Cloud Network Operation Management Technology to Support Trust of Services in the Era of SoE

● Naoki Oguchi ● Hiroshi Tomonaga ● Hitoshi Ueno ● Yasuhiko Aoki

Systems of engagement (SoE) are ICT systems for creating and maintaining connections between companies and customers and building trust. SoE must be able to offer flexibility and quickness in responding to changes in business, in addition to availability comparable to that of systems of record (SoR), which are ICT systems that support the mission-critical operations in an organization. For services provided by SoE to be used by more users, a system must essentially satisfy the security and integrity requirements that allow for a greater sense of security in users. However, grasping the configuration and requirements of an entire system is generally difficult for service developers, and it was difficult to redesign a system considering the security and performance based on the service every time improvements were made to the service. Therefore, Fujitsu Laboratories has developed a system of building SoE and supporting stable operations so that such service developers can quickly and flexibly provide a secure service in a multi-cloud environment. This paper presents network as code (NaC) and network verification technologies that assist even those service developers who lack an understanding of the entire system with the construction of a secure system. It also describes traffic prediction and anomaly detection technologies, which promptly recognize any failures and performance degradation in an SoE system to allow for the provision of integrity.

1. Introduction

Until now, accurate and stable operations were expected in ICT systems, such as corporate backbone systems and systems handling personal information. Therefore, they were constructed and operated based on redundancy design and sufficient testing. These systems are called systems of record (SoR).

At the same time, changes in business have also accelerated with the progress of recent Web technologies and the virtualization and container technologies mainly on clouds. The ability to quickly reflect customer feedback in systems and continuously update is required to improve user satisfaction and gain advantages in business. These systems are called systems of engagement (SoE).

SoE require a different type of system reliability ("trust") than before. "Trust" in this context means a system's ability to provide integrity to satisfy the given response time and performance requirements as well as security to safeguard the data.[1)] Therefore, SoE require a scheme that can flexibly and quickly provide these "trusts".

It is therefore necessary to first ensure security every time a service is updated, which includes changing the security level and checking for any security violations in relation to other services depending on the scale, content, and provision phase of the service. A scheme is required that enables even service developers who have difficulty in understanding the configuration and requirements of the entire system to easily and flexibly build a system that satisfies the security requirements.

Meanwhile, with a virtual infrastructure such as a cloud, it has been difficult for service developers to identify failure points and factors behind performance degradation not only because various SoE are multiplexed but also because adequate tools are not available. Therefore, technology to identify factors behind failures and resource shortages more quickly and deal with or avoid them is required to achieve integrity.

Fujitsu Laboratories has developed a system for

supporting the construction and stable operation of SoE so that such service developers can provide secure services in a multi-cloud environment quickly and flexibly.

This paper first presents network as code (NaC) and network verification technologies that assist even those service developers who lack an understanding of the entire system so that they can construct a secure system. Next, it presents a cloud network analysis technology, which quickly recognizes failures and performance degradation in SoE and allows integrity to be provided.

2. Reliability required for SoE

Generally, SoE are cloud-native business systems that are implemented by arranging various microservices consisting of containers in a virtual infrastructure composed of virtual machines (VMs), virtual networks, etc. and connecting them with virtual networks. The development cycle of such systems–supply of the system, customer feedback, and system modification–is repeated over a short period to promptly improve services offered to customers. However, improving the services requires extensive know-how regarding computing and network settings to ensure security. Furthermore, providing or outsourcing specialized staff costs time and money.

3. Virtual network construction and verification technology to provide quick services based on trust levels

This section gives an explanation about the NaC technology, which allows SoE to be constructed quickly according to the required trust level, and a network verification technology to ensure consistency in terms of security settings in the constructed system.

3.1 Outline of NaC technology

With SoE, services are composed of a combination of many independent functions (microservices). Generally, monolithic systems are tightly coupled in a single module and then service enhancement require additional work. On the other hand, SoE have each function constructed and loosely coupled together, which allows each microservice to be developed at its own speed. This makes it easier for the entire system to follow changes in the environment.

