4. Focused Research (3)

The IIJ Backbone—30 Years of Transformations

4.1 Introduction

The IIJ backbone, which started with just a few routers back in 1993, has now evolved into a large-scale network encompassing thousands of network devices. We have had to deal with a range of issues in the process due to technology and hardware being unable to keep pace with the rapid growth of the Internet—these issues have included communications and routing technologies, the limits of router hardware performance, and power supply issues. Looking back on these days, it was knowledge and ingenuity that got us through and enabled us to provide a stable Internet environment.

In the first half of this article, we discuss the background to and reasons for changes in the IIJ backbone over time as well as the innovations made, with some historical context mixed in. In the second half, we discuss the security measures IIJ has implemented in its network operations.

4.2 IIJ Backbone Through the Years

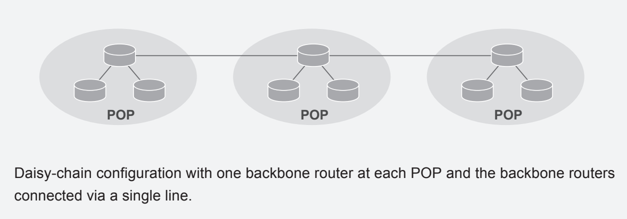
4.2.1 1993–2002: Early Years (Struggling with Resource Shortages)

Back when the Internet was edging toward a transition from academic to commercial use, people's awareness of the Internet was still low, system environments for connecting to the Internet left a lot to be desired[*1], and usage fees were high[*2], so the Internet was mainly being used on a trial basis.

Changes in Physical Configuration in the Early Years

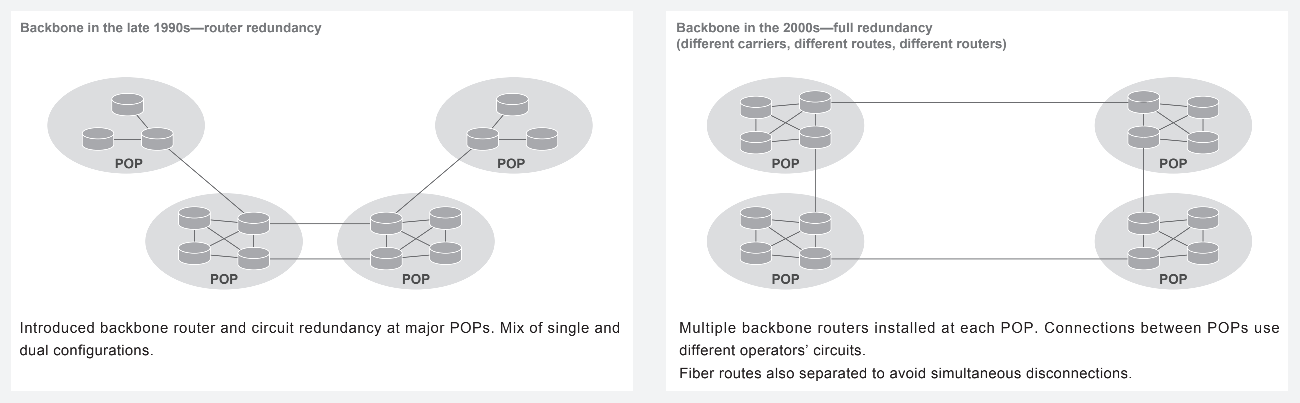
The backbone started with a single configuration, with one backbone router installed for each POP (point of presence) and one dedicated line connecting the POPs in a daisy chain.

The IIJ backbone continued to grow with this single configuration, but once it became more and more common



Daisy-chain configuration with one backbone router at each POP and the backbone routers connected via a single line.

Figure 1: The Initial Backbone


Backbone in the late 1990s—router redundancy
POP
POP
POP
POP
Backbone in the 2000s—full redundancy
(different carriers, different routes, different routers)
POP
POP
POP
POP

Introduced backbone router and circuit redundancy at major POPs. Mix of single and dual configurations.

Multiple backbone routers installed at each POP. Connections between POPs use different operators' circuits.
Fiber routes also separated to avoid simultaneous disconnections.

FIgure 2: Expansion of the Backbone

[*1] The TCP/IP stack became standard in Windows and Mac OSes around 1995.

[*2] Connecting to the Internet via a 45Mbps dedicated line cost 2,000 yen/month.

for companies to use the Internet for financial transactions and the like around 1999, quality demands on the Internet started to become more stringent, and this is when efforts to improve fault tolerance really started to ramp up. We started working on circuit and equipment redundancy at major POPs, and achieved full redundancy around 2002.

