

Módulo 1

Introducción

© 2024

Gabriel Guerrero-Contreras

`gabriel.guerrero@uca.es`

Sara Balderas-Díaz

`sara.balderas@uca.es`

Universidad de Cádiz

Escuela Superior de Ingeniería

Departamento de Ingeniería Informática

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike
4.0 International License.

To view a copy of this license, visit:

<http://creativecommons.org/licenses/by-nc-sa/4.0/legalcode>



Índice

1. Introducción	1
1.1. ¿Qué es ns-3?	1
1.2. Historia y evolución de ns-3	2
1.3. Características principales	3
1.4. Algunas desventajas	4
1.5. Instalación y configuración del entorno de desarrollo	4
1.5.1. Docker	4
1.5.2. Docker y ns-3	5
1.5.3. Ejecución de la imagen de Docker y ns-3	7

1. Introducción

En el ámbito de las telecomunicaciones y las tecnologías de la información, la simulación de redes desempeña un papel fundamental en la investigación, el diseño y la evaluación de sistemas de comunicación. En este contexto, ns-3 emerge como una herramienta esencial para investigadores y desarrolladores en la exploración y análisis de redes de comunicaciones. Desde sus inicios en la Universidad de Washington hasta su adopción generalizada en la comunidad académica, ns-3 ha evolucionado constantemente para convertirse en uno de los simuladores de redes más avanzados y respetados disponibles hoy en día. En este documento, exploraremos en detalle qué es ns-3, su historia y evolución, características principales, así como sus ventajas y desventajas, proporcionando una visión completa de esta herramienta de simulación de redes.

1.1. ¿Qué es ns-3?

ns-3, abreviatura de *Network Simulator 3*¹, es un simulador de redes de código abierto que se utiliza ampliamente en la investigación y el desarrollo en el campo de las comunicaciones de red. Se caracteriza por su **precisión** y **flexibilidad** en la simulación de redes, lo que lo convierte en una herramienta útil para diseñar, probar y analizar **protocolos** de red y **algoritmos** en un entorno de simulación controlado y reproducible.

ns-3 se desarrolla en C++ y está diseñado con una arquitectura orientada a objetos y basada en modelos. Esto permite representar con **precisión** los componentes de una red, como **nodos**, **enlaces** y **protocolos**, como objetos interconectados. La **modularidad** inherente de **ns-3** facilita la extensión con nuevos modelos de dispositivos y protocolos de red, lo que permite a los usuarios personalizar y adaptar el simulador según sus necesidades específicas de investigación o desarrollo.

Una de las características distintivas de **ns-3** es su capacidad para modelar la red a un nivel **granular**. Esto significa que los usuarios pueden controlar y modificar cada aspecto de la simulación, desde las características de los **enlaces** y los **nodos** hasta el comportamiento de los **protocolos** de red. Esto proporciona un alto grado de **flexibilidad** y **precisión** en la simulación, lo que lo hace ideal para estudiar el comportamiento de las redes en condiciones realistas.

ns-3 ofrece una amplia variedad de modelos de red predefinidos, que incluyen **protocolos** de capa de enlace, **enrutamiento**, **transporte** y **aplicación**, entre otros. Estos modelos se pueden modificar o extender fácilmente para adaptarse a requisitos específicos de investigación o desarrollo. Además, **ns-3** proporciona un entorno de scripting basado en Python para la configuración y ejecución de simulaciones, lo que facilita la experimentación y la automatización de tareas repetitivas.

En términos de **rendimiento**, **ns-3** se destaca por su capacidad para simular grandes redes con eficiencia. Esto se logra mediante técnicas de **paralelización** que permiten

¹<https://www.nsnam.org/>

distribuir la carga de la simulación en múltiples núcleos de CPU. Además, **ns-3** ofrece herramientas de visualización y análisis de datos que permiten a los usuarios entender y evaluar el comportamiento de la red simulada.

1.2. Historia y evolución de ns-3

ns-3 ha experimentado un largo viaje desde su concepción inicial hasta convertirse en uno de los simuladores de redes más utilizados y respetados en la comunidad académica y de investigación. Su historia está marcada por hitos significativos, colaboraciones clave y un crecimiento constante en funcionalidad.