On the other hand, not only does the number of services increase but each service is also updated frequently and the execution machine for the updated service is determined each time. This causes frequent changes in the execution machine. Therefore, in a virtual network connecting individual services, connections must be changed according to the position of the service execution machine so as not to disturb flexibility and agility, the advantages gained by adopting microservices.

This is where NaC comes in. NaC is a system for automatically deriving the detailed functions and settings required for the individual virtual networks simply by describing the service developers' own various requirements necessary to construct the service as intents, such as the connectivity, access control, and trust level (**Figure 1**). For example, intents may be described as allowing a certain service to only be used from a specific service or to be stopped during a failure because it is still in the development phase. It becomes possible to connect microservices quickly and at low cost simply by specifying these intents. There is no need to be aware of the configuration of the virtual infrastructure, which is composed of containers and virtual networks.

3.2 Realization of NaC

First, in order to automatically derive settings according to various virtual infrastructure configurations, the processing is separated into procedures that do not depend on the virtual infrastructure configuration (function derivation) and procedures that do depend on this configuration (topology matching), as shown in Figure 1.

The upper part of the figure illustrates function derivation. Function derivation is a phase in which network functions and their connections are derived from an intent. Functions are derived by applying the intent to derivation rules. Attempting to deal with various intents of service developers with a single derivation rule causes the rule to be complicated, which poses a problem. Accordingly, we define a number of simple derivation rules. Searching for rules to be used for derivation according to the intent, the detected rules are applied step by step (R5, R2, and R6 in Figure 1) to achieve gradual derivation. In this way, various intents can be dealt with by combining rules without the need for complicating a derivation rule.

The bottom part of the Figure illustrates topology

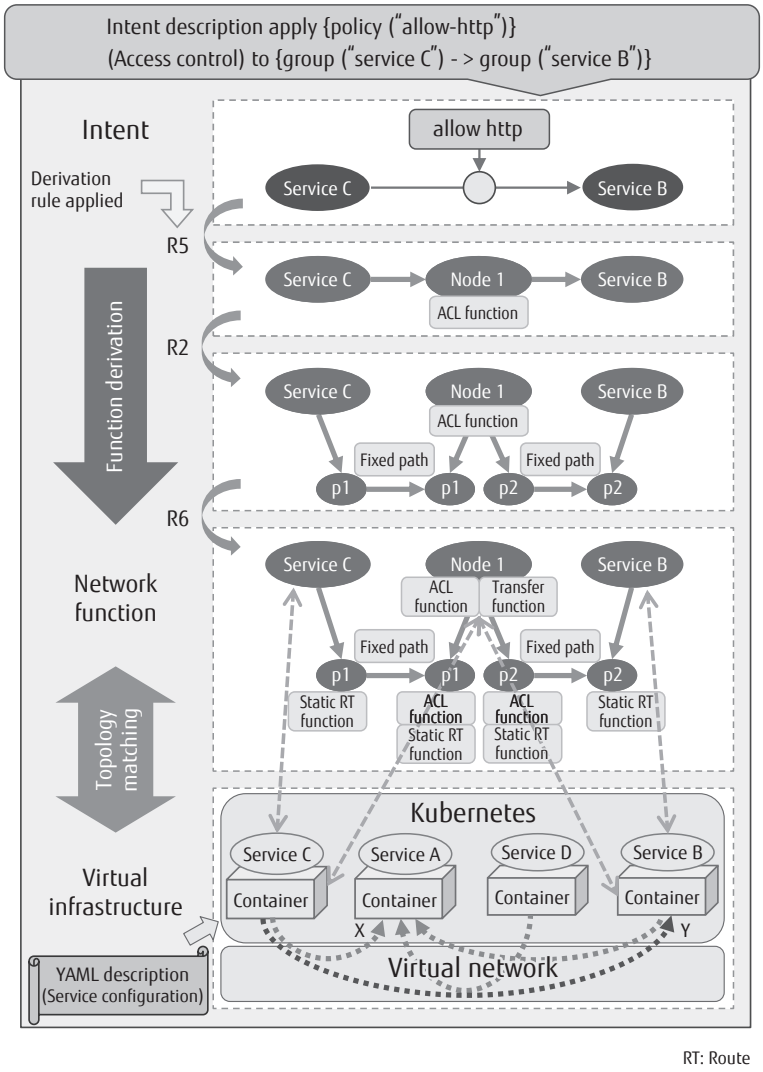


Figure 1
Realization of NaC and use cases.

