Cloud Native SDx Control Technology

Takeshi Hirota, Kenzo Okuda, Masataka Masuda, and Seisho Yasukawa

Abstract

As various services are introduced in the cloud environment, service provision can be simplified and accelerated through end-to-end automatic control of network services and the cloud environment that includes applications for providing services. This article provides an overview of cloud native SDx (software-defined anything) control technology and describes a technical verification of automatic control technology.

Keywords: cloud, automatic control, SDx

1. Carrier network issues

Existing business and social infrastructures will be shifted to mechanisms that assume digitization. Additionally, with the migration to a 5G (fifth-generation) mobile network and the further penetration of cloud services, we can expect the service provision format to be increasingly diversified and an even greater variety of services to be launched, for example, real-time processing for self-driving systems or the utilization of data from various types of sensors. Many of these services will be provided in a cloud environment, so we can envision the need not only for leveraging of cloud features to provide services rapidly but also the need for continuous adding and modification of services. Regarding networks for using services that are provided in a cloud environment, we can foresee that network connection points, quality level, and other factors will have to be changed in an on-demand manner depending on the service.

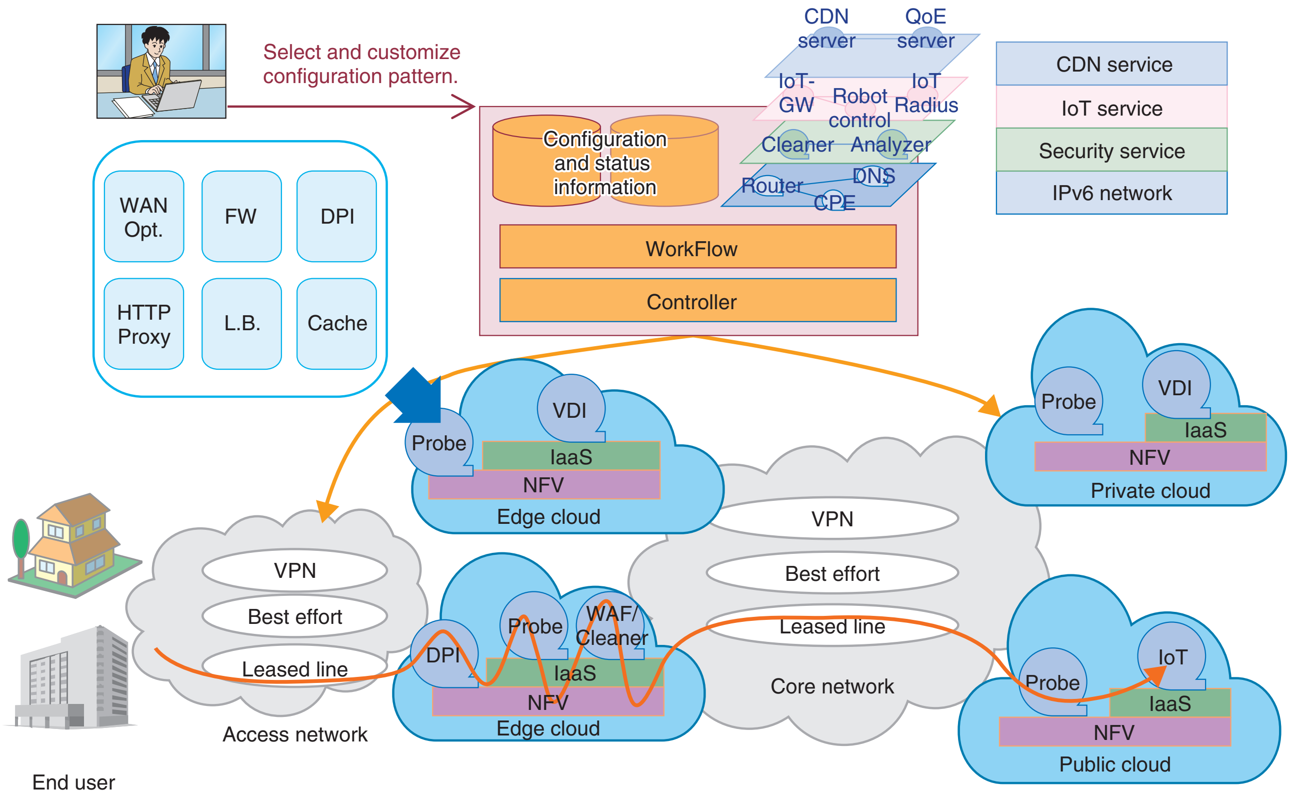
However, network services provided to date have only had a function for connecting the user and service provider, and the inability of the network and the service provision infrastructure on the cloud to sufficiently work together has hindered the rapid provision of services. Additionally, while the quality and reliability of carrier networks themselves have traditionally been high, the ability to control the network from outside the carrier has proved difficult due to various problems including a low degree of freedom, a relatively long time for providing a service or changing settings, and the difficulty of making on-demand changes.

