POLITECNICO DI TORINO

Master's Degree in Computer Engineering

Master's Degree Thesis

Designing a Plug&Play interconnection system to connect different datacenters with pure L2 links

Supervisors

Prof. Fulvio RISSO

Candidate

Luca Giuseppe ROCCO

Academic year 2022-2023

Abstract

In the last decade, the field of cloud computing has experienced rapid growth, witnessing the introduction of numerous innovative services each year. However, the market is currently dominated by a few major players who continue to consolidate their power. This dominance poses challenges, including the difficulty of countering vendor lock-in and the limited competitiveness of smaller cloud service providers. To address this issue, there are initiatives at the European level aimed at establishing an alternative or complementary cloud solution that empowers minor providers to reach a wider customer base.

The concept revolves around the creation of a federated cloud and network infrastructure through a distributed approach, involving European Internet Exchange Points and Cloud Service Providers. This approach brings forth both political and technical challenges. While it is important to acknowledge the broader political context, this thesis focuses primarily on the technical aspects.

Within this federated infrastructure initiative, a significant number of members are actively working towards the development of an initial, simplified solution to validate the concept of a distributed cloud and network infrastructure across Europe. The first milestone in this journey was reached at the end of 2022, with the presentation of a proof-of-concept for the Structura-X Lighthouse project at the Gaia-X summit. The Structura-X PoC provided valuable insights into possible infrastructural solutions, highlighting the need for a novel approach to cross-network automation that can enhance the speed and reliability of processes.

Within the realm of automation, there exist multiple layers, each serving a specific purpose. This thesis will primarily delve into the layer 3 automation problem, which revolves around addressing cross-network layer 3 reachability challenges. By thoroughly examining this issue, the thesis aims to propose an engineering solution that can effectively mitigate the challenges associated with achieving seamless layer 3 connectivity across a distributed cloud and network infrastructure. The automated layer 3 negotiation enables a scalable and rapid setup, replacing inefficient and time-consuming manual processes.

Layer 3 connectivity plays a crucial role in enabling seamless communication between different networks, as more than 90% of cloud services rely on layer 3 protocols. The proposed solution outlined in this thesis strives to offer the necessary flexibility to accommodate diverse network topologies among participating partners. This fully functional project serves as a practical demonstration of the solution's viability and practicality in real-world scenarios.

The final solution comprises two components, each serving a specific purpose. One key aspect is the implementation of a fixed and standardised interface that

triggers the negotiation process. This interface provides a consistent and predictable means of initiating the negotiation procedure across different deployments.

Furthermore, a highly customisable component is incorporated into the solution, responsible for handling the network configuration based on specific details of the network topology. Thanks to this component, the solution offers high adaptability and can be applied to various network topologies.

This thesis focuses on designing, implementing, and testing components for the layer 3 negotiation process. The solution will be part of a larger federated network infrastructure and efficient cloud ecosystem.

Table of Contents

List of Figures

Acronyms

IXP

Internet Exchange Point

CSP

Cloud Service Provider

L3

Layer 3

L2

Layer 2

BGP

Border Gateway Protocol

SC

Service Composer

API

Application Programming Interface

IXApi

Internet Exchange Application Programming Interface

PoC

Proof of Concept

NaaS

Network as a Service

VM

Virtual Machines

SaaS

Software as a Service

CLI

Command Line Interface

REST

Representational state transfer

HTTP

Hypertext Transfer Protocol

HTTPS

HTTP Secure

TLS

Transport Layer Security

gRPC

Google Remote Procedure Call

IP

Internet Protocol

NAT

Network Address Translation

BC

Broadcast

VRT

Virtual Routing Table

VPN

Virtual Private Network

SD-WAN

Software Designed - Wide Area Network

GCP

Google Cloud Platform

Chapter 1

Introduction

Cloud computing is an increasingly vital aspect of technology, with a multitude of cloud service providers operating worldwide. This advancement enables the creation of affordable and dependable services in less time and with less effort compared to traditional on-premises solutions. However, despite the wide range of CSPs available, a select few dominate the market.

In Europe, collaborative initiatives are underway to establish a new cloud ecosystem by uniting multiple small European CSPs. These companies are working together to develop a robust solution that embraces this innovative approach. This thesis aims to thoroughly explore key aspects of this collaborative endeavor, with a particular emphasis on addressing networking challenges.

Currently, three major cloud service providers dominate about 65% of the cloud services market. These companies hold significant control due to two main reasons. Firstly, they have complete control over the network infrastructure, allowing them to ensure high levels of security, availability, and performance. Secondly, they offer a wide range of services, making it convenient for businesses to work with a single provider.

However, these major cloud service providers operate in a closed system, without much compatibility between themselves or other cloud providers. As a result, once a business chooses a provider, they are usually tied to that provider unless they decide to move their entire infrastructure to a different provider. This is known as "vendor lock-in."

This thesis aims to propose an innovative solution by collaborating with three main entities:

- Small and medium-sized cloud service providers
- European Internet exchange points
- Telecommunication companies (Telcos)

By working together, these entities can establish a distributed and cooperative network and cloud infrastructure.

The main challenge in this distributed scenario is to create a strong underlying network. To address this challenge is needed an innovative approach based on cross-network automation which is essential for providing a scalable and reliable service.

The research acknowledges the importance of cross-network reachability and aims to propose solutions that can create a large scale networking system by connecting different CSPs networks. By doing so, the thesis aims to foster collaboration and innovation among the participating entities, contributing to the development of a distributed network model that offers a compelling alternative to the dominance of the major cloud service providers.

In this context, the thesis particularly focuses on Internet Exchange Points (IXPs) and addresses the challenges associated with automating communication between different networks. By tackling these challenges and finding ways to enable seamless cross-network reachability.

The complete solution consists of three main components:

- **Layer 2 automation**: This component focuses on the creation of on-demand layer 2 virtual links. It enables the establishment of virtual connections between different networks, allowing for efficient and flexible network connectivity.
- **Layer 3 automation**: The layer 3 automation component builds upon the layer 2 virtual links by creating a layer 3 network infrastructure. This infrastructure enables seamless communication and reachability between different networks. By establishing layer 3 connectivity on top of the layer 2 links, disparate networks can effectively communicate with each other.
- **Federated service catalog**: The federated service catalog encompasses a collection of cloud services provided by various cloud service providers within the federation. Leveraging the network federation, these cloud service providers collaborate and utilize the underlying infrastructure to offer their services to users. This federated approach expands the service offerings and allows clients to access a broader range of cloud services through a unified platform.

By integrating layer 2 and layer 3 automation techniques, along with a federated service catalog, the proposed solution aims to create a distributed and cooperative network and cloud infrastructure.

This thesis will discuss deeply the *layer 3 automation* and it is organized as follows:

- **IXPs Federation as an alternative to the Hyperscalers**: Introduces a new approach of building a distributed network. The network is particularly important because is the basement of the whole cloud infrastructure.
- **A new Network Fabric architecture**: To create a new network fabric some new components are required. This chapter explains all the components and their relationships
- **The L3 Negotiation Problem**: One of the problems that raises when we want to deal with a distributed network is the layer 3 reachability. The problem is explained along all the other challenges that it brings.
- **The proposed solution**: The proposed solution is the result of a Proof of Concept (PoC) that was initiated at the end of 2022. The PoC served as the starting point for developing and refining the solution. The experiences

and insights gained from the PoC were used as a foundation for elaborating, creating, and implementing the solution. In this chapter of the thesis, a detailed demonstration of the solution is provided. This demonstration showcases how the solution functions and highlights the reasons why it is recommended for adoption by the participating entities.

Chapter 2

Context and Current Situation

2.1 Importance of Cloud Computing and Networks Today

Cloud computing and networks play crucial roles in today's technology landscape for several reasons[1]:

1. **Scalability and Flexibility**: Cloud computing allows businesses to scale their resources up or down according to their needs. It provides the flexibility to quickly adjust computing power, storage, and network resources based on demand. This scalability enables businesses to efficiently handle fluctuating workloads, accommodate growth, and adapt to changing market conditions.

2. **Cost Efficiency**: Cloud computing offers cost advantages by eliminating the need for upfront hardware investments and reducing maintenance and operational expenses. With cloud services, businesses can pay for the resources they consume on a pay-as-you-go basis. This pay-per-use model allows for optimal resource allocation, minimizing wasted capacity and maximizing cost efficiency.

