VX—IIJ's New Backbone Network

2.1 Introduction

In June 2022, IIJ began offering a new network service called IIJ Private Backbone Service/Smart HUB (the "SHB service"), which provides flexible high-capacity connections with cloud services. We built and released a new backbone network, which we (internally) call VX (Virtualization eXchange), to provide the network infrastructure for the SHB service. This article gives an overview of this new VX backbone network and the background to its creation, and details, from various angles, how it differs from previous backbone networks.

2.2 History of the IIJ Backbone

I would like to start by unraveling the history of IIJ's backbone network, which I don't think has been told in much detail to date.

VX is the fourth-generation of network infrastructure within the context of IIJ's backbone network. Let's take a brief look at each generation. The first generation is the Layer 3 IP network ("BB") that has been around since IIJ's early days. BB started with a 192kbps circuit back when IIJ was established, and it has since expanded beyond Japan to now encircle the globe. While originally using 192kbps lines, as of 2022 it mainly runs on 100Gbps broadband, and we are now looking at deploying next-generation 400Gbps. It provides Internet infrastructure for a range of services, starting with IIJ's Internet connectivity services, and continues to expand in that regard.

The first-generation backbone network that is BB reached a turning point in the 2010s. Until then, BB followed a policy of using the same physical and logical topologies, and the network was designed to carry traffic as efficiently and stably as possible over its routers and circuits. Back then, BB took on a square configuration with two core routers each in Tokyo and Osaka, and the leaf nodes in eastern Japan were connected in a V-shaped configuration. Although this network topology may seem efficient at first glance, the bandwidth at the busiest point between Tokyo and Osaka had to be kept constantly at 50% or lower. Also, in the event of a failure or maintenance on one part of the system, all traffic had to be diverted to the other part of the system, and the operational burden and costs involved in doing this were high. So to maximize fault tolerance and traffic-balancing efficiency, we created a mesh network, called a backbone fabric (BF), between the core routers using carrier circuits and our own WDM equipment, making it possible to connect the core routers in an N+1 configuration. At the time, the BF was made up of six routers in

