IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 20, 2017, accepted March 24, 2017, date of publication April 24, 2017, date of current version June 7, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2696027

COSAP: Contract-Oriented Sensor-Based Application Platform

TAKUMA OIDE[1,2], TORU ABE[3], AND TAKUO SUGANUMA[3], (Member, IEEE)
[1]Japan Society for the Promotion of Science, Tokyo 102-0083, Japan
[2]Graduate School of Information Sciences, Tohoku University, Sendai 980-8577, Japan
[3]Cyberscience Center, Tohoku University, Sendai, Japan

Corresponding author: Takuma Oide (oide@ci.cc.tohoku.ac.jp)

This work was supported by JSPS KAKENHI under Grant 15J09912.

ABSTRACT Research and development on participatory sensing that accumulates a large amount of data acquired from users on the cloud and handles them as big data has proceeded due to the spread of sensor devices with miniaturized and advanced functions typified by smartphones. However, in service configuration models based on the current participatory sensing for which the cloud is a core component, there is a problem in that it is not possible to flexibly distribute information according to the data provisioning policy. In this paper, we propose a new sensor-based application platform based on a service configuration model that does not use a server or a cloud. In addition, we design and implement a contract-oriented information flow protocol, which realizes flexible reflection of the provisioning policy on that platform. Furthermore, we discuss feasibility and scope of the proposed protocol through simulation experiments.

INDEX TERMS Contract protocol, network service platform, participatory sensing, contents flow, mobile ad-hoc network, multi agent system, ubiquitous computing.

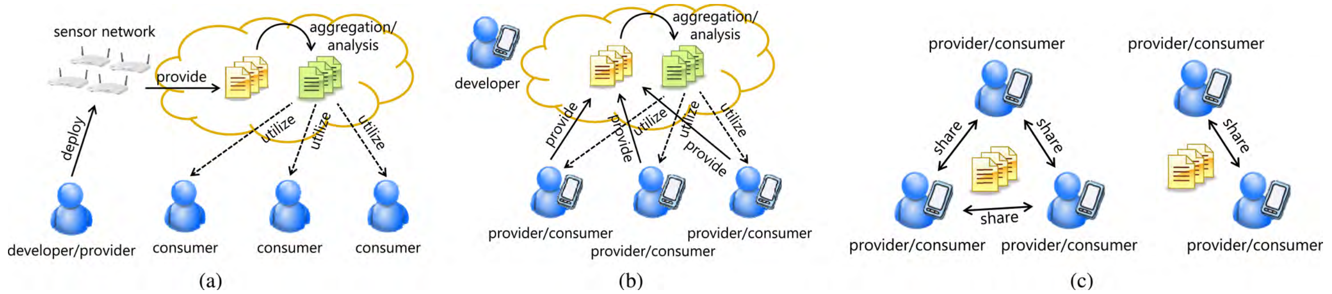
I. INTRODUCTION

Due to the widespread use of miniaturized and highly functional sensor devices, development of sensor-based applications that uses data acquired from these devices is underway. Examples include agricultural support systems [1], that monitor environmental information such as temperature, humidity, luminance, and rainfall to improve the efficiency of agricultural work; Home Energy Management System (HEMS) [2], that monitors and manages the power consumption of each appliance in a home network; and healthcare systems [3], that detect abnormal activities from physiological information such as heart rate and blood pressure. Therefore, the scope of these kinds of applications, recently called IoT applications, is diverse.

In this paper, we focus on service configuration model for sensor-based applications. In conventional applications [1]–[5], service configuration models are vertically integrated so that data acquisition is exclusively made by installing a sensor for each service as shown in Fig. 1(a). For this reason, there is an inefficient use of sensor and network resources. In addition, sensor data is acquired from the predetermined devices within a fixed frequency, spatial resolution, and accuracy, and then provided to users. Therefore, the realization of a user-oriented service that changes the service configuration according to the user's request is limited.

Consequently, in recent years, to effectively use the resources of sensor equipped devices, such as smartphones, research on mobile crowdsensing [6] has been conducted. Mobile crowdsensing, which utilizes a people-centric networking paradigm [7]–[9], is a form of sharing sensor data acquired from sensor devices owned by users, and can be broadly classified into two categories: Participatory Sensing [10] in which users themselves consciously provide sensor data, and Opportunistic Sensing [11], [12] in which users share sensor data based on different properties such as geographical or social relationship.

In the service configuration model of applications based on Participatory Sensing [13]–[17], users act as both provider and consumer of the sensor data as shown in Fig. 1(b). Sensor data acquired from users is stored on the *clouds*, and provided to users based on their requests. Therefore, service providers do not need to install a sensor device for each service, and it is possible to efficiently use sensor and network resources. However, it is difficult to grasp the amount of data that satisfies user's requirements, and moreover how much of that data is disseminated, which makes data distribution management more difficult. In addition, it is necessary to manually identify



(a) (b) (c)

FIGURE 1. **Service configuration model of applications based on (a) the conventional model, (b) participatory sensing model, and (c) opportunistic sensing model. (a) Conventional Model. (b) Participatory Sensing Model. (c) Opportunistic Sensing Model.**

which tasks can be provided, which is a great obstacle to motivate data provisioning.

On the other hand, in the service configuration model of applications based on Opportunistic Sensing, the information is shared in an ad-hoc manner through a very unstable Device-to-Device (D2D) network, composed by sensor devices owned by users as shown in Fig. 1(c). Therefore, users can share information without requiring special infrastructure or complicated operations. However, as in the previous case, the data distribution management is also difficult. Therefore, we need a new service configuration model such that reduces the disadvantages while using the best asset of both models, and a software platform to promote system development based on the new model.

In this research, we propose the Contract-Oriented Sensor-based Application Platform (COSAP) to share sensor data among users without requiring special infrastructure and operation. With COSAP, users are connected to each other by the Peer-to-Peer (P2P) network, and data is distributed by establishing a temporary relationship based on the ''contract''. The consumer's request and data provisioning policy of providers are modeled and registered on COSAP. Based on the policy, the data distribution route is dynamically determined by automatic negotiation among agent components by using the Contract-oriented Information Flow Protocol (CIFP). Therefore, COSAP allows highly efficient distribution of sensor data to all users while reflecting the user policy. Furthermore, COSAP needs no central server nor cloud infrastructure, the only requirement is the installation of COSAP on each user's device; therefore, it is easy to deploy highly on-demand applications for sharing data among other users in comparison with the traditional models.

We have proposed the basic concept of COSAP and CIFP in a previous publication [18]. The present paper first outlines COSAP. Next, we will design the CIFP to realize the sensor data distribution based on the policy of the provider. We will also discuss the feasibility and scope of CIFP through simulation experiments.

This paper is organized as follows: In Section I, the background and general outline of this research was shown. In Section II, research on conventional sensor-based applications and the sharing platforms of sensor data will be cited and their characteristics and problems will be described. In Section III, we describe a Contract-Oriented Sensor-based Application Platform for solving the evoked issues and the assumed environments. In Section IV, the design of the Contract-oriented Information Flow Protocol used in the proposed platform will be described in detail. In Section V, the results of simulation experiments conducted to confirm the effectiveness of the proposed protocol and its evaluation are described. In Section VI we will discuss the conclusion of this research and future challenges.

II. RELATED RESEARCH AND CHALLENGES

A. SENSOR-BASED APPLICATION

Research on sensor-based applications that utilize the enormous amount of sensor data acquired from sensor devices is being conducted in various fields [1]–[5]. In research on home networks [4], information obtained from sensors installed in homes is aggregated and analyzed by a server, and autonomous home appliance control is performed based on the results. In the research targeting weather data [5], applications that aggregate and share observation data for each region have been proposed. In the service configuration model of these studies, the system collects sensor data from known sensor devices installed in advance, and the acquired sensor data is exclusively used by the system. Meanwhile, efforts to disclose data held by government agencies and municipalities as open data have also progressed in recent years [19]–[21]. In these efforts, there are spatiotemporal restrictions on services constructed using these data to provide sensor data acquired by the provider with fixed frequency and accuracy.

In recent years, research and development of sensor-based applications based on the participatory sensing model [10] has been conducted to solve these problems. Predić *et al.* [22] proposes a life log system that links up with an existing participatory sensing network [23] that acquires positional information and acceleration information from a user's smartphone, performs behavior estimation, and shares atmospheric information such as CO_2 concentration. In addition to this, the use of sensor data-like provision of air pollution maps [15] created by collecting environmental information and noise maps [13], [14] created by collecting noise information, which are obtained from small-sized highly functionalized sensor devices such as smartphones, is being promoted.

However, in these systems, when acquiring data from users, they are fixedly collected without considering the provider's data provisioning policy. Moreover, to respond to various user demands, acquisition of data is performed with high frequency and high accuracy. Therefore, in addition to the possibility of providing unnecessary sensor data, there is a problem that the power consumption of the smartphone owned by the user increases. Also, with these systems, it is difficult for the provider to know the use of the data provided by the provider themselves. Therefore, there is a limit to the realization of an application that uses personal data with a higher level of privacy, such as continuous movement history, purchase history, usage history of public transportation, as well as positional information, and noise information.