3. **Accessibility and Remote Work**: Cloud computing enables remote access to applications and data from anywhere with an internet connection. This accessibility is particularly valuable in today's era of remote work and distributed teams. Cloud-based collaboration tools and virtual desktops facilitate seamless communication, collaboration, and productivity across geographically dispersed teams.

4. **Data Storage and Backup**: Cloud storage solutions provide secure and reliable data storage, eliminating the need for on-premises storage infrastructure. Cloud-based backups and disaster recovery services ensure data resilience and business continuity by offering automated backups, redundant storage, and geographically dispersed data centers.

5. **Global Reach**: Cloud providers maintain a global network of data centers, allowing businesses to deploy applications and services closer to their customers worldwide. This distributed infrastructure improves performance, reduces latency, and enhances the user experience by ensuring fast and responsive access to applications and content.

6. **Security and Compliance**: Cloud providers invest heavily in robust security measures to protect customer data and infrastructure. They employ advanced encryption, access controls, and monitoring systems to safeguard against cyber threats. Additionally, cloud services often comply with industry-specific

regulations, helping businesses meet their security and compliance requirements more easily.

7. **Innovation and Time-to-Market**: Cloud computing provides a platform for rapid innovation and faster time-to-market. By leveraging cloud services, businesses can quickly experiment, develop, and deploy applications, services, and solutions. The cloud's agility and ready-to-use infrastructure enable organizations to focus on innovation rather than managing underlying IT infrastructure.

In summary, cloud computing and networks are important today due to their scalability, cost efficiency, accessibility, data storage capabilities, computing power, global reach, security features, and ability to drive innovation. They empower businesses with the tools and resources necessary to thrive in the digital era.

2.2 Hyperscalers

Understanding what hyperscalers offer to clients and their market dominance is essential. These cloud service providers possess distinct qualities that set them apart and contribute to their formidable position. Hyperscalers leverage their unique advantages to deliver exceptional services and capture a significant market share. With their unwavering reliability, commitment to superior quality of service, optimized operations, and cost efficiencies, hyperscalers become the preferred choice for clients seeking top-tier cloud solutions.

The immense power of hyperscalers in the market is strongly tied to their unparalleled control over the network infrastructure. By having full control over this critical foundation, including services, servers, virtualization, and other components, hyperscalers gain a significant competitive advantage.

One of the key advantages resulting from this control is the ability to ensure unmatched reliability in their services. Hyperscalers can implement robust redundancy mechanisms, fault-tolerant architectures, and comprehensive disaster recovery strategies. This level of reliability instills confidence in clients, as they can rely on uninterrupted access to their critical applications and data.

Moreover, hyperscalers can deliver a superior quality of service due to their network control. They have the capability to optimize network performance, reduce latency, and ensure high throughput. This allows them to provide consistently excellent user experiences, especially for applications that demand low latency, high bandwidth, or real-time interactions.

The optimization of network topology is another significant benefit stemming from this control. Hyperscalers can design and implement highly efficient network architectures, leveraging their deep understanding of traffic patterns, data flows, and resource utilization. Such optimized topologies enable faster data transfers, efficient load balancing, and improved scalability, all of which contribute to superior performance and cost-effectiveness.

In addition, hyperscalers often leverage their network control to offer cost savings for end users. The largest hyperscalers typically waive network traffic charges when data moves between services within their own infrastructure. This eliminates or significantly reduces the costs associated with inter-service communication, making their services more affordable for clients and fostering a more cost-effective environment for businesses.

Overall, the comprehensive control over the network infrastructure empowers hyperscalers to dominate the market. It allows them to deliver robust, high-quality services, optimize network performance, and offer cost advantages that smaller cloud service providers struggle to match. This market power reinforces the position of hyperscalers as the go-to choice for clients seeking top-notch cloud solutions.

2.3 Attempts of creation of an alternative solution

Different solutions have been proposed at the European level and some of them are currently under development. Two projects that are worth to be mentioned are GAIA-X and one of its side projects, Structura-X.

2.3.1 GAIA-X and its Clearing Houses

GAIA-X is an initiative that aims to establish a secure and sovereign European data infrastructure. It is a project driven by European companies, research institutions, and governments with the goal of creating a federated and decentralized data ecosystem. The initiative seeks to address the challenges related to data sovereignty, privacy, and secure data sharing by developing a trusted and interoperable platform for data exchange.

The concept of clearing houses, in the context of GAIA-X, refers to intermediaries or platforms that facilitate the exchange and sharing of data among different participants within the GAIA-X ecosystem. These clearing houses serve as trusted intermediaries that provide services to enable secure data transactions and ensure compliance with data protection regulations.

Clearing houses in GAIA-X play a vital role in managing access rights, permissions, and data governance. They provide mechanisms for data owners to define policies and permissions regarding data usage, access, and sharing. Clearing houses may also offer services related to data discovery, data quality assurance, and compliance with regulatory frameworks.

The concept of clearing houses is intended to create a trustworthy and transparent environment where data can be shared and utilized by multiple entities while maintaining data sovereignty and compliance with privacy regulations[2].

2.3.2 Structura-X

Structura-X was born as a side project of GAIA-X and it aims at the creation of an underlying federated network infrastructure that the GAIA-X framework can rely on. While GAIA-X tries to create a data infrastructure the idea of running the GAIA-X framework on the public network is not accepted because there are no control or guarantees.

Chapter 3

IXPs Federation network model

3.1 A distributed approach

In contrast to hyperscalers, alternative initiatives take a different approach, primarily focusing on a distributed model. This model enables various cloud service providers to join a federation and participate in an environment where all providers are unified as one large Cloud Service Provider (CSP). The key belief among the partners involved in these initiatives is that constructing and consolidating the network infrastructure is the crucial first step toward achieving the desired outcome. In the broader context, the construction of the network infrastructure involves the collaborative efforts of Internet Exchange Points (IXPs) as neutral points of connection. These IXPs play a crucial role in facilitating the interconnection between various networks. In the upcoming chapter, a detailed discussion will be presented on the network fabric, which has been advocated as a potential solution.

3.2 Benefits and Disadvantages

Using a distributed approach is a challenge but it brings several benefits to the cause.

First and foremost, adopting a distributed approach enables Internet Exchange Points (IXPs) to join the network infrastructure at any time, contributing to enhanced reliability and reachability. Moreover, the governance structure will be fully decentralized, with each IXP taking responsibility for monitoring and maintaining its network, as is typically the case. From an economic perspective, each IXP will experience a boost in terms of selling ports, facilitating peering, establishing connections, and offering other network-related services.

In addition, adopting a distributed approach and sharing the network infrastructure with multiple IXPs will enable each Cloud Service Provider (CSP) to serve a larger user base from various parts of the world. By participating in the federation, each CSP contributes to a unified catalog of services, which combines the catalogs of all the CSPs involved in the federation.

One of the critical aspects of this solution is that each entity involved can continue focusing on its core activities without the need for major changes or additional training. For instance, IXPs can continue their primary function of providing connectivity through physical ports, leveraging their years of expertise in the field. By maintaining their established operations, there is no disruption to their existing business processes or the need for extensive retraining of personnel.

Similarly, other entities such as CSPs can continue to specialize in their respective areas of expertise, delivering their services to end customers without significant alterations. This approach recognizes and respects the wealth of knowledge and experience accumulated by each company in their specific domain.

By allowing entities to maintain their core activities and leveraging their existing know-how, the solution ensures a smooth transition and minimizes any potential resistance or challenges associated with adopting new practices. It also eliminates the need for extensive reorganization or retraining efforts, enabling a seamless integration of the federated network infrastructure while preserving the strengths and expertise of each participating entity.

However, constructing a distributed network through various IXPs that operate differently and employ different technologies presents a significant challenge.

Additionally, the final solution must have some characteristics to be adopted as a solution, including:

- High availability to ensure that the services will be always up and running
- Trustability to ensure that all the customers' data will be treated properly even if it passes through multiple entities
- High performances

It is important to note that the technology is not the only challenge in this case. IXPs, CSP, telcos, system integrators and so on have to be involved and convinced about this new innovative approach.