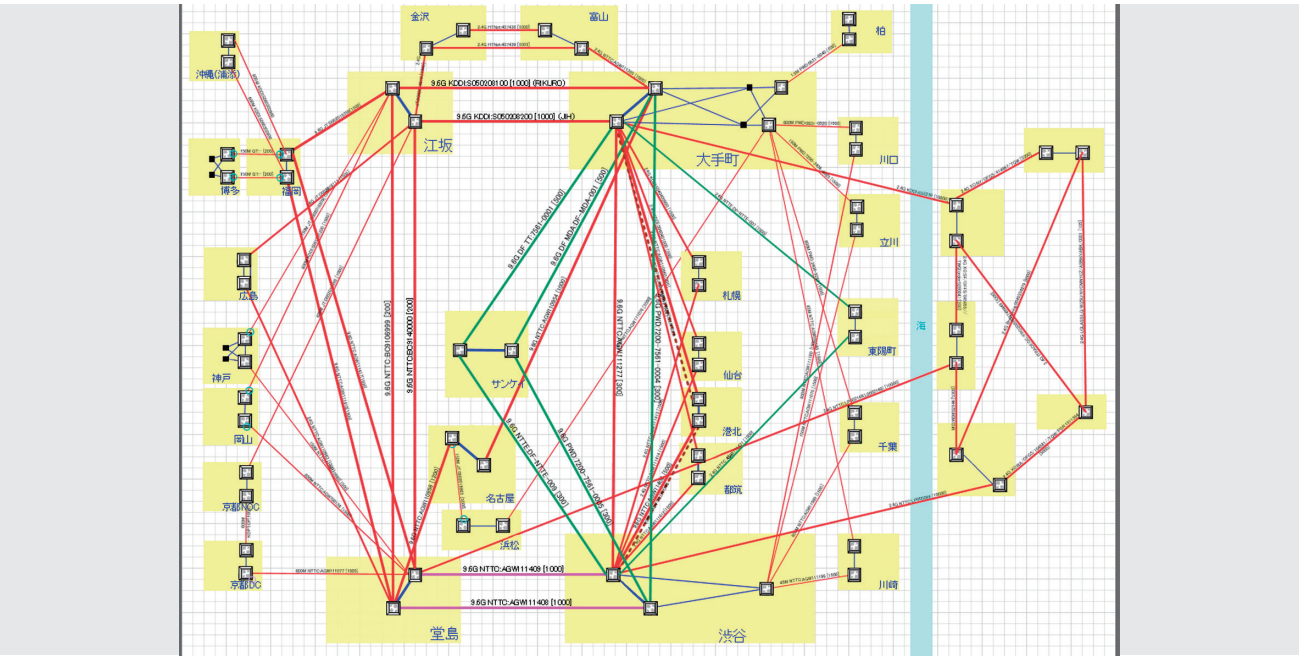

Figure 1: IP Backbone Map Circa 2006 (when Osaka–Tokyo was in a Square Topology)

Tokyo and four in Osaka, and we ultimately expanded it to the US East Coast and West Coast. Using the BF topology drove traffic efficiency on the Tokyo–Osaka leg all the way up and, I think, made it possible to achieve stable communications with minimal impact on traffic relative to how things were with the core routers in a square configuration. We created the BF using 10Gbps Ethernet and 9.6Gbps SONET/SDH circuits, but not being a carrier, IIJ did not have its own carrier network. A disadvantage of not having such a network was the increase in costs posed by circuit usage fees, but, more importantly, the advantages were the ability to freely select circuit operators and to procure from a wider range of circuit path options for the BF topology, which involves using a lot of circuits.

While BF did achieve this ideal, the time came when we started to run up against limits to maintaining this configuration. To make maximal use of BF, the core routers need to be fully meshed with the BF, and a lot of work needed to be done when increasing speed to create the many backbone links. The network topology policy at the time was that the physical topology should match the Layer 3 logical topology. But as we ran up against limits to maintaining this physical topology, we decided to rethink this notion that physical topology should mirror logical topology. This heralded the evolution of the IIJ backbone network into its second generation.

The second generation was a virtual Layer 2 network that we called WARP. The concept with WARP was to network sites in a way that is independent of the physical topology, and to thereby address issues experienced with BF. So, starting with WARP, we began creating logical paths using MPLS label switching technology, something that had not been used on the IIJ backbone network until then. In the BF era, we created Layer 3 load balancing paths along physical circuits, but with WARP, the virtualization of the network between sites meant that we could freely create Layer 2 connections between sites that were not directly physically connected, so we had a greater degree of freedom in terms of topology. WARP facilitated full-mesh connectivity between network nodes physically configured into a BF. As of 2022, we continue to maintain and expand our WARP-based network between real-world sites, and the core BB routers—those in Japan + key routers in the US—are fully meshed in the form of a virtual Layer 2 network. And with OSPF on network Layer 3, we have achieved optimal logical path topology, enabling precise traffic engineering.