B. DATA SHARING PLATFORM

To solve the problem raised in the previous section, research is being conducted on a data sharing platform that builds sensor-based applications on demand upon request from users and distributes only the necessary and sufficient data [16], [17], [24]. In the platform based on the participatory sensing model [16], [17], the data information that the user wishes to collect is registered as a task on the platform freely on its own. Other users who viewed this task information exchange sensor data via the platform; however, with these platforms, it is necessary to view the task information of others when data is provided, and the time and effort has increased compared with the use of a conventional sensor-based application that provides fixed data. It is important to motivate and encourage users to provide data in participatory sensing. However, it is conceivable that the fact that discovery of the tasks that it can provide itself is manual, and the fact that changes in the data distribution setting due to the change in the environment or its own provisioning policy being manual is a great obstacle to providing motivation.

C. CHALLENGES IN RELATED RESEARCH

Conventional sensor data usage models include fixed and exclusive applications that acquire data [4], [5] and data stores [19]–[21], but service providers led them to collect and provide sensor data. For this reason, there is a risk of data distribution in which the policy of the owner of the sensor device is not sufficiently reflected, which may lead to unnecessary consumption of resources and disclosure of undesirable personal data. On the other hand, in a data sharing platform that shares data on-demand by request from other users [16], [17], [24], the user who is the data provider leads and shares sensor data, and distribution of sensor data that reflects the policy of the owner is thus made possible. However, it is troublesome to find a task that matches the policy and to follow changes in policies and environments; consequently, there is a risk that the data provision itself from the user will not be performed. From the above, it can be said that the configuration model of the conventional sensor-based application has limitations on dynamic sensor data utilization which considers the provider's policy.

In this research, we model the requests from consumers and policies of the providers, register them on the platform, and realize the distribution of data by automatic negotiation using the Contract-oriented Information Flow Protocol between agents of users that hold these as knowledge. We summarize the problems in the existing research targeted by the proposed Contract-oriented Information Flow Protocol are organized as follows.

(P1) Distribution setting of sensor data according to the provider's policy is difficult: Since data generated from sensor devices owned by users includes privacy-sensitive data such as personal data and because power consumption costs, etc., are incurred when acquiring the data itself, the distribution of data that considers the policy of the provider and privacy, etc., is necessary; however, with the data-sharing approach in existing research, it was necessary for the data provider to determine the quality of the data themselves, and to discover the requirements satisfying the policy. These increase the burden on the providers. For this reason, it is difficult to distribute sensor data based on the provider's policy, which provides sensor data by limiting the scope of disclosure and usage to the provider without imposing a burden on the provider.

(P2) It is difficult to dynamically control the service according to changes in the provider's policy: In the service realized by distributing the sensor data based on the policy of the provider, it is necessary to control the distribution of sensor data in a flexible manner according to changes in the policy of the provider; however, since the quality of service to be provided changes by changing the distribution of data, it is necessary to control it while taking into consideration the balance with the user's request. However, with the approach for service construction in the existing research, since we use fixed and exclusively acquired sensor data, either the data provider is not considered for the control of the service, or it is necessary for the provider to manually reflect the policy themselves. For this reason, it is difficult for the existing technology to dynamically control the service according to changes in the provider's policy.

III. CONTRACT-ORIENTED SENSOR-BASED APPLICATION PLATFORM

A. OUTLINE OF THE PROPOSED PLATFORM

Figure 2 shows an outline of the Contract-Oriented Sensor-based Application Platform (COSAP). COSAP adopts a server-less architecture and operates on a user's sensor device that is connected by an arbitrary sensor network. Users can directly communicate each other using the functions of the transport layer, and the overlay layer can reach the target device through geographical range search, etc., by using the functions of the structured P2P network proposed in our previous publications [25], [26]. On the agent network infrastructure in the application layer, contractual relationships are

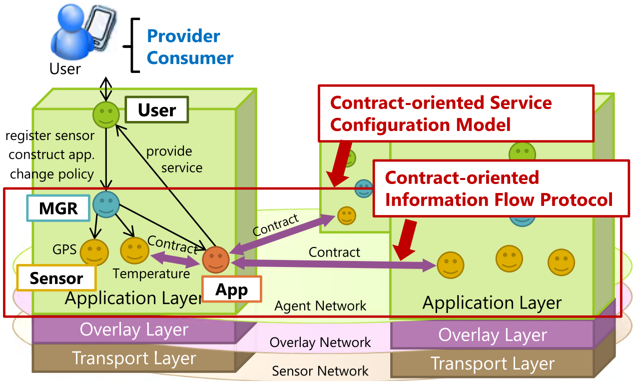


FIGURE 2. **Outline of the Contract-Oriented Sensor-based Application Platform (COSAP).**

established among users based on the contract-oriented service configuration model, thereby realizing the distribution of data and dynamically constructing sensor-based applications. As the provider of sensor data, the users register the sensor devices that they own and change the policy for providing sensor data generated from the devices. Furthermore, as a consumer of the application, they dynamically construct an application that collects sensor data from registered devices.

The following are the four types of configuration components on the COSAP;

User: Converts the request from the user into a format that other components can understand.

MGR: Manages the `Sensor` and `App` components dynamically generated according to requests from users, and manages the contracts they conclude.

Sensor: Functions as a sensor device registered by the user. Based on the policy for provision that has been set, contract(s) with other user's components are bound, and sensor data is provided based on the condition of the contract.

App: Functions as an application built by the user. Based on the set consumer's request, this concludes the contract with other user's components and acquires, processes, and presents sensor data distributed based on the condition of the contract.

B. CONTRACT-ORIENTED SERVICE CONFIGURATION MODEL

In this platform, it needs a flexible and highly coordinated model among components such as sensor devices, sensor networks, and applications. Therefore, we define a contract-oriented service configuration model that realizes applications by capturing and organizing the interaction between components as a relationship based on the contractual concept and controls the distribution of sensor data. The application in this model consists of three components: `Sender`, which is the provider of the data; `Receiver`, which is the consumer of the data; and `Manager`, which monitors both. There is only one Receiver per application, but there can be multiple Senders and Managers. Distribution of data is performed from the Sender to the Receiver, and the frequency and usage are determined based on the agreement concluded beforehand among the constituent components.

We propose the following protocol as the Contract-oriented Information Flow Protocol (CIFP) for contract management among constituent elements.

1) CONTRACT SIGNING PROTOCOL

A basic protocol for concluding a contract between components that control the distribution of data based on the policy of providing sensor data and the relationship with users. At the time of contract negotiation, it is decided whether or not the protocol is established based on both the Sender and Receiver's policies, as well as the relationship between the two. Application of this protocol realizes flexible sensor data distribution according to the provision policy and solves the problem (P1).

2) CONTRACT MANAGEMENT PROTOCOL

Protocol for making changes to contract conditions according to provision policy, consumer's request, changes in network environment, etc. The type of changes to the contract conditions include modifications to amend the conditions of contracts with the same partner, re-agreement to renew a contract with a different partner, and cancellation that stops the discarding of data distribution by discarding the contract. Application of this protocol realizes dynamic control of the function and quality of service according to changes in the provision policy and consumer's request, and solves the problem (P2).

C. ASSUMED ENVIRONMENT

In this research, the following user environment is assumed.

- As a provider of sensor data, the user registers arbitrary ones owned by the user in good faith
- From the viewpoint of the provider's privacy policy and the allowable power consumption, the registered sensor will have the disclosure range, disclosure limit, transfer permission, and frequency of provision set as the provision policy.
- As a user of the application, the user dynamically builds a sensor-based application that collects sensor data in real time in an arbitrary space and frequency
- In the application, sensor data is distributed as raw data, and distribution of processed data is not performed

As an example of a specific sensor-based application configured in the above environment, the following applications are assumed.

1) WEATHER MONITORING SYSTEM

Dynamically discovers environmental sensors such as temperature, illuminance, humidity, and rainfall around the target area, then collects sensor data and displays it in real time. Compared to conventional weather monitoring systems, it is expected that detailed weather data can be obtained for the point that fits the user's requirements.

2) HEALTH MANAGEMENT SYSTEM

Changes frequency and parameters of sensor data collected from biometric sensors such as heartbeat / blood pressure sensors and others according to the health condition of the observed person and displays them in real time. Compared to the conventional health management system, since it is possible to control the distribution of unnecessary sensor data, it is expected that the power consumption of the biosensor can be suppressed.

3) MULTIMEDIA STREAMING SYSTEM

Shares multimedia data among devices, limited to specific areas such as sports venues and live venues, and displays in real time. Compared to conventional multimedia streaming systems, it is not necessary to install infrastructure, and transmission by D2D communication makes it possible to expect improvement of Quality of Experience (QoE).

4) TOURIST INFORMATION SHARING SYSTEM

Shares information about places, limited to specific areas such as sightseeing spots, and shares it with on-site users. Compared to information sharing systems that use conventional Social Networking Services (SNS), etc., improvement in convenience can be expected because selection of related information is triggered by the user being in the position itself.

IV. CONTRACT-ORIENTED INFORMATION FLOW PROTOCOL

A. MODELING COMPONENT ELEMENTS

For the user environment mentioned in Section III-C, we model each component to express the requirements of both providers and consumers.

