

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Despliegue de una plataforma de demostración del encaminamiento
basado en contenidos

PROYECTO INTEGRADOR

Previo la obtención del Título de:

Ingeniero en Telemática

Presentado por:

Kevin Eduardo Alcívar Peña

Ernesto Celín Campoverde Rodríguez

GUAYAQUIL - ECUADOR

Año: 2021

DEDICATORIA

Dedico el presente proyecto a todas las personas que depositaron su confianza en mí y me brindaron ayuda de alguna manera, en especial a mis padres por ser el pilar principal.

Kevin Alcívar

Dedico el presente proyecto a mi familia, en especial a mis padres, por su entrega, espíritu perseverante y apoyo incondicional durante mi carrera universitaria. Su ejemplo me llena de motivación.

Ernesto Campoverde

AGRADECIMIENTOS

No existen las suficientes palabras para poder agradecer todo el apoyo y ayuda que recibo de mis padres, gracias por siempre estar ahí.

A Ernesto por su completa disposición a la hora de trabajar.

A la Ing. Jordán por guiarnos y ofrecernos sus consejos y experiencia para poder culminar de la mejor manera nuestra carrera.

Kevin Alcívar

Quiero expresar mi profundo agradecimiento a mis familiares, por siempre haberme brindado una mano cuando la necesitaba.

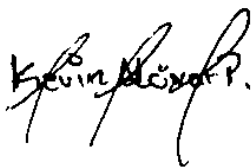
A mi compañero Kevin, quien me ayudó con buena voluntad en todo momento.

A la Ing. Jordán, por habernos guiado durante el desarrollo del proyecto, siempre ofreciéndonos consejos y pautas para alcanzar la meta final.

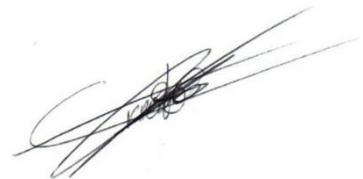
Ernesto Campoverde

DECLARACIÓN EXPRESA

"Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; *Kevin Eduardo Alcívar Peña* y *Ernesto Celín Campoverde Rodríguez* damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"



Kevin Eduardo
Alcívar Peña



Ernesto Celín
Campoverde Rodríguez

EVALUADORES

.....
José Eduardo Córdova García

PROFESOR DE LA MATERIA

.....
Paola Carolina Jordán Figueroa

PROFESOR TUTOR

RESUMEN

Actualmente, usuarios alrededor del mundo navegan en las redes a un ritmo vertiginoso, creando la necesidad de realizar grandes inversiones en réplicas de centros de datos en ubicaciones geográficas estratégicas para ofrecer un tiempo de respuesta óptimo a los datos solicitados. En este contexto, un nuevo paradigma denominado "Redes híbridas centradas en la información" (hICN) aparece, tratando de incorporar todas las características del modelo ICN en el núcleo IP existente.

En el presente proyecto, se simulan escenarios usando software de virtualización para comprender el funcionamiento al implementar el modelo híbrido en una red básica.

Dentro de la misma, se experimentó utilizando máquinas virtuales con Ubuntu, una tarjeta de desarrollo Raspberry Pi 3 B+ y un Smartphone. Además, se utilizó el encaminador "hICN-Light" para el procesamiento de dos nuevos tipos de paquetes. Posteriormente, se efectuaron pruebas de desempeño por medio de la herramienta "hiperf" para analizar el rendimiento del ancho de banda de la red.

Los resultados obtenidos demostraron la compatibilidad entre distintas plataformas y la red hICN. Por otro lado, en escenarios de distribución de video lineal desplegados por Cisco, se evidenció que el consumo de memoria se mantuvo estable mientras los visualizadores aumentaban, y que el tráfico enviado desde el servidor se redujo.

Finalmente, la red híbrida permite aprovechar las características del modelo centrado en la información, logrando una integración sólida con la infraestructura IP actual.

Palabras Clave: ICN, hICN, encaminador, plataformas

ABSTRACT

Today, users around the world surf the Internet at a dizzying pace, creating the need for large investments in data center replicas in strategic geographic locations to provide an optimal response time to requested data. In this context, a new paradigm called "Hybrid Information-Centric Networking" (hICN) appears, which tries to incorporate all the features of the ICN model into the existing IP core.

In the present project, scenarios are simulated using virtualization software to understand the performance of implementing the hybrid model in a small topology.

Within it, experiments were done using virtual machines with Ubuntu, a Raspberry Pi 3 B+ development board and a smartphone. In addition, the "hICN-Light" forwarder was used for processing two new packet types. Subsequently, performance tests were carried out using the "hiperf" tool to analyze the network bandwidth performance.

The results proved the compatibility between different platforms and the hICN network. On the other hand, on linear video distribution scenarios deployed by Cisco, it was noted that memory consumption remained stable while viewers increased, and that traffic sent from the server was reduced.

Finally, the hybrid network allows to take advantage of information-centric model's characteristics, achieving a solid integration with the current IP infrastructure.

Keywords: *ICN, hICN, forwarder, platforms*

ÍNDICE GENERAL

RESUMEN	I
ABSTRACT	II
ÍNDICE GENERAL	III
ABREVIATURAS	V
ÍNDICE DE FIGURAS	VI
ÍNDICE DE TABLAS	VII
CAPÍTULO 1	1
1. Introducción	1
1.1 Descripción del problema	2
1.2 Justificación del problema	3
1.3 Objetivos	4
1.3.1 Objetivo General	4
1.3.2 Objetivos Específicos	4
1.4 Marco teórico	5
1.4.1 The FD.io Community[4]:	5
1.4.2 Information-Centric Networking (ICN)[5]:	5
1.4.3 Hybrid Information-Centric Networking (hICN)[1]:	6
1.4.4 Content Distribution Network (CDN)[14]:	10
CAPÍTULO 2	11
2. Metodología	11
2.1 Sistema Operativo utilizado en las máquinas virtuales	11
2.2 Registro de repositorio de los binarios hICN	11
2.3 Instalación de pila de aplicaciones asociadas a hICN	11
2.3.1 Encaminador Hicn-Light	12
2.4 Configuración del encaminador hICN	13
2.4.1 Configuración del cliente	13
2.4.2 Configuración del router hICN	14

2.4.3	Configuración del servidor	15
2.5	Configuración de APP's Android hICN	16
2.5.1	Hybrid ICN Network Service:	16
2.5.2	hICN Speed Test & Tools	18
2.6	Configuración de la Raspberry Pi 3 B+	20
CAPÍTULO 3		21
3.	Resultados y análisis	21
3.1	Primer escenario	25
3.2	Segundo escenario	29
3.3	Tercer escenario	32
3.4	Análisis de costos.....	35
3.4.1	Virtualización de máquinas	35
3.4.2	Implementación en una Raspberry Pi 3 B+	36
3.4.3	Implementación en un Smartphone	36
3.4.4	Resumen de costos	37
CAPÍTULO 4		38
4.	Conclusiones Y Recomendaciones	38
	Conclusiones	38
	Recomendaciones	40
BIBLIOGRAFÍA		41

ABREVIATURAS

IoT	Internet of Things
ICN	Information-Centric Networking
hICN	Hybrid Information-Centric Networking
PC	Personal Computer
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
IP	Internet Protocol
PIT	Pending Interest Table
FIB	Forwarding Information Base
CS	Content Storage
CDN	Content Delivery Network
VPP	Vector Packet Processor
RTC	Red Telefónica Conmutada
5G	Fifth Generation
UDP	User Datagram Protocol
API	Application Programming Interface
ID	Identification
DHCP	Dynamic Host Configuration Protocol
SSH	Secure Shell
Wi-Fi	Wireless Fidelity
GB	Giga Bytes
RAM	Random Access Memory
LTE	Long Term Evolution
NAT	Network Address Translation
Mbps	Megabits per second
RAM	Random Access Memory

ÍNDICE DE FIGURAS

Figura 1.1: Torre de protocolos ICN e IP	5
Figura 2.1: Pantalla principal de la aplicación.....	16
Figura 2.2: Menú para navegar entre el Home y las configuraciones.	16
Figura 2.3: Menú de la opción configuraciones.	17
Figura 2.4: Menú de la opción WiFi Configuration.	17
Figura 2.5: Menú de la opción WiFi IPv4.	18
Figura 2.6: Menú de la herramienta Hiperf.	18
Figura 2.7: Menú de la herramienta Hiping.....	19
Figura 2.8: Menú de la herramienta Higet.	19
Figura 2.9: Pruebas realizadas por medio de sesiones SSH con la Raspberry Pi y las máquinas virtuales.	20
Figura 3.1: Red de distribución de video[15].	21
Figura 3.2: Memoria usada por hICN y TCP en un canal simple incrementando el número de visualizaciones[15].	22
Figura 3.3: Porcentaje del tráfico servido por cada componente en la red[15].....	22
Figura 3.4: (a) Red heterogénea de distribución de video con routers hICN negros habilitados[15].	23
Figura 3.5: TCP vs hICN sobre Wi-Fi con promedio de calidad de video recibido por el cliente y número de switches up (#SWu) y down (#SWd) con diferente tráfico cruzado y carga en el canal Wi-Fi. La parte de abajo muestra la ganancia del flujo hICN respecto a TCP[15].....	23
Figura 3.6: Acceso de la Red Heterogénea comparando entre hICN e ICN[15].	24
Figura 3.7: Topología del primer escenario.	25
Figura 3.8: Ejecución de la aplicación ping hICN en el cliente.....	26
Figura 3.9: Ejecución del servidor ping en el servidor.	26
Figura 3.10: Prueba Hiperf realizada en el cliente.	27
Figura 3.11: Gráfico de los datos de la prueba Hiperf en la máquina virtual del cliente.	28

Figura 3.12: Topología del segundo escenario.....29

Figura 3.13: Resultados de la prueba Hiperf en la aplicación móvil.....30

Figura 3.14: Gráfico de los datos de la prueba Hiperf en el Smartphone.....31

Figura 3.15: Topología del tercer escenario.32

Figura 3.16: Ejecución de la aplicación de ping hICN en la tarjeta de desarrollo.33

Figura 3.17: Resultados de la prueba Hiperf en la tarjeta de desarrollo.33

Figura 3.18: Gráfico de los datos de la prueba Hiperf en la tarjeta de desarrollo.....34

ÍNDICE DE TABLAS

Tabla 3.1: Costos Generales.....37

CAPÍTULO 1

1. INTRODUCCIÓN

A lo largo de su historia, la red de redes (también llamada Internet) ha experimentado cambios en su concepción como infraestructura de alcance global, partiendo desde la comunicación directa entre terminales fijos, hasta alcanzar complejos esquemas distribuidos capaces de servir contenido a usuarios alrededor del mundo.