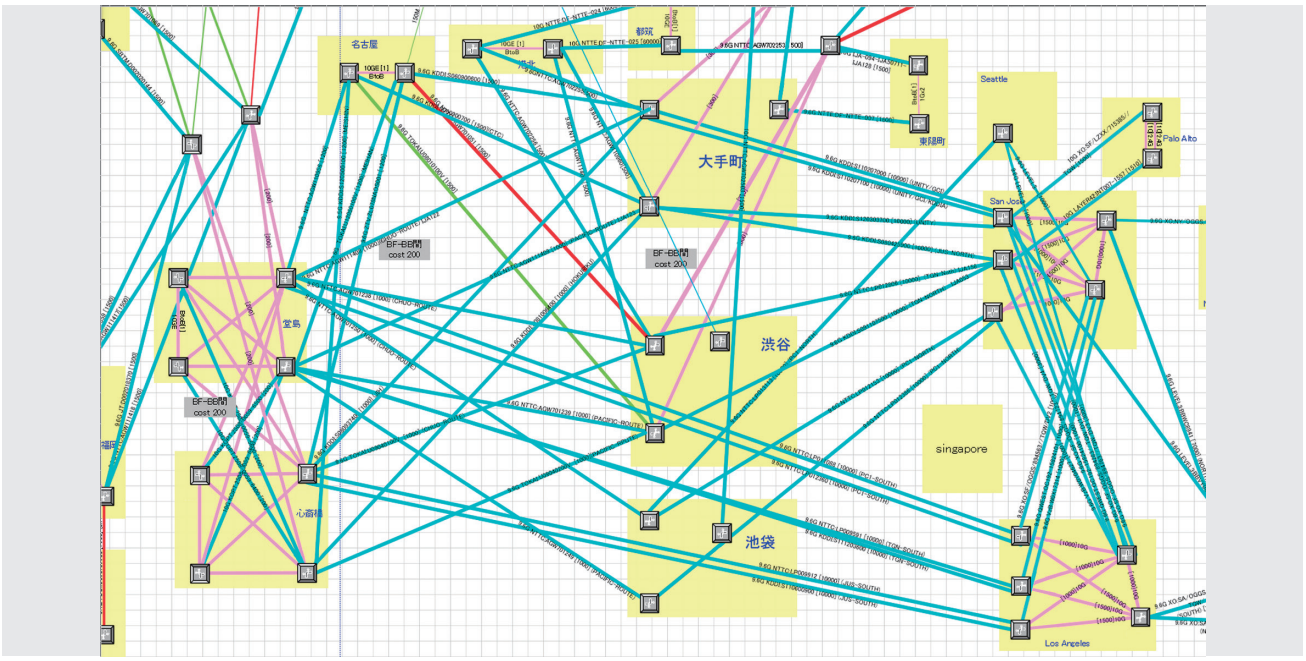

Figure 2: IP Backbone Map Circa 2013 (Backbone Fabric Era)

Our focus with the network up to this point had been on the efficient and stable transmission of Internet traffic generated between IIJ's sites, but with the third-generation backbone network we turned our attention to links between separate service infrastructure. And in the mid-2010s, we began providing our third-generation backbone network service infrastructure, which we called MATRIX. The concept with MATRIX was to provide a wide-area private network connecting multiple points, and to facilitate interconnections with separate service infrastructure networks, which had generally been independent until that point. WARP was a network for providing virtual Layer 2 connections, but MATRIX was a Layer 3 VPN infrastructure connecting different Layer 3 networks via private networks. Before the advent of MATRIX, connections between different Layer 3 networks (aside from those established via private networks set up for that purpose) required each set of service infrastructure to have a global IP address, and network connections between service infrastructure were generally made via the first-generation IP backbone. The issue with setting up a separate closed network is that this would always involve a bit of effort, in terms of preparing multiple WARP circuits and routing traffic through private-edge routers for an Internet VPN. So setting up an independent Layer 3 network as a backbone meant that it was easy for service infrastructure administrators to establish the necessary interconnections between networks without having to worry about private network issues. It may seem quite obvious to say this now that we are well into the heyday of the cloud, but there is a strong need for different private networks and private networks that do not go through the Internet. The flexibility in network connections between different sets of service infrastructure facilitated by MATRIX has bolstered the network linkages between those service infrastructures and made IIJ's services even more flexible. To this day, MATRIX is helping to facilitate the expansion of IIJ's GIO cloud services and services providing private connections with a range of public clouds, facilitating high-quality private network services that benefit many customers.