An instructive aside

In the winter of 2002, the Fukuoka POP, despite having two backbone circuits, ended up being isolated after both of those circuits were disconnected. This happened because water entered the fiber cables in a section of the fiber route that was common to both circuits, and then froze, causing damage. This lesson taught us to use separate routes for backbone circuits.

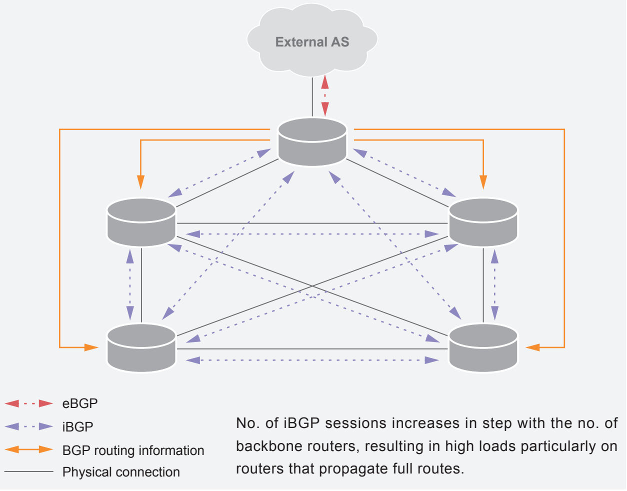
Changes in Routing Control During the Early Years

Since the beginning, we have continued to use BGP[*3] for EGP (Exterior Gateway Protocol) routing and OSPF (Open Shortest Path First) for IGP (Interior Gateway Protocol) routing.

In terms of the routing protocols used to control routes, we use EGP for user network routing information, such as customer and pool addresses, and we use IGP for routing information related to network configuration (devices and links between devices). This is also unchanged since the beginning.

IIJ backbone routing is designed with the idea of achieving a simple but robust network in mind, with the basic policy being to use EGP to propagate information on where networks are and IGP to control communication routes to the target networks. Initially, the number of routers and the total number of routes (full routes) on the Internet were small, BGP was still in development and thus only implemented the bare minimum of functionality, and we basically used a full-mesh iBGP configuration.

As awareness of the Internet began to grow globally, the IIJ backbone continued to expand, with the number of routers and routes increasing. As the number of neighbors and the amount of routing information sent and received increased, router restarts due to maintenance or failures put high demands on memory, and stability issues started to appear—for instance, it would take dozens of minutes or repeated restarts for routes to converge. The overseas routers, in particular, which are responsible for transmitting full routes to all routers in Japan, were coming up against their limits amid latency and the like. In the US, we had a pretty tough time as we went about procuring and increasing memory resources, and eventually we also brought in BGP



No. of iBGP sessions increases in step with the no. of backbone routers, resulting in high loads particularly on routers that propagate full routes.

Figure 3: iBGP Full Mesh Example

[*3] BGP (Border Gateway Protocol): Initially, BGP3 was the mainstream protocol. The implementation of CIDR and the like later led to BGP4 (RFC 1771). We have used BGP4 since beta testing.

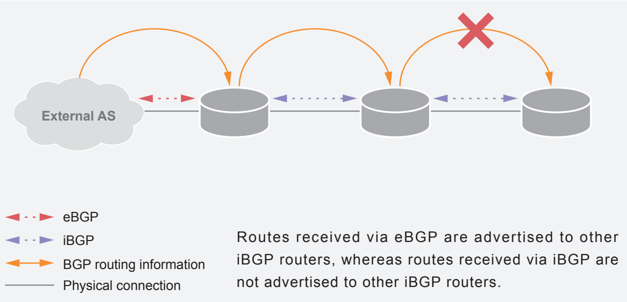
route reflection (RFC1966[*4]) with the aim of reducing loads associated with sending and receiving routing information.

The first thing we did was create a three-cluster configuration spanning East Japan, West Japan, and overseas (Figures 4, 5, and 6).

The use of broadband connections spread and traffic volumes increased substantially from around 2001, and we continued to strengthen and expand the IIJ backbone. It was clear to us that we were fast approaching system limits, so we subdivided the clusters and shifted to a configuration in which we have a cluster at each POP (Figure 7). And more than 15 years on, the backbone is still based on this configuration.