NTT Network Technology Laboratories has been studying cloud native software-defined anything (SDx)* control technology to resolve the above issues. This technology will enable service providers to use diverse functions provided by the network from the outside and to provide more attractive network services than ever. It will also simplify and accelerate the provision of services by enabling the cloud environment and applications for providing services to be collectively and automatically controlled.

2. Cloud native SDx control technology

The cloud native SDx control technology enables service providers to subjectively and uniformly control a network that connects end users to services and the cloud environment that is the infrastructure for providing services and service applications (**Fig. 1**).

* SDx: Generic term for technology that enables software-based control of information technology infrastructure resources (servers, storage, networks, etc.).



CDN: content delivery network
CPE: customer premises equipment
DNS: domain name system
DPI: deep packet inspection
FW: firewall
GW: gateway

HTTP: Hypertext Transfer Protocol
IaaS: infrastructure as a service
IoT: Internet of Things
IPv6: Internet protocol version 6
L.B.: load balancer
NFV: network functions virtualization

QoE: quality of experience
VDI: virtual desktop infrastructure
VPN: virtual private network
WAN Opt.: wide area network optimization
WAF: web application firewall

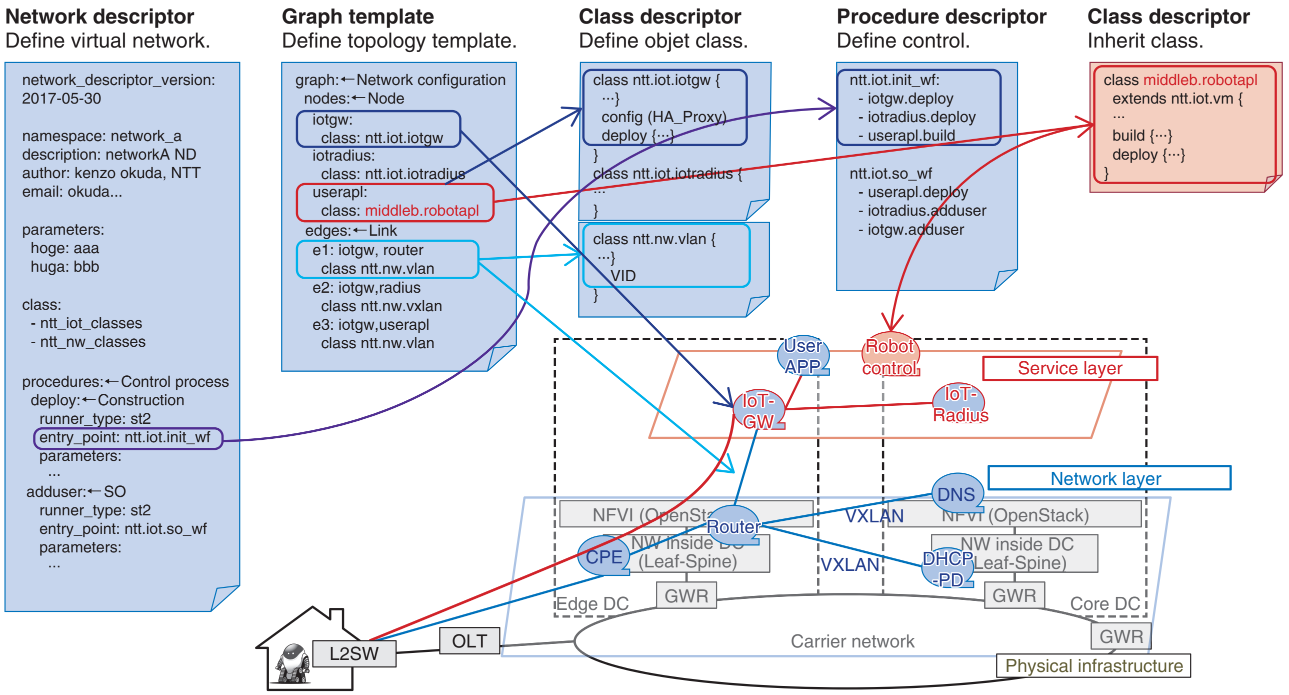
Fig. 1. Cloud native SDx control.