2.3 The VX Concept and the Introduction of VX Controllers

So far in this discussion, we've spent a bit of time looking back through the IIJ backbone network's generations, seeing how it has evolved along with the needs of the time and to address certain issues. Our current efforts to deploy VX are also aimed at resolving issues with the IIJ backbone network and realizing service concepts required by today's ICT services. IIJ's customers also continue to make use of the cloud, and use of the cloud has gained

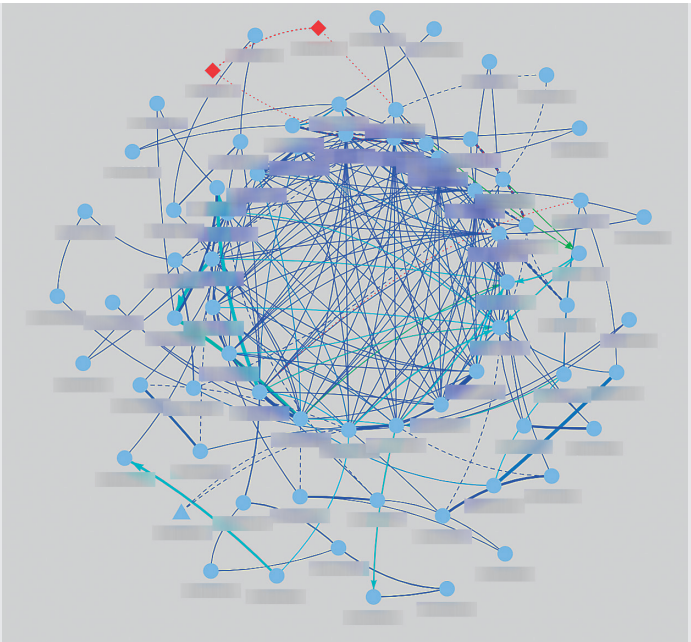

Figure 3: Logical Configuration of the IP Backbone as of 2022
*The meshed part in the center is on the WARP Layer 2 VPN

further steam amid the spread of Covid since 2020. The volume of business system cloud traffic is set to increase sharply under these circumstances, creating the need for greater network bandwidth. And when it comes to obtaining the necessary bandwidth and cloud resources, users are demanding services that give them the flexibility to use just the amount they need at the time they need it, and this is something IIJ's services need to embody as well. However, while we were able to provide service infrastructure offering network stability and security with MATRIX, we still had issues to address: increases in speed just weren't keeping up with customers' requests for more bandwidth, and operational workloads within IIJ were increasing because our operational style was one of handling the processes involved in delivering services to customers manually. In public clouds these days, the network is abstracted, and users have the freedom to use networks as a cloud feature without having to worry about the physical structure, and they can start up the instances they need from the control panel to make instant use of cloud resources. Hence, it is becoming the norm to provide services in a way that matches the sense of speed users expect. So to ensure that IIJ would be able to provide the services that customers demand, we began looking at the prospects for VX as the fourth generation of our backbone network.

VX is aimed at creating the network infrastructure IIJ needs to quickly and flexibly provide a whole range of services to customers. Our thinking was focused on providing a stable, high-capacity network, as discussed earlier, as well as network infrastructure that could meet the demands of service infrastructure when those services are provided to customers in NFV form. To realize the VX concept, in a first for the IIJ backbone network, we adopted SDN control using a network controller. With recent SDN technologies, you can put everything together from scratch using open source, but when it came to building VX, our chosen approach was to make full use of solutions from vendors that we work closely with on a regular basis. For the initial VX infrastructure, we selected Cisco ACI (Application Centric Infrastructure), an SDN solution for data centers from Cisco Systems. Cisco ACI makes it easy to build a network fabric by using an APIC (Application Policy Infrastructure Controller, an SDN controller) to control network configuration with Nexus series Layer 3 switches as the network nodes. Cisco ACI was originally a solution for making it easy to build IP-Clos networks with a basic spine-leaf topology to serve as server networks within data centers, but we use it for more than just data center purposes at IIJ. With customizations, we also use it for networks connecting POPs between multiple points terminating at end users, NVF server infrastructures, and external public clouds.

