Análisis de un modelo de redes de pares-a-pares bajo un ataque cibernético utilizando cadenas de Markov

Sánchez-Patiño Natalia[1], Gallegos-García Gina[2], Rivero-Ángeles Mario[2]

1. Universidad Nacional Autónoma de México, Facultad de Estudios Superiores Cuautitlán,
2. Instituto Politécnico Nacional, Centro de Investigación en Computación

Las redes pares-a-pares (P2P) son sistemas distribuidos donde ninguna autoridad central gobierna el comportamiento de los pares individuales. Una aplicación típica es el intercambio de archivos de alguna clase. Estos sistemas se basan en la participación voluntaria de los pares en el trabajo de subida y almacenamiento de estos archivos.
Estas redes representan actualmente un porcentaje considerable de todo el ancho de banda mundial. Sin embargo, este nuevo modelo de comunicaciones posee también un claro inconveniente que representa una multitud de vulnerabilidades y amenazas a la seguridad. La naturaleza de la propia filosofía P2P implica que no existe ningún servidor centralizado que se encarga de subir, almacenar y comprobar la autenticidad de los recursos. Una consecuencia directa de ello es que no hay ningún mecanismo para controlar qué contenido se comparte esto hace que las redes P2P sean óptimas para la propagación de un software malicioso o malware, en general.
En este trabajo, se presenta un esquema para un sistema de redes P2P basado en el software BitTorrent donde se pretende simular el tráfico de la red basado en el uso de cadenas de Markov, con el objetivo de poder analizar el comportamiento del sistema e identificar sus debilidades. Se empleó un enfoque con tres aproximaciones, la primera centrada tan solo en la red P2P, la segunda centrada en los ataques y la red y la tercera en posibles medidas de prevención a ataques.

Keywords: Cadenas de Markov; Modelos analíticos; Redes P2P; Criptografía

1. INTRODUCCIÓN

Una red de pares a pares (P2P) por sus siglas en inglés, es una red distribuida con nodos dinámicos, llamados pares. Los pares en el sistema proporcionan recursos como ancho de banda, espacio de almacenamiento y potencia de cómputo con el objetivo de intercambiar los datos o realizar alguna tarea colectiva (Esquivel et al. 2013). En los últimos años, los sistemas P2P han recibido una gran atención ya que las aplicaciones basadas en este tipo de redes están dominando una parte considerable del tráfico en Internet, y es usada para proveer servicios de transmisión (Adamu et al. 2011) de audio y video, como se puede observar en aplicaciones como Spotify y Skype.

Una gran ventaja de un sistema P2P sobre las aplicaciones cliente / servidor convencionales reside en su robustez. Esto se debe a que los datos ya no se concentran en una entidad central, como un servidor; en cambio, se distribuyen entre la población de usuarios. Otra ventaja es la escalabilidad. Esto se debe al hecho de que todos los pares del sistema comparten sus recursos y cualquier nuevo par que llega (Robayo Santana 2009), mientras que el aumento de la demanda también aumenta la capacidad del sistema gracias a la posibilidad de distribuir ejecución determinado servicio entre los múltiples pares de la red. Sin embargo, La gestión de las redes más compleja, siendo necesario, por ejemplo, establecer mecanismos para identificar donde se pueden encontrar los servicios, desarrollar protocolos específicos para las comunicaciones entre pares, etc. Esto suele derivar en redes menos eficientes que las basadas en el modelo tradicional, en relación al tiempo necesario para enviar una solicitud y recibir la respuesta correspondiente.

Una desventaja básica es que las redes P2P sufren por la ausencia total de control sobre la distribución de contenido, ya que se ejecutan en PC y son administradas por los propios usuarios. Las PC de los usuarios no están monitorizadas, no son seguras y son casi imposibles de controlar por un proveedor de red (Robayo Santana 2009, Torres-Cruz et al. 2018). La naturaleza de las redes por pares implica que no existe ningún servidor centralizado que se encargue de subir almacenar y comprobar la autenticidad de los recursos lo que hace que no hay un mecanismo para controlar qué contenido se comparte. Siendo esto una amenaza para la seguridad. El objetivo de este trabajo es mejorar la seguridad de las redes P2P analizándolas por medio del algoritmo obtenido mediante las cadenas de Markov.

2. TRABAJO RELACIONADO/ MARCO TEÓRICO

2.1. Redes de pares a pares

Las redes pares a pares están organizadas en una estructura plana que permite el intercambio directo de información entre clientes. Tomando a BitTorrent como ejemplo, cuando se descarga un archivo, se descarga parte del archivo y la otra parte es recibida al mismo tiempo por diferentes pares que pueden haber descargado o descargado el archivo (Wararkar et al. 2016). Este mecanismo hace que mejore la velocidad de descarga. Un Par actúa como servidor y como cliente, en una estructura auto organizada y que se coordinada dinámicamente, ya que el número de nodos se puede aumentar o disminuir en cualquier momento.

El par respectivo es responsable de encontrar la información correcta. Si los datos encontrados son comunes, es fácil encontrarlos, pero es difícil encontrar datos raros en otros pares. Las redes P2P se pueden dividir en categorías.

- Red P2P no estructurada: Este tipo de red no le dice la ubicación exacta de los datos, por lo que busca los datos solicitados en su propia red. Se comunican entre sí a través de la demanda fluctuante, lo que aumenta el tráfico y, por lo tanto, desperdicia ancho de banda. La desventaja de este tipo de red es que no parece saber si puede acceder a los datos.
- Red P2P estructurada: Este tipo de red utiliza un protocolo coherente globalmente para una búsqueda eficiente. Utiliza una tabla hash llamada DHT (Distributed Hash Table), esta tabla proporciona la información acerca de donde proviene un archivo en particular en función de varios factores importantes.
- Red P2P híbrida: Combina la topología de una estructurada y no estructurada. Este tipo de sistemas definen a los llamados Superpares, que actúan como servidores a una pequeña porción de la red y cada uno de ellos tiene una lista sobre la información de los archivos

2.2. Ataques cibernéticos en redes de pares a pares

Se han publicado diversos artículos que ilustran diferentes modelos de ataque P2P o que ejemplifican diferentes mecanismos de defensa en determinados aspectos. En esta sección citaremos algunos de esos trabajos. En (Wagner & Plattner 2002) se discute cómo un sistema P2P puede ser utilizado para generar un ataque DDoS. En (Douceur 2002), se ha analizado el Sybil, uno de los principales tipos de ataques en la red P2P. En (Singh et al. 2004) se examina otro ataque de red P2P denominado Eclipse. (Christin et al. 2005) examina los ataques basados en la disponibilidad de contenidos en la red P2P. (Nielson et al. 2005) proporciona una taxonomía de los ataques racionales encontrados en la red P2P. (Engle & Khan 2006) ilustra diferentes tipos de métodos de ataque P2P y sus soluciones. En (Keyani et al. 2002) se propone un método de recuperación distribuida si una red P2P sufre un ataque violento. En (Mishra 2003) se propone un sistema P2P resistente a los ataques. En (Ferdous et al. 2007) se propone una taxonomía completa de todos los métodos de ataque conocidos en la red P2P. Aunque hay muchos artículos en este campo respectivo, pero casi cada uno de ellos se limita a un aspecto diferente de una sola entidad de ataque. En este trabajo se propone estudiar distintas entidades de ataque bajo características que comparten entre si.

El tipo de red que nosotros tomamos para obtener los parámetros y desarrollar el modelo, fue a partir de la aplicación Bit Torrent, esta aplicación utiliza redes de tipo híbridas, por lo que también consideramos todos los posibles ataques para este tipo de redes.

1. Ataques generales a redes
   - *a*) DoS y DDoS (Denegación de servicio): Causan que el servicio dejen de funcionar al utilizar solicitudes de servicio razonables para agotar los recursos del anfitrión objetivo. Disponibilidad. Defensa: Valuación. El anfitrión envía acertijos a sus clientes antes de continuar con el cálculo solicitado, asegurando así que el cliente realice un cálculo igualmente costoso.
   - *b*) Hombre en el medio(MiTM): Un atacante se inserta entre otros dos nodos de la red, realiza conexiones independientes y transmite mensajes entre ellos. Confidencialidad, integridad. Defensa: uso de firmas digitales basadas en criptografía de clave pública.
   - *c*) Gusanos: Pieza de software complejo que es capaz de ataques mucho más complicados, como la recopilación de todo tipo de informa-

ción. Confidencialidad. Defensa: actualizaciones de seguridad periódicas.

d) Botnets/Zombies: redes de máquinas comprometidas bajo el control de un atacante. No repudio. Defensa: dividir nodos de alto grado para evitar respuestas específicas y diseñar conjuntos de pruebas de Turing.

e) Escuchando a escondidas: Los atacantes pueden obtener acceso a los datos dentro de una red y espiar el tráfico. Confidencialidad. Defensa: fuerte seguridad física y fuertes servicios de encriptación.

f) Mascarada: Una entidad del sistema se hace pasar ilegítimamente como otra entidad para obtener acceso a sistemas confidenciales. Confidencialidad, Autenticación, integridad. Defensa: filtrar los paquetes entrantes que parecen provenir de una dirección IP interna o de una dirección invalida.