3.2.1 Involved entities and their roles

There are several key entities involved in the solution, including:

- **Cloud Service Provider (CSP)**: CSPs play a crucial role as they are the primary entities that utilize the network infrastructure to distribute their services across Europe. They rely on the network infrastructure to establish connectivity and deliver their services to end customers.
- **Internet Exchange Point (IXP)**: IXPs serve as the backbone of the network infrastructure. Being neutral entities, IXPs are ideally positioned as core points within the network infrastructure. They facilitate the exchange of traffic between different networks and enable seamless interconnection between CSPs.

The overarching goal of the federation is to create a reliable and automated network that supports the interconnection of multiple CSPs, thereby forming a unified cloud ecosystem.

3.2.2 Economic considerations

While this thesis does not focus on economic aspects, it is important to understand how a federation can contribute to increased income for both IXPs and CSPs:

- **IXP**: The primary revenue stream for an IXP is derived from the sale of physical ports. By participating in the federation, the number of interconnected entities increases, resulting in a higher demand for ports at each IXP. This, in turn, leads to increased revenue for IXPs.
- **CSP**: The revenue generated by CSPs primarily comes from the services they offer to their customers. Joining the federation enables CSPs to reach a larger customer base, thus expanding their service offerings and generating more sales. As a CSP attracts more customers, the need to scale up resources becomes inevitable, facilitating easy growth for the CSP.

Chapter 4

A new Network Fabric architecture

As previously discussed, a distributed approach seems to be the right way to proceed and to involve as much entities as possible. This chapter will cover the most important details about the network fabric and its components.

Creating a real distributed network requires the definition of a commonly agreed network fabric, which is established through the participation of federated partners. The network fabric consists of architectural elements that each participant must deploy within their own network. The proposed federation is built upon existing technologies, but it leverages them in a different manner. The following diagram illustrates the interconnection among different IXPs, which has been widely distributed and accepted by the partners:

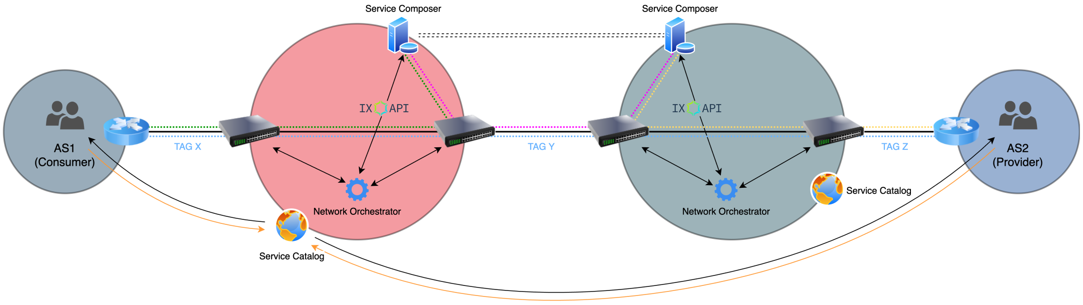


Figure 4.1: IXPs federation network fabric

This schema represents the established interconnections between multiple Internet Exchange Points (IXPs), showcasing the network fabric that forms the foundation of the distributed network. The acceptance and adoption of this schema by the federated partners ensure a unified approach to interconnecting their networks, facilitating seamless communication and collaboration across the federation.

The networking infrastructure is built upon BGP peering, which facilitates the exchange of routes among routers. BGP plays a crucial role in ensuring fault tolerance within the network. However, the goal is to enhance networking optimization capabilities by incorporating custom metrics that BGP does not inherently support. Therefore, the plan is to replace BGP with a custom product that leverages these custom metrics to compute optimal paths. Some of the custom metrics considered for path computation are:

- **Latency**: This metric focuses on minimizing the delay or latency experienced in routing traffic between different network nodes. By considering latency as a factor, the network can prioritize paths with lower latency to improve overall performance.

- **Redundancy**: Redundancy is an important metric for ensuring high availability and resilience within the network. By evaluating the level of redundancy along different paths, the network can prioritize routes that offer greater redundancy, minimizing the impact of failures or disruptions.
- **Maximum throughput**: Maximizing throughput involves optimizing network paths to accommodate higher data transfer rates. By considering the capacity and available bandwidth along different paths, the network can select routes that offer maximum throughput, ensuring efficient data transmission.

By incorporating these custom metrics into the path computation process, the network can achieve enhanced performance, resilience, and efficiency, tailored to the specific requirements of the federation.

The key infrastructural component responsible for replacing the BGP path computation is known as the Service Composer (SC). It plays a critical role in the overall infrastructure. Along with path computation, the SC offers several other important functionalities, including:

- **Coordinating internal network automation**: The SC serves as a central coordinator for internal network automation processes. It ensures the smooth orchestration of various automation tasks and components within the infrastructure.
- **Passing requests to the next Service Composer**: In order to compose the full path, the SC forwards requests to the next Service Composer in the chain. This collaborative approach enables the composition of complex network paths across multiple components.
- **Exposing API for initiating service composition**: The SC provides an API that allows users or other components to initiate the service composition process. This API serves as an interface for requesting specific network configurations and initiating the path computation.

To establish seamless communication with the network automation layer, a standardized solution is required. In this context, IXApi has been proposed as the standard solution. Developed by leading Internet Exchange Points (IXPs) in Europe, IXApi offers comprehensive support for various technical and economic aspects related to network automation. By adopting IXApi, the infrastructure ensures compatibility and interoperability with the automation layer, facilitating efficient and standardized operations.

Thanks to the utilization of IXApi, the automation processes within each Internet Exchange Point (IXP) network can be implemented in a flexible manner. There are no specific guidelines or standards imposed for performing network automation within the IXPs. The only requirement is to use IXApi as the designated entry point for automation activities.

The Service Composer (SC) plays a crucial role in the infrastructure by conducting path computations. The result of these computations is the establishment of a layer two link between the source and destination Cloud Service Providers (CSPs). This layer two link ensures physical connectivity between the CSPs, serving as a foundation for establishing higher-layer connections and communication. By enabling this layer two connectivity, the infrastructure facilitates the seamless transfer of data and network traffic between the participating entities.

Following the establishment of the layer two link, the next step involves creating a layer three network on top of the newly created layer two infrastructure. While the Service Composer (SC) does not directly handle this task, it can initiate the process if necessary. It is important to note that the creation of a layer three connection is not always required in every scenario.

In the case of the Proof of Concept (PoC) developed for Structura-X, a layer three interconnection was necessary to facilitate the interconnection of Kubernetes clusters. However, it is crucial to understand that the layer three negotiation is optional. If there is no need for layer three reachability or interconnection between specific entities or networks, it can be omitted. This flexibility allows for a tailored approach where layer three connectivity is enabled only when required, minimizing unnecessary configuration and resource utilization.

The proposed architecture, presented at the Euro-X conference, has received widespread acceptance as a robust solution. The primary entities involved in the development of this solution include Internet Exchange Points (IXPs) and Cloud Service Providers (CSPs). However, the potential for expansion and inclusion of additional stakeholders remains open. Some of the entities that could join the initiative include:

- **Telcos**: Telecommunication companies can contribute their infrastructure and expertise to enhance the federated cloud and network infrastructure.

- **Network as a Service (NaaS) providers**: NaaS providers can offer specialized networking services that complement the existing infrastructure and contribute to the overall functionality.

- **System integrators**: System integrators can play a crucial role in ensuring seamless integration and compatibility among various components of the federated infrastructure.

Moreover, the participation of hyperscale cloud providers should not be excluded. In fact, their involvement presents a unique opportunity to promote interoperability and collaboration among different cloud providers. By fostering a more interoperable ecosystem, the federated cloud infrastructure aims to improve the flexibility and efficiency of cloud services, benefiting both providers and users.

