

MÁSTER UNIVERSITARIO EN INGENIERÍA DE
TELECOMUNICACIONES

TRABAJO FIN DE MÁSTER

ACTUALIZACIÓN DE DYNPAC PARA INTEGRACIÓN CON SN4I (SMART NETWORKS FOR INDUSTRY) E INCORPORACIÓN DE NUEVOS SERVICIOS DE RED



Estudiante: Plaza Vas, Isabel

Directora: Higuero Aperribai, María Victoria

RESUMEN

En la actualidad, es innegable la creciente necesidad y dependencia de las redes y servicios de telecomunicaciones en todos los ámbitos. La evolución constante de las tecnologías de comunicación es fundamental para satisfacer las demandas de tráfico y las necesidades de servicio cada vez más exigentes.

En este contexto, el escenario *Smart Networks for Industry* (SN4I) es una red basada en tecnologías como la Virtualización de las Funciones de Red (NFV) y las Redes Definidas por *Software* (SDN), que permite la investigación y experimentación en comunicaciones para la Industria 4.0. SN4I está desplegada y operativa entre diversas sedes de la Universidad UPV/EHU en Bilbao y Leioa y el Centro de Fabricación Avanzada Aeronáutica (CFAA) en Zamudio.

Como parte de SN4I, la aplicación DynPaC ofrecía hace pocos años un servicio para el cálculo dinámico de rutas y el aprovisionamiento de recursos de red sobre esta infraestructura SDN. Sin embargo, debido a la evolución del escenario SN4I para adaptarse a las tecnologías más punteras de comunicación, la aplicación DynPaC ha quedado obsoleta para operar bajo los nuevos requerimientos y prestaciones que exige el escenario SN4I.

Ante esta problemática, este Trabajo Fin de Máster (TFM) propone una solución para la integración de DynPaC con la infraestructura de SN4I y la incorporación de nuevos servicios de red a este escenario, permitiendo así dotarlo de la capacidad de soportar las últimas tecnologías de comunicación y garantizar un entorno de investigación y experimentación de vanguardia.

Palabras clave: Redes Definidas por Software (SDN), Virtualización de las Funciones de Red (NFV), Ancho de Banda Bajo Demanda (BoD), Programación en el Plano de Datos (DPP), telemetría de red.

ABSTRACT

Currently, the escalating necessity and reliance on telecommunications networks and services in all domains are indisputable. The continual evolution of communication technologies is pivotal in meeting the ever-increasing demands of traffic and the increasingly exigent service requirements.

Within this context, the Smart Networks for Industry (SN4I) framework represents a network founded on cutting-edge technologies such as Network Function Virtualization (NFV) and Software-Defined Networking (SDN). It facilitates research and experimentation in communications for the Industry 4.0 sector. SN4I has been deployed and is operational across various campuses of the UPV/EHU University in Bilbao and Leioa, as well as the Advanced Aeronautical Manufacturing Center (CAFA) in Zamudio.

Previously, as part of SN4I, the DynPaC application provided a service for dynamic route calculation and network resource provisioning on this SDN infrastructure. Nevertheless, due to the evolution of the SN4I framework to accommodate the most advanced communication technologies, the DynPaC application has become outdated and incapable of meeting the new requirements and performance expectations demanded by the SN4I framework.

Given these circumstances, this Master's Thesis proposes a solution for integrating DynPaC with the SN4I infrastructure and incorporating new network services into this framework. This integration will empower SN4I with the ability to support the latest communication technologies and establish a cutting-edge environment for research and experimentation.

Keywords: *Software Defined Networks (SDN), Network Function Virtualization (NFV), Bandwidth on Demand (BoD), Data Plane Programming (DPP), network telemetry.*

LABURPENA

Gaur egun, ukaezina da telekomunikazio-sare eta -zerbitzuen beharra eta mendekotasuna handituz doala esparru guztietan. Komunikazio-teknologiaren etengabeko bilakaera funtsezkoa da trafiko eskariak eta gero eta exigenteagoak diren zerbitzu beharrak asetzeko.

Testuinguru honetan, Smart Networks for Industry (SN4I) teknologietan oinarrituta dagoen sare bat da, besteak beste Sare-Funtzioen Birtualizazioan (NFV) eta Softwareak Definitutako Sareetan (SDN). Honi esker 4.0 Industriarako komunikazioetan ikertu eta esperimentatu egin daiteke. SN4I sarea UPV/EHUko Bilboko eta Leioako egoitzen eta Zamudioko Aeronautikako Fabrikazio Aurreratuko Zentroaren (CFAA) artean hedatuta eta erabilgarri dago.

SN4Iren atal gisa, DynPaC aplikazioak duela urte batzuk SDN azpiegituraren gainean zerbitzu bat eskaintzen zuen ibilbideen kalkulo dinamikorako eta sare-baliabideen hornikuntzarako. Hala ere, punta-puntako komunikazio-teknologietara egokitzeko SN4I izan duen bilakaera dela eta, DynPaC aplikazioa zaharkituta geratu da SN4I agertokiak eskatzen dituen errekerimendu eta prestazio berriekin jarduteko.

Problematika honen aurrean, Master Amaierako Lan (TFM) honek irtenbide bat proposatzen du DynPaC SN4Iren azpiegiturarekin integrazteko eta sare-zerbitzu berriak gehitzeko egoera honetara. Honela, azkeneko komunikazio-teknologiak jasateko gaitasuna izango du, eta abangoardiako ikerketa- eta esperimentazio-ingurunea bermatuz.

Hitz gakoak: *Softwareak Definitutako Sareak (SDN), Sare-Funtzioen Birtualizazioa (NFV), Eskariaren Araberako Banda-Zabalera (BoD), Datu-Planoko Programazioa (DPP), sare-telemetrya.*

ÍNDICE DE CONTENIDO

1.	INTRODUCCIÓN	10
2.	CONTEXTO TECNOLÓGICO Y ESCENARIO DEL TFM	12
2.1.	Contexto tecnológico	12
2.1.1.	Redes Definidas por Software (SDN)	12
2.2.	Escenario del TFM	17
2.2.1.	<i>Dynamic Path Computation</i> (DynPaC)	19
2.2.2.	Evolución de SN4I	20
3.	OBJETIVOS Y ALCANCE	22
4.	BENEFICIOS	24
4.1.	Beneficios técnicos	24
4.2.	Beneficios económicos	25
4.3.	Beneficios sociales	26
5.	ANÁLISIS DE ALTERNATIVAS	27
5.1.	Estrategia de desarrollo	27
5.1.1.	Alternativas para la estrategia de desarrollo	27
5.1.2.	Selección de la alternativa óptima para la estrategia de desarrollo	28
5.2.	Técnica de migración	30
5.2.1.	Alternativas para la técnica de migración	30
5.2.2.	Selección de la alternativa óptima para la técnica de migración	31
5.3.	Tipo de implementación	32
5.3.1.	Alternativas para el tipo de implementación	32
5.3.2.	Selección de la alternativa óptima para el tipo de implementación	33
6.	ANÁLISIS DE RIESGOS	35
6.1.	Riesgos externos	35
6.2.	Riesgos internos	35
6.3.	Análisis de riesgos	36
6.4.	Plan de prevención	37
7.	DESCRIPCIÓN DE LA SOLUCIÓN	38
7.1.	Introducción	38
7.2.	Arquitectura DynPaC	39
7.2.1.	Módulos	39
7.2.2.	Interfaz con elementos externos	44

7.3.	Funcionamiento de DynPaC en SN4I	45
7.4.	Diseño de la solución	47
7.4.1.	Visión general	47
7.4.2.	Diseño de las fases de migración	49
7.5.	Implementación de la solución.....	54
7.5.1.	Plataforma de desarrollo	54
7.5.2.	Implementación de las fases de la migración.....	56
7.6.	Pruebas de validación.....	63
7.6.1.	Escenario diseñado para las pruebas	64
7.6.2.	Definición de las pruebas	65
7.6.3.	Resultados de las pruebas.....	66
7.6.4.	Conclusiones de los resultados.....	77
8.	METODOLOGÍA	78
8.1.	Equipo de proyecto	78
8.2.	Descripción de tareas y paquetes de trabajo	79
8.3.	Diagrama de Gantt	85
9.	ANÁLISIS DE COSTES	87
9.1.	Horas internas.....	87
9.2.	Amortizaciones	87
9.3.	Gastos.....	88
9.4.	Subcontrataciones	88
9.5.	Coste total del proyecto.....	89
10.	CONCLUSIONES.....	90
	BIBLIOGRAFÍA	92

ÍNDICE DE FIGURAS

<i>Figura 2.1: Arquitectura genérica de una SDN.</i>	13
<i>Figura 2.2: Visión general de la plataforma de ONOS.</i>	14
<i>Figura 2.3: Arquitectura de SN4I.</i>	17
<i>Figura 6.1: Matriz Probabilidad-Impacto de los riesgos del proyecto.</i>	36
<i>Figura 7.1: Arquitectura modular de DynPaC.</i>	40
<i>Figura 7.2: Arquitectura DynPaC como aplicación de ONOS.</i>	46
<i>Figura 7.3: Código modificado en el archivo pom.xml.</i>	56
<i>Figura 7.4: Código modificado en la clase DefaultDynpacTopologyAbstraction.java.</i>	57
<i>Figura 7.5: Código modificado en la clase DynpacTopologyAbstraction.java.</i>	58
<i>Figura 7.6: Dependencia Minimal JSON añadida en el archivo pom.xml.</i>	58
<i>Figura 7.7: Código modificado en la clase DynpacPce.java.</i>	59
<i>Figura 7.8: Código modificado en el archivo pom.xml.</i>	61
<i>Figura 7.9: Código modificado en DynpacConfigurationDefaultVlanManage.java.</i>	61
<i>Figura 7.10: Código modificado en DynpacConfigurationExceptionVlanManage.java.</i>	61
<i>Figura 7.11: Código modificado en el archivo pom.xml.</i>	62
<i>Figura 7.12: Código modificado en el archivo pom.xml.</i>	63
<i>Figura 7.13: Código modificado en el archivo pom.xml.</i>	63
<i>Figura 7.14: Escenario diseñado para las pruebas de validación.</i>	64
<i>Figura 7.15: VLANs configuradas en el escenario diseñado para las pruebas de validación.</i>	65
<i>Figura 7.16: Resultado de la compilación de DynPaC compatible con ONOS v1.15.0.</i>	67
<i>Figura 7.17: Aplicación DynPaC instalada y activa en ONOS v1.15.0.</i>	67
<i>Figura 7.18: Alta de un servicio DynPaC en ONOS v1.15.0.</i>	68
<i>Figura 7.19: Servicio DynPaC activo en ONOS v1.15.0.</i>	68
<i>Figura 7.20: Flujos instalados por el servicio DynPaC.</i>	69
<i>Figura 7.21: Resultado de la compilación de DynPaC compatible con ONOS v2.0.0.</i>	70
<i>Figura 7.22: Aplicación DynPaC instalada y activa en ONOS v2.0.0.</i>	71
<i>Figura 7.23: Alta de un servicio DynPaC en ONOS v2.0.0.</i>	71
<i>Figura 7.24: Servicio DynPaC activo en ONOS v2.0.0.</i>	72
<i>Figura 7.25: Flujos instalados por el servicio DynPaC.</i>	72
<i>Figura 7.26: Resultado de la compilación de DynPaC compatible con ONOS v2.1.0.</i>	73
<i>Figura 7.27: Aplicación DynPaC instalada y activa en ONOS v2.1.0.</i>	74
<i>Figura 7.28: Alta de un servicio DynPaC en ONOS v2.1.0.</i>	75

<i>Figura 7.29: Servicio DynPaC activo en ONOS v2.1.0.</i>	75
<i>Figura 7.30: Flujos instalados por el servicio DynPaC.</i>	76
<i>Figura 8.1: Paquetes de trabajo del proyecto (WBS).</i>	79

ÍNDICE DE TABLAS

<i>Tabla 5.1: Matriz de evaluación de alternativas para la estrategia de desarrollo.....</i>	<i>29</i>
<i>Tabla 5.2: Matriz de evaluación de alternativas para la técnica de migración.</i>	<i>32</i>
<i>Tabla 5.3: Matriz de evaluación de alternativas para el tipo de implementación.....</i>	<i>34</i>
<i>Tabla 8.1: Equipo de trabajo del proyecto.....</i>	<i>78</i>
<i>Tabla 8.2: Paquete de tareas P.T.1. del proyecto.</i>	<i>80</i>
<i>Tabla 8.3: Paquete de tareas P.T.2. del proyecto.</i>	<i>81</i>
<i>Tabla 8.4: Paquete de tareas P.T.3. del proyecto.</i>	<i>82</i>
<i>Tabla 8.5: Paquete de tareas P.T.4. del proyecto.</i>	<i>83</i>
<i>Tabla 8.6: Paquete de tareas P.T.5. del proyecto.</i>	<i>84</i>
<i>Tabla 8.7: Parámetros generales de la planificación del proyecto.</i>	<i>85</i>
<i>Tabla 8.8: Hitos del proyecto.</i>	<i>85</i>
<i>Tabla 9.1: Partida de horas internas del proyecto.....</i>	<i>87</i>
<i>Tabla 9.2: Coste unitario de las amortizaciones del proyecto.</i>	<i>87</i>
<i>Tabla 9.3: Partida de amortizaciones del proyecto.</i>	<i>88</i>
<i>Tabla 9.4: Partida de gastos del proyecto.....</i>	<i>88</i>
<i>Tabla 9.5: Partida de subcontrataciones del proyecto.</i>	<i>88</i>
<i>Tabla 9.6: Resumen de los costes del proyecto.</i>	<i>89</i>

1. INTRODUCCIÓN

En la actualidad, resulta innegable la creciente necesidad y dependencia de redes y servicios de telecomunicaciones en cualquier ámbito profesional, económico y social.

Estas redes constituyen elementos fundamentales en la era digital, permitiendo la transferencia instantánea de información y facilitando la interacción global entre individuos, organizaciones y entidades gubernamentales.

En consecuencia, la constante evolución de las tecnologías ha sido y es imperativa para satisfacer las demandas de tráfico cada vez más intensas y las necesidades de servicio cada vez más exigentes.

Asimismo, resulta imprescindible que las redes se adapten a velocidades cada vez mayores, mediante la implementación de mejoras en capacidad, eficiencia y seguridad. De esta forma, se garantiza un rendimiento óptimo y una experiencia ininterrumpida en este entorno interconectado y en constante progreso.

Para lograr todos estos avances, la investigación juega un papel clave a la hora de desarrollar nuevas tecnologías que soporten los requerimientos exigidos en cada momento.

En este contexto, el escenario *Smart Networks for Industry* (SN4I) es una red de comunicaciones basada en la Virtualización de las Funciones de Red (NFV, de sus siglas en inglés) y las Redes Definidas por *Software* (SDN, de sus siglas en inglés). SN4I conecta varias sedes de la Universidad del País Vasco (UPV/EHU) con el Centro de Fabricación Avanzada Aeronáutica (CFAA) de Zamudio, creando un entorno propicio para la investigación y experimentación en el ámbito de las comunicaciones en la Industria 4.0.

En este escenario, el Grupo de Investigación en Ingeniería Telemática (I2T), perteneciente a la Escuela de Ingeniería de Bilbao de la Universidad del País Vasco/Euskal Herriko Unibertsitatea (UPV/EHU) ha desarrollado y desarrolla actualmente diversos proyectos de investigación relacionados con la optimización del funcionamiento y operación de las SDN y NFV, la aplicación de técnicas de Ingeniería de Tráfico y la seguridad en la operación de las mismas.

Como parte de SN4I, la aplicación DynPaC permite el cálculo dinámico de rutas y el aprovisionamiento de recursos de red sobre esta infraestructura SDN.

Sin embargo, debido a la evolución constante de las telecomunicaciones, el escenario SN4I ha ido evolucionando y demandando nuevos requerimientos técnicos para adaptarse a las tecnologías más punteras de comunicación.

Como consecuencia de esta evolución, la aplicación DynPaC ha quedado obsoleta para operar bajo los nuevos requisitos y prestaciones que exige el escenario SN4I.

En este contexto, el objetivo principal de este Trabajo Fin de Máster (TFM) es lograr la compatibilidad entre DynPaC y la infraestructura de SN4I, así como dotar a este escenario de investigación y experimentación de la capacidad de soportar las tecnologías de comunicación más recientes, tales como la Programabilidad en el Plano de Datos (DPP, de sus siglas en inglés) y la monitorización de red basada en telemetría.

El presente documento constituye el informe final del mencionado TFM.

2. CONTEXTO TECNOLÓGICO Y ESCENARIO DEL TFM

En el presente apartado se describen los aspectos más relevantes de las tecnologías relacionadas con este TFM.

2.1. Contexto tecnológico

2.1.1. Redes Definidas por Software (SDN)

Las Redes Definidas por *Software* (SDN, de sus siglas en inglés) surgen como una nueva concepción de arquitectura de red, en respuesta al auge de las aplicaciones de red de naturaleza dinámica y a las cada vez más exigentes necesidades y demandas de los usuarios en cuanto a servicios de comunicaciones. Partiendo de esta visión, se define esta arquitectura de red, dinámica y adaptable, que constituyen las SDN.

Las SDN son redes en las que el *software* se encuentra desvinculado del *hardware*. La principal característica de SDN es la separación de los dos planos de los dispositivos de red:

- **Plano de control:** Se traslada el *software* a un plano de control centralizado, que establece dónde y cómo enviar el tráfico.

De esta manera, la visión y gestión de la red se centraliza en el plano de control, que obtiene una perspectiva global de la red y adquiere el poder y capacidad de decisión sobre la configuración y el funcionamiento del resto de la red.

- **Plano de datos:** Se deja en el *hardware* el plano de datos, que realiza la acción de reenviar el tráfico.

Así, la complejidad del plano de la red encargado de la transmisión de tráfico (plano de datos) se reduce significativamente. Los dispositivos de red que constituyen este plano se simplifican en su conjunto, puesto que no requieren de lógica específica para cada protocolo que sean capaces de procesar.

Además de estos dos planos, en esta arquitectura se incluye una capa de aplicación, que implementa la siguiente función:

- **Capa de aplicación:** En ella se incluyen un conjunto de aplicaciones y funciones de red que comunican instrucciones específicas al plano de control, como solicitudes de recursos o información sobre el estado o configuración de la red. Para ello se utilizan las Interfaces de Programación de Aplicaciones (API, de sus siglas en inglés), que pueden desempeñar diversas funciones.

Esta división en capas de la arquitectura de red permite que el operador de una red SDN disponga de capacidad para programar y controlar toda la red desde un solo punto de control lógico en lugar de hacerlo dispositivo por dispositivo. Es decir, la abstracción del plano de control sobre el plano de datos permite una visión global y centralizada de la red, a la vez que reduce sustancialmente su complejidad y facilita su programabilidad y monitorización.

A continuación, en la Figura 2.1, se muestra un diagrama de la arquitectura SDN, en el que se puede observar la composición de la red en diferentes capas, así como la interconexión entre ellas.

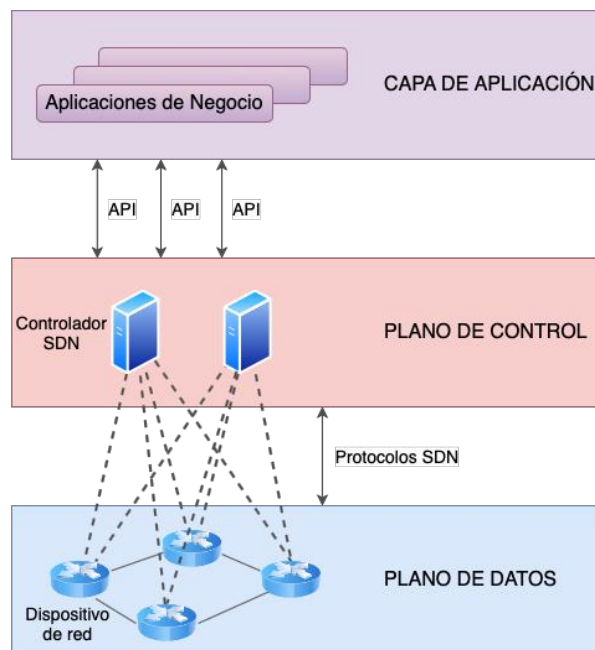


Figura 2.1: Arquitectura genérica de una SDN.

Migrar la capa de control, tradicionalmente muy vinculada a los dispositivos de red, a un punto de control centralizado permite abstraer la estructura subyacente para aplicaciones y servicios de red, pudiendo así tratarla como una entidad lógica o virtual. De esta manera, SDN no sólo facilita el diseño y la operación de la red, sino que también simplifica notablemente los dispositivos de red en sí mismos, ya que, con esta nueva arquitectura, no requieren capacidad de comprender y procesar múltiples estándares de protocolo; sino que únicamente deben ser capaces de obedecer las instrucciones de los controladores SDN.

2.1.1.1. Plano de control: ONOS

En las SDN, hay gran diversidad de opciones a la hora de implementar el plano de control. Entre ellas, se encuentra ONOS: una de las alternativas más utilizadas en la actualidad en despliegues reales. [1]

La palabra ONOS corresponde a las siglas de *Open Network Operating System*. La función de ONOS es proporcionar el plano de control para una SDN, controlando el funcionamiento de los componentes de red como *switches*, *routers* y enlaces y ejecutando programas o módulos *software* para proporcionar servicios de comunicación a los equipos finales y redes adyacentes.

ONOS es, en definitiva, lo que se entiende por un controlador SDN. En concreto, es el controlador SDN de código abierto líder actualmente para crear soluciones SDN, ya que ofrece servicios personalizados para la supervisión y gestión de red, además de aplicaciones de encaminamiento, para todo tipo de Redes Definidas por *Software*.

En la Figura 2.2, se muestra un diagrama de la plataforma que proporciona el sistema ONOS. Como puede observarse en dicha Figura, ONOS ofrece desde protocolos de gestión de dispositivos de red hasta un conjunto de aplicaciones que actúan de forma escalable, modular y distribuida.

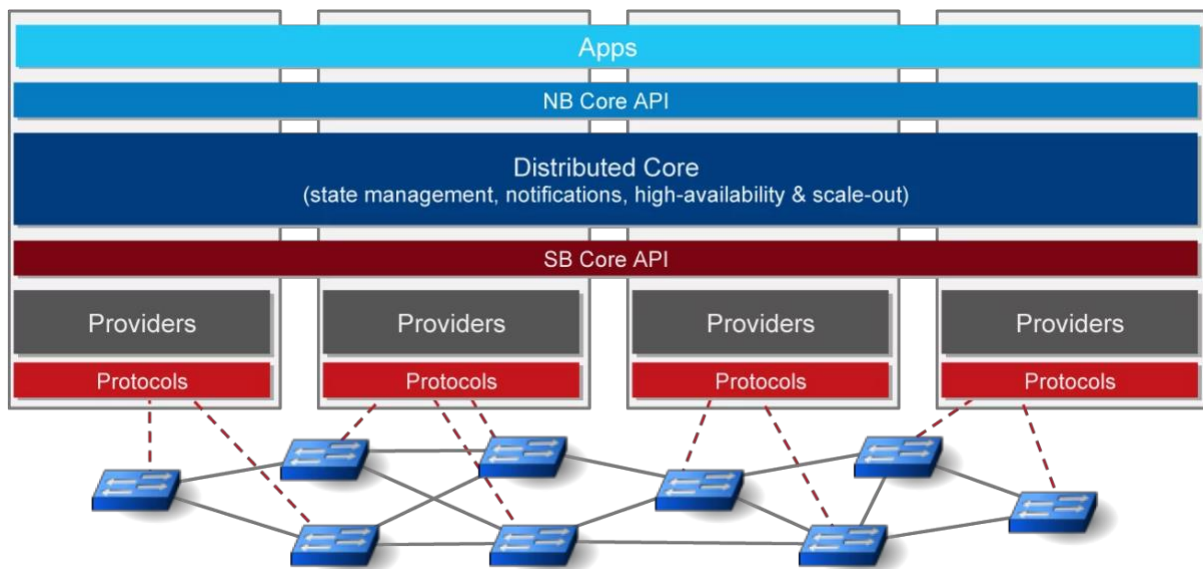


Figura 2.2: Visión general de la plataforma de ONOS.