2. Ataques específicos a redes P2P

A nivel de red

a) Sybil: El usuario malintencionado puede crear múltiples identidades falsas y pretender ser múltiples nodos físicos distintos en el sistema y atacan varios protocolos, como el almacenamiento distribuido para anular los mecanismos de replicación y fragmentación. Autenticación. Defensa: El sistema debe garantizar que las identidades distintas se refieran a entidades distintas y limitar la capacidad de una entidad para determinar la identidad.

b) Mapeo de Identidad: Permite a un atacante obtener algún identificador particular y ganar una posición estratégica en la red para ganar control sobre ciertos recursos. Defensa: el identificador depende de algún dato fuera del control de un nodo.

c) Eclipse: El atacante genera una gran cantidad de identidades falsas y colocar esas identidades en la red para mediar en la mayor parte del tráfico y eclipsar los nodos correctos. Autenticación. Defensa: firmas digitales y criptografía de clave pública.

d) Robo de identidad: El nodo malicioso en la ruta de un mensaje afirma que es el nodo de destino deseado, por lo que puede secuestrar solicitudes de ruta y búsqueda para falsificar y destruir datos. Defensa: utilizar pruebas, listas negras.

e) Batir: se generan pares que se unen y abandonan la red lo suficientemente rápido como para corromper la mejor función de la red. Disponibilidad. Defensa: diseño que pueda manejar de manera eficiente la gran cantidad de pares que se unen al sistema durante solo unos minutos.

f) Repetición: una transmisión de datos válida es maliciosa o fraudulentamente repetida, un adversario intercepta la información y la retransmite, posiblemente como parte de un ataque enmascarado. Integridad, confidencialidad, autenticación. Defensa: uso de la marca de tiempo para evitar que el P2P inalámbrico sufra ataques malintencionados.

g) Adivinación: El atacante intercepta mensajes de dos usuarios legales para obtener la contraseña de un usuario. Confidencialidad, autenticación. Defensa: la contraseña de un usuario se verifica localmente y no es necesario transmitirla en el canal.

h) Caché de datos: Aunque los cachés ofrecen un aumento del rendimiento, pueden ser explotados por atacantes y crear un degradamiento de la red.

i) Amplificación de tráfico: magnifica el efecto de un solo anfitrión atacante, un solo anfitrión atacante aparece como múltiples anfitriones y puede tirar redes enteras. Disponibilidad.

A nivel aplicación

a) Racional: El nodo intentará maximizar su consumo de recursos del sistema mientras minimiza el uso de los suyos. Disponibilidad. Defensa: sistema de trueque por fragmentos de datos, cuanto más comparta un nodo con otros, más recibirá.

b) Almacenamiento y recuperación: los usuarios malintencionados se niegan a prestar servicios a los demás nodos o niegan la existencia de datos de los que eran responsables. Disponibilidad, no repudio. Defensa: El sistema debe garantizar la replicación, de manera que ningún nodo sea responsable de facilitar el acceso a las réplicas.

c) Envenenamiento: Inyecta una gran cantidad de señuelos o archivos que no se pueden leer en la red. Disponibilidad, integridad. Defensa: descargar versiones de esos anuncios y luego intentar determinar si las versiones de descarga están limpias o envenenadas.

d) Contaminación: El atacante corrompe el contenido del archivo compartido, dejándolo inutilizable y reenvía el archivo dañado a otros pares. Disponibilidad, integridad. Defensa: listas negras, cifrado de tráfico, verificación hash, firma de fragmentos y detección sin descargar.

e) Inundación de consultas: El nodo malicioso genera tantas consultas como sea posible para inundar la red y no se puede establecer la sesión de descarga. Disponibilidad. Defensa: después de obtener el número máximo de consultas simplemente elimina el resto de

solicitudes.

f) Información de enrutamiento: El nodo malicioso reenvía las consultas al nodo incorrecto o inexistente y luego el nodo original puede que nunca encuentre el nodo de destino. Disponibilidad.

g) Conducción gratuita: hay pares consume principalmente recursos y produce muy pocos, ya sea no compartir contenido o recursos computacionales.

h) Política, auditoría: Explotan vulnerabilidades en las políticas de la red y los sistemas de auditorías y que no detecten comportamientos extraños. Autenticación. (Washbourne 2015, Ferdous et al. 2007, Lee & Jang 2020, Dinger & Hartenstein 2006, Liang et al. 2006, Chen et al. 2009, Faheem Rasheed 2012)]

Se decidió modelar de manera general un ataque, tomando las características principales y compartidas de varios de ellos, pues su comportamiento y forma de ataque son muy parecidas. De esta manera los parámetros son simples y los hallazgos obtenidos pueden ser ampliados a más de un ataque en este tipo de redes.

Se optó por tomar como base la dispersión de malware o virus dentro de la red, esto puede hacerse de diferentes maneras, a través de ataques de envenenamiento de índice, ataque contaminador, ataque eclipse, entre otros, pero todos causan efectos similares, que es infectar a un gran número de computadoras.

En la figura 1 se puede observar los principales estados tomados en consideración. El nodo o par puede estar expuesto a ser infectado por otro nodo o archivos que descargue de este, puede ser infectado o no dependiendo de otros factores pero una vez que se infecta la manera de cambiar de estado es desconectarse de la red. Los nodos pueden hacer varias acciones, ya sea que estén infectados o no, pueden descargar archivos, pedirlos de otros pares, subir archivos y enviarlos.

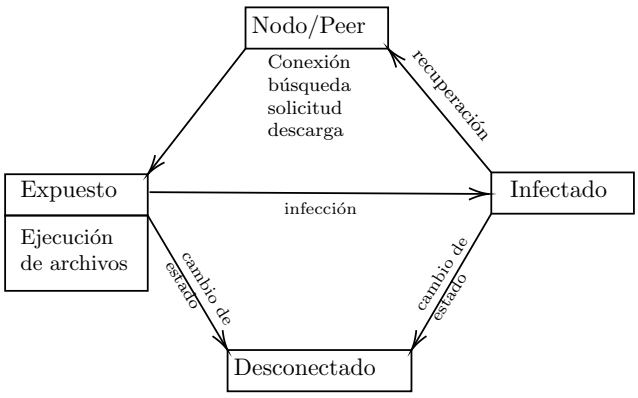

FIGURA 1: Propagación de un virus o malware dentro de la red

2.3. Cadenas de Markov

Las cadenas o series Markov son procesos estocásticos que representan un sistema con un conjunto finito de estados discretos y un conjunto de probabilidades de transición entre estados, generalmente representados como un gráfico dirigido. Cada nodo representa el estado del sistema y el arco direccional representa las posibles transiciones entre estados. A menudo se utilizan para describir un sistema mutuamente excluyente, pero se pueden describir con una sola declaración que describe el sistema en su conjunto (Bocek et al. 2008, Tosun 2005, Suñé Torrents et al. 2017). Algo que es importante señalar es que en una cadena de Markov todos los estados son observables.

Como muestra la literatura (Robayo Santana 2009, Moreno Mogollón et al. 2007) el diseño del modelo requiere dos pasos: en primer lugar, el ajuste del proceso estadístico que es la caracterización del fenómeno y en segundo, seleccionar el modelo (en este caso las cadenas de Markov o en dado caso una de sus variantes) y ajustar el proceso estadístico a ello. Implica resumir el tráfico existente en la red y analizar lo que se sabe que es normal y seguro. En algunos casos, puede optar por utilizar datos de tráfico conocidos para mostrar anomalías y determinar la ubicación del descriptor. Una vez conocidas las características de los datos de tráfico de la red, es necesario extraer los trazos generales (representados por ecuaciones y gráficos) que determinan los datos de tráfico reales y realizan el programa de simulación. A partir de ahí, se realizan pruebas de similitud para verificar el comportamiento generado y no se pueden realizar otras pruebas directamente en el modelo de tráfico de red real.

2.4. Tráfico real

En las redes P2P los archivos residen en los extremos de la red (en los clientes, a los que ahora se denomina pares, por su doble condición de clientes y servidores), y sólo en algunas arquitecturas se utilizan "servidores", que actúan como meros índices de los archivos, y cuyo rol es facilitar que los clientes puedan encontrarse entre sí, y no almacenar los archivos, como sucedería en una arquitectura cliente-servidor convencional.

Dado el esquema de comunicaciones de las aplicaciones P2P, el tráfico que inducen se caracteriza por su simetría en entornos cerrados de usuarios, puesto que todo el tráfico que reciba un usuario (par) determinado habrá sido inyectado por otros usuarios del grupo. Esta simetría supone que, a largo plazo, el tráfico que el usuario inyecta en la red, es similar al tráfico que el usuario recibe de la misma (Rivero-Angeles & Rubino 2010).