Chapter 5

Related Works

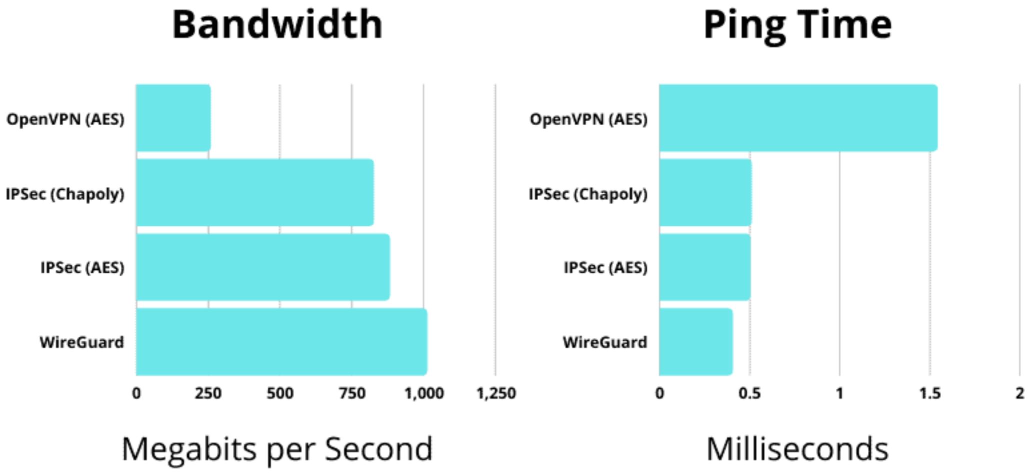
Interconnecting different networks is a well-known problem with multiple existing solutions. One common example is the use of Virtual Private Networks (VPNs) to connect networks located in different geographic locations. Additionally, there are other solutions such as MEF SD-WAN or simple Internet connections.

5.1 VPN

VPNs are a well-established technology that allows the interconnection of geographically dispersed networks. However, there are several drawbacks and limitations that make VPNs less suitable for our specific goals.

Firstly, using a VPN requires a pre-defined networking infrastructure between the Cloud Service Providers (CSPs). In the project's context, one of the objectives is to create a reliable and performant network where partners can manipulate and manage the optimal path based on custom metrics. This level of control is not feasible within the current implementation of the Internet.

Secondly, VPNs often suffer from performance limitations. Bandwidth capacity is a significant constraint in VPN technologies. The chart below illustrates the speed and latency characteristics of common VPN solutions[3]:



As shown, the maximum throughput typically reaches around 1 Gbps, which may not be sufficient for data-intensive scenarios in many cases.

5.2 MEF SD-WAN

MEF SD-WAN, also known as MEF 70, is a standard developed by the MEF (Metro Ethernet Forum) to define the requirements and specifications for software-defined wide area networking (SD-WAN) services. MEF is an industry consortium that focuses on developing and standardizing networking technologies and services.

SD-WAN is a technology that enables organizations to create virtualized and software-defined networks over wide area connections, such as MPLS (Multiprotocol Label Switching), internet, or cellular networks. It provides centralized control and management of network traffic, allowing organizations to optimize connectivity, improve application performance, and reduce costs. [4]

Some key aspects of MEF SD-WAN include:

- **Service Attributes**: MEF SD-WAN specifies a set of service attributes that define the behavior and performance of SD-WAN services. These attributes cover areas such as connectivity, performance, security, and traffic handling.
- **Service Orchestration**: MEF SD-WAN emphasizes the need for service orchestration capabilities, enabling the automation and management of SD-WAN services across multiple network domains. It promotes the use of open APIs (Application Programming Interfaces) for seamless integration with existing network management systems.
- **Interoperability**: MEF SD-WAN focuses on ensuring interoperability between different SD-WAN solutions from various vendors. This allows organizations to choose and integrate SD-WAN products and services from different providers while maintaining compatibility and avoiding vendor lock-in.
- **Security**: MEF SD-WAN addresses security considerations by defining requirements for encryption, authentication, access control, and threat detection. It promotes the use of industry-standard security mechanisms to protect SD-WAN deployments.

Using SD-WAN in a distributed and multi-provider scenario presents challenges, but this is not the primary reason for not adopting this technology. Several factors contribute to our decision, including the newness of the technology, ongoing development efforts, and limited vendor support.

- **Limited Vendor Support**: Although MEF SD-WAN aims to ensure interoperability between different SD-WAN solutions, the standard is relatively new, and not all vendors may have fully implemented it. As a result, there might be limitations in terms of vendor support and compatibility, which could impact the ability to mix and match different vendor solutions seamlessly. In a eterogeneous and distributed scenario the full compatibility is a MUST.
- **Evolving Standard**: MEF SD-WAN is a dynamic standard that continues to evolve over time. New updates and versions are released periodically

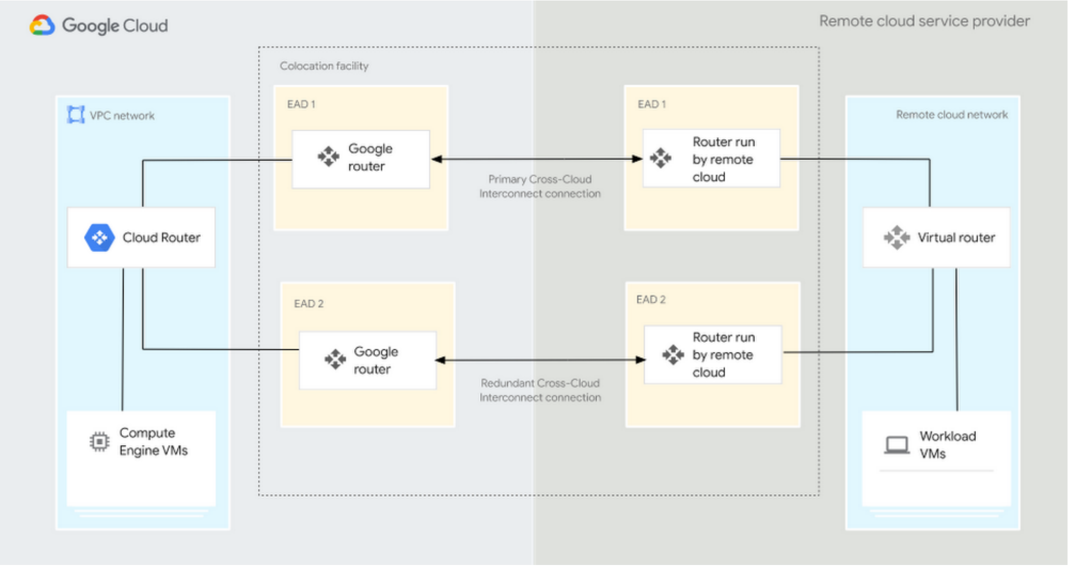
to address emerging challenges and industry requirements. However, these updates may introduce changes that require service providers and organizations to adapt their existing deployments, potentially causing compatibility issues or necessitating additional configuration changes.

- **Complexity in Implementation**: Implementing MEF SD-WAN may involve significant configuration and integration efforts. Organizations may need to make changes to their existing network infrastructure, introduce new management systems, and train their staff to work with the new standard. This complexity could lead to challenges during the deployment and require expertise in SD-WAN technologies.

In the scenario of a federation we are trying to adopt and develop ad-hoc technologies in order to keep everything simple, traceable and monitorable.

5.3 Cross-Cloud Interconnect

Cross-Cloud Interconnect is a newly released technology by Google that simplifies the interconnection between two different cloud service providers (CSPs). This fully managed service can be enabled with just a few clicks. It offers high-speed connectivity options, including 10Gbps and 100Gbps connections between Google Cloud Platform (GCP) and other partner providers.


Google Cloud
Remote cloud service provider
Colocation facility
VPC network
EAD 1
Google router
Primary Cross-Cloud Interconnect connection
EAD 1
Router run by remote cloud
Remote cloud network
Cloud Router
Virtual router
EAD 2
Google router
Redundant Cross-Cloud Interconnect connection
EAD 2
Router run by remote cloud
Compute Engine VMs
Workload VMs

However, it is important to note that while this solution facilitates interconnection between different providers, it remains under the control of a limited number of selected entities.

Chapter 6

The L3 Negotiation Problem

In the context of the network federation combined with the cloud federation, it is essential to enable the purchase of services from the catalog and ensure their accessibility to the buyers. Initially, the primary customers are Internet Exchange Points (IXPs) or Cloud Service Providers (CSPs). However, there is a future vision where end users will also have the ability to procure and utilize these services.

The accompanying image provides an illustrative example of virtual machines (VMs) being acquired from one CSP and hosted within the infrastructure of another CSP. This showcases the interoperability and cross-provider capabilities that the network and cloud federation aspire to achieve.