TABLE 1. Definition of user's policy.

Policy	Policy
`Scope`	Permission of the form of data distribution route. `PUBLIC`, `PROTECTED`, and `PRIVATE` can be selected.
`Frequency`	Frequency of transmitting data. `HIGH`, `MIDDLE`, and `LOW` can be selected.
`Area`	Geographical area where data can flow. The area is circular expressed by `Center` and `Range`.

1) POLICY

Attributes of the policy (`Policy`) in this service composition model is defined in TABLE 1. However, each element of `Scope` and `Frequency` shall have the following relation.

$$\texttt{Scope.PUBLIC} \supset \texttt{Scope.PROTECTED} \supset \texttt{Scope.PRIVATE}$$

$$\texttt{Frequency.HIGH} \supset \texttt{Frequency.MIDDLE} \supset \texttt{Frequency.LOW}$$

2) USER

The attributes of the data provider (`Sender`) and consumer (`Receiver`) in this service composition model are defined in TABLE 2, and the meaning of each element in both policies is defined in TABLE 3.

TABLE 2. Definition of sender and receiver.

Sender/Receiver	Provider/Consumer
`ID`	Identifier of users.
`DataType`	(`Sender`) Data type that the Sender can provide. (`Receiver`) Data type that the Receiver wants to obtain.
`Location`	The user's location.
`Policy`	(`Sender`) Provider's policy. (`Receiver`) Consumer's request.
`WhiteList`	(`Sender` only) Receivers the Sender allows to provide its data.

TABLE 3. Definition of sender's policy and receiver's policy. (a) Sender's Policy. (b) Receiver's Policy.

Sender.Policy	Provider's policy
`Scope.PUBLIC`	allows free provision.
`Scope.PROTECTED`	allows free provision after authorisation.
`Scope.PRIVATE`	denies any provision.
`Frequency`	Maximum frequency of data provision.
`Area`	Authorized area where the Sender's data can be received.

(a)

Receiver.Policy	Consumer's request
`Scope.PUBLIC`	utilizes and transfers received data freely.
`Scope.PROTECTED`	utilizes received data freely and transfers them in succession to the sender's policy.
`Scope.PRIVATE`	utilizes received data freely but does not transfer.
`Frequency`	Minimum frequency of data receiving.
`Area`	Geographical are where objective data generated.

(b)

3) CONTRACT

The form of the contract (`Contract`) in this service composition model is defined below.

Direct Contract: a contract that a provider (`Sender`) and a consumer (`Receiver`) directly negotiated and are connected by.

Multi-Stage Contract: a contract that a Receiver, who has a direct contract, negotiates with another Receiver as a relay of data acquired in the contract. Let Sender, which is the source of sensor data, be the `SourceSender` of its multi-stage contract, and the contract that is closer to SourceSender shall be the superior contract. In addition, when one Contract is used as a standard, a contract higher than that shall be declared as a `RelayContract`.

The attributes of Contract are defined in TABLE 4, and the elements in the conditions of the contract are defined in TABLE 5. In case of a direct contract, $\texttt{Contract.SourceSender} = \texttt{Contract.Sender}$ shall be established. In addition, contracts that satisfy the

TABLE 4. Definition of contract.

Contract	: Contract
`ID`	: Identifier of contracts.
`DataType`	: Data type that shared during the contract.
`Receiver`	: Consumer id.
`Sender`	: Provider id.
`SourceSender`	: Owner id of the shared data.
`Condition`	: Condition of the contract.

TABLE 5. Definition of policy condition.

Contract.Condition	: Condition of the contract
`Scope.PUBLIC`	: allows multi-stage contract.
`Scope.PROTECTED`	: allows multi-stage contract after authorisation.
`Scope.PRIVATE`	: allows just direct contract.
`Frequency`	: Frequency of transmitting data.
`Area`	: Geographical area where receivers are.

following conditions are defined as duplicate contracts. The existence of duplicate contracts means that the same sensor data is distributed through multiple routes.

$$\texttt{contractA.Receiver} = \texttt{contractB.Receiver}$$

$$\texttt{contractA.SourceSender} \\ = \texttt{contractB.SourceSender}$$

TABLE 6. Definition of sensor data.

Value	: Sensor Data
`DataType`	: Type of the data.
`Location`	: Place where the data generated.
`Timestamp`	: Generated time.
`SourceSender`	: Owner id of the data.
`data`	: Value of the data.

4) SENSOR DATA

Attributes of sensor data (`Value`) in this service configuration model are defined in TABLE 6.

TABLE 7. Definition of message.

Message	: Message
`Tag`	: Type of the message.
`Src`	: Sender id of the message.
`Location`	: Location of the sender.
`Payload`	: payload of the message.

5) MESSAGE

The attributes of the message (`Message`) exchanged between the component elements in this service composition model are defined in TABLE 7.

B. DESIGN

Design the Contract-oriented Information Flow Protocol exchanged between the component elements to manage the contract designed in Section IV-A. In this paper, we assume all users are polite and we do not consider mechanisms for detecting malicious users in order to simplify the sequence of our protocols.

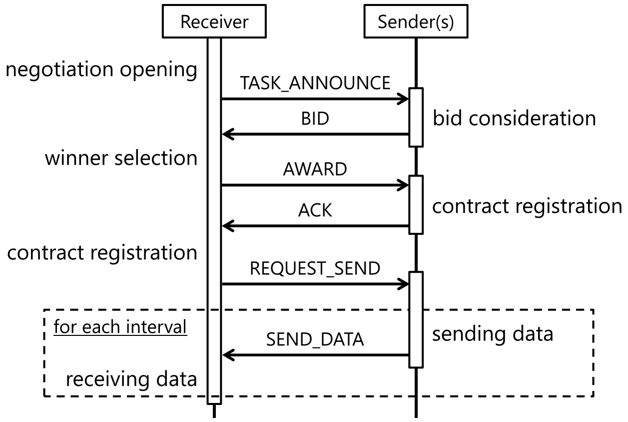


FIGURE 3. Sequence chart of the contract signing protocol.

1) CONTRACT SIGNING PROTOCOL

The Contract signing protocol defines the flow of concluding contracts between `Sender` and `Receiver`. Figure. 3 shows a sequence chart of the contract signing protocol.

Contract signing protocol presents `Receiver.Policy` to surrounding Sender(s) and asks whether to conclude a contract (`TASK_ANNOUNCE`). The Sender(s) to be presented are only the Sender in question if known, otherwise all Senders in `Receiver.Policy.Area` are presented. At this time, the geographic range search is realized by the function of the overlay layer. Sender(s) makes the following comparisons between the `Receiver.Policy` included in the received message and its own `Sender.Policy` and judges whether or not it is possible to conclude a contract.

Determine whether the Receiver meets the requirements of the Sender:

$$\texttt{Receiver.Policy.Scope} \subseteq \texttt{Sender.Policy.Scope}$$

$$\texttt{Receiver.Policy.Frequency} \\ \subseteq \texttt{Sender.Policy.Frequency}$$

$$\texttt{Receiver.Location} \in \texttt{Sender.Policy.Area}$$

$$\texttt{Sender.WhiteList} \neq \emptyset \wedge \texttt{Receiver} \\ \in \texttt{Sender.WhiteList}$$

When concluding direct contract, determine whether Sender meets the requirements of the Receiver:

$$\texttt{Sender.DataType} = \texttt{Receiver.DataType}$$

$$\texttt{Sender.Location} \in \texttt{Receiver.Policy.Area}$$

When concluding a multi-stage contract, determine if multi-stage contract is possible:

$$\texttt{RelayContract.Condition.Scope} \neq \texttt{Scope.PRIVATE}$$

$$\texttt{Sender.Policy.Area} \\ \subseteq \texttt{RelayContract.Condition.Area}$$

When concluding a multi-stage contract, execute the above judgment for the RelayContract as well:

$$\begin{aligned}
&\texttt{Receiver.Policy.Scope} \\
&\qquad \subseteq \texttt{RelayContract.Condition.Scope} \\
&\texttt{Receiver.Policy.Frequency} \\
&\qquad \subseteq \texttt{RelayContract.Condition.Frequency} \\
&\texttt{Receiver.Location} \in \texttt{RelayContract.Condition.Area} \\
&\texttt{RelayContract.DataType} = \texttt{Receiver.DataType} \\
&\texttt{RelayContract.SourceSender.Location} \\
&\qquad \in \texttt{Receiver.Policy.Area}
\end{aligned}$$

When all of the above conditions are satisfied, the Sender judges that it is possible to conclude a contract and creates a contract to bid to the Receiver according to the following:

$$\begin{aligned}
&\texttt{Contract.ID} \leftarrow \texttt{generateId()} \\
&\texttt{Contract.DataType} \leftarrow \texttt{Sender.DataType} \\
&\texttt{Contract.Receiver} \leftarrow \texttt{Receiver} \\
&\texttt{Contract.Sender} \leftarrow \texttt{Sender} \\
&\texttt{Contract.Condition.Scope} \leftarrow \texttt{Sender.Policy.Scope} \\
&\texttt{Contract.Condition.Frequency} \\
&\qquad\qquad \leftarrow \texttt{Receiver.Policy.Frequency} \\
&\texttt{Contract.Condition.Area} \leftarrow \texttt{Sender.Policy.Area}
\end{aligned}$$

if the bid is a direct contract:

$$\texttt{Contract.SourceSender} \leftarrow \texttt{Sender}$$

if the bid is a multi-stage contract:

$$\begin{aligned}
&\texttt{Contract.SourceSender} \\
&\qquad \leftarrow \texttt{RelayContract.SourceSender}
\end{aligned}$$

As a result, the following relation holds for arbitrary contracts, as well as for superior `contractA` and inferior `contractB` in arbitrary multi-stage contracts.

$$\begin{aligned}
&\texttt{contract.SourceSender.Policy} \\
&\qquad \supseteq \texttt{contract.Sender.Policy} \\
&\qquad \supseteq \texttt{contract.Receiver.Policy} \\
&\texttt{contractA.Condition} \supseteq \texttt{contractB.Condition}
\end{aligned}$$

With respect to the contract for bidding, in situations where `Contract.Condition.Scope = Scope.PROTECTED`, notify SourceSender and receive authentication. SourceSender authenticates the user based on `SourceSender.Policy` and `SourceSender.WhiteList` and reports the result as shared group information. At this time, a Contract which cannot receive authentication from SourceSender is excluded from the bid candidates.

Also, if there are multiple bid candidates when authentication from the SourceSender is received, duplicate contracts are detected and contracts other than those with the best distribution conditions are removed from the bid candidates. Thereafter, the remaining bid candidate is notified to the Receiver as a `BID`.

The Receiver that received the BID aggregates the BIDs collected from multiple Senders and likewise excludes the contracts other than the contract having the best distribution condition from the contract candidates when duplicate contracts are detected. A contract relationship is concluded with respect to the contract candidate that remains until the end, and the Sender is notified accordingly (`AWARD`). After that, registration of contract conditions is done for the Manager, and a request for data transmission is reported (`REQUEST_SEND`). Upon receiving the request, the Sender starts providing sensor data to the Receiver at a cycle based on the registered contract conditions (`SEND_DATA`).

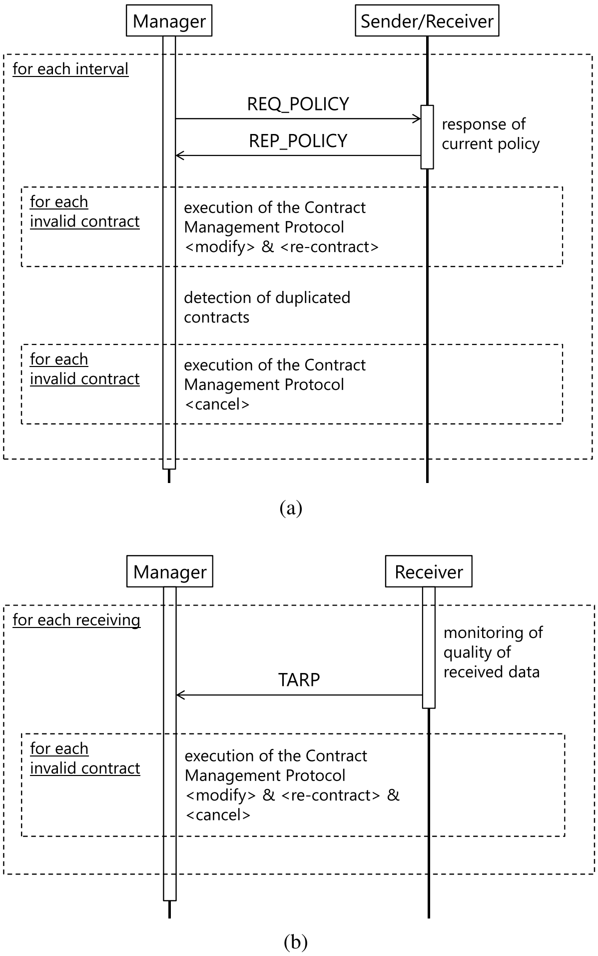


FIGURE 4. Sequence chart of monitoring before the contract management protocol. (a) Sequence chart of `Manager`'s monitoring. (b) Sequence chart of `Receiver`'s monitoring.

2) CONTRACT MANAGEMENT PROTOCOL

The Contract Management Protocol defines the flow of detecting the necessity for amending the condition of contracts that have been concluded, and making the actual changes. Figure. 4 shows a sequence chart at the time of monitoring in Contract Management Protocol.

In the monitoring of contract performance by the Manager as shown in Fig. 4(a), the Manager monitors the performance condition of the contract at regular intervals. First, the Manager inquires the Sender (or Receiver) that has concluded the Contract registered to itself about the current policy (`REQ_POLICY`). The Manager compares the policy based on the response (`REP_POILICY`) and the registered `Contract.Condition` in the following manner, and judges the necessity for amending the contract condition.

If Receiver to be monitored, detect violation of contractual terms:

$$\begin{gathered}\texttt{Receiver.Policy.Scope} \supset \texttt{Contract.Condition.Scope} \\ \Rightarrow \mathit{modify} \leftarrow \text{TRUE}\end{gathered}$$

$$\begin{gathered}\texttt{Receiver.Policy.Frequency} \\ \neq \texttt{Contract.Condition.Frequency} \\ \Rightarrow \mathit{modify} \leftarrow \text{TRUE}\end{gathered}$$

$$\begin{gathered}\texttt{Receiver.Location} \notin \texttt{Contract.Condition.Area} \\ \Rightarrow \mathit{modify} \leftarrow \text{TRUE}\end{gathered}$$

If Sender to be monitored, detect violation of contractual terms:

$$\begin{gathered}\texttt{Sender.Policy.Scope} \neq \texttt{Contract.Condition.Scope} \\ \Rightarrow \mathit{modify} \leftarrow \text{TRUE}\end{gathered}$$

$$\begin{gathered}\texttt{Sender.Policy.Frequency} \\ \subset \texttt{Contract.Condition.Frequency} \\ \Rightarrow \mathit{modify} \leftarrow \text{TRUE}\end{gathered}$$

$$\begin{gathered}\texttt{Sender.Policy.Area} \neq \texttt{Contract.Condition.Area} \\ \Rightarrow \mathit{modify} \leftarrow \text{TRUE}\end{gathered}$$

$$\begin{gathered}\texttt{Sender.WhiteList} \neq \emptyset \\ \wedge \texttt{Contract.Receiver} \notin \texttt{Sender.WhiteList} \\ \Rightarrow \mathit{cancel} \leftarrow \text{TRUE}\end{gathered}$$

When Sender is monitored and a multi-stage contract is being concluded, detect violation associated with RelayContract:

$$\begin{gathered}\texttt{Contract.SourceSender} \neq \texttt{Contract.Sender} \\ \wedge \texttt{RelayContract} = \text{NULL} \\ \Rightarrow \mathit{cancel} \leftarrow \text{TRUE}\end{gathered}$$

$$\begin{gathered}\texttt{RelayContract.Condition.Scope} = \texttt{Scope.PRIVATE} \\ \Rightarrow \mathit{cancel} \leftarrow \text{TRUE}\end{gathered}$$

$$\begin{gathered}\texttt{Sender.Policy.Area} \\ \supset \texttt{RelayContract.Condition.Area} \\ \Rightarrow \mathit{cancel} \leftarrow \text{TRUE}\end{gathered}$$

On the other hand, with respect to the monitoring of the contract performance by the Receiver as shown in Fig. 4(b), each time sensor data based on the contract is received, the following comparison is made to judge the necessity for amending the conditions of the contract.

$$\begin{gathered}\texttt{Value.DataType} \neq \texttt{Contract.DataType} \\ \Rightarrow \mathit{modify} \leftarrow \text{TRUE}\end{gathered}$$

$$\begin{gathered}\texttt{Value.Location} \notin \texttt{Receiver.Policy.Area} \\ \Rightarrow \mathit{modify} \leftarrow \text{TRUE}\end{gathered}$$

$$\begin{gathered}\texttt{Value.TimestamplastValue.Timestamp} \\ > \texttt{Contract.Condition.Frequency} * 2 \\ \Rightarrow \mathit{modify} \leftarrow \text{TRUE}\end{gathered}$$

$$\begin{gathered}\texttt{Value.SourceSender} \neq \texttt{Contract.SourceSender} \\ \Rightarrow \mathit{modify} \leftarrow \text{TRUE}\end{gathered}$$

$$\begin{gathered}\texttt{recvTime} - \texttt{lastRecvTime} \\ > \texttt{Contract.Condition.Frequency} * 2 \\ \Rightarrow \mathit{modify} \leftarrow \text{TRUE}\end{gathered}$$

$$\begin{gathered}\texttt{Message.src} \neq \texttt{Contract.Sender} \\ \Rightarrow \mathit{modify} \leftarrow \text{TRUE}\end{gathered}$$

$$\begin{gathered}\texttt{Message.Location} \notin \texttt{Receiver.Policy.Area} \\ \Rightarrow \mathit{modify} \leftarrow \text{TRUE}\end{gathered}$$

Note that `recvTime` is the time when the message was received, `lastRecvTime` is the message time immediately before is received, and `lastValue` is the sensor data acquired immediately before.

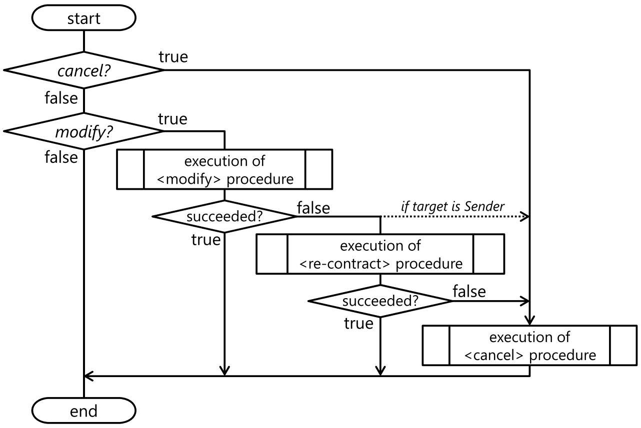


FIGURE 5. **Flow chart of decision making.**

After determining the necessity of changing the contract conditions, the Manager decides the procedure to amend the contract conditions according to FIGURE 5. Figure. 6 shows a sequence chart of modification, re-contract, and cancellation. When a multi-stage contract is canceled as in Fig. 6(c), the Receiver requests a direct contract with SourceSender through the Contract signing protocol. By doing this, even if all multi-stage contracts are canceled due to policy change, etc., of data relayers, it becomes possible to take over the contract with SourceSender again.

After completing the amendments to the condition of a contract, the Manager detects duplicate contracts against the updated contract and eliminates other contracts, leaving only the contract with the best conditionïijŐ

V. EXPERIMENTS

A. EXPERIMENT 1: EXPERIMENT TO CONFIRM OPERATION

1) OVERVIEW

In Experiment 1, to confirm the effectiveness of COSAP and CIFP, data is distributed among users with various policies, and we confirm to see whether they are correctly controlled by fluctuations in the environment (addition / deletion of users, policy change). Figure. 7 shows the assumed environment. In the assumed environment, there are nine users (Peer 1 to Peer 9), and they reside in each of the three areas, Area A, Area B, and Area C. Each Peer has sensors that collect two kinds of sensor data in its own device (Sensor 101, Sensor 102, $\cdots$, Sensor 901, Sensor 902), and each user sets

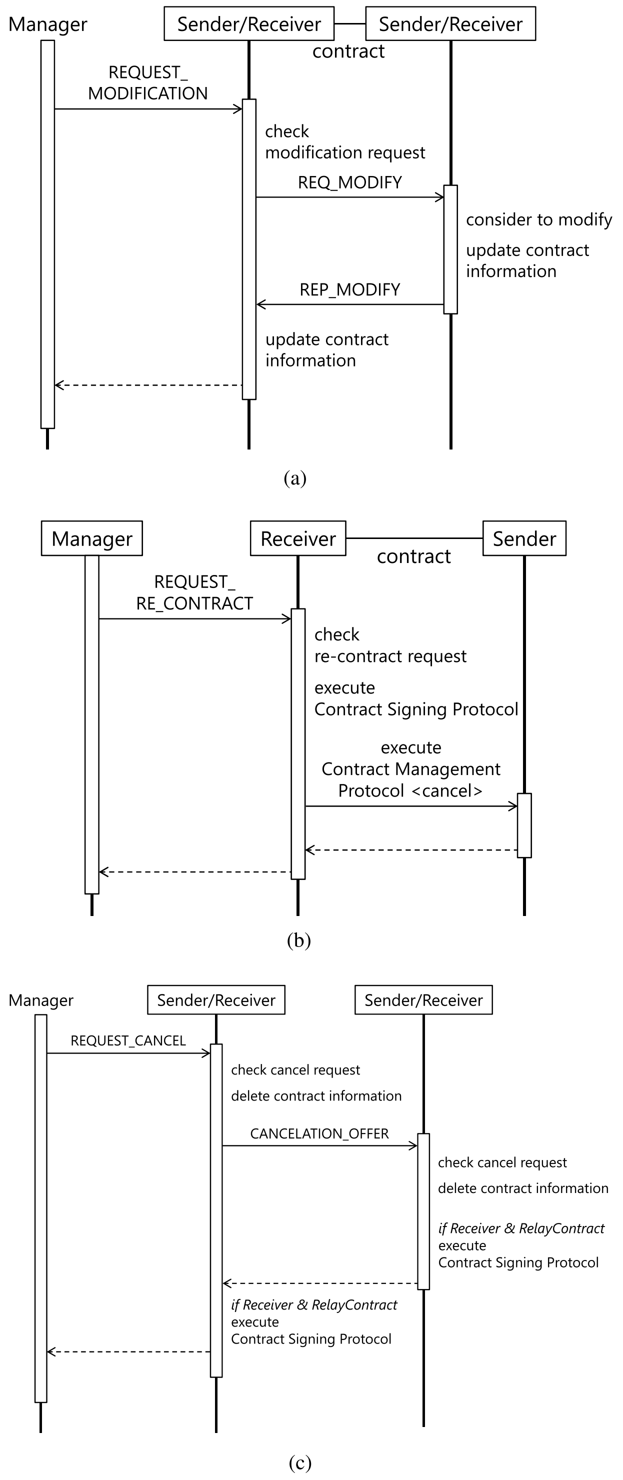


FIGURE 6. **Sequence chart of the contract management protocol. (a) ⟨modify⟩. (b) ⟨re-contract⟩. (c) ⟨cancel⟩.**

their own policy. At this time, some users (Peer 1 to Peer 3) limit the provision range to the area where they are in to suppress the power consumption of their own sensors. Peer 7 also limits the provision of sensor data to users who are acquaintances (Peer 1, Peer 2, Peer 3, Peer 6) for privacy protection.

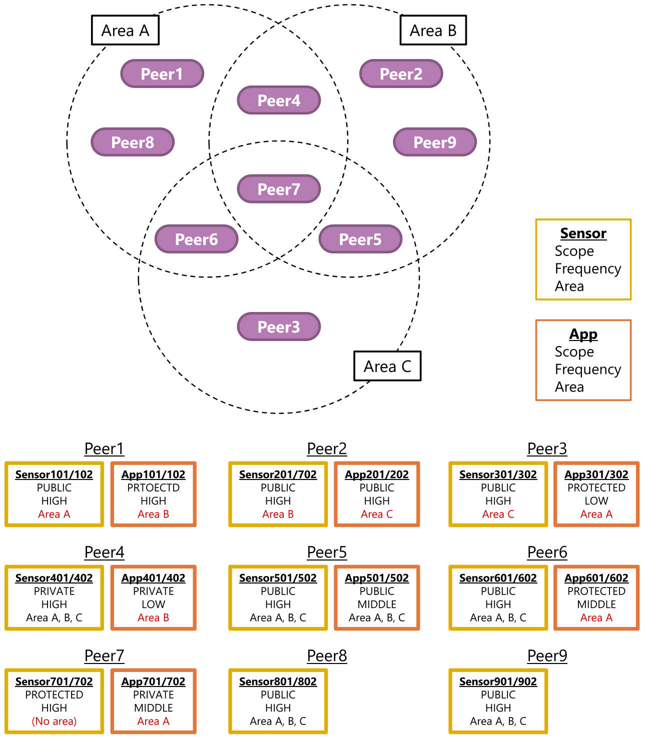


FIGURE 7. **Assumed environment of experiment 1.**

The scenario employed was as follows.

1) Peer 1 to Peer 7 register two sensors each, for 14 sensors in total (Sensor 101 to Sensor 702)
2) Peer 1 to Peer 7 construct the applications (App 101 to App 701)
3) Each application is used for 30 seconds (Environment 1)
4) The following changes are executed for the environment
    - Scope of Sensor 301 is changed to `PRIVATE`
    - Scope of Sensor 401 is changed to `PUBLIC`
    - Frequency of provision of Sensor 501 is reduced to `MIDDLE`
    - Frequency of provision of Sensor 601 is reduced to `MIDDLE`
    - Peer 7 releases the registration of two sensors (Sensor 701, Sensor 702)
    - Peer 8 newly registers two sensors to Area A (Sensor 801, Sensor 802)
    - Peer 9 newly registers two sensors to Area B (Sensor 901, 9ensor 902)
5) Each application is used for 30 seconds (Environment 2)

The parameters set for the experiment were as follows.

$$\begin{aligned}\texttt{Frequency.HIGH} &= 1000(\text{ms})\\ \texttt{Frequency.MIDDLE} &= 2000(\text{ms})\\ \texttt{Frequency.LOW} &= 4000(\text{ms})\\ \alpha &= 0.5\end{aligned}$$

Here, the parameter α is a ratio for selecting a multi-stage contract where direct contracts and multistage contracts with the same distribution conditions exist at the time of selecting a contract with the best distribution condition from the collected BID in the basic contract protocol. In other words, if $\alpha = 0.0$, all the contracts are direct contracts, and if $\alpha = 1.0$, a multi-stage contract is bound where possible.

In this experiment, evaluation is carried out based on contracts concluded in each environment. Experiments 2 and 3 respectively analyze the distribution of messages and elapsed time at the time of concluding a contract and at the time of amending the contract conditions. Also, for the names of Sensor and App, the first digit of the numerical value of each component shall represent DataType.

TABLE 8. **Result of Experiment 1 (Environment 1).**

Contract chain	Contract chain
Sensor101-App601	Sensor102-App602
Sensor101-App701	Sensor102-App702
Sensor201-App401	Sensor202-App402
Sensor201-App501	Sensor202-App502
Sensor301-App501	Sensor302-App502
Sensor401-App401	Sensor402-App402
Sensor401-App701	Sensor402-App702
Sensor501-App201-App101	Sensor502-App202-App102
Sensor501-App201-App401	Sensor502-App202-App402
Sensor501-App501	Sensor502-App502
Sensor601-App201	Sensor602-App202-App502
Sensor601-App301	Sensor602-App302
Sensor601-App501	Sensor602-App602
Sensor601-App601	Sensor602-App702
Sensor601-App701	Sensor702-App102-App302
Sensor701-App101-App301	Sensor702-App102-App602
Sensor701-App101-App601	

2) EXPERIMENTAL RESULTS

TABLE 8 and TABLE 9 show the contracts concluded in each environment. In the table, components with contract relationships are connected with a hyphen; the left side is Sender, and the right side is Receiver.

From the tables, 38 contracts were concluded in Environment 1 and 46 contracts in Environment 2. Also, each contract was concluded between the components with the same last digit (DataType are equal).

3) DISCUSSION

Based on the result in Environment 1, Sensor 101 exists in Area A (`Sensor101.Policy.Area = Area A`), and a contract was correctly concluded between App 601 and App 701 (`App601.Policy.Area = Area A`, `App701.Policy.Area = Area A`), which need the data from Area A. Furthermore, in the multi-stage agreement, in the contract between the Sensor 601 and the App 201, data is distributed every 1 second, which is the required frequency of App 201 (`App201.Policy.Frequency = HIGH`), and for the inferior contracts App 201 and App 501, data was distributed every 2 seconds, which is the required frequency of App 501 (`App501.Policy.Frequency = MIDDLE`). Similarly in all other contracts, we confirmed that matching was performed according to both policies, and contractual relationship was concluded, and distribution of data based on policy was started. Based on the above, it was possible to realize the distribution of sensor data according to the policy of the provider while satisfying the consumer's request, by applying the contract signing protocol, which led to resolution of "(P1) Setting of sensor data distribution in accordance with provider's policy is difficult".

TABLE 9. **Result of Experiment 1 (Environment 2).**

Contract chain	Contract chain
Sensor101-App601	Sensor102-App602
Sensor101-App701	Sensor102-App702
Sensor201-App401	Sensor202-App402
Sensor201-App501	Sensor202-App502
Sensor401-App101	Sensor302-App502
Sensor401-App301	Sensor402-App402
Sensor401-App401	Sensor402-App702
Sensor401-App501	Sensor502-App202-App102
Sensor401-App601	Sensor502-App202-App402
Sensor401-App701	Sensor502-App502
Sensor501-App401	Sensor602-App202-App502
Sensor501-App501	Sensor602-App302
Sensor601-App301	Sensor602-App602
Sensor601-App501	Sensor602-App702
Sensor601-App601	Sensor802-App302
Sensor601-App701	Sensor802-App502
Sensor801-App301	Sensor802-App602
Sensor801-App501	Sensor802-App702
Sensor801-App601	Sensor902-App102
Sensor801-App701	Sensor902-App402
Sensor901-App101	Sensor902-App502
Sensor901-App401	
Sensor901-App501	

Next, from the result in Environment 2, Sensor 301, whose Scope was changed to `PRIVATE`, confirmed that the contract with the App 501, whose Scope is `PUBLIC`, was canceled. Also, Sensor 501 and Sensor 602, for which the Frequency was reduced to `MIDDLE`, confirmed that the contract between App 101 and App 201 that had requested `HIGH` Frequency was canceled. Regarding other contracts, contract details were changed correctly according to policy changes. From the above, it was possible to flexibly control the distribution of the sensor data according to changes in the policy and environment of the provider, which led to resolution of "(P2) Dynamic control of the service according to changes in the provider's policy is difficult".

B. EXPERIMENT 2: EXPERIMENT TO EVALUATE THE CONTRACT SIGNING PROTOCOL

1) OVERVIEW

Experiment 2 evaluates the load of each component and the quality of the distribution data due to the difference in the form of the contract. There are six users (Peer 1 to Peer 6), Peer 1 acts as a provider of sensor data, and Peer 2 to Peer 6 act as consumers of sensor data. Peer 1 provides one sensor (Sensor 101) and has provision policies {`PUBLIC`, `HIGH`, `all areas`}. Peer 2 to Peer 6 each build 20 applications and have common user requests {`PUBLIC`, `HIGH`, `all areas`} in all.

The scenarios used for evaluation are as follows.

1) Peer 1 registers one sensor (Sensor 101)
2) Peer 1 to Peer 6 construct 20 applications each (App 201 to App 220, $\cdots$, App 601 to App 620)
3) Each application is used for 30 seconds

As targets of comparison, distribution of data when $\alpha = 0.0, 0.2, 0.4, 0.6, 0.8, 1.0$ was analyzed, and contract signing protocol was evaluated using the following evaluation parameters.

a: FAIRNESS INDEX (FI) [27] OF THE NUMBER OF TRANSMISSIONS

This is a normalized index showing the degree of dispersion of load and is defined by the following equation. In addition, n represents the number of components, and x_i represents the number of data transmissions of component i.

$$FI := \frac{(\sum x_i)^2}{n \sum x_i^2}$$

b: RECEPTION INTERVAL OF RECEIVE DATA (RIRD)

The interval between the times receiving contract data (`SEND_DATA`).

$$\texttt{recvTime} - \texttt{lastRecvTime}$$

c: GENERATION INTERVAL OF RECEIVED DATA (GIRD)

The interval between the times received data being generated.

$$\texttt{Value.Timestamp} - \texttt{lastValue.Timestamp}$$

d: FRESHNESS OF RECEIVED DATA (FRD)

The interval between the time the data was received and the time the data was generated. The larger this value, the more old data that has been distributed, signifying a decrease in freshness.

$$\texttt{recvTime} - \texttt{Value.Timestamp}$$

In this experiment, Contract Management Protocol was set to OFF, and the following parameters were set.

$$\begin{aligned} \texttt{Frequency.HIGH} &= 1000(\text{ms}) \\ \texttt{Frequency.MIDDLE} &= 2000(\text{ms}) \\ \texttt{Frequency.LOW} &= 4000(\text{ms}) \\ \alpha &= 0.5 \end{aligned}$$

2) EXPERIMENTAL RESULTS

The experimental results are shown in TABLE 10. From the table, FI increased as the multi-stage contract was given priority, that is, as α increased. Meanwhile, at the same time, decrease in freshness (average), generation interval (standard deviation), and increase in freshness (standard deviation) were confirmed.

TABLE 10. **Result of Experiment 2.**

α	FI	RIRD		GIRD		FRD	
		(avg)	(std)	(avg)	(std)	(avg)	(std)
1.0	0.321	1000.012	4.968	1000.223	29.695	1637.236	23.688
0.8	0.160	999.982	4.776	1000.680	43.470	1386.210	27.277
0.6	0.080	1000.013	4.561	1001.572	42.219	687.266	27.431
0.4	0.031	999.938	5.405	999.947	5.189	247.690	2.359
0.2	0.016	999.975	5.222	1000.012	4.973	0.062	0.267
0.0	0.010	1000.039	4.559	1000.051	4.432	0.040	0.161

3) DISCUSSION

Because FI increased with the prioritization of a multi-stage contract, the cost of communication necessary for data distribution could be distributed among the components by introducing a multi-stage contract. For this reason, we confirmed that the concept of a multi-stage contract is effective not only for sensor data but also for extending usage of public data, such as distributing image data, which is heavy, between components.

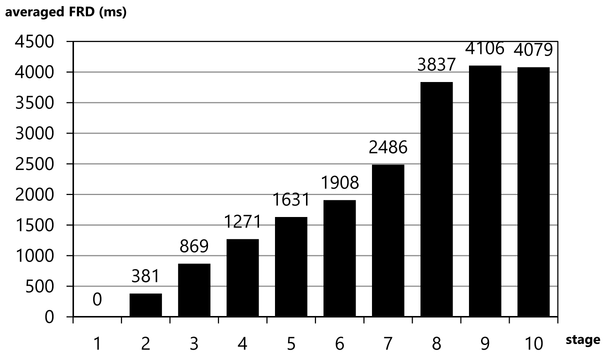


FIGURE 8. **Average of FRD for each contract stage ($\alpha = 1.0$).**

On the other hand, freshness decreased as the multi-stage contract was given priority. This is due to the fact that old data is being transferred as the number of multi-stage contracts increases. Figure. 8 shows the average of freshness for each contract stage at $\alpha = 1.0$. From the figure, the freshness of the data obtained by the first-stage Receiver who receives data directly from the Sender is 0, but the freshness value also increases as the number of stages increases. As a cause of this, one can raise the fact that since the component, which is a relay node of data, is not synchronized with the timing of receiving data with the preceding contract and the timing of sending data with the contract of the subsequent stage, accumulating a delay equivalent to the same amount of deviation between the two steps. Since the accumulated delay depends on the timing of concluding a contract, there are differences depending on the components. However, the maximum value of delay accumulated per stage is approximately 1,000 ms, which is the data transmission frequency.

In addition, the standard deviation of the generation interval and the standard deviation of the frequency increased as the multi-stage contract was prioritized. This is due to the fact that the data relayer mistakenly transmits the same data

TABLE 11. Number of erroneous transmissions for each contract stage.

Stage	1	2	3	4	5	6	7	8	9	10
Error	0	3	16	3	0	0	0	0	0	-

as the data previously transmitted when transferring the data. TABLE 11 shows the number of erroneous transmissions of the same data for each contract stage at $\alpha = 1.0$. From the table, it can be seen that erroneous data transmission was performed regardless of the number of contracted stages. As a cause of this, as described above, since the timing of the transmission and reception of the data relaying component is not synchronized, in the case where the timing is very close, before new data is received in the preceding contract, new data may be transmitted by the subsequent contract.

Based on the above results, a multi-stage agreement based on the Contract Signing Protocol is useful from the viewpoint of load balancing, but depending on the timing of data transmission, the transfer data may be become out-of-date, or the erroneous transmission of the same data may become more likely. As a solution to these problems, since the cause of both issues is the transmission timing of the multi-stage agreement, when data transmission is started by concluding a multi-stage agreement (at reception of `SEND_DATA` in FIGURE 3), it is conceivable that it is effective to start the distribution of sensor data according to the reception timing in the RelayContract, which is the superior contract.

C. EXPERIMENT 3: EVALUATION STUDY OF CONTRACT MANAGEMENT PROTOCOL

1) OVERVIEW

Here, we evaluate the time it takes to complete the amendment of contract conditions and the number of contracts to be concluded due to the difference in the means of amending the contract conditions. There are six users (Peer 1 to Peer 6), whereby Peer 1 acts as a provider of sensor data, and Peer 2 to Peer 6 act as consumers of sensor data. There are two sensors provided by Peer 1 (Sensor 101, Sensor 102); Sensor 101 has a provision policy {`PUBLIC`, `MIDDLE`, `all areas`}, and the Sensor 102 has a provision policy {`PUBLIC`, `LOW`, `all areas`}. Peer 2 to Peer 6 each build 10 applications and have common user requests {`PUBLIC`, `MIDDLE`, `all areas`} in all.

The following scenarios were used for evaluation.

1) Peer 1 registers 2 sensors (Sensor 101, Sensor 102)
2) Peer 2 to Peer 6 each constructs 10 applications (App 201 to App 210, $\cdots$, App 601 to App 610)
3) Each application is used for 30 seconds (Environment 1)
4) The following are implemented
   - Policy frequency of Sensor 102 is changed from `LOW` to `HIGH`
   - Policy frequency of App 201 to App 310 is changed from `MIDDLE` to `LOW`
   - Policy frequency of App 401 to App 610 is changed from `MIDDLE` to `HIGH`
5) Each application is used for 30 seconds (Environment 2)

TABLE 12. Result of Experiment 3 (Case A).

NCC	TDPC (ms)	TECA (ms)	NPV
50→50	1079.788	52.970	9
50→50	1246.719	164.344	14
50→50	1084.324	91.559	9
50→50	1053.000	84.606	11
50→50	1201.433	109.833	11

The following case is assumed as a target for comparison.

Case A: Execute ⟨modify⟩, ⟨re-contract⟩, and ⟨cancel⟩ as a means of amending contract conditions.

Case B: Execute ⟨modify⟩ and ⟨cancel⟩ as a means of amending contract conditions.

Case C: Execute ⟨cancel⟩ as a means of amending contract conditions.

After executing the scenario for each of the above cases, the Contract Management Protocol was evaluated according to the following parameters.

Number of Contracts Concluded (NCC): Number of contracts concluded after amending contract conditions.

Time taken to Detect Policy Change (TDPC); The time from amendment of policy to the start of amendment to contract conditions. Measure the time from when the component receives the `SET_POLICY` message until time when `REQUEST_MODIFICATION`, `REQUEST_RE_CONTRACT`, or `REQUEST_CANCEL` message is received for the first time.

Time taken to Execute Contract Amendment (TECA): the time from the start of amendment of contract conditions to the end the amendment process. Measure the time from when a component first receives a `REQUEST_MODIFICATION`, `REQUEST_RE_CONTRACT`, or `REQUEST_CANCEL` message until the completion of those sequences.

Number of Policy Violations (NPV): Number of data distributed based on old policy information.

Parameters set for the experiment are as follows.

$$\begin{aligned}\texttt{Frequency.HIGH} &= 1000(\text{ms})\\ \texttt{Frequency.MIDDLE} &= 2000(\text{ms})\\ \texttt{Frequency.LOW} &= 4000(\text{ms})\\ \alpha &= 0.5\\ \text{interval of MGR's observation} : &= 2000(\text{ms})\end{aligned}$$

2) EXPERIMENTAL RESULTS

TABLE 12, TABLE 13, and TABLE 14 show the experimental results when each case was attempted five times.

Regarding Case A, the number of contracts concluded remained unchanged before and after the environmental change at 50 contracts, and the policy change detection time was approximately 1,000 ms. In addition, the time taken to execute contract amendment was 100 ms, about 1/10 of

TABLE 13. Result of Experiment 3 (Case B).

NCC	TDPC (ms)	TECA (ms)	NPV
50→29	1952.658	11.395	18
50→23	867.750	9.063	11
50→27	870.563	14.906	3
50→26	390.923	20.731	5
50→27	953.281	13.313	8

TABLE 14. Result of Experiment 3 (Case C).

NCC	TDPC (ms)	TECA (ms)	NPV
50→05	856.886	7.429	5
50→02	762.595	16.297	4
50→04	1081.118	9.853	5
50→08	713.375	13.719	7
50→06	1272.108	14.568	8

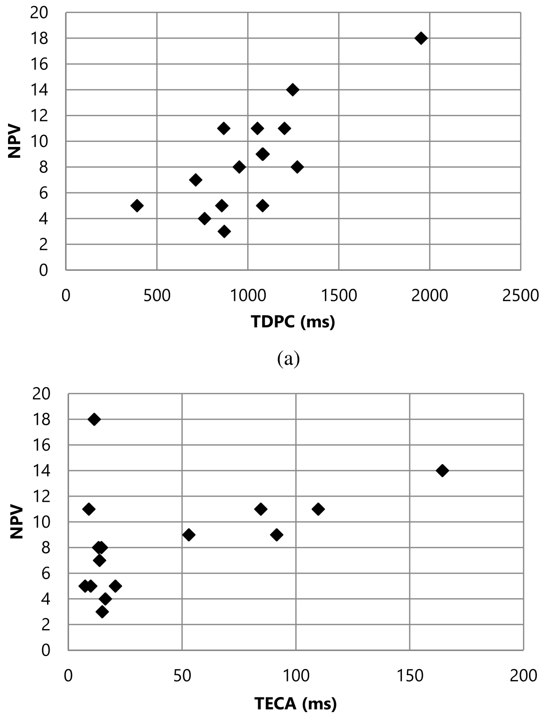
TABLE 15. Result of Experiment 3.

Case	NCC	TDPC (ms)	TECA (ms)	NPV
Case A	50→50	1130.784	100.043	10.8
Case B	50→26	1065.600	13.531	9.0
Case C	50→05	942.063	12.434	5.8

the policy change detection time, and the number of policy violations with the distributed data was about 10. Regarding Case B, the number of contracts concluded was about half after the change of environment, and the policy change detection time was about 1,000 ms on average, although it was somewhat varying. The time taken to execute contract amendment was around 15 ms, and the number of policy violations with the distributed data was about 10. With regard to Case C, the number of contracts concluded decreased sharply after the change of environment to around 5, and the policy change detection time was about 1,000 ms on average, unchanged from the other cases. The time taken to execute contract amendment was around 15 ms, and the number of policy violations with distributed data was around 5.

3) DISCUSSION

TABLE 15 shows the average of each parameter for each case. Case A, which executes ⟨modify⟩, ⟨re-contract⟩, and ⟨cancel⟩ as a means of amending contract conditions had the highest number of contracts concluded, and associated with this, it showed a tendency to increase the time taken to execute contract amendment. The policy change detection time is about 1,000 ms for each case, because it comes from the monitoring cycle (every 2,000 ms) of contract performance by MGR. In addition, it was found that the number of policy violations with distributed data was influenced by the length of policy change detection time, and was not greatly affected by the time taken to execute contract amendment (FIGURE 9). Therefore, in an environment with a scale of 50 units like in this experiment, we can expect that policy violations will almost completely not happen during the amendment of contract conditions by making the value of `Frequency.HIGH` sufficiently larger than the time taken to execute contract amendment, at 0.1 seconds.



(a)

(b)

FIGURE 9. Correlation with the number of policy violations. (a) Correlation between the time taken to detect policy change (TDPC) and the number of policy violation (NPV). (b) Correlation between the time taken to execute contract amendment (TECA) and the number of policy violation (NPV).

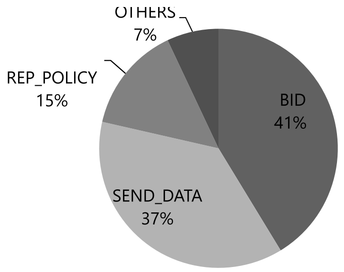


FIGURE 10. Ratio of the transmission message volume of App 201 in Case A.

To eliminate the number of policy violations, it is effective to shorten the monitoring cycle by the MGR and reduce the policy change detection time, but the increase in traffic volume accompanying it is not a negligible amount. Figure. 10 shows the ratio of the transmission message volume of App 201 in Case A. As can be seen from the figure, even in this setting where the monitoring cycle of MGR was about the same as `Frequency.MIDDLE`, the amount of monitoring response message to MGR occupied 15% of the total. Therefore, it can be said that further increase in maintenance packets not directly related to data distribution is not preferable.

From the above results, the Contract Management Protocol, which executes ⟨modify⟩, ⟨re-contract⟩, and ⟨cancel⟩, is effective for contract management, and it turned out that its execution time does not greatly affect the number of policy violations with distribution data. As a solution to reduce the number of policy violations, it would be effective to judge the transmission of the TRAP message to the MGR, which was conventionally judged every time the Receiver received the data, whenever each component amends its own policy, and minimize the time until MGR detects the policy change.

VI. CONCLUSION

In this paper, we aimed at realizing a software platform that dynamically builds applications that share sensor data based on the policy of providers, targeting sensor-based applications based on participatory sensing paradigm. To realize this objective, we have worked on the realization of dynamic sensor data utilization, taking into account the provider's policy, which was difficult with the conventional sensor-based application configuration model. Specifically, we proposed a Contract-Oriented Sensor-based Application Platform based on a service configuration model that does not install servers or clouds. We also designed and implemented the Contract-oriented Information Flow Protocol used for negotiations among users, and demonstrated that it is possible to realize the distribution of sensor data according to the policy of the provider through simulation experiments.

In this research, we automate the data distribution path reflecting the policy and reduce the burden on the data provider to prevent the phenomenon of motivation for data provision. On the other hand, there are many studies that directly give incentives to data providers by applying auction theory and mechanism design theory. Therefore, in the future, it will be necessary to advance the system design such that the provider receives some benefit by negotiating and actively providing their own data.

REFERENCES

[1] Y. Shouyi, L. Leibo, Z. Renyan, S. Zhongfu, and W. Shaojun, "Design of wireless multi-media sensor network for precision agriculture," *Commun., China*, vol. 10, no. 2, pp. 71–88, Feb. 2013.
[2] C. Suh and Y. B. Ko, "Design and implementation of intelligent home control systems based on active sensor networks," *IEEE Trans. Consum. Electron.*, vol. 54, no. 3, pp. 1177–1184, Aug. 2008.
[3] A. Pantelopoulos and N. G. Bourbakis, "A survey on wearable sensor-based systems for health monitoring and prognosis," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 40, no. 1, pp. 1–12, Jan. 2010.
[4] M. Nakamura, S. Matsuo, S. Matsumoto, H. Sakamoto, and H. Igaki, "Application framework for efficient development of sensor as a service for home network system," in *Proc. IEEE Int. Conf. Ser. Comput. (SCC)*, Jul. 2011, pp. 576–583.
[5] T. H. Ma, W. Tian, B. Wang, D. H. Guan, and S. Y. Lee, "Weather data sharing system: An agent-based distributed data management," *IET Softw.*, vol. 5, no. 1, pp. 21–31, Feb. 2011.
[6] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: Current state and future challenges," *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 32–39, Nov. 2011.
[7] M. Conti *et al.*, "From manet to people-centric networking: Milestones and open research challenges," *Comput. Commun.*, vol. 71, pp. 1–21, Nov. 2015.
[8] N. D. Lane, S. B. Eisenman, M. Musolesi, E. Miluzzo, and A. T. Campbell, "Urban sensing systems: opportunistic or participatory?" in *Proc. 9th Workshop Mobile Comput. Syst. Appl.*, 2008, pp. 11–16.
[9] F. Delmastro, V. Arnaboldi, and M. Conti, "People-centric computing and communications in smart cities," *IEEE Commun. Mag.*, vol. 54, no. 7, pp. 122–128, Jul. 2016.
[10] J. A. Burke *et al.*, "Participatory sensing," in *Proc. World Sensor Web Workshop (SenSys)*, Oct. 2006, pp. 1–5.
[11] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, and R. A. Peterson, "People-centric urban sensing," in *Proc. 2nd Annu. Int. Workshop Wireless Internet*, Aug. 2006, pp. 18–31.
[12] M. Conti and M. Kumar, "Opportunities in opportunistic computing," *Computer*, vol. 43, no. 1, pp. 42–50, 2010.
[13] (Feb. 20, 2017). *NoiseTube*. [Online]. Available: http://noisetube.net/
[14] R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu, "Ear-phone: An end-to-end participatory urban noise mapping system," in *Proc. Int. Conf. Inf. Process. Sensor Netw.*, Apr. 2010, pp. 105–116.
[15] P. Dutta *et al.*, "Common sense: Participatory urban sensing using a network of handheld air quality monitors," in *Proc. 7th ACM Conf. Embedded Netw. Sens. Syst. (SenSys)*, 2009, pp. 349–350.
[16] S. Kim, J. Mankoff, and E. Paulos, "Sensr: Evaluating a flexible framework for authoring mobile data-collection tools for citizen science," in *Proc. Conf. Comput. Supported Cooperat. Work*, Feb. 2013, pp. 1453–1462.
[17] J. Zaman and W. D. Meuter, "DisCoPar: Distributed components for participatory campaigning," in *Proc. 6th IEEE Workshop Pervasive Collaboration Soc. Netw.*, May 2015, pp. 160–165.
[18] T. Oide, T. Abe, and T. Suganuma, "A design of contract-oriented sensor application platform," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2015, pp. 172–177.
[19] Data go jp. Japanese Government. (Feb. 20, 2017). *DATA GO JAPAN*. [Online]. Available: http://www.data.go.jp/
[20] (Feb. 20, 2017). *DATA.GOV.UK Opening Up Government*. [Online]. Available: https://data.gov.uk/
[21] Data.gov. U.S. General Services Administration. (Feb. 20, 2017). *DATA.GOV*. [Online]. Available: https://www.data.gov/
[22] B. Predić, Z. Yan, J. Eberle, D. Stojanovic, and K. Aberer, "ExposureSense: Integrating daily activities with air quality using mobile participatory sensing," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun.Workshops (PERCOM Workshops)*, Mar. 2013, pp. 303–305.
[23] *OpenSense*. (Feb. 20, 2017). [Online]. Available: http://opensense.epfl.ch/
[24] V. Arnaboldi, M. Conti, and F. Delmastro, "CAMEO: A novel context-aware middleware for opportunistic mobile social networks," *Pervasive Mobile Comput.*, vol. 11, pp. 148–167, Apr. 2014.
[25] A. Takeda, T. Oide, and A. Takahashi, "Simple dynamic load balancing mechanism for structured P2P network and its evaluation," *Int. J. Grid Utility Comput.*, vol. 3, nos. 2–3, pp. 126–135, 2012.
[26] T. Oide, A. Takeda, A. Takahashi, and T. Suganuma, "A robust P2P information sharing system and its application to communication support in natural disasters," *Int. J. Softw. Sci. Comput. Intell.*, vol. 5, no. 4, pp. 20–39, Oct. 2013.
[27] R. K. Jain, D.-M. W. Chiu, and W. R. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," DEC Res., Hudson, MA, USA, Tech. Rep. DEC-TR-301, Sep. 1984.

TAKUMA OIDE received the M.S. degree in information science from Tohoku University in 2015, where he is currently pursuing the Ph.D. degree. Since 2015, he is a research fellow of the Japan Society for the Promotion of Science. His research interests include P2P network, flexible network middleware, wireless sensor network, and agent-based computing.

TORU ABE received the M.E. and Ph.D. degrees in information engineering from Tohoku University in 1987 and 1990, respectively. From 1990 to 1993, he was a Research Associate with the Education Center for Information Processing, Tohoku University. From 1993 to 2001, he was an Associate Professor with the Graduate School of Information Science, Japan Advanced Institute of Science and Technology. He is currently an Associate Professor with the Cyberscience Center, Tohoku University. His research interests include image processing and knowledge engineering.

TAKUO SUGANUMA (M'01) received the M.S. and Ph.D. degrees in engineering from the Chiba Institute of Technology, Japan, in 1994 and 1997, respectively. He is currently a Professor with the Cyberscience Center, Tohoku University, Japan. His main research interests include agent-based computing, flexible network middleware, network management, symbiotic computing, green ICT, and Disaster-resistant communications. He is a member of the IEICE and the IPSJ. He received the UIC-07 Outstanding Paper Award in 2007.

• • •