Concretamente, para este proyecto es crucial la funcionalidad que ofrece ONOS de ejecutar programas o módulos *software* para proporcionar servicios de comunicación a los equipos finales y redes adyacentes. A través de su API (*NB Core API*), ONOS permite interconectar su núcleo (*Distributed Core*) con aplicaciones personalizadas de servicios de red (*Apps*).

En la plataforma de ONOS (véase la Figura 2.2):

- Por un lado, los proveedores (*Providers*) son responsables de interactuar con la red utilizando varios protocolos de control y configuración (*Protocols*) y de suministrar dichos datos al núcleo de ONOS (*Distributed Core*).
- Por otro lado, el núcleo de ONOS (*Distributed Core*) es responsable de recopilar la información sobre el estado de la red y distribuirla a las aplicaciones (*Apps*), ya sea de forma síncrona a través de consultas (*queries*) o asíncrona a través de *callbacks*.

De esta forma, las aplicaciones proporcionan servicios específicos de red, pudiendo monitorizar la red, obtener datos en tiempo real de la misma y tomar decisiones para personalizar su funcionamiento en base a los datos obtenidos.

2.1.1.2. Plano de datos: OpenFlow

OpenFlow constituye un protocolo para la comunicación entre el plano de control y el plano de datos de las SDN. Asimismo, tal y como ya se ha mencionado, permite controlar las tablas de *forwarding* de los *switches*, *routers* y puntos de acceso de la red.

OpenFlow también incluye una API del grupo ONF (*Open Networking Foundation*) que permite abstraer las funcionalidades de capa 2 y capa 3 en *routers* y *switches*, de manera que el control de dichas funcionalidades reside en el *software*, mientras las acciones de reenvío permanecen en el *hardware*.

OpenFlow discrimina el tráfico en base a flujos. En consecuencia, un *switch* OpenFlow reenvía los paquetes según su tabla de flujo (*flow table*), la cual puede tener varias entradas que determinan cómo procesar dichos paquetes. Una entrada de una tabla de flujo OpenFlow contiene:

- **Campos de coincidencia (*Match fields*):** Especifica el puerto y las cabeceras del paquete con las que se debe contrastar el tráfico entrante.
- **Prioridad:** Indica la prioridad de la entrada en la tabla en caso de que el paquete entrante coincida (*match*) con varias entradas de la tabla.
- **Contadores:** Su valor se actualiza cuando se encuentra una coincidencia (*match*).
- **Instrucciones:** Indican la acción o conjunto de acciones a realizar en caso de coincidencia (*match*).
- **Temporizadores:** Indican el tiempo máximo antes de que el *switch* descarte una entrada de la tabla de flujos.
- **Cookie:** Es el valor que elige el controlador para filtrar las estadísticas, las modificaciones y el borrado de los flujos. No se utiliza durante el procesado de los paquetes.

En el caso de que un *switch* reciba tráfico que no coincida con ninguna de las entradas de su tabla de flujo, enviará el paquete al controlador (entrada por defecto) para que éste gestione qué debe hacer con él. El controlador responderá al *switch* y le especificará la entrada que debe añadir en su tabla para dicho flujo. Esta comunicación se lleva a cabo a través de los mensajes OpenFlow mencionados a continuación.

- **Packet-In:** Es el paquete que un *switch* envía al controlador cuando el paquete recibido por el *switch* no coincide con ninguna de las entradas de su tabla.
- **Packet-Out:** Es el paquete que el controlador envía a uno o más puertos del *switch*.
- **Flow-Mod:** Es el paquete que el controlador envía al *switch* para especificarle qué entrada en particular debe añadir a su tabla de flujo.

2.2. Escenario del TFM

Smart Networks for Industry (SN4I) [2] es una red de comunicaciones compatible con la Virtualización de las Funciones de Red (NFV, de sus siglas en inglés) y SDN, que interconecta dos sedes, Bilbao y Leioa, de la Universidad del País Vasco (UPV/EHU), con el Centro de Fabricación Avanzada Aeronáutica (CFAA) de Zamudio. En este contexto, NFV proporciona una forma flexible de desplegar servicios virtuales, mientras que SDN permite programar elementos de red para interconectar estos servicios virtuales mediante enlaces o redes virtuales. A continuación, en la Figura 2.3, se muestra un diagrama de la arquitectura de SN4I.

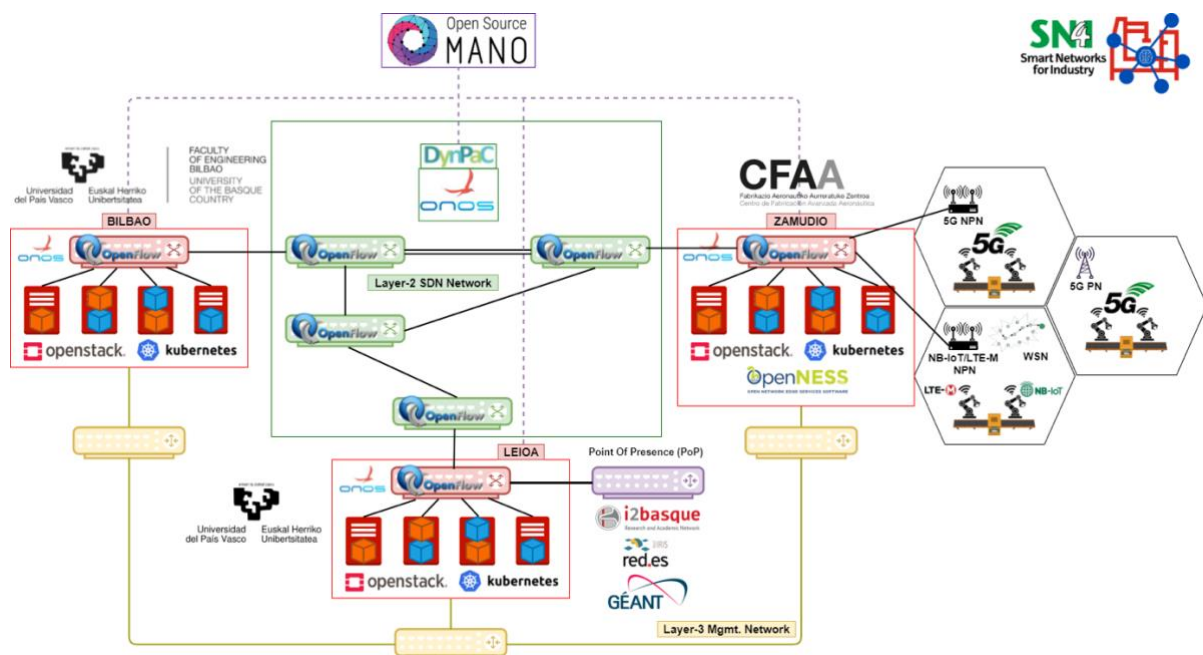


Figura 2.3: Arquitectura de SN4I.

Como resultado, SN4I proporciona recursos de red, computación y almacenamiento segmentados para el despliegue de servicios concurrentes aislados, así como para la creación de escenarios de experimentación. Este despliegue también permite la integración de tecnologías NFV con los protocolos del Internet industrial y la coexistencia con otras redes industriales como las *Time Sensitive Networks* (TSN).

SN4I complementa la investigación y la actividad tradicionales en el campo de la fabricación del CFAA, con tecnologías de vanguardia como la creación bajo demanda de servicios de gran ancho de banda y bajo retardo, la seguridad basada en SDN, etc. La integración de todas estas nuevas tecnologías con las actuales máquinas, herramientas y procesos de fabricación de última generación permite disponer de un centro mixto de I+D+i en tecnologías avanzadas de fabricación que constituye un referente a nivel internacional.

Esta infraestructura capitaliza el conocimiento acumulado en la anterior instalación, la OpenFlow Enabled Facility (EHU-OEF) de la UPV/EHU, desplegada por el Grupo I2T de esta Universidad, y representa la evolución de la aplicación de las tecnologías SDN y NFV a un caso de uso vertical (Industria 4.0), pero también a la Ciberseguridad y al despliegue de Servicios Genéricos de Red.

En 2021, SN4I se incluyó en la infraestructura SLICES-RI de la hoja de ruta ESFRI, que es una infraestructura de investigación a escala europea totalmente programable y virtualizada, accesible de forma remota, diseñada para apoyar las actividades de investigación experimental a gran escala, en temas relacionados con arquitecturas y servicios de red innovadores. La investigación SN4I SLICES-RI se dirige específicamente a la experimentación 5G Avanzada y 6G aplicada a la Fabricación (Industria 4.0 y más allá). El 1 de septiembre de 2022 se puso en marcha el proyecto SLICES-PP (Fase Preparatoria) (HORIZON-INFRA-2021-DEV-02), de 39 meses de duración, para preparar la infraestructura.

En 2022, SN4I se integró en Fed4FIRE+, una infraestructura federada en toda Europa que proporciona instalaciones de ensayo interoperables, abiertas, accesibles y fiables a investigadores e innovadores.

Tal y como se puede observar en la figura anteriormente adjunta (véase la Figura 2.3), la arquitectura de red de SN4I sigue el modelo previamente descrito en el apartado Redes Definidas por Software (SDN) de este documento. Así, la arquitectura de red se divide en tres estratos:

- **Plano de control:** ONOS proporciona el plano de control a la red de SN4I, controlando componentes de red como *switches*, *routers* y enlaces y ejecutando programas o módulos *software* para proporcionar servicios de comunicación a las tres sedes de SN4I (Bilbao, Leioa y CFAA).
- **Plano de datos:** OpenFlow proporciona el plano de datos a la red, abstrayendo las funcionalidades de capa 2 y capa 3 en los equipos de red (*routers* y *switches*), de manera que el control de dichas funcionalidades reside en el *software*, mientras las acciones de reenvío permanecen en el *hardware*.

Asimismo, OpenFlow constituye un protocolo para la comunicación entre el plano de control y el plano de datos de la SDN.

- **Capa de aplicación:** Se han incluido funcionalidades que facilitan la adopción de estrategias de Ingeniería de Tráfico (TE, de sus siglas en inglés) en SDN a través de la aplicación denominada DynPaC, que se describe en el siguiente subapartado.

2.2.1. *Dynamic Path Computation (DynPaC)*

Tal y como se ha mencionado en el anterior subapartado, en la capa de aplicación de la arquitectura SN4I, se encuentra la aplicación denominada DynPaC.

DynPaC es el acrónimo para Cálculo Dinámico de Rutas (del inglés, *Dynamic Path Computation*). DynPaC consiste en un *framework* genérico que permite el aprovisionamiento de servicios de nivel de enlace (capa 2) sobre una infraestructura OpenFlow SDN, haciendo posible la reserva de servicios con limitaciones de ancho de banda dentro de la franja horaria deseada, ya sea ésta presente o futura.

En definitiva, DynPaC facilita la adopción de estrategias de Ingeniería de Tráfico (TE, de sus siglas en inglés) en Redes Definidas por *Software*.

DynPaC es capaz de realizar complejos cálculos de ruta extremo a extremo, para así seleccionar y asignar la mejor ruta a cada servicio en función de sus necesidades de ancho de banda. Todo ello teniendo en cuenta los siguientes parámetros:

- **Ancho de banda:** La capacidad disponible en la red, tanto en los enlaces como en los nodos de red.
- **Número de saltos:** El número de saltos entre los puntos finales de la comunicación.
- **Estabilidad:** La cualidad de la ruta calculada de permanecer estable ante posibles perturbaciones en la red.

Asimismo, DynPaC tiene en cuenta la importancia de aislar los diferentes servicios de red entre sí para proporcionar seguridad a los mismos. Para ello, encapsula el tráfico de los servicios en diferentes VLAN.

El *framework* de DynPaC está diseñado para proporcionar la posibilidad de ser instalado como una aplicación en ONOS. Gracias a ello, DynPaC puede trabajar con un controlador ONOS mientras reside en el plano de aplicación de la arquitectura SDN. De esta manera, DynPaC es capaz de:

- Crear servicios de nivel de enlace sobre la infraestructura gestionada por el controlador ONOS.
- Intercambiar información e interactuar con ONOS para instalar las correspondientes reglas de flujo, que representan cada servicio creado, en los conmutadores OpenFlow.

2.2.2. Evolución de SN4I

Tal y como se ha mencionado previamente, SN4I constituye una infraestructura dedicada a la investigación y creación de escenarios de experimentación. En consecuencia, es crucial que SN4I soporte las tecnologías y servicios de red más novedosos y de última generación. La arquitectura de red SDN, donde la inteligencia reside en el plano de control, facilita sustancialmente esta tarea. Gracias a ONOS, el controlador SDN de *software* libre actualmente líder en el mercado [1], es posible mantener el plano de control de SN4I actualizado y compatible en todo momento con las tecnologías de vanguardia.

Es importante destacar que ONF, la fundación a cargo de ONOS, proporciona una nueva versión actualizada del *software* para el controlador ONOS cada 4-6 meses aproximadamente [3]. De esta manera, ONOS incorpora nuevas mejoras y funcionalidades regularmente, adaptándose a la evolución constante de las tecnologías de red y comunicación.

Algunas de las recientes mejoras e incorporaciones de ONOS más relevantes para las actividades llevadas a cabo en SN4I se refieren a aspectos tales como los mencionados a continuación [4]:

- **Mejoras en la monitorización de la red:** Se incorpora una tecnología de monitorización en banda, denominada INT, que permite la monitorización de aspectos avanzados de la operación y estado de la red desde el propio plano de datos. De esta manera, aporta una fiabilidad y una precisión superiores a las aportadas por otras tecnologías, lo que resulta de gran valor para la optimización del funcionamiento de muchos escenarios de red.
- **Programabilidad del Plano de Datos (DPP, de sus siglas en inglés):** Esta capacidad permite programar y *customizar* la forma en la que los dispositivos del plano de datos (*switches, routers...*) deben procesar los paquetes.
- **Mejoras en la interfaz gráfica de usuario:** Se han desarrollado algunas mejoras significativas en la herramienta en cuanto a aspectos de visualización e interacción con el usuario. Algunas de estas mejoras se refieren a aspectos como mejora de la vista topológica de la red, posibilidad de selección múltiple de nodos, incorporación de actualizaciones dinámicas de la topología, etc.

Todas estas mejoras mencionadas previamente suponen dotar a SN4I de capacidad para soportar dichas mejoras únicamente actualizando la versión del *software* de ONOS utilizada.

Sin embargo, en la capa de aplicación, donde opera DynPaC, esta actualización genera el conflicto descrito a continuación.

Como se ha destacado anteriormente en el subapartado *Dynamic Path Computation* (DynPaC) de este documento, una de las características más ventajosas de DynPaC es la posibilidad que proporciona de ser instalado como una aplicación en ONOS. Gracias a ello, es capaz de crear servicios sobre la infraestructura gestionada por el controlador ONOS e intercambiar información con el controlador para instalar las correspondientes reglas de flujo, que representan cada servicio creado, en los *switches* OpenFlow.

Cuando DynPaC se desarrolló en 2015-2017, la aplicación era plenamente compatible con las versiones de ONOS de entonces. En cambio, a medida que ONF fue publicando nuevas versiones del *software* de ONOS, DynPaC fue quedándose obsoleto, dejando de ser compatible con las versiones posteriores del controlador.

En este contexto y con el objetivo de la incorporación de las mejoras incluidas en las últimas versiones de ONOS, surge la necesidad de lograr la compatibilidad de DynPaC con versiones recientes de ONOS, de tal modo que se mantenga tanto la infraestructura de red de SN4I completa como su compatibilidad con las tecnologías de comunicación más novedosas.

3. OBJETIVOS Y ALCANCE

Como se ha destacado en el apartado anterior, las prestaciones de la aplicación DynPaC se consideran de gran interés para su incorporación en el escenario SN4I.

En consecuencia, el objetivo principal de este proyecto es la integración de DynPaC con el escenario SN4I y la incorporación de nuevos servicios de red a dicho escenario.

Es decir, de forma más específica, en el proyecto objeto de este TFM se buscan las características descritas a continuación:

- **Compatibilidad de DynPaC con SN4I:** Es el objetivo principal del proyecto. Integrar DynPaC como capa de aplicación en la arquitectura de red que proporciona SN4I permite dotar al escenario de red SN4I del aprovisionamiento de servicios de nivel de enlace, facilitando así la utilización de estrategias de TE. DynPaC como aplicación incorpora al escenario SN4I la posibilidad de calcular y seleccionar la mejor ruta extremo a extremo para cada servicio en función de sus necesidades de ancho de banda, y a su vez aislar los servicios entre sí para proporcionar seguridad a los mismos.

Como objetivos adicionales, destacan la incorporación de los servicios de red expuestos a continuación:

- **Monitorización de red basada en telemetría:** Hoy en día es inconcebible prescindir de la monitorización de la red con la que se opera. En escenarios SDN, como es el caso de SN4I, es muy común el uso de herramientas de telemetría para llevar a cabo la monitorización de la red. Estas herramientas proporcionan datos del estado de la red que permiten conocer en todo momento las circunstancias (de temperatura, alimentación eléctrica, uso del ancho de banda, estado de los enlaces...) bajo las que se encuentra dicha red, permitiendo así monitorizarla. Es por esto por lo que la monitorización de red basada en telemetría y, en consecuencia, la capacidad de incorporar herramientas de telemetría es otro de los objetivos de este proyecto.
- **Programabilidad en el plano de datos (DPP, de sus siglas en inglés):** El auge de los planos de datos programables los ha convertido en uno de los temas de investigación principales que se tratan en el grupo I2T dentro de la línea de investigación de las SDN. En consecuencia, es interesante dotar al escenario SN4I de la capacidad de soportar estos planos de datos programables. Así, éste constituye otro de los objetivos del proyecto.

Además de los objetivos descritos anteriormente, para este proyecto se han definido una serie de objetivos secundarios. Dichos objetivos secundarios son los expuestos a continuación.

- **Modularidad:** Un sistema basado en módulos facilita que la solución final esté mejor estructurada. Como consecuencia de esto, se consigue que la implementación de los diferentes elementos se pueda realizar con facilidad. De forma adicional, se pueden añadir, modificar y/o eliminar funcionalidades dependiendo de las necesidades de cada escenario.

Cabe mencionar también que un sistema modular facilita actualizaciones como las que se plantean en este TFM.

- **Flexibilidad:** Es conveniente que la solución proporcionada sea capaz de adaptarse a los distintos posibles escenarios que tengan lugar en la red SN4I. De esta manera, se consigue un escenario de investigación ideal para llevar a cabo pruebas de investigación relacionadas con tecnologías de red innovadoras como con las que trabaja el grupo I2T.

Todos los objetivos descritos anteriormente hacen referencia al diseño de propuesta de la solución. Complementariamente, otro de los objetivos adicionales de este proyecto es el mencionado a continuación:

- **Desarrollo de una plataforma de desarrollo que permita analizar la viabilidad de la propuesta desarrollada:** Una vez elaborada la solución propuesta, es interesante analizar su viabilidad. Para ello, resulta muy útil contar con una plataforma de desarrollo que permita realizar diversas pruebas que sirvan para verificar la viabilidad de la propuesta desarrollada.

En definitiva, este proyecto incluye el diseño, desarrollo e implementación de la solución que permita alcanzar los objetivos propuestos. El diseño de la solución deberá realizarse acorde con las funcionalidades y características requeridas, analizando las posibles alternativas que pueden llegar a utilizarse. Por otro lado, dicho diseño debe servir como esquema para la implementación final de la solución. Finalmente, se realizarán diversas pruebas que permitan verificar la viabilidad de la propuesta desarrollada.

4. BENEFICIOS

Como ya se ha indicado en el apartado anterior, el desarrollo de este proyecto busca satisfacer la necesidad actual de integrar la aplicación DynPaC con el escenario SN4I, así como incorporar a dicho escenario dispositivos DPP y herramientas de monitorización basadas en telemetría, manteniendo las funcionalidades y prestaciones de la aplicación DynPaC.

A continuación, se indican, desglosados por su naturaleza, los principales beneficios que derivan del trabajo realizado en este proyecto.

La posterior Descripción de la solución, definirá cómo alcanzar cada uno de los beneficios a continuación presentados.

4.1. Beneficios técnicos

Los principales beneficios técnicos que se derivan del desarrollo de este proyecto son los mencionados a continuación:

- **Innovación técnica:** La incorporación de la aplicación DynPaC al escenario tecnológico de SN4I supone una gran innovación técnica debido a la integración de los servicios que DynPaC permite ofrecer y el escenario puntero en tecnologías de comunicación que constituye SN4I.
- **Incremento del control y de la monitorización del estado de la red:** La solución propuesta en este proyecto proporciona al escenario SN4I el soporte de dispositivos DPP y herramientas de monitorización basadas en telemetría, que le permiten monitorizar la red de forma precisa y en tiempo real. De esta manera, se adquiere un mayor control sobre la red y, con ello, mayor capacidad de gestión. Un operador de red con un conocimiento detallado del estado de la red amplía sus posibilidades técnicas en términos de infraestructura de red y servicios soportados por la misma.

Además, la Programación del Plano de Datos (DPP) es una innovación tecnológica que ha comenzado a despertar interés en tiempos muy recientes y cuya forma de funcionamiento abre nuevas posibilidades en el campo de la operación y gestión de redes muy prometedoras, las cuales despiertan actualmente muchas expectativas.

- **Incremento en la seguridad de la red:** DynPaC es una aplicación de servicios de ancho de banda bajo demanda (BoD, de sus siglas en inglés) que, no sólo permite reservar recursos de la red para un determinado servicio, sino que también se encarga de aislarlo del resto de servicios, para proporcionar seguridad a los mismos.

En consecuencia, la integración de DynPaC con el escenario SN4I supone dotar al tráfico de dicha red de una mayor seguridad tanto frente a situaciones de congestión como frente a posibles ataques y vulnerabilidades.

- **Contribución a la innovación en el campo de las SDN:** Con la solución de actualización propuesta en este proyecto, se incorporan a la red SN4I tecnologías de comunicación novedosas en el campo de las SDN, tales como DPP, herramientas de telemetría... La utilización de estas tecnologías novedosas en este escenario de investigación permite mostrar las ventajas de las mismas, contribuyendo así a su desarrollo y, en consecuencia, su utilización.

4.2. Beneficios económicos

Los principales beneficios económicos que se derivan del desarrollo de este proyecto son los mencionados a continuación:

- **Optimización del funcionamiento de la red:** Las herramientas de monitorización de red basadas en telemetría proporcionan datos del estado de la red en tiempo real. Por lo tanto, al incorporar estas herramientas al escenario SN4I, es posible conocer en tiempo real el estado de la red, lo que permite reaccionar con rapidez y antelación a situaciones de potencial congestión u otro tipo de incidencias que comprometan el buen funcionamiento de la misma.

De esta manera, el operador de red es capaz de optimizar la configuración y el funcionamiento de la red, reduciendo así los costes derivados de incidencias, averías y otras situaciones que comprometan el servicio.

- **Mejora de la utilización de recursos de la red:** En relación con el punto anterior, la propuesta de incorporación de dispositivos DPP y herramientas de monitorización basadas en telemetría propuesta en este proyecto facilita la mejora de la utilización de recursos de la red. Una optimización de la utilización de los recursos de red repercute positivamente de manera significativa en los beneficios económicos del operador de la red.