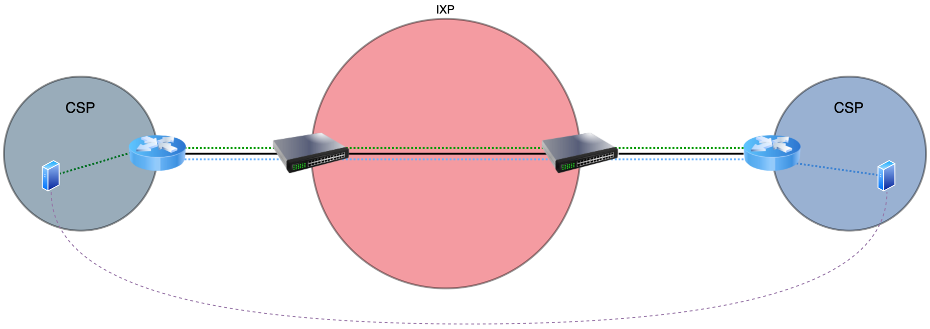


Figure 6.1: CSPs connected to the same IXP

In the previous scenario, several actions were taken to transition from *no connection between CSPs* to *reachability between CSPs*. It is important to remember that in this vision, both CSPs are part of the federation and possess the necessary set of tools for enabling cross-network automation. As discussed in the previous chapter, there are a few elements that facilitate the connection between two IXPs, including:

- **Service Composer**: Enables the best path computation through the connected IXPs. The path computation done by the SC is based on custom metrics to ensure a certain level of service quality.
- **L2 Automation**: Allows each IXP to create the resources needed for a logical L2 link from the source IXP to the destination IXP.
- **L3 Automation**: The final step when the bought service requires Layer 3 reachability. It is important to note that not all services need Layer 3 reachability (e.g., AI training).

These elements collectively contribute to enabling seamless connectivity and effective service delivery within the federation.

6.1 Why is L3 negotiation needed?

To understand why Layer 3 (L3) negotiation is necessary, it is important to consider the circumstances under which it arises. The primary services offered by CSPs are often Software-as-a-Service (SaaS), and in most cases, utilizing them as a service requires a Layer 3 connection. A prime example is a database, which is a software product that a CSP may purchase from another CSP. Once the database is acquired, the buyer naturally wishes to use it, necessitating a Layer 3 connection between the two CSPs.

During the initial Proof of Concept (PoC) phase of Structura-X, the partners encountered the challenge of establishing Layer 3 connectivity among themselves. A Significant manual effort was required to configure the network devices involved. The manual configuration process took over a week to complete because each partner had to share their internal subnet where the target machines were located, as well as their external subnet where their router could be reached.

After the Proof of Concept (PoC) phase of Structura-X, all the participating partners unanimously recognized that L3 automatic negotiation was an essential requirement. This decision was made because, when a customer purchases a service, it must be delivered in a matter of minutes. Waiting for days to establish the required connectivity is deemed unacceptable in terms of service delivery.

6.2 Implementation challenges

When all configurations are done manually, the process becomes simpler, and it helps avoid numerous addressing-related problems, especially in ad-hoc environments created for specific purposes. However, in real-world scenarios with multiple networks and customers, managing addressing problems can become a significant challenge. Some of the common challenges include:

- **Networks overlapping**: This occurs when the source network and destination network have the same subnet or overlapping subnets. In manual negotiations, resolving this issue is relatively straightforward through phone calls or email communication, often by implementing double NAT to circumvent the problem. However, automation introduces complexities in handling such situations.

- **Net-to-Net reachability**: Establishing connectivity between two IXPs via a logical Layer 2 link requires careful consideration. Choosing the appropriate approach becomes crucial. The commonly used solutions include "unnumbered ports" and "IP binding for each port."

- **Different network topologies**: Handling diverse network topologies adds another layer of complexity. The IP negotiation process must be adaptable to accommodate variations in network topologies and the specific network devices employed. It is possible that some IXPs use Cisco devices while others use Nokia devices, and the negotiation process must be compatible with both scenarios.

6.3 When L3 negotiation is needed

Another crucial requirement for the solution is the ability to offer optional negotiation. There are several scenarios in which L3 negotiation may not be necessary, including:

- **Pure computational tasks**: In cases where the service primarily involves computational tasks without the need for specific network connectivity, L3 negotiation may not be required.
- **Direct exposure to the internet**: If the bought service is directly exposed to the internet without the need for additional network-level connectivity, L3 negotiation may be unnecessary.

This means that the negotiation process should be optional within the broader connectivity process. This flexibility allows for different scenarios and use cases where L3 negotiation may not be essential.

Chapter 7

The proposed solution

7.1 The goals

The primary objective, as mentioned in the previous chapters, is to enable the auto-negotiation of layer 3 addresses, facilitating communication between two different remote networks managed by different Cloud Service Providers (CSPs). However, the solution encompasses additional technical goals. This chapter will outline these goals and explain how they have been addressed to create a functional and effective solution.

Creating a common standard interface

Presently, there is a lack of standardized Application Programming Interfaces (APIs) that serve as entry points for the purpose of negotiation. Therefore, a key goal is to establish a common standard interface that is flexible enough to accommodate various use cases of negotiation. The proposed solution features a versatile interface that is both easy to extend and modify. Moreover, the solution is entirely open source, allowing for continual improvement through contributions and suggestions from the community.

Ensuring interoperability

Another critical goal is to ensure interoperability between different systems and components within the network federation. With the participation of multiple IXPs and cloud service providers, it is vital to establish seamless interoperability across the federated network. This includes enabling communication and data exchange between disparate systems, network devices, and software solutions. By adhering to industry standards and employing compatible protocols, the solution fosters interoperability, promoting efficient collaboration among the various entities involved.

Enhancing scalability and flexibility

Scalability and flexibility are paramount in accommodating the growing demands and dynamic nature of the network federation. The solution aims to provide a scalable architecture capable of handling an increasing number of participants, services, and interconnected networks. It also emphasizes flexibility to adapt to evolving requirements and changing network topologies. By leveraging scalable and flexible design principles, the solution facilitates the seamless integration of new partners and services into the federation, ensuring its long-term viability and success.

Ensuring security and privacy

Security and privacy are of utmost importance in any network federation. As the solution facilitates communication and data exchange between different entities, it must incorporate robust security measures. This includes implementing secure communication protocols, encryption mechanisms, and access control mechanisms to safeguard sensitive information and protect against unauthorized access. By prioritizing security and privacy, the solution fosters trust among participants and instills confidence in the network federation.

Flexible implementation and templating

Flexibility is indeed a crucial property to achieve in the solution, as it allows for adoption by various partners with different technologies, topologies, and requirements. The negotiation process has undergone extensive testing and validation, particularly with respect to the commonly used network topology employed by the partners.

In the network topology depicted below, each Cloud Service Provider (CSP) is connected to a port of the nearest Internet Exchange Point (IXP). The IXP facilitates Layer 2 (L2) connectivity to the CSP router. Beyond the router lies the CSP network, which houses the services that can be accessed by potential customers seeking to purchase services from the CSP.

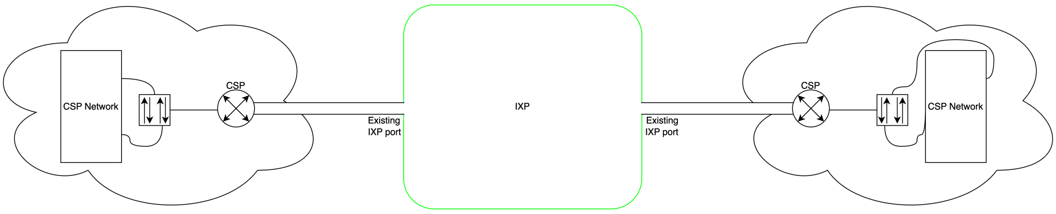


Figure 7.1: High-level network structure

This standardized network topology serves as the foundation for testing and validating the negotiation process. By establishing a common framework, it becomes easier to assess the functionality, efficiency, and compatibility of the solution within a well-defined network architecture.

Indeed, the proposed solution is designed to be adaptable and suitable for various network topologies. While the previously mentioned network topology serves as a common standard, the solution can be customized and configured by each Internet

Exchange Point (IXP) to accommodate the specific operations required to configure their network devices.

By allowing IXPs to implement custom configurators, the solution can effectively address the unique needs and complexities of different network environments. This flexibility ensures that the solution remains compatible and functional across heterogeneous environments, where network technologies, devices, and configurations may vary.