Con el creciente volumen de tráfico generado por dichos usuarios y la proyección de un aumento considerable de dispositivos Internet of Things (IoT) para los próximos años, la infraestructura actual que gobierna Internet será desafiada, por lo que surge la interrogante respecto a si la misma podrá estar a la altura de un nuevo escenario más exigente. A raíz de este hecho, empresas reconocidas en el ámbito de las redes han emprendido un periplo en la búsqueda de nuevas arquitecturas; en medio de este contexto aparece el paradigma de las redes centradas en la información (ICN), cuyo enfoque radica en el contenido como tal sin considerar una dependencia en su ubicación.

En el Capítulo 3 se realizan pruebas de las arquitecturas ICN, de forma más específica, la arquitectura Hybrid Information-Centric Networking (hICN), en un ambiente virtualizado.

Luego se procede a realizar las mismas pruebas sobre hardware de tarjetas de desarrollo, en este caso Raspberry Pi 3, y en máquinas virtuales o personal computers (PC's), con el fin de comprender el funcionamiento del encaminamiento basado en contenidos.

Finalmente se analizan los resultados de estas pruebas con respecto al rendimiento que tienen las arquitecturas actuales Content Delivery Network (CDN).

1.1 Descripción del problema

Es necesario encontrar una nueva arquitectura de red que permita distribuir contenido de manera eficiente y escalable según las necesidades tecnológicas emergentes de las redes del futuro.

En el contexto de las investigaciones sobre el internet del futuro, han aparecido sugerencias sobre un nuevo paradigma llamado ICN. Los investigadores notaron que la Internet tradicional ha tomado un modelo de comunicación host-to-host, que de una forma inherente tiene un enfoque en el que a los usuarios les importa la localización de la información, sin embargo, a los usuarios solo les interesa la información que ellos solicitan[1].

El paradigma sufre un cambio de “Dónde” a “Qué”, impulsando la aparición de nuevas capas de transportes que puedan manejar la comunicación entre grupos de redes con identificadores de información, y no por direcciones.

1.2 Justificación del problema

La importancia de encontrar una arquitectura de red que soporte las exigencias de las redes del futuro es clave.

Una idea intuitiva que puede aparecer al buscar una solución es preservar el esquema actual y escalar verticalmente, es decir, mejorar las capacidades de procesamiento en términos de hardware y software para dispositivos intermedios. Sin embargo, esta solución, aunque tentadora, trae consigo inversiones de dinero significativas, y, por otro lado, el protocolo de IPv4 quedará obsoleto ya que las direcciones Internet Protocol (IP) provistas no podrán abastecer el creciente número de dispositivos conectados a Internet[2].

Cabe destacar que la empresa Cisco estimó que para el año 2023 el número de dispositivos conectados a redes IP será tres veces más grande que la población mundial[3]. Las implicaciones de dicho aumento se traducirán en más peticiones hacia plataformas en la nube, lo cual puede llegar a perjudicar la experiencia del usuario, en caso de mantener la arquitectura contemporánea.

Al realizar pruebas con las arquitecturas CCN y hICN se puede obtener datos para realizar una comparación entre estas tecnologías en desarrollo y las actuales, permitiendo observar las ventajas y desventajas que tienen en un escenario donde se necesita una escalabilidad eficiente ante la proyección de la gran demanda de información de las tecnologías IoT, Smart Cities y a la par de las redes Fifth Generation (5G).

Al pensar en los retos que tiene la tecnología 5G, las próximas generaciones de redes de radio móviles cuentan con la ambición de servir a un gran número de casos y las ICN han sido identificadas como un candidato idóneo que trae beneficios en la red de borde en términos de rendimiento, escalabilidad y costo.

1.3 Objetivos

1.3.1 Objetivo General

Desplegar escenarios simulados utilizando software de virtualización para la comprensión del funcionamiento y desempeño de las redes centradas en los contenidos.

1.3.2 Objetivos Específicos

- Probar el funcionamiento de las hICN sobre ambientes virtualizados considerando una topología de red básica.
- Desplegar los componentes de una hICN sobre hardware de tarjeta de desarrollo y Smartphone con el fin de comprender y poner a prueba el funcionamiento del encaminamiento basado en contenidos.
- Comparar la eficiencia de las redes convencionales CDN frente a la red hICN.

1.4 Marco teórico

1.4.1 The FD.io Community[4]:

The Fast Data Project es un proyecto open source colaborativo con el fin de establecer un alto rendimiento en un framework de servicios para ambientes de cómputo dinámicos.

Cuenta con una wiki con soporte de la Linux Foundation en donde trabajan con diferentes proyectos sobre el desarrollo para la implementación de las ICN, y en este proyecto se utiliza el subproyecto hICN.

1.4.2 Information-Centric Networking (ICN)[5]:

Es un método de desarrollo de la infraestructura de Internet para apoyar directamente la comunicación centrada en datos e independiente de la ubicación. Introduce datos con nombres únicos como el principio básico de comunicación.

El acceso a los datos es independiente de la ubicación, la aplicación y el almacenamiento, lo que permite el almacenamiento en caché dentro de la red y la movilidad sin restricciones.

En la siguiente figura se puede observar una comparación de la pila de protocolos de la red ICN y una red IP:

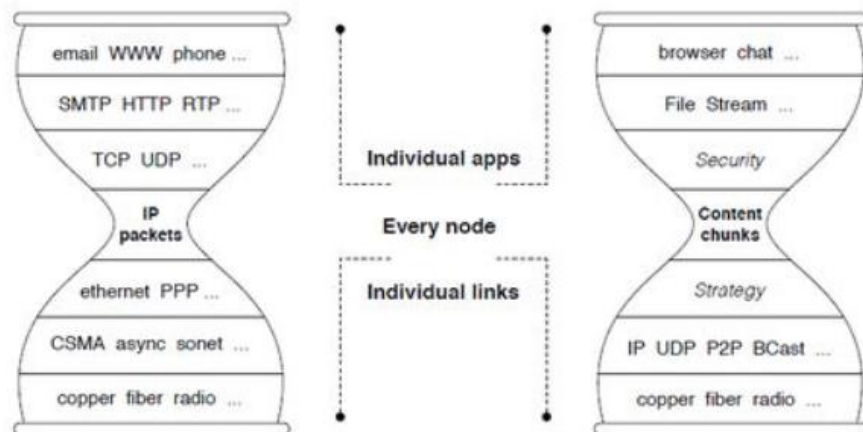


Figura 1.1: Torre de protocolos ICN e IP[6].

En ambas redes se pueden usar las mismas aplicaciones finales, siendo esta una importante característica de las redes ICN, pero los cambios más significativos se dan en las capas inferiores en donde los datos se dividen en fragmentos (Chunks) y son cifrados. Por medio de una interfaz se indica bajo qué protocolo serán enviados, permitiendo el uso de protocolos como: IP, User Datagram Protocol (UDP), entre otros.

1.4.3 Hybrid Information-Centric Networking (hICN)[1]:

La principal meta de las arquitecturas hICN es incorporar las capacidades que tienen las ICN dentro de las redes IP existentes, sin sacrificar alguna característica y reusando la mayoría de la tecnología de software y hardware existente. Como resultado, se reduciría el esfuerzo de adoptar las hICN en un futuro cercano.

En hICN se definen dos componentes: name prefix y name suffix. El name prefix es utilizado por la aplicación o servicios en la red. El name suffix se utiliza para indexar los segmentos de la información que utiliza la aplicación[7].

La red hICN cuenta con dos tipos de paquetes: interest packets y data packets. Estos paquetes logran viajar normalmente en la red IP debido a que son encaminados por nombre, y, como el nombre es una dirección IPv6, el enrutamiento se realiza de la misma manera que un paquete IP tradicional[8].

La comunidad de FD.io, partiendo desde una versión provista por Cisco como aporte inicial, es la responsable del desarrollo y mantenimiento del proyecto hICN, poniendo a disposición varias aplicaciones que pueden ser aprovechadas por terceros para desplegar escenarios reales y explorar las bondades de la arquitectura en cuestión.

La implementación del software es compatible con diversos sistemas operativos, entre ellos:

- Ubuntu 18.04 LTS (amd64, arm64)
- Red Hat Enterprise Linux 7
- CentOS 7
- Debian Stable/Testing
- Windows 10
- Android 10 (amd64, arm64)
- iOS 13
- macOS 10.15

El proyecto hICN está constituido por los siguientes módulos:

1.4.3.1 Core Library[9]

Abreviada como libhcn, Core Library es la librería núcleo de la implementación híbrida del modelo ICN. La misma proporciona un conjunto de dependencias codificadas en lenguaje C encargadas de abstraer mecanismos internos relacionados con el funcionamiento de sus componentes, además de proveer una Application Programming Interface (API) de sockets general para paquetes IPv4 o IPv6.

La librería se divide en varias capas, destacan: un formato estándar de paquetes hICN en conjunto con una API que gestiona las cabeceras de dichos paquetes, una capa de red responsable de traducir las direcciones de origen y destino conforme a las necesidades del esquema híbrido, y una API de transporte que permite generar paquetes señuelo (dummy packets) de tipo Interest y Data.

1.4.3.2 Transport Library[10]

Dentro de la librería se incluye una API de sockets para aplicaciones que requieran comunicarse usando el stack de protocolos hICN, así como la implementación de objetos indispensables (paquetes Interest y Data, y nombres/etiquetas de contenido) llevada a cabo de la mano de la Core Library. Adicionalmente, se encuentran embebidos protocolos y servicios de transporte, tales como red telefónica conmutada (RTC) y autenticación, respectivamente.

1.4.3.3 Portable Forwarder[11]

También llamado hicn-light, es un componente orientado al reenvío de datos que opera con comunicación entre procesos y sockets estándares. Portable Forwarder orquesta el direccionamiento de paquetes basándose en el contenido de la tabla Forwarding Information Base (FIB) respectiva.

Las acciones más importantes que se pueden llevar a cabo con el módulo Forwarder son:

- Escuchar conexiones TCP o UDP, configurando un nombre simbólico y una dirección IPv4 o IPv6.
- Escuchar conexiones de tipo hICN, asignando un nombre simbólico y una dirección IPv4 o IPv6.
- Realizar adyacencias con otros encaminadores.

Algunos de los comandos principales que se utilizan con el ejecutable `hicn-light-control` se detallan a continuación:

- `add listener hicn`: Crea una interfaz hICN para establecer una adyacencia con otro encaminador.
- `add punting`: Redirecciona los paquetes hICN hacia la interfaz especificada.
- `add connection hicn`: Crea una conexión hICN en el encaminador local con opciones especificadas.
- `add route`: Crea una ruta que redirecciona los paquetes con un prefijo hICN definido hacia la conexión especificada.