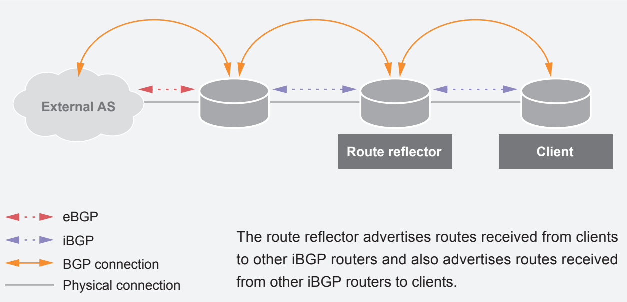
A memorable aside

At that time, the sheer growth in the number of routes was a serious problem. In our chassis-based routers, even our modular interface cards were on the verge of running out of memory, and we had to work late into the night to add memory to hundreds of cards to avert system malfunctions.



Routes received via eBGP are advertised to other iBGP routers, whereas routes received via iBGP are not advertised to other iBGP routers.

Figure 4: Normal BGP Adjacency Diagram



The route reflector advertises routes received from clients to other iBGP routers and also advertises routes received from other iBGP routers to clients.

Figure 5: RR-RC Adjacency Diagram

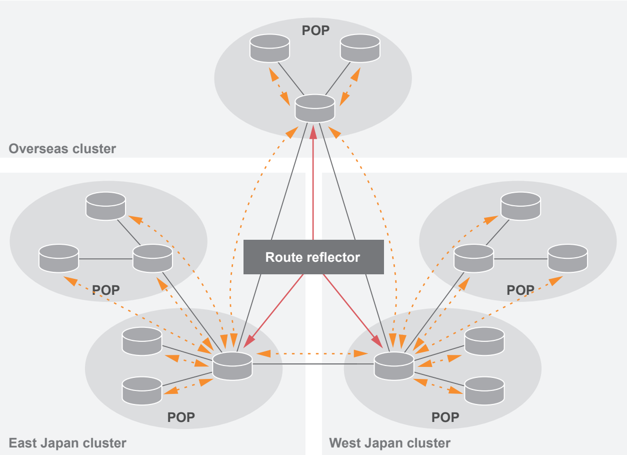


Figure 6: Overview of the Clusters

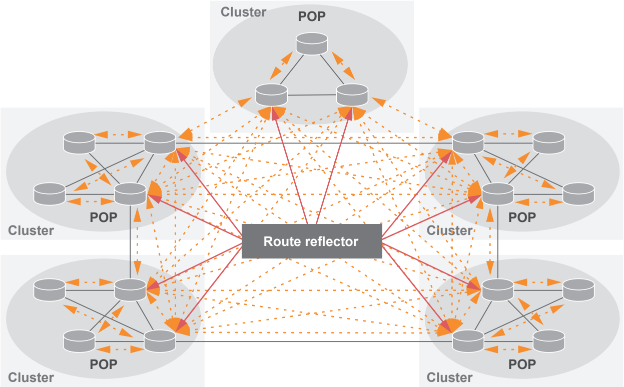


Figure 7: Overview of Cluster Subdivisions

[*4] RFC 1966 BGP Route Reflection—An alternative to full mesh IBGP.

A simple aside

It used to be whispered in the industry that only around 50 routers could coexist in any one OSPF backbone area because of the CPUs being underpowered. In light of this, we split up the OSPF areas on the IIJ backbone too, which dramatically increased the operational difficulty level, and we even experienced a number of accidents because of this. Fortunately, hardware subsequently evolved, and we were able to do away with the area divisions, but this is a prime example of why a simple configuration is best.

The Internet is an amalgamation of networks managed and operated by many different entities, and unintended route hijacking due to misoperations does occur on rare occasions. This can, in some cases, impact the entire Internet, so considerable care must be taken with respect to the routing information sent to and received from other ASes. At IIJ, we introduced route filtering on all edge routers (including within IIJ itself) early on to prevent users from becoming the cause of problems.

Initially, we did this using a combination of access control lists and AS path filtering, but the route filters on all edge routers need to be updated every time you added a new CIDR block, provider-independent address, or the like, so it was very complicated and a lot of work. So we decided to adopt BGP Communities Attribute (RFC 1997). It is quite simple to use. You add a BGP community at route inflow sources and route origins, propagate this inside the backbone, and control route advertising on the edge routers based on the BGP community. This made routing control much easier and greatly reduced the number of settings to be changed, helping to stabilize operations.

To recap, the various technologies were still being developed in these early years, and struggled to keep up with the Internet's growth. This was an era of working to solidify our foundations through trial and error while constantly battling resource shortcomings.

4.2.2 2003–2006: Popularization Era (Rising Quality and the IPv6 Rollout)

As use of the Internet spread, so did demands for quality. We progressively introduced redundancy into the backbone from around 2000, and although lengthy interruptions mostly disappeared, packet losses due to route changes started to become an issue.