matching. The connection for the functions derived with the Function derivation is compared with the actual configuration of the virtual infrastructure to determine the components that implement the functions. Once the function deployment has been determined, the settings and parameters for implementing the functions are derived to configure the virtual network. In this way, function derivation and topology matching are performed to automatically derive the network settings.

With Kubernetes, which is becoming the de facto standard for orchestration platforms of microservices, container deployment is determined based on the service configuration information described in YAML files. In addition, application programming interface (API) access control between dependent services is required. Use of NaC here allows this access control to be automatically built simply by describing the access control as a state to be satisfied by the network.

3.3 Microservice use cases

This subsection describes NaC use cases in a microservice environment.

For example, let's assume that API endpoint X provided by Service A is accessible from an arbitrary service, and that API endpoint Y provided by Service B is accessible only from Service C. To allow http access from Service C to Service B in this situation, the state is

described as an intent as shown by the callout shown in Figure 1. Based on this description, NaC automatically derives and configures the settings of the network, and http access from Service C to Service B is permitted.

This makes it possible to immediately realize SoE that response to business and environment changes.

3.4 Issues with network verification technology

This technology collects the content of forwarding tables of multiple routers and filters known as access control lists (ACLs) in firewalls (FWs) located in a virtual network to verify the consistency of the entire system.

An ACL is basically expressed as a five-tuple, i.e. source IP, source port, destination IP, destination port, and protocol. Here, the source IP and source port represent the IP address and port number of the source application in the communication using a TCP or UDP protocol. The destination IP and destination port represent the IP address and port number of the destination application. The protocol represents the protocol number to identify whether it is a TCP or UDP.

When a service is built on a large-scale virtual infrastructure, an ACL must be set on multiple FWs included in the system. In doing so, failure to ensure consistency in the settings between the multiple FWs may lead to the inability to communicate between microservices or cause the security of services to be compromised.

In addition, when various services are multiplexed into a virtual infrastructure, different service developers may make conflicting intent settings. In this case, the individual developers need to set an ACL in such a way that the communication generated by the services they have developed can pass through each FW. However, overwriting existing ACL settings with their own ACL settings that conflict with those of other developers may interfere with other services.

For example, in **Figure 2**, assume that Developer #1 sets ACL11 and ACL12 on FW1 and FW2 respectively so that Communication #1 (Session #1) generated by the service can pass through. Unaware of this, Developer #2 may set ACL21 and ACL22 on FW2 and FW3 so that Communication #2 (Session #2) generated by the service can pass through. As a result, if ACL21 and ACL12 have conflicting settings, Session #1 may