The introduction of SDN marked a transition in our backbone network operations from an era in which we mainly operated routers through a command-line interface to an era in which our operations are centered on using controllers, in which we use SDN controllers to, for example, configure network settings and monitor the status of entire networks. The biggest change with the introduction of SDN controllers is that we are now able to use APIs to control networks. The Cisco ACI internal settings are abstracted out to make it easy to use not only for network engineers but for application engineers as well, but even so, it's still fairly daunting for users to deal with directly. So with VX, we used the Cisco ACI API and let users define their own models from the ACI settings based on easy-to-use models, and thus abstracted out the structure to make it look simple. Services are provided in such a way that users can establish connections between the necessary points with only the minimum of VX connection elements and parameters. I think abstracting out and simplifying the structure made it easier to think about API links between VX and the service infrastructures, and easier to make effective use of VX in IIJ's service infrastructure as the core network NFV. VX provides an API interface called VX Controller. Up until now, backbone network operators configured the settings needed to connect customers and service infrastructure, but opening the API makes it possible for VX users to enter settings on demand. We believe that making it possible to use the network infrastructure with as little human intervention

as possible has shortened availability lead times and made it easier to think about and deploy IIJ services in a more NFV-like manner. We have actually already started using both VX and the API interface, and we use VX as the network infrastructure for providing the SHB service. By linking the back-end API server of the SHB control panel with VX Controller, we have been able to provide an environment that gives customers on-demand network control on the IIJ backbone network and successfully automated the delivery of internal IIJ services.

Figure 4 shows a sample VX Controller GUI screen. Rather than loading in a config, users can create end-to-end connections simply by entering the necessary parameters in the GUI settings panel. An API interface is also provided, which can be used to configure settings just as they would be via the GUI.

If VX is to serve as flexible, high-capacity NFV infrastructure, the network also needs to be easily expanded and extended. For the high-capacity component, we make full use of the high-port-density, high-capacity nature of Cisco Nexus products. Back in the MATRIX era, we also used high-performance routers designed for carrier applications to build the networks, but certain aspects of such high-performance routers got in the way of any meaningful increase in ports—they invariably had low port density and the price-per-port was high. On this point, as a Layer 3 switch, the Nexus is a network device suited specifically to configuring IP fabrics not based on full Internet routes. Deciding on a narrower set of network use cases meant that the requirements were different from those that applied with the high-performance carrier routers previously used on the IIJ backbone network, and this helped us achieve cheaper, high-capacity communications. By narrowing down the functions, as described above, we successfully reduced the initial costs involved in network expansion to an extent. We also need to expand connection points with VX to scale the network and make it easier to connect to VX. Making it easier to expand the network will also no doubt make it easier for IIJ to deploy service infrastructure and easier for more customers to connect to VX. In terms of creating a flexible NFV platform, the expansion of the network also fits in with the idea of moving the exchange of traffic to locations closer to the user in keeping with today's MEC (Multi-access Edge Computing) approach. MEC requires functionality such as routers, switches, and firewalls to be closer to the user, and IIJ can achieve this by deploying VX and an NFV platform as a service close to the customer. This