The equipment used in providing network services has generally consisted of dedicated devices, but with the progress being made in softwarization of hardware functions, it is becoming possible to replace those devices with a combination of general-purpose hardware and software. This makes it easier to control the network itself with software. With this change in the environment, and in view of the increasing number of services provided from a cloud, it will be impossible to provide more flexible and prompt services unless a means of integrating control from the network to the application is devised. Cloud native SDx control technology is targeted for all sorts of operators that provide services from a cloud environment, and it is aimed at providing network services that link a common infrastructure with the cloud while automating the immediate provision and maintenance of services.

3. Technical elements for SDx control

Achieving integrated control of a variety of targeted services requires (1) a mechanism for automatically controlling the resources (networks, resources on the cloud, etc.) needed for service provision and (2) a method for appropriately managing the information describing the resources targeted for control.

First, with regard to (1), various technologies already exist for implementing a mechanism for automatic control in a cloud environment. Service providers are already using such technologies, so we can use such cloud-based technologies as a basis for including the network as a target of control. Specifically, we are combining technologies such as network functions virtualization (NFV) and software-defined networking (SDN) and studying a mechanism for handling the network and cloud environment as well as



```
network_descriptor_version:
2017-05-30

namespace: network_a
description: networkA ND
author: kenzo okuda, NTT
email: okuda...

parameters:
  hoge: aaa
  huga: bbb

class:
  - ntt_iot_classes
  - ntt_nw_classes

procedures:←Control process
  deploy:←Construction
    runner_type: st2
    entry_point: ntt.iot.init_wf
    parameters:
      ...
  adduser:←SO
    runner_type: st2
    entry_point: ntt.iot.so_wf
    parameters:
      ...
```

```
graph:←Network configuration
  nodes:←Node
    iotgw:
      class: ntt.iot.iotgw
    iotradius:
      class: ntt.iot.iotradius
    userapl:
      class: middleb.robotapl
  edges:←Link
    e1: iotgw, router
      class ntt.nw.vlan
    e2: iotgw,radius
      class ntt.nw.vxlan
    e3: iotgw,userapl
      class ntt.nw.vlan
```

```
class ntt.iot.iotgw {
  ···}
  config (HA_Proxy)
  deploy {···}
}
class ntt.iot.iotradius {
···
}
```

```
class ntt.nw.vlan {
  ···}
  VID
}
```

```
ntt.iot.init_wf:
  - iotgw.deploy
  - iotradius.deploy
  - userapl.build

ntt.iot.so_wf
  - userapl.deploy
  - iotradius.adduser
  - iotgw.adduser
```

```
class middleb.robotapl
  extends ntt.iot.vm {
  ···
  build {···}
  deploy {···}
}
```

APP: application
DC: datacenter
DHCP-PD: Dynamic Host Configuration Protocol prefix delegation
GWR: gateway router
L2SW: layer 2 switch
NFVI: network functions virtualization infrastructure
NW: network
OLT: optical line terminal
VXLAN: Virtual eXtensible Local Area Network

Fig. 2. Modeling of controlled objects.

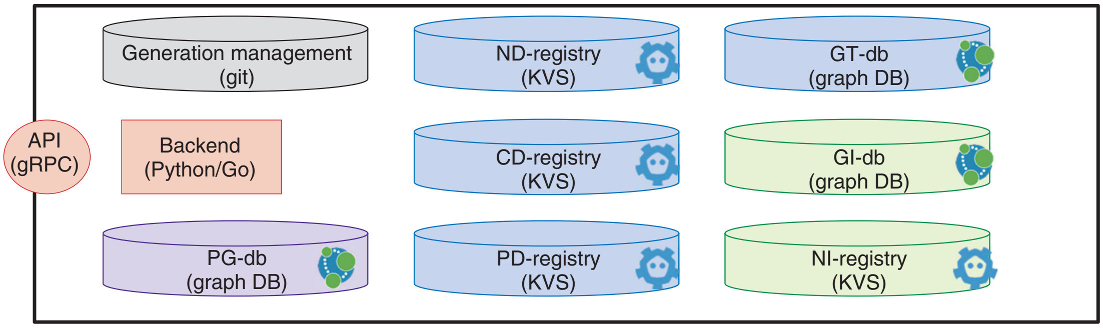
the applications corresponding to different types of services through a series of operations. However, various resources are necessary to provide services, including physical resources (computer, network equipment, etc.) and virtual resources (virtual computers, SDN functions, etc.), and some form of control is needed to link and coordinate them. The controlled objects differ for each service, so creating an automatic-control mechanism for each service would require a great deal of labor.

Consequently, to provide flexible support for diverse services by modeling controlled objects and handling them in a generalized manner, a management method that can uniformly handle resource information as described by requirement (2) above is needed (**Fig. 2**). To build a model, the first step would be to clarify the type of service targeted for control as a network descriptor, and to define what elements (network nodes, computer resources in the cloud environment, etc.) make up the service and the state of each element as a class descriptor. The next step would be to define a graph template that shows how each of those elements are connected, and the order they should be placed in is defined as a procedure descriptor. Because data characteristics differ for each type of information defined, we are also investigating the use of graph databases and key-value store schemes as methods for appropriately managing that information in forms that are easy to handle from the outside (**Fig. 3**).

4. Technical verification

We conducted a technical verification to assess the feasibility of achieving automatic control by combining a variety of technologies used on the cloud. For this verification, we created a scenario assuming specific service-provision scenes. The scenario was based on a service provider configuring a service that provides an application for controlling a robot installed in the user's home and that remotely provides robot control functions.

We assumed the following three processes would be carried out in the provision of this service: (1) infrastructure construction, (2) service configuration and provision based on user request, and (3) detection



API: application programming interface
CD: class descriptor
DB: database
GI: graph instance

git: A version control system.
gRPC: An open source remote procedure call (RPC) system.
GT: graph template
KVS: key-value store

NI: network instance
ND: network descriptor
PD: procedure descriptor
PG: physical graph

Fig. 3. Management of configuration information.

and handling of any unauthorized access occurring during the service. For each of these, we constructed an environment using actual equipment to assess operation.

(1) Infrastructure construction

In constructing an infrastructure for providing this service, we succeeded in automatically deploying on the cloud infrastructure gateways for making connections, an authentication function, and an application for robot control.

(2) Service configuration and provision based on user request

We made it possible to automatically launch the application for robot control, set user information to the authentication function, and enable the user to use the application.

(3) Detection and handling of unauthorized access

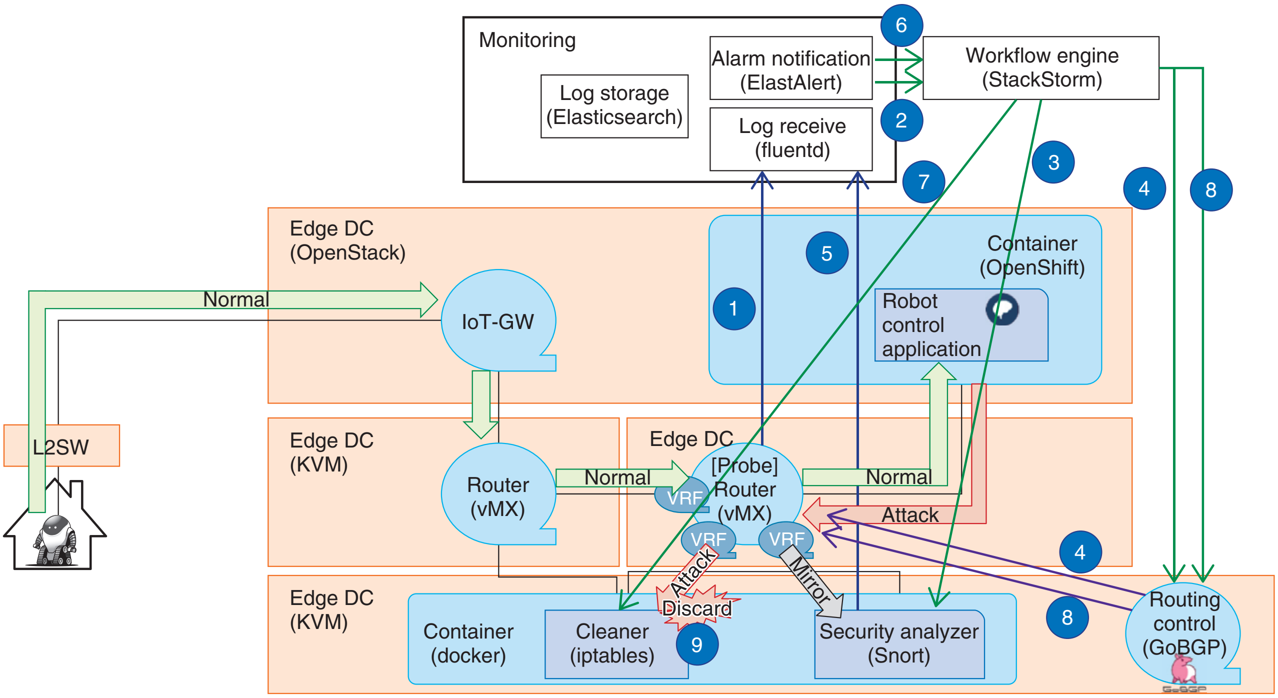
We made it possible when detecting anomaly traffic to automatically launch a function for checking the content of that traffic and to determine whether unauthorized access has occurred.

If unauthorized access is detected, we made it possible to use that detection as a trigger for automatically launching a function for removing the unauthorized access and recovering the system to a normal state. The configuration for technical verification of unauthorized access detection is shown in **Fig. 4**. In the past, infrastructure construction and service provision based on demand would require the service provider to establish various settings and install the application manually. Furthermore, the only way for the person in charge of system monitoring to discover an anomaly would be to notice indications that an unauthorized access was taking place and then collect and analyze various types of information. Responding to such an anomaly would also be centered around manual operations. Such tasks not only require a lot of work but also increase the possibility of human error in operations.

In this technical verification, we confirmed that many tasks from network settings to application installation and anomaly detection and response could be automated by combining various existing mechanisms (most implemented as open source software). Going forward, we aim to achieve efficient automation of diverse types of service provision by incorporating the modeling and configuration management methods that we are currently studying based on those mechanisms.

5. Future outlook

We plan to establish the technologies behind the modeling and configuration management methods now being studied and incorporate them in an automatic control mechanism. In this way, we aim to establish a control infrastructure that can uniformly manage and automatically control all of the resources needed for providing a service.



VRF: virtual routing and forwarding

Fig. 4. Technical verification configuration (unauthorized access detection).

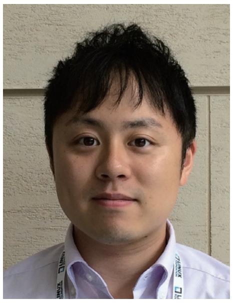
Trademark notes

All brand names, product names, and company names that appear in this article are trademarks or registered trademarks of their respective owners.

Takeshi Hirota

Senior Research Engineer, Network Architecture Design and Promotion Project, NTT Network Technology Laboratories.

He has worked on the development of a network management system for synchronous digital hierarchy, asynchronous transfer mode (ATM), and packet transport technology. He is currently researching automatic unified control from networks to cloud environments. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE).



Kenzo Okuda

Engineer, Network Architecture Design and Promotion Project, NTT Network Technology Laboratories.

He received a B.E and M.E in information and communication engineering from Osaka City University in 2012 and 2014. He joined NTT Network Technology Laboratories in 2014. He has researched future networking architectures, automation technologies for networks, and a network modeling language for telecommunications infrastructure and SDN/NFV. He is a member of IEICE.

Masataka Masuda

Senior Research Engineer, NTT Network Technology Laboratories.

He received a B.E. and M.E. in electrical engineering from the Shibaura Institute of Technology, Tokyo, in 1997 and 1999, and a Ph.D. in engineering from Tokyo University of Agriculture and Technology in 2011. He joined NTT in 1999 and has been working on IP networks and multimedia service quality evaluation and network service operation support architecture. He received the Young Researcher's Award in 2002 and the Technical Committee on Communication Quality's Award in 2004, both from IEICE.

Seisho Yasukawa

Group Leader, Senior Research Engineer, NTT Network Technology Laboratories.

He received a B.E. and M.E. in applied physics from the University of Tokyo in 1993 and 1995. Since joining NTT in 1995, he has conducted research and development of ATM based multimedia switching and NGN (Next Generation Network) architecture. He has also been involved in standardization efforts for P2MP-MPLS (Point-to-Multipoint Multiprotocol Label Switching). His current research includes 5G transport technology.