Se puede obtener la lista de comandos disponibles en el encaminador `Hicn-Light` de la siguiente manera:

```
$ hicn-light-control help
```

1.4.3.4 Utility Applications[12]

Este módulo está conformado por 3 herramientas para efectuar pruebas de conectividad y rendimiento sobre la topología a desplegar:

- `Hicn-ping-server`: Escucha peticiones de ping desde un servidor.
- `Hicn-ping-Client`: Genera peticiones de ping desde un cliente.
- `Hiperf`: Crea flujos de datos con la finalidad de medir el desempeño en una red. Este componente resulta indispensable ya que juega un rol crucial en uno de los objetivos específicos planteados, el cual consiste en la comparación entre la arquitectura hICN y la convencional basándose en métricas.

1.4.3.5 Applications[13]

En este último módulo se cuenta con dos herramientas de servicio y peticiones HTTP:

- Hicn-Http-Proxy: Es un proxy inverso que aumenta el rendimiento de un servidor HTTP/TCP.
- Higet: Es un cliente HTTP no interactivo que trabaja sobre hICN.

1.4.4 Content Distribution Network (CDN)[14]:

CDN distribuye los contenidos del servidor de origen hacia los servidores de réplica cercanos a los clientes finales. Los servidores de réplica en un CDN almacenan un conjunto muy selectivo de contenidos y sólo las peticiones de ese contenido son servidas para que el porcentaje de aciertos pueda acercarse al cien por ciento.

CAPÍTULO 2

2. METODOLOGÍA

Durante este capítulo se dará a conocer los archivos binarios y ejecutables necesarios, qué opciones de configuración contienen, y de qué manera se utilizan, para poder implementar una red básica con arquitectura hICN que permita demostrar y comprender el funcionamiento que tiene en distintas plataformas, debido a la versatilidad y compatibilidad que proporcionan los desarrolladores del proyecto.

2.1 Sistema Operativo utilizado en las máquinas virtuales

Se utilizaron máquinas virtuales en Virtual Box con el sistema operativo Ubuntu 18.04.5 LTS.

2.2 Registro de repositorio de los binarios hICN

Para poder instalar los binarios y ejecutables hICN se registraron los repositorios del proyecto con los siguientes comandos:

```
$ curl -s https://packagecloud.io/install/repositories/fdio/release/script
.deb.sh | sudo bash
$ curl -s https://packagecloud.io/install/repositories/fdio/hicn/script.d
eb.sh | sudo bash
```

2.3 Instalación de pila de aplicaciones asociadas a hICN

Existe poca información actualizada sobre la instalación y se encuentra dispersa en diferentes fuentes, por lo que se realizó una investigación para aplicar una correcta instalación. Se deben instalar las librerías estáticas o dinámicas junto con sus archivos cabecera en donde estos archivos tienen agregado -dev en el nombre de la librería.

La instalación que se realizó se detalla a continuación:

2.3.1 Encaminador Hicn-Light

El funcionamiento del encaminador Hicn-Light es basado en sockets, por lo que no es necesario una tarjeta de red dedicada para poder utilizarlo, sin embargo, el poder de procesamiento es menor que el encaminador Vector Packet Processing (VPP), debido a que este encaminador sí utiliza una tarjeta de red dedicada por la forma de procesar los paquetes en vectores, abarcando una tasa más alta de paquetes.

Si bien es cierto que VPP tiene más rendimiento, así mismo necesita más recursos para poder funcionar, debido a esto, se eligió el encaminador Hicn-Light por los recursos limitados disponibles.

El encaminador Hicn-Light necesita de algunas librerías (detalladas en la sección 1.4.3) para su correcto funcionamiento y se las puede instalar con los siguientes comandos:

```
$ sudo apt install libhicn=20.09-33 libhicn-dev=20.09-33
libhicntransport=20.09-33 libhicntransport-dev=20.09-33 hicn-
utils=20.09-33 libhicnctrl=20.09-33 libhicnctrl-dev=20.09-33 hicn-
apps=20.09-33 hicn-apps-dev=20.09-33
```

La aplicación principal del encaminador Hicn-Light se la puede instalar con el siguiente comando:

```
$ sudo apt install hicn-light=20.09-33
```


2.4 Configuración del encaminador hICN

Los encaminadores hICN se pueden configurar como equipo terminal o como un router hICN, para la presente implementación se hizo uso del encaminador Hicn-Light en el cliente, router y servidor. Las configuraciones que se realizaron en los encaminadores se detallan a continuación:

2.4.1 Configuración del cliente

Para el cliente, el encaminador Hicn-light se puede configurar con el ejecutable `hicn-light-control` que cuenta con diversas opciones (mencionadas en la sección 1.4.3), o por medio de la creación de un archivo `configFile.cfg`, pasando como parámetro al ejecutable `hicn-light-daemon` con la opción `-config <ruta del archivo>`.

El contenido del archivo `configFile.cfg` utilizado se detalla a continuación:

```
#Crea una interfaz principal para recibir paquetes de interés de la red
add listener udp remote0 192.168.100.86 12345 enp0s8

#Crea una conexión con encapsulamiento udp se indica primero la ip remota
#y luego la ip local con sus respectivos puertos
add connection udp conn0 192.168.100.87 12345 192.168.100.86 12345

#Se indica la ruta del prefijo IPv6 del contenido nombrado y el peso
#de la conexión
add route conn0 b001::/16 1
```

En el cliente se debe establecer una conexión con el router hICN, para ello se crea un socket que es capaz de escuchar el contenido determinado por el prefijo hICN indicado en la ruta de la conexión.

2.4.2 Configuración del router hICN

Al ser utilizado como router hICN se debe activar la funcionalidad de almacenamiento bajo demanda. Se creó una conexión por cada dispositivo utilizado para la red básica.

El contenido del archivo configFile.cfg utilizado se detalla a continuación:

```
#Cliente
add listener udp remote0 192.168.100.87 12345 enp0s8
add connection udp conn0 192.168.100.86 12345 192.168.100.87 12345
add route conn0 b001::/16 1

#Servidor
add listener udp remote1 192.168.56.102 12345 enp0s9
add connection udp conn0 192.168.56.103 12345 192.168.56.102 12345
add route conn1 b001::/16 1

#Celular
add connection udp conn0 192.168.100.67 49152 192.168.100.87 49152
add route conn2 b001::/16 1

#Raspberry
add listener udp remote0 192.168.100.75 12345 enp0s8
add connection udp conn0 192.168.100.74 12345 192.168.100.75 12345
add route conn0 b001::/16 1
```

En el router hICN se deben establecer todas las conexiones con los clientes y el servidor de la red básica, indicando el mismo prefijo hICN para una correcta transmisión de los datos.

2.4.3 Configuración del servidor

Para el servidor, el encaminador Hicn-light tuvo una configuración similar al cliente, en este caso solo se debe establecer una conexión que reciba los paquetes bajo el prefijo hICN asignado, debido a que no necesita crear otra adyacencia.

El contenido del archivo configFile.cfg utilizado se detalla a continuación:

```
#Router
add listener udp remote0 192.168.56.103 12345 enp0s9
add connection udp conn0 192.168.56.102 12345 192.168.56.103 12345
add route conn0 b001::/16 1
```

En el servidor se debe establecer la conexión con el router hICN para crear un camino que permita el envío de los datos solicitados, así mismo indicando el prefijo hICN correcto.

2.5 Configuración de APP's Android hICN

Para poder utilizar este tipo de redes desde un Smartphone, el equipo de desarrollo del proyecto Fast Data pone a disposición algunas aplicaciones:

2.5.1 Hybrid ICN Network Service:

Se deben activar los servicios hICN services.

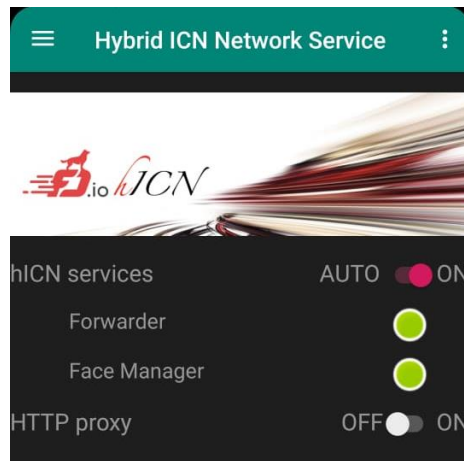


Figura 2.1: Pantalla principal de la aplicación.

Al dar clic en la esquina superior izquierda, se debe elegir la opción Settings para ingresar los datos necesarios y así crear la conexión con el router hICN.

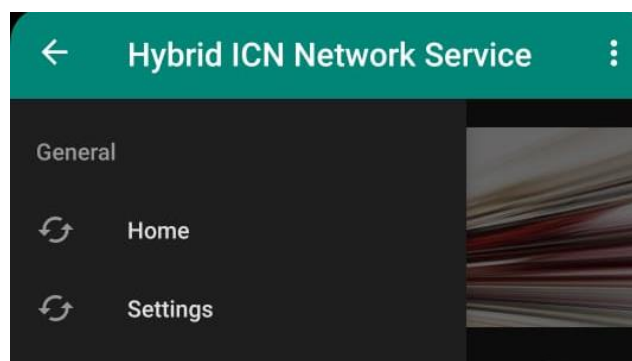


Figura 2.2: Menú para navegar entre el Home y las configuraciones.

Para indicar las direcciones IP locales y remotas se debe ir a la opción de Wi-Fi Configuration.

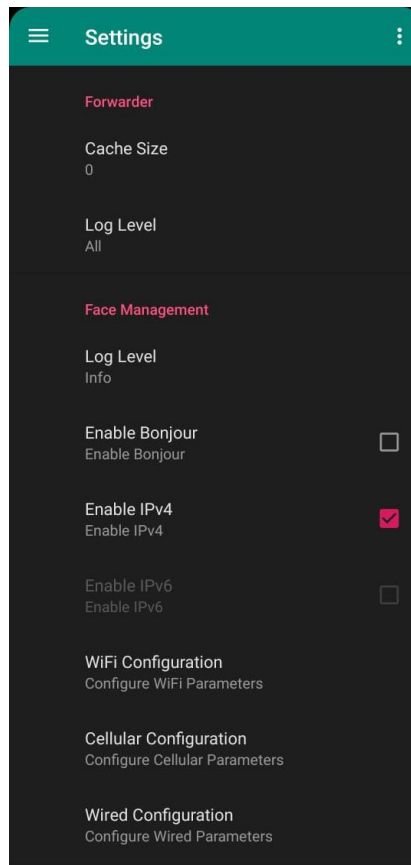


Figura 2.3: Menú de la opción configuraciones.

Actualmente solo está disponible la opción para configurar IPv4 y una configuración manual.

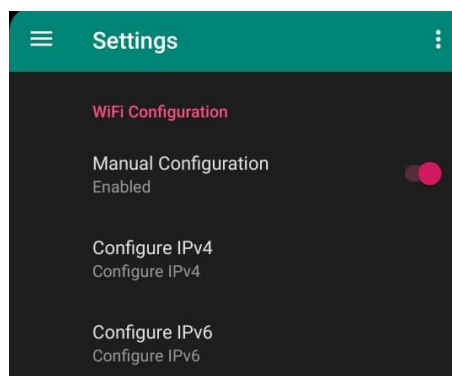


Figura 2.4: Menú de la opción Wi-Fi Configuration.

Aquí se debe especificar el puerto origen y la dirección IP remota con su respectivo puerto.

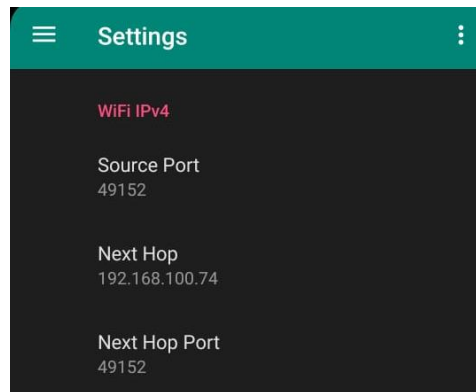


Figura 2.5: Menú de la opción Wi-Fi IPv4.

2.5.2 hICN Speed Test & Tools

Esta aplicación cuenta con 3 herramientas que realizan distintas pruebas utilizando los paquetes de la arquitectura hICN:

- Hiperf: Permite visualizar el uso del ancho de banda.

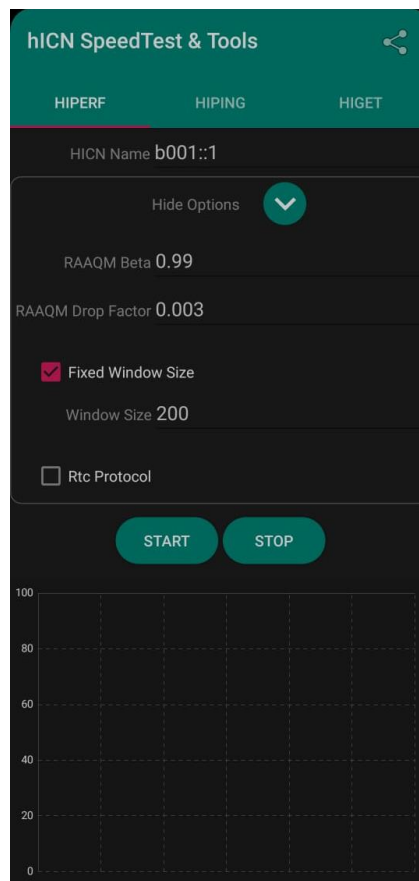


Figura 2.6: Menú de la herramienta Hiperf.

- Hiping: Permite realizar una prueba de ping entre dos dispositivos.

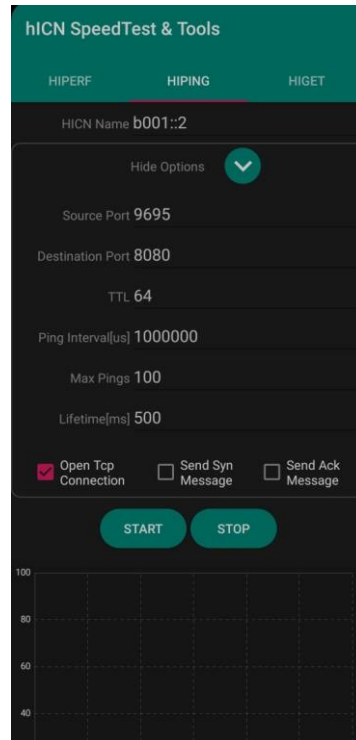


Figura 2.7: Menú de la herramienta Hiping.

- Higet: Permite realizar una petición HTTP.

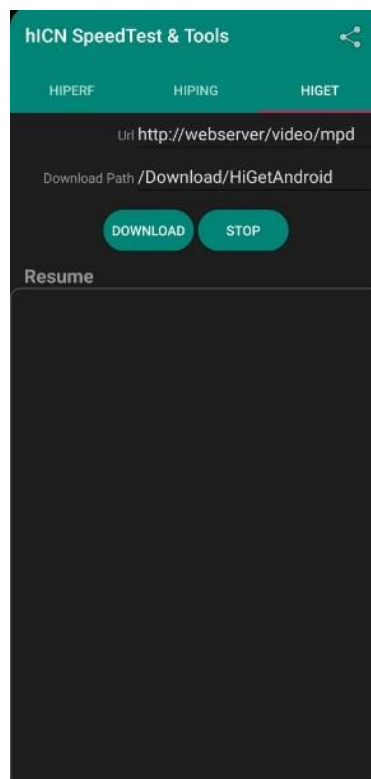
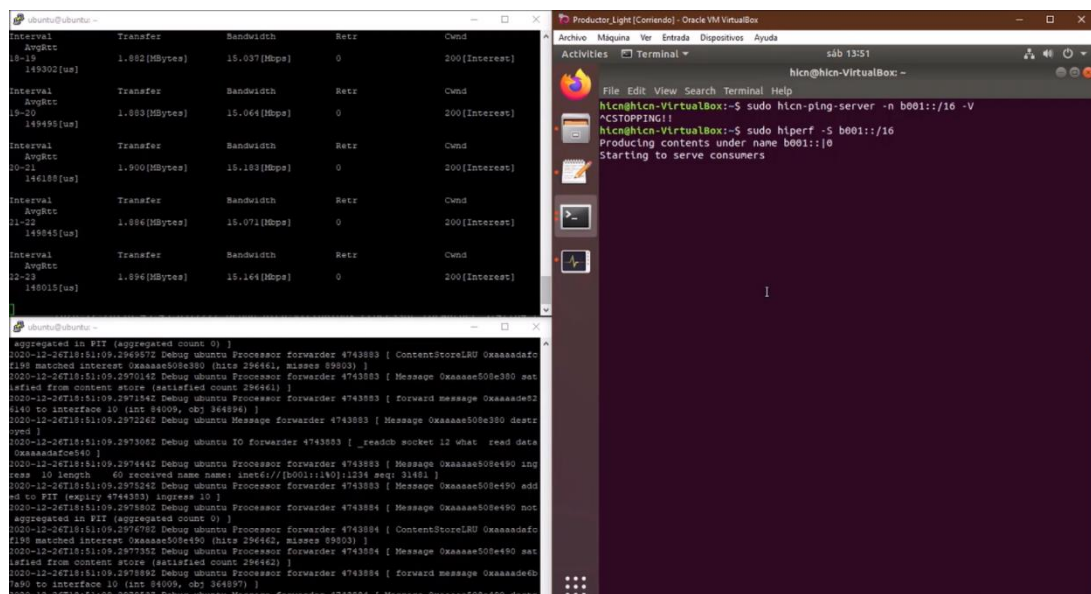


Figura 2.8: Menú de la herramienta Higet.

2.6 Configuración de la Raspberry Pi 3 B+

Con la finalidad de añadir la tarjeta de desarrollo a la red hICN básica, se instaló Ubuntu 18.04 para ARM64. Al ser una tarjeta con pocos recursos se usó una interfaz gráfica más ligera.

Luego de la instalación del sistema operativo, se procedió a habilitar el encaminador Hicn-Light de manera similar a las instalaciones de: el consumidor, productor y router hICN. Se estableció una conexión vía Secure Shell (SSH) para la instalación y ejecución de los programas por medio de la aplicación Putty.



The image shows two terminal windows. The left window displays the output of a network performance test, showing metrics for Transfer, Bandwidth, Retx, and Cwnd over several intervals. The right window shows the execution of commands to start a hICN server and a consumer, with output indicating that the server is producing contents and starting to serve consumers.

```
Interval      Transfer      Bandwidth      Retx      Cwnd
AvgRtc
18-19         1.882[Mbytes] 15.037[Mbps]   0         200[Interest]
149302[us]

Interval      Transfer      Bandwidth      Retx      Cwnd
AvgRtc
19-20         1.883[Mbytes] 15.064[Mbps]   0         200[Interest]
149495[us]

Interval      Transfer      Bandwidth      Retx      Cwnd
AvgRtc
20-21         1.900[Mbytes] 15.183[Mbps]   0         200[Interest]
146188[us]

Interval      Transfer      Bandwidth      Retx      Cwnd
AvgRtc
21-22         1.886[Mbytes] 15.071[Mbps]   0         200[Interest]
149845[us]

Interval      Transfer      Bandwidth      Retx      Cwnd
AvgRtc
22-23         1.896[Mbytes] 15.164[Mbps]   0         200[Interest]
148015[us]

Aggregated in PIT (aggregated count 0)
2020-12-26T18:18:10.9297446Z Debug ubuntu Processor forwarder 4743883 [ ContentStoreLRU Okaaaadafo
1198 matched interest Okaaaas50e380 (hits 296461, misses 89603) ]
2020-12-26T18:18:10.9297542Z Debug ubuntu Processor forwarder 4743883 [ Message Okaaaas50e380 sat
isfied from content store (satisfied count 296461) ]
2020-12-26T18:18:10.9297542Z Debug ubuntu Processor forwarder 4743883 [ forward message Okaaaade5
848 to interface 10 (int 86009, obj 164896) ]
2020-12-26T18:18:10.9297226Z Debug ubuntu Message Forwarder 4743883 [ Message Okaaaas50e380 desti
nyed ]
2020-12-26T18:18:10.9297308Z Debug ubuntu IO forwarder 4743883 [ _readcb socket 12 what read data
Okaaaadarc640 ]
2020-12-26T18:18:10.9297446Z Debug ubuntu Processor forwarder 4743883 [ Message Okaaaas50e490 ind
ress 10 lengh 60 received from name: int64://[8002:140]:1234 seq: 2142 ]
2020-12-26T18:18:10.9297524Z Debug ubuntu Processor forwarder 4743883 [ Message Okaaaas50e490 add
ed to PIT (expiry 4744383) ingress 10 ]
2020-12-26T18:18:10.9297882Z Debug ubuntu Processor forwarder 4743884 [ ContentStoreLRU Okaaaadafo
1198 matched interest Okaaaas50e490 (hits 296462, misses 89603) ]
2020-12-26T18:18:10.9297357Z Debug ubuntu Processor forwarder 4743884 [ Message Okaaaas50e490 sat
isfied from content store (satisfied count 296462) ]
2020-12-26T18:18:10.9297882Z Debug ubuntu Processor forwarder 4743884 [ forward message Okaaaade5
848 to interface 10 (int 86009, obj 164897) ]
2020-12-26T18:18:10.9297857Z Debug ubuntu Message Forwarder 4743884 [ Message Okaaaas50e490 desti
```

Figura 2.9: Pruebas realizadas por medio de sesiones SSH con la Raspberry Pi y las máquinas virtuales.

CAPÍTULO 3

3. RESULTADOS Y ANÁLISIS

Se realizó la implementación de una red básica con arquitectura hICN en diferentes plataformas con la intención de demostrar el funcionamiento del enrutamiento de contenido por nombre en este tipo de redes.

Las herramientas de medición de rendimiento son limitadas en cuanto a la versión empleada en el presente proyecto, por lo que solo se logró utilizar la aplicación Hiperf para medir el rendimiento con el uso de ancho de banda máximo.

Debido a esto, se analizarán pruebas realizadas por CISCO detalladas a continuación, las cuales evidencian las ventajas de esta arquitectura respecto del paradigma actual de las redes.

Como primer análisis tenemos al rendimiento de conjunto de pruebas aplicadas con N clientes:

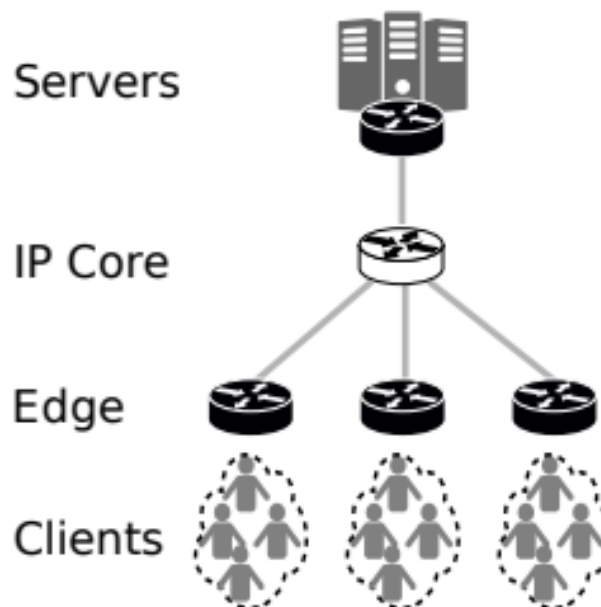


Figura 3.1: Red de distribución de video[15].

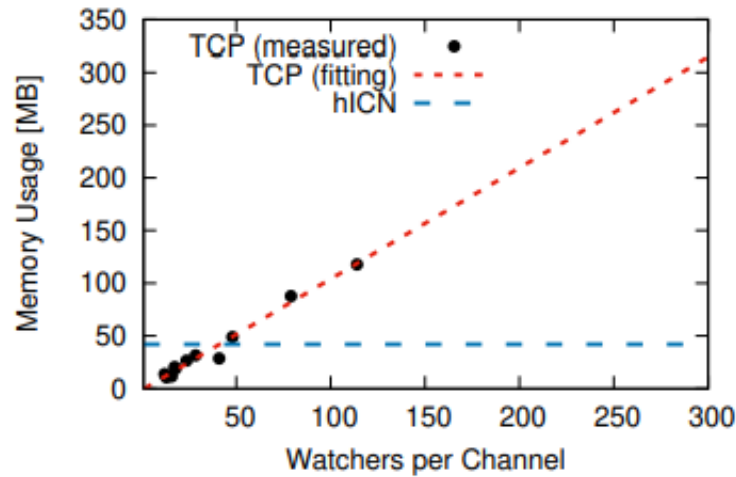


Figura 3.2: Memoria usada por h1CN y TCP en un canal simple incrementando el número de visualizaciones[15].

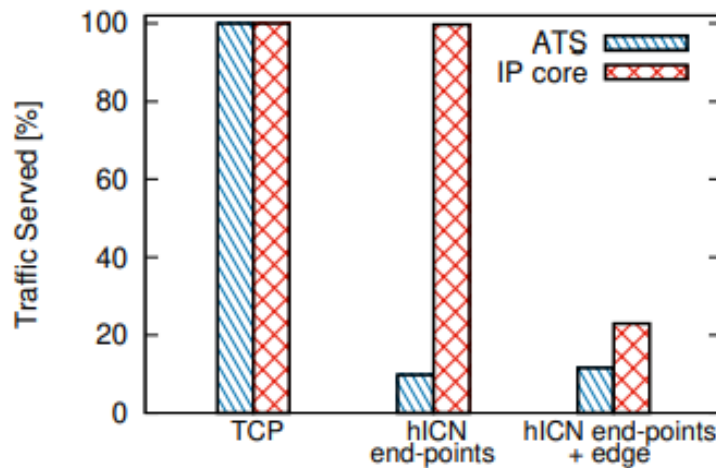


Figura 3.3: Porcentaje del tráfico servido por cada componente en la red[15].

El escenario planteado en la figura 3.1 es uno de los más comunes, teniendo una red de distribución de video.

De forma intuitiva se puede llegar a pensar que, al aumentar el número de visualizaciones como se observa en la figura 3.2, la memoria usada se elevaría de igual manera, y aquí es donde se destaca una de las características de las redes h1CN, al mantenerse constante el uso de memoria mientras las visualizaciones aumentan.

Al utilizar las redes h1CN con dispositivos en puntos finales y en el borde, se obtiene menor tráfico del servidor con respecto a una red actual TCP como se puede observar en la figura 3.3.

Como segundo análisis tenemos a la distribución de video lineal recibiendo los datos de una antena Wi-Fi y LTE:

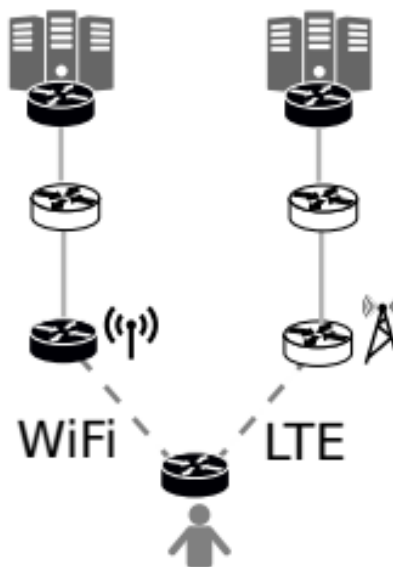


Figura 3.42: (a) Red heterogénea de distribución de video con routers hICN negros habilitados^[15].

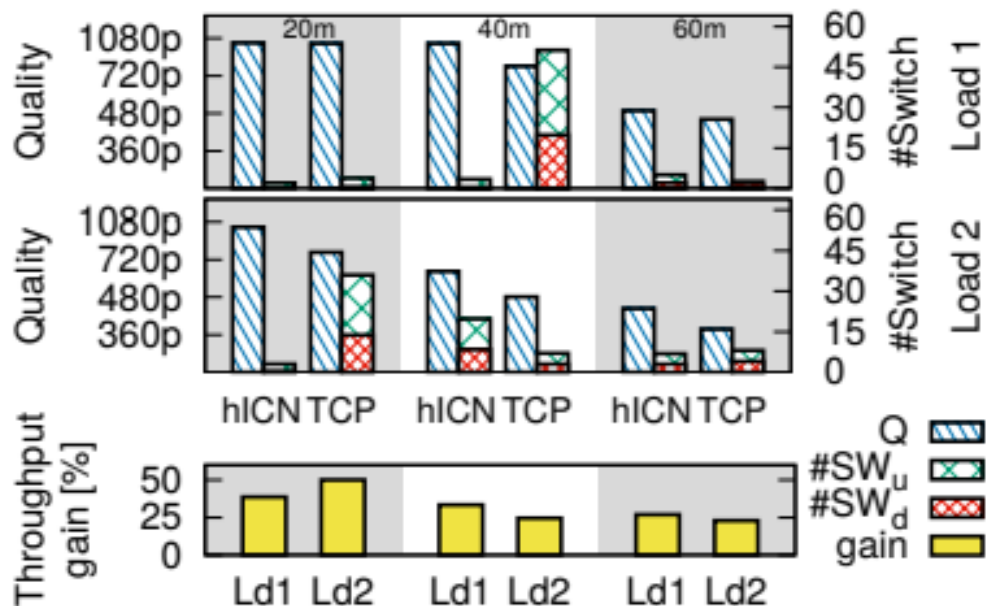


Figura 3.5: TCP vs hICN sobre Wi-Fi con promedio de calidad de video recibido por el cliente y número de switches up (#SW_u) y down (#SW_d) con diferente tráfico cruzado y carga en el canal Wi-Fi. La parte de abajo muestra la ganancia del flujo hICN respecto a TCP^[15].

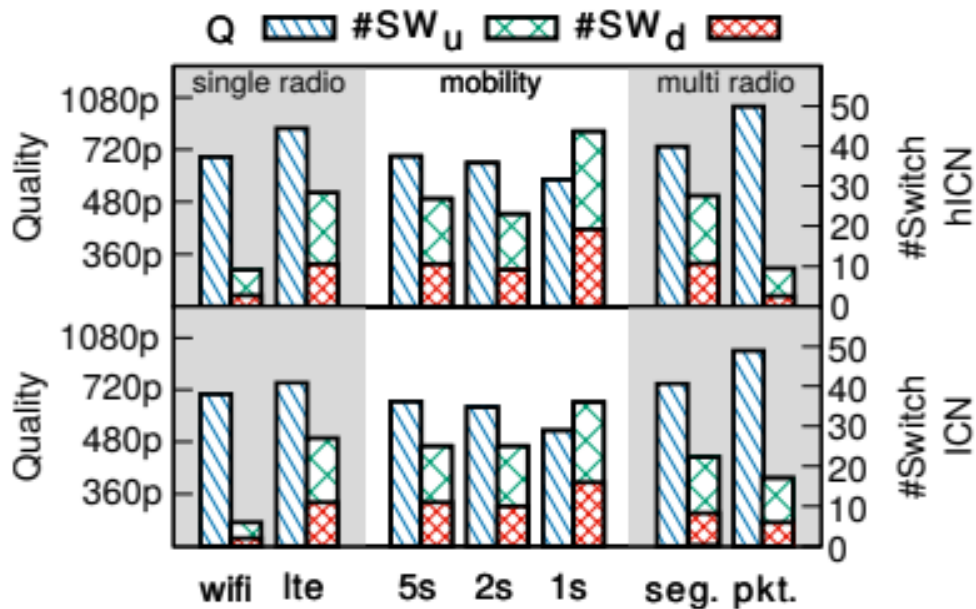


Figura 3.6: Acceso de la Red Heterogénea comparando entre hICN e ICN[15].