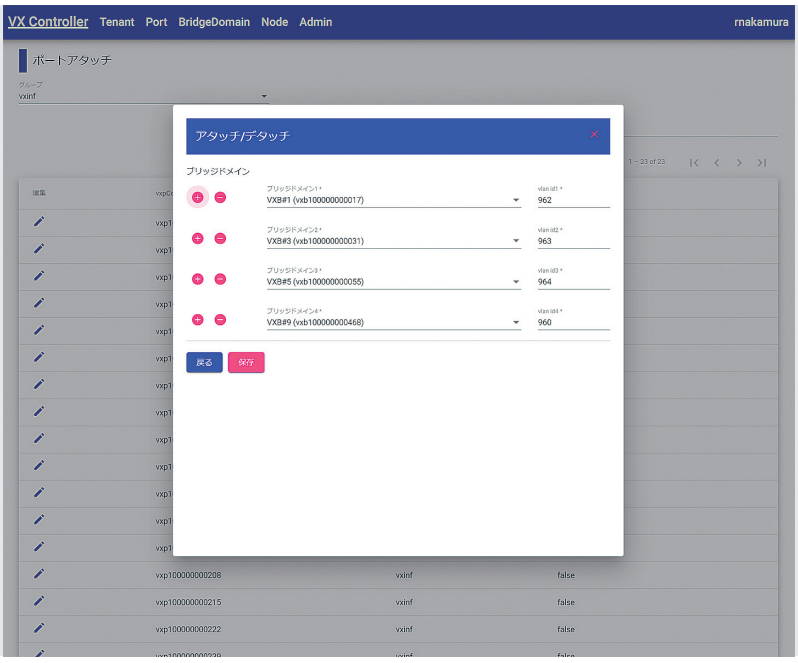

Figure 4: Image of the VX Controller GUI Screen

makes it easier not only to expand physically but also to logically accommodate the infrastructure for services in the form of an NFV platform. We logically divide each set of service infrastructure based on VX and provide it as a tenanted service. In addition to network divisions, limiting the scope of what can be done on each set of service infrastructure makes it possible to accommodate multiple services virtually. So the impact of one tenant's actions do not spill over to other services, and from the VX user's perspective, anything to do with service infrastructure can be thought of in terms of VX. Control of the API mentioned in the previous section is also limited to the tenant that owns the service, so there is no risk of connections being made to unexpected destinations, which makes it easier for us as the VX provider to pass control over to users.

Let's take a brief overview of VX (Figure 5). It has three layers: a layer for controlling the entire system via the SND controller / VX Controller, as mentioned earlier, a layer that interconnects the data centers, and the spine-leaf fabric network accommodating POPs / NFV platforms / public clouds within the data centers. The nodes are basically configured for redundancy, and thus the system is designed such that a single failure will not affect the provision of services. There are six SDN controllers, including one standby unit, deployed across three data centers. To ensure operational stability, at least three of these units must be running at the same time, and the design ensures that service provision will absolutely not be interrupted even if one of the data centers becomes unavailable. When the number of points or the number of services accommodated increases, we scale out the spine-leaf configuration. It is now easier to deploy services since we are able to expand VX connection points using only the minimum equipment necessary.

2.4 Network Monitoring on VX

Equipment and usage monitoring will also be important as the use of NFV platforms on VX accelerates. Until now, we have generally used ICMP echo for alive monitoring and SNMP for acquiring information and receiving traps when monitoring backbone network devices, but with VX, we have added frameworks to enable the monitoring of network quality from the user's perspective, namely metrics for monitoring network status and tools for