- **Aumento de prestaciones y QoS:** Al optimizar el funcionamiento de la red, es posible aumentar las prestaciones que se proporcionan al usuario que contrata el servicio de un ISP, así como la calidad de dicho servicio (QoS, de sus siglas en inglés). Una mayor variedad de servicios soportados, unida a la alta calidad de los mismos, permite al operador de la red incrementar sus ingresos y, con ello, sus beneficios económicos.

Asimismo, cabe recordar que la aplicación DynPaC permite reservar recursos en la red para garantizar un determinado ancho de banda a un servicio concreto. En consecuencia, dicho servicio cuenta con los recursos necesarios durante el tiempo requerido y de manera estable, lo que contribuye notablemente al aumento en la QoS proporcionada a sus usuarios.

En un escenario como SN4I, este tipo de servicio resulta prácticamente imprescindible, debido a los nuevos servicios de comunicación que proporciona.

4.3. Beneficios sociales

Los principales beneficios sociales que se derivan del desarrollo de este proyecto son los mencionados a continuación:

- **Incremento de los servicios disponibles y la calidad de los mismos:** El aumento de prestaciones y QoS que soporta la red repercute directamente en la cantidad y la calidad de los servicios de Internet de los que goza el usuario. Teniendo en cuenta la cada vez mayor dependencia de las personas en cualquier ámbito, ya sea profesional o personal, hacia las redes y servicios de Telecomunicación en general, las innovaciones y desarrollo en este campo contribuyen a satisfacer dichas demandas y necesidades.
- **Contribución a la evolución de las Telecomunicaciones:** Tal y como se ha mencionado en el subapartado Beneficios técnicos, este proyecto contribuye a la innovación en el campo de las SDN y, en consecuencia, al incremento del despliegue de este tipo de redes. Una infraestructura de red global definida por *software* permite por completo la implantación de NGN y, con ello, una evolución en las tecnologías de redes de telecomunicación.

5. ANÁLISIS DE ALTERNATIVAS

En este apartado se presentan los diferentes análisis de alternativas que se han llevado a cabo a la hora de tomar ciertas decisiones en cuanto al diseño e implementación desarrollados en este proyecto.

5.1. Estrategia de desarrollo

La primera decisión importante que se debe tomar en este proyecto es la estrategia de desarrollo por la que se va a optar para lograr la integración de DynPaC con el escenario SN4I. A continuación, en este apartado, se analizan las alternativas valoradas para dicho fin, que son las que se consideran más adecuadas y significativas para los objetivos y contexto de este TFM.

5.1.1. Alternativas para la estrategia de desarrollo

5.1.1.1. Solución *from scratch*

La primera alternativa a valorar para integrar DynPaC con el escenario SN4I es desarrollar una nueva aplicación *from scratch*. Es decir, crear un nuevo DynPaC, desarrollado desde cero, sin buscar la reutilización de ningún módulo o componente del DynPaC obsoleto.

Esta solución implica el diseñar una aplicación completamente nueva para DynPaC, de manera que cumpla con todos los requisitos necesarios para ser compatible con SN4I.

Una aplicación DynPaC *from scratch* proporcionaría una precisión muy elevada en cuanto al cumplimiento de los requerimientos del escenario SN4I. Asimismo, al crear una solución a medida para dichos requerimientos, la aplicación tendría un rendimiento mayor dentro de este escenario. Sin embargo, el coste de un desarrollo de este tipo es muy elevado en todos los sentidos, especialmente en tiempo.

5.1.1.2. Solución de migración

Otra alternativa de interés para alcanzar los objetivos que se plantean es partir de la versión de DynPaC obsoleta y migrarla a una versión compatible con versiones recientes de ONOS, de tal modo que se mantenga la aplicación original, pero con las modificaciones pertinentes para actualizarla a las necesidades actuales del escenario SN4I.

La migración de una aplicación de una magnitud como el DynPaC implica una gran complejidad. Además, en el contexto de este proyecto, se le suma la complejidad añadida de que las personas que desarrollaron este *software* no forman parte del grupo I2T en la actualidad, y conocer cada aspecto del mismo es una tarea que implica mucho tiempo y esfuerzo. Sin embargo, el coste de este tipo de solución seguiría siendo menor que el de desarrollar una aplicación de esta magnitud desde cero.

5.1.1.3. Solución híbrida

Otra de las opciones a valorar, dada la arquitectura modular del DynPaC actualmente obsoleto, es analizar el *software* disponible y los requisitos de las nuevas necesidades para mantener aquellos módulos que todavía son compatibles con el escenario SN4I y desarrollar desde cero aquellos que ya no cumplen con los requerimientos actuales de este escenario. De esta manera, se reduciría parte del coste de una solución *from scratch*, sin renunciar del todo al buen rendimiento y precisión que aporta una solución a medida.

Sin embargo, el estudio del código actual para identificar aquellas partes del mismo que pueden ser reaprovechadas supondría un elevado coste para el proyecto.

5.1.2. Selección de la alternativa óptima para la estrategia de desarrollo

Para la evaluación de las alternativas para el tipo de solución a desarrollar en este proyecto, se han establecido unos criterios de selección en base a los objetivos y necesidades de este TFM, con una ponderación asignada en función de la importancia de cada criterio respecto a los demás en esta evaluación.

Cada alternativa se puntúa del 1 al 5 en cada criterio de selección, siendo 1 la puntuación más desfavorable y 5 la más favorable.

Los criterios de selección considerados y sus correspondientes ponderaciones en esta evaluación son los siguientes:

- **Coste (0,45):** Este criterio se refiere al coste total, tanto en tiempo como en dinero, que conlleva llevar a cabo el tipo de solución seleccionado. Al tener un efecto directo sobre el presupuesto del proyecto, este criterio adquiere una ponderación elevada, siendo ésta del 45%.
- **Precisión (0,35):** Este criterio se refiere a la precisión que ofrece el tipo de solución seleccionado frente a los requerimientos establecidos al inicio del proyecto. Este criterio se valora con una ponderación media. Concretamente, del 35%.

- Rendimiento (0,20):** Este criterio representa la medida de eficacia en que la aplicación hace uso de los recursos del sistema en función del tipo de solución seleccionado. Dado que este criterio tiene un efecto indirecto sobre el coste del proyecto, recibe una ponderación del 20%.

A continuación, en la Tabla 5.1, se muestran las puntuaciones asignadas a cada alternativa, desglosadas por criterio y ponderación. Como se puede observar, la alternativa seleccionada para el tipo de solución a desarrollar es la migración. El principal punto determinante en la selección de esta alternativa, dadas las ponderaciones previamente establecidas para cada criterio de selección, ha sido su menor coste en comparación con las otras alternativas. Esto, sumado a su relativa precisión en cuanto al cumplimiento de requerimientos, así como su considerable rendimiento, hacen de la migración la alternativa más adecuada para la solución a desarrollar en este proyecto.

Tabla 5.1: Matriz de evaluación de alternativas para la estrategia de desarrollo.

	Solución <i>from scratch</i>	Solución de migración	Solución híbrida	Ponderación
Coste	1	4	2	0,45
Precisión	5	3	4	0,35
Rendimiento	5	4	4	0,20
TOTAL	3,20	3,65	3,10	-

5.2. Técnica de migración

Dado que la solución seleccionada para desarrollar en este proyecto es migrar la aplicación DynPaC actualmente obsoleta, una de las decisiones fundamentales a tomar es la técnica de migración que se va a seguir. A continuación, en este apartado, se analizan las alternativas valoradas para dicho fin, que son las que se consideran más adecuadas y significativas para los objetivos y contexto de este TFM.

5.2.1. Alternativas para la técnica de migración

5.2.1.1. Técnica Big-Bang

Realizar la migración de una aplicación siguiendo la técnica Big-Bang implica el salto brusco y directo de la versión actual de la mencionada aplicación a la versión deseada. De este modo, la aplicación migrada se integra con el resto del sistema al final de las fases de diseño y desarrollo. Es decir, se integran todos los componentes y, entonces, se prueba el sistema como un todo.

Esta técnica puede ser útil para la detección de errores cuando se encuentren todos los módulos ya en construcción. Además, permite evaluar la interacción entre módulos para agilizar los procesos de integración.

Sin embargo, cuando una aplicación se integra utilizando este enfoque, resulta muy difícil aislar las causas de los errores encontrados en las pruebas. Además, la técnica de migración Big-Bang supone que la integración de los módulos no se pueda comenzar hasta un momento avanzado del proyecto, porque requiere que todos los módulos o sub-módulos de la aplicación hayan sido desarrollados y probados de forma unitaria.

5.2.1.2. Técnica incremental

La técnica de migración incremental consiste, como su propio nombre indica, en la migración del producto siguiendo una estrategia gradual. Es decir, el programa es desarrollado y probado en fases incrementales, atendiendo a la información de soporte publicada para cada una de las actualizaciones de *software* que se han ido publicando.

La complejidad de esta técnica es considerable, debido a la planificación de fases y realización de pruebas intermedias. Además, el enfoque gradual supone extender en el tiempo el proceso de migración como tal.

No obstante, aunque esto pueda incrementar el coste del proyecto, facilita la realización de pruebas unitarias y, en consecuencia, el aislamiento de errores y las pruebas sistemáticas. Asimismo, garantiza el adecuado funcionamiento de cada aspecto modificado en cada actualización, minimizando la probabilidad de error posterior, en la fase de depuración y operación del *software* en el escenario real.

5.2.2. Selección de la alternativa óptima para la técnica de migración

Para la evaluación de las alternativas para la técnica de migración a realizar en este proyecto, se han establecido unos criterios de selección en base a los objetivos y necesidades de este TFM, con una ponderación asignada en función de la importancia de cada criterio sobre los demás en esta evaluación.

Cada alternativa se puntúa del 1 al 5 en cada criterio de selección, siendo 1 la puntuación más desfavorable y 5 la más favorable.

Los criterios de selección considerados y sus correspondientes ponderaciones en esta evaluación son los siguientes:

- **Aislamiento de errores (0,45):** Este criterio se refiere a la capacidad de la técnica utilizada de facilitar la identificación y localización de los errores detectados. Dado que la detección, aislamiento y corrección de errores durante el proceso de integración es fundamental para el éxito del proyecto y para minimizar la probabilidad de error posterior, este criterio recibe una ponderación elevada. Concretamente, de 45%.
- **Coste (0,35):** Este criterio se refiere al coste total, tanto en tiempo como en dinero, que conlleva el proceso de migración según la técnica seleccionada. Al tener un efecto directo sobre el presupuesto del proyecto, este criterio adquiere una ponderación media, siendo ésta de 35%.
- **Precisión (0,20):** Este criterio se refiere a la precisión en la actualización siguiendo la técnica seleccionada. Es decir, la garantía de aprovechamiento que provee en cuanto a cada mejora de cada versión. Este criterio se valora con una ponderación de 20%.

A continuación, en la Tabla 5.2, se muestran las puntuaciones asignadas a cada alternativa, desglosadas por criterio y ponderación. Como se puede observar, la alternativa seleccionada para la técnica de migración a seguir es la técnica incremental. El principal punto determinante en la selección de esta alternativa, dadas las ponderaciones previamente establecidas para cada criterio de selección, ha sido su aislamiento de errores, en comparación con el de la técnica Big-Bang. Esto, sumado a su menor coste en cuanto a desarrollo, hacen de la técnica de migración incremental la alternativa más adecuada para la migración.

Tabla 5.2: Matriz de evaluación de alternativas para la técnica de migración.

	Técnica Big-Bang	Técnica incremental	Ponderación
Aislamiento de errores	2	5	0,45
Coste	5	3	0,35
Precisión	4	3	0,20
TOTAL	3,45	3,90	-

5.3. Tipo de implementación

Otra de las principales problemáticas de ingeniería afrontadas en este TFM es la elección del tipo de implementación que se va a llevar a cabo para probar la viabilidad de la propuesta de diseño. A continuación, en este apartado, se analizan las alternativas consideradas para dicho fin.

5.3.1. Alternativas para el tipo de implementación

5.3.1.1. Implementación sobre equipamiento real

Desplegar el diseño de la solución propuesto en este proyecto enteramente sobre equipamiento real permitiría probar el funcionamiento de dicha solución bajo condiciones equiparables a las de una situación real similar a la situación de SN4I, que es el escenario que se desea desplegar. Por tanto, este tipo de implementación proporcionaría resultados veraces y que pondrían en valor el estudio.

Sin embargo, una implementación sobre equipamiento real tiene un coste elevado a tener en cuenta, tanto en tiempo como en dinero, especialmente si se considera la posibilidad de construir diversos escenarios con casuísticas diferentes para poder evaluar adecuadamente el funcionamiento de cada funcionalidad del diseño propuesto.

5.3.1.2. Implementación sobre equipamiento simulado

Desplegar el diseño de la solución propuesta en este proyecto enteramente sobre equipamiento simulado permitiría, con un coste temporal y económico relativamente reducidos, probar el funcionamiento de dicha solución en diversos escenarios diferentes y bajo casuísticas y condiciones moldeables. Esto proporcionaría resultados bajo un amplio espectro de posibilidades, permitiendo así valorar la adecuación de la solución a un mayor número de situaciones de red posibles y, por tanto, una evaluación más exhaustiva de la misma.

Sin embargo, la fidelidad de dichos resultados, en una implementación emulada mediante *software*, está sujeta a diversos factores que pueden poner en riesgo las evaluaciones del funcionamiento de algún aspecto del desarrollo que se lleve a cabo. De hecho, en despliegues reales, se considera de forma frecuente que este tipo de desarrollos constituyan un paso anterior a las pruebas sobre equipamiento real, permitiendo así ajustar y mejorar diferentes aspectos de la misma antes de su despliegue sobre equipamiento de red real.

5.3.2. Selección de la alternativa óptima para el tipo de implementación

Cada alternativa se puntúa del 1 al 5 en cada criterio de selección, siendo 1 la puntuación más desfavorable y 5 la más favorable.

Los criterios de selección considerados y sus correspondientes ponderaciones en esta evaluación son los siguientes:

- **Precisión de los resultados (0,45):** Este criterio se refiere a la capacidad del sistema de implementación utilizado de proporcionar resultados fidedignos. Dado que la veracidad y máxima similitud con la realidad de los resultados obtenidos en las pruebas a realizar es de vital importancia para la validez de este proyecto, este criterio recibe una ponderación elevada. Concretamente, de 45%.
- **Coste (0,35):** Este criterio se refiere al coste total, tanto en tiempo como en dinero, que conlleva el sistema de implementación utilizado. Al tener un efecto directo sobre el presupuesto del proyecto, este criterio adquiere una ponderación media, siendo ésta de 35%.
- **Versatilidad (0,20):** Este criterio se refiere a la flexibilidad que ofrece el sistema de implementación utilizado para configurar y probar diferentes escenarios, con el fin de dotar a la propuesta del proyecto de mayor ámbito de evaluación y, con ello, valor. Por esto, este criterio se valora con una ponderación de 20%.

A continuación, en la Tabla 5.3, se muestran las puntuaciones asignadas a cada alternativa, desglosadas por criterio y ponderación. La alternativa seleccionada para el tipo de implementación a realizar es la implementación sobre equipamiento real. El principal punto determinante en la selección de esta alternativa, dadas las ponderaciones previamente establecidas para cada criterio de selección, ha sido la precisión de los resultados que es capaz de proporcionar, en comparación con la de una implementación emulada mediante *software*. Esto, sumado a su relativa versatilidad para configurar diferentes escenarios para probar la viabilidad de la propuesta de diseño, hacen de la implementación sobre equipamiento real la alternativa óptima para el tipo de implementación.

Tabla 5.3: Matriz de evaluación de alternativas para el tipo de implementación.

	Implementación real	Implementación emulada	Ponderación
Precisión	5	2	0,45
Coste	2	5	0,35
Versatilidad	3	4	0,20
TOTAL	3,55	3,45	-

6. ANÁLISIS DE RIESGOS

El desarrollo de un proyecto de ingeniería siempre conlleva asumir una serie de riesgos. Por lo tanto, antes de definir una solución concreta en base a las alternativas presentadas con anterioridad, resulta necesario estudiar los riesgos que pueden derivarse del desarrollo de este proyecto. A continuación, se resumen los diferentes puntos a tener en cuenta:

6.1. Riesgos externos

Los riesgos externos tienen su origen fuera el equipo del proyecto. Esto significa que hasta cierto punto no pueden ser controlados por la propia Empresa o, en este caso, Grupo de Investigación. Entre este tipo de riesgos destaca:

- **(1) Incompatibilidad con nuevas versiones de ONOS:** Tal y como se ha mencionado anteriormente en este documento, ONOS proporciona actualizaciones de su *software* cada 4-6 meses para incorporar nuevas mejoras y funcionalidades que lo permitan adaptarse a la evolución constante de las tecnologías de red y comunicación.

Durante el desarrollo de este proyecto, cabe la posibilidad de que ONOS publique una nueva versión no completamente compatible con el trabajo realizado en este TFM.

6.2. Riesgos internos

Los riesgos internos tienen su origen, en este caso, dentro del propio equipamiento sobre el que se ha trabaja en este proyecto. Esto significa que tanto la prevención de los mismos como la solución a sus posibles efectos queda bajo la responsabilidad del personal miembro del equipo de proyecto. Entre este tipo de riesgos destaca:

- **(2) Problemas de *hardware*:** Los fallos en el equipamiento suelen ser uno de los problemas habituales en todo proyecto de ingeniería de redes. Tal y como se ha visto en el análisis de alternativas, la implementación de la solución objeto de este proyecto se va a realizar sobre equipamiento de real. Cabe la posibilidad de que dicho equipamiento de red sufra problemas técnicos que afecten al proyecto en cuanto a plazos y al coste asociado a dichos plazos.

- **(3) Pérdida de información:** De la misma manera que el equipamiento de red, el equipo (ordenador personal) utilizado para documentar todo el proyecto podría sufrir problemas técnicos, provocando una pérdida parcial del proyecto. Esto podría afectar al proyecto en cuanto a plazos y al coste asociado a dichos plazos.

6.3. Análisis de riesgos

Una vez determinados los principales riesgos del proyecto, se va a hacer uso de una herramienta de análisis de riesgos denominada Matriz Probabilidad-Impacto (Figura 6.1), para establecer prioridades en cuanto a los diferentes riesgos que pueden afectar al proyecto, considerando tanto la probabilidad de que ocurran, como el impacto que tiene su ocurrencia sobre el proyecto.

A continuación, en la Figura 6.1, puede observarse cómo el riesgo (1) tiene una probabilidad muy baja dentro del contexto de este proyecto, especialmente teniendo en cuenta el plazo de tiempo establecido para este TFM.

Sin embargo, el riesgo (2) tiene una probabilidad mayor que el riesgo (1). Esto, sumado a un impacto considerablemente elevado en el proyecto, provoca la necesidad de tomar medidas de prevención consistentes, con el fin de evitar a toda costa verse en la situación de afrontar dicho impacto.

Por otro lado, el riesgo (3) tiene la mayor probabilidad de ocurrir y un impacto notable en el proyecto. Por esto, será el riesgo -(3)- al que más atención será necesario prestar, para no comprometer el éxito del proyecto.

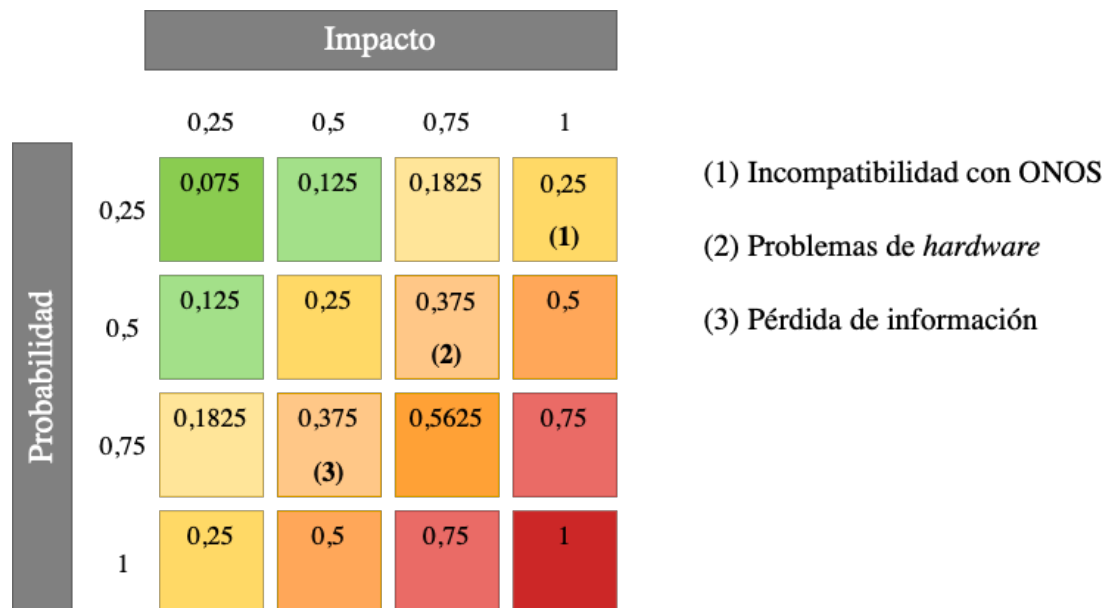


Figura 6.1: Matriz Probabilidad-Impacto de los riesgos del proyecto.

6.4. Plan de prevención

El último paso a realizar en el proceso de análisis de riesgos consiste en definir un plan de prevención que permita minimizar la probabilidad de que los riesgos ocurran, así como establecer un protocolo de actuación en caso de que uno de los riesgos mencionados se materialice. Para cada uno de los riesgos anteriormente citados se establecen las siguientes acciones:

- **(1) Diseño eficiente y migrable a otro tipo de plano de control:** La versión original de DynPaC constituye un *framework* genérico implementable sobre infraestructuras OpenFlow SDN. Es decir, la aplicación DynPaC no depende estrictamente de un plano de control ONOS para su funcionamiento.

Mantener este mismo enfoque para el diseño de la solución objeto de este TFM garantiza que, aunque ONOS publique una nueva versión incompatible con DynPaC, dicha solución pueda seguir siendo válida siempre y cuando se cumpla con los requisitos establecidos de monitorización basada en telemetría y DPP.

Además, proporcionar una solución con enfoque modular, de acuerdo con los objetivos adicionales de este TFM, incrementa las posibilidades de que dicho diseño sea más fácilmente implementable en nuevos planos de control.

- **(2) Hardware de respaldo:** En caso de que alguno de los equipos utilizados sufra problemas técnicos que puedan inutilizarlo, conviene contar con equipamiento de *back up* como reserva.
- **(3) Copias de seguridad y uso de repositorios en la nube:** Llevar un adecuado control de versiones en un repositorio (en la nube, por ejemplo) convierte en prácticamente nulo el impacto derivado de la posibilidad de perder el trabajo realizado a lo largo del proyecto por un problema técnico en el equipo utilizado.

7. DESCRIPCIÓN DE LA SOLUCIÓN

7.1. Introducción

Este proyecto propone una solución para la integración de DynPaC con SN4I y la incorporación de dispositivos DPP y herramientas de monitorización basadas en telemetría.

Cabe recordar que, tal y como se ha descrito previamente en este documento, el objetivo principal de este proyecto consiste en:

- **Compatibilidad de DynPaC con SN4I.**

Asimismo, como también se ha señalado en apartados anteriores, las versiones de ONOS posteriores a la aplicación original DynPaC (2017) incorporan mejoras y tecnologías que resultan de gran interés para este proyecto, ya que se alinean significativamente con los objetivos anteriormente destacados. Dentro de estas actualizaciones de ONOS, las incorporaciones de mayor relevancia para este proyecto son:

- **Mejoras en la monitorización de la red:** Incorporando una tecnología de monitorización en banda basada en telemetría, denominada INT, que permite la monitorización de aspectos avanzados de la operación y estado de la red desde el propio plano de datos.
- **Programabilidad del Plano de Datos (DPP, de sus siglas en inglés):** Incluyendo la compatibilidad con el marco P4Runtime, que permite especificar cómo los dispositivos del plano de datos (*switches, routers...*) deben procesar los paquetes.