Dynamic routing protocols like BGP and OSPF are used for Internet routing control, enabling automatic rerouting of communications in the event of failures. With this sort of dynamic routing, changes in the network are propagated throughout the network as routing information, and each router receiving the information creates its own routing table, ensuring that the entire network can function normally without any inconsistencies. The convergence of state changes resulting from this series of operations is called routing convergence, and the time taken to reach convergence (convergence time) is one measure of network quality and performance.

The state changes leading up to convergence can be roughly divided into the following phases.

- **Event detection (router addition/deletion, link up/down, configuration change, etc.)**
- **Injection into routing protocols**
- **Propagation of routing information**
- **Routing calculations (for each routing protocol)**
- **Incorporation into the routing table**

State changes occur frequently on the Internet due to maintenance and failures. When a state change occurs, convergence needs to be reestablished, and while this is happening, inconsistencies between different routers' routing tables can cause packet losses. The larger the network, the longer convergence times tend to be, and the greater the impact of convergence performance on network quality. So speeding up routing convergence is crucial to achieving a more stable, higher-quality network.

Technologies for speeding up routing convergence were just beginning to emerge at the time (circa 2003). We first decided to study IIJ's backbone performance, measuring it using equipment scheduled to be decommissioned. We checked the results against device debug logs to determine what was taking up time, and we then looked at potential countermeasures. We ended up taking the following three major actions.

- **Router upgrades**
- **Parameter tuning**
- **Switch to topology that makes it easier to detect outages**

It was impossible to tune the various parameters unless the routers were upgraded to the latest OS. We needed just under a year for the backbone routers alone, and several years to complete this on all routers. Alongside this upgrade, we also added IPv6 support. Starting with the upgraded routers, we set about tuning parameters and shifting to a dual-stack network. BFD (Bidirectional Forwarding Detection) was not yet available at the time, so we did the best we could, changing to point-to-point on L2 segments to the extent possible and implementing a topology that would not rely on keep-alive. The upshot of our efforts was that we achieved a convergence time of under a second.

Alongside our efforts to speed up routing convergence, we also worked on the development of a range of systems to improve quality.

- **System for monitoring state changes based on router logs**
- **System for recording routing updates**
- **System for measuring and monitoring packet losses and delays between points**

The sort of quality we achieved is taken for granted today, but it was through these efforts that we achieved it at an early point in time..
To recap, the various technologies were still being developed in these early years, and struggled to keep up with the Internet's growth. This was an era of working to solidify our foundations through trial and error while constantly battling resource shortcomings.

4.2.3 2007–2010: Battling with Traffic (Shift to BF Routers)

With traffic ever increasing, our routers were coming up against their limits, so in line with the design at the time, we considered using OC-768 (40G) as our next connection media after OC-192 (9.6G). Of the routers that met our requirements, only the Cisco CRS-1 supported OC-768. But we quickly gave up on that idea as it had a one-ton floor load and we could not install it. Hence, our only option was to add multiple 10GbE, and so we faced the need to rethink our design given issues with the routers' 10GbE port count and capacity.

Our solution was to use multiple routers to implement the 3-stage switch fabric architecture used to increase the capacity of the CRS-1 backplane, thus creating a giant virtual router out of a "router group" (Figure 8-1).

With this concept, the backbone routers (denoted BF) corresponding to the switch fabric must be connected to all the backbone routers (denoted BB) at the edge. Looking at it from the other end, however, the BB routers need to have as many ports as there are BF routers, but because they handle incoming/outgoing at the edge, you can't use all of the ports to connect to the BF routers. Based on what capacity we would need if the capacity of the router group were to double every year for four years (16 times current traffic), we calculated how many ports could be used to connect to the BF routers out of the BB routers' maximum port count.

Having solved the connection port issue, we then took advantage of the fact that we were using multiple routers and worked on the idea of distributing them across multiple locations, instead of having them all in the same place, to reduce the overall number of routers. We looked at distributing the BF routers across three locations in Tokyo where traffic was heavy (Figure 8-2). The problem with distributing the routers like this is that you need a huge number of 10G lines between sites. Given the unit cost of 10GbE circuits at the time, we surmised there would be a hefty price tag, such that it would be cheaper not to use a distributed deployment. This led us to speak to communications carriers about the number of circuits we envisioned and what the price per circuit would be. We compared this with what the per-circuit cost would be if we were to operate transmission equipment ourselves at scale, and we decided to have our own transmission equipment on some sections. While we had been using simple transmission equipment, full-fledged equipment presented both a high hurdle and a high price tag. But we spent some time approaching manufacturers, testing their equipment, asking questions, and having them explain things. Sensing how earnest we were, one of those manufacturers decided to work with us at the price level we were hoping for, and this ended up being a deciding factor.