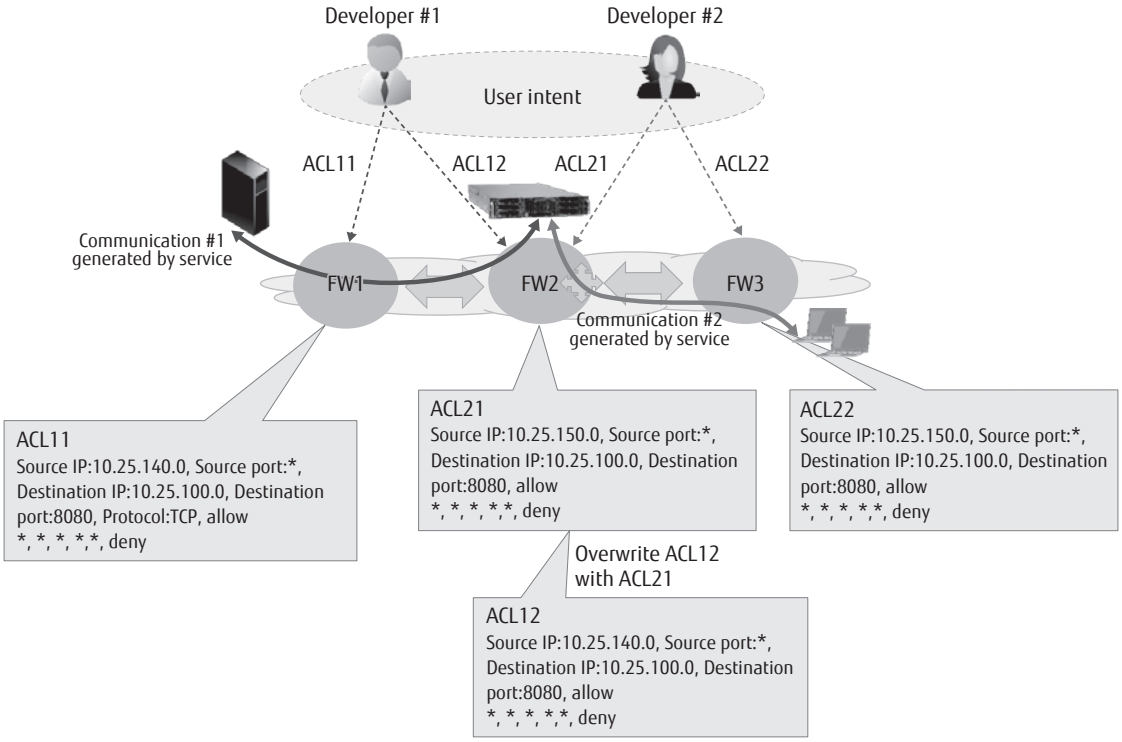


Figure 2
Issues with firewall settings.

be stopped from passing through. Accordingly, when a system has multiple FWs to allow passage of communications generated by various services, it is difficult for humans to check for consistency, duplication, and conflicts between these rules.

To deal with these issues, this technology mathematically verifies whether the network and system settings are correct and predicts the behavior of every packet. Specifically, the following is performed (**Figure 3**).

1) Configuration information (ACLs) is collected from devices in the network.
2) All ACLs are divided into sub-ACLs corresponding to disjoint address ranges (equivalence classes: ECs) that do not overlap.
3) Sub-ACLs are used to represent all ACLs in the system as a forwarding graph.
4) The forwarding graph is represented by a matrix. The reachability matrix is computed to check for any inconsistencies in reachability.

By computing these in real time every time any setting change is made, conflicts in the settings can be identified before packets actually flow.

4. Cloud network analysis technology

This section describes a cloud network analysis technology, indispensable in the SoE era and which allows for the provision of integrity based on the requirements of service providers.

4.1 Basic concept

In order to ensure stable operation of services, it is important to identify and isolate failure factors promptly by using information on the behavior of VMs, virtual networks, and containers. In particular, it is expected that, as systems become increasingly larger and cloud services more diversified in the future, the conventional fixed threshold setting for each measurement parameter or monitoring based on the know-how of skilled operators will be insufficient. Therefore, the upgrading of system and network operation management using AI technology such as machine learning is expected to handle real-time data collected from VMs and virtual networks.

Possible factors hindering the stable operation of services include problems such as setting errors by operators and exposure of software bugs and hardware failures. It is, however, difficult to always monitor the behaviors of all components. But when a component slows down, for example, it appears as a response degradation of a service or a sudden traffic decrease. In this way, assuming that an error in the individual component is exposed as a communication error at the service level, the traffic prediction technology and traffic anomaly detection technology are applied to services and virtual networks.