El desarrollo de ns-3 se remonta a principios de la década de 2000 en la Universidad de Washington en Seattle, donde un equipo de investigadores liderado por **George Riley** y **John Heidemann** comenzó a explorar la idea de un nuevo simulador de redes que pudiera abordar las limitaciones de los simuladores existentes en ese momento. Inspirados por el éxito de ns-2² (su predecesor), pero también conscientes de sus desafíos y limitaciones, el equipo se propuso desarrollar un simulador más moderno, flexible y potente.

ns-3 vio la luz con el lanzamiento de su primera versión pública en 2008. Esta versión inicial estableció las bases para lo que se convertiría en un simulador de redes altamente sofisticado y configurable. Aunque inicialmente recibió menos atención que su predecesor ns-2, rápidamente comenzó a ganar seguidores gracias a su arquitectura más moderna y su enfoque en la modularidad y la extensibilidad.

A medida que ns-3 maduraba y ganaba funcionalidad, su popularidad creció rápidamente en la comunidad académica. La creciente biblioteca de modelos y herramientas disponibles para ns-3 facilitó su adopción en una variedad de áreas de investigación, incluidas las redes inalámbricas, las redes de sensores, las redes de área local y las redes móviles, entre otras.

A lo largo de los años, ns-3 se ha enriquecido de contribuciones provenientes de investigadores y desarrolladores de todo el mundo. Estas aportaciones han fortalecido significativamente el simulador, introduciendo nuevos modelos de dispositivos que abarcan desde routers y switches hasta dispositivos móviles y antenas especializadas. Por ejemplo, se han desarrollado modelos más precisos de antenas **MIMO (Multiple Input Multiple Output)** que permiten simular con mayor fidelidad tecnologías de comunicación inalámbrica avanzadas como el **5G**.

Además de los avances en modelos de dispositivos, ns-3 ha sido enriquecido con la incorporación de protocolos de red. Entre estas contribuciones se incluyen nuevos protocolos de enrutamiento diseñados para redes inalámbricas ad hoc, como el protocolo **OLSRv2 (Optimized Link State Routing Protocol version 2)**, que mejora la eficiencia en el descubrimiento de rutas en redes dinámicas.

Junto con las mejoras en modelos de dispositivos y protocolos de red, ns-3 ha ex-

²<https://www.isi.edu/nsnam/ns/>

perimentado un progreso significativo en el ámbito de las herramientas de análisis. Se han desarrollado herramientas avanzadas de visualización y diagnóstico que facilitan la interpretación de los resultados de las simulaciones. Por ejemplo, la integración de herramientas de análisis de tráfico en tiempo real ha permitido a los usuarios monitorizar y analizar el rendimiento de la red durante la ejecución de las simulaciones.

Además, ns-3 ha sido el epicentro de una variedad de colaboraciones interdisciplinarias entre instituciones académicas, empresas e industrias. Estas colaboraciones han dado lugar a avances significativos en áreas como la **computación en la nube**, el **Internet de las cosas (IoT)** y las **redes definidas por software (SDN)**, enriqueciendo así la versatilidad y aplicabilidad del simulador en una amplia gama de campos de investigación y desarrollo.

1.3. Características principales

- **Simulación de protocolos de red:** Capacidad para simular una amplia variedad de protocolos de red, que abarcan desde los niveles más bajos, como la capa física y la capa de enlace de datos, hasta los niveles más altos, como la capa de aplicación. ns-3 proporciona implementaciones precisas de numerosos protocolos estándar, como TCP/IP, UDP, HTTP, DNS, entre otros, lo que permite a los usuarios evaluar el rendimiento y el comportamiento de estos protocolos en diferentes escenarios de red.
- **Arquitectura modular y extensible:** Permite a los usuarios extender y personalizar el simulador según sus necesidades específicas. ns-3 está construido alrededor de un conjunto de módulos independientes que representan diferentes aspectos de una red, como nodos, enlaces, dispositivos y protocolos. Esto facilita la adición de nuevos modelos y funcionalidades al simulador sin afectar el núcleo existente, lo que permite a los usuarios adaptar ns-3 a una amplia variedad de aplicaciones y escenarios de red.
- **Compatibilidad multiplataforma:** ns-3 se ejecuta en múltiples plataformas y sistemas operativos, lo que lo hace accesible para una amplia audiencia de usuarios.
- **Soporte para tecnologías emergentes:** Desde la simulación de redes inalámbricas y móviles hasta las redes de sensores y el Internet de las cosas (IoT), ns-3 proporciona modelos y herramientas para abordar una amplia variedad de tecnologías y escenarios de red.
- **Soporte de la comunidad y documentación extensa³:** ns-3 cuenta con una comunidad activa de desarrolladores, investigadores y usuarios que contribuyen al desarrollo y mantenimiento del simulador. Esta comunidad proporciona soporte técnico, recursos educativos y herramientas adicionales para ayudar a los usuarios

³<https://www.nsnam.org/documentation/>

a aprovechar al máximo ns-3. Además, ns-3 cuenta con una documentación detallada y exhaustiva que cubre todos los aspectos del simulador, desde la instalación y configuración hasta el desarrollo de modelos y la ejecución de simulaciones, lo que facilita el aprendizaje y la utilización del simulador para una amplia variedad de aplicaciones.