De forma adicional, otra de las mejoras que incluye ONOS y que, aunque de forma más indirecta, también contribuye a la mejora del escenario de este TFM y, por tanto, conviene tener en cuenta, es el señalado a continuación:

- **Mejoras en la interfaz gráfica de usuario:** Se han desarrollado algunas mejoras significativas en la herramienta en cuanto a aspectos de visualización e interacción con el usuario, incluyendo la mejora de la vista topológica de la red, posibilidad de selección múltiple de nodos, incorporación de actualizaciones dinámicas de la topología, etc.

Teniendo en cuenta esta similitud entre los objetivos marcados para este TFM y las mejoras incorporadas por ONOS en sus últimas versiones, este proyecto propone una solución basada en la actualización de DynPaC para lograr su integración con SN4I, así como la incorporación de dispositivos DPP y herramientas de monitorización basadas en telemetría.

A continuación, en este apartado, se describe en detalle la solución diseñada e implementada en este TFM.

En primer lugar, se explica la arquitectura modular de la aplicación DynPaC, describiendo cada uno de los módulos que la componen y su funcionamiento.

En segundo lugar, se describe el funcionamiento de DynPaC dentro de la arquitectura SN4I, escenario de este TFM.

En tercer lugar, se expone el diseño de la solución objeto de este proyecto, estructurado en diferentes fases incrementales, tal y como se ha decidido en el Análisis de alternativas.

Finalmente, se detalla el desarrollo llevado a cabo para implementar el diseño de la solución realizado en este proyecto, así como las pruebas realizadas para validar dicha solución.

7.2. Arquitectura DynPaC

Como ya se ha mencionado previamente, DynPaC constituye un *framework* genérico que facilita la adopción de estrategias de TE en SDN. Su arquitectura consiste en una serie de módulos bien definidos y una API REST que facilita su integración con elementos externos. En este apartado se expone en detalle dicha arquitectura.

7.2.1. Módulos

Como ya se ha adelantado, DynPaC tiene una arquitectura modular. Concretamente, consta de seis módulos, los cuales se describen en este subapartado. Dichos módulos son, tal y como se puede observar en la Figura 7.1: *Path Computation Element (PCE)*, *Topology Abstractor*, *Resilience Module*, *Monitoring Module*, *Network Programmer* y *Service Manager*.

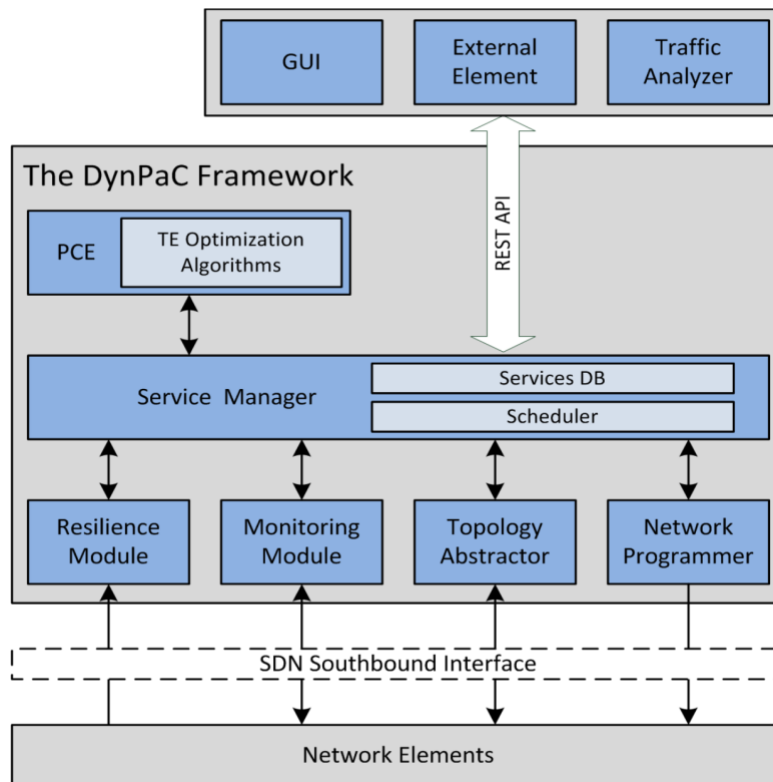


Figura 7.1: Arquitectura modular de DynPaC.

7.2.1.1. Path Computation Element (PCE)

El objetivo de la Ingeniería de Tráfico (TE) es optimizar el rendimiento de las comunicaciones en las redes en las que opere. Dado que el cálculo de rutas tiene un impacto directo sobre la utilización de los recursos de la red, se convierte en un punto clave de la TE.

Cabe recordar que la finalidad de DynPaC es facilitar la aplicación de estrategias de TE a las SDN. En consecuencia, DynPaC incorpora un módulo completo cuyo único objetivo es el cálculo de rutas para implementar esas TE: *Path Computation Element* (PCE).

Disponer de un módulo independiente especializado en el cálculo de rutas aporta múltiples beneficios. Entre ellos, destaca la posibilidad que brinda para modificar o actualizar el algoritmo en cuestión, sin tener que modificar otras partes de DynPaC. Dependiendo el caso de uso, puede convenir el uso de un algoritmo de encaminamiento u otro, según se priorice un menor consumo de potencia, la maximización de la Calidad de Experiencia (QoE), etc.

7.2.1.2. *Topology Abstractor*

Para calcular las diferentes rutas, el PCE necesita conocer en detalle la topología de la red. Para ello, se utiliza el módulo *Topology Abstractor*, cuya función principal es construir un modelo preciso de la red que facilite el cálculo de rutas.

Concretamente, en el *framework* de DynPaC, el *Topology Abstractor* dibuja el grafo de la red y lo abstrae como una matriz unidimensional, donde cada entrada representa el ancho de banda disponible en cada enlace bidireccional que interconecta dos nodos determinados, aprovechando el hecho de que los servicios se consideran, en este caso, simétricos y bidireccionales.

7.2.1.3. *Resilience Module*

Uno de los objetivos más habituales en la TE es la minimización de la pérdida de paquetes. Por ello, DynPaC incluye un módulo de resiliencia (*Resilience Module*) cuya función es minimizar el impacto derivado de la interrupción del servicio y las consecuentes pérdidas de paquetes en caso de un fallo en la red.

El *Resilience Module* monitoriza el estado de la infraestructura de la red e informa al gestor del servicio (*Service Manager*) cuando detecta algún fallo en un nodo o enlace, para que éste (*Service Manager*) pueda ejecutar sus mecanismos de recuperación. Este procedimiento se describe con mayor detalle en el subapartado referente al módulo *Service Manager*.

Dependiendo del caso de uso, se puede requerir una función de protección de la ruta o bien una función de restablecimiento de la ruta.

- **Protección de la ruta:** Mecanismo de recuperación en el que los recursos de reserva están precalculados y reservados. En el momento en el que se detecta un fallo en la red, los recursos reservados pasan a ser utilizados para proporcionar el servicio.
- **Restablecimiento de la ruta:** Mecanismo de recuperación en el que los recursos alternativos no están previamente reservados. En el momento en el que se detecta un fallo en la red, se calcula una nueva ruta teniendo en cuenta los recursos disponibles en dicho momento.

En ambos mecanismos, es necesario comprobar si existen suficientes recursos de red antes de proporcionar la ruta. Sólo en el caso de que los haya, la ruta de reserva será instalada en la red.

En cualquier caso, cuando el *Resilience Module* detecte el fallo, el *framework* de DynPaC procederá a reprogramar los elementos de red afectados para instalar rutas alternativas y así mantener la conectividad de los servicios afectados.

La funcionalidad implementada por el módulo de resiliencia es especialmente valiosa para aquellos casos de uso en los que se requiere una elevada disponibilidad.

7.2.1.4. *Monitoring Module*

DynPaC dispone de un módulo capaz de realizar un seguimiento de los recursos utilizados por cada servicio de conectividad proporcionado en la red. La monitorización de la red es crucial tanto para los usuarios finales como para los operadores de la red, ya que permite verificar que los Acuerdos a Nivel de Servicio (SLA, de sus siglas en inglés) entre ambos se cumplen.

El módulo de monitorización (*Monitoring Module*) de DynPaC está diseñado para realizar un seguimiento por flujo y por servicio de los recursos de red utilizados. Gracias a ello, dicho módulo es capaz de gestionar la disgregación de un servicio en varios subservicios.

7.2.1.5. *Network Programmer*

El módulo de programación de la red (*Network Programmer*) es el responsable de programar los elementos de red implicados en la provisión de un determinado servicio.

Como se ha mencionado previamente, en los sistemas de reserva anticipada de recursos, dichos recursos se reservan para el futuro y durante un período de tiempo determinado. Esto tiene las implicaciones mencionadas a continuación.

- Los recursos reservados para un servicio concreto pueden utilizarse para otras reservas hasta que llegue el momento de comienzo de dicho servicio.
- Una vez dé comienzo el servicio, la ruta óptima calculada por el PCE debe ser instalada en la red.
- Asimismo, una vez liberado el servicio, los recursos implicados pueden volver a ser objeto de otras reservas.

En definitiva, el *Network Programmer* debe extraer la información contenida en una ruta especificada (secuencia de elementos de red) y traducirla al formato adecuado necesario para programar dichos elementos de red. En el caso concreto de OpenFlow, deberá generar mensajes de tipo *flow modification* para instalar o eliminar las entradas en la tabla de flujos (*flow table*) de los *switches*.

7.2.1.6. *Service Manager*

El gestor de servicios *Service Manager* es el núcleo del *framework* de DynPaC. Las funciones de este módulo van desde la gestión de los flujos de trabajo a la gestión de los recursos de red, pasando por la gestión del ciclo de vida de los servicios.

Gestión de flujos de trabajo

Todos los módulos de DynPaC, así como la API REST, se comunican con el módulo de gestión de servicios, quien interacciona con los diferentes módulos para proveer el servicio de conectividad solicitado.

En primer lugar, el *Service Manager* activa el mecanismo en el *Topology Abstractor* para obtener una visión de la topología de la red que facilita el posterior cálculo de rutas que realizará el PCE. Una vez el PCE dispone de la información necesaria relativa a la topología para calcular las rutas, el *Service Manager* permite atender solicitudes de servicio.

Cada vez que se realiza una nueva solicitud de servicio, el *Service Manager* solicita una ruta o un conjunto de ellas al PCE.

Seguidamente, el *Service Manager* comprueba si dicha solicitud de servicio puede ser aceptada o no, teniendo en cuenta los recursos de red disponibles y las rutas posibles proporcionadas por el PCE.

Por último, el *Service Manager* se comunica con el *Network Programmer* tanto para instalar como para desinstalar el servicio.

Asimismo, cabe mencionar que, durante el tiempo que un servicio está aceptado, el *Service Manager* se mantiene a la escucha en todo momento de los diferentes eventos que puedan provenir de los módulos de resiliencia (*Resilience Module*) y monitorización (*Monitoring Module*) y que supondrían un riesgo o un cambio en el estado del servicio.

Gestión del ciclo de vida de los servicios

DynPaC soporta la reserva anticipada de servicios. Es decir, permite la reserva de servicios que van a ser automáticamente instalados o eliminados en la red en el futuro.

En consecuencia, se debe monitorizar el estado de cada servicio instalado en la red, ya que es necesario conocer si dicho servicio se encuentra reservado para el futuro, instalado en la red o ya ha expirado. Para ello, el *Service Manager* utiliza la Base de Datos *Services DB*.

Gestión de los recursos de red

Tal y como ya se ha mencionado anteriormente, una de las principales funcionalidades de DynPaC es permitir la reserva anticipada de recursos.

Para ello, es necesario realizar un seguimiento del consumo de recursos. Estos datos se almacenan en la base de datos denominada *Scheduler*, dentro del módulo *Service Manager*.

7.2.2. Interfaz con elementos externos

Una de las funcionalidades más destacables del *framework* de DynPaC es que proporciona una API REST abierta para su integración con diferentes elementos externos a través del módulo *Service Manager*.

Por ejemplo, un analizador de tráfico externo podría utilizar esta interfaz para enviar y/u obtener información sobre el tráfico de la red, que podría servir para optimizar el consumo de los recursos de la red.

Por otro lado, otro tipo de elementos externos de gestión, como un orquestador, podrían utilizar esta interfaz para la gestión remota del ciclo de vida de los distintos servicios.

En el escenario de este TFM, esta interfaz se podría utilizar para la comunicación de DynPaC con el controlador SDN ONOS. Sin embargo, tal y como se ha indicado en anteriores apartados, DynPaC también ofrece la posibilidad de ser instalado como una aplicación de ONOS, a través de la propia API del controlador. Ésta es la forma de interconexión utilizada en el escenario de este TFM, debido a las ventajas que ofrece en cuanto a GUI y usabilidad. Para mayor detalle, esta forma de interconexión se describe más adelante en el siguiente subapartado.

7.3. Funcionamiento de DynPaC en SN4I

Como ya se ha indicado anteriormente en este documento, DynPaC es un *framework* que permite la provisión de servicios de capa 2 sobre una infraestructura SDN basada en OpenFlow. DynPaC es capaz de programar la reserva de servicios con restricciones de ancho de banda dentro de un intervalo de tiempo deseado, ya sea presente o futuro.

Cabe recordar también que DynPaC utiliza el cálculo de ruta de extremo a extremo y tiene en cuenta el ancho de banda disponible en la red, el número de saltos entre los puntos finales y la estabilidad de la ruta, para asignar la mejor ruta posible a cada servicio según sus requisitos de ancho de banda. Asimismo, los servicios de DynPaC están aislados entre sí, al encapsular su tráfico en diferentes VLAN, para garantizar la seguridad mediante la diferenciación de los servicios.

En el escenario de este TFM, DynPaC se instala como una aplicación de ONOS, de manera que puede trabajar con el controlador SDN mientras reside en el plano de aplicaciones de la arquitectura SDN.

De esta manera, DynPaC es capaz de crear servicios sobre la infraestructura gestionada por el controlador ONOS y se comunica con él para instalar las reglas de flujo correspondientes en los *switches* OpenFlow. Estas reglas de flujo permiten ofrecer cada servicio creado.

A continuación, en la Figura 7.2, se ilustra la arquitectura de DynPaC (azul) sobre ONOS (rojo). Atendiendo a dicha figura, se pueden observar los módulos de DynPaC que interactúan con ONOS:

- **Topology Abstractor:** Este módulo recopila la información proporcionada por los módulos de topología de ONOS (*Topology Service*, *Edge Port Service*, etc) y los utiliza para abstraer la topología de la red para facilitar el cálculo de las rutas.
- **Resiliency Module:** Este módulo utiliza el módulo de monitorización de los enlaces de ONOS (*Link Listener*) para poder proceder a la instalación de rutas de respaldo en caso de que se detecte un fallo en la ruta principal.
- **Network Programmer:** Este módulo interactúa con los módulos de aplicación y de flujo de ONOS (*Application Service*, *Flow Objective Service*) para instalar en los *switches* OpenFlow las reglas de flujo que conforman un servicio durante la duración de dicho servicio.

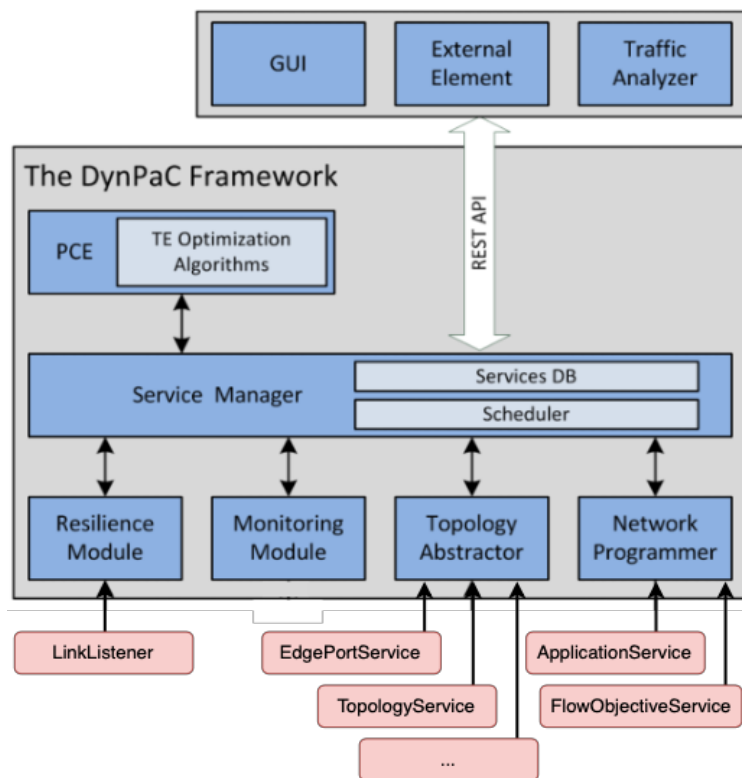


Figura 7.2: Arquitectura DynPaC como aplicación de ONOS.

Cabe destacar también que, cuando DynPaC se instala como una aplicación de ONOS, la Interfaz Gráfica de Usuario (GUI, de sus siglas en inglés) de ONOS muestra una nueva pantalla donde se muestra toda la información referente a los servicios de DynPaC: su estado, duración y ancho de banda reservado, así como los *switches*, puertos y etiquetas VLAN involucrados. De esta manera, esta forma de interconexión entre DynPaC y ONOS, facilita la gestión de los servicios de DynPaC, ya que el administrador de dichos servicios puede acceder a la información y gestión de los mismos, a través de la propia GUI de ONOS.

7.4. Diseño de la solución

A continuación, en este apartado, se expone el diseño de la solución desarrollado en este proyecto. Primero, se presenta una visión general del enfoque adoptado para el desarrollo de dicho diseño. Seguidamente, se describe detalladamente el diseño realizado, estructurado en diferentes incrementos.

7.4.1. Visión general

En las versiones de ONOS posteriores (v1.11.0 en adelante) a la versión originalmente compatible (v.1.10.0) con DynPaC, se incorporan múltiples mejoras referentes a diversos aspectos. En concreto, las más relevantes en el escenario de este TFM, de acuerdo con los objetivos marcados en este proyecto (véase el apartado Objetivos y alcance), debido los avances tecnológicos que suponen para las actividades de investigación llevadas a cabo en la red SN4I, son las expuestas a continuación. [4]

- **Servicio de ONOS de Telemetría de Red en Banda (INT, de sus siglas en inglés) e implementación del int.p4 de referencia [5]:** In-band Network Telemetry (INT) es una herramienta de telemetría para la monitorización de redes directamente a través del plano de datos. El funcionamiento de INT se basa en la inyección de cabeceras que aportan información sobre el estado de la red en los paquetes que se transmiten a través del plano de datos.

La inyección directa de información de telemetría en paquetes que se transmiten por la red supone una fiabilidad y precisión valiosas para la monitorización de la red, puesto que los datos obtenidos y recolectados no son estadísticas ni generalizaciones, sino que se puede observar en ellos las peculiaridades de cada ruta seguida y cada instante en el que un paquete ha sido transmitido.

- **Mejoras relacionadas con P4Runtime:** El acrónimo P4 viene del nombre completo Programming Protocol-independent Packet Processors, que significa “programación de procesadores de paquetes independientes del protocolo”. P4 es el lenguaje de programación de código abierto y de dominio específico líder para dispositivos de red [6]. P4 permite especificar cómo los dispositivos del plano de datos (*switches, routers...*) deben procesar los paquetes. Asimismo, P4 es el lenguaje de programación utilizado para la implementación de la herramienta de telemetría INT, según el estándar de IETF. [5]

- **Escalabilidad:** Se soportan hasta 200 instancias de BMv2 en un clúster de controladores ONOS de 3 nodos.
- **Soporte de reconexión a dispositivos perdidos:** Se permite la reconexión a dispositivos con los que se había perdido la conectividad.
- **Compatibilidad con varias API de *multicast*:** Gracias al nuevo sistema de replicación de paquetes.
- **Mejoras relacionadas con Stratum:** Stratum es actualmente el sistema operativo (OS, de sus siglas en inglés) de *software* libre líder para dispositivos del plano de datos (*switches*) de SDN [7]. Los dispositivos de red que funcionan con Stratum permiten la implementación de las tecnologías P4 (DPP) e INT (telemetría) anteriormente comentadas.
 - **Soporte para gNMI y el modelo de Interfaces OpenConfig:** Descubrimiento de puertos, sondeo de estadísticas, suscripción a eventos *up/down* de puertos.
 - **Driver de Stratum:** Extiende los *drivers* de P4Runtime y gNMI.
 - **Drivers de stratum-bmv2 y stratum-tofino:** Los switches BMv2 y Tofino son los más utilizados para arquitecturas SDN con planos de datos programables P4.
 - **Soporte para canales seguros gRPC via SSL/TLS.**
- **Mejoras de plataforma:**
 - **Actualización de Apache Karaf:** versión 4.2.3.
 - **Actualización de Bazel:** versión 0.25.
- **GUI:** Se incorporan diversas mejoras en la interfaz gráfica de usuario tales como mejora de la vista topológica de la red, posibilidad de selección múltiple de nodos, incorporación de actualizaciones dinámicas de la topología, etc.

Con el objetivo de la incorporación de todas estas mejoras incluidas en las últimas versiones de ONOS, se ha diseñado la migración de DynPaC a una versión más actualizada y compatible con estas versiones de ONOS. Concretamente, se parte de una versión de DynPaC compatible con ONOS v1.10.0 y la actualización objeto de este proyecto proporciona una versión de DynPaC compatible con ONOS v2.1.0. A continuación, en el siguiente subapartado, se describe en detalle el diseño de la solución de actualización propuesta en este TFM.

7.4.2. Diseño de las fases de migración

En el contexto de este TFM, tal y como se ha mostrado en el Análisis de alternativas, el enfoque óptimo para migrar la aplicación DynPaC es seguir un proceso de actualización incremental. Es decir, el diseño de la solución de actualización objeto de este proyecto se divide en varias fases incrementales, incorporando gradualmente las diferentes funcionalidades requeridas hasta llegar a una versión final que satisfaga todos los requerimientos marcados al inicio del proyecto (véase el apartado Objetivos y alcance).

Concretamente, el proceso de migración de DynPaC se ha dividido en tres fases de migración, de manera que cada una de ellas se centra en la incorporación de una parte de las funcionalidades requeridas para la versión final. A continuación, se describen brevemente las aportaciones de cada incremento según el diseño realizado.

- **Fase I:** En esta fase se parte de la versión original de DynPaC, compatible con ONOS v1.10.0, y se actualiza hasta alcanzar la compatibilidad con ONOS v.1.15.0. Esta fase se centra en los objetivos de incorporar programabilidad en el plano de datos y soportar herramientas de telemetría para mejorar la monitorización de la topología y estado de la red SN4I.
- **Fase II:** En esta fase se actualiza DynPaC a una versión compatible con ONOS v2.0.0. Esta fase se centra fundamentalmente en el objetivo de mejorar la programabilidad en el plano de datos. Asimismo, este incremento también proporciona mejoras en la interfaz gráfica de usuario a través de la cual se gestionará la aplicación DynPaC.
- **Fase III:** En esta fase se actualiza DynPaC a una versión compatible con ONOS v2.1.0. Esta fase se centra en el objetivo de integración de DynPaC con SN4I. También contribuye al objetivo de mejorar la programabilidad en el plano de datos y proporciona mejoras en la interfaz gráfica de usuario a través de la cual se gestionará la aplicación DynPaC.

A continuación, en los siguientes subapartados, se describe en detalle el diseño incremental llevado a cabo en este proyecto para la integración de DynPaC con SN4I y la incorporación de dispositivos DPP y herramientas de monitorización basadas en telemetría.