4.2 Architecture

Figure 4 shows the system architecture of the proposed method. Service quality information and container metrics are obtained from an Istio controller[2], an open-source software (OSS). Based on these, the response time of services and its variation are calculated. Regarding network quality information, the trace packet extraction unit and packet performance analysis server developed by Fujitsu Laboratories are used to calculate the packet loss rate and traffic volume. The trace packet extraction unit provides a function for grouping packets

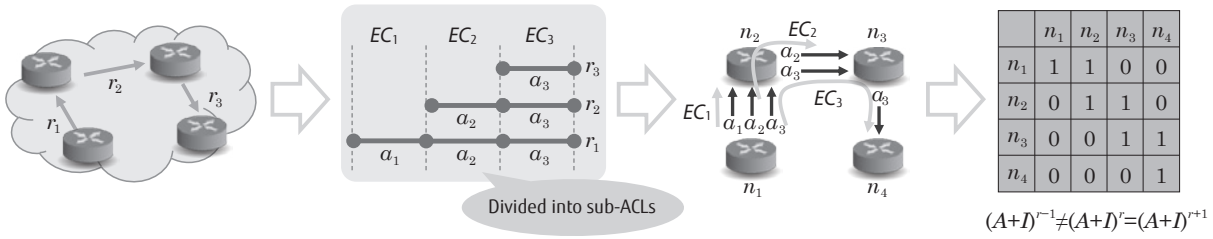


Figure 3
How network verification technology works.

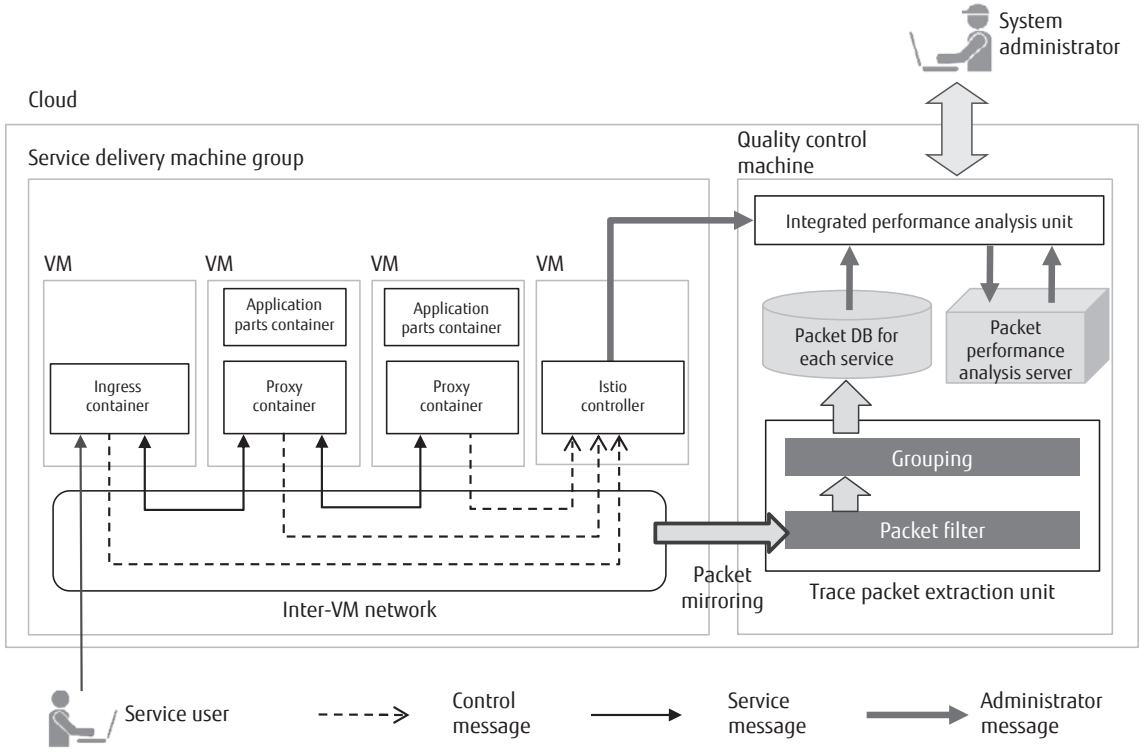


Figure 4
Proposed system architecture.

captured from an inter-VM network for each service in real time and storing them in a packet DB for each service. The packet performance analysis server provides a function for calculating packet response delays and packet loss rates. The integrated performance analysis unit aggregates both service quality information and network quality information and provides traffic predictions and anomaly analysis functions as shown in the next subsection. In this way, we have separated the function for collecting and calculating quality information from the function for carrying out the overall analysis to provide an architecture that facilitates the addition and extension of new means of analysis.