3. PLANTEAMIENTO DEL PROBLEMA

Las redes pares a pares son muy un sistema con principios de funcionamiento simples, su costo de mantenimiento es menor al no contar con servidores centralizados y por eso mismo muy populares, sobre todo en los servicios de transmisión de datos y compartición de archivos de cualquier tipo. Es por eso son ideales para la difusión a una gran cantidad de computadoras. Muchos de estos archivos que se suben a la red y se comparten pueden ser hechos de manera intencionada para infectar a otras computadoras. Por medio de este trabajo se investiga el alcance que podría tener este tipo de archivos corruptos a través de una red P2P, así como la importancia de los factores involucrados en el funcionamiento de la red. Para nuestro caso de estudio desarrollamos la simulación de un modelo que emula de manera sencilla el comportamiento de un ataque simple a una red de pares a pares, por lo que se escogió las cadenas de Markov como herramienta para modelar este tipo de redes y su comportamiento bajo un ciberataque general y conocido ampliamente.

4. JUSTIFICACIÓN

El primer modelo de comunicación utilizado en Internet fue el modelo cliente-servidor, sencillo de implementar pero con un gran problema de escalabilidad, que conlleva aumentar el número de usuarios la calidad del servicio ofrecido disminuye. Como solución a este problema de escalabilidad apareció el modelo P2P. Los ejemplos de redes P2P, como BitTorrent o Skype, entre otras. Estas redes representan actualmente un porcentaje considerable de todo el ancho de banda mundial. Sin embargo, este nuevo modelo de comunicaciones posee también un claro inconveniente que representa una multitud de vulnerabilidades y amenazas a la seguridad. La naturaleza de la propia filosofía P2P implica que no existe ningún servidor centralizado que se encarga de subir con a almacenar y comprobar la autenticidad de los recursos. Una consecuencia directa de ello es que no hay ningún mecanismo para controlar qué contenido se comparte esto hace que las redes P2P sean óptimas para la propagación de malware en general. Para ello se busca realizar un mecanismo de análisis que ayude a controlar el tráfico de las redes, analizando la cantidad de nodos infecciosos que pueden causarle daño a la red. Esto tendría un impacto al proporcionar información del comportamiento esperado de la red bajo ciertas condiciones, de manera que se puedan conocer estos datos rápidamente y considerar los diferentes escenarios posibles. Esto es aplicable en la utilización de este tipo de redes para la creación de nuevos servicios o la mejora de algunos ya existentes.

5. HIPÓTESIS

Para este trabajo se pretende presentar un modelo simplificado de una red P2P de manera que se compruebe su correcto funcionamiento para el enlace archivos en continua transmisión, haciendo uso del esquema de cadenas de Markov, además de comprender algunos aspectos del comportamiento de dicho esquema. Para fines de implementación, se usará un esquema similar a la aplicación BitTorrent, donde se tienen dos tipos de usuarios, unos con archivos completos y otros que únicamente guardan trozos de distintos archivos.

6. OBJETIVOS GENERALES

Desarrollar un modelo simplificado de una red P2P para poder verificar su rendimiento bajo ese primer esquema simple y entender su comportamiento bajo las condiciones de un ciber-ataque, pudiendo robustecer el modelo tras la identificación de sus debilidades en las simulación de un ataque dentro o hacia la red.

6.1. Objetivos particulares

- Analizar el modelo generado utilizando las cadenas de Markov relacionadas con ciber-ataques en redes P2P.
- Obtener medidas de desempeño de la red P2P bajo un ciber-ataque incluyendo:
  - Impacto del ataque
  - Impacto en el desempeño de la red
  - Retardo de dispersión del ataque.
- Mejorar el modelo propuesto al identificar variables adicionales que modelen de forma más precisa un ciber-ataque.

7. RECURSOS

7.1. Materiales

- Computadora personal

7.2. Software

- Anaconda
- Jupyter Notebook
- Python

8. MÉTODOS

8.1. Aproximación por cadenas de Markov al modelado de la red de pares a pares

8.1.1. Cadena simple de Markov

Esta primera cadena simple, simula el comportamiento de una red de pares a pares donde existen dos tipos de usuarios, sanguijuelas y semillas.

Para el desarrollo de este modelo se toman en cuenta dos escenarios de condiciones; Penuria, donde hay

escasez de usuarios y el comportamiento esta dato por la ecuación $M(Nx + y)$, donde M significa la tasa de subida de archivos, N la tasa de bajada, x los usuarios con trozos de archivos y y usuarios con los archivos completos a los que es más probable enviar peticiones para la descarga de sus archivos. Y la Abundancia, donde aumenta la cantidad de usuarios y por lo tanto también los recursos, representándose con la ecuación Cx, siendo C la tasa de baja de archivos en estos casos.

Otras variables a tomar en cuenta son la tasa de arribos (L), es decir cuántos usuarios llegan por segundo y los tiempos de conexión de los usuarios H para las sanguijuelas y G para las semillas.

Esta serie de variables se relacionan utilizando las siguientes ecuaciones:

$$\tau = \min(M(Nx + y), Cx)$$

$$T_1 = -\frac{1}{L}\log(1-u)$$

$$T_2 = -\frac{1}{Hx}\log(1-u)$$

$$T_3 = -\frac{1}{Gy}\log(1-u)$$

$$T_4 = -\frac{1}{\tau}\log(1-u)$$

$$T = \min(T_1, T_2, T_3, T_4)$$

Cada una de estas ecuaciones representa un cambio de estado en la cadena, cada uno de ellos recibe un valor en cada iteración realizada y aquel con el valor mínimo es el que determina a qué estado se hace la transición. Estos cambios son guardados dentro de una matriz para saber cuánto tiempo se permanece dentro de un estado. Algo importante para generar este cambio de estados es un número semi-aleatorio entre cero y uno con distribución uniforme que influirá en el peso que tendrán estas ecuaciones a cada paso.

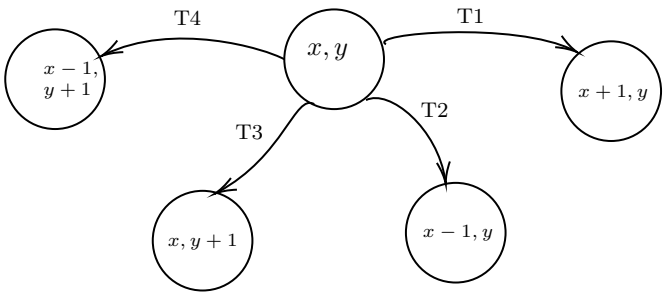

FIGURA 2: Modelado utilizando una cadena simple de Markov

Esta cadena es una cadena de Markov irreducible porque desde cualquier estado se puede acceder a cualquier otro y todos los estados se comunican entre sí. Los sanguijuelas son usuarios que tienen partes de archivos o ningún dato y los sedes son aquellos pares que ya han descargado el archivo completo y esperan en el sistema para compartir sus recursos. Ambos cooperan para llevar archivos a otros leeches o sanguijuelas. Entre más usuarios con archivos completos se encuentren dentro de la red, los tiempos de descarga serán más ágiles permitiendo que el tráfico y el ancho de banda aumenten. Esto sin embargo debe de alcanzar un límite en algún punto, que es cuando el tráfico alcanza la estabilidad y la red se mantiene en movimiento sin grandes cambios en el número de usuarios y es el tipo de comportamiento que se desea obtener.

8.1.2. Cadena de Markov con nodos iniciales infectados

$$X_n \rightarrow \text{sanguijuelas no infectadas}$$

$$X_n \rightarrow \text{sanguijuelas infectadas}$$

$$Y_n \rightarrow \text{semillas no infectadas}$$

$$Y_i \rightarrow \text{semillas infectadas}$$

1. Las sanguijuelas no están infectados
2. Si un par entra en contacto con otro par infectado se infecta. Tasa de infección = 1.
3. $X_i \geq N_i \rightarrow$ Debe haber al menos un par malicioso
4. $N_i \rightarrow$ número inicial de nodos maliciosos
5. No hay transiciones de $X_i \rightarrow X_n$ ya que si está infectado, ya no se puede desinfectar.
6. El nodo ya está infectado $\rightarrow$ no importa de donde descarga, se pasa a semilla infectada.

$$X = X_n + X_i$$

$$Y = Y_n + Y_i$$

$$\tau_i = \text{máx}\left\{CX_i, (X\eta + Y)\,\frac{X_i}{X}\right\}$$

7. Sólo los pares (sanguijuelas y semillas) infectados.
8. Si el peer ya está infectado $\rightarrow$ se mantiene infectado.
9. $\tau_{nn} \rightarrow$ Indica si el par actual no se infectaría.
10. $\tau_{ni} \rightarrow$ Indica la descarga de los pares, pero al menos uno de ellos debe de estar contaminado.

$$\tau_{nn} = \text{mín}\left\{CP_nX_n, \mu\left(X_n\eta + Y_n\right)\right\}$$

$$\tau_{ni} = \text{mín}\left\{C(1-P_n)X_i, \mu\left(X_i\eta + Y_n\right)(1-P_n)\right\}$$