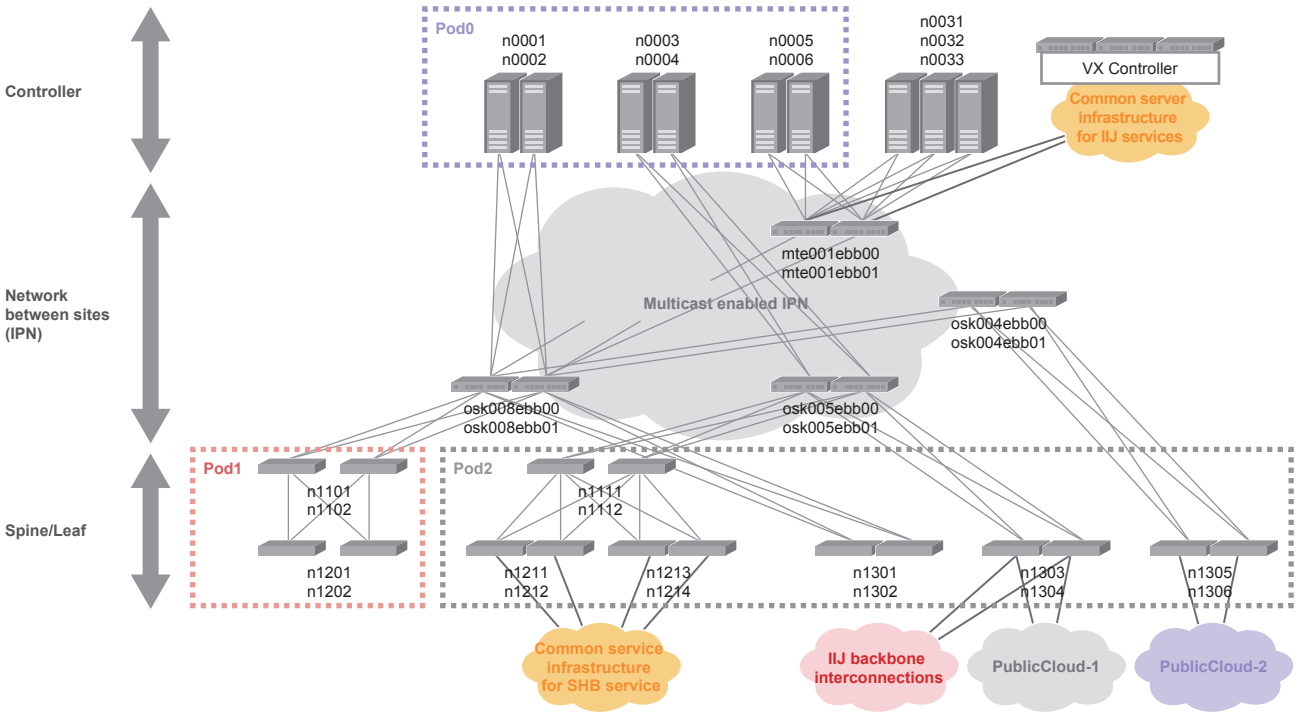

Figure 5: Structural Overview of VX

effectively visualizing the information. To acquire and store our metrics, we use Prometheus, an open-source tool that is starting to see widespread use of late. Cisco ACI also supports monitoring agents that acquire metrics, so it was easy for us to obtain time-series data. And for data that the monitoring agent is unable to acquire, Prometheus also provides exporters that make it easy to create time series in a specified data format for reading into Prometheus, so a whole range of data can be handled within Prometheus. The metrics collected can be visualized in a dashboard format using Grafana, an open source visualization tool, making it easy to check network health, and we have linked this with our own monitoring system so that alerts can be issued when the system detects monitored values falling below a given threshold. In addition to network device health and errors, the monitoring system can also collect data on service capacity, such as network bandwidth usage and connectable interfaces, to produce visualizations, and we are thus using Prometheus + Grafana to automate capacity checks.

Figure 6 is a sample screen from the VX monitoring dashboard that we use. It gives a comprehensive overview of the status of resources and alerts. By convention, network devices typically appear in green when operating normally, and in orange/red when any anomalies are detected.

We find it difficult to ascertain the status of the network IIJ provides to its customers in the same environment that users experience. While we can monitor service-providing equipment such as the routers and switches that make up the backbone network, there are always some things that monitoring of IIJ's equipment alone will miss. In rare cases, we do unfortunately discover faults only after a customer detects an anomaly and contacts us about it. These are known as silent failures, and they cause disruptions to customers' communications despite no device alarms being generated and no anomalies being present in the logs. Silent failures have long been an issue for us network engineers when it comes to providing services. We have been trying to find ways of detecting silent failures before customers do so that we can swiftly restore service availability. As an NFV platform, we envision VX flexibly interconnecting many different sets of service infrastructure, and we thus expect silent failures to have a significant impact. This is why, with VX, we have introduced mechanisms for monitoring communications status under conditions that are as close as possible to those experienced by the infrastructure users. Every VX service edge is connected to a quality monitoring server, and the servers monitor whether communications via the VX service edge are getting through properly. Since this makes it possible to see communications status from the

