

## Automatización del proceso de simulación para el análisis de rendimiento del protocolo TCP en redes inalámbricas

### Simulation process automation for TCP performance analysis in wireless networks

Franco A. Trinidad, franco.unpa@protonmail.com  
Claudia Gonzalez, lic\_claudiagonzalez@hotmail.com  
Carlos A. Talay, catedraunpa@yahoo.com.ar  
Diego Rodriguez Herlein, diegorh@gmail.com

Rectorado - Universidad Nacional de la Patagonia Austral  
Av. Lisandro de la Torre 860 - Río Gallegos - Santa Cruz - Argentina

Recibido: 23/02/2018. Aceptado: 14/11/2018

### RESUMEN

Al trabajar con distintas herramientas de simulación de redes, uno de los inconvenientes más notables es la necesidad de llevar a cabo la generación de un modelo, la ejecución de pruebas y la obtención de resultados para cada caso de estudio. Este proceso insume mucho tiempo debido a los desafíos técnicos involucrados y sólo una vez finalizado, es posible realizar un análisis valorativo de los datos obtenidos. Para detectar casos cuyos resultados pudieran ser de interés, el proceso deberá repetirse varias veces, lo que incrementa el tiempo entre la concepción de un problema y el análisis de los resultados.

El objetivo principal de este trabajo fue reducir el tiempo requerido para implementar y probar un modelo. Esto se logró al establecer una base de simulación con parámetros relevantes en el estudio de rendimiento de TCP, para poder automatizar de forma eficiente la generación de pruebas, el procesamiento y presentación de datos. Gracias a esto se pueden obtener los resultados en menos tiempo, dando paso a la etapa de evaluación.

Como se optó por NS-2, la base de simulación está implementada en el lenguaje *OTcl*. El procesamiento de datos se hizo mediante scripts de *awk*, *perl* y *python*. Para la presentación visual se utilizó *gnuplot*. Finalmente, se usó *bash* para unir todas las partes del proceso, coordinando su ejecución, ordenando los datos y ofreciendo información relevante en cada paso.

Como resultado se logró optimizar el flujo de trabajo para el análisis de rendimiento del protocolo TCP sobre diferentes topologías, reconociendo más rápidamente aquellos escenarios con resultados prometedores. A partir de los datos obtenidos, se exploraron hipótesis que sirvieron de base para la escritura de publicaciones científicas.

**Palabras clave:** TCP; Simulación; Rendimiento; Automatización.

### ABSTRACT

Whilst working with different network simulation tools, the necessity to generate a model, execute tests and obtain results for each case study, becomes an inconvenient. Due to the technical challenges involved, this process takes up too much time and only after it's done,



it's possible to analyze the data. In order to detect cases whose results might be interesting, this process must be repeated several times, which increases the amount of time between the conception of a problem and the analysis of the results.

The main goal is to reduce the time required to implement and test a model. This was achieved establishing a simulation base with parameters which are relevant for the study of TCP performance, and through the automation of test generation, data processing and presentation.

NS-2 was used and the model implemented in Otcl. Scripts in awk, perl and python were used in data processing. Gnuplot allowed a visual presentation. Finally, bash programs merged everything together, coordinating the execution, sorting the data and offering relevant information during the process.

As a result, work flow was optimized for performance analysis of TCP in different topologies, allowing promising scenarios to be easily recognized. Some of the hypothesis derived from these cases turned into scientific publications.

**Key words:** TCP; Simulation; Performance; Automation.

## 1. INTRODUCCIÓN

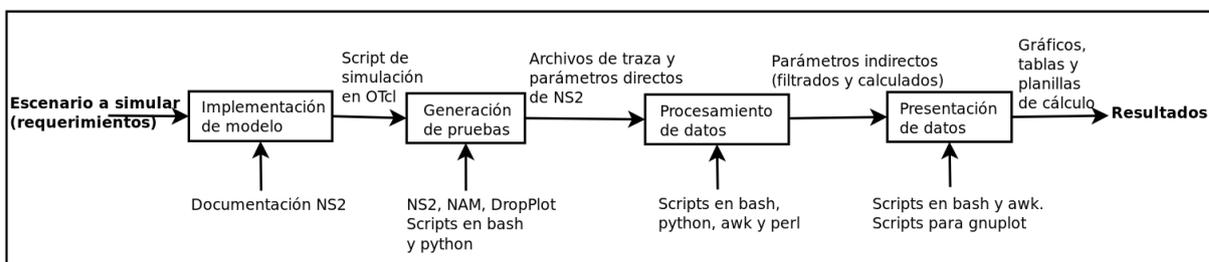
TCP es uno de los protocolos más antiguos de Internet (DARPA Internet Program, 1981). Su continuidad se debe a una serie de modificaciones propuestas para responder a fallas en el diseño original y a la aparición de nuevas tecnologías (Marrone et al, 2013). Uno de los problemas más relevantes en la actualidad consiste en mejorar su rendimiento en redes inalámbricas (Goswami et al, 2016).

El predominio de topologías cableadas en la década del '80 puso de relieve el problema de pérdida de paquetes por congestión de redes. La primera versión del estándar IEEE 802.11 (1997) dio lugar a una amplia adopción de tecnologías basadas en redes inalámbricas (WLAN) (Lorente y De Cerio, 2005). Tal crecimiento permitió reconocer las falencias del uso de TCP en ese medio (Tsaoussidis y Matta, 2001): por un lado, las pérdidas por errores de transmisión en redes inalámbricas no son despreciables; por otro lado, entran en juego factores como la interferencia de señales y las salidas del rango de cobertura (Xylomenos et al, 2001). En otras palabras, fenómenos característicos de medios inalámbricos dieron a conocer el rendimiento subóptimo de TCP en topologías de este tipo (Elaarag, 2002). Esto planteó la necesidad de modificar los algoritmos de control de congestión, entre otras cosas, para mejorarlo.

Entre las técnicas más utilizadas para el estudio de problemas en redes, se destaca el modelado de datos mediante el enfoque de simulación (Issariyakul y Hossain, 2012). Una de las herramientas más populares es Network Simulator 2 (NS-2) (Hernandez Benet y Vizcaino, 2014). NS-2 es un simulador de eventos discretos de código abierto desarrollado en 1989 que contiene módulos para simular numerosos aspectos de una red (UC Berkeley, LBL, 2011). El análisis de los datos obtenidos del simulador suele variar según el objeto de estudio y las herramientas conocidas por los investigadores.

Este trabajo surge de la necesidad de optimizar el flujo de trabajo para el análisis de rendimiento del protocolo TCP sobre diferentes topologías. Se buscó minimizar el tiempo requerido para implementar y probar un modelo, y maximizar la atención dedicada al análisis de resultados y obtención de conclusiones a partir de ellos.

El objeto de estudio es el proceso de simulación (véase Figura 1). Éste consiste en la implementación de un modelo a simular, la generación de pruebas, el procesamiento de los datos obtenidos y la presentación de los mismos. En particular, se busca determinar qué parámetros permiten evaluar el rendimiento del protocolo TCP, especialmente aquellos que sean relevantes en redes inalámbricas.



**Figura 1.** Etapas del proceso de simulación

La solución ofrecida consiste en automatizar el proceso de simulación para facilitar el estudio de distintos escenarios que involucren tráfico TCP.

El objetivo general es conformar un sistema base para la implementación de modelos de simulación. Éste debe poder adecuarse a los requerimientos de escenarios diseñados para el estudio de rendimiento de TCP con mínima modificación.

NS-2 es una herramienta compleja, por lo que el proceso de simulación puede tornarse dificultoso y extenderse más tiempo del necesario, debido a detalles técnicos que desvían la atención del fenómeno que se desea estudiar. El procesamiento de datos y la presentación de los mismos suelen hacerse por separado, lo que incrementa aún más el tiempo entre la decisión de implementar un modelo y el momento de analizar los resultados.

Los aportes de este trabajo se dividen en tres partes:

En primer lugar, se determinan los parámetros más importantes para el estudio de rendimiento de TCP.

En segundo lugar, se desarrolla una base de simulación para el estudio de TCP sobre distintas topologías utilizando NS-2 (teniendo en cuenta los parámetros relevantes).

Finalmente, se automatiza el procesamiento de datos obtenidos y se los presenta de forma tal que facilite su comprensión.