11. $P_n \rightarrow$ Probabilidad de no infectarse
12. El ancho de banda total es: $\mu(X\eta + Y)$
13. El ancho de banda infectado es: $\mu(X_i\eta + Y_i)$
14. El ancho de banda total es: $\mu(X_n\eta + Y_n)$

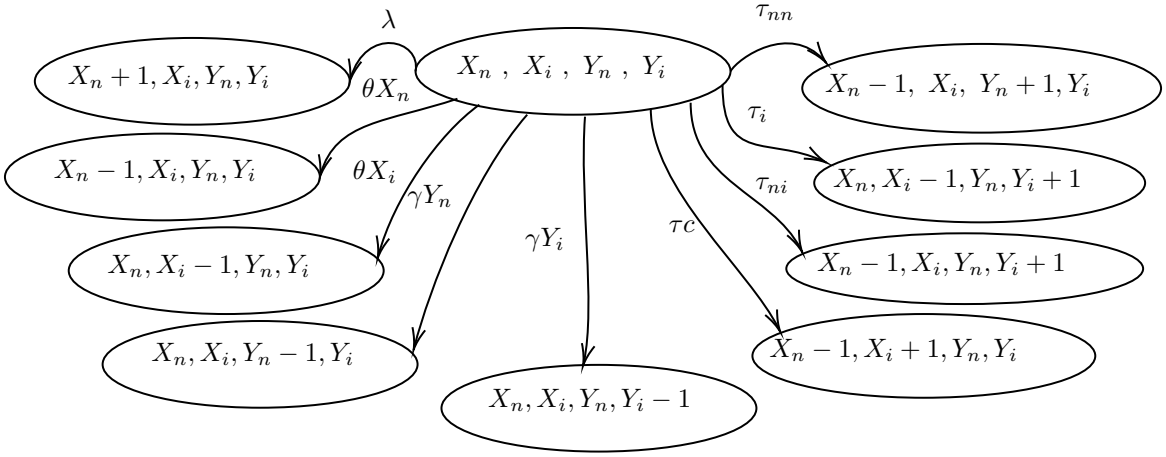

FIGURA 3: Diagrama de los posibles estados de la cadena infectada

$$\mu X_i \eta + \mu Y_i + \mu X_n \eta + \mu Y_n$$
$$= \mu\left[\eta(X_i + X_n) + Y_i + Y_n\right] = \mu\left[\eta X + Y\right]$$
$$\Rightarrow P_n = \frac{\mu(X_n\eta + Y_n)}{\mu(X\eta + Y)} = \frac{X_n\eta + Y_n}{X\eta + Y}$$

El número de pares infectados con los que se puede conectar:

$$\eta X_i + Y_i$$

El número de pares no infectados con los que se puede conectar:

$$\eta X_n + Y_n$$

La forma de conectarse con nodos no infectados:

$$P_n = \frac{X_n\eta + Y_n}{X\eta + Y}$$

13. $\tau_C \rightarrow$ Tasa con la que una sanguijuela se contamina.
    Se puede contaminar cada pedazo descargado. El archivo es de tamaño F (F: 1 normalizado). El tamaño del pedazo es B. El archivo tiene $K = \frac{F}{B}$ chuks.

    $$\tau_C = \min\left\{CK(1 - P_n)X_n, \mu K(\eta X + Y)(1 - P_n)\right\}$$

14. $CK, \mu K \rightarrow$ La tasa de descarga de un pedazo es K veces más rápida que la del archivo total.
15. $1 - P_n \rightarrow$ Probabilidad de infectarse

Estas operaciones determinan el comportamiento de los pares o usuarios dentro de la red pues determinan la probabilidad de cambiar de estado según las condiciones que se tengan en ese momento determinado.

8.1.3. Cadena de Markov con nodos iniciales infectados y un parámetro como contramedida

$P_I \rightarrow$ Probabilidad de que un nodo NO infectado que descarga de un nodo Infectado SÍ se contamine.

Este nuevo parámetro que se agrega a esta última aproximación representa aquellas medidas preventivas que se pueden tomar para que cuando un nodo sano entre en contacto con un nodo malicioso la probabilidad de contagio no sea del 100 %. Un ejemplo de esta clase de medidas pueden ser los antivirus o los protocolos de verificación de procedencia del archivo o protocolos de cuarentena de archivos de los que no se sabe si se pueden confiar.

Mientras más cercano a 1 se halle este parámetro mayor será la probabilidad de contagio, y si se encuentra más cerca del cero, esto indica que las medidas asumidas son de mayor efectividad.

8.2. Experimentos de simulación: resolución de la cadena de Markov propuesta

El software utilizado para desarrollar la simulación de las cadenas descritas anteriormente, se hizo a través de Python utilizando el manejador de paquetes Anaconda y la interfaz de Jupyter Notebook. Otros de los paquetes dentro de Python que son importantes para el funcionamiento del sistema son: random, math, numpy, matplotlib, pandas.

8.2.1. Cadena simple de Markov

Para la cadena de Markov se tomaron cada uno de los parámetros explicados en la tabla 1

De manera que los valores tomados por los parámetros anteriores estuvieran entre 0 y 1, combinándolos de manera que después de varias iteraciones se pudiera observar un comportamiento convergente por parte de la simulación. Además de los parámetros anteriores también denotamos a la cantidad de semillas como x y a la cantidad de sanguijuelas como y. Estos parámetros se pueden inicializar en cualquier número mientras ha-

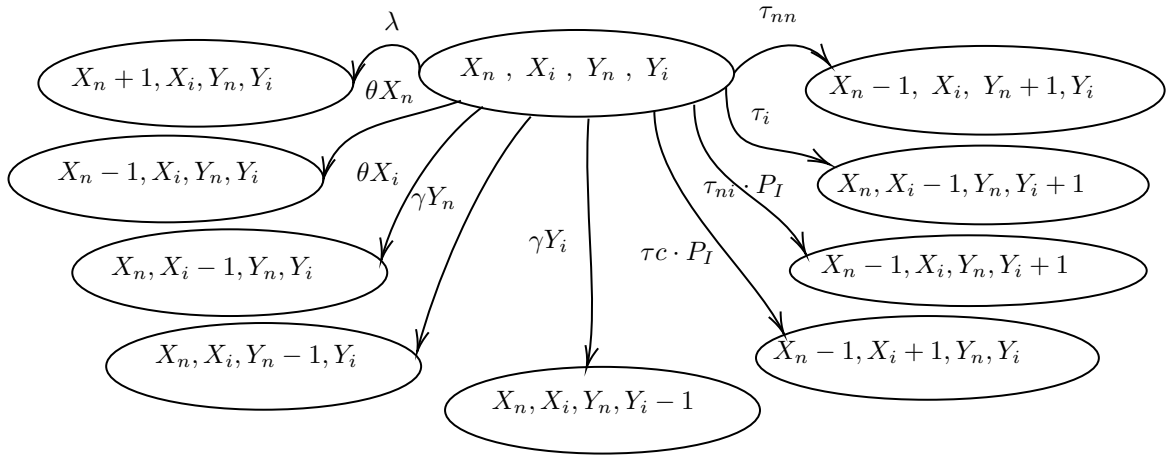

FIGURA 4: Diagrama de los posibles estados de la cadena con un factor de contramedida al ataque cibernético

Parámetro	**Descripción**	**Unidades**
C	Tasa de bajada	archivos/seg
L	Tasa de arribos	usuarios/seg
M	Tasa de subida	archivos/seg
H	Tiempo de conexión de las sanguijuelas	seg
G	Tiempo de conexión de las semillas	seg
N	Tasa de bajada	archivos/seg

CUADRO 1: Parámetros utilizados en la cadena simple

Algoritmo 1: Cadena simple

```
x=0;
y=1;
for i=1 to 100000 do
    tau = min(M·(N·x+y),C·x);
    T1 = -(1/L) · ln(1-u());
    T2 = -(1/(H·x)) · ln(1-u());
    T3 = -(1/(G·y)) · ln(1-u());
    T4 = -(1/tau) · ln(1-u());
    T = min(T1,T2,T3,T4);
    ;
    if T=T1 then
        x+=1 ;
    else if T=T2 then
        x-=1 ;
    else if T=T3 then
        x-=1 ;
        y-=1 ;
    else
        x+=1 ;
        y+=1 ;
    end
```

Resultado: Número de x y y en total y a cada paso

ya más de una sanguijuela. En este caso se inicializaron como 0 y 1 respectivamente.

El pseudocódigo 1 representa los estados que se presentaron en la figura 2 de una forma más similar a lo que se escribiría en el software a utilizar (en este caso Python).

Una vez que se tienen el número total de pares (semillas y sanguijuelas) se puede graficar su comportamiento a través de cada una de las iteraciones hechas para poder observar y determinar si sus valores se mantuvieron estables dentro de un rango y el promedio de dicha estabilización.