The concept of utilizing custom configurators enables the solution to maintain a fixed interface while still being adaptable to diverse network setups. This approach provides a consistent and standardized entry point for the negotiation process, allowing for seamless integration and interoperability between different partners and their respective networks.

By leveraging custom configurators, each IXP can define the specific operations and procedures necessary to configure their network devices appropriately. This customization ensures that the solution aligns with the specific requirements and protocols employed within each network environment.

In summary, the solution's adaptability, achieved through the utilization of custom configurators, allows for the proper configuration and programming of network devices in various network topologies. This approach ensures that the solution remains compatible and effective in heterogeneous environments, providing a fixed interface while accommodating the specific needs of different partners.

7.2 Components of the solution

The software solution developed is composed of two different components. Those components have to be running together, the solution cannot work if one of the components is not present.

- **Negotiator**: The negotiator serves as the entry point for the solution, providing a fixed interface to accept negotiation requests and manage them effectively. This component plays a crucial role in handling the negotiation process by receiving incoming requests, evaluating their feasibility, and communicating with the component responsible for configuring the network devices. Once the negotiator determines the feasibility of the request, it communicates with the component responsible for configuring the network devices. By providing a fixed interface, the negotiator offers a standardized entry point for negotiation requests, ensuring consistency and compatibility across various interactions.

- **Configurator**: Its primary role is to implement the necessary configurations on the network devices based on the specific requirements of each Internet Exchange Point (IXP). As the configurator operates at the network device level, it is programmed in an ad-hoc manner for each IXP, taking into account factors such as the device vendor, the number of devices involved, and the unique network topology. Since different vendors may have varying command sets and configuration protocols, the configurator is tailored to accommodate these specific requirements. It contains the necessary logic and programming instructions to interact with the network devices, ensuring the correct configurations are applied consistently across the infrastructure. Ultimately, the configurator plays an important role in translating the configuration requirements of the negotiation process into actionable commands for the network devices

The two components, the negotiator and the configurator, establish communication between them. In the proposed solution architecture, it is recommended to deploy these components in the same network for ease of communication. However, it is not a strict constraint, as long as they can establish network connectivity and reach each other. The communication between the negotiator and configurator is implemented using gRPC, a high-performance, open-source remote procedure call (RPC) framework. gRPC allows efficient and secure communication between distributed components by defining service interfaces and message types using Protocol Buffers. The data exchanged between the negotiator and configurator includes all the necessary parameters required for configuring a network device. This data encompasses a range of configuration details, such as network addresses, ports, and any other relevant parameters specific to the network device being configured.

7.2.1 The negotiator

The negotiator component serves as a standardized interface that remains consistent across different environments within the federation. It provides a set of configurable flags that allow participants in the federation to customize its behavior according to their specific requirements. Certain flags are mandatory and must be configured to ensure proper functionality. These mandatory flags include:

- **Local Subnet**: This flag specifies the subnet that the participant wants to make accessible to their customers. It defines the range of IP addresses that will be allocated to the customer's network.

- **Border Router Port**: This flag determines the port of the participant's border router that will be configured when a negotiation request is received. The traffic from the customer's network will be directed to this port, and the negotiator will configure the necessary routes and IP addresses on this interface.

- **List of Private Networks**: This flag comprises a list of private networks that are used when there is an overlap between the local network and the remote network. These private networks help resolve the issue of conflicting IP address ranges and ensure proper communication between the networks.
- **List of Negotiated Networks**: This flag consists of a list of private networks that will be negotiated with the remote configurator. These networks are employed in establishing a point-to-point link between the two networks in scenarios where using an unnumbered IP address is not feasible.

The mandatory flags serve as default values within the negotiator component. These default values are designed to provide a baseline configuration that can be adjusted and overridden during the negotiation phase. This flexibility is crucial as it enables participants to customize the negotiation process based on their specific service requirements.

During the negotiation phase, participants have the opportunity to provide specific configurations that deviate from the default values set by the mandatory flags. This means that if a particular service demands specific configurations or settings, the negotiator can adapt and accommodate those requirements.

By allowing participants to override the default values, the negotiation process becomes highly customizable and adaptable.

Interfaces

The Negotiator component offers two distinct interfaces to fulfill different purposes:

- **CLI Interface**: This interface is essential for configuring the Negotiator before initiating its operation. It allows users to set and specify the necessary flags discussed earlier, such as the local subnet, border router port, private networks, and other relevant parameters. The CLI interface ensures that the Negotiator is properly configured with the desired settings before it is started.
- **REST Interface**: This interface is used to interact with the Negotiator during the negotiation process. It exposes two endpoints that serve different functions. The first endpoint is responsible for initiating a negotiation request, while the second endpoint is designed to receive and handle incoming negotiation requests from other participants.

The decision to utilize the HTTP protocol for the REST interface is driven by several factors. Firstly, HTTP is widely supported and easily accessible, making it a convenient choice for communication between different components. Additionally, leveraging HTTPS (HTTP over TLS) provides a layer of security by encrypting the communication between the Negotiator and other entities. This ensures the confidentiality and integrity of the transmitted data. Even in a protected environment like the federation, HTTPS can be utilized, allowing the use of self-signed certificates to establish secure connections.

The REST interface exposes two endpoints:

- **/api/v1/start_negotiation**: This endpoint serves as the starting point for initiating a new negotiation process. It is designed to be called from an external entity, such as a catalog or a client seeking to acquire a service. When invoked, this endpoint triggers the negotiation process within the Negotiator, initiating the required actions and interactions.
- **/api/v1/handle_negotiation**: This endpoint is responsible for handling incoming negotiation requests from other instances of the Negotiator component. It enables communication and coordination between multiple Negotiators within the federation. When this endpoint is invoked by a remote Negotiator, the receiving Negotiator processes and responds to the negotiation request, facilitating the establishment of connectivity and configuration between the involved networks.

By offering these two endpoints, the REST interface provides the necessary functionality to initiate and manage the negotiation process.

7.2.2 The configuration agent

The Configuration Agent is a crucial component responsible for configuring the network devices within the federation. Due to the diverse network topologies and device types used by different IXPs, the Configuration Agent is designed to be programmable and replaceable by each IXP.

The communication between the Negotiator and the Configuration Agent is facilitated through the gRPC interface. This interface allows the exchange of data structures containing all the necessary parameters for configuring network devices. Importantly, these parameters are vendor-agnostic, meaning they can be applied regardless of the specific devices' vendor. This ensures that the gRPC interface can provide the required functionalities consistently across different types of network devices.

By implementing the Configuration Agent as a programmable and replaceable component, each IXP can tailor its configuration processes to match its specific network topology and device requirements.

The gRPC interface provides several important actions that can be performed by the Configuration Agent.

Each gRPC action accepts a message as input, which is defined in the protobuffer file. This approach ensures flexibility and ease of future updates to the interface.

The protobuffer definition includes message types such as Result, IPAssignment, Route, and NatConfiguration. These message types define the structure and parameters required for each specific action.

The Result message type includes fields for indicating the success or failure of the executed action and providing an associated message.

The usage of gRPC enables the implementation of the Configuration Agent in different programming languages, ensuring seamless communication between the Negotiator and the Configuration Agent components. This language-agnostic approach allows for greater flexibility and interoperability within the federation.

The protobuffer definition of the three actions described above is

Interfaces

The current implementation of the solution provides a subset of actions that are sufficient for configuring network devices. These actions, along with their corresponding messages, are:

- **AddIPToInterface**: This action assigns an IP address to a specific interface on a network device.
- **AddRoute**: This action creates a routing rule on a network device.
- **ConfigureNat**: This action configures a Network Address Translation (NAT) rule on a network device.

The protobuffer definition file serves as the description of the interface. It specifies the messages and actions that can be used for communication between the Negotiator (client) and the Configuration Agent (server). The protobuffer definition file acts as a contract between the client and server, ensuring compatibility and proper communication.

The gRPC architecture allows for flexibility in the implementation of the Configuration Agent. As long as the protobuffer definition remains unchanged, the implementation of the server can be modified without causing any issues with the Negotiator client.

This separation of concerns between the negotiator and configuration agent components, along with the use of gRPC and protobuffer definitions, enables the solution to adapt to different implementations and configurations while maintaining interoperability.

Changing the interface definition

When a new action is needed in the interface provided by the Configuration Agent, the first step is to update the protobuffer file to include the new action definition.

