover

Based on the results in the previous section, we propose metrics related to intermittency as a new quality index from the perspective of a social infrastructure that supports the operation of robotics equipment based on the use of PUB/ SUB message communication.

As an index for system evaluation, the downtime is well known as a measure of reliability. Based on this idea, we can define an index of how much intermittency occurs in the communication environment. First, it is necessary as a quality measure when it becomes impossible to communicate or how long it can keep continuing the connectivity. We can define them as mean offline time and mean online time, respectively. In order to grasp the characteristics in detail, we must consider not only the mean but also the distribution. However, considering the situation of moving at a constant speed while using mobile communication, of course it depends on the configuration of the base station, it may be possible to encounter an offline section at a relatively similar timing due to the design of cellular system. Thus, as an index, we can assume that this mean online time and mean offline time should be appropriate as the bottom line. Moreover, by using mean online time and mean offline time, we can form an index called intermittency ratio. More specifically, we can define the intermittency ratio as

$$\begin{aligned} &\textit{Intermittency ratio} \\ &= \textit{mean offline time} \\ &\quad / (\textit{mean online time} + \textit{mean offline time}). \end{aligned} \tag{1}$$

This intermittency ratio can be used to show how much communication is impossible on average in the overall connectivity. The larger this value, the shorter the communication time. Conversely, if this value is small, we can assume that communication is possible almost stably. Therefore, we can use it as an indicator of whether or not we can use a robotics device that requires real-time control.

From the viewpoint of controlling moving robotic devices, we treat the duration when the delay due to handover is increased as the offline time since such duration affect the control so much as we saw in the Section III. The amount of delay that should be taken depends on the application because it depends on how the control model is built. For example, in the case of the model shown in the simulation in Section III, the average delay time is about 100ms, but the arm position diverges after a delay exceeding 2,000ms occurs. Therefore, we can set at least 2,000ms as a threshold.

Next, we define the following index to measure the impact on data transmission when using intermittent communication.

$$\begin{aligned} &\textit{Intermittency index} \\ &= \textit{transmission data size} \\ &\quad / (\textit{transmission speed} * \textit{mean online time}). \end{aligned} \tag{2}$$

If this value exceeds 1, we are not able to complete the transmission of the information during online. When considering the control of robotics devices, the size of the video frame, not the command information for control, is affected if the environment changes Online / Offline in a short time. However, in the case of handover a certain level of communication speed is secured, so this effect is unlikely. As described in [9], a robotics device, such as a drone, offloads information processing, which is necessary for subsequent autonomous control, to the edge server in a short period of time when the drone stays in a coverage area. The processing result needs to be available to the drone during its stay in a coverage area. This indicator is important to decide offloading whether it can retrieve the result while it is in a coverage area.

On the other hand, if the index is less than 1, we can complete the transmission of information while connectivity is maintained. Therefore, if there is no need for immediate transmission, we can keep availability of the connection. However, if a communication requires immediacy, we have to take into account whether the waiting time is within the allowable range even if the index is less than 1. It also applies to the robot arm in this paper.

When investigating a control system, it is necessary to consider in advance how much the sensing information and feedback information that are continuously generated are sensitive to delay. For that purpose, the various quality indicators mentioned above are useful to evaluate the appropriateness of the underlining network.

V. Conclusions

When controlling robotics devices remotely, we usually assume always-on network. However, in terms of the stability of robotics device control, we encounter an intermittency problem caused by handover. This problem is different from a clear communication interruption caused by well-known out of communication coverage problem and have never mentioned in previous researches for the context of robotics control. In this paper, we illustrated it by both simulation and measurement. Then, we proposed a new quality metrics for evaluating network intermittency to support remotely controlled robotics devices.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Both AK and YS jointly planned and conducted this research. Both of us jointly performed and investigated the work in Section III while AK performed the experiment in Section IV. Both of us discussed and studied the metrics in Section V. AK wrote this manuscript and YS reviewed the text and both of us revised the contents.

Acknowledgment

Part of this work was supported by JSPS KAKENHI Grant Number JP 18K11279.

REFERENCES

[1] T. Fujimoto "Architecture, Capability, and Competitiveness of Firms and Industries," Discussion paper *CIRJE-F-182*, Center for International Research on the Japanese Economy (CIRJE), The University of Tokyo Nov. 2002.

[2] Bell Telephone Laboratories, *Unix(TM) Time-Sharing System: Unix Programmer's Manual*, Seventh Edition, Volume 1, Murray Hill, New Jersey: Bell Telephone Laboratories, Incorporated, 1979.

[3] Y. Kantaros and M. M. Zavlanos, "Distributed intermittent connectivity control of mobile robot networks, " *IEEE Trans. on Automatic Control*, vol. 62, no. 7, pp. 3109-3121, Jul. 2017.

[4] S. Carrilho and H. Esaki, "A pub/sub message distribution architecture for disruption tolerant networks," *IEICE Trans Inf. & Syst.*, vol. E92-D, no. 10, pp. 1888-1896, Oct. 2009.

[5] E. Monticelli, B. M. Schubert, M. Arumaithurai, X. Fu, and K. K. Ramakrishnan, "An information centric approach for communications in disaster situations," in *Proc. 20th IEEE International Workshop on Local & Metropolitan Area Networks*, 2014, pp. 1-6.

[6] Robot Operating System. [Online]. Available: http://www.ros.org

[7] N. Higo, Y. Sueda, and A. Koike, "COG: Overlay network functions assisting COnnected next generation society systems," in *Proc. 23rd IEEE International Symposium on Local and Metropolitan Area Networks*, 2017.

[8] NS3. [Online]. Available: https://www.nsnam.org

[9] A. Koike, "Infrastructure for supporting networked robotics system relying on intermittent communication," in *Proc. IEEE International Conference on Big Data, Cloud Computing, Data Science & Engineering*, 2019.

Arata Koike Arata Koike was born in Tokyo, Japan in 1964. He received B.S. and M.S. degree in Physics from St. Paul's (Rikkyo) University, Tokyo, Japan in 1989 and 1991, respectively. He received Ph. D in Electro Engineering from the University of Tokyo, Japan in 2009. He joined Nippon Telegraph and Telephone Corp. (NTT) in 1991, and conducted research and development of Personal Handy-phone System, Asynchronous Transfer Mode (ATM), the Internet, and Machine-to-Machine (M2M) protocols and their architectures. From 2017, he has been working at Tokyo Kasei University, and is currently serving as a professor on Informatics at the Faculty of Humanities.

Yoshiko Sueda was born in Hiroshima, Japan in 1969. She received B.S. and M.S. degree in Physics from Tokyo Denki University, Tokyo, Japan in 1993 and 1995, respectively. She received Ph. D in Engineering from Shibaura Institute of Technology, Japan in 2008. She joined Nippon Telegraph and Telephone Corp. (NTT) in 1995, and conducted research and development of Advanced Intelligent Network, communication services, and Machine-to-Machine (M2M) protocols and their architectures. From 2018, she has been working at Meisei University, and is currently serving as an associate professor on Informatics at the School of Information Science.