8.2.2. Cadena de Markov con nodos iniciales infectados

En la figura 5 se puede ver un ejemplo de una cadena de Markov que toma en cuenta dos estados, el susceptible y el infectados. Estos mismos estados son los que se consideran para la simulación de la cadena de Markov con nodos iniciales infectados.

dos estados.png

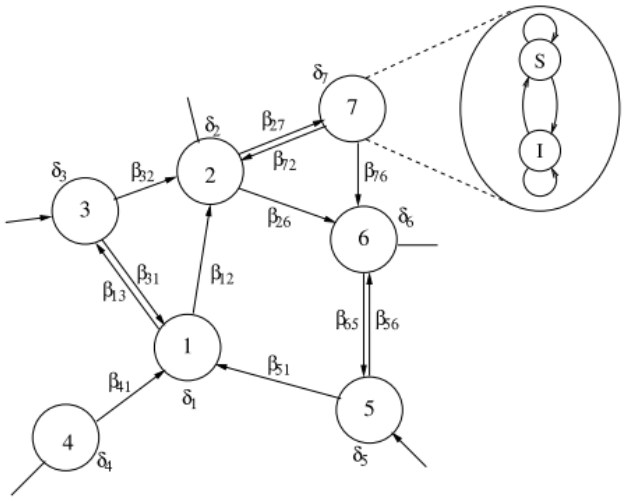

FIGURA 5: Ejemplo de una red considerando dos estados

Parámetro	**Descripción**	**Unidades**
K	Pedazos de un archivo	tamaño/divisiones
N_i	Nodos infectados	pares (semillas o sanguijuelas)

CUADRO 2: Parámetros agregados a la cadena con nodos infectados

Para la cadena de Markov con nodos iniciales infectados se tomaron los parámetros de la cadena simple más los parámetros explicados en la tabla 2

Los valores obtenidos de la cadena simple son utilizados como referencia en esta aproximación e igualmente son ajustados ligeramente para que el comportamiento de las cuatro gráficas en este caso, siga un comportamiento similar al de la cadena simple desarrollada en el paso anterior. A esta aproximación se le agregan además los parámetros descritos en la tabla 2, que nos ayudaran a describir y agregar estados como se muestra en la figura 4. Además de los parámetros anteriores también denotamos a la cantidad de semillas sanas como xn, a la cantidad de sanguijuelas sanas como yn, a las semillas infectadas xi y a la cantidad de sanguijuelas infectadas como yi. Estos parámetros se pueden inicializar en cualquier número mientras haya más de una sanguijuela sana y una sanguijuela infectada. En este caso se inicializaron las semillas con el valor de 0 y a las sanguijuelas con el valor de 1.

El pseudocódigo 2 representa los estados que se presentaron en la figura 3 de una forma más similar a lo que se escribiría en Python.

Para esta cadena además se variaron tres distintos parámetros en cierto rango. Estos parámetros son G que representa el tiempo de conexión de las semillas, N que indica la tasa de bajada o de descarga y L que es la tasa de nuevos usuarios que se conectan a la red por segundo. Se hicieron 3 experimentos distintos para variar cada uno de los parámetros mencionado contra el número de nodos infectados (Ni) de manera inicial con el fin de obtener como resultado el comportamiento de los pares ya fueran sanos o infectados y determinar qué valor o rango de valores favorecía o perjudicaba el crecimiento de los nodos infectados, buscando a su vez llegar a un estado estable. El valor de K se estableció en 0.1, aunque esto variaría en un sistema real dependiendo del tamaño y tipo de archivo.

Para cada uno de estos parámetros (G, N y L) se tomó en cuenta un rango de 0.2 a 1 con un paso de 0.1. Y para los nodos iniciales maliciosos se inició desde 1 hasta 10 nodos.

8.2.3. Cadena de Markov con nodos iniciales infectados y un parámetro como contramedida

A partir de los resultados obtenidos de la cadena anterior se determinó que los mejores parámetros para el tiempo de conexión de las semillas (G) sea de 1, la tasa de descarga (N) tenga un valor de 0.85 y la tasa de nuevos usuarios(L) sea igualmente de 1.

Para esta aproximación se realizó de manera similar al experimento anterior, una comparación entre el número de nodos infectado iniciales y el PI o la probabilidad de que un nodo sano que descarga archivos de un nodo infectado llegue a contaminarse, observando de que manera afecta el comportamiento de las semillas y sanguijuelas, ya sea sanas o infectadas.

En este experimento se extendió el rango de nodos infectados iniciales, se comenzó desde 1 hasta 14 nodos y para la tasa de infección se llevó a cabo dentro del rango del 0 al 1 con pasos de 0.01.

En este nuevo algoritmo (3) se puede ver que el nuevo parámetro PI se multiplica por los factores el estado 6 y 7 pues estos estados afectan el crecimiento de los pares infectados.

Algoritmo 2: Cadena con nodos infectados

```
xn=0;
yn=1;
xi=0;
yi=1;
for i=1 to 100000 do
    Pn = (xn·M+yn)/(xn+(yn+yi));
    tauni = min(C·(1-Pn)·xi,M·(N·xn+yn)·(1-Pn));
    taunn = min(C·Pn·xn,M·(N*xn+yn));
    taui =
     min(C·xi,((xn+xi)·N+(yn+yi)·(xi/(xn+xi))));
    tauC = min(C·K·(1-
     Pn)·xn,M·K·(M·(xn+xi)+(yn+yi))·(1-Pn));

    T1 = -(1/L) · ln(1-u());
    T2 = -(1/(H·xn)) · ln(1-u());
    T3 = -(1/(H·xi)) · ln(1-u());
    T4 = -(1/G·yn) · ln(1-u());
    T5 = -(1/(G·yi)) · ln(1-u());
    T6 = -(1/(tauC)) · ln(1-u());
    T7 = -(1/(tauni)) · ln(1-u());
    T8 = -(1/taui) · ln(1-u());
    T9 = -(1/taunn) · ln(1-u());
    T = min(T1,T2,T3,T4,T5,T6,T7,T8,T9);
    if T=T1 then
        xn += 1 ;
    else if T=T2 then
        xn -= 1 ;
    else if T=T3 then
        xi -= 1 ;
    else if T=T4 then
        yn -= 1 ;
    else if T=T5 then
        yi -= 1 ;
    else if T=T6 then
        xn -= 1 ;
        xi += 1 ;
    else if T=T7 then
        xn -= 1 ;
        yi += 1 ;
    else if T=T8 then
        xi -= 1 ;
        yi += 1 ;
    else
        xn -= 1 ;
        yn += 1 ;
    end
```

Resultado: Número de xn, xi, yn y yi en total y a cada paso

Algoritmo 3: Cadena agregando un parámetro como contramedida

```
for Ni=1 to 15 do
    for PI=0 to 1 do
        for i=1 to 100000 do
            Pn = (xn·M+yn)/(xn+(yn+yi));
            tauni =
             min(C·(1-Pn)·xi,M·(N·xn+yn)·(1-Pn));
            taunn = min(C·Pn·xn,M·(N*xn+yn));
            taui =
             min(C·xi,((xn+xi)·N+(yn+yi)·(xi/(xn+xi))));

            tauC = min(C·K·(1-
             Pn)·xn,M·K·(M·(xn+xi)+(yn+yi))·(1-
             Pn));

            T1 = -(1/L) · ln(1-u());
            T2 = -(1/(H·xn)) · ln(1-u());
            T3 = -(1/(H·xi)) · ln(1-u());
            T4 = -(1/G·yn) · ln(1-u());
            T5 = -(1/(G·yi)) · ln(1-u());
            T6 = -(1/(tauC·PI)) · ln(1-u());
            T7 = -(1/(tauni·PI)) · ln(1-u());
            T8 = -(1/taui) · ln(1-u());
            T9 = -(1/taunn) · ln(1-u());
            T =
             min(T1,T2,T3,T4,T5,T6,T7,T8,T9);
            if T=T1 then
                xn += 1 ;
            else if T=T2 then
                xn -= 1 ;
            else if T=T3 then
                xi -= 1 ;
            else if T=T4 then
                yn -= 1 ;
            else if T=T5 then
                yi -= 1 ;
            else if T=T6 then
                xn -= 1 ;
                xi += 1 ;
            else if T=T7 then
                xn -= 1 ;
                yi += 1 ;
            else if T=T8 then
                xi -= 1 ;
                yi += 1 ;
            else
                xn -= 1 ;
                yn += 1 ;
            end
```

Resultado: Número de xn, xi, yn y yi en total y a cada paso

9. RESULTADOS

9.0.1. Cadena simple de Markov

Durante la realización de los experimentos los valores de los parámetros iniciales de la cadena o sus condiciones se fueron variando dentro del rango de 0 a 1, los valores que mejor resuelven la cadena y con los que se obtienen resultados estables son los siguientes:

C = 0.02 (tasa de bajada, archivos/seg)
L = 1 (tasa de arribos, usuarios/seg)
M = 0.00125 (tasa de subida, archivos/seg)
H = 0.01 (tiempo de conexión sanguijuelas, seg)
G = 0.01 (tiempo de conexión semillas, seg)
N = 0.85 (tasa de bajada, archivos/seg)

Como se puede ver en la figura 6, sobre todo en la primera gráfica que representan a las sanguijuelas dentro de la red, se ha alcanzado cierta estabilidad haciendo uso de este modelo, lo cual indica que la relación entre las variables fue correcta, sin embargo al ser poco el tiempo de simulación (25 minutos en total, con 100,000 iteraciones) así como escasas las condiciones iniciales el estado máximo de usuarios tampoco fue muy alto, sobre todo en los usuarios que actuaban como semillas o seeds.