7.4.2.1. Fase I

Este proyecto de migración parte de una versión de DynPaC compatible con ONOS v1.10.0. El objetivo de la primera fase de migración diseñada en este TFM es actualizar DynPaC a una versión compatible con ONOS v1.15.0.

En lo que respecta a este proyecto, la versión de partida carece de las siguientes características requeridas actualmente en el escenario SN4I:

- **Soporte de dispositivos DPP:** ONOS v1.10.0 y sus versiones anteriores no soportan tecnologías relacionadas con DPP.
 - No existe compatibilidad con ninguna tecnología que permita programar los dispositivos del plano de datos.
 - No soportan el uso de dispositivos que implementen DPP.
- **Soporte de herramientas de monitorización basadas en telemetría:** La versión de ONOS 1.10.0 y sus anteriores no son compatibles con herramientas que permitan monitorizar el estado de la red mediante telemetría.
- **Soporte de herramientas de virtualización:** Aunque esta versión soporta las herramientas de virtualización utilizadas en SN4I (Apache Karaf, Kubernetes, OpenStack...), lo hace en versiones considerablemente antiguas. En los campos de la informática y las comunicaciones, siempre es aconsejable trabajar con las versiones más recientes de los *softwares* utilizados. Por ello, resulta de gran interés para el escenario SN4I soportar las últimas versiones de estas herramientas.

En la versión objetivo de esta primera fase de migración, se dota al escenario SN4I de las características mencionadas a continuación:

- **Soporte de P4Runtime:** Mediante el soporte de P4Runtime se logra dar soporte al lenguaje de programación P4, que permite especificar cómo los dispositivos del plano de datos (*switches, routers...*) deben procesar los paquetes. De esta manera, se logra dotar de programabilidad al plano de datos (DPP) de la red SN4I.
- **Soporte de INT:** La compatibilidad con INT permite la mejora de la monitorización del estado de la red SN4I, mediante la incorporación de esta herramienta de telemetría.

Para lograr estas características, la actualización de la aplicación de DynPaC requiere las modificaciones indicadas a continuación:

- **Modificación del archivo de configuración de la aplicación para indicar la nueva versión de ONOS a utilizar.**
- **Eliminación de paquetes que quedan obsoletos y la adecuada importación de los nuevos paquetes que incorporan esas funcionalidades. En concreto, los módulos afectados son:**
 - *Topology Abstractor*
 - PCE
- **Gestión y procesado de nuevas dependencias.**

No obstante, esta versión de DynPaC diseñada en la Fase I todavía cuenta con algunas carencias con respecto al soporte de los dispositivos DPP y las herramientas de virtualización utilizados en el escenario SN4I, tal y como se detalla, a continuación, en el siguiente subapartado.

7.4.2.2. Fase II

En la fase II de migración se parte de la versión de DynPaC que ha resultado de la primera fase, anteriormente descrita; es decir, se parte de la versión compatible con ONOS v1.15.0. El objetivo de esta fase es actualizar DynPaC a una versión compatible con ONOS v2.0.0.

En lo que respecta a este proyecto, la versión de partida de esta fase carece de las siguientes características requeridas actualmente en el escenario SN4I:

- **Soporte de Stratum:** Esta versión no soporta el sistema operativo Stratum. Cabe recordar que es el sistema operativo de *software* libre líder actualmente para dispositivos DPP. Por lo tanto, dado el objetivo de este proyecto de incorporar DPP al escenario SN4I, es conveniente el soporte de este SO para poder trabajar con dispositivos que lo utilicen, tales como BMv2 o Tofino.
- **Soporte de herramientas de virtualización:** Esta versión no soporta las últimas versiones de las herramientas de virtualización utilizadas en SN4I (Apache Karaf, Kubernetes, OpenStack...).

La versión objetivo de esta segunda fase de migración incorpora al escenario SN4I las características mencionadas a continuación:

- **Soporte de Stratum:** Esta versión soporta el sistema operativo Stratum, de manera que se dota al escenario SN4I de compatibilidad con todos los dispositivos de red del plano de datos (*switches*) que utilicen este sistema operativo para implementar DPP.

- **Soporte de P4 e INT sobre Stratum:** Esta versión también incluye las funcionalidades y los *drivers* necesarios para poder implementar P4 e INT sobre dispositivos que funcionan con Stratum OS.
- **Mejora de la GUI:** De forma adicional, al actualizar DynPaC a ONOS v2.0.0, esta fase también proporciona mejoras en la interfaz gráfica de usuario de ONOS, a través de la cual se gestiona la aplicación DynPaC. En consecuencia, se aportan mejoras en la GUI del plano de control del escenario de red SN4I.

Para lograr estas características, la actualización de la aplicación de DynPaC requiere las modificaciones indicadas a continuación:

- **Modificación del archivo de configuración de la aplicación para indicar la nueva versión de ONOS a utilizar.**
- **Gestión y procesado de nuevas dependencias.**
- **Actualización de clases y métodos de determinados paquetes. En concreto, los módulos afectados son:**
 - *Service Manager*
 - *Network Programmer*
 - *Topology Abstractor*

Sin embargo, cabe mencionar que esta versión de DynPaC diseñada en la Fase II todavía cuenta con algunas carencias con respecto al soporte de las herramientas de virtualización utilizadas en el escenario SN4I, tal y como ocurría en versiones anteriores.

7.4.2.3. Fase III

En la fase III de migración se parte de la versión de DynPaC que ha resultado de la fase anterior (Fase II); es decir, se parte de la versión compatible con ONOS v2.0.0. El objetivo de esta fase es actualizar DynPaC a una versión compatible con ONOS v2.1.0.

Esta fase de migración se centra en resolver la carencia en la compatibilidad con las últimas versiones de las herramientas de virtualización utilizadas en el escenario SN4I, tales como Apache Karaf, Kubernetes u OpenStack.

Para ello, la actualización de la aplicación de DynPaC requiere las modificaciones indicadas a continuación:

- **Modificación del archivo de configuración de la aplicación para indicar la nueva versión de ONOS a utilizar.**
- **Gestión y procesado de nuevas dependencias.**

De esta manera, la versión objetivo de esta fase de migración, la cual constituye la versión final del proceso de actualización presentado en este proyecto, incorpora todos los requerimientos del escenario SN4I expuestos al inicio de este documento:

- **Programabilidad en el Plano de Datos:** Mediante la migración diseñada en este proyecto, se ha dotado al escenario SN4I de la posibilidad de incorporación de dispositivos DPP, gracias a la incorporación de las siguientes tecnologías:
 - **P4Runtime:** La compatibilidad con P4Runtime permite utilizar el lenguaje P4 para programar los dispositivos del plano de datos.
 - **Stratum:** La compatibilidad con Stratum OS permite utilizar los dispositivos DPP que hacen uso de este sistema operativo para implementar DPP.

De forma adicional en esta última actualización se incluyen nuevas mejoras para Stratum OS relacionadas con el soporte de P4Runtime y *switches* programables como Stratum-BMv2 y Stratum-Tofino, los cuales se utilizan en la red SN4I.

- **Monitorización de red basada en telemetría:** La migración diseñada en este proyecto ha logrado mejorar la monitorización del estado de la red SN4I mediante la posibilidad de utilizar la herramienta de telemetría INT.
- **Integración de DynPaC en el escenario SN4I:** A lo largo de toda la migración objeto de este TFM se ha ido logrando la integración de la aplicación DynPaC con el escenario SN4I, que ahora soporta dispositivos DPP y herramientas de monitorización basadas en telemetría.
- **Mejoras en la GUI:** De forma adicional, las dos últimas fases de esta migración proporcionan mejoras en la interfaz gráfica de usuario de ONOS, a través de la cual se gestiona la aplicación DynPaC. Así, una vez más, se aportan mejoras en la GUI del plano de control del escenario de red SN4I.

7.5. Implementación de la solución

Una vez finalizado el diseño de la solución, se puede proceder al desarrollo de la propuesta. A continuación, en este subapartado, se describe la implementación llevada a cabo en este proyecto.

7.5.1. Plataforma de desarrollo

Para llevar a cabo el desarrollo de este proyecto, se han utilizado una serie de herramientas *hardware* y *software* que constituyen la plataforma de desarrollo. A continuación, en el presente apartado, se exponen los componentes de dicha plataforma.

7.5.1.1. Plataforma *hardware*

En este subapartado se describe la plataforma *hardware* utilizada para llevar a cabo el desarrollo del proyecto.

Ordenador personal

El equipo sobre el que se ha desplegado toda la plataforma de desarrollo necesaria para llevar a cabo el proyecto, así como para el desarrollo de la documentación, es un MacBook Pro de 2017 de 13 pulgadas.

Dicho equipo cuenta con un procesador Intel Core i5 a 2,3 GHz de doble núcleo y una memoria de 16 GB 2133 MHz LPDDR3.

El sistema operativo nativo es macOS Monterey Versión 12.3.1.

Servidores

Para alojar el controlador de la red (ONOS), así como compilar e instalar la aplicación DynPaC, se han utilizado varias máquinas virtuales que se encuentran albergadas en los servidores pertenecientes al Grupo de Investigación I2T.

Dichos servidores son de la marca Dell. Concretamente, del modelo PowerEdge R720. Su procesador es un Intel(R) Xeon(R) CPU E5-2699 v4 a 2,20 GHz.

Equipamiento de red

Por último, para el equipamiento de red se han utilizado *switches* de la marca HP. El modelo concreto de dichos *switches* es Aruba 3810M JL171A.

7.5.1.2. Plataforma *software*

En este subapartado se describe la plataforma *software* utilizada para llevar a cabo el desarrollo del proyecto.

Entorno de Desarrollo Integrado (IDE)

En primer lugar, para desarrollar cualquier proyecto *software* de grandes dimensiones como éste, es crucial el uso de un entorno de desarrollo integrado (IDE, de sus siglas en inglés) adecuado a las características del proyecto en particular.

En el caso específico de este proyecto, se ha utilizado el IDE IntelliJ IDEA Ultimate instalado en el sistema operativo nativo del ordenador personal.

Gracias a las funcionalidades avanzadas que ofrece este IDE, se ha podido llevar a cabo el desarrollo *software* de forma remota vía SSH en lugar de directamente sobre la máquina virtual que alberga el proyecto DynPaC.

Herramienta de gestión de proyectos Java

En este punto, cabe mencionar que DynPaC es un proyecto desarrollado mediante Java. En consecuencia, dada la complejidad del mismo, resulta interesante el uso complementario de una herramienta *software* para la gestión y construcción de proyectos Java. En este caso, la herramienta de elección es Maven, de Apache Software Foundation.

Apache Maven es una herramienta de código abierto que permite estandarizar la configuración de un proyecto en todo su ciclo de vida, como por ejemplo en todas las fases de compilación y empaquetado y la instalación de mecanismos de distribución de librerías, para que puedan ser utilizadas por otros desarrolladores y equipos de desarrollo.

Apache Maven utiliza un *Project Object Model* (POM) para describir el proyecto de *software* a construir, sus dependencias de otros módulos y componentes externos, y el orden de construcción de los elementos.

Este modelo se implementa mediante un archivo de extensión `xml` al que suele denominarse: “`pom.xml`”. Se utiliza para administrar código fuente, archivos de configuración, información y roles del desarrollador, sistema de seguimiento de problemas, información de la organización, autorización del proyecto, URL del proyecto, dependencias del proyecto, etc.

El archivo `pom.xml` de un proyecto se ubica en el directorio raíz de dicho proyecto.

Software de red

Por último, en cuanto al equipamiento de red, cabe mencionar dos *softwares* utilizados.

Por un lado, en el plano de control, se utiliza un controlador ONOS. Como ya se adelantaba en el apartado Contexto tecnológico, este tipo de controlador permite la instalación de aplicaciones externas que proporcionan servicios específicos de red, como es DynPaC. Las versiones que se han utilizado en el desarrollo de este proyecto son: 1.15.0, 2.0.0 y 2.1.0.

Por otro lado, para permitir la comunicación entre el plano de datos (*switches* Aruba HP) y el plano de control (ONOS), se hace uso del protocolo de código abierto OpenFlow en su versión 1.3.

7.5.2. Implementación de las fases de la migración

En este apartado se describe en detalle la implementación del diseño incremental de actualización de DynPaC para la integración con SN4I y la incorporación de dispositivos DPP y herramientas de monitorización basadas en telemetría.

7.5.2.1. Fase I

Como ya se ha mencionado anteriormente en este documento, el punto de partida del proyecto de migración es una versión de DynPaC compatible con ONOS v1.10.0. La primera fase del proceso de actualización, tal y como se ha indicado en el subapartado Diseño de la solución, requiere lograr la compatibilidad con ONOS v1.15.0.

Para realizar dicha actualización de versiones, en primer lugar, se debe configurar la versión de ONOS a utilizar en la aplicación de DynPaC.

Para ello, es esencial modificar el archivo `pom.xml` del código de DynPaC. En este archivo, se especifica la versión de ONOS con la que trabaja la aplicación. A continuación, en la Figura 7.3, se muestra la modificación de código realizada en el archivo `pom.xml` para implementar el cambio de versión mencionado. Aparece resaltada en rojo la línea eliminada y en verde la línea añadida.

22	22		<code><properties></code>
23		-	<code><onos.version>1.10.0</onos.version></code>
	23	+	<code><onos.version>1.15.0</onos.version></code>
24	24		<code><onos.app.name>org.geant.dynpac</onos.app.name></code>
25	25		<code><onos.app.origin>GEANT</onos.app.origin></code>

Figura 7.3: Código modificado en el archivo `pom.xml`.

Por otro lado, al realizar dicha actualización de versiones, quedan obsoletas varias clases e interfaces del paquete que proporciona un servicio de interfaz a las aplicaciones de ONOS, como DynPaC. A partir de la versión 1.11.0 de ONOS, este servicio pasa a proporcionarlo otro paquete.

Los paquetes en cuestión son `org.onosproject.incubator.net.intf` y `org.onosproject.net.intf`, respectivamente, y es necesario intercambiar las clases obsoletas por las actualizadas para implementar este cambio de versiones. En concreto, en este proyecto, este cambio afecta a las clases del módulo *Topology Abstraction* mencionadas a continuación.

- Clase `DefaultDynpacTopologyAbstraction.java`: Se eliminan los paquetes mencionados a continuación.
 - `org.onosproject.incubator.net.intf.Interface`.
 - `org.onosproject.incubator.net.intf.InterfaceService`.

Se importan los paquetes mencionados a continuación.

- `org.onosproject.net.intf.Interface`.
- `org.onosproject.net.intf.InterfaceService`.
- Clase `DynpacTopologyAbstraction.java`: Se elimina el paquete mencionado a continuación.
 - `org.onosproject.incubator.net.intf.Interface`.

Se importa el paquete mencionado a continuación.

- `org.onosproject.net.intf.Interface`.

A continuación, en la Figura 7.4 y Figura 7.5, se muestra el código modificado para implementar los cambios mencionados para lograr la compatibilidad de DynPaC con ONOS 1.15.0.

14	- import org.onosproject.incubator.net.intf.Interface;
15	- import org.onosproject.incubator.net.intf.InterfaceService;
14	+ import org.onosproject.net.intf.Interface;
15	+ import org.onosproject.net.intf.InterfaceService;

Figura 7.4: Código modificado en la clase `DefaultDynpacTopologyAbstraction.java`.

```

3      - import org.onosproject.incubator.net intf.Interface;
3      + import org.onosproject.net intf.Interface;
  
```

Figura 7.5: Código modificado en la clase `DynpacTopologyAbstraction.java`.

De forma adicional, cabe mencionar que en las aplicaciones de ONOS posteriores a la versión 1.12.0, es necesaria una dependencia adicional a las requeridas para versiones anteriores. Se trata de una dependencia de Eclipse Source, llamada Minimal JSON, que proporciona un analizador y escritor de JSON. Es necesaria la gestión de esta nueva dependencia para actualizar DynPaC a la versión objetivo de esta primera fase de migración.

Como ya se ha visto en el subapartado Herramienta de gestión de proyectos Java, las dependencias se introducen en el archivo `pom.xml` del código de la aplicación. A continuación, en la Figura 7.6, se muestra el código introducido en el POM para añadir la mencionada dependencia.

```

111 +      <dependency>
112 +          <groupId>com.eclipsesource.minimal-json</groupId>
113 +          <artifactId>minimal-json</artifactId>
114 +          <version>0.9.1</version>
115 +      </dependency>
  
```

Figura 7.6: Dependencia Minimal JSON añadida en el archivo `pom.xml`.

Finalmente, en lo que respecta a este proyecto, a partir de la versión 1.13.1 de ONOS, se sustituye la librería encargada de medir el coste de los enlaces de la red en forma de recuento de saltos. Por lo tanto, se debe gestionar este cambio de librerías dentro del código de DynPaC.

En versiones anteriores, esta función la realizaba el paquete `org.onosproject.net.topology.HopCountLinkWeight`, que a partir de ONOS 1.13.1 pasa a ser sustituido por el paquete `org.onosproject.net.topology.HopCountLinkWeighter`. Esta actualización afecta de forma directa a la clase `DynpacPce.java` del módulo PCE de DynPaC. Esta clase requiere los cambios mostrados, a continuación, en la Figura 7.7.

```

25 25 import org.onosproject.net.Link;
26 26 import org.onosproject.net.Path;
27 27 import org.onosproject.net.provider.ProviderId;
28 28 - import org.onosproject.net.topology.HopCountLinkWeight;
29 28 + import org.onosproject.net.topology.HopCountLinkWeigher;
29 29 import org.onosproject.net.topology.LinkWeigher;
30 30 import org.onosproject.net.topology.TopologyEdge;
31 31 import org.onosproject.net.topology.TopologyVertex;
... .. @@ -46,7 +46,8 @@ import java.util.Set;
46 46 import java.util.SortedMap;
47 47 import java.util.TreeMap;
48 48
49 49 - import static org.onosproject.net.topology.AdapterLinkWeigher.adapt;
50 49 + //import static org.onosproject.net.topology.AdapterLinkWeigher.adapt;
51 50 + import static org.geant.dynpac.pce.AdapterLinkWeigher.adapt;
52 51 import static org.slf4j.LoggerFactory.getLogger;
53 52
54 53 @Component
... .. @@ -177,7 +178,7 @@ public class DynpacPce implements DynpacPceService {
177 178
178 179         if (!src.equals(dst)) {
179 180
180 180 -         LinkWeigher hopCountWeigher = adapt(new
181 181 +         LinkWeigher hopCountWeigher = adapt(new
182 182 HopCountLinkWeigher(graph.getVertexes().size()));
183 182 HopCountLinkWeigher(graph.getVertexes().size()));
  
```

Figura 7.7: Código modificado en la clase *DynpacPce.java*.

7.5.2.2. Fase II

Tras la generación de versiones de ONOS v1.15, por parte de ONF se dio un salto importante en términos de actualización, que dio lugar a la generación ONOS v2.0. Este gran salto afecta a DynPaC en múltiples aspectos.

En primer lugar, algunas de las dependencias ya existentes en versiones anteriores, quedan obsoletas. Por tanto, es necesario sustituirlas por versiones más novedosas de las mismas.

Para ello, se deben modificar las dependencias pertinentes en el archivo pom.xml, tal y como se puede observar en la Figura 7.8.

En segundo lugar, la clase que desempeñaba la función de abstracción base de los comandos del *shell* Karaf, se convierte en una clase abstracta. Por lo tanto, las clases que la extienden también deben convertirse en abstractas.

La clase en cuestión es *AbstractShellCommand.java*, y, concretamente, en este proyecto, la clase afectada es *DynpacConfigurationDefaultVlanManage.java* del módulo *Service Manager*. En la Figura 7.9, se muestra la modificación realizada.

Por último, el método encargado de ejecutar los comandos abstraídos del *shell* Karaf es sustituido por otro, dentro de la misma clase.

La clase implicada es, de nuevo, la clase `AbstractShellCommand.java` y el método obsoleto es `execute()`. Este método se debe sustituir por el nuevo método `doExecute()`. Este cambio afecta a las clases relacionadas con el CLI de DynPaC, pertenecientes a los módulos *Service Manager*, *Topology Abstractor* y *Network Programmer*, mencionadas a continuación.

- Clase `DynpacConfigurationDefaultVlanManage.java`.
- Clase `DynpacConfigurationFlowPriorityShow.java`.
- Clase `DynpacConfigurationShow.java`.
- Clase `DynpacNetworkSnapshotShow.java`.
- Clase `DynpacServiceActiveResourceShow.java`.
- Clase `DynpacServiceCommit.java`.
- Clase `DynpacServiceRelease.java`.
- Clase `DynpacServiceRequest.java`.
- Clase `DynpacServiceResubmit.java`.
- Clase `DynpacServiceTerminate.java`.
- Clase `DynpacServicesList.java`.
- Clase `DynpacTopologyEdgeNodesShow.java`.
- Clase `DynpacTopologyShow.java`.

En todas ellas, el cambio a realizar en términos de código es el mismo: se sustituye el método `execute()` por el método `doExecute()`. En la Figura 7.10, se muestra, a modo de ejemplo, dicho cambio realizado en la clase `DynpacConfigurationExceptionVlanManage.java`.

72	73	<dependency>
73	74	<groupId>org.apache.karaf.shell</groupId>
74	75	<artifactId>org.apache.karaf.shell.console</artifactId>
75	-	<version>3.0.3</version>
	76	+ <version>4.2.2</version>
	77	+ <scope>provided</scope>
	78	+ </dependency>
	79	+ <dependency>
	80	+ <groupId>org.apache.karaf.shell</groupId>
	81	+ <artifactId>org.apache.karaf.shell.commands</artifactId>
	82	+ <version>4.2.2</version>
76	83	<scope>provided</scope>
77	84	</dependency>
78	85	
...	...	@@ -209,7 +216,7 @@
209	216	<plugin>
210	217	<groupId>org.apache.felix</groupId>
211	218	<artifactId>maven-bundle-plugin</artifactId>
212	-	<version>3.0.1</version>
	219	+ <version>4.0.0</version>
213	220	<extensions>>true</extensions>
214	221	<configuration>
215	222	<instructions>
...	...	@@ -218,7 +225,7 @@
218	225	<_wab>src/main/webapp</_wab>
219	226	<Import-Package>
220	227	org.glassfish.jersey.servlet,
221	-	*
	228	+ *;resolution:=optional
222	229	</Import-Package>
223	230	<Include-Resource>
224	231	WEB-INF/classes/apidoc/swagger.json=target/swagger.json,

Figura 7.8: Código modificado en el archivo pom.xml.

13	13	@Command(scope = "dynpac", name = "dynpac-configuration-default-vlan-manage",
14	14	description = "Manages the default VLAN range of the DynPaC configuration")
15	-	public class DynpacConfigurationDefaultVlanManage extends AbstractShellCommand {
	15	+ public abstract class DynpacConfigurationDefaultVlanManage extends AbstractShellCommand {

Figura 7.9: Código modificado en DynpacConfigurationDefaultVlanManage.java.

32	32	@Override
33	-	protected void execute() {
	33	+ protected void doExecute() {

Figura 7.10: Código modificado en DynpacConfigurationExceptionVlanManage.java.

Por último, una vez más, es esencial indicar en el código de la aplicación DynPaC la nueva versión de ONOS que va a utilizar.

Para ello, se debe modificar en el archivo pom.xml. A continuación, en la Figura 7.11, se muestra la modificación de código realizada en el archivo pom.xml para implementar el cambio de versión mencionado.

22	22		<code><properties></code>
23	-		<code><onos.version>1.15.0</onos.version></code>
	23	+	<code><onos.version>2.0.0</onos.version></code>
24	24		<code><onos.app.name>org.geant.dynpac</onos.app.name></code>
25	25		<code><onos.app.origin>GEANT</onos.app.origin></code>

Figura 7.11: Código modificado en el archivo `pom.xml`.