1.4. Algunas desventajas

Como hemos visto, ns-3 presenta una serie de ventajas significativas que lo hacen destacar como una herramienta fundamental para la investigación y el desarrollo. Sin embargo, también hay desventajas importantes que deben considerarse al utilizar ns-3 en proyectos de desarrollo de software.

- **Curva de aprendizaje inicial:** Pronunciada, especialmente para aquellos que no están familiarizados con la simulación de redes o el desarrollo de software en C++. Los usuarios pueden necesitar tiempo para familiarizarse con los conceptos y la arquitectura de ns-3 antes de poder utilizarlo de manera efectiva.
- **Entorno de desarrollo complejo:** Configurar y compilar ns-3 desde el código fuente puede requerir la instalación de varias dependencias y la resolución de problemas de compatibilidad.
- **Requisitos de hardware:** ns-3 puede requerir recursos computacionales significativos para ejecutar simulaciones complejas, lo que puede limitar su uso. Se pueden necesitar grandes recursos hardware para ejecutar simulaciones a gran escala con tiempos de ejecución razonables.

1.5. Instalación y configuración del entorno de desarrollo

La instalación de **ns-3** puede variar dependiendo del sistema operativo utilizado. Se proporcionan instrucciones detalladas en la documentación oficial⁴ de **ns-3** para instalar las dependencias necesarias y configurar el entorno de desarrollo.

1.5.1. Docker

Docker⁵ es una herramienta de virtualización ligera que facilita el desarrollo, la ejecución y el despliegue de aplicaciones en entornos consistentes y aislados, conocidos como contenedores. El uso de Docker para ns-3 ofrece varias ventajas, incluida la facilidad de configuración del entorno de simulación, la portabilidad entre diferentes sistemas operativos y la capacidad de compartir configuraciones de entorno de manera reproducible con otros usuarios o colaboradores. Esta aproximación ayuda a minimizar

⁴<https://www.nsnam.org/wiki/Installation>

⁵<https://www.docker.com/>

las diferencias entre entornos de trabajo y reduce los problemas de "funciona en mi máquina", haciendo que el proceso de desarrollo y prueba sea más eficiente.

Al optar por Docker como solución para instalar y ejecutar ns-3, es importante tener en cuenta las implicaciones de rendimiento asociadas con la virtualización. Mientras que Docker ofrece una manera conveniente de configurar un entorno de desarrollo para ns-3, la capa de abstracción que añade puede, en algunos casos, influir en el rendimiento de las simulaciones, especialmente en aquellas que son intensivas en recursos computacionales. Para usuarios cuyos proyectos de simulación demandan el máximo rendimiento o aquellos que requieren una integración avanzada con herramientas de desarrollo, sistemas de control de versiones, o bibliotecas de software específicas, instalar ns-3 directamente sobre el sistema operativo del host podría ser más adecuado.

1.5.2. Docker y ns-3

Un **Dockerfile** es un archivo de texto plano que contiene una serie de instrucciones que **Docker** utiliza para construir automáticamente una imagen de contenedor. Estas instrucciones describen los pasos necesarios para configurar y preparar el entorno en el que se ejecutará la aplicación dentro del contenedor. Un **Dockerfile** suele incluir comandos para especificar la base de la imagen, copiar archivos y directorios, configurar variables de entorno, instalar dependencias y ejecutar comandos para configurar la aplicación.

A continuación se muestra un ejemplo de un **Dockerfile** que se puede utilizar para instalar y configurar el framework de simulación **ns-3**:

```
FROM ubuntu:20.04

ENV DEBIAN_FRONTEND=noninteractive

RUN apt-get update && apt-get install -y \
    g++ \
    python3 \
    python3-dev \
    python3-setuptools \
    git \
    mercurial \
    qt5-default \
    bison \
    flex \
    gir1.2-goocanvas-2.0 \
    python3-gi \
    python3-pygraphviz \
    python3-gi-cairo \
    python3-pybind11 \
    gnuplot \
    python3-scipy \
```

```

python3-numpy \
libc6-dev \
libxml2 \
libxml2-dev \
cmake \
libc6-dev \
libgtk2.0-0 \
libgtk2.0-dev \
vtun \
lxc \
wget \
&& apt-get clean && rm -rf /var/lib/apt/lists/*

WORKDIR /ns3

RUN wget https://www.nsnam.org/release/ns-allinone-3.35.tar.bz2
    ↪ \
    && tar -xjf ns-allinone-3.35.tar.bz2 \
    && rm ns-allinone-3.35.tar.bz2

RUN cd ns-allinone-3.35/ns-3.35 \
    && ./waf configure --enable-examples --enable-tests \
    && ./waf

ENV NS3_PATH=/ns3/ns-allinone-3.35/ns-3.35
ENV PATH=${NS3_PATH}:${PATH}

CMD ["/bin/bash"]