With the circuits for our distributed deployment sorted out, we set about designing the circuits between the router group and each of our locations. A conventional 1+1 redundancy design would require a huge number of circuits, so we looked at N+1 redundancy to reduce the cost, but figuring out how to distribute things with an odd number of circuits was extremely difficult, and we couldn't find a good method for this (Figure 8-3). In the end, we gave up on part of the 3-stage fabric concept and decided to implement N+1 redundancy by connecting the BF routers between major locations like Tokyo and Osaka. Because connections run through the BF routers, without a clear picture of which Tokyo/Osaka circuit traffic coming in from the BB routers is using, and what the volume of that traffic is, we would not be able to properly plan for capacity expansions or traffic rerouting during outages and maintenance, and this creates a very difficult problem. NetFlow analysis is the only way to solve this problem, and if that analysis is time consuming, you can't cope with sudden traffic spikes. It happened that right around that time we were developing a system for high-speed analysis of distributed systems[*5], and this helped us avoid creating congestion during outages and maintenance. We designed our system to last four years, but we ended up doubling that as we were able to continue expanding it, without any changes to the design, for eight years.

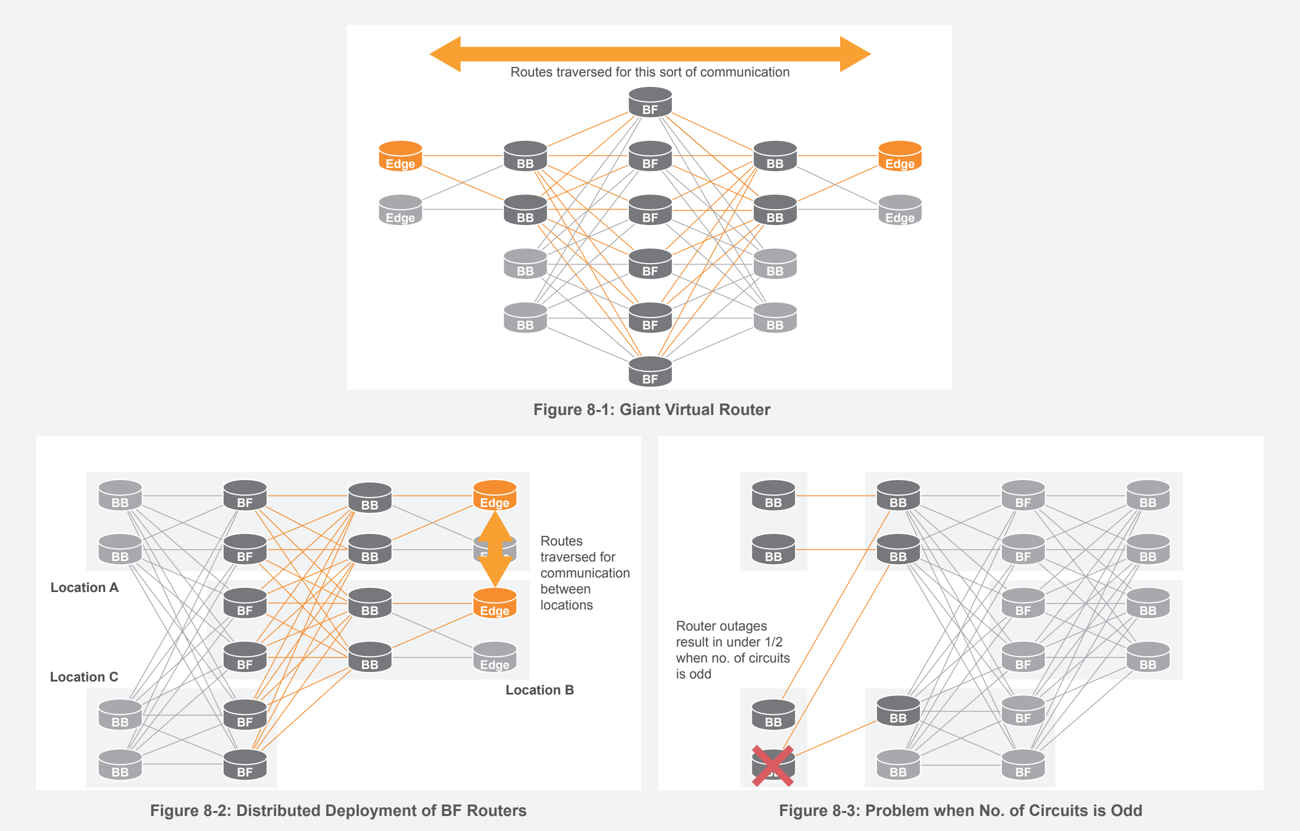

Routes traversed for this sort of communication
BF
Edge
BB
Location A
Location C
Location B
Routes traversed for communication between locations
Router outages result in under 1/2 when no. of circuits is odd

Figure 8-1: Giant Virtual Router

Figure 8-2: Distributed Deployment of BF Routers

Figure 8-3: Problem when No. of Circuits is Odd

Figure 8: Overview of Internet Backbone with Fabric Configuration

[*5] Refer to Chapter 3, Cloud Computing Technology "Implementation and Application of the DDD Distributed System" in IIR Vol. 4 (https://www.iij.ad.jp/en/dev/iir/004.html).