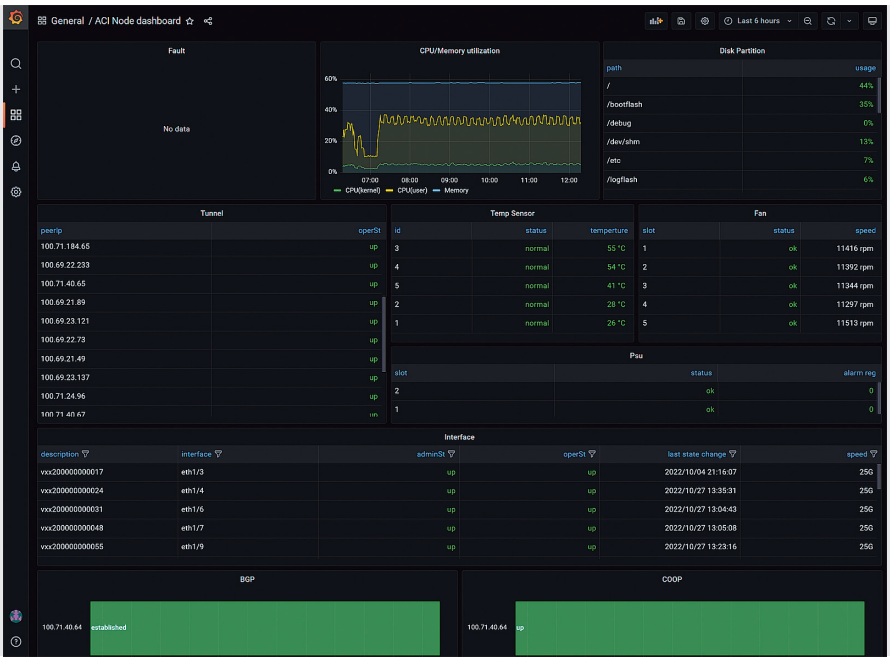

Figure 6: Image of the VX Monitoring Dashboard

same perspective as users, we can tell if any problems are happening even if the service monitoring system does not detect any alerts. The time and effort required to add on a means for monitoring the status of communications between all service nodes increases as the network grows larger. From the VX initial design phase, we also worked on including a means of monitoring the network from the infrastructure user's perspective, and so we were able to put this into action smoothly to coincide with the VX launch.

Figure 7 shows the screen for monitoring the network from the user's perspective, giving a visualization of the status of communications between nodes over time. The red boxes on the screen correspond to when we actually performed network maintenance, and you can see that there was a partial impact on communications. Also, alerts are sent to the operations center when values fall below set thresholds.

2.5 Conclusion

This article has taken a look back at each generation of IIJ's backbone network and discussed our efforts and concepts for the newly released VX. Last but not least, IIJ has built multiple backbone network platforms across generations one to four, but the release of a new network generation does not mean that we will be discontinuing or merging previous generations. Each network has an optimal role to play and functionality to provide, and our approach aims to use each of the backbone networks synergistically so as to optimize the overall system. All of these networks represent infrastructure that is essential for providing IIJ's services. The recently released VX is not intended to replace previous backbone network generations. Instead, we intend to use this new backbone network to link a whole range of networks, NFV platforms, and cloud services to enable IIJ to deliver services that provide value to its customers.

Figure 7: Screen for Monitoring Network Status between VX Service Edges

Yuichi Yomogida
Part of the team at AS2497. Previously involved in running IX services at JPNAP.
Currently working on the IIJ backbone network and serving as peering coordinator.