7.5.2.3. Fase III

El diseño incremental realizado en este TFM especifica que en este tercer y último incremento es necesario actualizar DynPaC a una versión compatible con ONOS v2.1.0.

Como ya se ha mencionado en el subapartado anterior, la generación 2.0 de ONOS implementa importantes cambios en el *software* del controlador, así como en la API Java que proporciona para aplicaciones externas (como es DynPaC, en el contexto de este proyecto). En consecuencia, se requieren nuevas dependencias que permitan la compatibilidad de DynPaC con ONOS v2.1.0.

En primer lugar, se debe proceder a la inclusión de dichas dependencias necesarias, que, en el escenario de este TFM, son las indicadas a continuación.

- **ONOS Core Common:** Se trata de una dependencia necesaria para el correcto funcionamiento de otras dependencias tales como la Interfaz de la Línea de Comandos (CLI, de sus siglas en inglés) de ONOS, la API REST, utilidades de diferentes *drivers*, etc. [8]
- **ONOS ONlab Misc:** Se trata de una dependencia necesaria para el correcto funcionamiento de otras dependencias tales como la API de ONOS, la comunicación REST de ONlab, diversos *serializers*, el CLI de ONOS y el propio ONOS Core Common. [9]

Como ya se ha visto anteriormente, las dependencias se introducen en el archivo `pom.xml` del código de la aplicación. A continuación, en la Figura 7.12Figura 7.6, se muestra el código introducido en el POM para añadir las mencionadas dependencias.

```

157 +     <dependency>
158 +         <groupId>org.onosproject</groupId>
159 +         <artifactId>onlab-misc</artifactId>
160 +         <version>${onos.version}</version>
161 +         <scope>provided</scope>
162 +     </dependency>
163 +     <dependency>
164 +         <groupId>org.onosproject</groupId>
165 +         <artifactId>onlab-misc</artifactId>
166 +         <version>${onos.version}</version>
167 +         <scope>test</scope>
168 +         <classifier>tests</classifier>
169 +     </dependency>
170 +
171 +     <dependency>
172 +         <groupId>org.onosproject</groupId>
173 +         <artifactId>onos-core-common</artifactId>
174 +         <version>${onos.version}</version>
175 +         <scope>provided</scope>
176 +     </dependency>
177 +     <dependency>
178 +         <groupId>org.onosproject</groupId>
179 +         <artifactId>onos-core-common</artifactId>
180 +         <version>${onos.version}</version>
181 +         <scope>test</scope>
182 +         <classifier>tests</classifier>
183 +     </dependency>
  
```

Figura 7.12: Código modificado en el archivo pom.xml.

En segundo lugar, una vez más, es esencial indicar en el código de la aplicación DynPaC la versión de ONOS que va a utilizar.

Para ello, se debe modificar en el archivo pom.xml del. A continuación, en la Figura 7.13, se muestra la modificación de código realizada en el archivo pom.xml para implementar el cambio de versión mencionado.

22	22		<properties>
23		-	<onos.version>2.0.0</onos.version>
	23	+	<onos.version>2.1.0</onos.version>
24	24		<onos.app.name>org.geant.dynpac</onos.app.name>
25	25		<onos.app.origin>GEANT</onos.app.origin>

Figura 7.13: Código modificado en el archivo pom.xml.

7.6. Pruebas de validación

A partir de la implementación descrita previamente, se han desarrollado una serie de pruebas para mostrar su correcto funcionamiento y verificar su validez. Para ello, a continuación, se describe el escenario diseñado para las pruebas y se definen las pruebas realizadas, así como los resultados obtenidos en cada una de ellas. Finalmente, se comentan las conclusiones obtenidas.

7.6.1. Escenario diseñado para las pruebas

Para el desarrollo de las pruebas de validación, se ha llevado a cabo el despliegue de una maqueta de red con equipamiento real. Esta maqueta constituye un escenario con las características necesarias para poder probar sobre él la validez de la propuesta de solución presentada en este proyecto.

Esta maqueta consta de dos *hosts*, cada uno conectado a un *switch*. Ambos *switches* son *switches* OpenFlow y están conectados entre sí mediante un cable ethernet. Asimismo, la maqueta cuenta con un controlador ONOS que gestiona los *switches* OpenFlow.

De esta manera, esta maqueta sirve para emular el escenario de comunicación de dos usuarios de la red SN4I. A continuación, en la Figura 7.14, se muestra un diagrama que ilustra el escenario diseñado para las pruebas de validación.

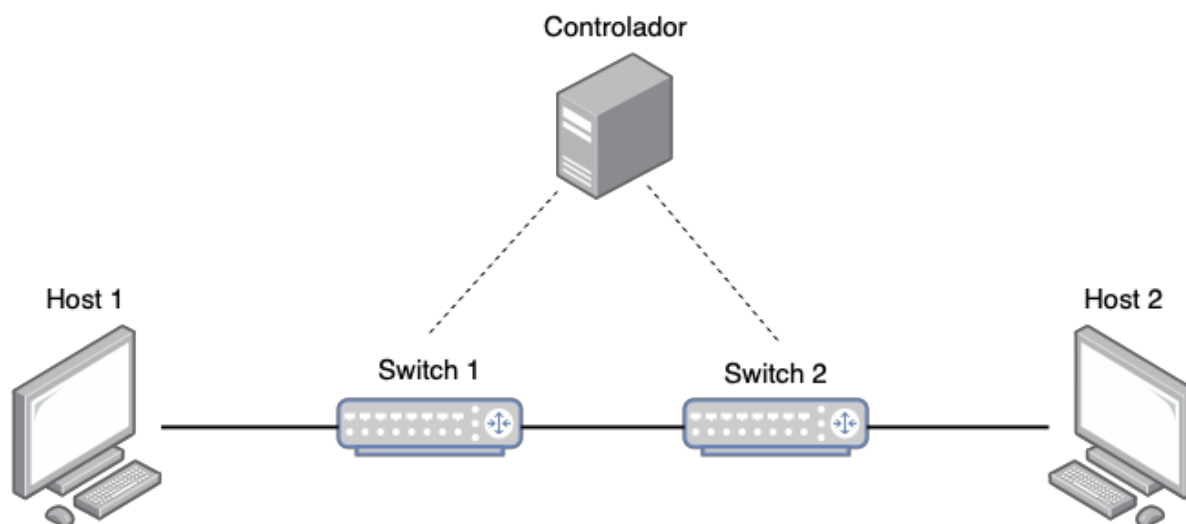


Figura 7.14: Escenario diseñado para las pruebas de validación.

En cuanto a la configuración de la maqueta de red, cabe recordar la propiedad de DynPaC de aislar los servicios entre sí mediante el encapsulamiento del tráfico de cada servicio en una VLAN independiente. En consecuencia, se han habilitado una serie de VLANs en los enlaces del escenario de red para que DynPaC pueda funcionar adecuadamente durante las pruebas de validación. A continuación, en la Figura 7.15, se muestra la configuración de las VLANs llevada a cabo en el escenario de red diseñado para las pruebas de validación.

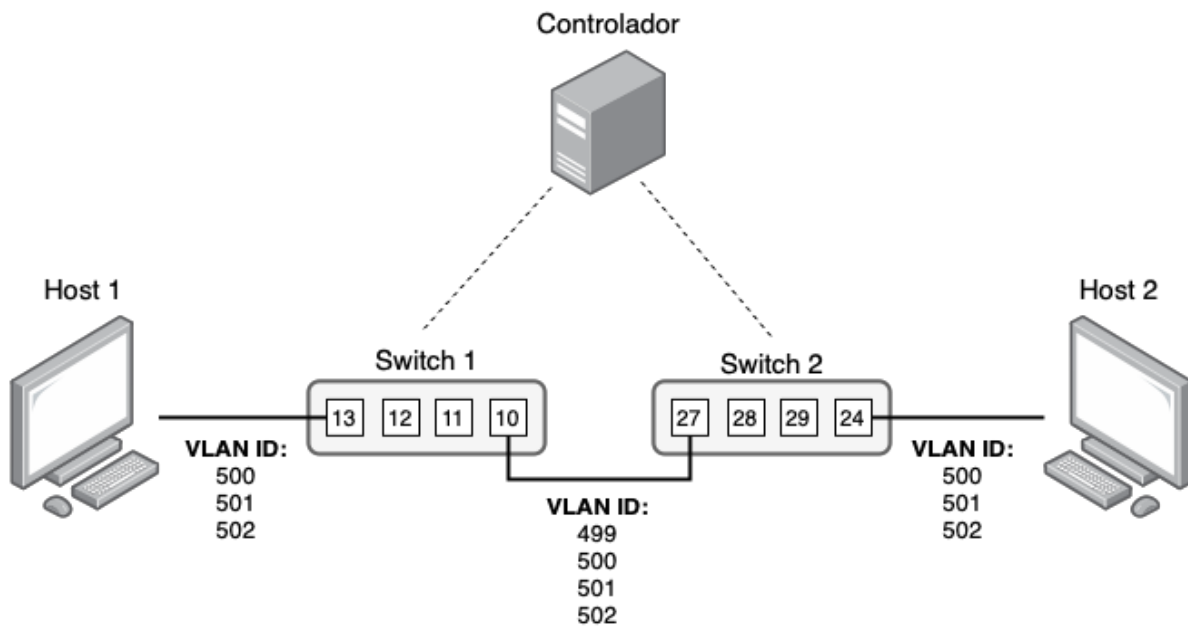


Figura 7.15: VLANs configuradas en el escenario diseñado para las pruebas de validación.

7.6.2. Definición de las pruebas

A continuación, en este subapartado, se describen las pruebas definidas para mostrar la validez de la solución de actualización propuesta en este proyecto.

7.6.2.1. Prueba de compilación

La primera prueba para verificar la validez de la solución es comprobar que es posible compilar el código de DynPaC de manera exitosa, es decir, sin que la compilación reporte ningún tipo de error.

Para ello, en primer lugar, para compilar DynPaC es necesario ejecutar el comando expuesto a continuación:

```
ubuntu@onos-dynpac:~/dynpac$ mvn clean install
```

Seguidamente, se debe comprobar que el resultado de la compilación es “BUILD SUCCESS”.

7.6.2.2. Prueba de instalación

Una vez compilada la aplicación, se debe demostrar que es posible la instalación de DynPaC como aplicación de ONOS.

Para ello, es necesario ejecutar el comando expuesto a continuación:

```
ubuntu@onos-dynpac:~/dynpac$ /opt/onos/bin/onos-app 10.98.1.212  
reinstall! /home/ubuntu/dynpac/target/dynpac-1.6.30.oar
```

Una vez ejecutado dicho comando, se debe acceder al menú de aplicaciones de ONOS para comprobar que la aplicación DynPaC se encuentra instalada y activa en el controlador.

7.6.2.3. Prueba de alta de un servicio

Por último, para mostrar que DynPaC es compatible con la versión de ONOS pertinente y, por tanto, se ha integrado adecuadamente con el escenario SN4I correspondiente a cada fase de la migración diseñada en este proyecto, se debe demostrar que es posible llevar a cabo la creación y configuración de un servicio.

Para ello, se debe acceder a la GUI de DynPaC habilitada dentro de la propia GUI de ONOS y dirigirse a la sección “*Service Creation*”. Allí, se deberán rellenar todos los campos requeridos, indicando las características del servicio a crear, y se procederá a dar de alta el servicio en cuestión (pulsando el botón “*Create Service*”).

Finalmente, se deberá activar el servicio creado desde la pestaña “*List Services*”, mediante el botón “*Commit*” y comprobar, desde la pestaña “*Service Info*” que los flujos insertados son los adecuados.

7.6.3. Resultados de las pruebas

A continuación, en el presente subapartado, se exponen los resultados obtenidos en las pruebas de validación, anteriormente descritas, realizadas al final de cada una de las fases de migración de este proyecto.

7.6.3.1. Fase I

Al finalizar la Fase I de migración, la aplicación DynPaC debe ser compatible con ONOS v1.15.0. A continuación, se muestran los resultados obtenidos en las pruebas realizadas para verificar dicha compatibilidad.

Prueba de compilación

Tras la ejecución del comando de compilación, el resultado es el esperado: “*BUILD SUCCESS*”, tal y como se muestra, a continuación, en la Figura 7.16.

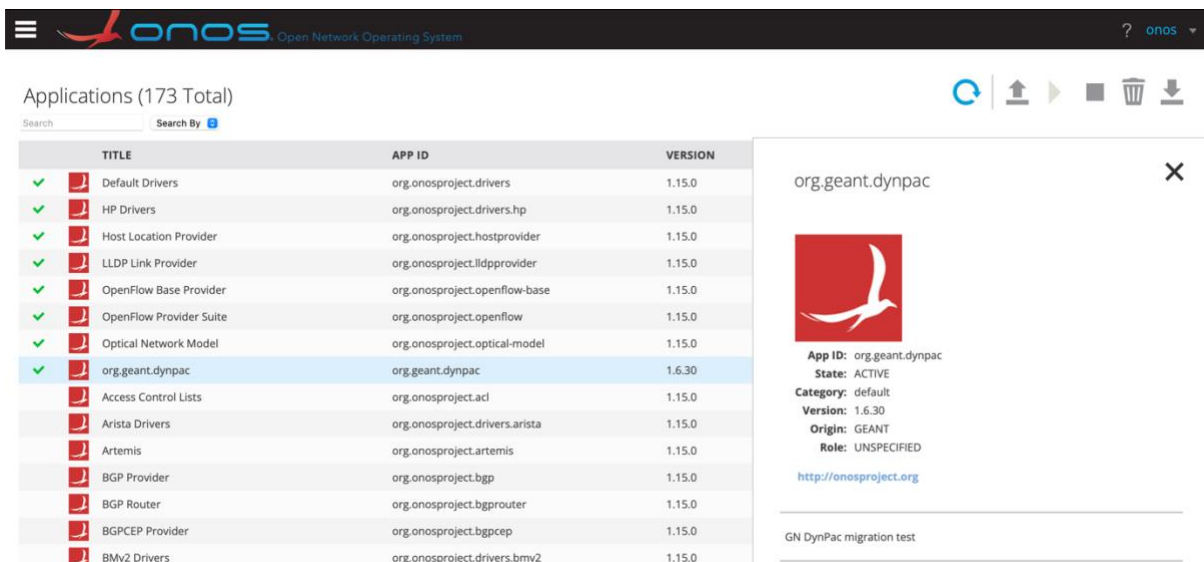
```

[INFO] --- maven-resources-plugin:3.2.0:testResources (default-testResources) @ dynpac ---
[INFO] Not copying test resources
[INFO] --- maven-compiler-plugin:2.5.1:testCompile (default-testCompile) @ dynpac ---
[INFO] Not compiling test sources
[INFO] --- maven-surefire-plugin:3.0.0-M5:test (default-test) @ dynpac ---
[INFO] Tests are skipped.
[INFO] --- maven-bundle-plugin:3.0.1:bundle (default-bundle) @ dynpac ---
[INFO] --- onos-maven-plugin:1.10:app (app) @ dynpac ---
[INFO] Building ONOS application package for org.geant.dynpac (v1.6.30)
[INFO] --- maven-install-plugin:3.0.0-M1:install (default-install) @ dynpac ---
[INFO] Installing /home/ubuntu/dynpac/target/dynpac-1.6.30.jar to /home/ubuntu/.m2/repository/org/geant/dynpac/1.6.30/dynpac-1.6.30.jar
[INFO] Installing /home/ubuntu/dynpac/pom.xml to /home/ubuntu/.m2/repository/org/geant/dynpac/1.6.30/dynpac-1.6.30.pom
[INFO] Installing /home/ubuntu/dynpac/target/dynpac-1.6.30.oar to /home/ubuntu/.m2/repository/org/geant/dynpac/1.6.30/dynpac-1.6.30.oar
[INFO] --- maven-bundle-plugin:3.0.1:install (default-install) @ dynpac ---
[INFO] Installing org/geant/dynpac/1.6.30/dynpac-1.6.30.jar
[INFO] Writing OBR metadata
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 14.030 s
[INFO] Finished at: 2023-06-02T16:50:58Z
[INFO] -----
  
```

Figura 7.16: Resultado de la compilación de DynPaC compatible con ONOS v1.15.0.

Prueba de instalación

Tras la ejecución del comando de instalación, la aplicación de DynPaC aparece activa (“ACTIVE”) dentro de las aplicaciones instaladas en ONOS, tal y como se muestra, a continuación, en la Figura 7.17.



The screenshot shows the ONOS Open Network Operating System interface. At the top, there is a navigation bar with the ONOS logo and the text 'Open Network Operating System'. Below the navigation bar, there is a section titled 'Applications (173 Total)' with a search bar and a 'Search By' dropdown. A table lists various applications, including 'org.geant.dynpac' which is highlighted in blue. To the right of the table, a detailed view of the 'org.geant.dynpac' application is shown, including its logo, App ID, State (ACTIVE), Category, Version (1.6.30), Origin (GEANT), and Role (UNSPECIFIED). The URL 'http://onosproject.org' is also visible.

TITLE	APP ID	VERSION
Default Drivers	org.onosproject.drivers	1.15.0
HP Drivers	org.onosproject.drivers.hp	1.15.0
Host Location Provider	org.onosproject.hostprovider	1.15.0
LLDP Link Provider	org.onosproject.lldpprovider	1.15.0
OpenFlow Base Provider	org.onosproject.openflow-base	1.15.0
OpenFlow Provider Suite	org.onosproject.openflow	1.15.0
Optical Network Model	org.onosproject.optical-model	1.15.0
org.geant.dynpac	org.geant.dynpac	1.6.30
Access Control Lists	org.onosproject.acl	1.15.0
Arista Drivers	org.onosproject.drivers.arista	1.15.0
Artemis	org.onosproject.artemis	1.15.0
BGP Provider	org.onosproject.bgp	1.15.0
BGP Router	org.onosproject.bgprouter	1.15.0
BGPCEP Provider	org.onosproject.bgpcep	1.15.0
BMV2 Drivers	org.onosproject.drivers.bmv2	1.15.0

Figura 7.17: Aplicación DynPaC instalada y activa en ONOS v1.15.0.

Prueba de alta de un servicio

Con la aplicación correctamente instalada y activa, es posible acceder a la GUI de la aplicación (accesible desde la propia GUI de ONOS v1.15.0). Desde esta interfaz gráfica, se ha realizado una prueba de alta de un servicio entre el *Host 1* y el *Host 2* del escenario de red desplegado, introduciendo la información de configuración pertinente en la pestaña “Service Creation”, tal y como se puede observar, a continuación, en la Figura 7.18.

Bandwidth on Demand

List Services | **Service Creation** | **Service Info** | **Configuration**

Name: PruebaTFM_IsabelPlaza

Source STP: of:01f41c98ec99be80 | 13 | 500

Destination STP: of:01f4eceb81b1f80 | 24 | 500

Country: ES-Spain

Time zone: Europe/Madrid

Bandwidth: 100

Resilience

Protected Service

Schedule

Start Date: 25/05/2023 | + 1 Hour | + 1 Day | + 1 Week | + 1 Month | Current Date

Start Time: 16:50 | - 1 Hour | - 1 Day | - 1 Week | - 1 Month

End Date: 25/05/2023 | + 1 Hour | + 1 Day | + 1 Week | + 1 Month | Current Date

End Time: 17:50 | - 1 Hour | - 1 Day | - 1 Week | - 1 Month

Developed by the GEANT project

Figura 7.18: Alta de un servicio DynPaC en ONOS v1.15.0.

Una vez dado de alta el servicio, se ha comprobado que éste se encuentra activo, desde la pestaña “List Services”, la cual se muestra, a continuación, en la Figura 7.19.

Bandwidth on Demand

List Services | **Service Creation** | **Service Info** | **Configuration**

Items (1 total)

ID	COOKIE	NAME	START	END	INGRESS	EGRESS	BW	STATE	ACTIONS
0: <input type="checkbox"/>	1311673391471656960	PruebaTFM_IsabelPlaza	25-05-2023 14:50:00	25-05-2023 15:50:00	of:01f41c98ec99be80 Port: 13 / VLAN: 500	of:01f4eceb81b1f80 Port: 24 / VLAN: 500	100	ACTIVE	<input type="button" value="Delete"/> <input type="button" value="Release"/> <input type="button" value="Topo"/> <input type="button" value="Info"/>

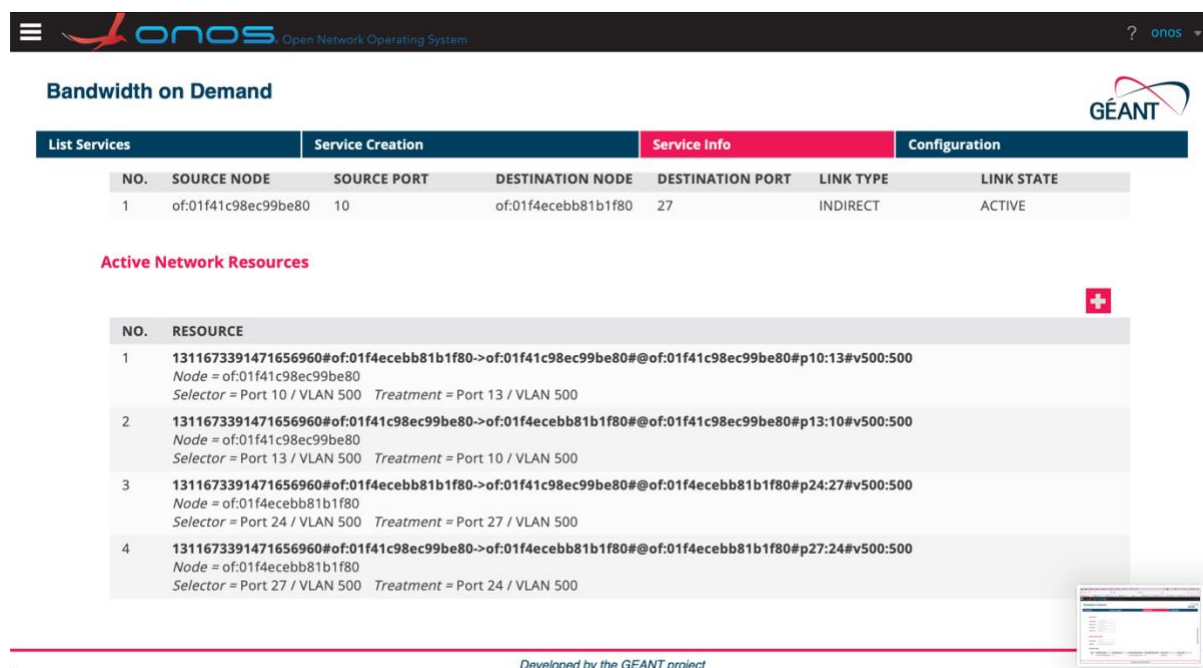
Developed by the GEANT project

Figura 7.19: Servicio DynPaC activo en ONOS v1.15.0.

Finalmente, se ha comprobado que los flujos introducidos son los adecuados. En este punto, cabe recordar que, en el escenario de red desplegado para la realización de estas pruebas (véase la Figura 7.14) la conectividad física se proporciona de la siguiente manera:

- **Host 1:** Se encuentra conectado al puerto 13 de *Switch 1*.
- **Host 2:** Se encuentra conectado al puerto 24 de *Switch 2*.
- **Switch 1 y Switch 2:** Están conectados entre sí a través del puerto 10 y el puerto 27, respectivamente.

En consecuencia, se deben insertar un flujo con la etiqueta VLAN indicado en la configuración del servicio (VLAN ID: 500) en cada uno de los mencionados puertos. Atendiendo a la Figura 7.20, adjunta a continuación, se puede observar que los flujos instalados son, efectivamente, los adecuados.



The screenshot shows the ONOS Bandwidth on Demand configuration page. It features a table for 'List Services' and a section for 'Active Network Resources'.