4.2.4 2011 Onward: Network Cloud (Building an Integrated Core and Expanding Private Areas)

Cloud services were going into full swing around this time, with AWS, GCP, Azure, and the rest already on the scene, and IIJ had also released IIJ GIO. Demand for communications between private sites isolated from Internet traffic was growing in conjunction with this, and we used MPLS/L3VPN to expand our private backbone separate from the Internet backbone.

The Internet backbone uses a fabric configuration as described above, and it was designed to be capable of transporting overall traffic as the fabric routers were scaled out, but as we only had 10G media, the more traffic grew, the more operational issues we encountered. For connections within POPs, we used load-balancing methods such as LAG (Link Aggregation) and IGP/BGP multipath, but there are limits to how well you can distribute traffic with traffic flow hashes, and you end up consuming too many ports. You also don't know which links IP packets are flowing through, so it's difficult to confirm that the system is running normally, and thus we were very much looking forward to consolidating everything on 100G.

We actually started using 100G around 2012, opening a 100G connection to JPNAP. Our Internet backbone was already designed with fault tolerance in mind between Tokyo, Nagoya, and Osaka—we used around 20 OC-192 circuits spanning different carriers, different routes, and different locations. Simply switching to 100G with this configuration would be too costly, so we took advantage of the capacity offered by 100G to achieve the following without losing any redundancy in terms of routes or locational distribution.

- **Integrated different locations' backbone network circuits**
- **Eliminated Tokyo/Osaka-dependent structure**

Since IIJ does not have its own fiber, we had to procure carrier circuits for long-distance sections. So that we would not have to obtain separate carrier circuits for each of the multiple network planes, we enabled MPLS/L2VPN pseudowires (PW), allowing bandwidth to be shared by multiple network planes. Further, because maintaining route and locational distribution redundancy for each network would increase the operational effort and costs involved, L2VPN handles most of the route distribution and traffic engineering, and MPLS high-speed rerouting conceals topology changes due to circuit failures and the like, and this configuration makes it easier to control each network. On our high-traffic Internet backbone, we shifted to a simpler configuration whereby core POPs are fully meshed, eliminating transit traffic between core locations.