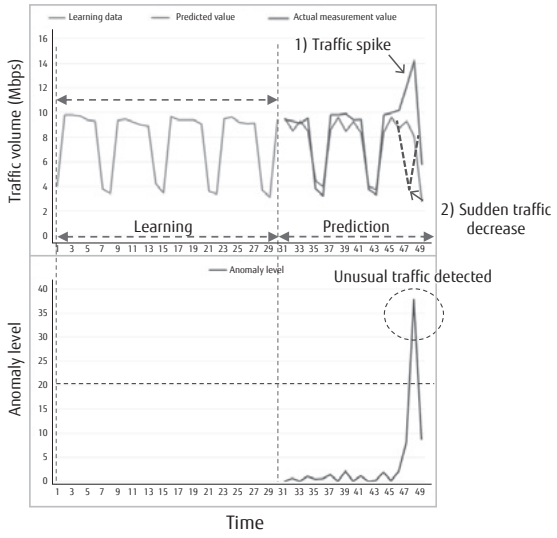
4.3 Traffic prediction technology

This subsection presents a traffic prediction technology that uses an autoregressive-moving-average (ARIMA) model as a means to realize highly accurate traffic change detection for virtual networks. This technology observes time-varying objects such as traffic and first models the periodic pattern and its trends based on autocorrelation and moving averages of past time-series data. Next, an autoregressive prediction is made for future time and the difference between the prediction and the actual measurement is determined to detect any changes.[3)]

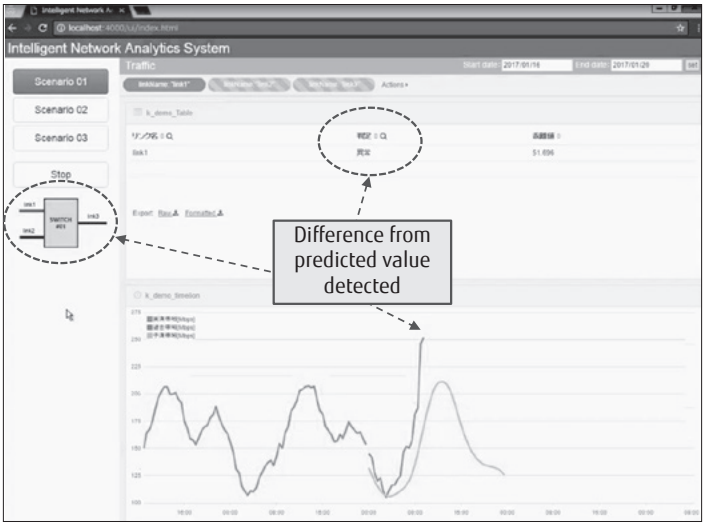
Figure 5 shows the results of the prediction based on the developed model and of the traffic anomaly detection based on the difference from the traffic prediction. Figure 5 (a) indicates values calculated from the difference between the predicted traffic volume and the actual traffic volume as an anomaly level. Figure 5 (b) shows an example of the anomaly level indicated on the GUI. By providing this capability of quickly detecting unusual behavior, it becomes possible to make prompt recommendations, such as the redeployment of VMs and containers.

4.4 Traffic anomaly detection technology

In addition to the method using time-series data described in the previous subsection, there is a method for detecting unusual states by focusing on the changes



(a) Difference between predicted and actual measurement values



(b) Example of anomaly level

Figure 5
Traffic prediction example.