La tecnología de las redes hICN se encuentran en una etapa temprana de desarrollo y están pensadas para funcionar a gran escala con un número grande de visualizaciones. Se considera una red Heterogénea con usuarios que tienen acceso a Wi-Fi y Long Term Evolution (LTE radio).

hICN obtiene un rendimiento consistente con respecto a TCP en el rango de 23% (teniendo los nodos a 60 metros y con 75% de carga) a 50% (teniendo los nodos a 20m y con 75% de carga) de ganancia.

Los resultados demuestran que hICN obtiene un mejor promedio en calidad de video con respecto a TCP la mayoría del tiempo, resultando en una mejora en la calidad de experiencia para el cliente.

Estas pruebas fueron realizadas con versiones ya no disponibles de la pila hICN debido a los cambios y evoluciones que sufren los distintos aplicativos de la arquitectura. Teniendo estas consideraciones y utilizando las versiones que se encuentran disponibles se plantearon 3 escenarios distintos que son detallados a continuación:

3.1 Primer escenario

Para efectos prácticos y conocer el funcionamiento de la pila hICN, lo más sencillo es realizar pruebas en un ambiente virtualizado implementando la topología más básica de una red, desplegando un cliente, un servidor y un nodo intermedio con el papel de router hICN.

Al cliente se le configuró en la máquina virtual dos interfaces de red: una de tipo adaptador puente para realizar una conexión con el router hICN y una de tipo Network Address Translation (NAT) para obtener salida a internet. El router hICN tuvo una configuración de 3 interfaces de red: una de tipo adaptador puente para estar en el mismo segmento de red con el cliente y poder realizar una conexión, también una red de solo anfitrión para estar en el mismo segmento de red que el servidor y una red NAT para obtener salida a internet. Finalmente, al servidor se le configuró dos interfaces de red: una red solo anfitrión para realizar la conexión con el router hICN y una red NAT para tener salida a internet.

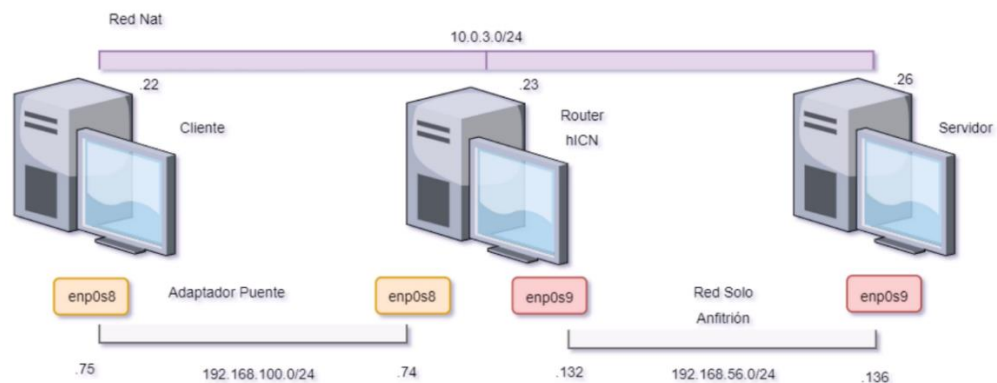


Figura 3.7: Topología del primer escenario.

Al tener desplegado el primer escenario es necesario conocer si existe comunicación de extremo a extremo.

```
hcn@hcn-VirtualBox:~$ sudo hcn-ping-client -n b001::1 -V
start ping
>>> send interest b001::1|0 src port: 9695 dst port: 8080 flags: TTL: 64

<<< received object.
<<< interest name: b001::1|0 src port: 9695 dst port: 8080 flags:
<<< object name: b001::1|0 src port: 8080 dst port: 9695 flags: path label 335
54432 (2) TTL: 64

>>> send interest b001::1|1 src port: 9695 dst port: 8080 flags: TTL: 64

<<< received object.
<<< interest name: b001::1|1 src port: 9695 dst port: 8080 flags:
<<< object name: b001::1|1 src port: 8080 dst port: 9695 flags: path label 335
54432 (2) TTL: 64
```

Figura 3.8: Ejecución de la aplicación ping hICN en el cliente.

```
hcn@hcn-VirtualBox:~$ sudo hcn-ping-server -n b001::/16 -V
```

Figura 3.9: Ejecución del servidor ping en el servidor.

Se utilizó una utilidad de la pila hICN que permite realizar pruebas de ping con los paquetes de interés y de datos de la arquitectura. En la figura 3.2 se utilizó en la máquina del cliente la herramienta hcn-ping-cliente que se encarga de enviar paquetes de interés en la red bajo el prefijo hICN que indica como parámetro con la opción “-n”. En la figura 3.3 se utilizó la herramienta hcn-ping-server en la máquina del servidor que se encarga de escuchar las peticiones de los paquetes interés que contengan el prefijo hICN indicado como parámetro con la opción “-n” y si existe alguna coincidencia envía la respuesta de los datos como se observa en la figura 3.2 en la parte que dice received object.

Se obtuvo como resultado que existe conexión entre el cliente y el servidor. La prueba también nos brinda información con la opción -V, sobre que puertos utiliza y el tiempo que se demora cada petición-respuesta.

Otra herramienta disponible en la pila hICN es una prueba de rendimiento del ancho de banda de la red llamada Hiperf.

```
h1cn@h1cn-VirtualBox:~$ sudo h1perf -C b001::1 -W 50
Starting download of b001::1|0
Interval      Transfer      Bandwidth      Retr
Cwnd          AvgRtt
0-1           1.069[MBytes] 8.447[Mbps]     0
50[Interest] 57248.3[us]
Interval      Transfer      Bandwidth      Retr
Cwnd          AvgRtt
1-2           1.042[MBytes] 8.324[Mbps]     0
50[Interest] 62430.6[us]
Interval      Transfer      Bandwidth      Retr
Cwnd          AvgRtt
2-3           1.085[MBytes] 8.671[Mbps]     0
50[Interest] 60005.3[us]
Interval      Transfer      Bandwidth      Retr
Cwnd          AvgRtt
3-4           1.071[MBytes] 8.559[Mbps]     0
50[Interest] 64906.6[us]
Interval      Transfer      Bandwidth      Retr
Cwnd          AvgRtt
4-5           1.110[MBytes] 8.855[Mbps]     0
50[Interest] 59780.1[us]
```

Figura 3.10: Prueba Hiperf realizada en el cliente.

La prueba Hiperf tiene un comportamiento similar a la prueba de ping hICN. Se ejecuta en el cliente indicando la opción “-C” y en el servidor la opción “-S”, otra opción básica que se debe indicar como se observa en la figura 3.4 es la “-W” que se refiere a una ventana de congestión fija que se encarga de recuperar los datos, también podemos observar de la misma figura que los datos son mostrados por consola sin ofrecer una gráfica para una mejor visualización.



Figura 3.11: Gráfico de los datos de la prueba Hiperf en la máquina virtual del cliente.

Se obtiene como resultado que la máxima velocidad en la red local es de aproximadamente uno 18 Megabits per second (Mbps), en la figura 3.5 se observa la recuperación de los datos de cada ventana de congestión de 50. Se utilizaron aproximadamente 50 datos para poder realizar la gráfica que en el eje horizontal dan como resultado aproximadamente 1.65 segundos, el descenso en el uso del ancho de banda se debe por errores de time out en las peticiones. Las versiones de la pila hICN disponibles que fueron utilizadas no estaban estables por lo que ocurrían errores de time out a menudo resultando en un menor uso del ancho de banda de la red.

3.2 Segundo escenario

Luego de conocer sobre el funcionamiento de la red hICN en el primer escenario se desea conocer el comportamiento de la arquitectura al ser implementado en un dispositivo no virtualizado, en este caso en un smartphone Android.

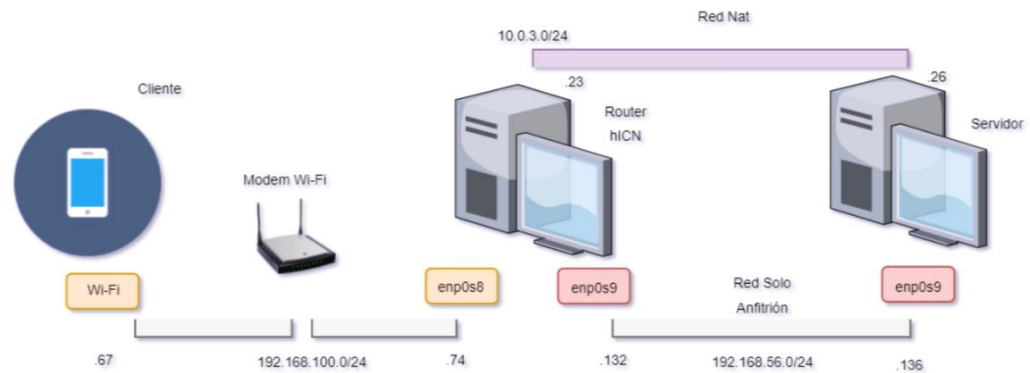


Figura 3.12: Topología del segundo escenario.

La configuración del router hICN y del servidor es la misma que en primer escenario.

Para lograr una conexión entre el smartphone y las máquinas virtuales, como se observa en la figura 3.6 el router hICN en su interfaz enp0s8 se le asignó una red de adaptador fuente haciendo posible estar en el mismo segmento de red al estar conectado al mismo módem Wi-Fi.

La pila hICN cuenta con herramientas para los smartphones Android como se indica en la sección 2.5 que nos permiten analizar el comportamiento de una manera más sencilla sin tener que estar familiarizado con el manejo comandos de una consola.



Figura 3.13: Resultados de la prueba Hiperf en la aplicación móvil.

Para realizar la prueba es necesario conocer el prefijo hICN para indicarlo en la interfaz y poder ejecutarla con el botón start. Una ventaja con respecto a realizar la prueba en la máquina virtual es que nos ofrece una gráfica con los datos recuperados, sin embargo, al ser una versión inicial todavía no cuenta con un buen formato de visualización como se puede observar en la figura 3.7 al no indicar a que corresponde los valores del eje horizontal.

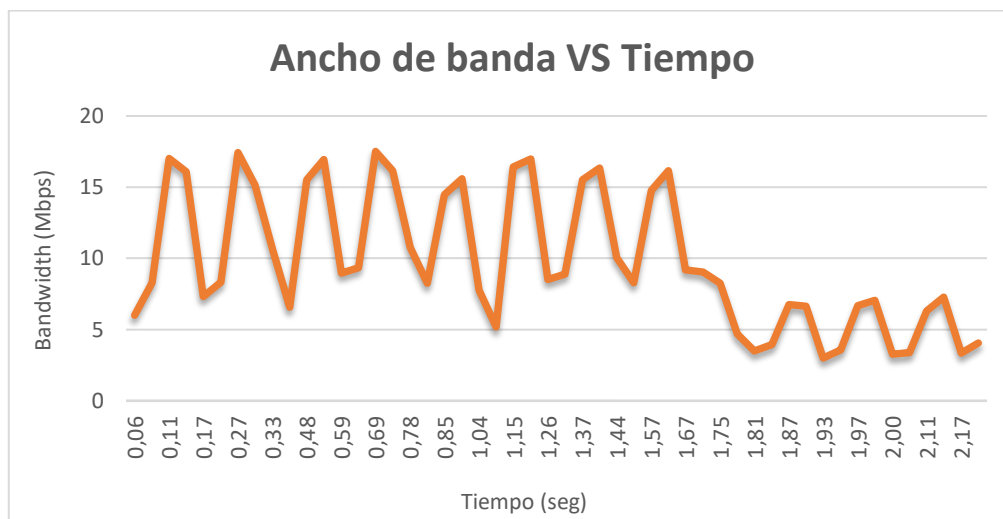


Figura 3.14: Gráfico de los datos de la prueba Hiperf en el Smartphone.

La figura 3.8 presenta el mismo uso de ancho de banda alrededor de 18 Mbps, pero teniendo variaciones más frecuentes con respecto a la figura 3.5, esto puede ser debido a las transiciones entre un dispositivo físico hacia el entorno virtualizado, otro factor que se puede tomar en cuenta son las limitaciones que tienen las aplicaciones móviles. Faltan desarrollar características del modelo hICN en las aplicaciones, como agregar opciones que permitan la configuración de direcciones IPv6 y conexiones hICN, por ahora solo está disponible las configuraciones con direcciones IPv4 y conexiones udp.

La prueba se realizó también con 50 datos, pero configurando una ventana de congestión de 200 por lo que la duración es mayor respecto a la del primer escenario en la figura 3.5 aumentando el tiempo entre cada petición.

3.3 Tercer escenario

El desarrollo de la pila hICN no se limita solo a plataformas con sistemas operativos que funcionan con arquitectura AMD64 o en el sistema operativo de Android, también tiene compatibilidad con sistemas operativos que funcionan en arquitectura ARM64. En el presente escenario se utilizó una tarjeta de desarrollo Raspberry Pi 3 B+ por su bajo costo y fácil obtención.

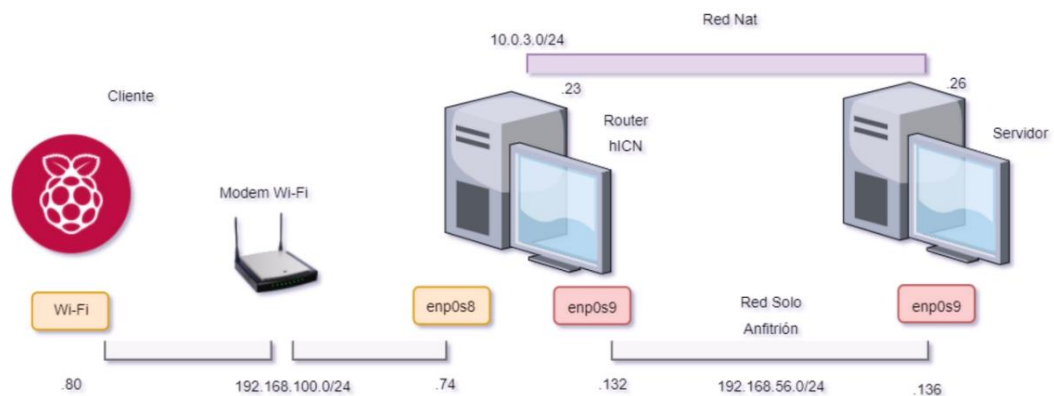


Figura 3.15: Topología del tercer escenario.

Se tiene la misma implementación de la figura 3.6 intercambiando al celular por la tarjeta de desarrollo Raspberry Pi 3 B+.

Se realizaron las mismas pruebas del primer escenario para observar el comportamiento de la arquitectura hICN en otra plataforma, comparando los resultados de las pruebas.

Al tener pocos recursos la tarjeta de desarrollo y para evitar comprometer el rendimiento de las pruebas al instalar una interfaz gráfica pesada se decidió instalar una interfaz gráfica liviana y realizar las pruebas via SSH.

```

ubuntu@ubuntu:~$ sudo hcn-ping-client -n b001::1 -V
start ping
>>> send interest b001::1|0 src port: 9695 dst port: 8080 flags: TTL: 64
<<< received object.
<<< interest name: b001::1|0 src port: 9695 dst port: 8080 flags:
<<< object name: b001::1|0 src port: 32768 dst port: 128 flags: S path label 150994944 (9) TTL: 2
64
>>> send interest b001::1|1 src port: 9695 dst port: 8080 flags: TTL: 64
<<< received object.
<<< interest name: b001::1|1 src port: 9695 dst port: 8080 flags:
<<< object name: b001::1|1 src port: 32768 dst port: 128 flags: S path label 150994944 (9) TTL: 2
64
>>> send interest b001::1|2 src port: 9695 dst port: 8080 flags: TTL: 64
<<< received object.
<<< interest name: b001::1|2 src port: 9695 dst port: 8080 flags:
<<< object name: b001::1|2 src port: 32768 dst port: 128 flags: S path label 150994944 (9) TTL: 2
64

```

Figura 3.36: Ejecución de la aplicación de ping hICN en la tarjeta de desarrollo.

Los resultados presentados en la figura 3.10 son similares a los presentados en la figura 3.2 al determinar que existe una conexión entre los extremos de la red local.

```

ubuntu@ubuntu:~$ sudo hperfb -C b001::1 -W 200
Starting download of b001::1|0
Interval      Transfer      Bandwidth      Retr      Cwnd
  AvgRtt
0-1          1.870[MBytes]  14.948[Mbps]    0         200[Interest]
  148387[us]

Interval      Transfer      Bandwidth      Retr      Cwnd
  AvgRtt
1-2          1.890[MBytes]  15.104[Mbps]    0         200[Interest]
  147903[us]

Interval      Transfer      Bandwidth      Retr      Cwnd
  AvgRtt
2-3          1.880[MBytes]  15.041[Mbps]    0         200[Interest]
  147598[us]

Interval      Transfer      Bandwidth      Retr      Cwnd
  AvgRtt
3-4          1.891[MBytes]  15.116[Mbps]    0         200[Interest]
  148993[us]

```

Figura 3.47: Resultados de la prueba Hiperfb en la tarjeta de desarrollo.

Luego de confirmar que existe una conexión entre los extremos, se realizó la prueba de rendimiento Hiperfb indicando una ventana de congestión de 200 con la opción “-W” obteniendo los resultados observados en la figura 3.11 que serán exportados para realizar una gráfica para una mejor visualización.

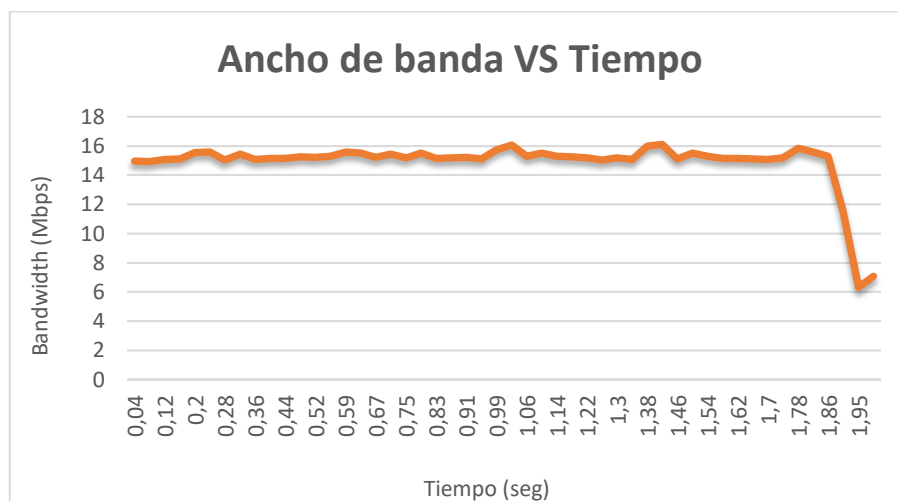


Figura 3.58: Gráfico de los datos de la prueba Hiperf en la tarjeta de desarrollo.

Se puede observar que el máximo ancho de banda utilizado este alrededor de 15 Mbps, siendo menor entre los datos presentados en los primeros escenarios. También que el uso del ancho de banda es más estable y que a pesar de usar también solo 50 datos con una ventana de congestión de 200, tiene una menor duración que la prueba realizada en la figura 3.8.

Los resultados nos indican una mejor integración en los sistemas operativos para ARM64 que en el sistema operativo que maneja Android.

3.4 Análisis de costos

Para el presente proyecto solo se consideran los costos de comprar los dispositivos o recursos necesarios para poder implementar los escenarios indicados en las secciones 3.1, 3.2 y 3.3, debido a que los objetivos no toman en consideración ofrecer o brindar algún servicio de consultas que necesite de personas o cualquier otro gasto, el enfoque es solo comprender el funcionamiento y comparar las ventajas y debilidades de las redes hICN con respecto a las redes existentes.

3.4.1 Virtualización de máquinas

Se utilizó una PC de escritorio para realizar la virtualización de las 3 máquinas debido a que es más fácil realizar una actualización de los recursos en caso de ser necesario. Las máquinas virtuales son utilizadas en los 3 escenarios.

Es complicado indicar algo específico sobre cuantos recursos consumirá la pila hICN debido a que su desarrollo esta en una etapa inicial y no se tiene conocimiento ni esta detallado en algún lado esta información. Teniendo en cuenta estas consideraciones lo que se detalla a continuación son los recursos necesarios solo para virtualizar de una manera fluida 3 máquinas para poder desplegar la red hICN básica.

La PC de escritorio utilizada fue comprada por partes por lo cual se realizó una búsqueda de una computadora para estimar el precio y obtener un precio referencial de aproximadamente 650 dólares[16]. A este precio se le debe agregar el de una memoria Random Access Memory (RAM) de 8gb con un precio aproximadamente de 33 dólares[17], debido a que la computadora utilizada tiene 16 GB de memoria RAM.