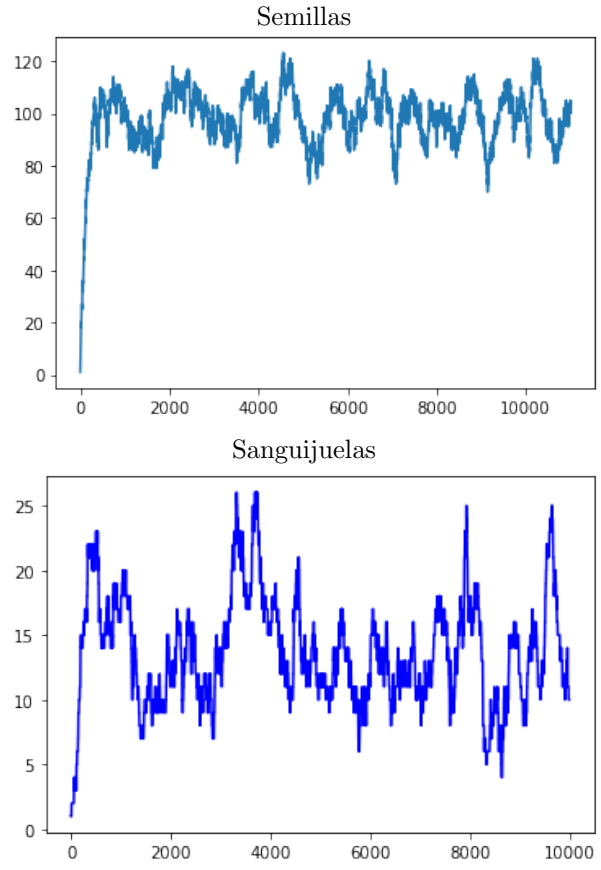

FIGURA 6: Resultados cadena simple

Para esta primera aproximación el promedio de sanguijuelas obtenido fue de 100 y el promedio de semillas fue de 16 como se muestra en la tabla 3.

Promedio en 100000 iteraciones	
Sanguijuelas	**Semillas**
100 ± 15	16 ± 3
Tiempo aproximado: 25 minutos	

CUADRO 3: Promedios y tiempo de simulación de la cadena de Markov simple

9.0.2. Cadena de Markov con nodos iniciales infectados

Al realizar los experimentos se usaron los valores obtenidos de la implementación de la cadena simple, aunque aun así estos valores debieron ajustarse y reducirse un poco para poder continuar con la simulación de la segunda aproximación

Esta simulación se efectuó en aproximadamente 40 minutos, los promedios obtenidos se muestran en la tabla

Promedio en 100000 iteraciones			
Sanguijuelas sanas	**Semillas sanas**	**Sanguijuelas infectadas**	**Semillas infectadas**
33 ± 8	2 ± 1	10 ± 5	2 ± 2
Tiempo aproximado: 37 minutos			

CUADRO 4: Promedios y tiempo de simulación de la cadena de Markov con nodos maliciosos.

Después de realizar ajustes en los valores anteriores lo que finalmente resuelven mejor la cadena en esta segunda aproximación y con los que se obtienen resultados estables son los siguientes:

C = 0.002 (tasa de bajada, archivos/seg)
L = 1 (tasa de arribos, usuarios/seg)
M = 0.05 (tasa de subida, archivos/seg)
H = 0.005 (tiempo de conexión sanguijuelas, seg)
G = 0.3 (tiempo de conexión semillas, seg)
N = 0.85 (tasa de bajada, archivos/seg)
K = 1/10 (longitud de archivos, normalizado)

La figura 7 muestra los resultados que corresponden a esta tabla (4) donde se observar la evolución y crecimiento de los nodos así como su permanencia a lo largo de la simulación. Esta es una de las diferentes gráficas obtenidas pues se pudo notar que incluso una pequeña variación en los parámetros iniciales podía hacer que la gráfica se comportara de maneras muy distintas.

Finalmente, en la segunda parte del experimento realizado con la caracterización del sistema con nodos maliciosos iniciales, se puede observar en las figuras 8, 9 y 10, un mapeo del comportamiento de los pares en

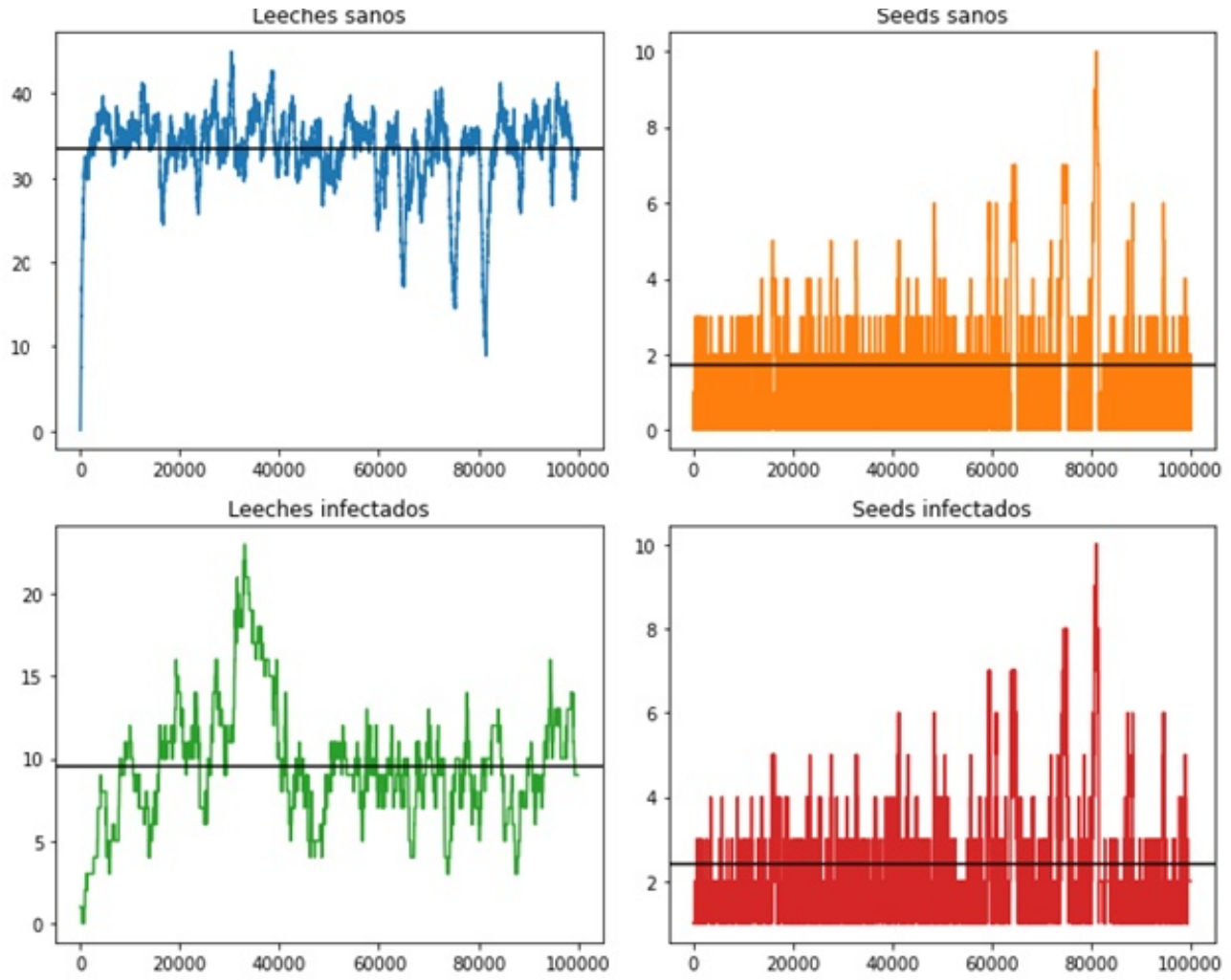

FIGURA 7: Gráficas de los resultados obtenidos para la cadena infectada variando los parámetros iniciales