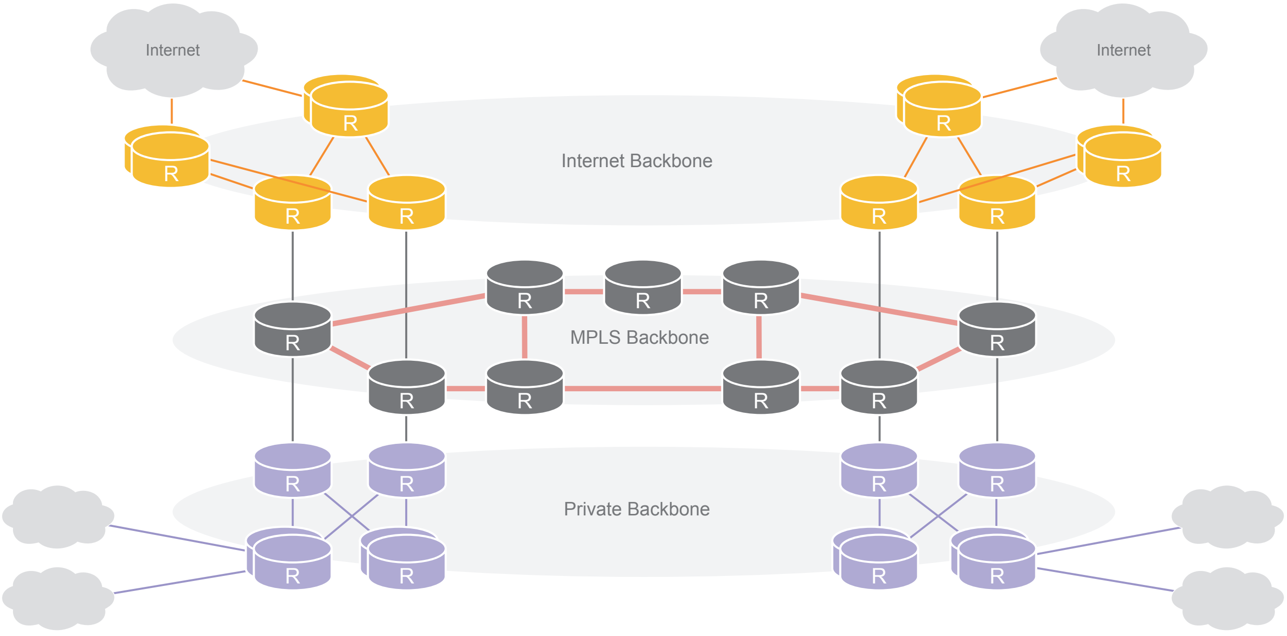


Figure 9: Overview of 100G Core Backbone

Before we eliminated the Tokyo/Osaka-dependent structure, the backbone was configured so that even locations outside the Tokyo/Osaka vicinity were tied to Tokyo or Osaka respectively. That was fine from the perspective of traffic efficiency, but it also meant that communications would go out in those non-central locations in the event of disasters affecting Tokyo or Osaka. We have spent several years addressing this. To increase fault tolerance, we extended our Sapporo and Sendai circuits to Nagoya via non-Kanto (i.e., non-Tokyo) routes, and our Okayama, Hiroshima, Fukuoka, Matsue, and other circuits to Nagoya via non-Osaka routes. We have been working on increasing fault tolerance for several years, and our international circuits between Japan and the US are distributed across Tokyo, Osaka, and Nagoya.

IIJ's network originally only had a location in the US, the central hub of Internet traffic, but alongside these efforts, we also extended our network to other regions—Europe in 2013, and Hong Kong and Singapore in 2014. This means that traffic can now be exchanged directly with Asia and Europe, so we are not reliant solely on the US for international connectivity, and this has improved Internet connectivity.

Implementing an integrated 100G backbone like this meant we could also smoothly expand our private backbone, which is small in comparison with Internet traffic. We have progressively expanded as a cloud exchange to facilitate interconnectivity with public clouds, and we have expanded as a network cloud to meet the needs of today's increasingly diverse workstyles.

We began this section with the IIJ backbone's early years, following its history up to the present and looking at the repeated improvements and changes made over the years to address the prevailing issues of the day. The Internet is an integral part of society's infrastructure, and people will no doubt demand even greater levels of reliability going forward. IIJ will continue to expand its systems to provide reliable social infrastructure that serves the needs of people everywhere.

4.3 IIJ's Network Security Measures

IIJ has also worked to improve security to ensure its networks can be used appropriately. In this section we look at some of the measures IIJ has taken with respect to the security of network operations.

4.3.1 Source Address Validation

On the Internet, routing information for delivering IP packets to their destination is basically searched for based on the destination IP address given in the IP packet header. The IP packet header also contains information on the source IP address, but the IP packets will be delivered to the destination even if this information is incorrect. Upon receiving the IP packet, the destination determines where it came from by looking at the source IP address in the IP packet header and, if necessary, sends out a response packet. If the source IP address is wrong, the system will still take the incorrect IP address information to be true and send the response packet to entirely the wrong host. This behavior is exploited by malicious attackers, and a variety of attack methods have been devised and used in real-world attacks. Attackers can use these methods, for example, to make it difficult to identify the source of an attack, to hijack communications by spoofing another host, or to have response packets sent to a specific host.

DNS reflection attacks (DNS amplification attacks) exploiting the DNS system have been observed since around 2005. These attacks involve spoofing the source IP address to be the IP address of the attack target and sending DNS queries to nameservers as a stepping stone. The nameservers respond, with the name resolution resulting in an increased amount of data, efficiently exhausting the attack target's bandwidth, the aim being to disable service. Attackers hijack Internet-connected hosts in advance and then carry out such attacks by sending packets with spoofed source IP addresses from those hosts. To ensure that IIJ's connectivity services are not exploited in such attacks, we decided to introduce technology that prevents source IP address spoofing.

Problems associated with IP spoofing were recognized early on, with certain problems and countermeasures being documented in RFC 2827 (BCP38) and RFC 3704 (BCP84). To combat this, you need to verify whether an appropriate source IP address is used as close as possible to where the connection service is terminated. The methods of source address validation available on the equipment IIJ was using at the time were unicast reverse path forwarding (uRPF), which uses a route search mechanism, and access control lists (ACLs), which use packet filtering. We implemented these as appropriate given the functional limitations of the different equipment models and software versions. In March 2006, we announced[*6] a rollout of sender verification on all connection services, which we subsequently completed. This has prevented IIJ's connection services from being exploited in attacks, improving the security and facilitating the stable operation of the overall network.