This can be done by adding the necessary message and action definition to the existing protobuffer file.

Once the protobuffer file is updated, you can use the protoc command, which is the Protocol Buffer Compiler, to generate the stub code. The stub code includes the client-side code that the Negotiator will use to communicate with the Configuration Agent.

After generating the stub code, the Negotiator needs to be updated to incorporate the new interface provided by the Configuration Agent. This includes updating the client-side code to handle the new action and message types defined in the protobuffer file.

By following this process, you can easily extend the interface of the Configuration Agent by adding new actions, generating the updated stub code, and updating the Negotiator to use the new interface.

7.3 Implementation for Linux machines

The example scenario implemented for the demonstration of the solution replicates the commonly used network topology in the Structura-X Proof of Concept (PoC). This scenario also includes partners who were using Linux-based routers. The network topology is represented by the following diagram

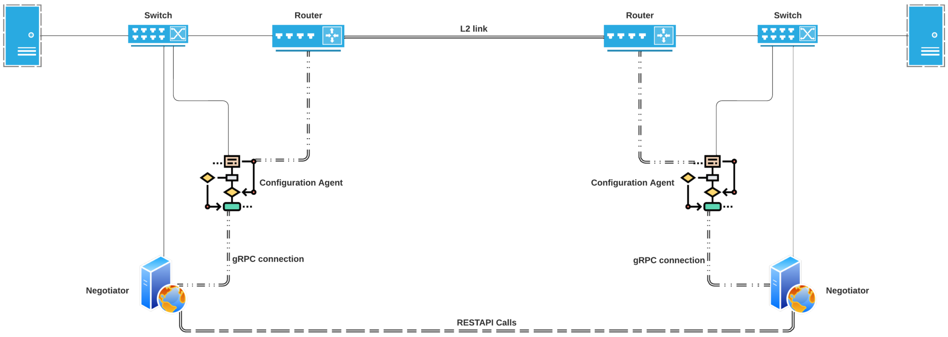


Figure 7.2: Network topology used to develop the negotiation process

In this scenario, there are two CSPs connected to their respective IXP. Each CSP is equipped with a Linux Router, which serves as the border router for the internal subnet where their services are located. The goal of the solution is to enable layer 3 connectivity between the internal subnets, allowing them to communicate with each other.

This example demonstrates the effectiveness of the solution in handling the network topology and configurations of different partners, including those using Linux-based routers. It showcases how the negotiator and configuration agent components can be utilized to establish and configure the necessary network connections.

The scenario has been replicated using Kathara[5], an open-source software for network virtualization. Kathara allows you to define the network infrastructure using a text file, where you can specify the different broadcast domains. In this scenario, four distinct broadcast domains have been created to represent the network topology:

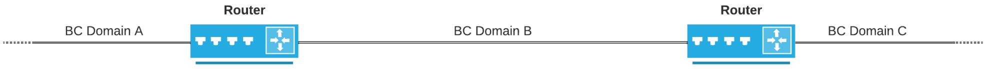


Figure 7.3: Broadcast domains between CSP

- **Broadcast domain A and C**
  - It includes the Linux router and the internal subnet
  - The Linux router serves as the border router
- **Broadcast domain B**: allows the communication between the negotiators
- **Broadcast domain D**: this BC domain abstracts the layer 2 interconnection provided by the IXPs

The broadcast domain B has been introduced to establish reachability for the negotiators. This ensures that the negotiator components of each CSP can effectively communicate and interact with one another. The discussion on negotiators' reachability alternatives will be deferred to a later section, where we will explore and analyze various approaches to establish connectivity between negotiators situated in different networks.

The Kathara definition is as follows

```
vm1[0]=A

```

```
router1[0]=A
router1[1]=B
router1[2]=D

router2[0]=C
router2[1]=B
router2[2]=D

vm2[0]=C
```

7.3.1 Negotiator and Configurator placement

In this example, simplicity has been prioritized to emphasize the functionality of the solution rather than its complexity. As a result, both the negotiator and configurator components have been placed on the Linux router machines. This decision has no impact on the functionality of the solution but helps avoid introducing unnecessary machines into the network. Furthermore, it allows for a clear and straightforward presentation of the solution flow.

By keeping the setup streamlined, the focus remains on showcasing the workings of the solution without unnecessary distractions. This approach enables a more concise and effective demonstration of the solution's capabilities.

7.3.2 Initial Layer 3 Networks

To replicate a real scenario, we need to assign layer 3 networks to the CSPs. In this use case, the configuration is straightforward as we need to assign layer 3 networks to broadcast domains A, B, and C. Broadcast domain D emulates a pure layer two link between the CSPs.

Broadcast Domain A Configuration

In this domain, there are 2 machines to be configured. The chosen subnet is 10.0.0.0/24.

- **Router**: This machine serves as the default router for the server connected to the switch, so it must have an assigned IP. In this case, we assign 10.0.0.1/24 to the interface connected to domain A.

- **Server**: This machine has the Router as its default router, and it also has an IP in the same subnet. In this case, we assign 10.0.0.2/24.

Broadcast Domain B Configuration

This domain enables communication between the two negotiators. The chosen subnet for this domain is 10.0.1.1/30.

- **Left Router**: The interface connected to domain B is assigned the IP 10.0.1.1/30.
- **Right Router**: The interface connected to domain B is assigned the IP 10.0.1.2/30.

Broadcast Domain C Configuration

In this domain, there are 2 machines to be configured. The chosen subnet is 10.0.0.2/24.

- **Router**: This machine serves as the default router for the server connected to the switch, so it must have an assigned IP. In this case, we assign 10.0.0.1/24 to the interface connected to domain A.
- **Server**: This machine has the Router as its default router, and it also has an IP in the same subnet. In this case, we assign 10.0.0.2/24.

The decision to use the same layer 3 network for both sides is intentional, as it allows us to demonstrate how the one-side NAT strategy works. By using the same layer 3 network, we can showcase how the negotiation process handles the overlapping IP addresses and successfully establishes connectivity between the two networks. This scenario helps highlight the effectiveness of the solution in managing and resolving IP conflicts in a heterogeneous environment.

7.3.3 Configuration of the Negotiator and Configurator

With the network scenario in place, the next step is to configure the two components: the Negotiator and the Configurator. In this example, both components are located on the same machine, but their functionalities remain the same regardless of their placement. In a real scenario, they could be deployed on separate machines or within a Kubernetes cluster.

The Negotiator, being a fixed interface that facilitates the negotiation process, does not require any additional configurations. It can simply be executed as a background process.

On the other hand, the Configurator needs to be programmed to accommodate the network topology and configure Linux machines. Since the Configurator runs on the Linux machine acting as the Router in this implementation, commands can be executed directly on the machine without the need for a remote connection.

IP Assignment

Assigning an IP to the Linux machine is accomplished using the *ip* command. The Configurator receives a message from the Negotiator containing all the necessary information to bind the IP to the appropriate interface. The message, defined in the protobuffer file, consists of two fields:

```
message IPAssignment {
  string ip;
  string interface;
}
```

With this information at hand, it becomes possible to assign an IP to the specified interface by executing the following *ip* command:

```
ip addr add <IPAssignment.ip> dev <IPAssignment.interface>
```

By running this command, the IP address specified in the message is assigned to the corresponding interface.

Route Creation

To create a routing rule on the Linux machine, the Configurator receives a message from the Negotiator that contains the necessary information. The message structure, defined in the protobuffer file, includes two fields:

```
message Route {
  string destinationNetwork = 1;
  string nextHop = 2;
}
```

Using this message, the Configurator can create the desired routing rule. The following command demonstrates how to add a route based on the information provided:

```
ip route add <Route.destinationNetwork> via <Route.nextHop>
```

By executing this command, a route is added to the Linux machine's routing table, directing traffic destined for the specified destination network to the specified next hop.

Handling Network Overlapping and Single-Side NAT

To address the issue of network overlapping, a single-side NAT strategy has been adopted. This strategy allows for the separation of responsibilities and eliminates the need for mutual agreement on the NAT process between the participating CSPs. The approach offers several benefits, including:

- **Debugging**: With a single CSP responsible for enabling the NAT process, troubleshooting and debugging become easier. The responsible CSP can focus on resolving any issues related to network overlapping and NAT configuration.
- **Controllability**: In case of failures or unexpected behavior, a single CSP has control over the NAT process. This simplifies the management and control of the NAT configuration.