```

Veamos, línea por línea, en qué consiste este fichero, para comprender mejor cómo se configura y construye la imagen de contenedor para el framework de simulación ns-3.

Selecciona Ubuntu 20.04 como la imagen base. Ubuntu es una elección popular para este propósito debido a su amplia compatibilidad con las dependencias requeridas por ns-3:

```
FROM ubuntu:20.04
```

Configura la variable de entorno 'DEBIAN_FRONTEND' para evitar cualquier pregunta interactiva durante la instalación de paquetes⁶:

```
ARG DEBIAN_FRONTEND=noninteractive
```

Instala las dependencias necesarias para ns-3. La lista incluye herramientas de desarrollo como compiladores y bibliotecas específicas que ns-3 utiliza para la simulación.

⁶Aunque esto es útil para automatizar la instalación sin supervisión, es importante usar este enfoque con precaución. La ausencia de interacción puede ocultar advertencias o errores que normalmente requerirían la intervención del usuario durante la instalación de paquetes.


```
RUN apt-get update && apt-get install -y \  
    g++ \  
    python3 \  
    ... \  
    wget \  
    && apt-get clean && rm -rf /var/lib/apt/lists/*
```

Establece el directorio de trabajo dentro del contenedor Docker. Esto significa que cualquier comando que se ejecute posteriormente se hará desde este directorio.

```
WORKDIR /ns3
```

Descarga y compila ns-3.35. Esto incluye clonar el repositorio de ns-3, y luego utilizar scripts específicos del proyecto para compilarlo.

```
RUN wget https://www.nsnam.org/release/ns-allinone-3.35.tar.bz2  
    ↪ \  
    && tar -xjf ns-allinone-3.35.tar.bz2 \  
    && rm ns-allinone-3.35.tar.bz2  
  
RUN cd ns-allinone-3.35/ns-3.35 \  
    && ./waf configure --enable-examples --enable-tests \  
    && ./waf
```

Configura las variables de entorno.

```
ENV NS3_PATH=/ns3/ns-allinone-3.35/ns-3.35  
ENV PATH=${NS3_PATH}:${PATH}
```

Define el comando por defecto que se ejecutará cuando el contenedor inicie. En este caso, inicia una shell Bash.

```
CMD ["/bin/bash"]
```

1.5.3. Ejecución de la imagen de Docker y ns-3

Para ejecutar la imagen de Docker creada a partir del Dockerfile proporcionado, se pueden seguir los siguientes pasos

1. Abre una terminal en tu sistema operativo.
2. Asegúrate de tener Docker instalado en tu máquina⁷. Puedes verificarlo ejecutando el siguiente comando en la terminal:

```
docker --version
```

⁷<https://www.docker.com/products/docker-desktop/>

3. Si Docker está instalado, procede a construir la imagen de Docker desde el Dockerfile. Para hacerlo, navega hasta el directorio que contiene el Dockerfile en tu terminal y ejecuta el siguiente comando:

```
docker build -t ns3-container .
```

4. Una vez que la imagen de Docker se haya construido correctamente, puedes ejecutar un contenedor basado en esa imagen. Para ello, utiliza el siguiente comando:

```
docker run -it ns3-container
```

5. Esto iniciará un nuevo contenedor basado en la imagen `ns3-container` y abrirá una sesión de shell dentro del contenedor. Ahora estás dentro del contenedor y puedes ejecutar comandos de ns-3 como lo harías en un entorno de desarrollo normal.
6. Una vez dentro del contenedor, puedes verificar que ns-3 se ha instalado correctamente y cuál es la versión actualmente instalada. Para esto, navega al directorio de ns-3 (`/ns3/ns-allinone-3.35/ns-3.35/`) y ejecuta el script de versión:

```
cd ns-allinone-3.35/  
cd ns-3.35/  
./waf --version
```

7. Para salir del contenedor y volver al terminal del sistema operativo host, simplemente ejecuta el comando:

```
exit
```