NO.	SOURCE NODE	SOURCE PORT	DESTINATION NODE	DESTINATION PORT	LINK TYPE	LINK STATE
1	of:01f41c98ec99be80	10	of:01f4eceb81b1f80	27	INDIRECT	ACTIVE

NO.	RESOURCE
1	1311673391471656960#of:01f4eceb81b1f80->of:01f41c98ec99be80#@of:01f41c98ec99be80#p10:13#v500:500 Node = of:01f41c98ec99be80 Selector = Port 10 / VLAN 500 Treatment = Port 13 / VLAN 500
2	1311673391471656960#of:01f41c98ec99be80->of:01f4eceb81b1f80#@of:01f41c98ec99be80#p13:10#v500:500 Node = of:01f41c98ec99be80 Selector = Port 13 / VLAN 500 Treatment = Port 10 / VLAN 500
3	1311673391471656960#of:01f4eceb81b1f80->of:01f41c98ec99be80#@of:01f4eceb81b1f80#p24:27#v500:500 Node = of:01f4eceb81b1f80 Selector = Port 24 / VLAN 500 Treatment = Port 27 / VLAN 500
4	1311673391471656960#of:01f41c98ec99be80->of:01f4eceb81b1f80#@of:01f4eceb81b1f80#p27:24#v500:500 Node = of:01f4eceb81b1f80 Selector = Port 27 / VLAN 500 Treatment = Port 24 / VLAN 500

Figura 7.20: Flujos instalados por el servicio DynPaC.

Conclusiones derivadas de las pruebas realizadas en la Fase I

Las pruebas realizadas sobre la versión de DynPaC resultante del diseño e implementación de la fase I de la migración objeto de este proyecto muestran que se ha logrado la compatibilidad de DynPaC con ONOS v1.15.0.

En consecuencia, de acuerdo con el Diseño de la solución expuesto en este documento, se puede afirmar que en esta fase se ha logrado dotar de programabilidad al plano de datos de la red SN4I, mediante el soporte de la tecnología P4Runtime. Asimismo, también se ha conseguido la mejora de la monitorización de la red mediante la incorporación de herramientas de telemetría, a través de la compatibilidad con INT. Ambos objetivos se han cumplido actualizando DynPaC de la versión originalmente compatible con ONOS v1.10.0 a una nueva versión compatible con ONOS v1.15.0.

7.6.3.2. Fase II

Al finalizar la Fase II de la migración, la aplicación DynPaC debe ser compatible con ONOS v2.0.0. A continuación, se muestran los resultados obtenidos en las pruebas realizadas para verificar dicha compatibilidad.

Prueba de compilación

Tras la ejecución del comando de compilación, el resultado es el esperado: “*BUILD SUCCESS*”, tal y como se muestra, a continuación, en la Figura 7.21.

```
[INFO] --- maven-resources-plugin:3.2.0:testResources (default-testResources) @ dynpac ---
[INFO] Not copying test resources
[INFO] --- maven-compiler-plugin:2.5.1:testCompile (default-testCompile) @ dynpac ---
[INFO] Not compiling test sources
[INFO] --- maven-surefire-plugin:3.0.0-M5:test (default-test) @ dynpac ---
[INFO] Tests are skipped.
[INFO] --- maven-bundle-plugin:4.0.0:bundle (default-bundle) @ dynpac ---
[INFO] --- onos-maven-plugin:1.10:app (app) @ dynpac ---
[INFO] Building ONOS application package for org.geant.dynpac (v1.6.30)
[INFO] --- maven-install-plugin:3.0.0-M1:install (default-install) @ dynpac ---
[INFO] Installing /home/ubuntu/dynpac/target/dynpac-1.6.30.jar to /home/ubuntu/.m2/repository/org/geant/dynpac/1.6.30/dynpac-1.6.30.jar
[INFO] Installing /home/ubuntu/dynpac/pom.xml to /home/ubuntu/.m2/repository/org/geant/dynpac/1.6.30/dynpac-1.6.30.pom
[INFO] Installing /home/ubuntu/dynpac/target/dynpac-1.6.30.oar to /home/ubuntu/.m2/repository/org/geant/dynpac/1.6.30/dynpac-1.6.30.oar
[INFO] --- maven-bundle-plugin:4.0.0:install (default-install) @ dynpac ---
[INFO] Installing org/geant/dynpac/1.6.30/dynpac-1.6.30.jar
[INFO] Writing OBR metadata
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 20.010 s
[INFO] Finished at: 2023-06-02T17:53:37Z
[INFO] -----
```

Figura 7.21: Resultado de la compilación de DynPaC compatible con ONOS v2.0.0.

Prueba de instalación

Tras la ejecución del comando de instalación, la aplicación de DynPaC aparece activa (“*ACTIVE*”) dentro de las aplicaciones instaladas en ONOS, tal y como se muestra, a continuación, en la Figura 7.22.

The screenshot shows the ONOS Applications management interface. A table lists various applications, with 'org.geant.dynpac' highlighted in blue. To the right, a detailed view of this application is shown, including its logo, ID, state (ACTIVE), category, version (1.6.30), origin (GEANT), and role (UNSPECIFIED). It also lists features like 'dynpac' and a note about 'GN DynPac migration test'.

TITLE	APP ID	VERSION
Default Drivers	org.onosproject.drivers	1.15.0
HP Drivers	org.onosproject.drivers.hp	1.15.0
Host Location Provider	org.onosproject.hostprovider	1.15.0
LLDP Link Provider	org.onosproject.ldpprovider	1.15.0
OpenFlow Base Provider	org.onosproject.openflow-base	1.15.0
OpenFlow Provider Suite	org.onosproject.openflow	1.15.0
Optical Network Model	org.onosproject.optical-model	1.15.0
org.geant.dynpac	org.geant.dynpac	1.6.30
Access Control Lists	org.onosproject.acl	1.15.0
Arista Drivers	org.onosproject.drivers.arista	1.15.0
Artemis	org.onosproject.artemis	1.15.0
BGP Provider	org.onosproject.bgp	1.15.0
BGP Router	org.onosproject.bgprouter	1.15.0
BGPCEP Provider	org.onosproject.bgpcep	1.15.0
BMv2 Drivers	org.onosproject.drivers.bmv2	1.15.0
Barefoot Drivers	org.onosproject.drivers.barefoot	1.15.0
Basic Optical Drivers	org.onosproject.drivers.optical	1.15.0
Basic Pipelines	org.onosproject.pipelines.basic	1.15.0
CORD Support	org.onosproject.cord-support	1.15.0
Factory	org.onosproject.factory	1.15.0

Figura 7.22: Aplicación DynPaC instalada y activa en ONOS v2.0.0.

Prueba de alta de un servicio

De forma análoga a la prueba realizada en la Fase I, desde la GUI de DynPaC, se ha realizado una prueba de alta de un servicio entre el *Host 1* y el *Host 2* del escenario de red desplegado, introduciendo la información de configuración pertinente en la pestaña “*Service Creation*”, tal y como se puede observar, a continuación, en la Figura 7.23.

The screenshot shows the 'Service Creation' configuration page. It includes fields for Name, Source STP, Destination STP, Country, Time zone, and Bandwidth. Below these are sections for Resilience (Protected Service) and Schedule (Start Date, Start Time, End Date, End Time) with navigation buttons for time intervals.

Service Creation

Name: PruebaTFM_Fase2

Source STP: of:01f31c98ec99be80 | 13 | 500

Destination STP: of:01f3eceb81b1f80 | 24 | 500

Country: ES-Spain

Time zone: Europe/Madrid

Bandwidth: 100

Resilience

Protected Service

Schedule

Start Date: 02/06/2023 | +1 Hour | +1 Day | +1 Week | +1 Month | Current Date

Start Time: 20:16 | -1 Hour | -1 Day | -1 Week | -1 Month

End Date: 02/06/2023 | +1 Hour | +1 Day | +1 Week | +1 Month | Current Date

End Time: 21:16 | -1 Hour | -1 Day | -1 Week | -1 Month

Figura 7.23: Alta de un servicio DynPaC en ONOS v2.0.0.

Una vez dado de alta el servicio, se ha comprobado que éste se encuentra activo, desde la pestaña “List Services”, la cual se muestra, a continuación, en la Figura 7.19.

Developed by the GEANT project

Figura 7.24: Servicio DynPaC activo en ONOS v2.0.0.

Finalmente, atendiendo a la Figura 7.20, adjunta a continuación, se comprueba que los flujos instalados son, efectivamente, los adecuados: en los puertos 10 y 13 del Switch 1, y en los puertos 24 y 27 del Switch 2, todos ellos con el VLAN ID 500.

Developed by the GEANT project

Figura 7.25: Flujos instalados por el servicio DynPaC.

Conclusiones derivadas de las pruebas realizadas en la Fase I

Las pruebas realizadas sobre la versión de DynPaC resultante del diseño e implementación de la fase II de la migración objeto de este proyecto muestran que se ha logrado la compatibilidad de DynPaC con ONOS v2.0.0.

En consecuencia, de acuerdo con el Diseño de la solución expuesto en este documento, se puede afirmar que se ha mejorado la programabilidad en el plano de datos proporcionada en la Fase I, incluyendo el soporte del sistema operativo Stratum en los dispositivos de red de la arquitectura SN4I.

Cabe recordar que el sistema operativo Stratum permite programar dispositivos del plano de datos mediante el lenguaje de programación P4, haciendo posible así la implementación de DPP.

De forma adicional, al actualizar DynPaC a ONOS v2.0.0, indirectamente se han implementado mejoras en la interfaz gráfica de usuario de ONOS, a través de la cual se gestiona la aplicación DynPaC. En consecuencia, se aportan mejoras en la GUI del plano de control del escenario de red SN4I.

7.6.3.3. Fase III

Al finalizar la Fase III de la migración, la aplicación DynPaC debe ser compatible con ONOS v2.1.0. A continuación, se muestran los resultados obtenidos en las pruebas realizadas para verificar dicha compatibilidad.

Prueba de compilación

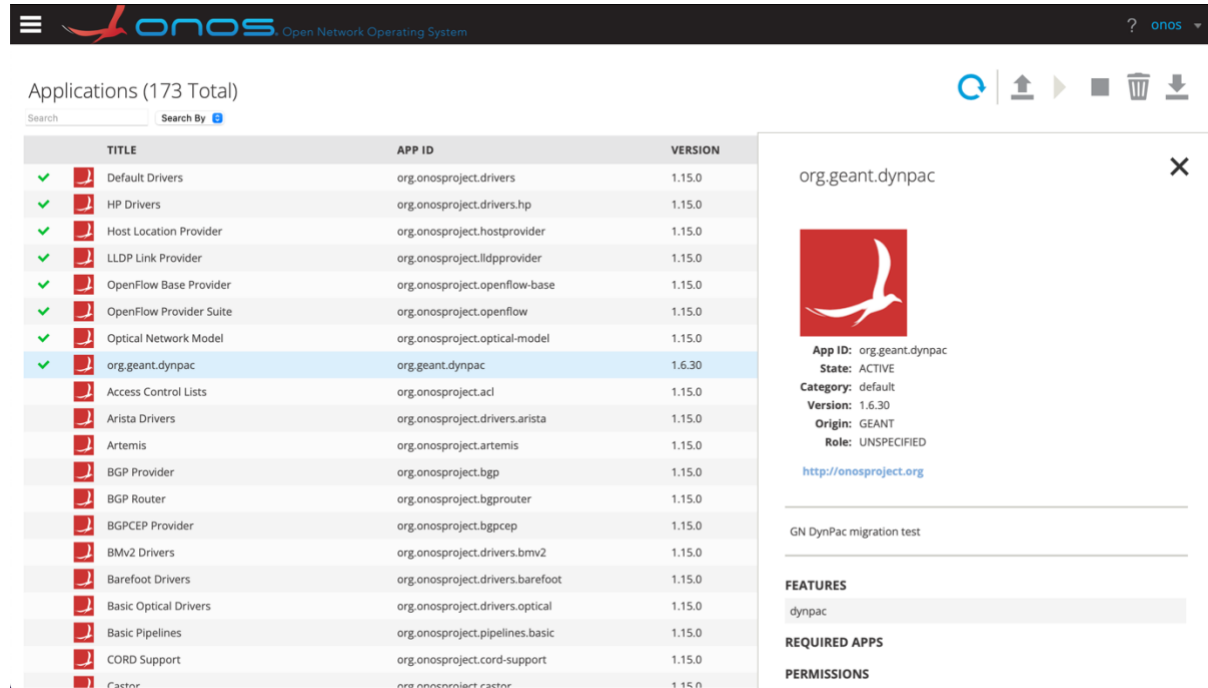
Tras la ejecución del comando de compilación, el resultado es el esperado: “*BUILD SUCCESS*”, tal y como se muestra, a continuación, en la Figura 7.26.

```
[INFO] --- maven-resources-plugin:3.2.0:testResources (default-testResources) @ dynpac ---
[INFO] Not copying test resources
[INFO] --- maven-compiler-plugin:2.5.1:testCompile (default-testCompile) @ dynpac ---
[INFO] Not compiling test sources
[INFO] --- maven-surefire-plugin:3.0.0-M5:test (default-test) @ dynpac ---
[INFO] Tests are skipped.
[INFO] --- maven-bundle-plugin:4.0.0:bundle (default-bundle) @ dynpac ---
[INFO] --- onos-maven-plugin:1.10:app (app) @ dynpac ---
[INFO] Building ONOS application package for org.geant.dynpac (v1.6.30)
[INFO] --- maven-install-plugin:3.0.0-M1:install (default-install) @ dynpac ---
[INFO] Installing /home/ubuntu/dynpac/target/dynpac-1.6.30.jar to /home/ubuntu/.m2/repository/org/geant/dynpac/1.6.30/dynpac-1.6.30.jar
[INFO] Installing /home/ubuntu/dynpac/pom.xml to /home/ubuntu/.m2/repository/org/geant/dynpac/1.6.30/dynpac-1.6.30.pom
[INFO] Installing /home/ubuntu/dynpac/target/dynpac-1.6.30.oar to /home/ubuntu/.m2/repository/org/geant/dynpac/1.6.30/dynpac-1.6.30.oar
[INFO] --- maven-bundle-plugin:4.0.0:install (default-install) @ dynpac ---
[INFO] Installing org/geant/dynpac/1.6.30/dynpac-1.6.30.jar
[INFO] Writing OBR metadata
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 13.237 s
[INFO] Finished at: 2023-06-02T18:04:14Z
[INFO] -----
```

Figura 7.26: Resultado de la compilación de DynPaC compatible con ONOS v2.1.0.

Prueba de instalación


Tras la ejecución del comando de instalación, la aplicación de DynPaC aparece activa (“ACTIVE”) dentro de las aplicaciones instaladas en ONOS, tal y como se muestra, a continuación, en la Figura 7.22.



Applications (173 Total)

TITLE	APP ID	VERSION
✓ Default Drivers	org.onosproject.drivers	1.15.0
✓ HP Drivers	org.onosproject.drivers.hp	1.15.0
✓ Host Location Provider	org.onosproject.hostprovider	1.15.0
✓ LLDP Link Provider	org.onosproject.lldpprovider	1.15.0
✓ OpenFlow Base Provider	org.onosproject.openflow-base	1.15.0
✓ OpenFlow Provider Suite	org.onosproject.openflow	1.15.0
✓ Optical Network Model	org.onosproject.optical-model	1.15.0
✓ org.geant.dynpac	org.geant.dynpac	1.6.30
Access Control Lists	org.onosproject.acl	1.15.0
Arista Drivers	org.onosproject.drivers.arista	1.15.0
Artemis	org.onosproject.artemis	1.15.0
BGP Provider	org.onosproject.bgp	1.15.0
BGP Router	org.onosproject.bgprouter	1.15.0
BGPCEP Provider	org.onosproject.bgpcep	1.15.0
BMv2 Drivers	org.onosproject.drivers.bmv2	1.15.0
Barefoot Drivers	org.onosproject.drivers.barefoot	1.15.0
Basic Optical Drivers	org.onosproject.drivers.optical	1.15.0
Basic Pipelines	org.onosproject.pipelines.basic	1.15.0
CORD Support	org.onosproject.cord-support	1.15.0
Factor	org.onosproject.factor	1.15.0

org.geant.dynpac



App ID: org.geant.dynpac
State: ACTIVE
Category: default
Version: 1.6.30
Origin: GEANT
Role: UNSPECIFIED
<http://onosproject.org>

GN DynPac migration test

FEATURES

dynpac

REQUIRED APPS

PERMISSIONS

Figura 7.27: Aplicación DynPaC instalada y activa en ONOS v2.1.0.

Prueba de alta de un servicio

De forma análoga a las pruebas realizadas en la Fase I y la Fase II, desde la GUI de DynPaC, se ha realizado una prueba de alta de un servicio entre el *Host 1* y el *Host 2* del escenario de red desplegado, introduciendo la información de configuración pertinente en la pestaña “*Service Creation*”, tal y como se puede observar, a continuación, en la Figura 7.28.

The screenshot shows the 'Service Creation' tab in the ONOS interface. The configuration fields are as follows:

- Name: PruebaTFM_Fase3
- Source STP: of:01f31c98ec99be80, 13, 500
- Destination STP: of:01f3eceb81b1f80, 24, 500
- Country: ES-Spain
- Time zone: Europe/Madrid
- Bandwidth: 100

Additional options include:

- Resilience:** Protected Service
- Schedule:** Start Date: 02/06/2023, Start Time: 20:27, End Date: 02/06/2023, End Time: 21:27. Navigation buttons for +/- 1 Hour, Day, Week, Month, and Current Date are present.

Developed by the GEANT project

Figura 7.28: Alta de un servicio DynPaC en ONOS v2.1.0.

Una vez dado de alta el servicio, se ha comprobado que éste se encuentra activo, desde la pestaña “List Services”, la cual se muestra, a continuación, en la Figura 7.29.

The screenshot shows the 'List Services' tab in the ONOS interface. It displays a table with one active service:

ID	COOKIE	NAME	START	END	INGRESS	EGRESS	BW	STATE	ACTIONS
0:	1311673391471656960	PruebaTFM_Fase3	02-06-2023 18:27:00	02-06-2023 19:27:00	of:01f31c98ec99be80 Port: 13 / VLAN: 500	of:01f3eceb81b1f80 Port: 24 / VLAN: 500	100	ACTIVE	Delete, Release, Topo, Info

Figura 7.29: Servicio DynPaC activo en ONOS v2.1.0.

Finalmente, atendiendo a la Figura 7.30, adjunta a continuación, se comprueba que los flujos instalados son, efectivamente, los adecuados: en los puertos 10 y 13 del Switch 1, y en los puertos 24 y 27 del Switch 2, todos ellos con el VLAN ID 500.

Bandwidth on Demand

DynPaC Links

NO.	SOURCE NODE	SOURCE PORT	DESTINATION NODE	DESTINATION PORT	LINK TYPE	LINK STATE
1	of:01f31c98ec99be80	10	of:01f31c98ec99be80	27	INDIRECT	ACTIVE

Active Network Resources

NO.	RESOURCE
1	1311673391471656960#of:01f31c98ec99be80->of:01f31c98ec99be80#@of:01f31c98ec99be80#p10:13#v499:500 Node = of:01f31c98ec99be80 Selector = Port 10 / VLAN 499 Treatment = Port 13 / VLAN 500
2	1311673391471656960#of:01f31c98ec99be80->of:01f31c98ec99be80#@of:01f31c98ec99be80#p24:27#v500:499 Node = of:01f31c98ec99be80 Selector = Port 24 / VLAN 500 Treatment = Port 27 / VLAN 499
3	1311673391471656960#of:01f31c98ec99be80->of:01f31c98ec99be80#@of:01f31c98ec99be80#p13:10#v500:499 Node = of:01f31c98ec99be80 Selector = Port 13 / VLAN 500 Treatment = Port 10 / VLAN 499
4	1311673391471656960#of:01f31c98ec99be80->of:01f31c98ec99be80#@of:01f31c98ec99be80#p27:24#v499:500 Node = of:01f31c98ec99be80 Selector = Port 27 / VLAN 499 Treatment = Port 24 / VLAN 500

Figura 7.30: Flujos instalados por el servicio DynPaC.

Conclusiones derivadas de las pruebas realizadas en la Fase III

Las pruebas realizadas sobre la versión de DynPaC resultante del diseño e implementación de la fase III de la migración objeto de este proyecto muestran que se ha logrado la compatibilidad de DynPaC con ONOS v2.1.0.

En consecuencia, de acuerdo con el Diseño de la solución expuesto en este documento, se puede afirmar que se han logrado incorporar a la red SN4I las herramientas de virtualización y servicios en la nube deseados (Apache Karaf, Kubernetes, OpenStack...), al contar con una plataforma que soporta las últimas versiones de estas herramientas, gracias a la actualización de DynPaC a una versión compatible con ONOS v2.1.0.

Cabe mencionar también que, mediante esta actualización, no sólo se ha conseguido la integración de DynPaC con SN4I, sino que también se consigue mejorar la programabilidad en el plano de datos, gracias a que ONOS v2.1.0 incluye nuevas mejoras para Stratum OS relacionadas con el soporte de P4Runtime y switches programables como Stratum-BMv2 y Stratum-Tofino, los cuales se utilizan en la red SN4I.

Sin olvidar también que esta fase de migración proporciona mejoras en la interfaz gráfica de usuario de ONOS, a través de la cual se gestiona la aplicación DynPaC, aportando mejoras en la GUI del plano de control del escenario de red SN4I.

7.6.4. Conclusiones de los resultados

Las pruebas realizadas para verificar la validez de la solución propuesta en este proyecto han obtenido resultados exitosos en todos y cada uno de sus apartados.

Por lo tanto, es posible afirmar que se ha comprobado, en un escenario de red con equipamiento real, que las versiones de la aplicación DynPaC resultantes de cada fase de la migración incremental diseñada e implementada en este TFM son plenamente compatibles con las versiones de ONOS correspondientes (1.15.0, 2.0.0 y 2.1.0).

En consecuencia, dado el Diseño de la solución desarrollado en este proyecto, se afirma que la versión final de DynPaC, resultante de la migración realizada en la Fase III de este proyecto, se ha integrado satisfactoriamente con el escenario SN4I y se ha dotado al mismo de la capacidad de incorporar dispositivos DPP y herramientas de monitorización de red basadas en telemetría.

Asimismo, también gracias al Diseño de la solución desarrollado en este TFM, en el que se ha partido de la versión original de DynPaC, cuya arquitectura era modular, la versión final de la aplicación ha mantenido la modularidad que la versión obsoleta originalmente ofrecía.

8. METODOLOGÍA

8.1. Equipo de proyecto

A continuación, en la Tabla 8.1, se muestra el equipo de trabajo encargado tanto del diseño e implementación de la solución propuesta en este proyecto, como de la redacción de la documentación pertinente.

Tabla 8.1: Equipo de trabajo del proyecto.

Nivel	Apellido 1 Apellido 2, Nombre	Responsabilidad
L1	Higuero Aperribai, María Victoria	Directora de proyecto
L1	Sasiain García, Jorge	Director técnico
L2	Plaza Vas, Isabel	Proyectista

El nivel se clasifica en L1 y L2 y dicha clasificación indica, en cada caso, lo siguiente:

- **Nivel de conocimiento L1:** ingeniera e ingeniero senior con un alto nivel de conocimiento en el campo de las Telecomunicaciones y, en especial, SDN. Su labor principal consiste en guiar a la proyectista, estableciendo las pautas y tareas esenciales que deben llevarse a cabo para lograr un desempeño satisfactorio del proyecto. Por otro lado, su experiencia ha servido de ayuda para resolver determinadas problemáticas surgidas durante el diseño e implementación de la solución.

La directora de proyecto determina el plan de trabajo, así como los objetivos a alcanzar dentro del mismo. Es la encargada tanto de supervisar las tareas de la proyectista, como de corregir y calificar la documentación del proyecto.

El director técnico guía a la proyectista en el diseño e implementación llevados a cabo durante el proyecto.