To implement the single-side NAT strategy, the network device must support two key technologies:

- **Network Address Translation (NAT)**: The NAT functionality allows for the translation of IP addresses between different networks. It enables the mapping of private IP addresses to public IP addresses, facilitating communication between overlapping networks.
- **Virtual Routing Table**: The virtual routing table feature enables the creation of separate routing tables within the network device. This allows for the proper routing of traffic between the overlapping networks, ensuring that packets are forwarded to their intended destinations.

Fortunately, these technologies are supported by most network devices, making the implementation of the single-side NAT strategy replicable across various environments, not limited to Linux-based systems.

The action responsible for performing NAT configuration in the configurator is in charge of creating a Virtual Routing Table (VRT), enabling policy-based routing, and configuring the NAT process. In Linux environments, the *iptables* utility is used to configure this process. The message received from the negotiator for NAT configuration is defined as follows:

```
message NatConfiguration {
  string localNetwork;
  string remoteNetwork;
  string localInterface;
  string tableNumber;
  string packetMark;
  string nextHop;
}
```

To configure NAT, the following commands are executed in the configurator:

```
iptables -t nat -A PREROUTING -d <NatConfiguration.localNetwork> -j
  NETMAP --to <NatConfiguration.remoteNetwork>

iptables -t nat -A POSTROUTING -d <NatConfiguration.remoteNetwork> -j
  NETMAP --to <NatConfiguration.localNetwork>

iptables -t mangle -A PREROUTING ! -i <NatConfiguration.localInterface
  > -j MARK --set-mark <NatConfiguration.packetMark>

ip route add default via <NatConfiguration.nextHop> table <
  NatConfiguration.tableNumber>

ip rule add fwmark <NatConfiguration.packetMark> lookup <
  NatConfiguration.tableNumber>
```

7.4 Visibility of Negotiators

The visibility of negotiators is an important aspect to consider as it directly impacts their ability to communicate with each other. In order for negotiators to work properly, there needs to be an existing network through which they can reach each other. This visibility is crucial for successful negotiation processes.

There are two main alternatives for achieving negotiators' visibility, each with its own pros and cons.

7.4.1 Using Internet Connectivity

One option is to utilize the existing internet connectivity. This may appear to be the easiest alternative since internet connections are already widespread and all the CSPs can potentially expose a service to the public internet. The positive aspect of this approach is that it does not require any additional configuration.

However, there are some considerations to keep in mind:

- **Security**: Exposing the negotiator to the public internet without proper authentication or authorization methods is not safe. The current implementation lacks these security measures as it was designed for a protected and restricted environment. This exposes the negotiator to potential risks, including unauthorized access and security breaches.

- **Control-plane Optimization**: Utilizing the internet as the control-plane for negotiators may not be optimal in the context of a private and internet-independent federation. Depending on the internet for control-plane communication introduces dependencies and may lead to connectivity issues in certain scenarios.

Given these considerations, it is important to carefully evaluate the security risks and the impact on the control-plane architecture before deciding to rely on internet connectivity for negotiators' visibility.

7.4.2 Using a Private and Pre-existing Control-plane Network

The second alternative is to utilize a pre-defined private network among the federated entities. This approach ensures that all traffic remains within the private network, enhancing security and maintaining control over the communication between negotiators. However, there are some challenges associated with this option:

- Network Maintenance: One important consideration is the maintenance of the control-plane network. Since it is a private network, the responsibility falls on the federated partners to ensure its reliability and availability. This requires careful monitoring, proactive management, and timely troubleshooting to address any issues that may arise. Establishing service level agreements (SLAs) and implementing robust network management practices can help guarantee the reliability of the network.

- Scalability: Another aspect to consider is the potential growth of the control-plane network, especially as new partners are acquired across Europe. As the network expands, it may become more challenging to control and manage effectively. Proper network design principles, such as segmentation, scalability planning, and centralized management, can help mitigate these difficulties and ensure the network remains manageable and efficient.

While utilizing a private and pre-existing control-plane network offers greater control and security, it requires ongoing attention and investment in network maintenance and scalability planning. By addressing these challenges proactively, the federated partners can establish a reliable and robust control-plane network to facilitate negotiators' visibility and secure communication within the federation.

7.5 Initiating the L3 Negotiation Process

It is crucial to determine which component is responsible for initiating the L3 negotiation when it is required. In the overall architecture, there are three main components involved:

- Virtual L2 link
- L3 negotiation
- Federated catalog

The federated catalog serves as the entry point for the entire process, allowing users to browse and purchase services. Based on the selected service, a series of actions are executed, including the L3 negotiation. One example of a service that requires L3 reachability is a managed VM. The federated catalog is divided into two primary components:

- **Client**: This is the user-facing web interface that enables browsing and selection of available services.
- **Server**: This component is deployed in each IXP and CSP network. It is publicly accessible to the clients, but it also has access to the private network and can communicate with the L3 negotiator.

When a user purchases a service, the server-side of the federated catalog identifies the necessary actions to be performed. In a complete scenario, two main asynchronous actions take place:

- **Virtual Layer 2 Link Creation**: The *service composer* is responsible for creating the L2 links as the underlying infrastructure element. The catalog server awaits a response from the service composer before proceeding.
- **L3 Negotiation**: Once the L2 link has been created, the L3 negotiation can commence. The catalog server invokes the negotiator by calling the */api/v1/start_negotiation* endpoint. This invocation occurs through the private network.

Chapter 8

Conclusions

In this thesis, we have explored the concept of a federated cloud and network infrastructure, aimed at creating a reliable and scalable solution for interconnecting multiple Cloud Service Providers (CSPs) across Europe. Through our discussions, we have examined various aspects of this infrastructure, including its architecture, components, and the benefits it offers to the involved entities.

The proposed solution emphasizes flexibility, adaptability, and automation. By adopting a distributed approach and leveraging existing technologies, the federated infrastructure enables seamless interconnection among CSPs, Internet Exchange Points (IXPs), and other entities, such as Telcos, Network as a Service (NaaS) providers, and system integrators.

One of the key components in this infrastructure is the Service Composer (SC), which plays a critical role in path computation and coordination of network automation processes. The SC, along with the IXApi standard solution for network automation, ensures efficient and standardized operations across the network.

Through automated layer 3 negotiation, the solution enables rapid and scalable setup, replacing manual and time-consuming processes. The Negotiator and Configuration Agent components work in tandem to provide a standardized interface and handle network configuration details, making the solution adaptable to different network topologies.

The integration of this solution into the larger ecosystem involves collaboration among CSPs, IXPs, and other entities. The federation opens up opportunities for increased income for both IXPs and CSPs. IXPs can sell more physical ports as the number of interconnected entities grows, while CSPs can reach more customers and expand their service offerings.

Furthermore, the federated infrastructure respects the expertise and core activities of each entity, allowing them to continue their main business activities without the need for significant changes or additional training.

In conclusion, the proposed federated cloud and network infrastructure offers a promising solution for creating a unified cloud environment in Europe. Its flexibility, automation capabilities, and focus on collaboration among entities make it an attractive option for CSPs, IXPs, and other stakeholders. The successful development of a Proof of Concept (PoC) has provided valuable insights and serves as a foundation for further advancements in this field.

As future work, it is essential to continue refining and expanding the federated infrastructure, involving more partners and addressing any challenges that may arise. This includes further exploration of the layer 3 negotiation process, continuous improvement of network automation, and integration with emerging technologies and industry standards.

Overall, the federated cloud and network infrastructure presented in this thesis hold great potential for transforming the cloud computing landscape, fostering interoperability, and creating new opportunities for collaboration and growth among European entities.

Bibliography

[1] Google. *Advantages and Disadvantages of Cloud Computing.* https://cloud.google.com/learn/advantages-of-cloud-computing (cit. on p. 6).

[2] *GAIA-X.* https://www.data-infrastructure.eu/GAIAX/Navigation/EN/Home/home.html (cit. on p. 9).

[3] *VPN.* https://www.purevpn.com/blog/wireguard-vs-openvpn/ (cit. on p. 20).

[4] *MEF.* https://www.mef.net/service-standards/overlay-services/sd-wan/ (cit. on p. 21).

[5] *Kathara.* https://www.kathara.org/ (cit. on p. 36).