4.3.2 Internet Routing Registry (IRR)

With a variety of different networks connecting to the Internet and expanding, one major consideration is how to go about coordinating BGP routing control policies among networks. To address this, the IRR publishes routing policies as objects and provides functionality allowing network operators to query each other's policies. Objects registered in the IRR database can be used to automatically generate route filters, perform checks when failures occur, etc. At IIJ, we register the main types of objects commonly looked up in the IRR database—such as route, route6, and as-set—and keep the information up to date. We have been using Merit RADb, a service run by Merit since the 1990s, to register our IRR routing objects. Alongside this, since 2005 we have also used the JPIRR operated by JPNIC, and at present, we mainly use these two IRRs.

If objects registered in the IRR database are rewritten without permission, other networks that look those objects up could generate the incorrect route filters, which could cause reachability problems for IIJ. With both Merit RADb and JPIRR, objects are basically updated by sending emails to the administration system. There are several authentication options available when doing this. At IIJ, we use the strongest one: Pretty Good Privacy (PGP). This sort of authentication involves verifying PGP digital signatures. To use it, you first register a PGP public key with object modification permission in the IRR database. IIJ completed the transition to PGP authentication in 2003.

4.3.3 Resource Public Key Infrastructure (RPKI)

RPKI is a public key infrastructure for certifying the distribution of number resources such as IP addresses and AS numbers, and using RPKI can help improve routing control security. An organization that receives an IP address can issue a Route Origination Authorization (ROA) from the RPKI system to indicate which AS should be advertising the network. Using this information, it is possible to verify whether route information received via BGP was generated by a valid origin AS. IRR is more widely used at present, but RPKI is better for automation and can use more reliable information, so we expect the use of RPKI to spread.

IIJ completed the issuance of basic ROAs in 2020. The ROAs include a maximum prefix length indicating the extent to which the AS can split routes it advertises, but since IIJ does not split advertisements, we leave the prefix length of advertised routes as is. This is also what is recommended in RFC 9319 (BCP185). Also in 2020, we introduced a policy for using ROA information to verify BGP route advertisements received from peers and upstream and discarding route information that is inconsistent with the ROAs. On the IIJ network, this makes it possible to identify and discard routes even if the ROA-issuing network receives a route advertisement from an incorrect origin AS. Merit RADb also implements a feature that automatically removes objects that are inconsistent with ROAs,

[*6] IIJ to Roll Out Source Address Validation on All Connection Services (https://www.iij.ad.jp/news/pressrelease/2006/pdf/0308.pdf, in Japanese).

so issuing ROAs is also a way to avoid registering incorrect objects in the IRR database.

4.3.4 Mutually Agreed Norms for Routing Security (MANRS)

Network security measures used in the operation of the Internet become more useful when a large number of network operators adopt them in concert with one another. MANRS is a voluntary global initiative promoting the introduction of such measures. With support from the Internet Society (ISOC), MANRS sets out recommended security measures (actions) for different areas, which it asks organizations involved in the Internet's operation to put into practice. Organizations that approve of this approach can become a participant by informing MANRS of the actions they have implemented.

IIJ appropriately implements security measures suited to its operations. These are consistent with the practices recommended by MANRS, and IIJ joined MANRS in 2015 as the first participant from Japan[*7]. Looking ahead, IIJ will continue to review its operations and continuously make improvements to ensure the stable operation of the Internet.

1993–2002: Early Years (Struggling with Resource Shortages)
Toshio Iwasaki
Manager, Operation Technology Department, Infrastructure Engineering Division, IIJ

2003–2006: Popularization Era (Rising Quality and the IPv6 Rollout)
Yoshio Asano
Infrastructure Technology Department, Network Division, IIJ

2007–2010: Battling with Traffic (Shift to BF Routers)
Kunio Kataoka
Infrastructure Engineering Division, IIJ

2011 Onward: Network Cloud (Building an Integrated Core and Expanding Private Areas)
Fumiaki Tsutsuji
Network Planning Manager, Network Technology Department, Infrastructure Engineering Division, IIJ

IIJ's Network Security Measures
Yoshinobu Matsuzaki
Technology Development Section, Operation Technology Department, Infrastructure Engineering Division, IIJ

[*7] MANRS Turns 1 and First Japanese Operator, IIJ, Joins (https://www.manrs.org/2015/11/manrs-turns-1-and-first-japanese-operator-iij-joins/).