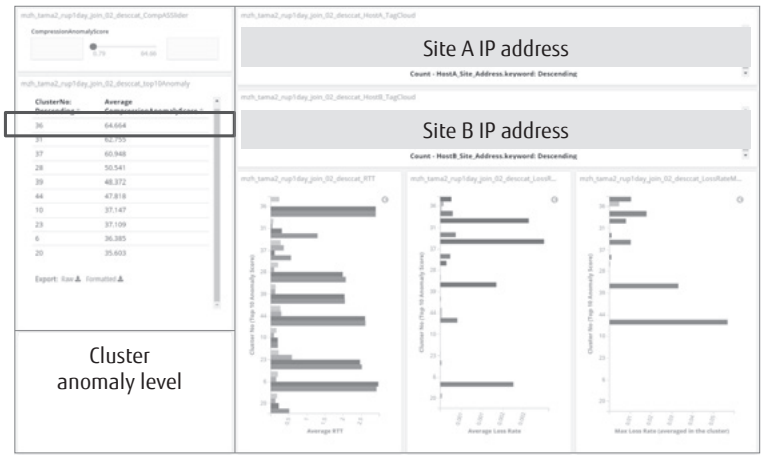


Figure 6
Anomaly detection GUI.

in the distribution of certain measured values such as packet capture data. Application of this monitoring of packet capture or other measurements to services, VMs, or inter-VM networks requires seasoned operation skills with virtual infrastructure. This is a complex, time-consuming task for service developers.

Accordingly, Fujitsu Laboratories developed a traffic anomaly detection technology that applies outlier structure detection technology[4]. Specifically, this technology is applied to log data obtained by the packet capture function provided as a virtual appliance to detect log data showing behavior different from the usual network operation. **Figure 6** shows the results of the capture data analysis based on the anomaly level

ranking and the statistical information on the GUI. This figure shows, regarding the set (cluster) of packet capture data identified by this technology as having the highest anomaly level, the events actually caught individually at Sites A and B visualized as measurement data.

This enables service developers who lack an understanding of the entire system to prioritize analysis of the logs that lead to impacts on the service without having to scrutinize an enormous amount of data in order to search for specific failure factors. This in turn allows the system to be rebuilt to avoid the resource that may have an anomaly occurring.

5. Conclusion

This paper described technology to support even service developers who have difficulty in grasping the configuration and requirements of entire systems so that they can easily construct and operate secure services in the SoE era. Specifically, it presented NaC technology for quickly building a virtual network across multiple clouds and network verification technology for verifying virtual networks. It also presented traffic prediction and anomaly analysis technology for performing entire processes–from anomaly detection to factor identification–for virtual networks built by NaC.

In the future, we envision the easy provision of appropriate combinations of these technologies depending on the characteristics of customers' businesses, or characteristics including whether latency guarantee in transaction or processing throughput is given priority, and the level of trust required by customers. We intend to develop these technologies as a next-generation service operation platform (suite) utilizing machine learning and other AI technologies. The aim is to make it available in FY 2020.

References

1) Y. Miyanishi et al.: A Proposal for Trust-ability Evaluation Model of Information Systems. IEICE Technical Report, Vol. 116, No. 200, SWIM2016-9, p. 15–22, August 2016. (in Japanese)
2) Istio.
https://istio.io/
3) S. Yamashita et al.: Analytics Framework for AI-based Network Operation and Management. IEICE Society Conference (2017). (in Japanese)
4) K. Maruhashi et al.: Compression-based Discovery of Anomalous Behavior in Large-scale Categorical Data. DEIM Forum (2017). (in Japanese)
https://db-event.jpn.org/deim2017/papers/64.pdf

Naoki Oguchi
Fujitsu Laboratories Ltd.
Dr. Oguchi is currently engaged in research related to SoE system operations in multi-cloud environments.

Hiroshi Tomonaga
Fujitsu Laboratories Ltd.
Mr. Tomonaga is currently engaged in research and development related to virtual resource analysis in multi-cloud environments.

Hitoshi Ueno
Fujitsu Laboratories Ltd.
Mr. Ueno is currently engaged in research related to service quality analysis in multi-cloud environments.

Yasuhiko Aoki
Fujitsu Ltd.
Dr. Aoki is currently engaged in research on network architecture for the beyond 5G/6G era.