La metodología utilizada consistió en cuatro actividades principales: investigación, diseño, implementación y evaluación. Éstas se llevaron a cabo de forma iterativa e incremental para conformar un modelo que abarque todos los aspectos de interés. Una vez tomada la decisión de implementar una nueva característica o funcionalidad, se investigó si ésta estaba contemplada por las herramientas usadas; se modificó el diseño del modelo para incluirla; se implementó en el simulador; y se la probó para validar que los resultados obtenidos se correspondan con la teoría.

Existen muchas investigaciones relacionadas al tema, debido tanto a la naturaleza del problema como a la disponibilidad de las herramientas de trabajo. Las distintas topologías, la complejidad de TCP, y la variedad de comportamientos y disposición de elementos en una red, dan lugar a una gran cantidad de escenarios de simulación posible. Suele estudiarse el comportamiento de un protocolo de comunicación en distintas situaciones para detectar fenómenos inusuales o fallas a mejorar. En este trabajo se usa el mismo enfoque.

Tal como se esperaba, la posibilidad de implementar y ejecutar pruebas complejas en menos tiempo facilitó el análisis de una mayor cantidad de escenarios, y permitió priorizar aquellas líneas de trabajo con resultados prometedores.

El documento está organizado de la siguiente manera: en la sección 2, se presenta el marco de referencia (conceptos básicos de TCP y NS-2). En la sección 3 se habla de la metodología utilizada. En la sección 4 se muestran resultados obtenidos, seguidos del análisis y discusión. En la sección 5 se exponen las conclusiones obtenidas. Finalmente, en la sección 6 se ofrecen algunas recomendaciones para futuros desarrollos.

## 2. MARCO DE REFERENCIA

### 2.1. Marco histórico

#### 2.1.1. TCP y problemas en redes inalámbricas

En 1981 se presentó la propuesta del protocolo TCP (DARPA Internet Program, 1981). Aunque varias partes del protocolo han sido mejoradas desde entonces (Internet Engineering Task Force, 2015), toda nueva implementación debe ser interoperable con las anteriores. Ya que TCP es un componente crítico en Internet, una gran parte de las comunicaciones depende de su correcto funcionamiento. Las modificaciones al protocolo surgen en respuesta a fallas de diseño o al desarrollo de nuevas tecnologías que acarrear problemas técnicos inesperados. Las comunicaciones inalámbricas son un ejemplo de ello (Network Working Group, 2004).

Las redes inalámbricas conllevan una serie de inconvenientes debido a la naturaleza del medio físico de transmisión. La movilidad, el rango de cobertura y la interferencia de señales son sólo algunos de los nuevos factores a considerar cuando hay pérdida de paquetes (Pentikousis, 2001). En otras palabras, los errores en tránsito no son despreciables como en las redes cableadas. A pesar de ello, los algoritmos de control de congestión no fueron diseñados para detectar la causa de la pérdida y se disparan de todas formas. Algunas de las modificaciones mencionadas para TCP buscan remediar este problema (Network Working Group, 2007).

A finales de la década del 90, luego de la primera versión del estándar IEEE 802.11, se observó un incremento en la adopción de tecnologías inalámbricas. Al haber cada vez más dispositivos móviles conectados a redes como Internet, sólo se acentuaron los problemas del uso de TCP en topologías inalámbricas.

Protocolos de transporte como TCP fueron pensados para redes fijas. Esto hace que las pérdidas y retardos por errores en tránsito sean interpretadas como congestión, y en consecuencia, se invoquen algoritmos para controlarla (Chandran et al, 1998). Este es un problema de interés para muchos investigadores.

#### 2.1.2. El uso de NS-2 en el ámbito académico

NS-2 es un simulador pensado para el trabajo con redes que requiere un alto nivel de detalle, como por ejemplo, en el estudio de protocolos de transporte y algoritmos de control de congestión. Fue desarrollado a finales de la década del '80 en base a REAL Network Simulator y recibió el apoyo de DARPA a mediados de los '90 (Bonastre y Salvador, 2005), lo que sin duda contribuyó significativamente a su popularidad y calidad. Otro factor que contribuyó significativamente a su adopción en el ámbito académico, la industria y el gobierno, es su licencia: el código está disponible para cualquiera que quiera utilizarlo, modificarlo y compartirlo, sin costo alguno.



El soporte de NS-2 cesó en el 2010 para dar lugar a NS-3, que no es compatible con su antecesor (Proyecto NS-3, 2016). Sin embargo, estuvo bajo arduo desarrollo durante dos décadas, lo que dio lugar a numerosos módulos para componentes de red, protocolos de encaminamiento, transporte, etc.

NS-3 se encuentra en desarrollo y los módulos existentes pueden no satisfacer las necesidades de los investigadores, motivo por el cual NS-2 continúa teniendo alta demanda. Debido a esto y a la gran cantidad de documentación disponible, se eligió como herramienta de trabajo.

Por la naturaleza misma de las investigaciones, la experimentación requiere capacidades que se encuentran más allá de los límites del simulador, y durante mucho tiempo, dio lugar a su expansión. Vale notar que una consecuencia de la riqueza en funcionalidad, es la complejidad de la herramienta. Aprender a utilizarla puede consumir mucho tiempo, lo que repercute negativamente en la productividad de los investigadores durante este período.

Inevitablemente para los investigadores, utilizar NS-2 implica comprender aspectos técnicos de la arquitectura del simulador, lo que reduce la atención al problema que se desea estudiar. Resulta necesario un marco de trabajo que permita automatizar aquellas tareas cuya complejidad se interpone entre los investigadores y el objeto de estudio. El presente trabajo intenta dar respuesta a este problema.

## 2.2. Marco conceptual

Como se dijo anteriormente, en este trabajo se presenta el problema de rendimiento de TCP inherente a su uso en redes inalámbricas. En su estudio se utilizan conceptos asociados a redes de computadoras, protocolos de comunicación y modelado de datos con enfoque de simulación.

El marco conceptual está dividido en tres partes bien diferenciadas por su carácter:

### 2.2.1. Redes de computadoras

Según Stallings (2000), una red es un conjunto de nodos conectados mediante enlaces. Estos nodos pueden ser celulares, notebooks, impresoras, o cualquier otro dispositivo capaz de enviar o recibir datos.

El rendimiento de una red puede medirse de diferentes formas, entre las que se destacan el tiempo que tarda un mensaje en viajar desde un dispositivo a otro, y el tiempo entre una solicitud y una respuesta. Entre las métricas más comunes, Forouzan (2007) destaca Throughput y Delay. Se busca maximizar el primero y minimizar el segundo.

Los parámetros más comunes en el estudio de rendimiento de redes según algunos autores (Valiska et al, 2012) son:

- *Throughput*: Indica la tasa de envío o recepción de paquetes de una red. Se mide típicamente en bits por segundo.
- *End-to-End Delay*: El retraso de extremo a extremo se refiere al tiempo que un paquete tarda en ser transmitido a través de una red, de una punta a la otra. Suele medirse en segundos.
- *Jitter*: Para evitar confusiones, suele llamarse Packet Delay Variation y se refiere a la fluctuación del retraso de extremo a extremo.
- *Packet Loss*: Es la cantidad de tráfico que no llega a destino en tiempo aceptable.

Uno de los problemas más comunes que afecta el rendimiento de las redes es la congestión. Ésta suele deberse a dos motivos (Torkey et al, 2008): la velocidad a la que se transmiten los paquetes es mayor a la velocidad de procesamiento de la red; o el ancho de banda resulta insuficiente para soportar la cantidad de paquetes enviados al mismo tiempo.

Un estudio de la Universidad de Statson (Elaarag, 2002) destaca algunos de los desafíos en redes inalámbricas:

- Alta tasa de errores en bits que lleva a la pérdida de paquetes o ACKs.
- Desconexiones debidas a fenómenos de handoff (cuando un nodo móvil se mueve de una celda a otra y se conecta a una nueva estación base); cuando un nodo se encuentra fuera del área de cobertura; cuando las señales son bloqueadas por objetos o edificios; cuando el ancho de banda es insuficiente para servir a la cantidad de usuarios conectados a una misma celda.
- Ancho de banda limitado y variable, dependiente de la ubicación y el número de usuarios existentes.
- Tamaño de la celda, que en caso de estar mal diseñada puede llevar a desconexiones constantes.
- Reducción del tiempo de transmisión y recepción, teniendo en cuenta que la mayoría de los dispositivos móviles funcionan a batería.
- El movimiento contante de los nodos conduce a una topología de red dinámica.

### **2.2.2. Transmission Control Protocol**

*Transmission Control Protocol* (TCP) es un protocolo de la capa de transporte, orientado a la conexión, seguro y confiable para la comunicación. Es uno de los más utilizados en Internet, y sirve de soporte para una amplia variedad de aplicaciones (Bonaventure, 2001): e-mail (SMTP, POP, IMAP), World Wide Web (HTTP), protocolos de transferencia de archivos (FTP), acceso remotos (telnet, ssh, VNC), etc.

Gran parte de su popularidad se debe a que permite transmitir datos de forma completa y ordenada, mediante mecanismos de control de flujo y recuperación de errores. Antes de transmitir, TCP establece una conexión creando un circuito lógico de punta a punta, sin asumir confiabilidad en los protocolos de capas más bajas como IP. Es decir, asume la responsabilidad de asegurar la integridad del mensaje (Parziale et al, 2006).

Además de los parámetros de rendimiento mencionados para redes inalámbricas, en el estudio de TCP deben considerarse:

- Round Trip Time (RTT): es el tiempo que pasa desde que se envía una señal hasta que se recibe una respuesta. Puede pensarse como el tiempo entre que se envía SYN y se recibe el ACK correspondiente.
- Retransmission Timeout: es el tiempo que se espera para recibir un ACK antes de reenviar un paquete.

Uno de los rasgos distintivos de TCP son los algoritmos de control de congestión. Estos previenen que el nodo fuente sobrepase la capacidad de la red, adaptando la tasa de envío, lo que evita potenciales situaciones de congestión.

Las variantes de TCP se diferencian entre ellas en base a su estrategia de control de congestión (Sharma, 2012). A grandes rasgos se clasifican en variantes basadas en pérdidas y variantes basadas en retardos. Las primeras usan la pérdida de paquetes como factor de ajuste

de la ventana de congestión, como Reno y Tahoe; las segundas son menos agresivas, y buscan evitar la congestión manteniendo una ventana y un throughput constante, como Vegas.

Para implementar TCP en redes móviles deben considerarse (Elaarag, 2002):

- Retardos y pérdidas de paquetes no causados por congestión.
- Desconexiones frecuentes causadas por tiempos muertos en el emisor (cuando el tiempo de retransmisión se duplica con cada intento de transmisión erróneo). TCP puede tardar mucho tiempo en restablecer la conexión, durante el cual no se transmiten datos (Dalal et al, 2001).
- Variación en el tamaño de los paquetes. Los paquetes en transmisiones inalámbricas suelen ser mucho menores que en redes cableadas. Encontrar el tamaño óptimo es fundamental para el rendimiento de la red.

### 2.2.3. Network Simulator 2

Network Simulator 2 (NS-2) es un simulador de eventos discretos (véase marco teórico), orientado a objetos, de código abierto, escrito en C++ y con un intérprete en OTcl. Utiliza dos lenguajes debido a las tareas que debe realizar: por un lado, C++ permite la manipulación de bytes, encabezados, y la implementación de algoritmos que se ejecuten sobre grandes cantidades de datos, lo que resulta conveniente en la programación de protocolos; por otro lado, al ser un lenguaje interpretado, OTcl permite variar los parámetros de configuración fácilmente para explorar múltiples escenarios en poco tiempo.

Cabe aclarar que NS-2 permite simular varios tipos de tráfico, pero este trabajo está centrado únicamente en TCP.

Para establecer un tráfico entre dos nodos, ya sea entre cableado-cableado, cableado-inalámbrico, inalámbrico-inalámbrico, los pasos son similares: debe crearse un objeto<sup>1</sup> llamado “TCP Agent” encargado de enviar paquetes de TCP y uno denominado “TCP Sink” capaz de recibirlos; éstos son asociados a nodo fuente y destino respectivamente. La vinculación entre ellos define el sentido de tráfico TCP. Finalmente, se crea un agente de aplicación FTP (típico para TCP en NS-2) en donde se origina el tráfico, que luego es empaquetado por el agente de TCP para ser enviado.

## 2.3. Marco teórico

Para estudiar un fenómeno determinado, típicamente se recurre al modelado de sistemas. Esto quiere decir que en lugar de tener que lidiar con la complejidad de un sistema real, se puede utilizar una versión simplificada del mismo que permita observar con mayor facilidad los aspectos de interés. Para modelar un sistema deben hacerse ciertas suposiciones sobre la naturaleza del mismo, que deben tenerse en cuenta a la hora de inferir conclusiones en base a los resultados obtenidos: un modelo demasiado simplificado puede llevar a conclusiones poco precisas.

Existen dos enfoques tradicionales para el modelado de sistemas: el analítico y el de simulación.

El enfoque analítico consiste en describir un sistema en términos matemáticos y aplicar métodos numéricos para estudiar el comportamiento del mismo al alterar distintas variables.

---

1 Las consideraciones de diseño orientado a objetos utilizadas en NS-2 escapan a los objetivos de este trabajo. Issariyajul y Hossain logran sintetizarlas adecuadamente en su introducción al simulador [8].

Este enfoque resulta ineficiente para sistemas demasiado complejos como es el caso de redes de computadoras. Para crear un modelo matemático de una red de computadoras, debe simplificarse el sistema al punto que se pone en duda la precisión de las inferencias que se puedan obtener.

El enfoque de simulación permite recrear escenarios del mundo real mediante programas de computadoras. De esta forma se pueden estudiar sistemas complejos con menos suposiciones, ya que la mayoría de los factores que influyen en su comportamiento pueden ser incorporados en el modelo de simulación.

Según el manejo de tiempo, las herramientas de simulación se dividen en dos grupos: manejadas por el tiempo, donde se utilizan intervalos fijos y los eventos se acomodan al principio o al final de los mismos; y manejadas por eventos, que ejecutan eventos en intervalos de tiempo arbitrarios.

NS-2 pertenece al segundo grupo. Los simuladores manejados por eventos tienen cuatro características importantes:

- Cada evento tiene una etiqueta de tiempo y es almacenado en una lista de eventos.
- La simulación procede mediante la ejecución de los eventos en dicha lista, haciendo que avance el reloj.
- Durante la ejecución, un evento puede desencadenar otros.
- La simulación continúa hasta finalizar la lista o hasta que se cumpla alguna otra condición.

Muchos estudios de rendimiento de TCP han sido realizados mediante simulaciones (Allman y Falk, 1999). Esto se debe a las ventajas que conlleva:

- Los simuladores no requieren equipos costosos. Una estación de trabajo basta para ejecutar las simulaciones y analizar los datos.
- Mediante simuladores se puede examinar una amplia variedad de escenarios en poco tiempo.
- Se puede simular el comportamiento de TCP en redes no convencionales que resultarían difíciles de acceder en el mundo real.
- Las limitaciones físicas no se aplican al simulador, por lo que se puede trabajar con una velocidad esperada en el futuro para anticipar posibles problemas.
- En los simuladores se pueden modificar fácilmente los protocolos de ruteo y políticas de colas.

Sin embargo, también deben considerarse las desventajas de este enfoque (Pandey y Tyagi, 2013):

- Las implementaciones de TCP utilizadas en los simuladores pueden ser abstractas y no coincidir necesariamente con las que se encuentran en los sistemas operativos modernos.
- Los simuladores no suelen considerar eventos externos a la red que pueden llegar a influir en el rendimiento de TCP. Por ejemplo, un retardo en el planificador de tareas del sistema operativo puede afectar la velocidad de la red.
- Un simulador trabaja con condiciones ideales, no replica el mundo real.

- Como todos los programas de computadoras, NS-2 puede tener bugs. Estos pueden causar resultados inesperados, inhabilitar alguna funcionalidad o impedir la correcta ejecución de la simulación.



### 3. METODOLOGÍA

#### 3.1. Materiales y métodos

Como punto de partida para el trabajo se tomaron como referencia: publicaciones acerca del estado del arte en el estudio de rendimiento de TCP (en particular aquellos que utilizan una metodología similar); libros de textos sobre redes de computadoras; manuales, tutoriales y foros que abordan problemas típicos en el uso de NS-2.

Antes de implementar cualquier modelo en un simulador, es conveniente entender el origen y la naturaleza del problema a resolver, los aspectos a estudiar y el objetivo de la simulación. Comprender el problema da una idea general acerca del tipo de escenario que debe plantearse para abordarlo. Los aspectos a estudiar determinan las variables a ajustar y los parámetros a obtener para el procesamiento de datos posterior. El objetivo de la simulación servirá para estructurar y ordenar los datos obtenidos en una presentación que resulte conveniente para el análisis y obtención de conclusiones.

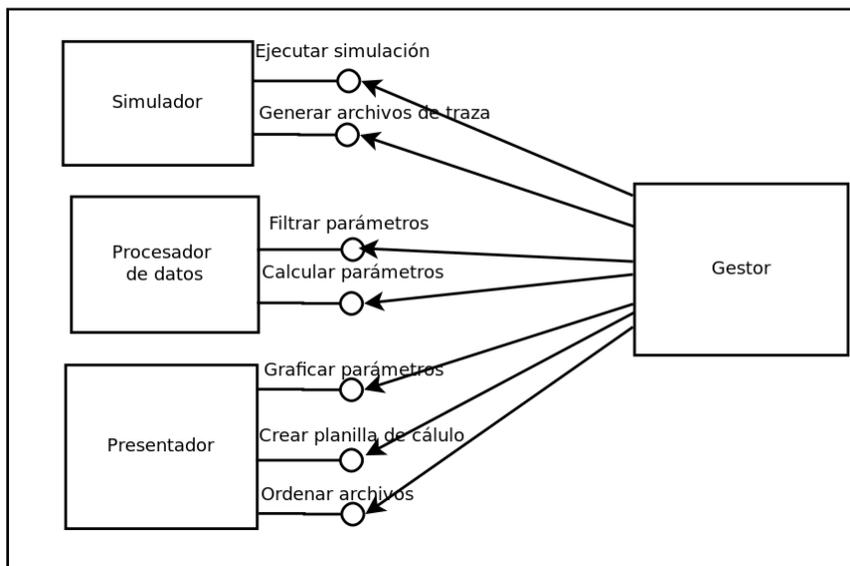
Las herramientas utilizadas durante el trabajo fueron:

- Network Simulator 2
- Network Animator
- Gnuplot y PlotDrop
- Tracegraph
- Bash, python, AWK, perl y Otcl
- CentOS 7 (distribución de GNU/Linux)

Hay cuatro actividades principales que predominan en la metodología utilizada: investigación, diseño, implementación, y evaluación. Éstas se llevaron a cabo de forma iterativa e incremental para conformar un modelo que abarque todos los aspectos de interés.

Durante la **investigación** se determinan los alcances y limitaciones de las herramientas utilizadas. Para ello, se acude a la documentación de las herramientas utilizadas, a publicaciones sobre el estado del arte acerca de estudio de rendimiento de TCP y a libros de texto sobre redes de computadoras. Adicionalmente, se pueden consultar listas de correos, foros y tutoriales para disipar dudas acerca de errores comunes, o para entender detalles sobre cómo implementar funcionalidades atípicas que no estén cubiertas adecuadamente en la documentación.

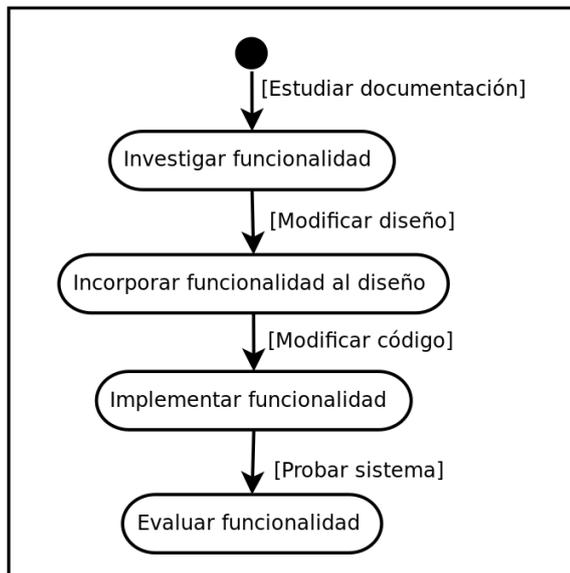
En el **diseño** se definen los componentes del sistema, las funciones que cumple cada uno y las relaciones entre ellos. Los componentes se organizan según el rol que ocupan en el proceso de simulación. En base a ello, como puede verse en la Figura 2, se determinan los siguientes subsistemas: gestor, simulador, procesador de datos, presentador de datos. El gestor contiene los componentes que coordinan la ejecución y presentan información relevante durante la misma; los componentes del subsistema simulador definen las entradas y salidas de la simulación, así como las variables que permiten adaptar el modelo base a los requisitos de uno específico; el subsistema para el procesamiento de datos está compuesto por scripts que obtienen los parámetros de rendimiento a partir de los archivos de traza generados por el simulador, ya sea mediante cálculos o filtrado de datos; el presentador grafica y ordena los resultados.



**Figura 2.** Componentes del modelo base

La **implementación** consiste en la escritura y/o modificación de código. Esto puede ser en un script de OTcl para ejecutar en NS2, un script de AWK, perl o python para filtrar datos en un archivo de traza o calcular algún parámetro a partir de datos existentes, o un script para generar un gráfico en gnuplot, o en bash para coordinar nuevas acciones.

En la **evaluación** se pone a prueba la funcionalidad implementada y se validan los resultados obtenidos contra los esperados a partir de la teoría.



**Figura 3.** Actividades realizadas para expandir o adaptar el modelo base.

Para comprender mejor la metodología utilizada, es necesario comprender cómo estas actividades se aplicaron en tres circunstancias diferentes: metodología de creación; metodología de extensión; metodología de adaptación.

### 3.1.1. Creación del modelo

La metodología de creación del modelo contempla los pasos iniciales para establecer la base de simulación.

La investigación inicial fue acerca de los conceptos y funcionalidades básicas de NS-2, y el lenguaje de programación Otcl. Se estudiaron textos introductorios al simulador, manuales, tutoriales, foros y listas de correo.

Tomando como objetivo el estudio de rendimiento de TCP sobre diferentes topologías, se crearon scripts en Otcl para simular redes cableadas, redes móviles y redes híbridas en NS-2. En ellos se podían ajustar variables que definían tipo de tráfico, cantidad de nodos, formato de los archivos de salida, etc. Se creó una función genérica para el tráfico TCP para poder definir fácilmente la cantidad de paquetes enviados, por tamaño o por tiempo, nodo fuente y destino, variante y tamaño de cada paquete, entre otras cosas.

Algunos de los parámetros útiles para el análisis de rendimiento podían obtenerse directamente del simulador, al igual que los archivos de traza. Entre ellos se encuentran la Ventana de congestión, Round Trip Time, Retransmission Time Out, y número de secuencia. Se buscaron ejemplos de scripts para procesar distintos tipos de archivos de traza. En base a ellos y a los manuales de NS-2, se crearon otros en awk, pelr y python para filtrar y calcular: throughput, goodput, delay, packet loss ratio, packet delivery ratio, y jitter. Luego se escribieron scripts que permitieran una representación gráfica de los datos procesados.

En resumen, el primer sistema consistió en tres modelos básicos, uno para cada tipo de topología. El procesamiento de datos y las pruebas se realizaron por separado, de forma manual. Al verificar que funcionaba, se generó un script en bash para automatizar el proceso.

### **3.1.2. Extensión del modelo**

La metodología para extender el modelo describe las actividades necesarias para incorporar nuevas funcionalidades.

Los requerimientos para simular un escenario particular pueden descomponerse en características o en funcionalidades a implementar. La viabilidad de dicha implementación puede verificarse al ver si se encuentra contemplada por la herramienta. Si es posible, se modifica el diseño y el código acorde. Finalmente, se realiza una prueba con cada topología para verificar que la nueva versión funcione adecuadamente y los resultados sean coherentes con la teoría (por ejemplo, que el orden de magnitud de un dato se encuentre dentro de los parámetros esperados).

Algunas de los aspectos que se incorporaron o modificaron en orden cronológico:

- Funciones genéricas que simplifiquen la adición de tráfico y la obtención de parámetros de rendimiento directamente del simulador.
- Capacidad de filtrar parámetros de rendimiento a partir de archivos de traza.
- Capacidad de calcular parámetros de rendimiento a partir de archivos de traza.
- Posibilidad de graficar parámetros de rendimiento a partir de archivos de traza.
- Estructuras de datos que faciliten la creación de enlaces de forma escalable, la elección de variantes de TCP.
- Capacidad de definir ancho de banda entre nodos móviles.
- Nombres de variables más representativos de su función, coherentes a través de funciones y archivos.
- Capacidad de agregar o eliminar nodos cableados o inalámbricos sin modificar script de simulación.

- Posibilidad de guardar valores específicos de cada simulación en una planilla de cálculo durante la ejecución de las pruebas.
- Capacidad de introducir errores estadísticos o en ráfaga, así como definir el valor de los mismos, sin modificar el script de simulación.
- Capacidad de definir parámetros alfa y beta para variable TCP Vegas.

### 3.1.3. Adaptación del modelo

La metodología de adaptación del modelo es la más importante, ya que permite validar la utilidad del modelo para implementar escenarios complejos con pocas modificaciones requeridas.

En este caso, la investigación está más enfocada en TCP. Se asume que las funcionalidades necesarias para satisfacer los requerimientos ya están incorporadas en el modelo base, y sólo deben ajustarse los valores utilizados. Lo que se busca es una estimación teórica para los valores que se deben obtener en la prueba y/o posibles explicaciones del comportamiento estudiado.

Si el sistema carece de lo necesario para satisfacer los requisitos de la prueba, se estudia la posibilidad de modificarlo o la necesidad de volver a diseñarla en base a las funcionalidades disponibles.

## 3.2. Escenarios estudiados

Se presentan dos casos que incentivaron la incorporación de nuevas funcionalidades así como la adaptación del modelo. A continuación se exponen los requerimientos de cada uno. En la sección de resultado se muestran las diferencias entre cada uno de ellos y el modelo base. En la discusión se abordan qué cambios fueron necesarios para satisfacer los requisitos.

**Escenario 1:** Consideraciones sobre el comportamiento del protocolo TCP en sus variantes Vegas, Reno, Cubic y Westwood ante errores en ráfaga en una topología híbrida.

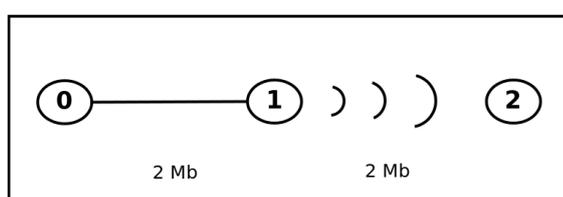
El escenario consiste en una topología híbrida de tres nodos.

Los nodos 0 y 1 debe estar vinculados por un enlace cableado de tipo dúplex, con un ancho de banda 2 Mb/s, retardo de propagación 2 ms, y política de servicio de las colas DropTail.

El enlace entre los nodos 1 y 2 debe ser inalámbrico, con modo de propagación TwoRayGround, la capa física WirelessPhy, MAC 802.11, la antena OmniAntenna y el nodo inalámbrico se encuentra fijo.

Se considera que la pérdida de paquetes se debe únicamente a los errores en ráfaga introducidos. Se utiliza una longitud variable de 0, 5, 10, 15, 20 y 25 paquetes.

Se deben realizar pruebas para Vegas, Reno, Cubic y Westwood para cada una de las longitudes de errores en ráfaga definidas.



**Figura 4.** Esquema del escenario 1.

**Escenario 2:** Desempeño de TCP Vegas según parámetros alfa y beta en escenarios híbridos con errores en ráfagas.

El escenario consiste en una topología híbrida de tres nodos.

Los nodos 0 y 1 deben estar vinculados por un enlace cableado de tipo dúplex, con un ancho de banda 10 Mb/s, retardo de propagación 2 ms, y política de servicio de las colas DropTail.

El enlace entre los nodos 1 y 2 debe ser inalámbrico, con modo de propagación TwoRayGround, la capa física WirelessPhy, MAC 802.11, la antena OmniAntenna y el nodo inalámbrico se encuentra fijo.

Se considera que la pérdida de paquetes se debe únicamente a los errores en ráfaga introducidos. Se utiliza una longitud variable de entre 0 y 20 paquetes.

Se estudia el comportamiento de la variable TCP Vegas. Los valores de alfa y beta correspondientes a la misma fluctúan de 1 a 10 y de 3 a 12, respectivamente. Para cada combinación posible de estos parámetros se realizan simulaciones con cada longitud de ráfaga definida. En total, se realizan 1534 simulaciones.

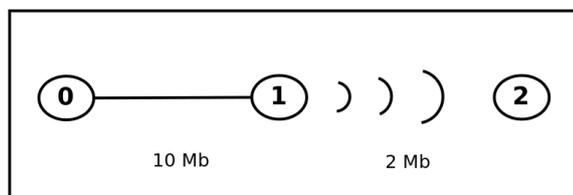


Figura 5. Esquema

## 4. RESULTADOS Y DISCUSIÓN

### 4.1. Resultados

A continuación se muestran los resultados de adaptar el modelo de base a cada escenario mediante tablas que incluyen las funcionalidades, los valores predeterminados, los valores utilizados y observaciones correspondientes.

Los valores en la tabla pueden ser booleanos (SÍ-NO), números específicos o intervalos (dependiendo de la prueba realizada), dejarse indefinidos. Las funcionalidades agregadas como consecuencia de los requisitos solicitados se destacan en negrita. En los casos que corresponde se indica la unidad utilizada entre paréntesis. Se asume un enlace por par de nodos cableados o estación base. La variante por defecto es TCP Tahoe. Las consideraciones particulares de cada prueba se indican en la sección de observaciones.

En la tabla se dejan de lado detalles específicos como:

- Posiciones iniciales (para nodos móviles)
- Enlaces entre nodos cableados y/o estaciones base.
- Dirección del tráfico (nodo origen y destino)
- En caso de realizar múltiples pruebas, no se especifican las configuraciones utilizadas para el tráfico en cada una.

**Escenario 1:**

Este escenario fue diseñado para estudiar el comportamiento de distintas variantes de TCP ante errores en ráfaga.

Se alteró el modelo para incluir la capacidad de introducir distintos tipos de errores en el tráfico, y en particular, errores en ráfaga. También se incorporó la posibilidad de utilizar las variantes de TCP desarrolladas sobre TCP Linux en NS-2, como YeAH, Scalable, HTCP, Westwood, Hybla, Cubic, Bic, Highspeed, Vegas, entre otros (Wei y Cao, 2006). Finalmente, en base a este escenario se introdujo por primera vez una planilla de cálculo con un resumen centralizado de los datos.

| Categoría                   | Variable o parámetro ( <i>unidad o aclaración</i> ) | Valor predeterminado     | Valor utilizado |
|-----------------------------|---|--------------------------|-----------------|
| Topología                   | Cableada  | No                       | Predeterminado  |
|                             | Inalámbrica   | No                       | Predeterminado  |
|                             | Híbrida   | No                       | Si              |
| Nodos                       | Cableados   | 0                        | 1               |
|                             | Inalámbricos  | 0                        | 1               |
|                             | Estaciones base                                     | 0                        | 1               |
| Configuración de nodo móvil | Canal   | Channel/Wireless Channel | Predeterminado  |
|                             | Modelo de propagación                               | Propagation/TwoRayGround | Predeterminado  |
|                             | Interfaz de red                                     | Phy/WirelessPhy          | Predeterminado  |
|                             | Tipo de control de acceso al medio                  | Mac/802_11               | Predeterminado  |
|                             | Política de manejo de cola                          | Queue/DropTail/PriQueue  | Predeterminado  |
|                             | Modelo de antena                                    | Antenna/OmniAntenna      | Predeterminado  |

|   |   |                       |                       |
|---|---|-----------------------|-----------------------|
| Configuración de enlace entre nodos cableados o estaciones base | Tamaño de de buffer (paquetes)          | 50                    | Predeterminado        |
|   | Protocolo de enrutamiento               | DSDV                  | Predeterminado        |
|   | Ancho de banda (Mb)                     | 2                     | Predeterminado        |
|   | Ancho de banda (Mb)                     | 2                     | Predeterminado        |
|   | Delay de propagación (ms)               | 2                     | Predeterminado        |
|   | Tipo de enlace                          | Duplex                | Predeterminado        |
|   | Política de manejo de cola              | DropTail              | Predeterminado        |
| Tráfico (Variantes de TCP usadas)                               | Tahoe                                   | Sí                    | No                    |
|   | Reno                                    | No                    | Sí                    |
|   | New Reno                                | No                    | Predeterminado        |
|   | Sack                                    | No                    | Predeterminado        |
|   | Fack                                    | No                    | Predeterminado        |
|   | Vegas                                   | No                    | Sí                    |
|   | <b>Linux</b>                            | <b>No</b>             | <b>Predeterminado</b> |
|   | <b>Cubic</b>                            | <b>No</b>             | <b>Si</b>             |
|   | <b>Bic</b>                              | <b>No</b>             | <b>Predeterminado</b> |
|   | <b>Hybla</b>                            | <b>No</b>             | <b>Predeterminado</b> |
|   | <b>Westwood</b>                         | <b>No</b>             | <b>Si</b>             |
|   | <b>Highspeed</b>                        | <b>No</b>             | <b>Predeterminado</b> |
|   | <b>Veno</b>                             | <b>No</b>             | <b>Predeterminado</b> |
|   | <b>Illinois</b>                         | <b>No</b>             | <b>Predeterminado</b> |
|   | <b>Compound</b>                         | <b>No</b>             | <b>Predeterminado</b> |
|   | <b>YeAH</b>                             | <b>No</b>             | <b>Predeterminado</b> |
|   | <b>HTCP</b>                             | <b>No</b>             | <b>Predeterminado</b> |
| <b>Low Priority</b>   | <b>No</b>                               | <b>Predeterminado</b> |                       |
| Tráfico (Variables)   | Comienzo de tráfico ( <i>segundos</i> ) | 0.0                   | 5.0                   |
|   | Fin del tráfico ( <i>segundos</i> )     | 100.0                 | No definido           |

|  |  |   |                              |  |
|--|--|---|------------------------------|--|
| Error  | Cantidad de paquetes ( <i>paquetes</i> ) <sup>2</sup>          | 3000  | Predeterminado               |  |
|  | Tamaño de paquete de TCP ( <i>Bytes</i> )                      | 1000  | Predeterminado               |  |
|  | TCP Tick ( <i>segundos</i> )                                   | 0.1   | Predeterminado               |  |
|  | <b>Estadístico (porcentaje de paquetes perdidos)</b>           | <b>0</b>  | <b>0</b>                     |  |
|  | <b>Ráfaga (paquetes)</b>                                       | <b>0</b>  | <b>0, 5, 10, 15, 20 y 25</b> |  |
|  | <b>Corte de línea (segundos)</b>                               | <b>0</b>  | <b>0</b>                     |  |
| Salidas de simulación  | Archivo de traza “old” (cableado)                              | No  | Predeterminado               |  |
|  | Archivo de traza “new” (cableado y/o inalámbrico)              | Sí  | Predeterminado               |  |
|  | Archivo de traza para NAM                                      | Sí  | Predeterminado               |  |
| Parámetros de rendimiento de TCP (incluye archivo de traza y gráfico de ser posible) | <b>Planilla de cálculo con resumen de resultados promedios</b> | <b>No</b>   | <b>Si</b>                    |  |
|  | Throughput promedio ( <i>Mbps</i> )                            | Sí  | Predeterminado               |  |
|  | Throughput instantáneo ( <i>Mbps</i> )                         | Sí  | No                           |  |
|  | Delay promedio ( <i>segundos</i> )                             | Sí  | Predeterminado               |  |
|  | Delay instantáneo ( <i>segundos</i> )                          | Sí  | No                           |  |
|  | Jitter ( <i>segundos</i> )                                     | Sí  | No                           |  |
|  | Ventana de congestión ( <i>paquetes</i> )                      | Sí  | Predeterminado               |  |
|  | RTT ( <i>segundos</i> )  | Sí  | No                           |  |
|  | RTO ( <i>segundos</i> )  | Sí  | No                           |  |
|  | <i>PLR</i>   | Sí  | No                           |  |
|  | <i>PDR</i>   | Sí  | No                           |  |
|  | Número de secuencia  | Sí  | No                           |  |
|  | Observaciones  | Topología híbrida de tres nodos: nodo cableado, estación base y nodo móvil. |                              |  |
|  |  | Los 3000 paquetes se envían desde el nodo 0 (cableado) al nodo 2            |                              |  |

- 2 Al definir el tráfico, se puede indicar cantidad de paquetes a transmitir o tiempo de transmisión de paquetes. Estas opciones son mutuamente excluyentes.

|            |   |
|------------|---|
| Resultados | (móvil). El error en ráfaga comienza al alcanzar el número de secuencia 999.  |
|            | Para cada variante de TCP utilizada se ejecuto una simulación por ráfaga de error definida (0, 5, 10, 15, 20 y 25), 24 en total.<br>Para cada una se obtuvo Delay, Throughput y Venana de congestión. |

Se generaron 24 simulaciones en total. El sistema organizó los resultados en directorios con nombres representativos de cada una. Para cada simulación se obtuvieron: archivo de traza del simulador, archivo de traza para NAM, archivos de datos de throughput y delay promedios, ventana de congestión.

A partir de la planilla de cálculo donde se guardaron los valores de cada prueba se generaron gráficos que permitieron hacer un análisis comparativo de los comportamientos ante errores en ráfaga.

### Escenario 2:

Este caso fue pensado a partir de los resultados del escenario anterior. Se observó que debido al comportamiento proactivo de TCP Vegas, el efecto negativo de las ráfagas de errores se acentuaba. Se diseñó un nuevo escenario para explorar valores de alfa y beta que optimicen el comportamiento ante pérdidas de paquetes continuas.

Gracias a este escenario, se incorporó la definición de los parámetros alfa y beta para la variante TCP Vegas. La conveniencia de la automatización del proceso se pone de manifiesto en este caso debido a la magnitud de pruebas que deben realizarse (1534).

| Categoría                   | Variable o parámetro ( <i>unidad o aclaración</i> ) | Valor predeterminado      | Valor utilizado |
|-----------------------------|---|---------------------------|-----------------|
| Topología                   | Cableada  | No                        | Predeterminado  |
|                             | Inalámbrica   | No                        | Predeterminado  |
|                             | Híbrida   | No                        | Sí              |
| Nodos                       | Cableados   | 0                         | 1               |
|                             | Inalámbricos  | 0                         | 1               |
|                             | Estaciones base                                     | 0                         | 1               |
| Configuración de nodo móvil | Canal   | Channel/Wireless Channel  | Predeterminado  |
|                             | Modelo de propagación                               | Propagation/Two RayGround | Predeterminado  |

|   |                                    |                         |                |                |
|---|------------------------------------|-------------------------|----------------|----------------|
| Configuración de enlace entre nodos cableados o estaciones base | Interfaz de red                    | Phy/WirelessPhy         | Predeterminado |                |
|   | Tipo de control de acceso al medio | Mac/802_11              | Predeterminado |                |
|   | Política de manejo de cola         | Queue/DropTail/PriQueue | Predeterminado |                |
|   | Modelo de antena                   | Antenna/OmniAntenna     | Predeterminado |                |
|   | Tamaño de de buffer (paquetes)     | 50                      | Predeterminado |                |
|   | Protocolo de enrutamiento          | DSDV                    | Predeterminado |                |
|   | Ancho de banda (Mb)                | 2                       | Predeterminado |                |
|   | Ancho de banda (Mb)                | 2                       | 10             |                |
|   | Delay de propagación (ms)          | 2                       | Predeterminado |                |
|   | Tipo de enlace                     | Duplex                  | Predeterminado |                |
|   | Política de manejo de cola         | DropTail                | Predeterminado |                |
|   | Tráfico (Variantes de TCP usadas)  | Tahoe                   | Si             | No             |
|   |                                    | Reno                    | No             | Predeterminado |
| New Reno  |                                    | No                      | Predeterminado |                |
| Sack  |                                    | No                      | Predeterminado |                |
| Fack  |                                    | No                      | Predeterminado |                |
| Vegas   |                                    | No                      | Si             |                |
| Linux   |                                    | No                      | Predeterminado |                |
| Cubic   |                                    | No                      | Predeterminado |                |
| Bic   |                                    | No                      | Predeterminado |                |
| Hybla   |                                    | No                      | Predeterminado |                |
| Westwood  | No                                 | Predeterminado          |                |                |
| Highspeed   | No                                 | Predeterminado          |                |                |
| Veno  | No                                 | Predeterminado          |                |                |

|   |   |          |  |
|---|---|----------|--|
| Tráfico (Variables)   | Illinois  | No       | Predeterminado   |
|   | Compound  | No       | Predeterminado   |
|   | YeAH  | No       | Predeterminado   |
|   | HTCP  | No       | Predeterminado   |
|   | Low Priority  | No       | Predeterminado   |
|   | Comienzo de tráfico ( <i>segundos</i> )                 | 0.0      | 5.0  |
|   | Fin del tráfico ( <i>segundos</i> )                     | 100.0    | No   |
|   | Cantidad de paquetes ( <i>paquetes</i> ) <sup>3</sup>   | 3000     | Predeterminado   |
|   | Tamaño de paquete de TCP ( <i>Bytes</i> )               | 1000     | Predeterminado   |
|   | TCP Tick ( <i>segundos</i> )                            | 0.1      | Predeterminado   |
| Error   | <b>Alfa para Vegas</b>                                  | <b>1</b> | <b>1, 2, 3, 4, 5, 6, 7, 8, 9, 10</b>                                     |
|   | <b>Beta para Vegas</b>                                  | <b>3</b> | <b>3, 4, 5, 6, 7, 8, 9, 10, 11, 12</b>                                   |
|   | Estadístico ( <i>porcentaje de paquetes perdidos</i> )  | 0        | 0  |
|   | Ráfaga ( <i>paquetes</i> )                              | 0        | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20 |
| Salidas de simulación   | Corte de línea ( <i>segundos</i> )                      | 0        | 0  |
|   | Archivo de traza “old” (cableado)                       | No       | Predeterminado   |
|   | Archivo de traza “new” (cableado y/o inalámbrico)       | Si       | Predeterminado   |
|   | Archivo de traza para NAM                               | Si       | Predeterminado   |
|   | Planilla de cálculo con resumen de resultados promedios | No       | Si   |
| Parámetros de rendimiento de TCP (incluye archivo de traza y gráfico de ser | Throughput promedio ( <i>Mbps</i> )                     | Si       | Predeterminado   |
|   | Throughput instantáneo ( <i>Mbps</i> )                  | Si       | Predeterminado   |
|   | Delay promedio ( <i>segundos</i> )                      | Si       | Predeterminado   |

- 3 Al definir el tráfico, se puede indicar cantidad de paquetes a transmitir o tiempo de transmisión de paquetes. Estas opciones son mutuamente excluyentes.

|               |   |    |                |
|---------------|---|----|----------------|
| posible)      | Delay instantáneo ( <i>segundos</i> )   | Si | Predeterminado |
|               | Jitter ( <i>segundos</i> )  | Si | No             |
|               | Ventana de congestión ( <i>paquetes</i> )   | Si | No             |
|               | RTT ( <i>segundos</i> )   | Si | No             |
|               | RTO ( <i>segundos</i> )   | Si | No             |
|               | <i>PLR</i>  | Si | No             |
|               | <i>PDR</i>  | Si | No             |
|               | Número de secuencia   | Si | No             |
| Observaciones | <p>Topología híbrida de tres nodos: nodo cableado, estación base y nodo móvil.</p> <p>Se varía alfa entre 1 y 10, y beta entre 3 y 12.</p> <p>Los 3000 paquetes se envían desde el nodo 0 (cableado) al nodo 2 (móvil). El error en ráfaga comienza al alcanzar el número de secuencia 999.</p> <p>El ancho de banda entre enlaces cableados es mucho mayor que entre los inalámbricos. Esto permite estudiar el efecto de las pérdidas cuando el cuello de botella está en la parte inalámbrica.</p> |    |                |
| Resultados    | <p>Para cada combinación de alfa y beta se realizan pruebas con ráfagas de errores entre 0 y 20. En total, se realizaron 1534 simulaciones.</p> <p>De cada una se obtuvo Delay y Throughput promedio e instantáneo.</p>   |    |                |

Se generaron 1534 simulaciones en total. El sistema organizó los resultados en directorios con nombres representativos de cada una. Para cada simulación se obtuvieron: archivo de traza del simulador, archivo de traza para NAM, archivos de datos de throughput y delay instantáneos y promedios.

A partir de la planilla de cálculo donde se guardaron los valores de cada prueba se generaron gráficos que permitieron ver qué valores de alfa y beta podían optimizar el comportamiento de Vegas ante errores en ráfaga.

## 4.2. Discusión

La discusión no abarca resultados específicos a las pruebas realizadas, ya que lo que se busca mostrar es la conveniencia de tener un modelo base. Como se puede observar en los escenarios estudiados, éste requiere pocas modificaciones para estudiar distintos fenómenos.

En ambos casos se puede observar que la mayoría de los valores utilizados se corresponden a los predeterminados. El modelo base ya era capaz de satisfacer gran parte de los requisitos.

El sistema organiza los resultados de cada simulación en directorios cuyos nombres indican los valores utilizados. Estos varían según lo que se estudia y en la práctica sólo requiere cambiar el nombre de una variable en el código.

La cantidad de veces que debe ejecutarse un script de simulación con diferentes valores se maneja con el gestor. El primer escenario tardó aproximadamente 10 minutos en generarse. El

segundo casi 4 horas. Sin embargo, la diferencia en el código se reduce a la inclusión de una función que haga variar los valores de alfa y beta antes de ejecutar la siguiente simulación.

Luego de incorporar las funcionalidades al modelo base, la adaptación tardó menos de 15 minutos. Se ajustaron los valores y se descartaron algunos gráficos, pero luego de haber preparado el escenario para generar las pruebas, todas las modificaciones estuvieron motivadas por la exploración del fenómeno estudiado. Los trasfondos de funcionamiento de NS-2 no interrumpieron el flujo de trabajo.

Vale notar que la incorporación de nuevas funcionalidades es lo que toma más tiempo, principalmente a causa de la investigación involucrada. El tiempo de adaptación del modelo base a un escenario que requiere funcionalidades existentes es despreciable en comparación.

El modelo base fue adaptado con éxito para estudiar una amplia variedad de escenarios. Algunos de ellos dieron lugar a presentaciones y publicaciones científicas durante el año 2017, en eventos como JI (UNPA)<sup>4</sup>, JBI (UNPA-UMAG)<sup>5</sup>, WICC, TE&ET, CACIC e INCISCOS. El análisis de resultados y conclusiones derivadas ellos puede hallarse en las referencias.

## 5. CONCLUSIONES

Automatizar el proceso de simulación permite enfocar la atención en los aspectos del escenario a simular, no en los detalles técnicos de la herramienta utilizada.

Desarrollar un modelo base, que abarque una amplia variedad de las funcionalidades disponibles en la herramienta seleccionada, facilita el trabajo necesario para probar nuevos escenarios de simulación.

La reducción en el tiempo de preparación de un escenario que cumpla con determinados requisitos se traduce en una mayor variedad de pruebas posibles en menor tiempo.

La automatización del proceso -generación de modelo, ejecución de simulación, obtención de resultados- posibilita reducir el tiempo que transcurre desde que se plantea un problema hasta que se obtienen los resultados de la simulación. Esto se traduce en la posibilidad de estudiar una mayor variedad de escenarios en menor tiempo, dando lugar a un análisis más amplio de cada caso. De esta forma, se pueden explorar más variantes del escenario original en busca de fenómenos de interés o anomalías.

## 6. RECOMENDACIONES

Se recomienda extender el sistema para automatizar la obtención de parámetros no específicos al protocolo TCP para ayudar en el estudio de otros aspectos del comportamiento de redes, protocolos, topologías, etc.

---

4 Trabajos “La utilización del ns2 para el análisis de parámetros de dos tipos de topologías comparadas con distintas variantes del protocolo Reno vs Vegas” y “Una revisión de las versiones del protocolo TCP y los ámbitos en los cuales se aplican”, presentados en las XIV Jornadas de Informática de la UNPA-UARG (2017).

5 Trabajo “Congestión: falso positivo. El caso del protocolo TCP en redes inalámbricas”, presentado en las I Jornadas Binacionales de Investigadores “Encuentro en el extremo austral” de la UNPA y la UMAG (2017).

## 7. AGRADECIMIENTOS

Se agradece el soporte de la Universidad Nacional de la Patagonia Austral, y la supervisión de la Lic. Claudia Gonzalez, el Mg. Carlos Talay y el Ing. Diego Rodriguez Herlein.

## 6. REFERENCIAS

- ALLMAN, M., FALK, A. (1999). *On the effective evaluation of TCP*. ACM Computer Communication Review. Recuperado el 26 de marzo de 2017, de <https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20000059208.pdf>.  
<https://doi.org/10.1145/505696.505703>
- BONASTRE, O., SALVADOR, C. (2005). *Introducción a la programación de protocolos de comunicaciones con Network Simulator 2*. Editorial Club Universitario. ISBN: 978-84-9948-485-3.
- BONAVENTURE, O., (2001). *Computer Networking. Principles, Protocols and Practice*. The Saylor Foundation. Recuperado el 26 de marzo de 2017, de <http://www.saylor.org/courses/cs402/>
- CHANDRAN, K., RAGHUNATHAN, S., VENKATESAN S., PRAKASH, R., (1998). *A feedback Based Scheme for improving TCP performance in ad-hoc wireles Networks*. University of Texas, Dallas. Procedente de “18th International Conference on Distributed Computing Systems”. Recuperado el 8 de junio de 2016, de [https://www.researchgate.net/publication/3751440\\_A\\_Feedback\\_Based\\_Scheme\\_for\\_Improving\\_TCP\\_Performance\\_in\\_Ad-Hoc\\_Wireless\\_Networks](https://www.researchgate.net/publication/3751440_A_Feedback_Based_Scheme_for_Improving_TCP_Performance_in_Ad-Hoc_Wireless_Networks)
- DALAL, P., KOTHARI, N., DASGUPTA, K., (2001). *Improving TCP Performance over Wireless Networks with frequent disconnections*. International Journal of Computer Networks & Communications (IJCNC) Vol.3, No.6, November 2011.
- DARPA Internet Program (1981). *RFC 793: Transmission Control Protocol*. Formation Sciences Institute, University of Southern California. Recuperado el 18 de mayo de 2017, de <https://tools.ietf.org/html/rfc793>
- ELAARAG, H. 2002. *Improving TCP Performance over Mobile Networks*. Revista ACM Computing Surveys, Vol. 34, No. 3, September 2002, pp. 357-374.  
<https://doi.org/10.1145/568522.568524>
- FOROUZAN B. A., FEGAN, S. (2007). *Transmisión de datos y redes de computadoras (4ta. Edición)*. New York: McGraw-Hill. ISBN 978-0-07-296775-3
- GOSWAMI, B., ASADOLLAHI, S., ASARE, I, (2016). *Performance evaluation of widely implemented congestion control algorithms over diversified networking situations*. Procedente de “International Conference on Computer Science Networks and Information Technology” realizada 23 y 24 Febrero 2016, en Pattaya, ISBN: 9788193137338
- HERNANDEZ BENET, C., VIZCAINO, F., (2014). *Study of TCP Available Bandwidth using NS2 and its forecasting based on Genetic Algorithm*. Tesis de maestría de la Universidad de Karlstad, Suecia.
- Internet Engineering Task Force. (2015). *RFC 7414: A roadmap for Transmission Control Protocol Specification Documents*. Recuperado el 18 de mayo de 2017, de <https://tools.ietf.org/html/rfc7414>

- ISSARIYAKUL, T., HOSSAIN, E. (2012). *Introduction to Network Simulartor NS2 (2nd edition)*. New York: Springer. 978-1-4614-1405-6. <https://doi.org/10.1007/978-1-4614-1406-3>
- LORENTE, O., DE CERIO, D. (2005). *Implementación de un modelo de canal inalámbrico para redes 802.11 bajo el simulador ns-2*. Escuela Politécnica Superior de Catalunya.
- MARRONE, L., BARBIERI, A., ROBLES, M., (2013). *TCP Performance Cubic, Vegas & Reno*. Revista JCS&T Vol. 13 No. 1. April, 2013
- Network Working Group. (2004). *RFC 3819: Advice for Internet Subnetwork Designers*. Recuperado el 18 de mayo de 2017, de <https://tools.ietf.org/html/rfc3819>
- Network Working Group. (2007). *RFC 5033: Specifying New Congestion Control Algorithms*. Recuperado el 18 de mayo de 2017, de <https://tools.ietf.org/html/rfc5033>
- PANDEY, S., TYAGI, V., (2013). *Performance Analysis of Wired and Wireless Networks using NS2 Simulator*. International Journal of Computer Applications. Volume72–No.21, June 2013.
- PARZIALE, L., BRITT, D., DAVIS, C., FORRESTER J., LIU, W., MATTHEWS, C., ROSSELOT N., (2006). *TCP/IP Tutorial and Technical Overview*. International Technical Support Organization, IBM. Sitio Web: [ibm.com/redbooks](http://ibm.com/redbooks)
- PENTIKOUSIS, K. (2001). *TCP in wired-cum-wireless environments*. State University of New York. Publicado en IEEE Communications Surveys & Tutorials. Recuperado el 26 de marzo de 2017, de [https://www.researchgate.net/publication/2356074\\_TCP\\_in\\_Wired-cum-Wireless\\_Environments](https://www.researchgate.net/publication/2356074_TCP_in_Wired-cum-Wireless_Environments)
- Proyecto NS-3. (2016). *ns-3 tutorial 3-27*. Recuperado 26 de marzo de 2017, de <https://www.nsnam.org/ns-3-27/documentation/>
- RODRIGUEZ HERLEIN, D., TALAY, C., (2016). *Explorando posibles mejoras de protocolo TCP en redes móviles*. Presentado en WICC 2016. URL de descarga: <http://sedici.unlp.edu.ar/handle/10915/52801>
- RODRIGUEZ HERLEIN, D., TALAY, C., GONZALES, C., TRINIDAD, F., MARRONE, L., (2017). *Un caso práctico del simulador ns-2 en el análisis del comportamiento de variantes del protocolo TCP con fines de investigación y enseñanza*. XII Congreso de Tecnología en Educación y Educación en Tecnología. Universidad Nacional de la Matanza, Buenos Aires. URL de descarga: <http://sedici.unlp.edu.ar/handle/10915/63370>
- RODRIGUEZ HERLEIN, D., TALAY, C., GONZALES, C., TRINIDAD, F., MARRONE, L., (2017). *Consideraciones sobre el comportamiento del protocolo TCP en sus variantes Vegas, Reno, Cubic y Westwood ante errores en ráfaga en una topología híbrida*. URL de descarga: <http://sedici.unlp.edu.ar/handle/10915/63846>
- RODRIGUEZ HERLEIN, D., TALAY, C., GONZALES, C., TRINIDAD, F., MARRONE, L., (2017). *Desempeño de TCP Vegas según parámetros alfa y beta en escenarios híbridos con errores en ráfagas*. Presentado en INCISCOS.
- STALLINGS W. (2000). *Comunicaciones y Redes de Computadoras (6ta edición)*. Prentice-Hall. ISBN: 84-205-2986-9
- SHARMA, J., SAINI, H., ARORA, D., (2012). *Analysis of Reno: A TCP Variant*. Presentado en International Journal of Electronics and Communication Engineering. ISSN 0974-2166 Volume 5, Number 3 (2012), pp. 267-277

- TORKEY, H., ATTIYA, G., Z. MORSE., (2008). *Performance Evaluation of End-to-End Congestion Control Protocols*. Presentado en “18th International Symposium on the Mathematical Theory of Networks and Systems”. Recuperado el 26 de marzo de 2017, de <http://scholar.lib.vt.edu/MTNS/Papers/047.pdf>
- TSAOUSSIDIS, V., MATTA, I., (2001). *Open Issues on TCP for Mobile Computing*. Boston University and Northeastern University. Recuperado el 26 de marzo de 2017, de <http://www.cs.bu.edu/techreports/pdf/2001-013-open-issues-tcp-wireless.pdf>
- UC Berkeley, LBL, USC/ISI, Xerox PARC. (2011). *The NS Manual, (formerly ns Notes and Documentation) ns-2.35*. Editores: Kevin Fall y Kannan Varadhan. Recuperado el 13 de mayo de 2016, de [https://isi.edu/nsnam/ns/doc/ns\\_doc.pdf](https://isi.edu/nsnam/ns/doc/ns_doc.pdf)
- VALISKA J., HRUSOVSKY, B., MARCHEVSKY, S., PILLAR, S., (2012). *Error Models Simulations in transmission channels using Network Simulation environment*. Revista Acta Electrotechnica et Informatica (Department of Electronic and Multimedia Communications, Faculty of Electrical Engineering and Informatics, Technical University), Vol. 12, No. 2, 2012, 51–58, DOI: 10.2478/v10198-012-0019-1. <https://doi.org/10.2478/v10198-012-0019-1>
- WEI, D., CAO, P. 2006. *NS-2 TCP-Linux: An NS-2 TCP implementation with Congestion Control Algorithms from Linux*. Stanford University. Recuperado el 20 de noviembre de 2017, de <http://netlab.caltech.edu/publications/ns2linux06.pdf>
- XYLOMENOS G., MÄHÖNEN P., SAARANEN, M., (2001). *TCP performance issues over Wireless link*. Recuperado el 8 de junio de 2017, de <https://cseweb.ucsd.edu/groups/csl/pubs/journal/commag01.pdf>. S/R.