- **Nivel de conocimiento L2:** ingeniera junior en Telecomunicaciones. Dispone de los conocimientos de SDN y programación adecuados para el diseño e implementación de la solución presentada en este documento. Su labor es desarrollar las tareas que conforman el proyecto, así como seguir las indicaciones provistas por la directora para obtener resultados acordes a los objetivos establecidos. La proyectista desarrollará las tareas que la directora de proyecto le encomiende y redactará los informes necesarios para, finalmente, elaborar el documento final de Trabajo Fin de Máster.

8.2. Descripción de tareas y paquetes de trabajo

En este apartado se determinan las diferentes tareas y paquetes de trabajo en los que ha sido dividido el diseño e implementación del proceso de actualización de DynPaC propuesto en este proyecto. Para facilitar la jerarquización del trabajo dentro de diferentes fases y tareas, se va a utilizar una herramienta de gestión de proyectos denominada WBS. Este organigrama permite organizar de manera visual las diferentes actividades que conforman el desarrollo del proyecto, tal y como se puede observar, a continuación, en la Figura 8.1.

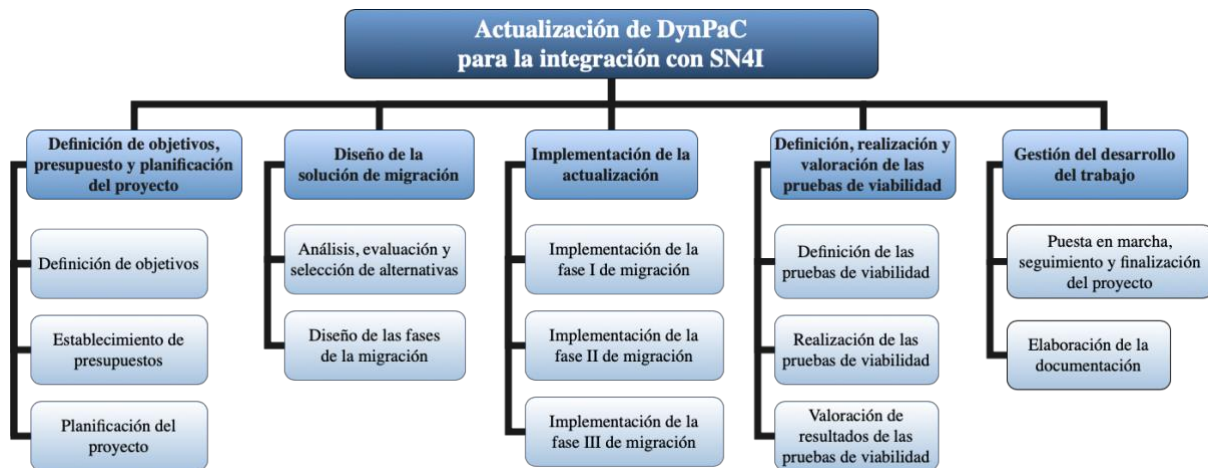


Figura 8.1: Paquetes de trabajo del proyecto (WBS).

En relación con la Figura anterior, se incluye, a continuación, en la Tabla 8.2, Tabla 8.3, Tabla 8.4, Tabla 8.5 y Tabla 8.6, una descripción detallada de los paquetes de trabajo y las tareas a realizar en cada uno de ellos, así como su duración, los recursos y la carga de trabajo necesarios para ello.

Tabla 8.2: Paquete de tareas P.T.1. del proyecto.

P.T.1.	<p>Definición de objetivos, presupuesto y planificación del proyecto. Este paquete de trabajo incluye las tareas previas al desarrollo del proyecto como tal. En ellas, se establecen los objetivos a alcanzar durante el desarrollo del proyecto, el presupuesto para alcanzar los objetivos en el plazo adecuado y la planificación necesaria para la realización del trabajo. (60 h)</p>
T.1.1.	<p>Definición de objetivos. Esta tarea consiste en definir los diferentes objetivos que se quieren lograr con el desarrollo del proyecto, así como las características necesarias que va a requerir la solución de propuesta para alcanzarlos. Recursos humanos: Directora de proyecto y proyectista. Recursos técnicos: herramientas ofimáticas (Microsoft Office, PowerPoint). Duración: 30 días. Carga de trabajo: 20 h.</p>
T.1.2.	<p>Establecimiento de presupuestos. Esta tarea es ejecutada al comienzo de la duración del proyecto para conocer el coste que va a suponer su realización y desarrollo. Por otro lado, también resulta de utilidad para el establecimiento de medidas correctoras en caso de ser requerido (imprevistos, retrasos...).</p> <p>Recursos humanos: Directora de proyecto y proyectista. Recursos técnicos: herramientas ofimáticas (Microsoft Office, Excel...).</p> <p>Duración: 10 días. Carga de trabajo: 20 h.</p>
T.1.3.	<p>Planificación del proyecto. En esta tarea se plantean las pautas de trabajo y la organización necesarias para llevar a cabo dicho desarrollo. Se desglosa todo el desarrollo del proyecto en diferentes paquetes de tareas y se establecen los plazos disponibles para cada uno.</p> <p>Recursos humanos: Proyectista. Recursos técnicos: herramientas ofimáticas (Microsoft Office, PowerPoint, Project Libre...).</p> <p>Duración: 20 días. Carga de trabajo: 20 h.</p>
H.1.	<p>Documento de objetivos, presupuesto y planificación del proyecto.</p>

Tabla 8.3: Paquete de tareas P.T.2. del proyecto.

<p>P.T.2.</p>	<p>Diseño de la solución de migración. Este paquete de trabajo constituye una de las fases de mayor relevancia en el proyecto, puesto que es donde se detalla el conjunto de metodologías, técnicas de actualización y tecnologías de comunicación a integrar en la solución propuesta. (150 h)</p>
<p>T.2.1.</p>	<p>Análisis, valoración y selección de las alternativas. Una parte importante del proyecto consiste en analizar las diferentes alternativas que pueden utilizarse a la hora de realizar el diseño e implementación, valorar las características de cada una de ellas, y seleccionar finalmente aquella que mejor se adapte a las necesidades de cada escenario. Esta parte resulta fundamental para elaborar el análisis de alternativas y el posterior diseño de la solución.</p> <p>Recursos humanos: Proyectista. Recursos técnicos: Internet, artículos de investigación. Duración: 30 días. Carga de trabajo: 30 h.</p>
<p>T.2.2.</p>	<p>Diseño de las fases de la migración. Esta tarea se consiste en la definición de las fases incrementales necesarias para configurar el diseño de la solución, así como el diseño de las mismos.</p> <p>Recursos humanos: Director técnico y proyectista. Recursos técnicos: Internet, artículos de investigación. Duración: 105 días. Carga de trabajo: 120 h.</p>
<p>H.2.</p>	<p>Descripción del diseño de la solución.</p>

Tabla 8.4: Paquete de tareas P.T.3. del proyecto.

P.T.3.	Implementación de la solución de migración. Este paquete de trabajo incluye las tareas necesarias para llevar a cabo la implementación de la actualización de DynPaC para la integración con SN4I y la incorporación de DPP y herramientas de monitorización basadas en telemetría. (250 h)
T.3.1.	Implementación de la Fase I de la migración. En esta tarea se lleva a cabo la implementación de la primera fase de la migración, diseñada en la T.2.2. Recursos humanos: Proyectista. Recursos técnicos: Ordenador personal. Duración: 30 días. Carga de trabajo: 100 h.
T.3.2.	Implementación de la Fase II de la migración. En esta tarea se lleva a cabo la implementación de la segunda fase de la migración, diseñada en la T.2.2. Recursos humanos: Proyectista. Recursos técnicos: Ordenador personal. Duración: 30 días. Carga de trabajo: 75 h.
T.3.3.	Implementación de la Fase III de la migración. En esta tarea se lleva a cabo la implementación de la tercera fase de la migración, diseñada en la T.2.2. Recursos humanos: Proyectista. Recursos técnicos: Ordenador personal. Duración: 30 días. Carga de trabajo: 75 h.
H.3.	Software actualizado de la aplicación DynPaC.

Tabla 8.5: Paquete de tareas P.T.4. del proyecto.

<p>P.T.4.</p>	<p>Definición, realización y valoración de resultados de las pruebas de viabilidad. Este paquete de trabajo incluye las tareas necesarias para la definición y el desarrollo de la fase de pruebas de viabilidad de la solución propuesta en este proyecto. (80 h)</p>
<p>T.4.1.</p>	<p>Definición de las pruebas de viabilidad. En esta tarea se diseñan las pruebas a realizar para probar la viabilidad de la solución propuesta en este proyecto. Recursos humanos: proyectista. Recursos técnicos: equipamiento de red (<i>switches, hosts, servidores, cableado</i>). Duración: 10 días. Carga de trabajo: 20 h.</p>
<p>T.4.2.</p>	<p>Realización de las pruebas de viabilidad. En esta tarea se ejecutan las pruebas definidas en la T.4.1. Recursos humanos: Proyectista. Recursos técnicos: Equipamiento de red (<i>switches, hosts, servidores, cableado</i>). Duración: 10 días. Carga de trabajo: 50 h.</p>
<p>T.4.3.</p>	<p>Valoración de los resultados de las pruebas de viabilidad. Esta tarea se evalúan los resultados obtenidos en las pruebas ejecutadas en la T.4.2. y se valora si concluyen la viabilidad de la solución propuesta en este proyecto. Recursos humanos: Proyectista. Recursos técnicos: Equipamiento de red (<i>switches, hosts, servidores, cableado</i>). Duración: 10 días. Carga de trabajo: 10 h.</p>
<p>H.4.</p>	<p>Plan de pruebas, resultados y conclusiones.</p>

Tabla 8.6: Paquete de tareas P.T.5. del proyecto.

P.T.5.	Gestión del desarrollo del proyecto. Este paquete de trabajo incluye una serie de tareas que se centran en la organización y gestión del trabajo a realizar dentro del mismo. (200 h)
T.5.1.	<p>Puesta en marcha, seguimiento y finalización del proyecto. Esta tarea consiste en definir los diferentes objetivos que se quieren lograr con el desarrollo del proyecto, así como las características necesarias que va a requerir la solución de propuesta para alcanzarlos. También se plantean las pautas de trabajo y la organización necesaria para llevar a cabo dicho desarrollo.</p> <p>Recursos humanos: Directora de proyecto y proyectista.</p> <p>Recursos técnicos: herramientas ofimáticas (Microsoft Office, PowerPoint, Excel, Project Libre...).</p> <p>Duración: 15 días. Carga de trabajo: 50 h.</p>
T.5.2.	<p>Elaboración de la documentación. En esta tarea se realiza el documento que recoge toda la información del proyecto.</p> <p>Recursos humanos: proyectista.</p> <p>Recursos técnicos: herramientas ofimáticas (Microsoft Office, PowerPoint, Excel, Project Libre...).</p> <p>Duración: 255 días. Carga de trabajo: 150 h.</p>
H.5.	Memoria y entrega final del proyecto

8.3. Diagrama de Gantt

En este apartado, se incluye un cronograma donde se muestran de manera más intuitiva las diferentes tareas que conforman el proyecto, así como el tiempo dedicado a las mismas. Esto permite organizar de manera más visual la planificación del proyecto.

El proyecto finaliza el día 6 de junio de 2023 con la entrega de la memoria final en la plataforma ADDI. En la Tabla 8.7, mostrada a continuación, se pueden observar las fechas inicio y fin de proyecto, así como la duración y la carga de trabajo del mismo.

Tabla 8.7: Parámetros generales de la planificación del proyecto.

Fecha de inicio	Fecha de finalización	Duración	Carga de trabajo
1 de septiembre de 2022	6 de junio de 2023	260 días	740 horas

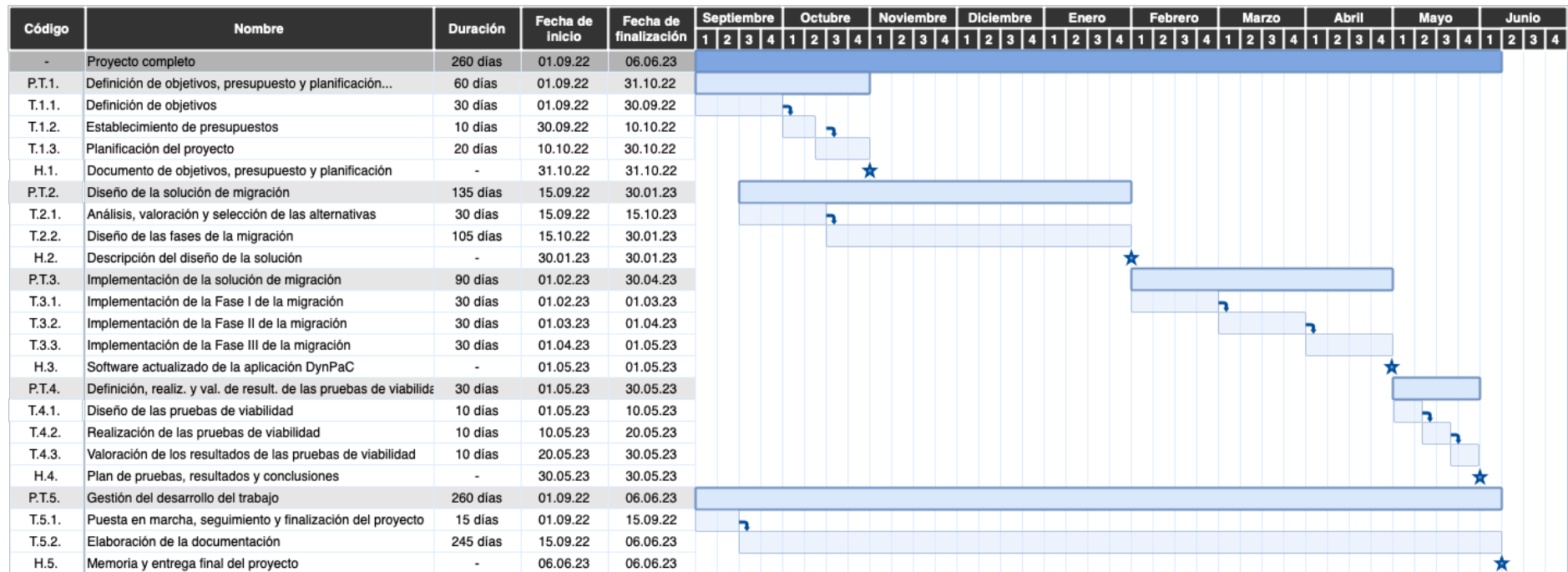
En lo referente a los hitos del proyecto, se han definido los mostrados, a continuación, en la Tabla 8.8.

Tabla 8.8: Hitos del proyecto.

Código	Descripción
H.1.	Documento de objetivos, presupuesto y planificación
H.2.	Descripción del diseño de la solución
H.3.	<i>Software</i> actualizado de la aplicación DynPaC
H.4.	Plan de pruebas, resultados y conclusiones
H.5.	Memoria y entrega final del proyecto

A continuación, en la siguiente página, se muestra el cronograma antes mencionado: el diagrama de Gantt de la planificación del proyecto presentado en este documento.

Figura 8.2: Diagrama de Gantt del proyecto.



9. ANÁLISIS DE COSTES

El desarrollo de este proyecto ha supuesto una serie de costes, los cuales se ven reflejados, a continuación, en este apartado. En primer lugar, se desglosan los costes de cada partida. Posteriormente, se expone el coste total del proyecto desglosado por dichas partidas.

9.1. Horas internas

Esta partida se refiere al coste de los recursos humanos pertenecientes al equipo de proyecto (véase Tabla 8.1) necesarios durante el desarrollo del proyecto. El desglose de costes de dicha partida se muestra, a continuación, en la Tabla 9.1.

Tabla 9.1: Partida de horas internas del proyecto.

Concepto	Número de horas [h]	Coste unitario [€/h]	Coste total [€]
Ingeniera junior	600	10	6.000
Ingeniera/o senior	200	40	8.000
TOTAL			14.000

9.2. Amortizaciones

Esta partida se refiere al coste amortizado de los recursos materiales empleados durante el desarrollo del proyecto. El coste unitario de cada concepto, así como el desglose de costes de la partida de amortizaciones se muestra, a continuación, en la Tabla 9.2 y en la Tabla 9.2 respectivamente.

Tabla 9.2: Coste unitario de las amortizaciones del proyecto.

Concepto	Coste [€]	Vida útil [años*]	Coste unitario [€/h]
Ordenador personal	1500	4	0,21
Switch HP Aruba (x2 unidades)	5.085 (x2)	6	0,96
Servidor	3.000	6	0,28
Cableado	100	5	0,01

* NOTA: Para el cálculo del coste unitario (coste / vida útil), se ha considerado que un año de vida útil equivale a 1760 horas laborables.

Tabla 9.3: Partida de amortizaciones del proyecto.

Concepto	Número de horas [h]	Coste unitario [€/h]	Coste total [€]
Ordenador personal	600	0,21	126,00
Switches HP Aruba	50	0,96	48,00
Servidor	50	0,28	14,00
Cableado	50	0,01	0,50
TOTAL			188,50

9.3. Gastos

Esta partida se refiere al coste de los gastos no incluidos en otras partidas que ha supuesto el desarrollo del proyecto. El desglose de costes de dicha partida se muestra, a continuación, en la Tabla 9.4.

Tabla 9.4: Partida de gastos del proyecto.

Concepto	Número de horas [h]	Coste unitario [€/h]	Coste total [€]
Facturas Internet	-	-	450
Facturas electricidad	-	-	530
TOTAL			980

9.4. Subcontrataciones

Esta partida se refiere al coste de los trabajos relativos al proyecto realizados por personal externo al equipo de proyecto (véase Tabla 8.1). Como puede observarse, a continuación, en la Tabla 9.5, en este proyecto no se ha subcontratado ningún trabajo y, por lo tanto, el coste derivado de las subcontrataciones es nulo.

Tabla 9.5: Partida de subcontrataciones del proyecto.

Concepto	Número de horas [h]	Coste unitario [€/h]	Coste total [€]
-	-	-	-
TOTAL			0

9.5. Coste total del proyecto

Visto el desglose de los costes de cada partida del proyecto, se muestra, a continuación, en la Tabla 9.6, el resumen de todos los costes del proyecto, desglosado por partidas.

A la hora de calcular el coste completo, se ha añadido una partida de costes indirectos (10 % del coste directo), con la intención de considerar gastos no atribuibles a un solo proyecto, así como los gastos de problemas que han surgido a lo largo del proyecto y que han penalizado el seguimiento riguroso de los plazos establecidos en la planificación del proyecto.

Tabla 9.6: Resumen de los costes del proyecto.

Partida	Coste total [€]
Horas internas	14.000,00
Amortizaciones	188,50
Gastos	980,00
Subcontrataciones	0,00
SUBTOTAL	15.168,50
Costes indirectos (10%)	1.516,85
TOTAL	16.685,35

El desarrollo de este proyecto supone un coste total que asciende a la cifra de DIECISÉIS MIL SEISCIENTOS OCHENTA Y CINCO EUROS CON TREINTA Y CINCO CÉNTIMOS (IVA incluido).

Tal y como se puede apreciar en el desglose previo, la mayoría de los recursos económicos son utilizados para hacer frente a los gastos en recursos humanos (horas internas). El diseño e implementación de la solución propuesta en este proyecto requiere un número elevado de horas de trabajo, lo cual incrementa de manera notable los costes de personal. El resto de los costes presentados se corresponden con elementos básicos presentes en la mayoría de los proyectos de ingeniería.

10. CONCLUSIONES

El desarrollo de este proyecto ha permitido mantener las funcionalidades, servicios y prestaciones de la aplicación DynPaC y a su vez incorporar las nuevas funcionalidades requeridas en el nuevo escenario de operación SN4I.

Por un lado, la integración de DynPaC como capa de aplicación en la arquitectura de red de SN4I ha permitido el aprovisionamiento de servicios de nivel de enlace, facilitando estrategias de TE y proporcionando cálculo y selección de la mejor ruta extremo a extremo para cada servicio, así como la capacidad de aislar servicios de comunicaciones para brindar seguridad a distintos usos de la infraestructura de red.

Por otro lado, se ha conseguido la incorporación de herramientas de telemetría al escenario SN4I, las cuales le aportan la capacidad de obtener datos sobre el estado de la red en tiempo real, lo cual es esencial para la monitorización y gestión eficiente de la red en escenarios SDN.

Asimismo, se ha logrado el objetivo de incorporar programabilidad en el plano de datos (DPP) a la infraestructura de red. El escenario SN4I ha sido dotado de la capacidad de soportar planos de datos programables, lo cual es fundamental en la investigación de tecnologías de red innovadoras y permite una mayor flexibilidad y adaptabilidad a diferentes escenarios a la vez que soporta la prestación de servicios de red innovadores.

De manera adicional, los objetivos secundarios de modularidad y flexibilidad se han cumplido satisfactoriamente. La solución propuesta se ha diseñado de forma modular, lo que facilita su estructuración, implementación y posibilidad de añadir, modificar o eliminar funcionalidades según las necesidades de cada escenario. Además, se ha asegurado que la solución sea flexible y capaz de adaptarse a los distintos posibles escenarios de la red SN4I, proporcionando un entorno de investigación y experimentación ideal para pruebas relacionadas con tecnologías de red innovadoras.

Finalmente, se ha desarrollado una plataforma de desarrollo para analizar la viabilidad de la propuesta. Esta plataforma ha permitido realizar diversas pruebas que han servido para verificar la viabilidad de la solución desarrollada y garantizar su eficacia en el entorno de SN4I.

En conclusión, se han conseguido alcanzar los objetivos definidos al principio del proyecto en los plazos y costes previstos.

Desde el punto de vista técnico, el proyecto ha logrado diseñar, desarrollar e implementar una solución que cumple con los objetivos propuestos. Se ha demostrado la viabilidad de la propuesta mediante pruebas realizadas sobre equipamiento de red real y, en consecuencia, es posible afirmar que se ha creado un entorno de investigación óptimo para la exploración de tecnologías de red innovadoras.

BIBLIOGRAFÍA

- [1] Open Networking Foundation, «ONOS: Open Network Operating System,» [En línea]. Available: <https://opennetworking.org/onos/>. [Último acceso: 30 marzo 2023].
- [2] I2T Research Group, «Smart Networks for Industry - SN4I,» [En línea]. Available: <https://i2t.ehu.eus/es/resources/6/sn4i>. [Último acceso: 27 marzo 2023].
- [3] ONOS Project, «ONOS Downloads,» 21 marzo 2022. [En línea]. Available: <https://wiki.onosproject.org/display/ONOS/Downloads>. [Último acceso: 30 marzo 2023].
- [4] Open Networking Lab, «ONOS Release Notes,» Enero 2020. [En línea]. Available: <https://github.com/opennetworkinglab/onos/blob/master/RELEASES.md#22-lts---september-2019>. [Último acceso: Marzo 2023].
- [5] IETF, «In-band Network Telemetry,» 28 abril 2020. [En línea]. Available: <https://datatracker.ietf.org/doc/html/draft-tian-bupt-inwt-mechanism-policy-00>. [Último acceso: 30 marzo 2023].
- [6] P4, «P4 Open Source Programming Language,» [En línea]. Available: <https://p4.org>. [Último acceso: 24 mayo 2023].
- [7] ONF, «Stratum,» [En línea]. Available: <https://opennetworking.org/stratum/>. [Último acceso: 24 mayo 2023].
- [8] Maven, «ONOS Core Common usages,» [En línea]. Available: <https://mvnrepository.com/artifact/org.onosproject/onos-core-common/usages>. [Último acceso: 2023 mayo 28].
- [9] Maven, «ONOS ONlab Misc usages,» [En línea]. Available: <https://mvnrepository.com/artifact/org.onosproject/onlab-misc/usages>. [Último acceso: 2023 mayo 28].