3.4.2 Implementación en una Raspberry Pi 3 B+

Se eligió la tarjeta de desarrollo Raspberry Pi 3B+ por dos sencillas razones por su compatibilidad con el software y hardware necesario para el funcionamiento de la pila hICN. Con un precio referencial de aproximadamente 75 dólares[18], relativamente bajo con respecto a otras tarjetas con más recursos computacionales.

Luego de la realización de las pruebas, la tarjeta obtuvo un rendimiento adecuado, pero no está de más indicar que si se tiene a disposición otra tarjeta que cumpla con la arquitectura ARM64 y con más recursos no existe algún problema en usarla, en este caso no hubo necesidad alguna.

3.4.3 Implementación en un Smartphone

Para utilizar esta arquitectura es necesario tener un dispositivo con Android mayor a 8.0, y un dispositivo con estos requerimientos cuesta alrededor de 150 dólares.

Para poder utilizar la pila hICN en un smartphone es necesario que el mismo tenga una versión mayor o igual a Android 8.0. Se realizó una búsqueda con relación calidad precio y se determinó que el Xiaomi Redmi 9 con un precio referencial de aproximadamente 143 dólares[19], cumple con los requisitos necesarios.

3.4.4 Resumen de costos

Tabla 3.1: Costos Generales

Escenarios	Precio (\$)
Virtualización de Máquinas	683
Raspberry Pi 3 B+	75
Smartphone	143
Total	901

Se tiene que la inversión para replicar este trabajo y poder realizar pruebas para conocer sobre el funcionamiento de la red hICN estaría alrededor de los 901 dólares. Es necesario mencionar y recalcar que el desarrollo y los avances presentados son de una etapa temprana y que los requerimientos pueden variar al pasar el tiempo.

Algo que se debe tener en cuenta y que debe quedar muy claro es que el proyecto que se esta desarrollando por Fast Data no es algo que una persona normal vaya a implementar a gran escala. Es un proyecto que tiene un enfoque mucho mas grande al querer revolucionar y cambiar un paradigma para que pueda ser compatible y eficiente con las tecnologías futuras que requieran un procesamiento de información masivo, y es ahí donde realmente se puede observar lo viable del proyecto. Con estas consideraciones estimar un costo de la implementación de algo así no es posible a estas alturas.

CAPÍTULO 4

4. CONCLUSIONES Y RECOMENDACIONES

Conclusiones

1. Los desarrolladores del proyecto trabajan en la compatibilidad con diferentes plataformas para que la transición a esta nueva arquitectura requiera cambios mínimos.
2. La reducción de tráfico servido permite el ahorro de recursos de red y de cómputo, lo cual implica un ahorro económico a futuro.
3. La red híbrida permite utilizar las características de las redes centradas en la información con la arquitectura de red actual logrando una buena integración.
4. Para obtener una mejora considerable respecto a consumo de memoria o tráfico servido, se debe habilitar nodos hICN en puntos finales en conjunto con nodos en el borde de la red.
5. hICN es un proyecto muy sólido de acuerdo con la extensa documentación sobre su funcionamiento de una manera muy técnica, hoy en día 3 de febrero del 2021, sigue en desarrollo el software que permite la administración de estos nuevos paquetes, contando con características básicas para todo lo que sería su implementación final.
6. Al tratarse de un nuevo paradigma sobre cómo se distribuyen los paquetes de contenido por la red, hubo dificultades al realizar la implementación para entender su funcionamiento debido a que la información disponible no estaba actualizada, y existían muchos cambios entre las versiones actualizadas y las descritas en la página web del proyecto.

7. Si bien es cierto que están disponibles los binarios de la pila hICN para realizar pruebas, estos aún no tienen todas las características desarrolladas y están sujetos a actualizaciones frecuentes, por ejemplo, los comandos utilizados para su configuración.
8. En los distintos escenarios implementados, ocurrieron algunos errores al momento de realizar las pruebas de rendimiento debido a las versiones disponibles de la pila hICN, pero se lograron obtener suficientes datos para realizar gráficas y así poder observar el rendimiento del ancho de banda de la conexión.
9. Los resultados de las pruebas realizadas por CISCO demuestran el potencial que tiene este tipo de arquitectura, no solo por el hecho de utilizar la infraestructura IP actual completa, pero también por poder aprovechar el modelo de las redes centradas en la información sobre IP, evitando tener que cambiar desde cero el núcleo de la red para poder implementarse.
10. Al realizar las pruebas con sistemas basados en arquitectura ARMx64, particularmente la Raspberry Pi 3 B+, se pudo constatar que el funcionamiento es idéntico que en su versión para sistemas con arquitectura AMDx64. Para sus versiones en dispositivos móviles la instalación fue más sencilla, pero tiene características más limitadas.

Recomendaciones

1. Para más información sobre el funcionamiento del código base del desarrollo de hICN se puede consultar la página [hicn-dev@lists.fd.io | Home](mailto:hicn-dev@lists.fd.io). Este sitio fue de mucha ayuda durante la investigación del proyecto al ofrecer respuestas directamente de los desarrolladores.
2. Para consultar información sobre el proyecto Fast Data se puede ingresar a la página [HICN - fd.io](https://fd.io/hicn), aquí se encuentran varios proyectos todavía vigentes y también culminados.
3. En la página [Getting started — Hybrid ICN 20.01 documentation \(hicn.readthedocs.io\)](https://hicn.readthedocs.io) se irá actualizando la información detallada sobre cada binario y su funcionamiento a medida que se avance en el desarrollo de la tecnología.
4. Si se desea realizar pruebas, lo mejor es procurar trabajar con las versiones más actualizadas, mismas que se las puede encontrar en la página [fdio/hicn - Packages - packagecloud](https://fd.io/hicn-Packages-packagecloud). En caso de tener alguna duda al respecto, referirse a la página descrita en la recomendación número 1.

BIBLIOGRAFÍA

- [1] M. Sardara, “Towards a scalable and programmable incremental deployment of ICN in the real world To cite this version :,” 2020.
- [2] U. Carofilis, “Estudio para la migración del protocolo IPv4 al protocolo IPv6. Caso de estudio Plenario de la Asamblea Nacional,” p. 104, 2017, [Online]. Available: [http://repositorio.puce.edu.ec/bitstream/handle/22000/14002/Tesis Ulises Carofilis capt 1-2-3-4 UV.pdf?sequence=1&isAllowed=y](http://repositorio.puce.edu.ec/bitstream/handle/22000/14002/Tesis%20Ulises%20Carofilis%20capt%201-2-3-4%20UV.pdf?sequence=1&isAllowed=y).
- [3] “The role of technology in powering an Inclusive Future.”
- [4] “fd.io.” https://wiki.fd.io/view/Main_Page (accessed Nov. 03, 2020).
- [5] “Cicn - fd.io.” <https://wiki.fd.io/view/Cicn> (accessed Nov. 03, 2020).
- [6] S. Jordan, Paola; Gastezzi, Adriana; Sarmiento, “Despliegue de una plataforma de demostración del encaminamiento basado en contenidos.,” pp. 1–68.
- [7] M. Papalini and M. Sardara, “draft-muscariello-intarea-hicn-04 - Hybrid Information-Centric Networking,” 2020. Accessed: Dec. 05, 2020. [Online]. Available: <https://trustee.ietf.org/>.
- [8] “FW: [hICN] Questions hicn-light forwarder.” <https://lists.fd.io/g/hicn-dev/message/31> (accessed Dec. 05, 2020).
- [9] “Core library — Hybrid ICN 20.01 documentation.” <https://hicn.readthedocs.io/en/latest/lib.html> (accessed Dec. 03, 2020).
- [10] “Transport library — Hybrid ICN 20.01 documentation.” <https://hicn.readthedocs.io/en/latest/transport.html> (accessed Dec. 03, 2020).
- [11] “Portable forwarder — Hybrid ICN 20.01 documentation.” <https://hicn.readthedocs.io/en/latest/hicn-light.html> (accessed Dec. 03, 2020).
- [12] “Utility applications — Hybrid ICN 20.01 documentation.” <https://hicn.readthedocs.io/en/latest/utills.html> (accessed Dec. 03, 2020).
- [13] “Applications — Hybrid ICN 20.01 documentation.” <https://hicn.readthedocs.io/en/latest/apps.html> (accessed Dec. 03, 2020).
- [14] G. Peng, “CDN: Content Distribution Network,” pp. 1–26, 2004, [Online]. Available: <http://arxiv.org/abs/cs/0411069>.

- [15] G. Carofiglio, L. Muscariello, J. Augé, M. Papalini, M. Sardara, and A. Compagno, "Enabling ICN in the internet protocol: Analysis and evaluation of the hybrid-ICN architecture," *ICN 2019 - Proc. 2019 Conf. Information-Centric Netw.*, pp. 55–66, 2019, doi: 10.1145/3357150.3357394.
- [16] "Amazon.com: NEXSMART Ordenador de sobremesa todo en uno con Intel i7 y Windows 10 preinstalado, 8 GB DDR3 480 SSD compatible con WiFi y 23,8 pulgadas 1920 x 1080 FHD (táctil): Electronics." https://www.amazon.com/-/es/NEXSMART-Ordenador-sobremesa-preinstalado-compatible/dp/B08PYTF2HZ/ref=sr_1_12?__mk_es_US=ÅMÅŽÕÑ&dchild=1&keywords=escritorio+i3+octava&qid=1612824628&s=electronics&sr=1-12 (accessed Feb. 08, 2021).
- [17] "Amazon.com: v-Color 8 GB (1 x 8 GB) DDR4 DRAM 2666MHz (PC4-21300) CL19 1.2V U-DIMM Módulo de actualización de RAM de escritorio (TD48G26S819-VC): Computers & Accessories." https://www.amazon.com/-/es/v-Color-PC4-21300-actualización-escritorio-TD48G26S819-VC/dp/B07YP6CW8P/ref=sr_1_3?__mk_es_US=ÅMÅŽÕÑ&dchild=1&keywords=8gb+ram&qid=1612824818&sr=8-3 (accessed Feb. 08, 2021).
- [18] "Amazon.com: Canakit Raspberry Pi 3 Kit principiante completo–32 GB Edition: Computers & Accessories." <https://www.amazon.com/-/es/Canakit-Raspberry-Kit-principiante-completo-32/dp/B01C6Q2GSY> (accessed Feb. 08, 2021).
- [19] "Amazon.com: Xiaomi Redmi 9 Unlocked RAM Dual Sim 32GB 3GB RAM 6.53" International Global Version (Sunset Purple)." https://www.amazon.com/-/es/Redmi-9/dp/B089WBCZRD/ref=sr_1_3?__mk_es_US=ÅMÅŽÕÑ&dchild=1&keywords=redmi+9&qid=1612826097&sr=8-3 (accessed Feb. 08, 2021).