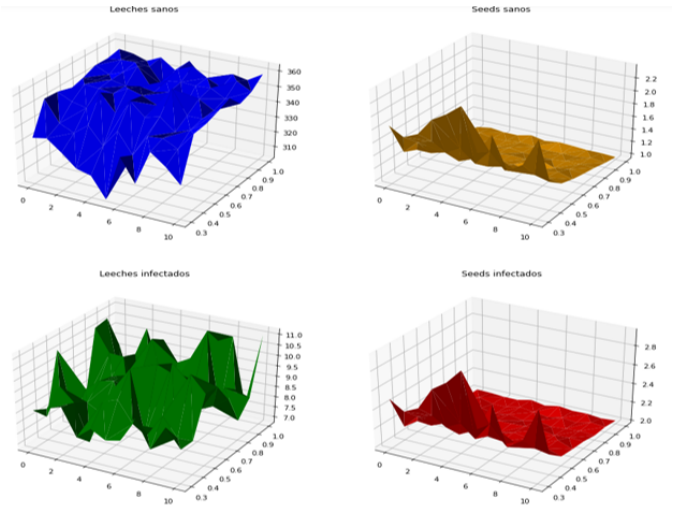

FIGURA 8: Variación de nodos infectados y γ

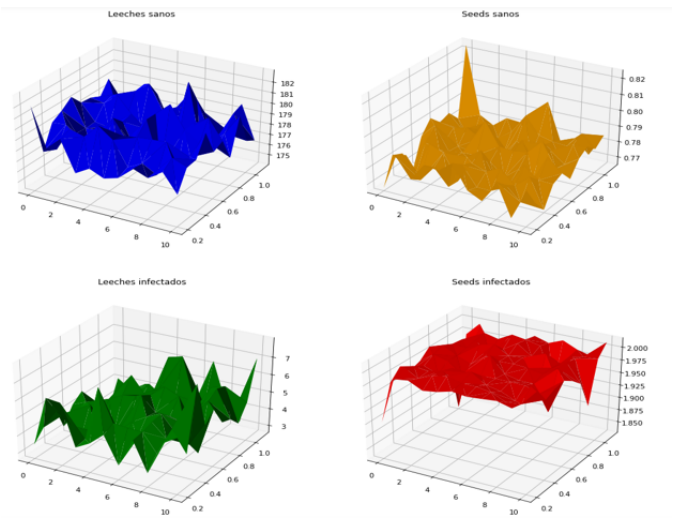

FIGURA 9: Variación de nodos infectados y η

la red.

Estas gráficas representan la alza y caída de pares dentro de la red P2P a lo largo de los rangos especificados anteriormente en la sección de experimentación

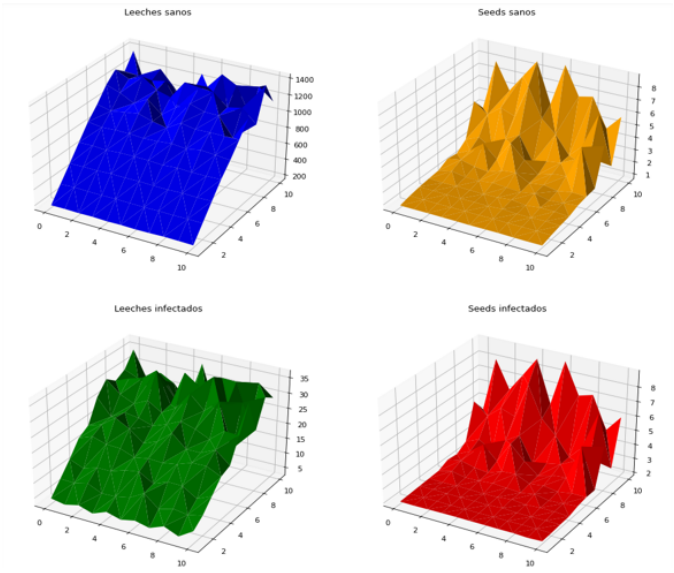

FIGURA 10: Variación de nodos infectados y λ

dentro de la metodología, siendo estos de 0.1 a 1. La primera figura de ellas, la número 8 representa al parámetro G o γ, que es el tiempo de conexión de las semillas. La siguiente figura, número 9, representa al parámetro N o η, que se refiere a la tasa de descarga de archivos dentro de la red. Y por último, la figura 10 se refiere al parámetro L o λ que es la tasa de nuevos usuarios en la red por segundo.

9.0.3. Cadena de Markov con nodos iniciales infectados y un parámetro como contramedida

La simulación de esta última aproximación, donde se agregó un parámetro representando la probabilidad de infectarse en base a medidas preventivas, se efectuó en 1 hora y 33 minutos, lo cual supero los tiempos de ejecución anteriores. Uno de los motivos de esto fue que el rango de nodos infecciosos iniciales esta vez fue de 1 a 15 nodos y el rango para la probabilidad de infección fue de 0 a 1, similar a los rangos de los parámetros en la aproximación anterior, pero esta vez el paso para cada iteración fue de 0.01.

En las gráficas producidas (figura 11), se puede apreciar el efecto de un rango más amplio y un paso menor, al producir gráficas que son menor abruptas, aún así no tienen una superficie suavizada del todo.

10. ANÁLISIS DE RESULTADOS

A través de los resultados obtenidos en los experimentos y simulaciones realizados se pudo notar de manera bastante clara el efecto de mover aumentar o disminuir el valor de los parámetros iniciales del sistema que emulan los tiempos de conexión de los usuarios y las tasas de subida y descarga de archivos dentro de la red P2P.

Los resultados sugieren que los valores tomados para la resolución de la cadena de Markov en el sistema simple donde solo se simula la red P2P y el tráfico normal, son adecuados para ese tipo de sistema y variaciones en números cercanos a los finalmente tomados provocaban ligeras alteraciones pero sin afectar en casi nada al sistema.

Conforme se continúan agregando parámetros y variables, sus valores comienzan a ser más sensibles a los cambios en ellos, aunque sea mínimos y se alterar de manera más significativa el sistema, debido a eso es que se tuvieron que reajustar algunos de los valores tomados anteriormente.

Observando el mapeo de los pares en las gráficas 3D en la segunda aproximación, se puede notar que hay una mayor relación en cuestión de comportamiento en sanguijuelas, sanas e infectadas y a su vez entre semillas sanas e infectadas. El parámetro que parece ser más sensible a los cambios es λ o L, sobre todo traspasando la mitad del rango ocupado en la simulación. El siguiente parámetro más reactivo a las variaciones el parámetro γ o G, donde se ve un aumento en las semillas cuando sus valores son pequeños y un comportamiento bastante más irregular para las sanguijuelas.

Para la última aproximación tomada, que es la simulación de la cadena infectada más contramedidas, se tiene que el parámetro PI (probabilidad de infección) causa un cambio muy notorio en las cuatro gráficas. Tres de ellas tienden a subir conforme la probabilidad de infección aumenta, esto se observa sobre todo en el conteo de pares infectados, mientras que las semillas sanas, aunque aumentan, lo hacen en una menor proporción y por el contrario el conteo de sanguijuelas sanas baja drásticamente. Esto concuerda con lo encontrado en la literatura (Fan & Xiang 2010, Grishunina et al. 2017), coincidiendo también con algunos modelos epidemiológicos utilizados. Por el otro lado, aunque si se alcanza a ver un aumento en la cantidad de usuarios infectados al aumentar los nodos infecciosos iniciales, no es un aumento demasiado grande o notorio, por lo que dentro de esta escala y rango, su efecto podría despreciarse.

11. CONCLUSIÓN

En este trabajo se detallan los resultados que se obtienen al analizar en términos de una cadena de Markov, el comportamiento de las redes de pares a pares de una manera sencilla en tres escenario distintos. El primero, el funcionamiento normal de una red P2P, donde se observó que después de 100,000 iteraciones ya se llega a un estado estable. En segundo, la red ya probada que converge a un estado se le aumentan

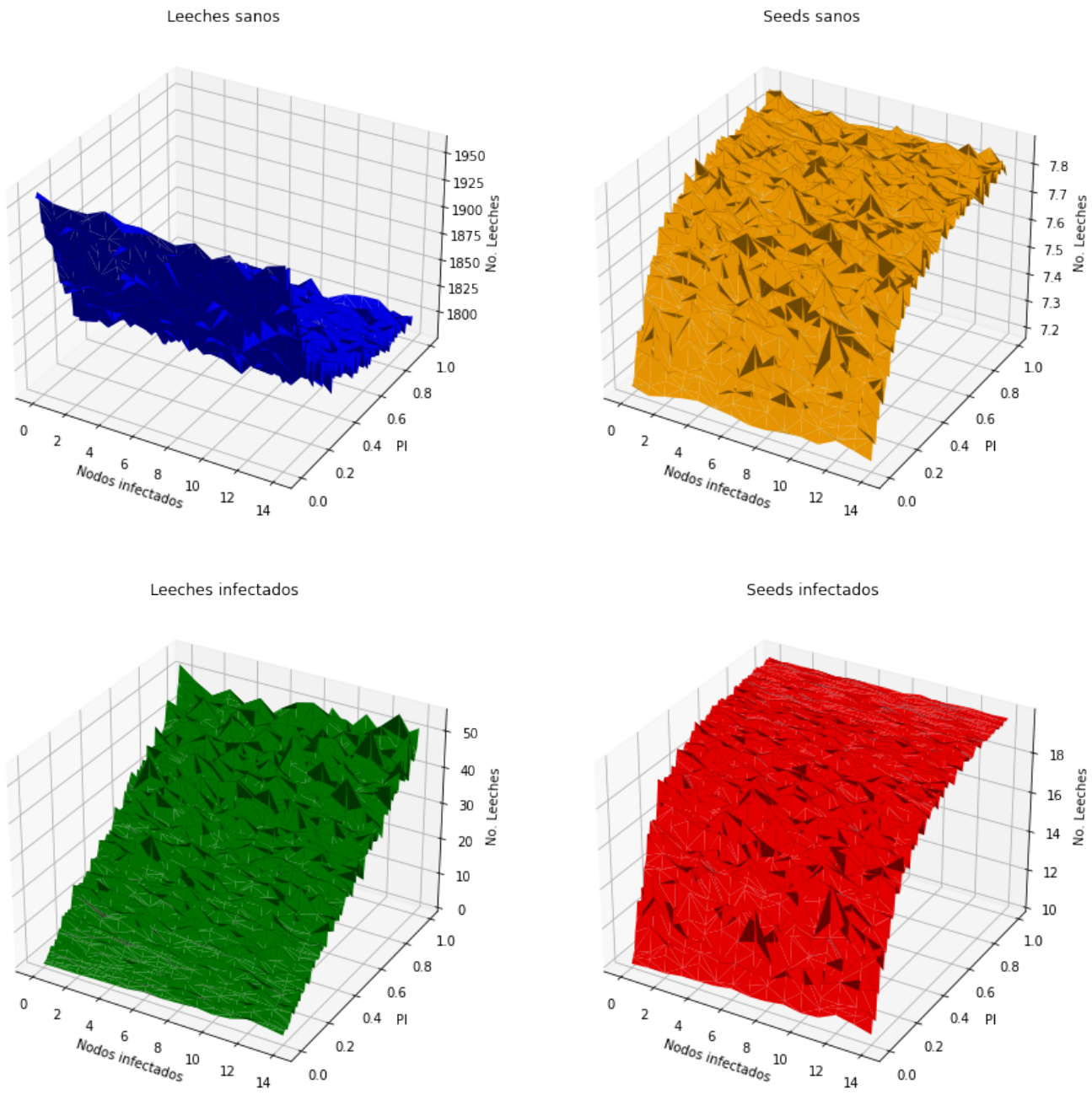

FIGURA 11: Resultados de la resolución de la cadena con PI y Ni variables.

parámetros, esto ocasiona el surgimiento de un nuevo tipo de pares infecciosos invadiendo el sistema, desestabilizando la red y afectando el posible crecimiento o conexión de los pares sanos. Finalmente, se adicionó un parámetro para conocer el impacto de posibles medidas de seguridad, ya sea a nivel de red o a nivel usuario.

A través de lo observado dentro del análisis del sistema, se puede concluir que el parámetro Ni no es tan significativo para el comportamiento general, al menos dentro del rango probado que en realidad fue bastante corto.

De acuerdo a los objetivos planteados, se logró completar el esquema simple del sistema, analizando su comportamiento bajo distintas condiciones, identificando en cuales de estas condiciones la red se encuentra más vulnerable y cuál es la causa de la debilidad.

Nuestra mayor contribución es proponer un sistema de modelado simple, para conocer la propagación de virus o malware a través de una red P2P, utilizando un método estocástico como son las cadenas de Markov, que permiten observar la evolución de un sistema en el tiempo y puede ser usado para determinar el umbral de nodos infectados necesarios para invadir una red. La

facilidad de empleo del enfoque, demostrada por sus aplicaciones en nuestros experimentos de simulación, lo convierte en un instrumento atractivo para realizar investigaciones sobre la propagación de malware e inclusive de otro tipo aún más complejo de ataque.

REFERENCIAS

Adamu, A., Gaidamaka, Y. & Samuylov, A. (2011), 'Discrete Markov Chain Model for Analyzing Probability Measures of P2P Streaming Network', *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)* **6869 LNCS**, 428–439.
URL: *https://link-springer-com.pbidi.unam.mx:2443/chapter/10.1007/978-3-642-22875-9_39*

Bocek, T., Shann, M., Hausheer, D. & Stiller, B. (2008), Game theoretical analysis of incentives for large-scale, fully decentralized collaboration networks, *in* '2008 IEEE International Symposium on Parallel and Distributed Processing', IEEE, pp. 1–8.

Chen, T., Zhang, X., Zheng, J. & Li, H. (2009), *Active Worm Propagation Modeling in Unstructured P2P Networks.*
URL: *https://www.researchgate.net/publication/266268815*

Christin, N., Weigend, A. S. & Chuang, J. (2005), Content availability, pollution and poisoning in file sharing peer-to-peer networks, *in* 'Proceedings of the 6th ACM conference on Electronic commerce', pp. 68–77.

Dinger, J. & Hartenstein, H. (2006), 'Defending the sybil attack in P2P networks: Taxonomy, challenges, and a proposal for self-registration', *Proc. - First Int. Conf. Availability, Reliab. Secur. ARES 2006* **2006**, 756–763.

Douceur, J. R. (2002), The sybil attack, *in* 'International workshop on peer-to-peer systems', Springer, pp. 251–260.

Engle, M. & Khan, J. I. (2006), 'Vulnerabilities of p2p systems and a critical look at their solutions', *Kent State University, Tech. Rep* .

Esquivel, E. B., Rivero-Angeles, M. E. & Rubino, G. (2013), 'Priority scheme for window-based video-on-demand transmission on BitTorrent-like Peer-to-Peer networks', *IEEE Int. Conf. Commun.* pp. 3000–3005.

Faheem Rasheed, M. (2012), Modelling Virus Propagation in P2P Networks, Technical report.
URL: *www.IJCSI.org*

Fan, X. & Xiang, Y. (2010), 'Modeling the propagation of Peer-to-Peer worms', *Futur. Gener. Comput. Syst.* **26**, 1433–1443.
URL: *www.elsevier.com/locate/fgcs*

Ferdous, S., Chowdhury, F. & Moniruzzaman, M. (2007), 'A Taxonomy of Attack Methods on Peer-to-Peer Network', *Proc. 1th Indian Conf. Comput. Intell. Inf. Secur.* **2007**(April 2016), 132–138.

Grishunina, Y., Manita, L. & Elizarov, S. A. M. (2017), 'Stochastic Models of Virus Propagation In Computer Networks : Algorithms of Protection and Optimization', **38**(5), 906–909.

Keyani, P., Larson, B. & Senthil, M. (2002), Peer pressure: Distributed recovery from attacks in peer-to-peer systems, *in* 'International Conference on Research in Networking', Springer, pp. 306–320.

Lee, M. S. & Jang, D. J. (2020), 'A survey of blockchain security issues', *JP J. Heat Mass Transf.* **2020**(Special Issue 1), 29–35.

Liang, J., Naoumov, N. & Ross, K. W. (2006), 'The index poisoning attack in P2P file sharing systems', *Proc. - IEEE INFOCOM* **00**(c).

Mishra, M. (2003), Cascade: an attack resistant peer-to-peer system, *in* 'the 3rd New York Metro Area Networking Workshop'.

Moreno Mogollón, J. A., Padilla Aguilar, J. J., Escobar Ordóñez, V. & Correo Montesino, A. F. (2007), 'Characterization and simulation of the lan traffic by mmpp model', *Revista Facultad de Ingeniería Universidad de Antioquia* (42), 7–29.

Nielson, S. J., Crosby, S. A. & Wallach, D. S. (2005), A taxonomy of rational attacks, *in* 'International Workshop on Peer-to-Peer Systems', Springer, pp. 36–46.

Rivero-Angeles, M. E. & Rubino, G. (2010), 'Priority-based scheme for file distribution in peer-to-peer networks', *IEEE Int. Conf. Commun.* pp. 1–6.

Robayo Santana, E. L. (2009), 'Detección de intrusos en redes de telecomunicaciones ip usando modelos ocultos de markov/intrusion detection in ip network telecommunications using hidden markov models', *Departamento de Ingeniería de Sistemas e Industrial* .

Singh, A., Castro, M., Druschel, P. & Rowstron, A. (2004), Defending against eclipse attacks on overlay networks, *in* 'Proceedings of the 11th workshop on ACM SIGOPS European workshop', pp. 21–es.

Suñé Torrents, A., Fonollosa Guardiet, J. B., Fernández Alarcón, V. & Sallán Leyes, J. M. (2017), *Cadenas de markov: Métodos cuantitativos para la toma de decisiones III*, Universitat Politècnica de Catalunya. Iniciativa Digital Politècnica.

Torres-Cruz, N., Rivero-Angeles, M. E., