Object-based Information Flow Control in Peer-to-peer Publish/Subscribe Systems

著者	NAKAMURA Shigenari
著者別名	中村 繁成
page range	1-84
year	2020-03-24
学位授与番号	32675甲第484号
学位授与年月日	2020-03-24
学位名	博士(工学)
学位授与機関	法政大学 (Hosei University)
URL	http://doi.org/10.15002/00022974

法政大学審査学位論文

Object-based Information Flow Control in Peer-to-peer Publish/Subscribe Systems

Shigenari Nakamura

HOSEI UNIVERSITY

Object-based Information Flow Control in Peer-to-peer Publish/Subscribe Systems

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

Doctor of Engineering
(Systems Engineering and Science)

in Advanced Sciences

by

Shigenari Nakamura

Dissertation committee:

Professor Makoto Takizawa, Committee Chair
Professor Takao Miura
Professor Makoto Kanazawa
Professor Leonard Barolli

2020

Object-based Information Flow Control in Peer-to-peer Publish/Subscribe Systems

This dissertation of Shigenari Nakamura
is approved and is acceptable
in quality and form for publication:

Makoto Takizawa 滝沢
Professor Makoto Takizawa, Committee Chair

Takao Miura 三浦
Professor Takao Miura

Makoto Kanazawa
Professor Makoto Kanazawa

Professor Leonard Barolli

Hosei University
2019

Contents

List of Figures

List of Tables

Acknowledgments

The author would like to thank Professor Makoto Takizawa, Hosei University, the author's supervisor for his constant support and encouragement at all levels during the duration of this project "Object-based Information Flow Control in Peer-to-peer Publish/Subscribe Systems". He extended his support whenever needed, both for my Ph.D research as well as in other associated work that was part of this the project but not of this thesis. Then, the author also would like to appreciate Professor Takao Miura, Hosei University, Professor Makoto Kanazawa, Hosei University, and Professor Leonard Barolli of Fukuoka Institute of Technology for their helpful suggestions and careful review of this thesis.

Curriculum Vitae

Shigenari Nakamura

Education

2015	B.E. in Science and Engineering, Hosei University, Japan
2017	M.E. in Engineering, Hosei University, Japan
2020	Ph.D. in Engineering, Hosei University, Japan

Research Experience

Research Fellow of Japan Society for the Promotion of Science 2017-present

Field of Study

access control model, information flow control, distributed systems.

Abstract

Distributed systems are getting so scalable like IoT (Internet of Things) and P2P (Peer-to-Peer) systems that millions of devices are connected and support various types of applications. Here, distributed systems are required to be secure in addition to increasing the performance, reliability, and availability and reducing the energy consumption. In distributed systems, information in objects flows to other objects by transactions reading and writing data in the objects. Here, some information of an object may illegally flow to a subject which is not allowed to get the information of the object. Especially, a leakage of sensitive information is to be prevented from occurring. In order to keep information systems secure, illegal information flow among objects has to be prevented. Types of synchronization protocols are so far discussed based on read and write access rights in the RBAC (Role-Based Access Control) model to prevent illegal information flow.

In this thesis, we newly propose a P2PPSO (P2P type of topic-based PS (Publish/Subscribe) with Object concept) model and discuss the models and protocols for information flow control. A P2PPSO model is composed of peer processes (peers) which communicate with one another by publishing and subscribing event messages. Each peer can both publish and receive event messages with no centralized coordinator compared with traditional centralized PS models. Each event message published by a source peer carries information to a target peer. The contents carried by an event message are considered to be composed of objects. An object is a unit of data resource. Objects are characterized by topics, and each event message is also characterized by topics named publication topics.

In order to make a P2PPSO system secure, we first newly propose a TBAC

(Topic-Based Access Control) model. Here, an access right is a pair $\langle t, op\rangle$ of a topic t and a publish or subscribe operation op. A peer is allowed to publish an event message with publication topics and subscribe interesting topics only if the publication and subscription access rights are granted to the peer, respectively. Suppose an event message e_j published by a peer p_j carries an object on some topics into a target peer p_i. Here, information in the peer p_j illegally flows to the peer p_i if the target peer p_i is not allowed to subscribe the topics. An illegal object is an object whose topics a target peer is not allowed to subscribe. Even if an event message is received by a target peer by checking topics, objects carried by the event message may be illegal at the target peer. Hence, first, we propose a TOBS (Topics-of-Objects-Based Synchronization) protocol to prevent target peers from being delivered illegal objects in the P2PPSO system. Here, even if an event message is received by a target peer, illegal objects in the event message are not delivered to the target peer.

In the TOBS protocol, every event message is assumed to be causally delivered to every common target peer in the underlying network. Suppose an event message e_2 is delivered to a target peer p_i before another event message e_1 while the event message e_1 causally precedes the event message e_2 ($e_1 \rightarrow_c e_2$). Here, the event message e_2 is premature at the peer p_i. Hence, secondly, we propose a TOBSCO (TOBS with Causally Ordering delivery) protocol where the function to causally deliver every pair of event messages is added to the TOBS protocol. Here, we assume the underlying network supports reliable communication among every pair of peers, i.e. no event message loss, no duplicate message, and the sending order delivery of messages. Every pair of event messages received by using topics are causally delivered to every common target peer by using the vector of sequence numbers.

In the TOBS and TOBSCO protocols, objects delivered to target peers are held as replicas of the objects by the target peers. If a peer updates data of an object, the peer distributes event messages, i.e. update event messages, to update every replica of the object obtained by other peers. If a peer updates an object without changing topics, the object is referred to as altered. Here, an update event

message for the altered object is meaningless since peers check only topics to exchange event messages. Hence, thirdly, we propose an ETOBSCO (Efficient TOBSCO) protocol where update event messages of objects are published only if topics of the objects are updated to reduce the network overhead.

In the evaluation, first, we show how many numbers of event messages and objects are prevented from being delivered to target peers in the TOBS protocol. Next, we show every pair of event messages are causally delivered but it takes longer to deliver event messages in the TOBSCO protocol than the TOBS protocol. Finally, we show the fewer number of event messages are delivered while it takes longer to update replicas of altered objects in the ETOBSCO protocol than the TOBSCO protocol.

Keywords: Information flow control, Access control model, RBAC (Role-Based Access Control) model, Illegal information flow, P2P (Peer-to-Peer) model, PS (Publish/Subscribe) model, TBAC (Topic-Based Access Control) model, Causally ordering delivery, Meaningless update, TOBS (Topics-of-Objects-Based Synchronization) protocol, TOBSCO (TOBS with Causally Ordering delivery) protocol, ETOBSCO (Efficient TOBSCO) protocol

Chapter 1

Introduction

Distributed systems are composed of processes on computers which are interconnected in networks [1] and cooperate with one another to achieve some objectives. Here, a process is a unit of work, which is execution state of a program on a computer. An object is a unit of data resource which is an encapsulation of data and operations for manipulating the data [2]. Distributed systems are getting scalable like IoT (Internet of Things) which includes millions to billions of devices [3]. There are two types of distributed systems, CC (Cloud Computing) model [4] and P2P (Peer-to-Peer) model [5, 6]. In the CC model, each computer is either a service provider or a client. On the other hand, in the P2P model, each process is peer, i.e. each process can play both service provider and client roles. In addition, peer process (peer) is autonomous and there is no centralized coordinator. In this thesis, we consider the P2P type of distributed systems because it is more flexible, scalable, and reliable [6]. In distributed systems, peer processes (peers) are cooperating with one another by manipulating objects and exchanging messages in networks to realize some objective.

Distributed systems are required to be secure in addition to increasing the performance, reliability, and availability. In secure information systems [2], every data has to be manipulated by only the users which are allowed to access the data. For this aim, various types of methods are proposed, such as, cryptography [7, 8, 9], access control [10, 11, 12, 13], and so on. Cryptography is used to prevent every information from being stolen or disclosed without permission. Access control models [10] are used to make peers secure. Here, only a peer granted an access right on an object is allowed to manipulate the object. Even if a peer is not allowed to read data in an object o_i by the access control models [10], the peer can read the data by reading another object o_j if the data are written in the object o_j [2]. Here, illegal information flow occurs from the object o_i via the object o_j to the peer. We have to prevent illegal information flow among peers and objects in the access control models. In order to keep information systems secure by preventing illegal information flow among objects, types of protocols are proposed [14, 15, 16, 17] based on the RBAC (Role-Based Access Control) model [11, 12, 13, 18]. On the other hand, content-based systems like PS (Publish/Subscribe) systems [19, 20, 21, 22] are getting more important in various applications. Here, the PS model is an event-driven model [23] of a distributed system and a process is modeled to be a sequence of publication and receipt events of event messages. Peers publish and receive event messages in the publication and receipt events, respectively. Event messages published by a source peer carry objects to a target peer. Peers receive only event messages which carry objects in which the peers are interested. In the topic-based PS system [24, 25], objects are characterized by topics. Each event message is also characterized by topics named *publication* topics which are topics of objects carried by the event message. A peer only receives an event message whose publication topics interest the peer. Topics in which a peer is interested are *subscription* topics. Suppose an event message e carries objects. The publication topics of the event message e may not be the same as a collection of topics of all the objects. Here, even if a peer receives an event message in terms of the publication topics, the peer may not be allowed to take objects carried by the event message. We newly discuss how to prevent illegal information flow of objects among peers caused by publishing and receiving event messages.

In this thesis, we consider a P2PPSO (P2P (Peer-to-Peer) [6] of topic-based PS [24] with Object concept) model. Here, each peer can play both publisher and subscriber roles with no centralized coordinator. Peers exchange event messages with one another by publishing and receiving the event messages which carry objects. By exchanging event messages, objects are brought to the target peers of the event messages. On receipt of an event message with objects, a target peer holds replicas of the objects. Thus, replicas of an object are distributed to peers by exchanging event messages among peers. A peer which creates an object is referred to as a creator peer of the object. In this thesis, we assume that only the creator peer of an object can update data in the object. If a creator peer updates data in an object, topics may be changed as well as the data. Furthermore, every replica of the object in another peer is also required to be updated to keep every replica mutually consistent with the object. For instance, if a creator peer changes some data in an object, the data in every replica of the object is changed and then, the topics of the data are also changed from the topic sets of both the object and every replica of the object.

We newly propose a TBAC (Topic-Based Access Control) model to control publication and subscription of topics [26] by peers in topic-based PS systems [19, 20, 21, 22] in this thesis. Here, only a peer granted publication and subscription rights on a topic t is allowed to publish and subscribe an event message with the topic t, respectively. The topic sets $p_i.P$ and $p_i.S$ are sets of publication and subscription topics of a peer p_i, respectively. An event message e_i published by a peer p_i is delivered to a target peer p_j if the subscription $p_j.S$ and the publication $e_i.P$ include at least one common topic.

Suppose a peer p_i publishes an event message e_i including an object o whose data are related with a topic t in the P2PPSO system. Each object o is characterized by a set of topics. The topics show the meanings of the object o. We also suppose a target peer p_j receives the event message e_i but the subscription $p_j.S$ does not include the topic t. Here, the data of the object o on the topic t are delivered to the peer p_j although the peer p_j is not allowed to subscribe the topic t. An *illegal* object of a peer p_j is an object whose topics are not allowed to be subscribed by the peer p_j. An event message e_i is illegal at a target peer p_j if some objects carried by the event message e_i are illegal at the target peer p_j. Here, information of the peer p_i illegally flows to the peer p_j, i.e. illegal objects in the peer p_i are carried to the target peer p_j by the event message e_i. An event message e_i is delivered to a target peer p_j if the publication $e_i.P$ and the subscription $p_j.S$ have at least one common topic. However, even if an event message e_i is delivered to a target peer p_j, the event message e_i may carry illegal objects to the target peer p_j, i.e. the event message e_i is illegal.

In this thesis, we newly propose a TOBS (Topics-of-Objects Based Synchronization) protocol in order to prevent illegal objects from being delivered to target peers in the P2PPSO system. In the TOBS protocol, illegal objects are not delivered to target peers. The topics of objects carried by an event message e_i and the subscription of a target peer p_j of the event message e_i have to be compared to check whether or not the event message e_i carries illegal objects to the target peer p_j. Objects carried by event messages are stored in a storage of each target peer. Replicas of the object are required to be updated in the storage of a peer if the creator peer of the object updates the object. We propose a mechanism in the TOBS protocol to check if an object carried by an event message is illegal and to make every replica of an object distributed in peers mutually consistent.

In the P2PPSO system, event messages are required to be causally delivered to every common target peer to synchronize each object and every replica of the object because a peer may publish an event message after receiving another event message. Suppose an event message e_2 is delivered to a target peer p_i before another event message e_1 while the event message e_1 causally precedes the event message e_2 ($e_1 \rightarrow_c e_2$) [27]. Here, the event message e_2 is *premature* at the peer p_i. In the TOBS protocol, every event messages is assumed to be causally delivered to every common target peer in the underlying network. Hence, secondly, we propose a TOBSCO (TOBS with Causally Ordering delivery) protocol to causally deliver every pair of event messages. Here, we assume the underlying network supports reliable communication among every pair of peers, i.e. no event message loss, no duplicate message, and the sending order delivery of event messages. Every pair of event messages received by using topics are causally delivered to every common target peer by using the vector of sequence numbers.

If a peer updates data of an object, the peer distributes update event messages to every peer which holds a replica of the object to update the replica. Even if some data are updated in an object, any topics of the object may not be changed. If a peer updates an object without changing topics, the object is referred to as *altered*. Here, an update event message for the altered object is *meaningless* since peers check only topics to exchange event messages. In the TOBS and TOBSCO protocols, the meaningless update event messages are published. Hence, thirdly, we propose an ETOBSCO (Efficient TOBSCO) protocol where every peer does not publish meaningless event messages in order to reduce the network overhead.

We evaluate the TOBS, TOBSCO, and ETOBSCO protocols proposed in this thesis. First, we evaluate the TOBS protocol in terms of the numbers of illegal event messages and objects. We show how many event messages and objects which are not delivered to peers to prevent illegal information flow in the TOBS protocol. Next, we evaluate the TOBSCO protocol in terms of the number of premature event messages and delivery time of event messages. We show every pair of event messages are causally delivered but it takes longer to deliver event messages in the TOBSCO protocol than the TOBS protocol. Finally, we evaluate the ETOBSCO protocol in terms of the number of event messages delivered and update delay time of altered objects. We show the fewer number of event messages are delivered while it takes longer to update replicas of altered objects in the ETOBSCO protocol than the TOBSCO protocol.

The remaining part of this thesis is organized as follows.

In chapter 2, we overview research studies related with this thesis. Types of traditional access control models are discussed to make information systems secure. Based on the access control models, various types of approaches to preventing illegal information flow are presented. In addition, system models where the access control models and information flow controls are used are also described.

In chapter 3, we propose the P2PPSO model of a distributed system and the TBAC model as an access control model. We also discuss the causally ordered relation among event messages by taking advantage of the traditional causality theory. In the P2PPSO model, peers exchange objects. Each object is characterized by a set of topics. What objects each peer can publish and subscribe is determined by the TBAC model. Only a peer granted publication and subscription rights on a topic is allowed to publish and subscribe an event message with the topic. Event messages published may not be received by every common target peer in the same order because there is no centralized coordinator in the P2PPSO model. Hence, event messages are required to be causally delivered to every common target peer.

In chapter 4, we newly define the information flow relations, objects, and event messages based on the TBAC model. If an event message e_j carries an object o on topics, which a peer p_i is not allowed to subscribe, to the peer p_i, illegal information flow occurs. Here, the object o is illegal at the peer p_i. Also, the event message e_j is also illegal at the peer p_i because the event message e_j carries the illegal object o.

In chapter 5, we propose the TOBS, TOBSCO, and ETOBSCO protocols to prevent illegal information flow in the P2PPSO model. In the TOBS protocol, illegal objects are not delivered to the target peers in order to prevent illegal information flow. Here, the underlying network is assumed to support peers with the causally ordered delivery of event messages in addition to the reliable one-to-one communication. On the other hand, in the TOBSCO and ETOBSCO protocols, the underlying network is assumed to just support the reliable one-to-one communication. In the TOBSCO protocol, every event message is causally delivered on reliable one-to-one networks. If an object is altered, replicas on peers have to be updated. In the ETOBSCO protocol, update event messages are sent to peers holding replicas only if topics of the objects are changed to reduce the number of event messages.

In chapter 6, we evaluate the TOBS, TOBSCO, and ETOBSCO protocols proposed in this thesis. In order to evaluate the protocols, we develop a time-based simulator by using C language. About 30% of objects are illegal and not delivered to peers in the TOBS protocol. Every pair of event messages are causally delivered but it takes longer time to deliver event messages in the TOBSCO protocol than the TOBS protocol. Fewer number of event messages are delivered while it takes longer to update replicas of altered objects in the ETOBSCO protocol than the TOBSCO protocol.

In chapter 7, we conclude this thesis and discuss the future studies. In this thesis, the TBAC model is newly proposed for the topic-based PS model. We newly define information flow relations based on the TBAC model. In addition, we propose the TOBS, TOBSCO, and ETOBSCO protocols to prevent illegal information flow based on the information flow relations.

Chapter 2

Related Studies

2.1 Distributed systems

In the distributed systems [1] where multiple processes are cooperating, each process is autonomous because there is no centralized coordinator. The two main models of distributed systems are CC (Cloud Computing) model [4] and P2P (Peer-to-Peer) model [6]. A CC system is composed of a cloud of servers and clients. The cloud provides the ease to access shared resources and common infrastructure to offer services on demand to clients over the network. Servers in the cloud are cooperating with one another to meet the requests sent by clients. Here, there is no need to know specific locations of physical resources and devices accessed for clients. On the other hand, a P2P system is composed of peers which are interconnected in overlay networks. In P2P systems, multiple peers are cooperating with one another by exchanging messages in networks. A *peer* is an autonomous process which makes a decision by itself through communicating with other peers. There is no centralized coordinator and peers autonomously leave and join the system.

Distributed systems are getting scalable like IoT (Internet of Things) [3] which is composed of processes on not only computers like servers but also various types and millions of devices like sensors and actuators. Subjects like users and applications manipulate devices by issuing operations to the devices. For example, a subject gets data from a sensor and puts the data to an actuator to act the actuator based on the data. Here, data flow among subjects and devices.

In distributed systems, processes exchange messages with one another. Traditional networks like TCP [28] provide processes with reliable one-to-one communication. Here, messages sent by a process are delivered to another process in a sending order with neither message loss nor duplication. In distributed systems where more than two processes are cooperating with one another, messages are required to be causally delivered to destination processes. In paper [27], a partially ordered relation, i.e. happened-before relation ($\rightarrow_e$) on events is defined. For each peer p_i and message m, $s_i[m]$ and $r_i[m]$ show the sending and receipt events of the message m in the peer p_i, respectively. One sending event $s_i[m]$ exists for every receipt event $r_j[m]$. This means, a process p_i sends a message m and a process p_j receives the message m. For every pair of events e_1 and e_2, e_1 causally precedes e_2 ($e_1 \rightarrow_e e_2$) iff (if and only if) one of the following conditions holds:

1. The event e_1 happens before the event e_2 in the peer p_i.
2. For some peers p_i and p_j (not necessarily different), there is a message m such that $e_1 = s_i[m]$ and $e_2 = r_j[m]$.
3. There is an event e_3 such that $e_1 \rightarrow_e e_3$ and $e_3 \rightarrow_e e_2$.

A causal relation ($\rightarrow_c$) among messages is defined based on the happened-before relation [29]. A message m_1 *causally precedes* a message m_2 ($m_1 \rightarrow_c m_2$) iff $s_i[m_1] \rightarrow_e s_j[m_2]$ holds. In Figure 2.1 (1), both sending events $s_i[m_1]$ and $s_i[m_2]$ occur in the peer p_i. According to the condition 1, the sending event $s_i[m_1]$ happens before $s_i[m_2]$ ($s_i[m_1] \rightarrow_e s_i[m_2]$). Hence, $m_1 \rightarrow_c m_2$ holds. In Figure 2.1 (2), a process p_j receives a message m_1 sent by a process p_i. According to the condition 2, $s_i[m_1] \rightarrow_e r_j[m_1]$. Similarly, $r_j[m_1] \rightarrow_e s_j[m_2]$ and $s_j[m_2] \rightarrow_e r_k[m_2]$. According to the condition 3, $s_i[m_1] \rightarrow_e s_j[m_2]$. Hence, $m_1 \rightarrow_c m_2$ holds. The message m_1 may arrive at the process p_k after the message m_2 due to network delay. The message m_1 has to be delivered to the process p_k before the message m_2 since $m_1 \rightarrow_c m_2$. In order to causally deliver messages, types of logical clocks like linear clock [27] and vector clock [30] are proposed.

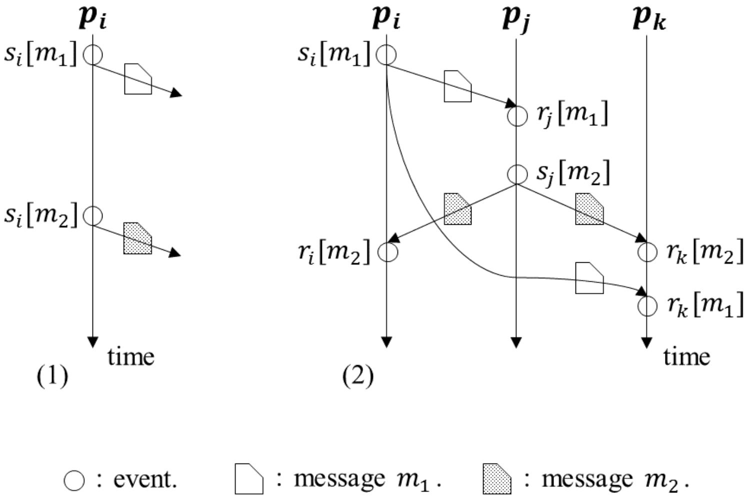

Figure 2.1: Causal relation among messages.

A PS (Publish/Subscribe) model [19, 20, 22, 31] is an event-driven, content-based model of a distributed system which is composed of processes interconnected in a network of brokers. There are publisher and subscriber processes [Figure 2.2]. A publisher process publishes an event message. An event message is delivered to only a subscriber process which is interested in the event message. In topic-based PS systems [24], a subscriber process specifies a subscription in terms of topics in which the subscriber process is interested. A publisher process publishes an event message with a publication which is also specified in terms of topics. If a publication of an event message and a subscription of the subscriber process have a common topic, the event message is delivered to the subscriber process. In this thesis, we discuss a P2PPS (P2P model of topic-based PS) system [25, 32, 33]. Here, every peer can publish and subscribe event messages and there is no centralized coordinator. In Figure 2.2, a process p_i publishes an event message on a topic t. The event message is delivered to a subscriber p_j which is interested in the topic t. On the other hand, the event message is not delivered to a subscriber p_k which is not interested in the topic t.

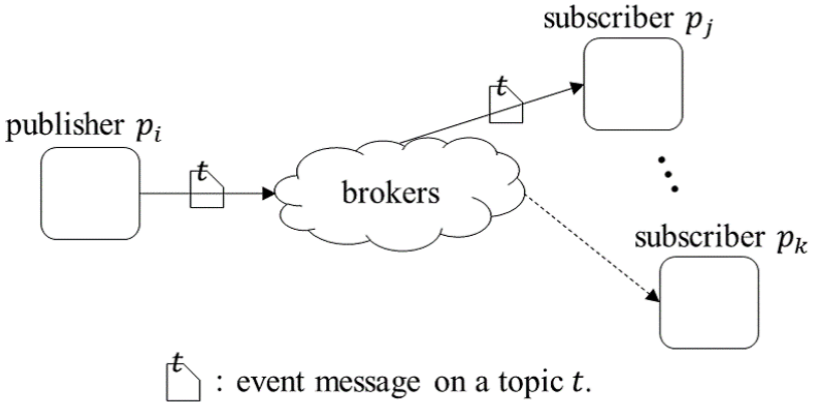

Figure 2.2: PS model.

2.2 Access control models

In distributed systems, processes exchange messages in networks to decide by themselves how to perform in the system because there is no centralized coordinator. Here, various types of information including confidential one are exchanged among entities of the system. Especially, it is critical to prevent illegal information flow from occurring. Illegal information flow means that an entity can get information even if the entity is not allowed to get the information in an access control model.

An information system is composed of *subjects* and *objects* [2]. An object is an encapsulation of data and operations to manipulate the data. A subject issues an operation to an object to manipulate the data. Then, the operation is performed on the object [2]. Users and transactions are examples of subjects. Databases and files are examples of objects. Let S and O be sets of subjects and objects in a system, respectively. Let OP be a set of operations on objects. Each object o supports a pair a of basic operations read (rd) and write (wr), i.e. $OP = \{rd$, $wr\}$. An access rule is a tuple $\langle s, o, op \rangle$ ($\in S \times O \times OP$) of a subject s, an object

o, and an operation op in the BAC (Basic Access Control) model [2]. An access rule $\langle s, o, op\rangle$ means that a subject s is allowed to manipulate an object o in an operation op. A pair $\langle o, op\rangle$ of an object o and an operation op is an *access right* (or permission). An authorizer grants an access right $\langle o, op\rangle$ to a subject s, i.e. an access rule $\langle s, o, op\rangle$ is specified by the authorization. A subject s is allowed to manipulate an object o in an operation op only if the subject s is granted an access right $\langle o, op\rangle$. Otherwise, the subject s is not allowed to manipulate the object o in the operation op. A system is *secure* iff every object o is manipulated by an authorized subject s in an authorized operation op according to an access rule $\langle s, o, op\rangle$.

In the RBAC (Role-Based Access Control) model [11, 12, 13] which is widely used in information systems like relational database systems [34], a role r ($\subseteq O \times OP$) is a set of access rights. An authorizer grants a role r, i.e. set of access rights to a subject s without granting each access right to the subject s. Each person plays a role r in a society, e.g. a president role in a company. Each role r shows what can be done by a subject which plays the role r in a society. Let R be a collection of roles in a system, $R \subseteq 2^{O \times OP}$. A subject s is granted a collection $s.R$ ($\subseteq R$) of roles and issues a transaction T to manipulate objects. Here, a transaction is a sequence of operations on objects [35]. A subject s grants a transaction T a subset $T.P$ ($\subseteq s.R$) of the roles $s.R$. A subset $T.P$ of the roles is referred to as *purpose* [36, 37] of the transaction T. A transaction T is allowed to issue an operation op to an object o only if an access right $\langle o, op\rangle$ is in the purpose $T.P$.

2.3 Information flow

Illegal information flow to occur in the access control models are discussed as confinement problem [2]. Suppose a subject s_i is granted a pair of a read access right $\langle f, rd\rangle$ on a file object f and a write access right $\langle g, wr\rangle$ on another file object g. Here, rd and wr show read and write operations, respectively, $OP = \{rd, wr\}$. Suppose another subject s_j is granted an access right $\langle g, rd\rangle$. Here, suppose the subject s_i reads data d in the file f and then writes the data d to the file g. The subject s_j is not allowed to read data in the file f. However, the subject s_j can obtain the data d in the file f by reading the data d stored in the file g. That is, information in the file f *illegally flows* into the subject s_j via the subject s_i and the file g.

In order to prevent illegal information flow, the LBAC (Lattice-Based Access Control) model [38] is proposed. Here, every entity e, i.e. subject or object belongs to a security class sc in a system. Let SC be a set of security classes. A legal information flow relation ($sc_1 \rightarrow sc_2$) from a security class sc_1 to a security class sc_2 ($\rightarrow \subseteq SC \times SC$) is defined by an administrator. The information flow relation $sc_1 \rightarrow sc_2$ means that information of an entity of a security class sc_1 is allowed to flow into an entity of a security class sc_2. Based on the information flow relation ($\rightarrow$), access rules are defined. Suppose a subject s and an object o belong to security classes sc_1 and sc_2, respectively. The subject s is allowed to read data in the object o if $sc_2 \rightarrow sc_1$. The subject s is allowed to write data to the object o if $sc_1 \rightarrow sc_2$. The subject s is allowed to modify the object o if $sc_1 \rightarrow sc_2$ and $sc_2 \rightarrow sc_1$.

In papers [36, 37, 39], the RBL (Role-Based Locking) protocol and scheduler of transactions are discussed to prevent illegal information flow to occur by performing transactions in the RBAC model [11, 12, 13]. Here, a role which includes more number of write access rights is more important. A transaction granted more important roles manipulates an object before another transaction.

In papers [14, 37], the illegal information flow relation from a role r_i to a role r_j ($r_i \mapsto r_j$) is defined. Let $In(r_i)$ and $Out(r_i)$ ($\subseteq O$) be sets of objects whose data are allowed to be read and written by a subject granted a role r_i, respectively, i.e. $In(r_i) = \{o \mid \langle o, rd \rangle \in r_i\}$ and $Out(r_i) = \{o \mid \langle o, wr \rangle \in r_i\}$. A role r_i *illegally flows* to a role r_j ($r_i \mapsto r_j$) iff $Out(r_i) \cap In(r_j) \neq \phi$ but $In(r_i) \not\subseteq In(r_j)$. Here, suppose a transaction T_1 with the role r_i reads data in an object o_1 and writes data to an object o_2. Here, some data x in the object o_1 may be brought to the object o_2. Then, suppose another transaction T_2 with the role r_j reads data in the object o_2. If the role r_j includes a read access right $\langle o_1, rd \rangle$, no illegal information flow occurs because the transaction T_2 is allowed to read data in the object o_1. However, if

$\langle o_1, rd \rangle \notin r_j$, the transaction T_2 may illegally get the data x from the object o_2 as shown in Figure 2.3.

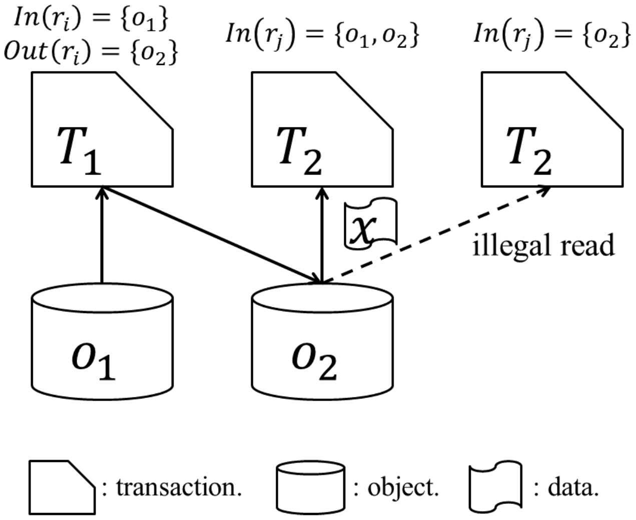

Figure 2.3: Legal and illegal information flow among objects.

A transaction *illegally reads* data in an object iff the transaction reads data in the object which includes data in another object which is not allowed to be read [14]. Allowable information relation flow from an object o_1 to an object o_2 is also *a priori* defined by an administrator. A transaction *suspiciously reads* data in an object iff the transaction reads data in the object whose data is not allowed to be brought to other objects [15]. A transaction *illegally writes* data to an object iff the transaction writes data to the object after illegally reading data in another object [Figure 2.4] [15]. A transaction *impossibly writes* data to an object iff the transaction writes data to the object after suspiciously reading data in another object [Figure 2.5] [15].

The WA (Write-Abortion) [15], RWA (Read-Write-Abortion) [16], and FRWA (Flexible Read-Write-Abortion) [17] protocols are proposed to prevent illegal information flow. For each object o_i and each transaction T_t, a pair of variables $o_i.R$ and $T_t.R$ are manipulated. The variables $o_i.R$ and $T_t.R$ denote roles in the role set R. Initially, the variable $o_i.R$ is empty and $T_t.R$ is a purpose $T_t.P$ of the

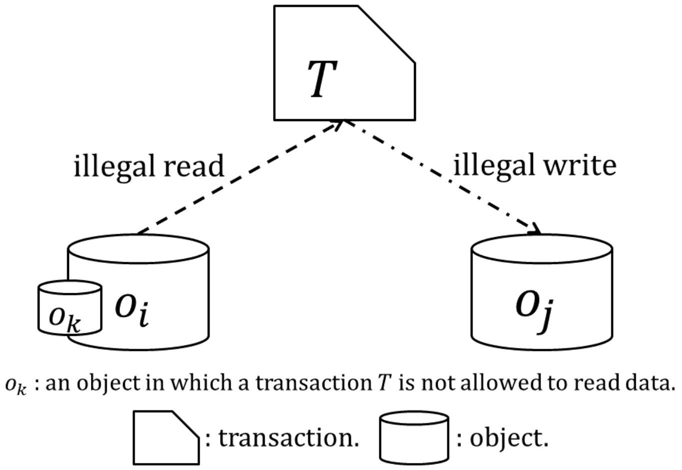

Figure 2.4: Illegal read and write operations.

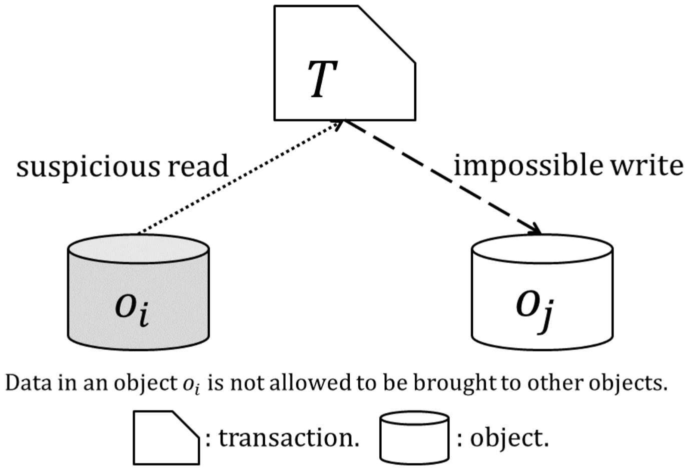

Figure 2.5: Suspicious read and impossible write operations.

transaction T_t. Each time a transaction T_t writes data to an object o_i, roles in the variable $T_t.R$ are added to the variable $o_i.R$, i.e. $o_i.R = o_i.R \cup T_t.R$. If a transaction T_t reads data in an object o_i, $T_t.R = T_t.R \cup o_i.R$. Here, if some role r_1 in $o_i.R$ illegally flows to a role r_2 in $T_t.R$ ($r_1 \mapsto r_2$), the read operation is illegal. In the WA protocol, a transaction aborts once issuing an illegal or impossible write operation to an object. Even if a transaction illegally reads data in an object, the transaction can commit if the transaction does not issue a write operation. Read operations performed after an illegal read operation before a write operation are *meaningless*. Because the transaction aborts once issuing the write operation and the read operations performed are rolled back. In the RWA protocol, a transaction aborts once issuing an illegal read operation or impossible write operation. Read operations are *lost*, which can be performed but are not performed after an illegal read operation is issued to an object. In the FRWA protocol, a transaction aborts if the transaction issues an illegal or impossible write operation to an object as well as the WA protocol. Furthermore, the transaction aborts with some probability ap once issuing an illegal read operation. If $ap = 1$ and $ap = 0$, the FRWA protocol is the same as the RWA protocol and the WA protocol, respectively.

The concepts of sensitivity of an object and safety of a role in the FRWA-O [40] and FRWA-RS [41] protocols are discussed. In the FRWA-O protocol, the abortion probability ap of a transaction T_t issuing an illegal read operation to an object o_i depends on the sensitivity of the object o_i. Here, the sensitivity of an object o_i just monotonically increases each time a transaction aborts by issuing an illegal read operation to the object even if the transaction commits. On the other hand, in the FRWA-RS protocol, the role safety of a role r_i increases and decreases each time a transaction T_t holding the role r_i commits and aborts, respectively, in order to reduce the number of transactions to abort. The abortion probability of each transaction T_t is decided by the role safety of roles in the variable $T_t.R$.

The SBS (Subscription-Based Synchronization) [26], TBS (Topic-Based Synchronization) [42], and FS-H (Flexible Synchronization for Hidden topics) [43] protocols are proposed to prevent illegal information flow caused by exchanging event messages carrying hidden topics in the P2PPS systems. If a peer p_i publishes an event message, the topics in the subscription $p_i.S$ of the peer p_i but not in the subscription $p_j.S$ of the target peer p_j are referred to as hidden topics for the peer p_j. In the SBS protocol, the delivery of an event message which may cause illegal information flow is prohibited. It is checked whether or not an event message causes illegal information flow in terms of subscription and publication access rights granted to each peer. This means, the topics in the subscription $p_i.S$ of a peer p_i indicates that data on the topics are obtained by the peer p_i. Here, even if the peer p_i neither subscribes some of the topics in the subscription $p_i.S$ nor obtains the data on the topics in reality, the data are considered as obtained by the peer p_i. Therefore, the data which the peer p_i does not obtain are considered as transferred to target peers by an event message published by the peer p_i. Hence, even the delivery of some legal event messages which indicate the topics of the data which are not carried by the event messages in reality is unnecessarily prohibited. On the other hand, in the TBS protocol, it is checked whether or not an event message causes illegal information flow in terms of topics which are really manipulated by each peer. Hence, a fewer number of event messages are prohibited than the SBS protocol because only and every illegal event messages are prohibited differently from the SBS protocol. In the FS-H protocol, even if an event message carries hidden topics which are strongly related with some topics subscribed by a target peer, the event message is delivered to the target peer and the hidden topics are added to the subscription of the target peer. The number of event messages prohibited is more reduced by using the learning mechanism compared with the SBS and TBS protocols.

Chapter 3

System Models

3.1 P2PPSO (P2P (Peer-to-Peer) model of a topic-based PS (Publish/Subscribe) with Object concept) model

A PS (Publish/Subscribe) model is a model of a content-based system [44, 45]. In traditional PS systems, each process is either a publisher or a subscriber [19, 20, 22, 31]. Event messages published by publishers are first sent to a broker network. Then, the broker network delivers event messages published by publishers to subscribers [Figure 3.1]. Here, event messages are delivered to subscribers in the receipt order of the broker network.

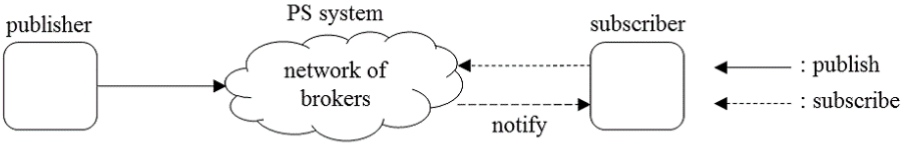

Figure 3.1: Centralized PS system.

On the other hand, in a P2PPS (P2P (Peer-to-Peer) model of a topic-based PS) system [32, 33], each process is peer, i.e. can play both publisher and subscriber roles. A P2PPS system is composed of peer processes (peers) p_1, . . ., p_{pn} ($pn \geq 1$). Let P be a set $\{p_1, \ldots, p_{pn}\}$ of all the peers in a system. Furthermore, there is no centralized coordinator like brokers. A peer p_i publishes an event message e. Then, a peer p_j receives the event message e only if the peer p_j is interested in the contents of the event message e. Here, the peer p_j is a target peer of the event message e.

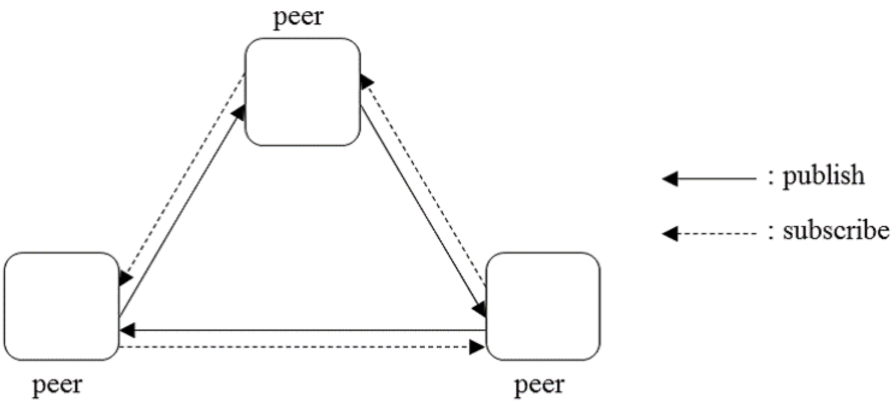

Figure 3.2: P2PPS system.

Event messages published by a peer are delivered to every target peer in the publishing order. However, a pair of event messages e_i and e_j published by different peers p_i and p_j may be delivered to different target peers in different orders. In papers [32, 33], the authors propose how to causally [27] deliver event messages related with topics to target peers by using the topic vector and physical time.

In the P2PPS system, event messages in which each peer is interested are characterized by topics [24]. Let T be a set $\{t_1, \ldots, t_{tn}\}$ ($tn \geq 1$) of all topics in a system. A peer p_i specifies the publication $e.P$ for an event message e in a subset of the topic set T ($e.P \subseteq T$). Each peer p_i then publishes an event message e with the publication $e.P$. Each peer p_i also specifies the subscription $p_i.S$ in a subset of the topic set T ($p_i.S \subseteq T$). An event message e is delivered to a target peer p_i if the publication $e.P$ and the subscription $p_i.S$ include at least one common topic, i.e. $e.P \cap p_i.S \neq \phi$. Here, the peer p_i is a *target* peer of the event message e. A peer which publishes an event message e is a *source* peer of the event message

e. Only an event message e which includes interesting information, i.e. whose publication $e.P$ includes a topic subscribed by the peer p_i, i.e. the topic is in $p_i.S$ is delivered to a each peer p_i.

In this thesis, we consider P2PPSO (P2PPS with Object concept) model where the contents of an event message are composed of objects. A peer p_i includes objects in an event message e and then publishes the event message e. An object is an unit of data resource. Let $e.O$ be a set of objects carried by an event message e. If the event message e is delivered to another peer p_j, the objects carried by the event message e are stored as replicas of the objects in the storage $p_j.D$ of the peer p_j. Let o_i be an object created by a peer p_i. Let o_i^j show a replica of an object o_i which is held by a peer p_j, i.e. $o_i^j \in p_j.D$. Here, o_i^i stands for an object o_i. Thus, replicas of objects are distributed to peers by publishing and receiving event messages. We have to discuss whether or not a peer can deliver objects in addition to receiving each event message. A set $o_i^j.T$ indicates topics of a replica o_i^j. Here, $e.T$ shows a set of topics of objects which an event message e carries, i.e. $e.T = \cup_{o \in e.O}\ o.T$. If an event message e_i from a peer p_i is delivered to a peer p_j, a replica o_k^i of each object o_k in the set $e_i.O$ is stored in the storage $p_j.D$. Here, $o_k^j.T = o_k^i.T$. The peer p_j is a holder peer of a replica o_k^j. The notations used in this thesis are summarized as follows:

- $T =$ set $\{t_1, \ldots, t_{tn}\}$ of topics in a system.
- $p_i.P =$ topics which a peer p_i is allowed to publish ($\subseteq T$).
- $p_i.S =$ topics which a peer p_i is allowed to subscribe ($\subseteq T$).
- $p_i.D =$ objects obtained by a peer p_i ($\subseteq O$).
- $e.P =$ topics which characterize an event message e ($\subseteq T$).
- $O =$ set of objects in a system.
- $o_i =$ an object created by a peer p_i ($\subseteq O$).
- $o_i^i =$ an object o_i ($\subseteq O$).

- o_i^j = an object whose creator is a peer p_i, obtained by a peer p_j, i.e. replica of an object o_i^i ($\subseteq p_j.D$, O).
- $o_i^j.T$ = topics of an object o_i^j ($\subseteq T$).
- $e.O$ = objects carried by an event message e ($\subseteq O$).
- $e.T$ = topics of objects carried by an event message e, i.e. $e.T = \cup_{o \in e.O} o.T$ ($\subseteq T$).

Each object o_i is created by a peer p_i. Here, the peer p_i is a creator peer of the object o_i. We assume only a creator peer p_i of each object o_i can update the object o_i in this thesis. A holder peer p_j of a replica o_i^j cannot update the replica o_i^j. There are two cases. In one case, only the state of the object o_i is updated but the topics $o_i.T$ are not changed. In another case, both the state and topics of the object o_i are changed. A replica o_i^j is *strictly consistent* with an object o_i iff $o_i^j = o_i$ and $o_i^j.T = o_i.T$. In order to keep a replica o_i^j strictly consistent with an object o_i, once the object o_i is updated, both the state o_i^j and topics $o_i^j.T$ have to be updated. In the topic-based PS systems, only topics are manipulated to publish and receive event messages. It is significant to know whether or not the topics of each replica o_i^j are the same as the object o_i. A replica o_i^j is *weakly consistent* with an object o_i iff $o_i^j.T = o_i.T$. Even if the state of a replica o_i^j is not the same as an object o_i, i.e. $o_i^j \neq o_i$ the replica o_i^j can be considered to be consistent with the object o_i in the topic-based systems only if o_i^j is weakly consistent with o_i, i.e. $o_i^j.T = o_i.T$.

3.2 TBAC (Topic-Based Access Control) model

It is significant how to control publication and subscription rights in the PS model. However, access control models on the PS model are so far not discussed. In this thesis, we newly propose a TBAC (Topic-Based Access Control) model [26] to make clear authorized access in the P2PPSO system. Let OP be a set of operations, i.e. $OP = \{$publish (*pb*), subscribe (*sb*)$\}$. Let T and P be sets of topics and peers, respectively, in a PS system S. A TBAC access rule $\langle p_i$, t, $op\rangle$ ($\in P \times T \times OP$) means that a peer p_i is allowed to manipulate a topic t in an operation op. Here, an operation op is a publish (pb) or subscribe (sb) operation, i.e. $op \in OP$. Let A be a set of access rules in a system. A pair $\langle t$, $op\rangle$ of a topic t and an operation op shows an access right in the TBAC model. In this thesis, we assume a centralized authorizer grants a peer p_i an access right $\langle t$, $op\rangle$ ($\in T \times OP$) where t is a topic ($t \in T$) and op is an operation ($op \in OP$). A peer p_i is allowed to perform an operation op on a topic t only if $\langle p_i$, t, $op\rangle \in A$, i.e. an access right $\langle t$, $op\rangle$ is granted to the peer p_i.

A peer p_i is allowed to publish an event message e with publication $e.P$ ($\subseteq T$) only if the peer p_i is granted an access right $\langle t$, $pb\rangle$ for every topic t in the publication $e.P$. The publication $p_i.P$ ($\subseteq T$) of a peer p_i is a subset $\{t \mid \langle p_i$, t, $pb\rangle \in A$, i.e. an access right $\langle t$, $pb\rangle$ is granted to the peer $p_i\}$ of topics on which the peer p_i is allowed to publish an event message.

A peer p_i is allowed to subscribe a topic t only if an access right $\langle t$, $sb\rangle$ is granted to the peer p_i. The subscription $p_i.S$ ($\subseteq T$) of a peer p_i is a subset of topics on which a peer p_i is allowed to receive event messages, i.e. $\{t \mid \langle t$, $sb\rangle$ is granted to p_i, i.e. $\langle p_i$, t, $sb\rangle \in A$ $\}$.

Suppose a peer p_i publishes an event message e with a publication $e.P$ ($\subseteq p_i.P$). Here, the peer p_i is allowed to publish an event message e only if the publication $e.P$ is a subset of the publication $p_i.P$, i.e. $e.P \subseteq p_i.P$. The subscription $p_j.S$ of a peer p_j shows topics in which the peer p_j is interested. That is, an event message e is delivered to a peer p_j if $e.P \cap p_j.S \neq \phi$. A peer p_j is a $target$ peer of an event message e iff $e.P \cap p_j.S \neq \phi$, i.e. the subscription $p_j.S$ of a peer p_j has a common topic with the publication $e.P$ of an event message e. Here, an event message e is only delivered to a target peer p_j in a system.

3.3 Causally ordered relation of event messages

In the P2PPSO system where there is no centralized coordinator, event messages may not be received by every common target peer in the same order. For instance, a target peer p_i receives an event messages e_1 before an event message e_2 while another target peer p_j receives the event message e_1 after the event message e_2, respectively. According to the causality theory [27], an event message e_1 *causally precedes* another event message e_2 ($e_1 \rightarrow_c e_2$) iff the publication event of e_1 happens before e_2. Hence, if the event message e_1 is published before the event message e_2 ($e_1 \rightarrow_c e_2$), the event message e_1 has to be delivered before the event message e_2 in the peer p_j even if the peer p_j receives the event message e_1 after the event message e_2. If the event message e_2 is delivered to the peer p_j before the event message e_1, the event messages e_1 and e_2 are delivered to the peer p_j out of publication order.

Definition 1. A peer p_i *correctly receives* an event message e_j iff (if and only if) p_i receives e_j and p_i knows every target peer receives e_j.

Definition 2. An event message e_j is *matured* at a destination peer p_i iff p_i *correctly receives* e_j and every event message received by p_i which causally precedes e_j is delivered to p_i.

Definition 3. An event message e_j is *premature* at a peer p_i iff e_j is delivered to p_i although e_j is not matured.

In Figure 3.3 (1), a pair of peers p_j and p_k publishes event messages after receipt of an event message e_j. After a peer p_i receives both event messages published by the peers p_j and p_k, the peer p_i knows the peers p_j and p_k already receive the event message e_j. Hence, the peer p_i correctly receives the event message e_j. In Figure 3.3 (2), an event message e_k such that e_k causally precedes e_j ($e_k \rightarrow_c e_j$) is delivered to the peer p_i before the peer p_i correctly receives the event message e_j. Hence, the event message e_j is matured. In Figure 3.3 (3), the event message e_j is delivered to the peer p_i before the the event message e_k although $e_k \rightarrow_c e_j$ holds. Hence, the event message e_j is premature.

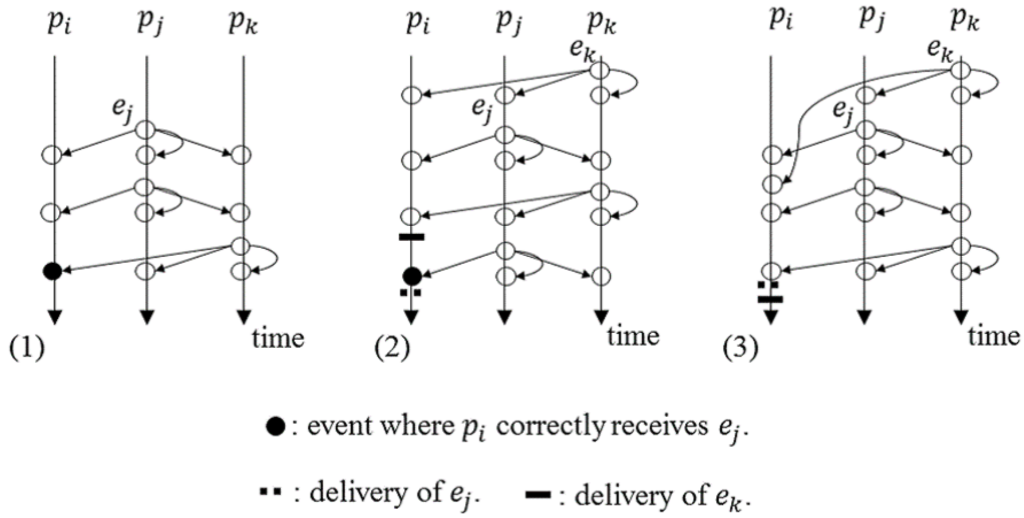

Figure 3.3: Causally precedent relation of event messages.

Chapter 4

Information Flow

4.1 Information flow relations

In the P2PPSO system, event messages carry objects to target peers. In order to check if each peer can take objects carried by event messages, the peer has to keep in record the objects and compare the objects with objects kept in the peer. However, it is not easy for each peer to hold every object carried by event messages. Each object o_i^j is characterized in terms of topics and has a variable $o_i^j.T$ which denotes topics. Topics in the variable $o_i^j.T$ indicate what data the object o_i^j includes. Each event message e_j has a variable $e_j.T$. The variable $e_j.T$ shows what data every object o_i^j in the event message e_j includes. A set $p_i.S$ of topics of the peer p_i indicates topics which the peer p_i is allowed to subscribe.

In this section, we define information flow relations on objects and topics based on the TBAC model. First, an information flow relation ($\rightarrow$) on event messages and peers is defined as follows:

Definition 4. Let e_i be an event message published by a peer p_i. The event message e_i *flows* to a peer p_j ($e_i \rightarrow p_j$) iff $e_i.O \neq \phi$ and $e_i.P \cap p_j.S \neq \phi$.

If an event message e_i flows to a peer p_j ($e_i \rightarrow p_j$), the event message e_i published by the peer p_i can be delivered to the peer p_j. Here, some information obtained by the peer p_i flows into the peer p_j. Otherwise, no information from the peer p_i flows into the peer p_j because the event message e_i is not delivered to the peer p_j.

Next, a legal information flow relation ($\Rightarrow$) on event messages and peers is defined as follows:

Definition 5. Let e_i be an event message published by a peer p_i. The event message e_i *legally flows* to a peer p_j ($e_i \Rightarrow p_j$) iff $e_i \rightarrow p_j$ and $e_i.T \subseteq p_j.S$.

The condition $e_i.T \subseteq p_j.S$ shows that every topic in the variable $e_i.T$ is also in the subscription $p_j.S$. This means, the event message e_i carries no data on topics which the target peer p_j is not allowed to subscribe. Hence, no information illegally flows into the peer p_j by delivering the event message e_i.

Finally, an illegal information flow relation ($\mapsto$) on event messages and peers is defined as follows:

Definition 6. Let e_i be an event message published by a peer p_i. The event message e_i *illegally flows* to a peer p_j ($e_i \mapsto p_j$) iff $e_i \rightarrow p_j$ and $e_i.T \not\subseteq p_j.S$.

The condition $e_i.T \not\subseteq p_j.S$ means that the event message e_i carries objects on topics, which the target peer p_j is not allowed to subscribe into the target peer p_j.

4.2 Objects

An event message carries objects in a source peer p_i to a target peer p_j. The target peer p_j has to decide if the objects can be delivered to the peer p_j. We define legal and illegal objects in terms of the topics. First, a legal object o^i for the target peer p_j ($o^i \overset{o}{\Rightarrow} p_j$) is defined as follows:

Definition 7. Let o^i be an object carried by an event message e_i published by a peer p_i [Figure 4.1]. Here, the object o^i is *legal* at a target peer p_j of the event message e_i ($o^i \overset{o}{\Rightarrow} p_j$) iff $o^i \in e_i.O$, $e_i \rightarrow p_j$, and $o^i.T \subseteq p_j.S$.

Next, an illegal object o^i for the target peer p_j ($o^i \overset{o}{\mapsto} p_j$) is defined as follows:

Definition 8. Let o^i be an object carried by an event message e_i published by a peer p_i [Figure 4.1]. Here, the object o^i is *illegal* at a target peer p_j of the event message e_i ($o^i \overset{o}{\mapsto} p_j$) iff $o^i \in e_i.O$, $e_i \rightarrow p_j$, and $o^i.T \not\subseteq p_j.S$.

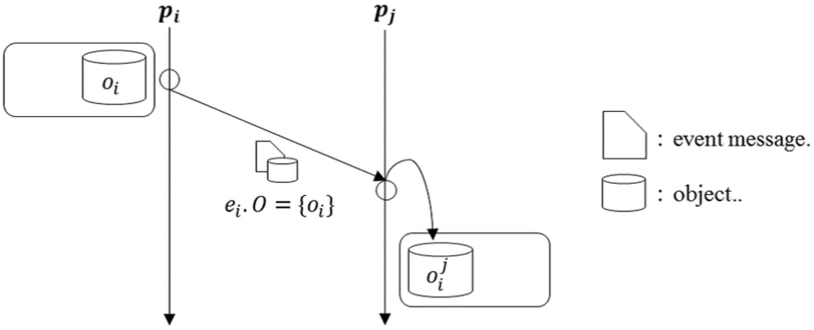

Figure 4.1: Object replication.

4.3 Event messages

Then, we define legal and illegal event messages in terms of the objects carried by the event messages. An legal event message e_i for the target peer p_j is defined as follows:

Definition 9. Let e_i be an event message published by a peer p_i and p_j be a target peer of the event message e_i. The event message e_i is *legal* at the target peer p_j iff the event message e_i carries no illegal object to the peer p_j.

Next, an illegal event message e_i for the target peer p_j is defined as follows:

Definition 10. An event message e_i is *illegal* at a target peer p_j iff the event message e_i is not legal at the peer p_j.

Example 1. Suppose there are three peers p_i, p_j, and p_k and three topics x, y, and z in a system, i.e. $P = \{p_i, p_j, p_k\}$ and $T = \{x, y, z\}$ as shown in Figure 4.2. We also suppose $p_i.S = p_i.P = \{x, y\}$, $p_j.S = p_j.P = \{x, y, z\}$, and $p_k.S = p_k.P = \{y, z\}$.

First, a pair of peers p_i and p_j create objects o_i^i and o_j^j and then store the objects o_i^i and o_j^j in their storage $p_i.D$ and $p_j.D$, respectively. The objects o_i^i and o_j^j include data on a pair of topics x and y and a pair of topics y and z, respectively, i.e. $o_i^i.T = \{x, y\}$ and $o_j^j.T = \{y, z\}$.

Next, the peer p_i publishes an event message e_i where $e_i.P = \{x\}$, $e_i.O = \{o_i^i\}$, and $e_i.T = o_i^i.T = \{x, y\}$. Since $e_i.O = \{o_i^i\} \neq \phi$ and $e_i.P \cap p_j.S = \{x\} \neq \phi$ ($e_i \rightarrow p_j$), the event message e_i is delivered to the peer p_j. In addition, since $e_i.T \subseteq p_j.S$ ($e_i \Rightarrow p_j$), the peer p_j is allowed to subscribe every topic carried by the event message e_i. Here, the object o_i^i is legal at the peer p_j ($o_i^i \overset{o}{\Rightarrow} p_j$). Since $e_i.O = \{o_i^i\}$, the event message e_i is legal at the peer p_j. Hence, no information illegally flows to the peer p_j from the peer p_i. The peer p_j stores the object o_i^i in its storage $p_j.D$. Here, $p_j.D = \{o_i^j, o_j^j\}$.

Next, the peer p_j includes a pair of objects o_i^j and o_j^j into an event message e_j and publishes the event message e_j where $e_j.P = \{z\}$, $e_j.O = \{o_i^j, o_j^j\}$, and $e_j.T = o_i^j.T \; (= \{x, y\}) \cup o_j^j.T \; (= \{y, z\}) = \{x, y, z\}$. Since $e_j \rightarrow p_k$, the event message e_j is delivered to the peer p_k. However, the peer p_k is not granted the subscription right $\langle x, sb\rangle$, i.e. $e_j.T \nsubseteq p_k.S$. Hence, $e_j \mapsto p_k$. The pair of objects o_i^j and o_j^j are illegal and legal at the peer p_k, respectively ($o_i^j \overset{o}{\mapsto} p_k$, $o_j^j \overset{o}{\Rightarrow} p_k$). The event message e_j is illegal at the peer p_k because the event message e_j carries the illegal object o_i^j. This means, data on the topic x which the peer p_k is not allowed to subscribe can be delivered to the peer p_k via the peer p_j. Here, information illegally flows to the peer p_k from the peer p_j.

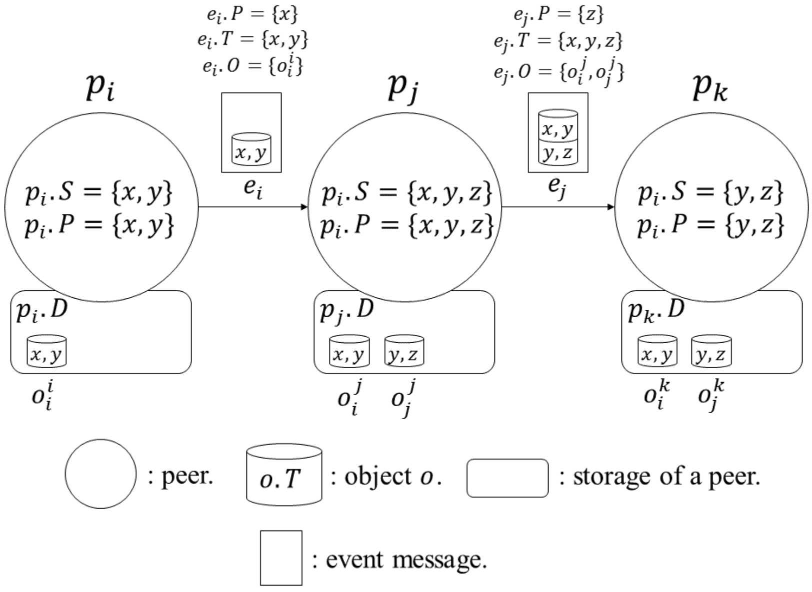

Figure 4.2: Information flow among peers.

Chapter 5

Protocols for Information Flow Control

5.1 Protocol stack

In this chapter, we discuss how to prevent illegal information flow from occurring in the P2PPSO systems. The illegal information flow relation ($\mapsto$) on peers and objects is defined based on the TBAC model in section 4.1. Information illegally flows to a target peer if an event message carries at least one illegal object into the target peer, i.e. the event message is illegal at the target peer.

In order to prevent illegal information flow, we propose a TOBS (Topics-of-Objects Based Synchronization) [46], TOBSCO (TOBS with Causally Ordering delivery) [47], and ETOBSCO (Efficient TOBSCO) [48] protocols in this thesis. The protocols are realized by taking advantage of underlying services. Figure 5.1 shows the protocol stack of the TOBS, TOBSCO, and ETOBSCO protocols. The protocol stack is composed of the following protocols:

- Network: The underlying network provides peers with the reliable one-to-one communication like TCP [28]. Here, a pair of event messages published by a common peer are delivered in sending order without message loss and duplication.

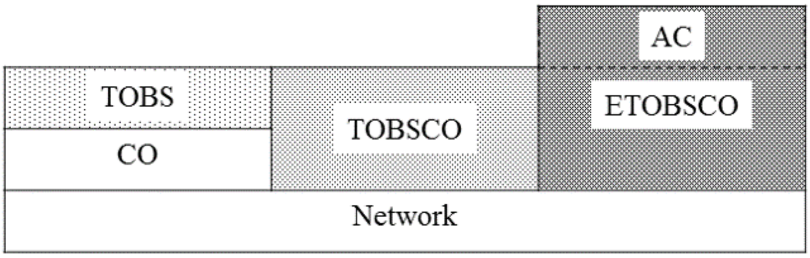

Figure 5.1: Protocol stack.

- CO (Causally Ordering): The causally ordered delivery of event messages [27] is provided on the underlying network service. Here, even if a pair of event messages e_1 and e_2 such that e_1 causally precedes e_2 ($e_1 \rightarrow_c e_2$) are published by different peers, the event message e_1 is delivered before the other event message e_2 to every common target peers.
- AC (Alteration Check): After a peer updates an object, it is checked whether or not the topics of the object are also updated. Here, if the topics are not updated, the object is referred to as altered. Only if the topics of the object are not changed, i.e. the object is altered the peer avoids publishing an update event message to reduce the network overhead.

In the TOBS, TOBSCO, and ETOBSCO protocols, only objects which interest the peer and are legal can be delivered. In the TOBS protocol, it is assumed the causal delivery service is supported by the underlying system. On the other hand, in the TOBSCO protocol, a function to causally deliver event messages is implemented. In the ETOBSCO protocol, the alteration check function is implemented in addition to the causal delivery function. If a peer updates only the state of an object, i.e. the peer alters the object the peer avoids publishing update event messages for the altered object. Hence, the network overhead is reduced since the number of update event messages published is reduced.

5.2 TOBS (Topics-of-Objects Based Synchronization) protocol

In this section, we propose a TOBS (Topics-of-Objects Based Synchronization) protocol [46] to prevent illegal information flow in the P2PPSO system based on the illegal information flow relation. In the P2PPSO system, a peer p_i is granted topics in its publication $p_i.P$ and subscription $p_i.S$. A peer p_i is allowed to publish and subscribe a topic t in the publication $p_i.P$ and subscription $p_i.S$, respectively. $e.T$ shows a set of topics carried by the event message e. Here, $e.T$ is composed of topics of the objects included in the event message e. In the TOBS protocol, on receipt of an event message e_i, the target peer p_j checks the condition $e_i.T \subseteq p_j.S$ to detect the event message e_i includes illegal objects. If the event message e_i carries illegal objects to a target peer p_j, the objects are not delivered to the target peer p_j while legal objects are delivered to the target peer p_j. Here, objects obtained by each peer through receiving event messages are synchronized so that no illegal object is delivered to target peers. If a peer p_i updates data in an object o_i^i, the peer p_i publishes an *update* event message ue_i to make mutually consistent every replica o_i^j of the object o_i^i obtained by each peer p_j. In this thesis, we assume only the creator peer p_i of the object o_i^i can update the data in the object o_i. If an object o_i^i is updated by a creator peer p_i, every replica o_i^j of the object o_i^i is also updated.

Algorithms 1, 2, and 3 show how each peer behaves in the TOBS protocol. A peer p_i publishes an event message e_i as shown in Algorithm 1. An event message e_j from a peer p_j is delivered to a target peer p_i as shown in Algorithm 2. A peer p_i updates an object o_i^i in its storage $p_i.D$ as shown in Algorithm 3.

In the P2PPSO system, if a peer p_i updates data in an object o_i^i, every replica o_i^j of the object o_i^i obtained by every other peer p_j $(i \neq j)$, is synchronized to be consistent with the object o_i^i. In the TOBS protocol, every replica o_i^j of the object o_i^i is updated through exchanging event messages among peers.

Example 2. Suppose there are three peers p_i, p_j, and p_k as shown in Figure 5.2. We also suppose the peers p_i, p_j, and p_k have publications and subscriptions as

Algorithm 1: Publication of a peer p_i

```
$e_i.O$ = set of objects or replicas included in the event message $e_i$ from the storage $p_i.D$, i.e. $e_i.O \subseteq p_i.D$;
$e_i.T = \{t \mid t \in o^i.T$ and $o^i \in e_i.O\}$;
$e_i.P$ = publication topics of $e_i$, i.e. $e_i.P \subseteq p_i.P$;
$p_i$ **publishes** the event message $e_i$;
```

$p_i.P = p_i.S = \{y, z\}$, $p_j.P = p_j.S = \{x, y, z\}$, and $p_k.P = p_k.S = \{x, y\}$, respectively.

First, a pair of peers p_j and p_k create objects o^j_j and o^k_k where $o^j_j.T = \{y, z\}$ and $o^k_k.T = \{x\}$. The peers p_j and p_k store the objects o^j_j and o^k_k in their storages $p_j.D$ and $p_k.D$, respectively. Next, the peer p_k publishes an event message e_k where $e_k.O = \{o^k_k\}$, $e_k.T = o^k_k.T = \{x\}$, and $e_k.P = \{x\}$. Here, the event message e_k flows to the peer p_j ($e_k \rightarrow p_j$). In addition, since the condition $e_k.T \subseteq p_j.S$ is satisfied, the event message e_k legally flows to the peer p_j ($e_k \Rightarrow p_j$). Therefore, the event message e_k is delivered to the peer p_j and the replica of the object o^k_k is stored in the storage $p_j.D$ of the peer p_j. Here, $p_j.D = \{o^j_j, o^j_k\}$.

Next, suppose the peer p_k updates data in the object o^k_k. The data of the object o^k_k on the topic x are changed with the data on a pair of the topics x and y. Here, the variable $o^k_k.T$ is changed with the topics $\{x, y\}$. The peer p_k publishes an update event message ue_k to make another peer p_m synchronize the replica o^m_k with the object o^k_k. Here, the publication $ue_k.P$ is same as the variable $o^k_k.T$ of the unupdated object o^k_k, i.e. $ue_k.P = \{x\}$. Since the update event message ue_k legally flows to the peer p_j ($ue_k \Rightarrow p_j$), the replica o^j_k in the storage $p_j.D$ of the peer p_j is updated. Hence, $o^j_k.T$ ($= \{x\}$) is changed with topics $\{x, y\}$.

Then, the peer p_j publishes an event message e_j where $e_j.O = \{o^j_j, o^j_k\}$, $e_j.T = o^j_j.T \cup o^j_k.T = \{x, y, z\}$, and $e_j.P = \{z\}$. Here, the event message e_j flows to the peer p_i ($e_j \rightarrow p_i$). However, the peer p_i is not allowed to subscribe the topic x in the variable $e_j.T$, i.e. the event message e_j illegally flows to the peer p_i ($e_j \mapsto p_i$). Here, the replica o^j_k is illegal at the target peer p_i because $o^j_k.T \not\subseteq p_i.S$ and the replica o^j_k is not delivered to the peer p_i. On the other hand, the object o^j_j is

Algorithm 2: Delivery of an event message e_j to a peer p_i

```
if $e_j \Rightarrow p_i$ then
    if $e_j$ is an update event message $ue_j$ then
        $o_j^i = o_j^j$;
        $o_j^i.T = o_j^j.T$;
    else
        if $o_k^j \in e_j.O$ ($k \neq i$) then
            if $o_k^i \in p_i.D$ then
                $o_k^i = o_k^j$;
                $o_k^i.T = o_k^j.T$;
            else
                add replica $o_k^j$ to $p_i.D$;
else
    if $e_j$ is an update event message $ue_j$ then
        if $o_j^j \overset{o}{\Rightarrow} p_i$ then
            $o_j^i = o_j^j$;
            $o_j^i.T = o_j^j.T$;
        else
            delete every replica $o_j^i$ from $p_i.D$;
    else
        if $o_k^j \overset{o}{\Rightarrow} p_i$ ($k \neq i$) then
            if $o_k^j \in e_j.O$ then
                if $o_k^i \in p_i.D$ then
                    $o_k^i = o_k^j$;
                    $o_k^i.T = o_k^j.T$;
                else
                    add replica $o_k^j$ to $p_i.D$;
        else
            delete every replica $o_k^i$ from $p_i.D$;
```

Algorithm 3: Update of an object o_i^i

```
$p_i$ makes an update event message $ue_i$ where $ue_i.O = \{o_i^i \mid o_i^i$ is being updated by $p_i\}$, $ue_i.T = \{t \mid t \in o_i^i.T$ and $o_i^i \in ue_i.O\}$, and $ue_i.P = ue_i.T$;
$p_i$ **updates** data in an object $o_i^i$ where $o_i^i.T \subseteq p_i.S$;
$ue_i.O = \{o_i^i \mid o_i^i$ is an updated object$\}$;
$ue_i.T = \{t \mid t \in o_i^i.T$ and $o_i^i \in ue_i.O\}$;
$p_i$ **publishes** an update event message $ue_i$;
```

delivered to the peer p_i because $o_j^j.T \subseteq p_i.S$, i.e. the object o_j^j is legal.

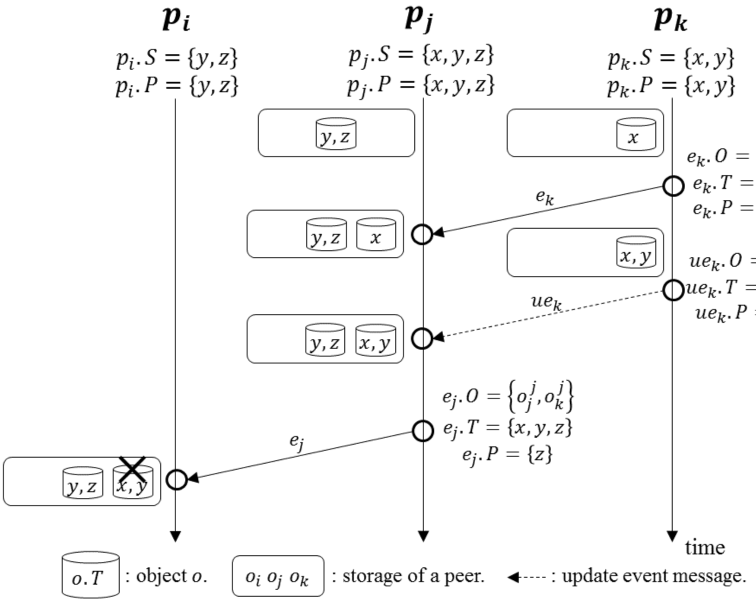

Figure 5.2: TOBS protocol.

In the SBS [26], TBS [42], and FS-H [43] protocols which we have so far proposed, every data exchanged among peers is not regarded as a collection of objects. Data obtained by each peer are denoted by topics. Even if a peer removes data after getting the data, the topics of the data are not deleted. This means, the number of topics carried into every peer monotonically increases. Here, even if data related with the topics which a target peer can subscribe are carried into the target peer, the data might be prohibited to be delivered to the peer because data carried into the peer are denoted by the topics which a source peer already obtains. Hence, some legal information flow is referred to as illegal.

5.3 TOBSCO (TOBS with Causally Ordering delivery) protocol

In the TOBS protocol, every event message is assumed to be causally delivered to every common target peer in the underlying networks. Hence, as soon as an event message arrives at a peer, the event message is delivered to the peer. Suppose a peer p_i receives an event message e_1 before another event message e_2. We also suppose another peer p_j receives the event messages e_2 before the event message e_1. Here, even if the event message e_1 causally precedes the other event message e_2 ($e_1 \rightarrow_c e_2$), the event message e_2 is delivered before the event message e_1 to the peer p_j. In addition, although the peers p_i and p_j receive common event messages e_1 and e_2, the event messages e_1 and e_2 are delivered to the peers in different orders.

In this section, we propose a TOBSCO (TOBS with Causally Ordering delivery) protocol [47] to causally deliver every event message. Here, we assume the underlying network provides every pair of peers with reliable communication service, i.e. every peer receives event messages in the sending order with neither message loss nor duplication. However, event messages published by different peers may be received by different target peers in different orders. Hence, every pair of event messages have to be causally delivered to every common target peer. In the paper [49], the CO (Causally Ordering broadcast) protocol is proposed where the vectors of sequence numbers of messages are used. We also assume each event message is broadcast to every peer. Then, only an event message whose publication includes some subscription topic is delivered to a peer. In this thesis, $e_i.SEQ$ denotes the sequence number of an event message e_i published by a peer p_i. If a peer p_i publishes an event message e_2 just after the event message

e_1, $e_2.SEQ = e_1.SEQ + 1$. Let $e_i.ACK_j$ be a sequence number of an event message e_i which the peer p_i expects to receive next from a peer p_j (j = 1, . . ., pn). Each peer p_i also obtains a sequence number $p_i.SEQ$ which the peer p_i expects to publish next. $p_i.SEQ$ is initially 0. Let $p_i.REQ_j$ be a sequence number of an event message which the peer p_i expects to receive next from a peer p_j (j = 1, . . ., pn). A peer p_i manipulates a $pn \times pn$ matrix $p_i.AL$. Each element $p_i.AL_{k,j}$ shows a sequence number of an event message which the peer p_i knows that a peer p_j expects to receive next from a peer p_k (j, k = 1, . . ., pn). $min(p_i.AL_k)$ is a minimum one of $p_i.AL_{k,1}$, . . ., $p_i.AL_{k,pn}$. Each element $p_i.AL_{k,j}$ is initially 1. The sequence numbers are manipulated by each peer p_i as shown in Algorithm 4.

Algorithm 4: Manipulation of sequence numbers

```
/*A peer $p_i$ publishes an event message $e_i$*/
$e_i.SEQ = p_i.SEQ$;
$p_i.SEQ = p_i.SEQ + 1$;
$e_i.ACK_j = p_i.REQ_j$ ($j$ = 1, . . ., $pn$);
/*$p_i$ receives an event message $e_j$ from a peer $p_j$*/
$p_i.REQ_j = e_j.SEQ + 1$;
$p_i.AL_{k,j} = e_j.ACK_k$ ($k$ = 1, . . ., $pn$);
```

In order to guarantee that event messages published by a peer p_j are delivered to a peer p_i in the publication order of the peer p_j, only an event message e_j which holds the condition "$e_j.SEQ = p_i.REQ_j$" is delivered to the peer p_i. Every event message arriving at the peer p_i in the underlying network is kept in the buffer RBF_i of the peer p_i. Here, it is guaranteed that every pair of event messages published by each peer are stored in the buffer RBF_i in the publishing order. Then, it is checked whether or not the event message e_j satisfies the condition "$e_j.SEQ < min(p_i.AL_j)$ $(= min\{p_i.AL_{j,1}, \dots, p_i.AL_{j,pn}\})$". If the condition holds, the peer p_i correctly receives the event message e_j. After that, only the event message e_j whose publication $e_j.P$ includes some common topic with the subscription $p_i.S$ is moved to a second buffer SBF_i of the peer p_i. Every event message in the second buffer SBF_i is reordered in the causally precedent order.

The peer p_i dequeues a top event message e from the second buffer SBF_i and the event message e is delivered to the peer p_i in the causally precedent order. Figure 5.3 shows how to deliver event messages to a peer.

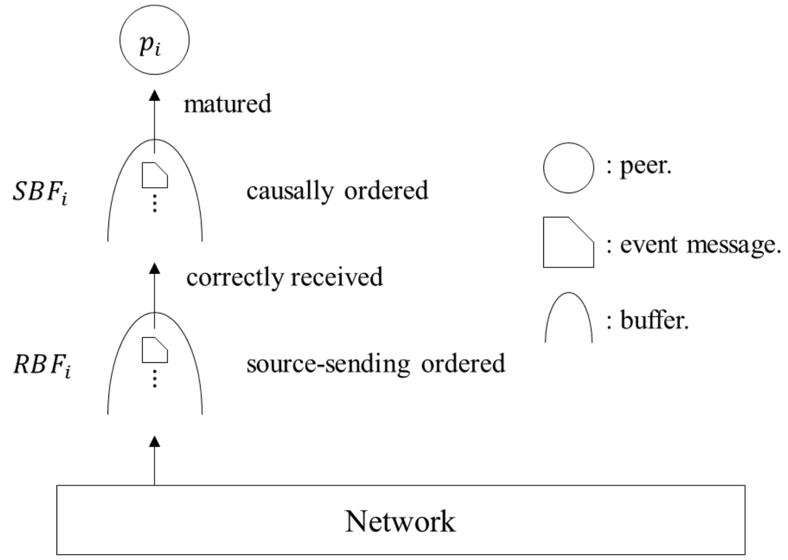

Figure 5.3: Delivery of event messages in the TOBSCO protocol.

Example 3. Suppose there are three peers p_1, p_2, and p_3 as shown in Figure 5.4. We also suppose the peers p_1, p_2, and p_3 have publications and subscriptions as $p_1.P = p_1.S = \{w, x\}$, $p_2.P = p_2.S = \{w, x, y, z\}$, and $p_3.P = p_3.S = \{w, x, y\}$, respectively.

First, the peers p_1, p_2, and p_3 create objects o_1^1, o_2^2, and o_3^3 where $o_1^1.T = \{w\}$, $o_2^2.T = \{x, y\}$ and $o_3^3.T = \{y\}$ and store them in their storages $p_1.D$, $p_2.D$, and $p_3.D$, respectively. Next, the peer p_1 publishes an event message e_1 where $e_1.O = \{o_1^1\}$, $e_1.T = o_1^1.T = \{w\}$, and $e_1.P = \{w\}$. Then, the peer p_1 updates data in the object o_1^1 and the topic set $o_1^1.T$ ($= \{w\}$) is changed with $\{x\}$. The peer p_1 publishes an update event message ue_1 to make another peer p_i synchronize the replica o_1^i with the object o_1^1. Here, the publication $ue_1.P$ is same as the variable $o_1^1.T$ of the unupdated object o_1^1, i.e. $ue_1.P = \{w\}$. After that, the peer p_2 publishes an event message e_2 where $e_2.O = \{o_2^2\}$, $e_2.T = o_2^2.T = \{x, y\}$, and $e_2.P = \{x, y\}$. Next, the peer p_2 updates data in the object o_2^2 and the topic set $o_2^2.T$ $(= \{x, y\})$ is changed with $\{w, x\}$. The peer p_2 publishes an update event message ue_2 to make another peer p_i synchronize the replica o_2^i with the object o_2^2. Then, the peer p_3 publishes an event message e_3 where $e_3.O = \{o_3^3\}$, $e_3.T = o_3^3.T = \{y\}$, and $e_3.P = \{y\}$. Finally, the peer p_1 publishes an event message e_4 where $e_4.O = \{o_1^1\}$, $e_1.T = o_1^1.T = \{x\}$, and $e_1.P = \{x\}$.

The variable REQ of each peer p_i is updated as shown in Figure 5.5. Table 5.1 shows the parameters of each event message. The peer p_3 receives event messages e_1, e_2, ue_1, and ue_2, until the peer p_3 receives the event message e_3. After the peer p_3 receives the event message e_3, the peer p_3 obtains the following matrix $p_3.AL$:

$$p_3.AL = \begin{bmatrix} 2 & 3 & 3 \\ 1 & 2 & 3 \\ 1 & 1 & 1 \end{bmatrix} \tag{5.1}$$

Here, $min(p_3.AL_1)$ is $min\{p_3.AL_{1,1}, p_k.AL_{1,2}, p_k.AL_{1,3}\} = 2$. The condition "$e_1.SEQ$ $(= 1) < min(p_3.AL_1)$" holds. Hence, only the event message e_1 is moved to the second buffer SBF_3 of the peer p_3 and then the event message e_1 is delivered to the peer p_3.

After the peer p_3 receives the event message e_4, the peer p_3 obtains the following matrix $p_3.AL$:

$$p_3.AL = \begin{bmatrix} 3 & 3 & 3 \\ 3 & 2 & 3 \\ 2 & 1 & 1 \end{bmatrix} \tag{5.2}$$

The conditions "$e_2.SEQ$ $(= 1) < min(p_3.AL_2)$ $(= 2)$" and "$ue_1.SEQ$ $(= 2)$ $< min(p_3.AL_1)$ $(= 3)$" hold and the event messages e_2 and ue_1 are moved to the second buffer SBF_3 of the peer p_3. Here, the peer p_3 recognizes $ue_1 \rightarrow_c e_2$ because $ue_1.SEQ$ $(= 2) < e_2.ACK_1$ $(= 3)$. Hence, ue_1 is delivered to the peer p_3 before e_2 differently from the TOBS protocol although the peer p_3 receives ue_1 after e_2. In the peer p_1, the event message e_1 and a pair of event messages

ue_1 and e_2 are delivered after the peer p_1 receives the event message e_3 and the event message e_4, respectively. Here, the illegal information flow relation "$e_2 \mapsto p_1$" holds. Since $o_2^2 \overset{o}{\mapsto} p_1$, the object o_2^2 is not delivered to the peer p_1 to prevent illegal information flow. In the peer p_2, the event message e_1 and a pair of event messages ue_1 and e_2 are delivered after the peer p_2 receives the event message e_3 and the event message e_4, respectively.

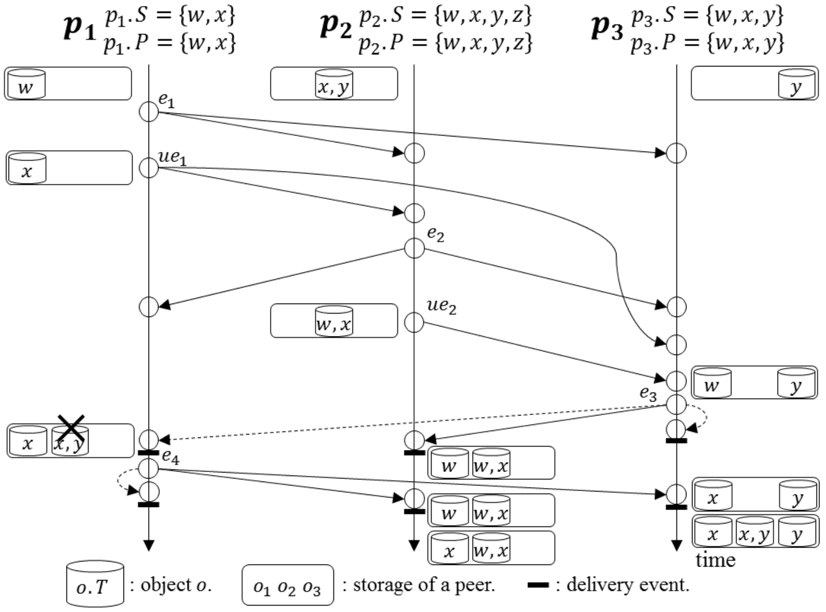

Figure 5.4: TOBSCO protocol.

In Example 3, the peer p_2 publishes an event message e_2 after receiving the update event message ue_1. Here, the causally precedent relation $ue_1 \rightarrow_c e_2$ holds. However, the event message e_2 arrives at the peer p_3 before the update event message ue_1. Therefore, if the TOBS protocol is performed in Example 3, the event message e_2 is delivered before the update event message ue_1 although $ue_1 \rightarrow_c e_2$ holds in the peer p_3. Here, the event message e_2 is *premature*, i.e. e_2 is delivered to the peer p_3 although the event message e_2 is not matured at the peer p_3.

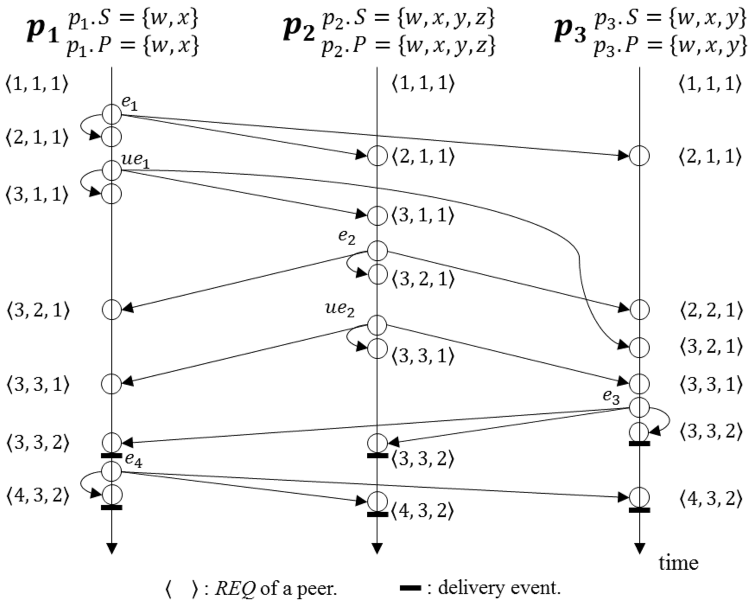

Figure 5.5: CO protocol.

Table 5.1: Parameters of every event message.

Name	O	P	T	SEQ	ACK
e_1	o_1^1	w	w	1	$\langle 1, 1, 1 \rangle$
ue_1	o_1^1	w	x	2	$\langle 2, 1, 1 \rangle$
e_2	o_2^2	x, y	x, y	1	$\langle 3, 1, 1 \rangle$
ue_2	o_2^2	x, y	w, x	2	$\langle 3, 2, 1 \rangle$
e_3	o_3^3	y	y	1	$\langle 3, 3, 1 \rangle$
e_4	o_1^1	x	x	3	$\langle 3, 3, 2 \rangle$

5.4 ETOBSCO (Efficient TOBSCO) protocol

In this section, we propose an ETOBSCO (Efficient TOBSCO) protocol [48] in order to deliver only legal objects to target peers. In the P2PPSO system, a peer p_i is allowed to publish and subscribe event messages on a topic t only if the topic t is included in the publication $p_i.P$ and the subscription $p_i.S$ of the peer p_i, respectively. If an event message e_i published by a source peer p_i flows to a target peer p_j ($e_i \rightarrow p_j$), the event message e_i is delivered to the peer p_j. Here, if at least one illegal object o_i in the event message e_i is delivered to the peer p_j ($o_i \overset{o}{\mapsto} p_j$), information of the source peer p_i illegally flows into the peer p_j. Illegal objects have to be not delivered to a target peer even if an event message is delivered. Here, only legal objects in event messages are delivered to target peers and are stored in the storages of the target peers.

Suppose a peer p_i publishes an event message e_i with a pair of objects o_1^i and o_2^i. Suppose, $o_1^i.T = \{x,\, y\}$ and $o_2^i.T = \{x,\, z\}$ where x, y, and z are topics. First, suppose a subscription $p_j.S$ of a target peer p_j of the event message e_i is a set $\{w,\, x,\, y\}$ of topics. Here, $o_1^i \overset{o}{\Rightarrow} p_j$ and $o_2^i \overset{o}{\mapsto} p_j$ since $o_1^i.T \subseteq p_j.S$ and $o_2^i.T \nsubseteq p_j.S$. Here, a pair of the objects o_1^i and o_2^i are legal and illegal at the peer p_j, respectively.

As we make the assumption in the preceding section, every object o_i is assumed to be updated by only the creator peer p_i of the object o_i. Once an owner peer p_i updates an object o_i, the peer p_i is required to publish an *update* event message ue_i to every peer p_j holding a replica o_i^j so that the replica o_i^j is mutually consistent with the object o_i. On receipt of an update event message of an object o_i, a peer p_j updates a replica o_i^j. Here, the topics of the object o_i may not be changed even if the state of the object o_i is updated. If the topics of the object o_i are not changed, an update event message to inform the update of the object o_i is *meaningless* to prevent illegal objects from being delivered since only topics of objects are checked. A replica o_i^j is *weakly consistent* with an object o_i iff the topics $o_i^j.T$ of the replica o_i^j is the same as the topics $o_i.T$ of the object o_i. Even if the replica o_i^j is weakly consistent with an object o_i, data in the replica o_i^j may not be the same as the object o_i. Thus, in order to reduce the overhead to update every replica, each replica is kept weakly consistent with an object.

Definition 11. A peer p_i *alters* an object o_i iff the peer p_i updates the state of the object o_i but the topics in $o_i.T$ are not changed.

Definition 12. An update event message ue_i of an object o_i published by a peer p_i is *meaningless* iff the peer p_i alters the object o_i.

If the state of the object o_i is changed but the topics in $o_i.T$ are not changed, it is meaningless to distribute an update event message ue_i. In the ETOBSCO protocol, every peer avoids publishing meaningless update event messages of an object o_i to reduce the number of event messages exchanged even if the peer p_i alters an object o_i. Here, holder peers of replicas of the object o_i cannot recognize that the object o_i is altered by a peer p_i unless the object o_i is delivered to the holder peers. In order to make every target peer know that the object o_i is altered, the peer p_i publishes an update event message ue_i carrying the altered object o_i if the object o which has at least one common topic with the object o_i is delivered to the peer p_i.

In the ETOBSCO protocol, each peer behaves as shown in Algorithms 1, 2, and 3 to prevent illegal information flow. In addition, every pair of event messages are causally delivered to every common target peer even if the event messages are published by different peers. In order to causally deliver event messages, the sequence numbers of event messages which we discussed in the section 5.3 are used. The sequence numbers of event messages are manipulated as shown in Algorithm 4.

Example 4. Suppose three peers p_1, p_2, and p_3 are cooperating by manipulating objects and publishing and receiving event messages as shown in Figure 5.6. The peers p_1, p_2, and p_3 have publications and subscriptions as $p_1.P = p_1.S = \{w, x, z\}$, $p_2.P = p_2.S = \{w, x, y, z\}$, and $p_3.P = p_3.S = \{w, x, y\}$, respectively. The peers p_1, p_2, and p_3 create objects o_1, o_2, and o_3 where $o_1.T = \{w\}$, $o_2.T = \{x\}$, and $o_3.T = \{x, y\}$, respectively. First, the peer p_1 publishes an event message e_1 with the object o_1. Here, $e_1.P = e_1.T = o_1.T = \{w\}$. Then, the peer p_1 alters the object o_1. The peer p_1 does not publish the update event message ue_2 because the update event message ue_2 is meaningless.

Then, the peers p_1, p_2, and p_3 publish event messages e_3, e_4, and e_5 where $e_3.O = \{o_1\}$, $e_4.O = \{o_2\}$, and $e_5.O = \{o_3\}$, respectively. In the peer p_2, the event message e_1 is delivered after the event message e_5 arrives at the peer p_2 because $e_1.P \cap p_2.S \neq \phi$. Here, the object o_1 carried by the event message e_1 is delivered to the peer p_2 and a replica o_1^2 is stored in the storage $p_2.D$ of the peer p_2. Similarly, a replica o_1^3 is stored in the peer p_3.

Finally, the peers p_1, p_2, and p_3 publish event messages e_6, e_7, and e_8 where $e_6.O = \{o_1\}$, $e_7.O = \{o_2\}$, and $e_8.O = \{o_3\}$, respectively. In the peer p_1, the pair of event messages e_4 and e_5 are delivered after the event message e_8 arrives at the peer p_1. Here, $e_4 \Rightarrow p_1$ holds and the object o_2 is delivered to the peer p_1. Thus, a replica o_2^1 is stored in the storage $p_1.D$. On the other hand, the illegal information flow relation $e_5 \mapsto p_1$ holds because the object o_3 such that $o_3 \overset{o}{\mapsto} p_1$ is carried by the event message e_5. Here, the illegal object o_3 is not delivered to the peer p_1 to prevent illegal information flow.

In the peer p_2, the replica o_1^2 of the object o_1 in the peer p_2 is synchronized with the altered object o_1 after the event message e_8 arrives at the peer p_2. On the other hand, the replica o_1^3 of the object o_1 in the peer p_3 is synchronized with the altered object o_1 after the event message e_7 arrives at the peer p_3.

Suppose the peers publish event messages and update objects as shown in Example 4. In the TOBSCO protocol, the peer p_1 publishes an update event message ue_2 after the peer p_1 alters the object o_1. Hence, more number of event messages are exchanged among peers in the TOBSCO protocol compared with the ETOBSCO protocol. However, the replica o_1^2 of the object o_1 in the peer p_2 is synchronized with the altered object o_1 after the event message e_5 arrives at the peer p_2. This means, it takes shorter time to synchronize every replica with the altered object in the TOBSCO protocol than the ETOBSCO protocol.

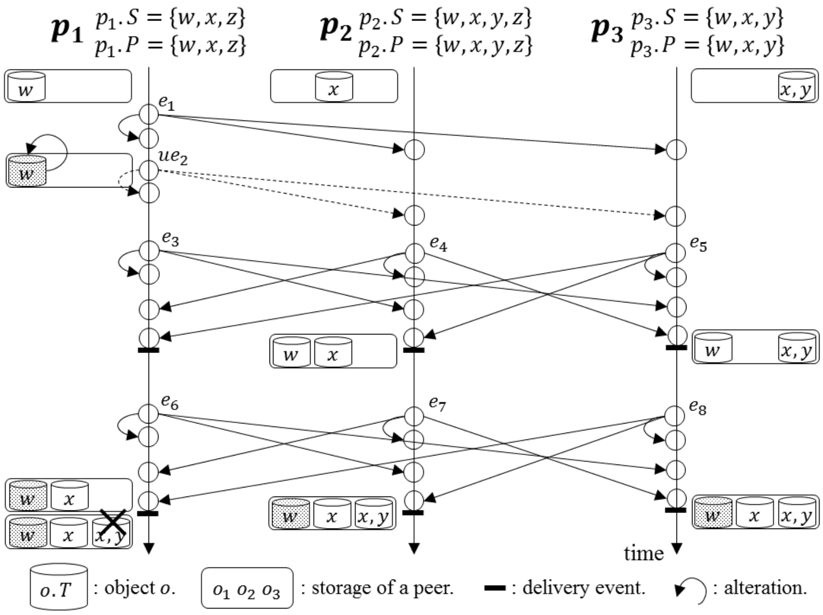

Figure 5.6: ETOBSCO protocol.

Chapter 6

Evaluation

6.1 Properties of protocols

First, the following property holds for the TOBS, TOBSCO, and ETOBSCO protocols:

Property 1. In the TOBS, TOBSCO, and ETOBSCO protocols, no illegal information flow occurs.

Proof 1. Suppose an object o such that $o.T \not\subseteq p_j.S$ is carried by an event message e_i to a target peer p_j. According to Definition 8, the object o is illegal. In the TOBS, TOBSCO, and ETOBSCO protocols, every illegal object o is not delivered to peers as shown in Algorithm 2. Therefore, no illegal information flow occurs in every protocol. $\square$

This property means that every illegal object is not delivered to any target peer. On the other hand, some legal object might be not delivered. The protocol stack of the TOBS, TOBSCO, and ETOBSCO protocols is shown in Figure 5.1.

In the TOBS protocol, the function to causally deliver every pair of event messages does not exist. Hence, the TOBS protocol is supported on the CO protocol by which event messages are causally delivered to every peer. On the other hand, the causally ordering delivery function is also supported by the TOBSCO protocol. Hence, the causally ordering delivery of event messages is guaranteed by using only the TOBSCO protocol. In the ETOBSCO protocol, the function to reduce the network overhead is supported in addition to the causally ordering delivery function. Table 6.1 summarizes the properties of the TOBS, TOBSCO, and ETOBSCO protocols.

Table 6.1: Summary of the TOBS, TOBSCO, and ETOBSCO protocols.

Protocol	**Prevention of illegal information flow**	**Causally ordering delivery**	**Alteration check**
TOBS	○	×	×
TOBSCO	○	○	×
ETOBSCO	○	○	○

6.2 TOBS protocol

6.2.1 Environment

In this subsection, we evaluate the TOBS protocol on a topic set $T = \{t_1, \ldots, t_{tn}\}$ ($tn \geq 1$) and a peer set $P = \{p_1, \ldots, p_{pn}\}$ ($pn \geq 1$) in terms of the numbers of event messages and objects which are not delivered to peers. Suppose a peer p_i receives an event message e and the event message e carries an object on a topic t. Here, if the peer p_i is not allowed to subscribe the topic t, the object is not delivered to the peer p_i in the TOBS protocol. We assume each event message can be reliably broadcast to every target peer in a system. An event message e is delivered to each peer p_i only if the publication $e.P$ and the subscription $p_i.S$ include at least one common topic.

In the evaluation, access rights are randomly granted to each peer p_i, i.e. topics in the publication $p_i.P$ and the subscription $p_i.S$ of each peer p_i are randomly taken in the topic set T. Let stn_i be the number of topics in the subscription $p_i.S$. The number stn_i is randomly selected out of numbers 1, …, $mstn$. Here, $mstn$

is the maximum number of topics which can be included in the subscription $p_i.S$ and publication $p_i.P$ of each peer p_i. Let ptn_i be the number of topics in the publication $p_i.P$. The publication $p_i.P$ of each peer p_i includes at least one topic. Topics in the publication $p_i.P$ are randomly selected so that the publication $p_i.P$ is a subset of the subscription $p_i.S$, i.e. $1 \leq ptn_i \leq stn_i$ and $p_i.P \subseteq p_i.S$. After publication and subscription rights are granted to a peer p_i, the peer p_i creates one object o_i. The topic set $o_i.T$ of an object o_i includes at least one topic. Topics in the set $o_i.T$ are randomly taken so that $o_i.T$ is a subset of the publication $p_i.P$, i.e. $1 \leq |o_i.T|$ and $o_i.T \subseteq p_i.P$.

Let cp be a creation probability. This means, each peer p_i creates an object with probability cp at each time.

Let up be an update probability. This means, each peer updates its own object with probability up at each time. We consider a pair of update operations on an object, *full* and *partial* update operations. If a peer p_i issues a full update operation to an object o^i_i, the whole data of the object o^i_i is fully overwritten. This means, the object o^i_i is deleted and created. Since every data is changed in the object o^i_i, topics on the object o^i_i are totally changed with new ones. Hence, the variable $o^i_i.T$ gets empty, i.e. $o^i_i.T = \phi$. Then the variable $o^i_i.T$ randomly includes topics so that $o^i_i.T$ is a subset of the subscription $p_i.S$, i.e. $o^i_i.T \subseteq p_i.S$. On the other hand, if the peer p_i issues a partial update operation to an object o^i_i, only some data of the object o^i_i is overwritten. Topics on the object o^i_i are considered to be not deleted and just new topics are given to the object o^i_i. Hence, some topics which the peer p_i is allowed to subscribe are added to the variable $o^i_i.T$.

Table 6.2 shows the parameters pn, tn, n, $mstn$, stn_i, and ptn_i used in the simulation. In the evaluation, we consider fifty peers $p_1, \ldots, p_{50}$ ($pn = 50$) and one hundred topics $t_1, \ldots, t_{100}$ ($tn = 100$). We evaluate the TOBS protocol in the following procedure:

[Simulation procedure]

1. One peer p_i is randomly selected in the peer set P and the peer p_i randomly includes some object or replica o^i such that $o^i.T \subseteq p_i.P$ in an event message

Table 6.2: Parameters used for simulation.

Parameters	**Values**
Number pn of peers in the system	50
Number tn of topics in the system	100
Number n of publication events	0, 100, 200, 300, 400, 500
Maximum number $mstn$ of topics in a subscription	40
Number stn_i of topics in a subscription of each peer p_i	1, . . ., $mstn$
Number ptn_i of topics in a publication of each peer p_i	1, . . ., stn_i
Creation probability cp	0.01
Update probability up	0.02

e_i. Publication $e_i.P$ is decided so that $e_i.P$ is same as the topic set $e_i.T$ of the event message e_i. The peer p_i publishes the event message e_i.

2. Each peer p_i creates an object o where $o.T$ is a subset of $p_i.P$ with probability cp.

3. Each peer p_i fully or partially updates data in the object o_i^i where p_i is the creator peer with probability up. If the peer p_i updates the object o_i^i, the peer p_i publishes an update event message ue_i to make another peer p_j synchronize the replica o_i^j with the object o_i^i, i.e. $o_i^j.T$ is updated as $o_i^i.T$.

4. An event message e_j is delivered to a target peer p_i if $p_i.S \cap e_j.P \neq \phi$, i.e. $e_j \rightarrow p_i$. Only the object o^j such that $o^j.T \subseteq p_i.S$ carried by the event message e_j is delivered to the peer p_i.

Let n be the number of publication events to occur in the simulation ($0 \leq n \leq 500$). For each n, two hundred different peer sets P_1, . . ., P_{200} of fifty peers p_1, . . ., p_{50} are randomly generated. For each set P_k, n publication events randomly occur two hundred times. After that, we calculate the average numbers of event messages and objects in the TOBS protocol.

6.2.2 Evaluation results

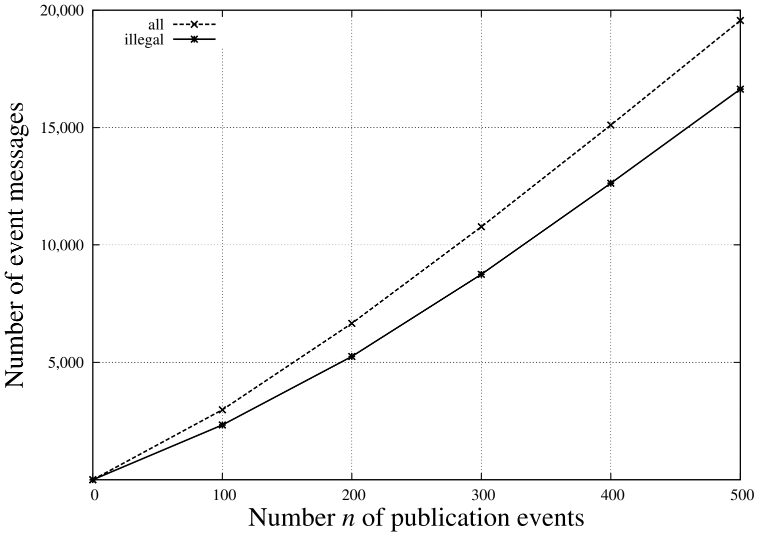

Figure 6.1: Number of event messages in the TOBS protocol.

Figure 6.1 shows the numbers of published event messages and illegal event messages in the TOBS protocol. The dotted line with crosses ($\times$) shows the total number of event messages published by the fifty ($pn = 50$) peers. On the other hand, the straight line with stars ($*$) shows the total number of illegal event messages published by the fifty ($pn = 50$) peers. The number of illegal event messages monotonically increases as the number n of publication events increases. For example, about 2,330 event messages are illegal for one hundred publication events ($n = 100$) and about 16,640 event messages are illegal for five hundred publication events ($n = 500$) in the TOBS protocol. That is, about 80% of the total number of event messages published by the peers are illegal. This means, 20% of event messages are legal.

Figure 6.2 shows the number of objects and replicas in the TOBS protocol for number n of publication events. Each time a peer receives an event message, replicas of objects carried by the event message are stored in the storage. In addition, a

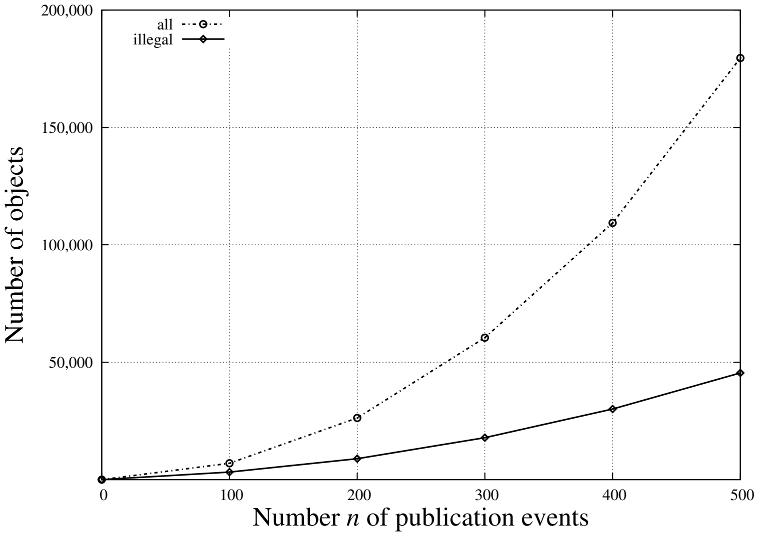

Figure 6.2: Number of objects in the TOBS protocol.

peer creates a new object. The dotted line with circles (○) shows the total number of objects and replicas held by every target peer. The number of objects and replicas carried exponentially increases for the number n of publication events. On the other hand, the straight line with diamonds (◇) shows the total number of illegal objects which are carried by event messages but are not delivered. The number of illegal objects which are not delivered about 30% slowly increases for the total number of objects and replicas as the number n of publication events increases. For example, about 3,210 objects are illegal for one hundred publication events ($n = 100$) and about 45,430 objects are for five hundred publication events ($n = 500$) in the TOBS protocol.

6.3 TOBSCO protocol

6.3.1 Environment

In this subsection, we evaluate the TOBSCO protocol in terms of the number of premature event messages and the average delivery time of event messages. In the TOBSCO protocol, every illegal object is not delivered to a target peer. In addition, every event message is causally delivered to every target peer.

Table 6.3: Parameters used for simulation.

Parameters	**Values**
Number pn of peers in the system	10
Number tn of topics in the system	50
Number n of publication events	0, 100, 200, 300, 400, 500
Maximum number $mstn$ of topics in a subscription	20
Number stn_i of topics in a subscription of each peer p_i	1, . . ., $mstn$
Number ptn_i of topics in a publication of each peer p_i	1, . . ., stn_i
Creation probability cp	0.01
Update probability up	0.02

Table 6.3 shows the parameters used in the simulation. In the evaluation, we consider ten peers p_1, . . ., p_{10} (pn = 10) and fifty topics t_1, . . ., t_{50} (tn = 50).

The simulation procedure is the same as that of the TOBS protocol described in subsection 6.2.1. For each n, one hundred different peer sets P_1, . . ., P_{100} of ten peers p_1, . . ., p_{10} are randomly generated. For each set P_k, n publication events randomly occur one hundred times. After that, we calculate the average number of premature event messages and delivery time in the TOBSCO protocol.

In the simulation, we make the following assumptions:

- No event message is lost in a system.

- Every peer does not publish any event message while the peer recognizes at least one event message exists which is already published but is not received yet by the peer. For instance, the peer p_3 does not publish any event message after receiving the event message e_2 and before receiving the update event message ue_1 in Figure 5.4.

- The simulation steps 1 to 4 are in one time unit [tu]. Let mdt be the maximum delay time. This means, delay time of each event message for each peer is randomly decided out of 1 to mdt [tu]. However, the delay time of the event message e_i for the peer p_i is 0 [tu].

6.3.2 Evaluation results

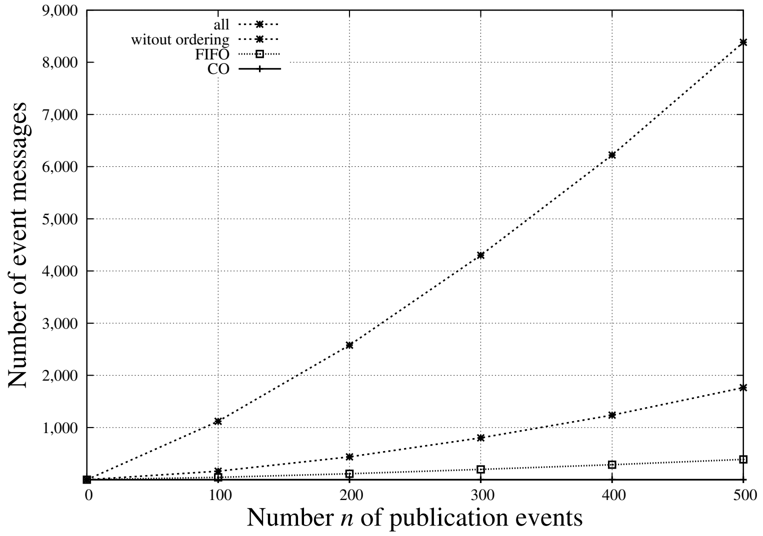

Figure 6.3: Number of premature event messages.

Figure 6.3 shows the numbers of premature event messages for number n of publication events where the maximum delay time mdt is ten [tu] in the TOBSCO protocol. The dotted line with the label "all" shows the total number of event messages delivered. The dotted line with the label "without ordering" shows the number of premature event messages when every event message is delivered to a peer as soon as the event message arrives at the peer. The dotted line with the label "FIFO" shows the number of premature event messages when every event message is delivered with FIFO ordering delivery in the reliable one-to-one communication. This means, the event messages published by a common peer are delivered in sending order while event messages published by different peers may be delivered in different orders. The straight line with the label "CO" shows the number of premature event messages when every event message is causally delivered. In the causally ordering delivery, no premature event message exists for any number of publication events. On the other hand, if event messages are not causally delivered, the number of premature event messages increases as the number n of publication events increases.

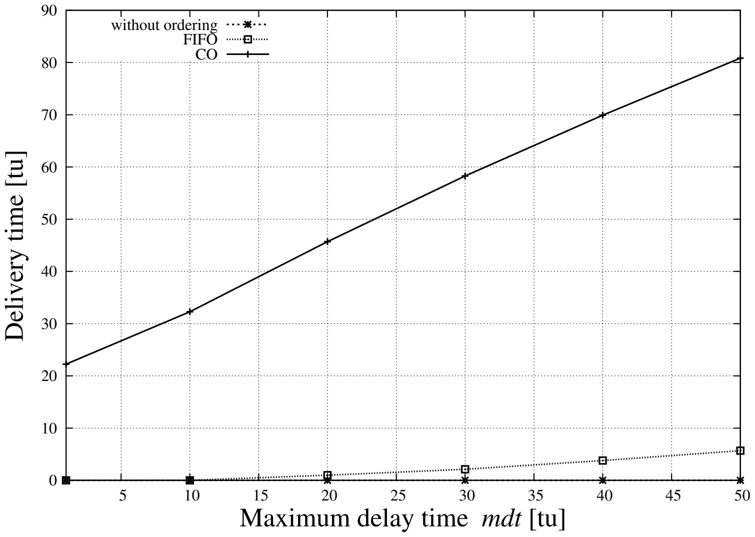

Figure 6.4: Delivery time.

Figure 6.4 shows the average delivery time of event messages delivered for maximum delay time mdt where n is one hundred in the TOBSCO protocol. Delivery time means how many time units it takes until an event message is delivered to a peer from time the event message arrives at the peer. If event messages are not sorted in the buffer before the event messages are delivered, delivery time is zero, i.e. every event message is delivered to a peer as soon as the event message arrives at the peer. On the other hand, delivery time becomes longer as the maximum delay time mdt becomes longer if event messages are sorted in the buffer before the event messages are delivered. The delivery time to support the causally ordering delivery is longer than that with just FIFO ordering delivery of the reliable one-to-one communication.

6.4 ETOBSCO protocol

6.4.1 Environment

In this subsection, we evaluate the ETOBSCO protocol in terms of the number of event messages delivered and the average update delay time of every object altered. In the ETOBSCO protocol, every illegal object is not delivered to peers to prevent illegal information flow. Every pair of event messages are causally delivered to peers. The meaningless update event messages are not published to reduce the number of event messages exchanged among peers.

Let ap be an alteration probability of an update operation. This means, an update operation is alteration with probability ap in full or partial update operation, respectively. Here, each peer p_i alters an object o_i with probability $up \cdot ap$.

Table 6.4 shows the parameters used in the simulation. In the evaluation, we consider ten peers $p_1, \ldots, p_{10}$ ($pn = 10$) and fifty topics $t_1, \ldots, t_{50}$ ($tn = 50$). The simulation procedure is the same as those of the TOBS and TOBSCO protocols described in subsection 6.2.1.

For each n, one hundred different peer sets $P_1, \ldots, P_{100}$ of ten peers $p_1, \ldots, p_{10}$ are randomly generated. For each set P_k, n publication events randomly occur one hundred times. After that, we calculate the average number of event messages delivered and update delay time of altered objects in the ETOBSCO protocol.

Table 6.4: Parameters used for simulation.

Parameters	**Values**
Number pn of peers in the system	10
Number tn of topics in the system	50
Number n of publication events	200, 400, 600, 800, 1,000
Maximum number $mstn$ of topics in a subscription	20
Number stn_i of topics in a subscription of each peer p_i	1, . . ., $mstn$
Number ptn_i of topics in a publication of each peer p_i	1, . . ., stn_i
Maximum delay time mdt of an event message e_i for a peer p_j $(i \neq j)$ [tu]	10
Creation probability cp	0.01
Update probability up	0.02
Alteration probability ap of an operation	0.5

6.4.2 Evaluation results

Figure 6.5 shows the numbers of event messages delivered in the TOBSCO and ETOBSCO protocols. The dotted line with the label "em" shows the total number of event messages delivered which are not update event messages. Here, the difference between em-line and ETOBSCO-line shows how many update event messages are published in the ETOBSCO protocol. The number of event messages delivered in the ETOBSCO protocol is fewer than the TOBSCO protocol. For example, about 20,300 event messages are delivered in the TOBSCO protocol but about 13,800 event messages are in the ETOBSCO protocol for one thousand publication events (n = 1,000).

Figure 6.6 shows the average update delay time of altered objects in the TOBSCO and ETOBSCO protocols. Update delay time means how many time units it takes until an event message carrying the altered object is delivered to a peer from time the object is altered. The update delay time of altered objects in the ETOBSCO protocol is longer than the TOBSCO protocol. For example, update delay

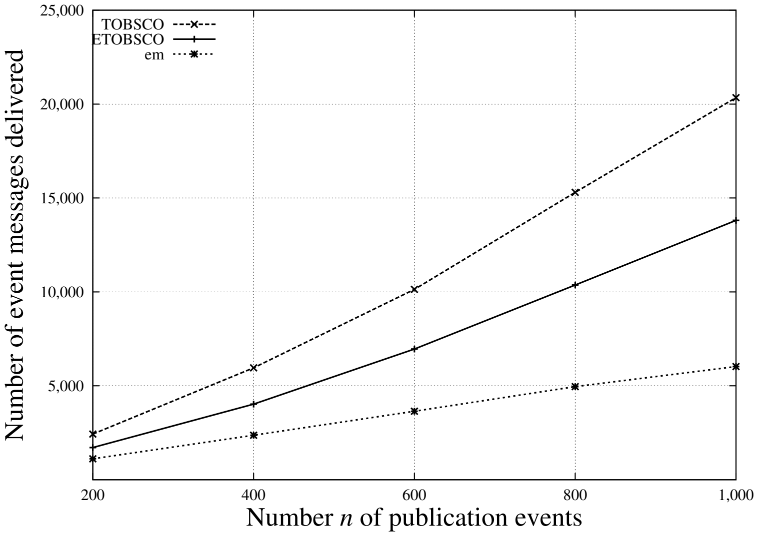

Figure 6.5: Number of event messages delivered.

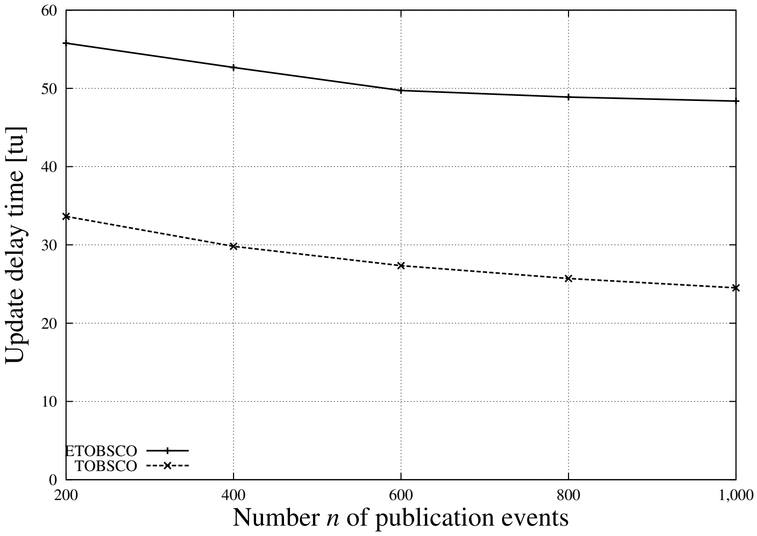

Figure 6.6: Update delay time.

time is about 25 [tu] in the TOBSCO protocol but about 48 [tu] in the ETOBSCO protocol for one thousand publication events ($n = 1{,}000$).

Chapter 7

Conclusions and Future Studies

7.1 Conclusions

The PS (Publish/Subscribe) model is a new event-driven, content-based model of a distributed system. In this thesis, we proposed the P2PPSO (P2P type of topic-based PS with Object concept) model where each peer can publish and receive event messages in a distributed manner. In addition to increasing performance, reliability, and availability, distributed systems have to be secure in presence of malicious peers. Here, we have to prevent illegal information flow to occurring among peers and objects in the access control models like the RBAC model. In the P2PPSO model where information exchanged among peers are considered as objects which are characterized by topics. Here, each peer exchanges objects with other peers by publishing and subscribing event messages with no centralized coordinator. Peers manipulate objects and event messages carry objects to target peers. Illegal information flow occurs if the objects related with topics which the target peers are not allowed to subscribe are delivered to the target peers by receiving event messages with the objects.

Next, we proposed a TBAC (Topic-Based Access Control) model as an access control model of the PS model. Here, an access right is a pair $\langle t, op \rangle$ of a topic t and a publish or subscribe operation op. A peer is allowed to publish an event message with publication topics and subscribe interesting topics only if the publication and subscription access rights are granted to the peer, respectively. Event messages carry objects to a target peer. If the information related with the topics which the target peer is not allowed to subscribe is carried by event messages to the peer, illegal information flow occurs. Based on the TBAC model, we defined legal and illegal information flows among peers and event messages. We also defined illegal objects whose topics are not allowed to be subscribed by the target peer. If at least one illegal object is carried by an event message to a target peer, information of the object illegally flows to the peer.

In order to prevent illegal information flow from occurring in the P2PPSO system, first, we proposed the TOBS protocol. In the TOBS protocol, even if an event message is received by a target peer, illegal objects in the event message are not delivered to the target peer.

In the TOBS protocol, every event message is assumed to be causally delivered to every common target peer in the underlying network. Hence, secondly, we proposed the TOBSCO (TOBS with Causally Ordering delivery) protocol where the function to causally deliver every pair of event messages is added to the TOBS protocol. Here, every pair of event messages received by using topics are causally delivered to every common target peer by using the vector of sequence numbers.

In the TOBS and TOBSCO protocols, illegal objects are not delivered to the target peers. However, update event messages of an object are published to update every replica of the object each time the object is updated. Hence, thirdly, we proposed the ETOBSCO (Efficient TOBSCO) protocol where an update event message of the object is not published if the topics of the object are not changed, i.e. the object is altered. Every peer is forced to avoid publishing meaningless event messages to reduce the network overhead. In the TOBS, TOBSCO, and ETOBSCO protocols, every illegal event message is not delivered to any target peer. However, some legal event message might be not delivered.

In the evaluation, first, we show how many event messages and objects which are not delivered to target peers in the TOBS protocol. Next, we show every pair of event messages are causally delivered but it takes longer to deliver event messages in the TOBSCO protocol than the TOBS protocol. Finally, we show the fewer number of event messages are delivered while it takes longer to update replicas of altered objects in the ETOBSCO protocol than the TOBSCO protocol.

The protocols which we newly discussed and proposed in this thesis are theoretical foundations to design, implement, and evaluate secure distributed systems.

7.2 Future studies

In this thesis, we evaluated the TOBS, TOBSCO, and ETOBSCO protocols in the simulation. In order to make the performance clearer, we are now designing a test bed to evaluate the protocols in terms of the number of event messages delivered, update delay time, and energy consumption of peers and systems.

Information systems are composed of various types of nodes like sensors and actuators in addition to servers and clients as discussed in the IoT (Internet of Things). In the IoT, a CapBAC (Capability-Based Access Control) model [50, 51] is proposed to make devices secure. Here, subjects like users and applications send access requests to devices and data are exchanged among subjects and devices. Hence, a subject may get data which the subject is not allowed to get, i.e. illegal information flow occurs. We would like to define information flow relations among subjects and devices and then propose a protocol to prevent illegal information flow based on the relations in the IoT.

Bibliography

[1] M. V. Steen, A. S. Tanenbaum, Distributed Systems: Pearson New International Edition: Principles and Paradigms, 2nd Edition, Pearson Education Limited, London, England, UK, 2013.

[2] D. E. R. Denning, Cryptography and Data Security, Addison Wesley, Boston, MA, USA, 1982.

[3] D. Hanes, G. Salgueiro, P. Grossetete, R. Barton, J. Henry, IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things, Cisco Press, Indianapolis, IN, USA, 2018.

[4] R. L. Grossman, The case for cloud computing, IT Professional 11 (2) (2010) 23–27.

[5] L. Barolli, F. Xhafa, A p2p platform for distributed, collaborative and ubiquitous computing, IEEE Trans. on Industrial Electronics 58 (6) (2011) 2063–2172.

[6] A. B. Waluyo, D. Taniar, W. Rahayu, A. Aikebaier, M. Takizawa, B. Srinivasan, Trustworthy-based efficient data broadcast model for p2p interaction in resource-constrained wireless environments, Journal of Computer and System Sciences 78 (6) (2012) 1716–1736.

[7] L. Ogiela, Intelligent techniques for secure financial management in cloud computing, Electronic Commerce Research and Applications 14 (6) (2015) 456–464.

[8] L. Ogiela, M. R. Ogiela, Bio-inspired cryptographic techniques in information management applications, Proc. of IEEE the 30th International Conference on Advanced Information Networking and Applications, 2016, pp. 1059–1063.

[9] M. R. Ogiela, L. Ogiela, On using cognitive models in cryptography, Proc. of IEEE the 30th International Conference on Advanced Information Networking and Applications, 2016, pp. 1055–1058.

[10] E. B. Fernandez, R. C. Summers, C. Wood, Database Security and Integrity, Adison Wesley, Boston, MA, USA, 1980.

[11] D. F. Ferraiolo, D. R. Kuhn, R. Chandramouli, Role-based Access Control (2nd ed.), Artech, Norwood, MA, USA, 2007.

[12] S. Osborn, R. S. Sandhu, Q. Munawer, Configuring role-based access control to enforce mandatory and discretionary access control policies, ACM Transactions on Information and System Security 3 (2) (2000) 85–106.

[13] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, C. E. Youman, Role-based access control models, IEEE Computer 29 (2) (1996) 38–47.

[14] S. Nakamura, D. Duolikun, M. Takizawa, Read-abortion (ra) based synchronization protocols to prevent illegal information flow, Journal of Computer and System Sciences 81 (8) (2015) 1441–1451.

[15] S. Nakamura, D. Duolikun, T. Enokido, M. Takizawa, A write abortion-based protocol in role-based access control systems, International Journal of Adaptive and Innovative Systems 2 (2) (2015) 142–160.

[16] S. Nakamura, D. Duolikun, T. Enokido, M. Takizawa, A read-write abortion protocol to prevent illegal information flow in role-based access control systems, International Journal of Space-Based and Situated Computing 6 (1) (2016) 43–53.

[17] S. Nakamura, D. Duolikun, T. Enokido, M. Takizawa, A flexible read-write abortion protocol to prevent illegal information flow among objects, Journal of Mobile Multimedia 11 (3&4) (2015) 263–280.

[18] J. Bacon, D. M. Eyers, J. Singh, P. R. Pietzuch, Access control in publish/subscribe systems, Proc. of the 2nd International Conference on Distributed Event-based Systems, 2008, pp. 23–34.

[19] Google alert, `http://www.google.com/alerts`, Accessed August 1, 2018.

[20] R. Blanco, P. Alencar, Event models in distributed event based systems, Principles and Applications of Distributed Event-Based Systems, 2010, pp. 19–42.

[21] P. T. Eugster, P. A. Felber, R. Guerraoui, A.-M. Kermarrec, The many faces of publish/subscribe, ACM Computing Surveys 35 (2) (2003) 114–131.

[22] S. Tarkoma, Publish/Subscribe System : Design and Principles (First Edition), John Wiley and Sons, Hoboken, NJ, USA, 2012.

[23] C. Esposito, A. Castiglione, F. Palmieri, M. Ficco, K.-K. R. Choo, A publish/subscribe protocol for event-driven communications in the internet of things, Proc. of the IEEE 14th International Conference on Dependable, Autonomic and Secure Computing, 2016, pp. 376–383.

[24] V. Setty, M. V. Steen, R. Vitenberg, S. Voulgaris, Poldercast: Fast, robust, and scalable architecture for p2p topic-based pub/sub, Proc. of ACM/IFIP/USENIX 13th International Conference on Middleware, 2012, pp. 271–291.

[25] Y. Yamamoto, N. Hayashibara, Merging topic groups of a publish/subscribe system in causal order, Proc. of the 31st International Conference on Advanced Information Networking and Applications Workshops, 2017, pp. 172–177.

[26] S. Nakamura, L. Ogiela, T. Enokido, M. Takizawa, An information flow control model in a topic-based publish/subscribe system, Journal of High Speed Networks 24 (3) (2018) 243–257.

[27] L. Lamport, Time, clocks, and the ordering of event in a distributed systems, Communications of the ACM 21 (7) (1978) 558–565.

[28] D. E. Comer, Internetworking with TCP/IP Vol.1: Principles, Protocols, and Architecture, 2nd Edition, Prentice Hall, Englewood Cliffs, NJ, USA, 1990.

[29] K. Birman, A. Schiper, P. Stephenson, Lightweight causal and atomic group multicast, ACM Transactions on Computer Systems 9 (3) (1991) 272–314.

[30] F. Mattern, Virtual time and global states of distributed systems, Proc. of Workshop on Parallel and Distributed Algorithms, 1989, pp. 215–226.

[31] S. Tarkoma, M. Ain, K. Visala, The publish/subscribe internet routing paradigm (psirp) : Designing the future internet architecture, Future Internet Assembly, 2009, pp. 102–111.

[32] H. Nakayama, D. Duolikun, T. Enokido, M. Takizawa, Selective delivery of event messages in peer-to-peer topic-based publish/subscribe systems, Proc. of the 18th International Conference on Network-Based Information Systems, 2015, pp. 379–386.

[33] H. Nakayama, D. Duolikun, T. Enokido, M. Takizawa, Reduction of unnecessarily ordered event messages in peer-to-peer model of topic-based publish/subscribe systems, Proc. of IEEE the 30th International Conference on Advanced Information Networking and Applications, 2016, pp. 1160–1167.

[34] C. J. Date, An Introduction to Database Systems (8th Edition), Addison Wesley, Boston, MA, USA, 2003.

[35] J. Gray, A. Reuter, Transaction Processing: Concepts and Techniques, Morgan Kaufmann, Burlington, MA, USA, 1993.

[36] T. Enokido, M. Takizawa, A purpose-based synchronization protocol for secure information flow control, International Journal of Computer Systems Science and Engineering 25 (2) (2010) 25–32.

[37] T. Enokido, M. Takizawa, Purpose-based information flow control for cyber engineering, IEEE Transactions on Industrial Electronics 58 (6) (2011) 2216–2225.

[38] R. S. Sandhu, Lattice-based access control models, IEEE Computer 26 (11) (1993) 9–19.

[39] T. Enokido, M. Takizawa, A legal information flow (lif) scheduler based on role-based access control model, International Journal of Computer Standard and Interfaces 31 (5) (2009) 906–912.

[40] S. Nakamura, T. Enokido, M. Takizawa, Sensitivity-based synchronisation protocol to prevent illegal information flow among objects, International Journal of Web and Grid Services 13 (3) (2017) 315–333.

[41] S. Nakamura, T. Enokido, M. Takizawa, A flexible read-write abortion protocol with role safety concept to prevent illegal information flow, Journal of Ambient Intelligence and Humanized Computing 9 (5) (2018) 1415–1425.

[42] S. Nakamura, T. Enokido, M. Takizawa, A topic-based synchronisation protocol in peer-to-peer publish/subscribe systems, International Journal of Communication Networks and Distributed Systems 24 (1) (2019) 106–121.

[43] S. Nakamura, T. Enokido, M. Takizawa, A flexible synchronization protocol to learn hidden topics in p2pps systems, Transactions on Computational Collective Intelligence 33 (2019) 52–70.

[44] A. S. M. Kayes, J. Han, W. Rahayu, T. Dillon, M. S. Islam, A. Colman, A policy model and framework for context-aware access control to information resources, The Computer Journal 62 (5) (2019) 670–705.

[45] A. S. M. Kayes, W. Rahayu, T. Dillon, Critical situation management utilizing iot-based data resources through dynamic contextual role modeling and activation, Computing 101 (7).

[46] S. Nakamura, T. Enokido, M. Takizawa, Information flow control in object-based peer-to-peer publish/subscribe systems, Concurrency and Computation: Practice and Experience(accepted).

[47] S. Nakamura, T. Enokido, M. Takizawa, Causally ordering delivery of event messages in p2ppso systems, Cognitive Systems Research 56 (2019) 167–178.

[48] S. Nakamura, T. Enokido, M. Takizawa, Protocol to efficiently prevent illegal flow of objects in p2p type of publish/subscribe (ps) systems, Service Oriented Computing and Applications(accepted).

[49] A. Nakamura, M. Takizawa, Causally ordering broadcast protocol, Proc. of IEEE the 14th International Conference on Distributed Computing Systems, 1994, pp. 48–55.

[50] J. L. Hernández-Ramos, A. J. Jara, L. Marín, A. F. Skarmeta, Distributed capability-based access control for the internet of things, Journal of Internet Services and Information Security 3 (3/4) (2013) 1–16.

[51] S. Nakamura, T. Enokido, M. Takizawa, Information flow control based on the capbac (capability-based access control) model in the iot, International Journal of Mobile Computing and Multimedia Communications 10 (4) (2019) 13–25.

List of Publications and Researches

Refereed Journal Papers
(Total:18, First author papers:14, 2017-2020:12)

1. S. Nakamura, T. Enokido, and M. Takizawa, “Protocol to Efficiently Prevent Illegal Flow of Objects in P2P Type of Publish/Subscribe (PS) Systems,” accepted for publication at *Service Oriented Computing and Applications* (*SOCA*).

2. S. Nakamura, T. Enokido, and M. Takizawa, “Information Flow Control in Object-Based Peer-to-Peer Publish/Subscribe Systems,” accepted for publication at *Concurrency and Computation: Practice and Experience* (*CPE*), 2019.

3. S. Nakamura, T. Enokido, and M. Takizawa, “Information Flow Control Based on the CapBAC (Capability-Based Access Control) Model in the IoT,” *International Journal of Mobile Computing and Multimedia Communications* (*IJMCMC*), Vol.10, No.4, Dec. 2019, pp.13-25.

4. S. Nakamura, T. Enokido, M. Takizawa, “A Topic-Based Synchronisation Protocol in Peer-to-Peer Publish/Subscribe Systems,” *International Journal of Communication Networks and Distributed Systems* (*IJCNDS*), Vol.24, No.1, Nov. 2019, pp.106-121.

5. S. Nakamura, T. Enokido, and M. Takizawa, “Causally Ordering Delivery of Event Messages in P2PPSO Systems,” *Cognitive Systems Research* (*CSR*), Vol.56, Aug. 2019, pp.167-178.

6. R. Oma, S. Nakamura, D. Duolikun, T. Enokido, and M. Takizawa, “A Fault-Tolerant Tree-Based Fog Computing Model,” *International Journal of Web and Grid Services* (*IJWGS*), Vol.15, No.3, June 2019, pp.219-239.
7. S. Nakamura, T. Enokido, and M. Takizawa, “A Flexible Synchronization Protocol to Learn Hidden Topics in P2PPS Systems,” *Transactions on Computational Collective Intelligence* (*TCCI*), Vol.33, June 2019, pp.52-70.
8. S. Nakamura, T. Enokido, and M. Takizawa, “A Flexible Read-Write Abortion Protocol with Role Safety Concept to Prevent Illegal Information Flow,” *Journal of Ambient Intelligence and Humanized Computing* (*AIHC*), Vol.9, No.5, Oct. 2018, pp.1415-1425.
9. R. Oma, S. Nakamura, D. Duolikun, T. Enokido, and M. Takizawa, “An Energy-Efficient Model for Fog Computing in the Internet of Things (IoT),” *Internet of Things; Engineering Cyber Physical Human Systems*, Vol.1-2, Sept. 2018, pp.14-26.
10. S. Nakamura, L. Ogiela, T. Enokido, and M. Takizawa, “An Information Flow Control Model in a Topic-based Publish/Subscribe System,” *Journal of High Speed Networks* (*JHS*), Vol.24, No.3, June 2018, pp.243-257.
11. S. Nakamura, M. Sugino, and M. Takizawa, “Algorithms for Energy-efficient Broadcasting Messages in Wireless Networks,” *Journal of High Speed Networks* (*JHS*), Vol.24, No.1, Jan. 2018, pp.1-15.
12. H. Kataoka, S. Nakamura, D. Duolikun, T. Enokido, and M. Takizawa, “Multi-level Power Consumption Model and Energy-Aware Server Selection Algorithm,” *International Journal of Grid and Utility Computing* (*IJGUC*), Vol.8, No.3, Oct. 2017, pp.201-210.
13. S. Nakamura, T. Enokido, and M. Takizawa, “Sensitivity-Based Synchronisation Protocol to Prevent Illegal Information Flow among Objects,” *International Journal of Web and Grid Services* (*IJWGS*), Vol.13, No.3, June 2017, pp.315-333.

14. S. Nakamura, D. Duolikun, T. Enokido, and M. Takizawa, "A Read-Write Abortion Protocol to Prevent Illegal Information Flow in Role-based Access Control Systems," *International Journal of Space-Based and Situated Computing* (*IJSSC*), Vol.6, No.1, May 2016, pp.43-53.

15. S. Nakamura, D. Duolikun, T. Enokido, and M. Takizawa, "A Flexible Read-Write Abortion Protocol to Prevent Illegal Information Flow among Objects," *Journal of Mobile Multimedia* (*JMM*), Vol.11, No.3&4, Nov. 2015, pp.263-280.

16. S. Nakamura, D. Duolikun, T. Enokido, and M. Takizawa, "A Write Abortion-based Protocol in Role-based Access Control Systems," *International Journal of Adaptive and Innovative Systems* (*IJAIS*), Vol.2, No.2, Sept. 2015, pp.142-160.

17. D. Duolikun, S. Nakamura, T. Enokido, and M. Takizawa, "An Energy-Efficient Process Migration Approach to Reducing Electric Energy Consumption in a Cluster of Servers," *International Journal of Communication Networks and Distributed Systems* (*IJCNDS*), Vol.15, No.4, Sept. 2015, pp.400-420.

18. S. Nakamura, D. Duolikun, and M. Takizawa, "Read-Abortion (RA) Based Synchronization Protocols to Prevent Illegal Information Flow," *Journal of Computer and System Sciences* (*JCSS*), Vol.81, No.8, Dec. 2015, pp.1441-1451.

Refereed International Conference Papers
(Total:76, First authors:24, 2017-2020:46)

1. S. Nakamura, T. Enokido, and M. Takizawa, "A TBOI (Time-Based Operation Interruption) Protocol to Prevent Late Information Flow in the IoT," *Proc. of the 14th International Conference on Broad-Band Wireless Computing, Communication and Applications* (*BWCCA-2019*), Antwerp, Belgium, Nov. 2019, pp.93-104.

2. K. Gima, R. Oma, S. Nakamura, T. Enokido, and M. Takizawa, “Parallel Data Transmission Protocols in the Mobile Fog Computing Model,” *Proc. of the 14th International Conference on Broad-Band Wireless Computing, Communication and Applications* (*BWCCA-2019*), Antwerp, Belgium, Nov. 2019, pp.494-503.

3. Y. Guo, R. Oma, S. Nakamura, T. Enokido, and M. Takizawa, “Data Exchange Algorithm at Aggregate Level in the TWTBFC Model,” *Proc. of the 14th International Conference on Broad-Band Wireless Computing, Communication and Applications* (*BWCCA-2019*), Antwerp, Belgium, Nov. 2019, pp.114-124.

4. H. Ishii, R. Oma, S. Nakamura, T. Enokido, and M. Takizawa, “Algorithm for Detecting Implicitly Faulty Replicas Based on the Power Consumption Model,” *Proc. of the 14th International Conference on Broad-Band Wireless Computing, Communication and Applications* (*BWCCA-2019*), Antwerp, Belgium, Nov. 2019, pp.483-493.

5. R. Oma, S. Nakamura, T. Enokido, and M. Takizawa, “A Node Selection Algorithm for Fault Recovery in the GTBFC Model,” *Proc. of the 14th International Conference on Broad-Band Wireless Computing, Communication and Applications* (*BWCCA-2019*), Antwerp, Belgium, Nov. 2019, pp.81-92.

6. T. Saito, S. Nakamura, T. Enokido, and M. Takizawa, “The Group-Based Linear Time Causally Ordering Protocol in a Scalable P2PPS System,” *Proc. of the 14th International Conference on Broad-Band Wireless Computing, Communication and Applications* (*BWCCA-2019*), Antwerp, Belgium, Nov. 2019, pp.471-482.

7. S. Nakamura, T. Enokido, and M. Takizawa, “Evaluation of an OI (Operation Interruption) Protocol to Prevent Illegal Information Flow in the IoT,” *Proc. of the 22nd International Conference on Network-Based Information Systems* (*NBiS-2019*), Oita, Japan, Sept. 2019, pp.3-14.

8. K. Gima, R. Oma, S. Nakamura, T. Enokido, and M. Takizawa, "A Model for Mobile Fog Computing in the IoT," *Proc. of the 22nd International Conference on Network-Based Information Systems* (*NBiS-2019*), Oita, Japan, Sept. 2019, pp.447-458.
9. H. Ishii, R. Oma, S. Nakamura, T. Enokido, and M. Takizawa, "Fault Detection of Process Replicas on Reliable Servers," *Proc. of the 22nd International Conference on Network-Based Information Systems* (*NBiS-2019*), Oita, Japan, Sept. 2019, pp.423-433.
10. R. Oma, S. Nakamura, D. Duolikun, T. Enokido, and M. Takizawa, "Evaluation of Data and Subprocess Transmission Strategies in the Tree-Based Fog Computing Model," Proc. of the 22nd International Conference on Network-Based Information Systems (NBiS-2019), Oita, Japan, Sept. 2019, pp.15-26.
11. Y. Guo, R. Oma, S. Nakamura, D. Duolikun, T. Enokido, and M. Takizawa, "Data and Subprocess Transmission on the Edge Node of TWTBFC Model," Proc. of the 11th International Conference on Intelligent Networking and Collaborative Systems (INCoS-2019), Oita, Japan, Sept. 2019, pp.80-90.
12. T. Saito, S. Nakamura, T. Enokido, and M. Takizawa, "A Hierarchical Group of Peers in Publish/Subscribe Systems," Proc. of the 11th International Conference on Intelligent Networking and Collaborative Systems (INCoS-2019), Oita, Japan, Sept. 2019, pp.3-13.
13. R. Oma, S. Nakamura, D. Duolikun, T. Enokido, and M. Takizawa, "Subprocess Transmission Strategies for Recovering from Faults in the Tree-Based Fog Computing (TBFC) Model," Proc. of the 13th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2019), Sydney, Australia, July 2019, pp.50-61.
14. T. Saito, S. Nakamura, T. Enokido, and M. Takizawa, "Message Ordering Based on the Object-Based-Causally (OBC) Precedent Relation," Proc. of

the 13th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2019), Sydney, Australia, July 2019, pp.62-72.

15. S. Nakamura, T. Enokido, L. Barolli, and M. Takizawa, “Capability-Based Information Flow Control Model in the IoT,” Proc. of the 13th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS-2019), Sydney, Australia, July 2019, pp.63-71.

16. Y. Guo, R. Oma, S. Nakamura, D. Duolikun, T. Enokido, and M. Takizawa, “Evaluation of a Two-Way Tree-Based Fog Computing (TWTBFC) Model,” Proc. of the 13th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS-2019), Sydney, Australia, July 2019, pp.72-81.

17. S. Nakamura, T. Enokido, L. Barolli, and M. Takizawa, “Efficient Information Flow Control by Reducing Meaningless Messages in P2PPSO Systems,” Proc. of the 33rd International Conference on Advanced Information Networking and Applications (AINA-2019), Matsue, Japan, Mar. 2019, pp.108-119.

18. R. Oma, S. Nakamura, D. Duolikun, T. Enokido, and M. Takizawa, “Energy-Efficient Recovery Algorithm in the Fault-Tolerant Tree-Based Fog Computing (FTBFC) Model,” Proc. of the 33rd International Conference on Advanced Information Networking and Applications (AINA-2019), Matsue, Japan, Mar. 2019, pp.132-143.

19. Y. Guo, R. Oma, S. Nakamura, D. Duolikun, T. Enokido, and M. Takizawa, “A Two-Way Flow Model for Fog Computing,” Proc. of the 33rd International Conference on Advanced Information Networking and Applications Workshops (WAINA-2019), Matsue, Japan, Mar. 2019, pp.612-620.

20. T. Saito, S. Nakamura, D. Duolikun, T. Enokido, and M. Takizawa, “Evaluation of TBC and OBC Precedent Relations Among Messages in P2P Type of Topic-Based Publish/Subscribe System,” Proc. of the 33rd International

Conference on Advanced Information Networking and Applications Workshops (WAINA-2019), Matsue, Japan, Mar. 2019, pp.570-581.

21. R. Chida, Y. Guo, R. Oma, S. Nakamura, D. Duolikun, T. Enokido, and M. Takizawa, “Implementation of Fog Nodes in the Tree-Based Fog Computing (TBFC) Model of the IoT,” Proc. of the 7th International Conference on Emerging Internet, Data, and Web Technologies (EIDWT-2019), Fujairah, United Arab Emirates (UAE), Feb. 2019, pp.92-102.

22. S. Hayashi, S. Nakamura, D. Duolikun, T. Enokido, and M. Takizawa, “Evaluation of a Protocol to Prevent Illegal Information Flow Based on Maximal Roles in the RBAC Model,” Proc. of the 7th International Conference on Emerging Internet, Data, and Web Technologies (EIDWT-2019), Fujairah, United Arab Emirates (UAE), Feb. 2019, pp.80-91.

23. S. Nakamura, T. Enokido, and M. Takizawa, “Evaluation of Object-Based Information Flow Control in P2PPS Systems,” Proc. of the 13th International Conference on Broad-Band Wireless Computing, Communication and Applications (BWCCA-2018), Taichung, Taiwan, Oct. 2018, pp.125-134.

24. S. Hayashi, S. Nakamura, D. Duolikun, T. Enokido, and M. Takizawa, “A Synchronization Protocol to Prevent Illegal Information Flow Based on Maximal Roles in the Role-Based Access Control Model,” Proc. of the 13th International Conference on Broad-Band Wireless Computing, Communication and Applications (BWCCA-2018), Taichung, Taiwan, Oct. 2018, pp.525-533.

25. T. Saito, S. Nakamura, D. Duolikun, T. Enokido, and M. Takizawa, “Object-Based Selective Delivery of Event Messages in Topic-Based Publish/Subscribe Systems,” Proc. of the 13th International Conference on Broad-Band Wireless Computing, Communication and Applications (BWCCA-2018), Taichung, Taiwan, Oct. 2018, pp.444-455.

26. R. Oma, S. Nakamura, D. Duolikun, T. Enokido, and M. Takizawa, “Fault-Tolerant Fog Computing Models in the IoT,” Proc. of the 13th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC-2018), Taichung, Taiwan, Oct. 2018, pp.14-25.

27. S. Nakamura, T. Enokido, and M. Takizawa, “Object-Based Information Flow Control Model in P2PPS Systems,” Proc. of the 21st International Conference on Network-Based Information Systems (NBiS-2018), Bratislava, Slovakia, Sept. 2018, pp.110-121.

28. D. Duolikun, S. Nakamura, T. Enokido, and M. Takizawa, “An Energy-Efficient Dynamic Live Migration of Multiple Virtual Machines,” Proc. of the 21st International Conference on Network-Based Information Systems (NBiS-2018), Bratislava, Slovakia, Sept. 2018, pp.87-98.

29. R. Oma, S. Nakamura, D. Duolikun, T. Enokido, and M. Takizawa, “Evaluation of an Energy-Efficient Tree-Based Model of Fog Computing,” Proc. of the 21st International Conference on Network-Based Information Systems (NBiS-2018), Bratislava, Slovakia, Sept. 2018, pp.99-109.

30. T. Saito, S. Nakamura, T. Enokido, and M. Takizawa, “A Causally Precedent Relation Among Messages in Topic-Based Publish/Subscribe Systems,” Proc. of the 21st International Conference on Network-Based Information Systems (NBiS-2018), Bratislava, Slovakia, Sept. 2018, pp.543-553.

31. S. Nakamura, T. Enokido, and M. Takizawa, “Evaluation of a Protocol to Prevent Malicious Information Flow in P2PPS Systems,” Proc. of the 12th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2018), Matsue, Japan, Jul. 2018, pp.102-114.

32. R. Oma, S. Nakamura, T. Enokido, and M. Takizawa, “A Tree-Based Model of Energy-Efficient Fog Computing Systems in IoT,” Proc. of the 12th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2018), Matsue, Japan, Jul. 2018, pp.991-1001.

33. E. Ogawa, S. Nakamura, T. Enokido, and M. Takizawa, “One-to-One Routing Protocols for Wireless Ad-Hoc Networks Considering the Electric Energy Consumption,” Proc. of the 12th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS-2018), Matsue, Japan, Jul. 2018, pp.105-115.

34. S. Nakamura, L. Ogiela, T. Enokido, and M. Takizawa, “A Protocol to Prevent Malicious Information Flow in P2PPS Systems,” Proc. of the 32nd International Conference on Advanced Information Networking and Applications (AINA-2018), Cracow, Poland, May 2018, pp.24-31.

35. E. Ogawa, S. Nakamura, T. Enokido, and M. Takizawa, “Unicast Routing Protocols to Reduce Electric Energy Consumption in Wireless Ad-hoc Networks,” Proc. of the 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA-2018), Cracow, Poland, May 2018, pp.533-538.

36. R. Oma, S. Nakamura, T. Enokido, and M. Takizawa, “An Energy-efficient Model of Fog and Device Nodes in IoT,” Proc. of the 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA-2018), Cracow, Poland, May 2018, pp.301-306.

37. S. Nakamura, L. Ogiela, T. Enokido, and M. Takizawa, “Malicious Information Flow in P2PPS Systems,” Proc. of the 6th International Conference on Emerging Internet, Data, and Web Technologies (EIDWT-2018), Tirana, Albania, Mar. 2018, pp.119-129.

38. S. Nakamura, L. Ogiela, T. Enokido, and M. Takizawa, “A Flexible Synchronization Protocol for Hidden Topics to Prevent Illegal Information Flow in P2PPS Systems,” Proc. of the 12th International Conference on Broad-Band Wireless Computing, Communication and Applications (BWCCA-2017), Barcelona, Spain, Nov. 2017, pp.138-148.

39. E. Ogawa, S. Nakamura, T. Enokido, and M. Takizawa, “A Low-Energy Unicast Ad-Hoc Routing Protocol in Wireless Networks,” Proc. of the 12th

International Conference on Broad-Band Wireless Computing, Communication and Applications (BWCCA-2017), Barcelona, Spain, Nov. 2017, pp.173-184.

40. R. Oma, S. Nakamura, T. Enokido, and M. Takizawa, "Hybrid Replication Schemes of Processes for Fault-Tolerance Systems in Energy-Efficient Server Clusters," Proc. of the 12th International Conference on Broad-Band Wireless Computing, Communication and Applications (BWCCA-2017), Barcelona, Spain, Nov. 2017, pp.597-607.
41. S. Nakamura, L. Ogiela, T. Enokido, and M. Takizawa, "Evaluation of Flexible Synchronization Protocol to Prevent Illegal Information Flow in P2PPS Systems," Proc. of the 20th International Conference on Network-Based Information Systems (NBiS-2017), Toronto, Canada, Aug. 2017, pp.66-77 (Best paper award).
42. E. Ogawa, S. Nakamura, T. Enokido, and M. Takizawa, "An Energy-Aware One-to-one Routing Protocol in Wireless Ad-Hoc Network," Proc. of the 20th International Conference on Network-Based Information Systems (NBiS-2017), Toronto, Canada, Aug. 2017, pp.102-113.
43. R. Oma, S. Nakamura, T. Enokido, and M. Takizawa, "Hybrid Replication Schemes of Processes in Energy-Efficient Server Clusters," Proc. of the 20th International Conference on Network-Based Information Systems (NBiS-2017), Toronto, Canada, Aug. 2017, pp.699-710.
44. S. Nakamura, L. Ogiela, T. Enokido, and M. Takizawa, "Flexible Synchronization Protocol to Prevent Illegal Information Flow in Peer-to-Peer Publish/Subscribe Systems," Proc. of the 11th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2017), Turin, Italy, Jul. 2017, pp.82-93.
45. E. Ogawa, S. Nakamura, and M. Takizawa, "An Energy-Saving Unicast Routing Protocol in Wireless Ad-hoc Network," Proc. of the 11th Interna-

tional Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS-2017), Turin, Italy, Jul. 2017, pp.110-120.

46. A. Sawada, S. Nakamura, D. Duolikun, T. Enokido, and M. Takizawa, “Simple Energy-Aware Algorithms to Selecting a Server for Storage and Computation Processes in a Cluster,” Proc. of the 11th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS-2017), Turin, Italy, Jul. 2017, pp.98-109.

47. S. Nakamura, L. Ogiela, T. Enokido, and M. Takizawa, “Evaluation of Protocols to Prevent Illegal Information Flow in Peer-to-Peer Publish/Subscribe Systems,” Proc. of IEEE the 31st International Conference on Advanced Information Networking and Applications (AINA-2017), Taipei, Taiwan, Mar. 2017, pp.631-638.

48. E. Ogawa, S. Nakamura, and M. Takizawa, “A Trustworthiness-based Adhoc Routing Protocol in Wireless Networks,” Proc. of IEEE the 31st International Conference on Advanced Information Networking and Applications (AINA-2017), Taipei, Taiwan, Mar. 2017, pp.1162-1168.

49. H. Kataoka, S. Nakamura, T. Enokido, and M. Takizawa, “Simple Energy-aware Algorithms for Selecting a Server in a Scalable Cluster,” Proc. of the 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA-2017), Taipei, Taiwan, Mar. 2017, pp.146-153.

50. H. Nakayama, E. Ogawa, S. Nakamura, T. Enokido, and M. Takizawa, “Topic-based Selective Delivery of Event Messages in Peer-to-peer Model of Publish/Subscribe Systems in Heterogeneous Networks,” Proc. of the 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA-2017), Taipei, Taiwan, Mar. 2017, pp.327-334.

51. A. Sawada, S. Nakamura, H. Kataoka, T. Enokido, and M. Takizawa, “Algorithms to Energy-efficiently Select a Server for a General Process in a Scal-

able Cluster," Proc. of the 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA-2017), Taipei, Taiwan, Mar. 2017, pp.138-145.

52. S. Nakamura, T. Enokido, and M. Takizawa, "Topic-based Synchronization (TBS) Protocols to Prevent Illegal Information Flow in Peer-to-Peer Publish/Subscribe Systems," Proc. of the 11th International Conference on Broad-Band Wireless Computing, Communication and Applications (BWCCA-2016), Asan, Korea, Nov. 2016, pp.57-68.

53. D. Duolikun, S. Nakamura, R. Watanabe, T. Enokido, and M. Takizawa, "Energy-aware Migration of Virtual Machines in a Cluster," Proc. of the 11th International Conference on Broad-Band Wireless Computing, Communication and Applications (BWCCA-2016), Asan, Korea, Nov. 2016, pp.21-32.

54. E. Ogawa, S. Nakamura, and M. Takizawa, "An Energy-efficient and Reliable Protocol in Wireless Networks," Proc. of the 11th International Conference on Broad-Band Wireless Computing, Communication and Applications (BWCCA-2016), Asan, Korea, Nov. 2016, pp.585-592.

55. S. Nakamura, T. Enokido, and M. Takizawa, "Subscription Initialization (SI) Protocol to Prevent Illegal Information Flow in Peer-to-Peer Publish/Subscribe Systems," Proc. of the 19th International Conference on Network-Based Information Systems (NBiS-2016), Ostrava, Czech Republic, Sept. 2016, pp.42-49.

56. H. Nakayama, S. Nakamura, T. Enokido, and M. Takizawa, "Topic-based Causally Ordered Delivery of Event Messages in a Peer-to-peer (P2P) Model of Publish/Subscribe Systems," Proc. of the 19th International Conference on Network-Based Information Systems (NBiS-2016), Ostrava, Czech Republic, Sept. 2016, pp.348-354.

57. S. Nakamura, T. Enokido, and M. Takizawa, "Information Flow Control

Models in Peer-to-Peer Publish/Subscribe Systems," Proc. of the 10th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2016), Fukuoka, Japan, Jul. 2016, pp.167-174.

58. H. Nakayama, S. Nakamura, T. Enokido, and M. Takizawa, "Scalable Group Communication Protocols in the Peer-to-peer Model of Topic-based Publish/Subscribe Systems," Proc. of the 10th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2016), Fukuoka, Japan, Jul. 2016, pp.142-149.

59. S. Nakamura, D. Duolikun, T. Enokido, and M. Takizawa, "Influential Abortion Probability in a Flexible Read-Write Abortion Protocol," Proc. of IEEE the 30th International Conference on Advanced Information Networking and Applications (AINA-2016), Crans-Montana, Switzerland, Mar. 2016, pp.1-8 (国際交流援助).

60. D. Duolikun, R. Watanabe, H. Kataoka, S. Nakamura, T. Enokido, and M. Takizawa, "An Energy-aware Migration of Virtual Machines," Proc. of IEEE the 30th International Conference on Advanced Information Networking and Applications (AINA-2016), Crans-Montana, Switzerland, Mar. 2016, pp.557-564.

61. H. Honda, S. Nakamura, H. Nakayama, D. Duolikun, T. Enokido, and M. Takizawa, "A Scalable Group Communication Protocol in Heterogeneous Networks," Proc. of the 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA-2016), Crans-Montana, Switzerland, Mar. 2016, pp.294-299.

62. M. Sugino, S. Nakamura, T. Enokido, and M. Takizawa, "Protocols for Energy-efficiently Broadcasting Messages in Wireless Networks," Proc. of the 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA-2016), Crans-Montana, Switzerland, Mar. 2016, pp.286-293.

63. S. Nakamura, D. Duolikun, T. Enokido, and M. Takizawa, "Role Safety in a Flexible Read-Write Abortion Protocol," Proc. of the 10th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA-2015), Krakow, Poland, Nov. 2015, pp.333-340.
64. D. Duolikun, H. Kataoka, S. Nakamura, T. Enokido, and M. Takizawa, "Energy-aware Migration and Replication of Processes in a Cluster," Proc. of the 10th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA-2015), Krakow, Poland, Nov. 2015, pp.283-287.
65. S. Nakamura, D. Duolikun, T. Enokido, and M. Takizawa, "A Flexible Read-Write Abortion Protocol with Sensitivity of Roles," Proc. of the 18th International Conference on Network-Based Information Systems (NBiS-2015), Taipei, Taiwan, Sept. 2015, pp.132-139.
66. H. Honda, S. Nakamura, D. Duolikun, T. Enokido, and M. Takizawa, "Reduction of Unnecessarily Ordered Messages in Scalable Group Communication," Proc. of the 18th International Conference on Network-Based Information Systems (NBiS-2015), Taipei, Taiwan, Sept. 2015, pp.99-106.
67. S. Nakahira, S. Nakamura, T. Enokido, and M. Takizawa, "Trustworthiness in Peer-to-Peer Systems," Proc. of the 18th International Conference on Network-Based Information Systems (NBiS-2015), Taipei, Taiwan, Sept. 2015, pp.652-657.
68. A. Sawada, S. Nakamura, T. Enokido, and M. Takizawa, "Eco Models of Storage-based Servers," Proc. of the 18th International Conference on Network-Based Information Systems (NBiS-2015), Taipei, Taiwan, Sept. 2015, pp.407-411.
69. M. Sugino, S. Nakamura, T. Enokido, and M. Takizawa, "Energy-efficient Broadcast Protocols in Wireless Networks," Proc. of the 18th International Conference on Network-Based Information Systems (NBiS-2015), Taipei, Taiwan, Sept. 2015, pp.357-364.

70. S. Nakamura, D. Duolikun, T. Enokido, and M. Takizawa, “A Flexible Read-Write Abortion Protocol with Sensitivity of Objects to Prevent Illegal Information Flow,” Proc. of the 9th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2015), Blumenau, Brazil, Jul. 2015, pp.289-296 (Best paper award, C&C 若手優秀論文賞, 国際会議論文発表者助成).

71. D. Duolikun, S. Nakamura, T. Enokido, and M. Takizawa, “Energy-efficient Replication and Migration of Processes in a Cluster,” Proc. of the 9th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2015), Blumenau, Brazil, Jul. 2015, pp.118-125.

72. M. Sugino, S. Nakamura, T. Enokido, and M. Takizawa, “Trustworthiness-based Broadcast Protocols in Wireless Networks,” Proc. of the 9th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS-2015), Blumenau, Brazil, Jul. 2015, pp.125-132.

73. S. Nakamura, D. Duolikun, T. Enokido, and M. Takizawa, “A Flexible Read-Write Abortion Protocol to Prevent Illegal Information Flow,” Proc. of IEEE the 29th International Conference on Advanced Information Networking and Applications (AINA-2015), Gwangju, Korea, Mar. 2015, pp.155-162.

74. S. Nakamura, D. Duolikun, A. Aikebaier, T. Enokido, and M. Takizawa, “Read-Write Abortion (RWA) Based Synchronization Protocols to Prevent Illegal Information Flow,” Proc. of the 17th International Conference on Network-Based Information Systems (NBiS-2014), Salerno, Italy, Sept. 2014, pp.120-127.

75. S. Nakamura, D. Duolikun, A. Aikebaier, T. Enokido, and M. Takizawa, “Synchronization Protocols to Prevent Illegal Information Flow in Role-based Access Control Systems,” Proc. of the 8th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2014), Birmingham, UK, Jul. 2014, pp.279-286.

76. S. Nakamura, D. Duolikun, A. Aikebaier, T. Enokido, and M. Takizawa, "Role-based Information Flow Control Models," Proc. of IEEE the 28th International Conference on Advanced Information Networking and Applications (AINA-2014), Victoria, Canada, May 2014, pp.1140-1147.

Awards

1. Best paper award, The 20th International Conference on Network-Based Information Systems (NBiS-2017), Toronto, Canada, Aug. 2017.
2. Outstanding Paper Award for Young C&C Researchers (C&C 若手優秀論文賞), The NEC C&C Foundation (公益財団法人 NEC C&C 財団), Jan. 2016.
3. Best paper award, The 9th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2015), Blumenau, Brazil, July 2015.

Research Grant

1. Grant-in-Aid for JSPS Research fellow (特別研究員奨励費 (DC1)), Japan Society for the Promotion of Science (独立行政法人 日本学術振興会), Apr. 2017-Mar. 2020.

Scholarship

1. Repayment Exemption for Students with Excellent Grades (特に優れた業績による返還免除 (全額免除)), Japan Student Services Organization (日本学生支援機構), Aug. 2017.

Travel Grants

1. Grants for Researchers Attending International Conferences (国際交流援助), Research Foundation for the Electrotechnology of Chubu (中部電気利用基礎研究振興財団), The 30th IEEE International Conference on Advanced Information Networking and Applications (AINA-2016), Dec. 2015.

2. Grants for Researchers Attending International Conferences (国際会議論文発表者助成), The NEC C&C Foundation (公益財団法人 NEC C&C 財団), The 9th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2015), Apr. 2015.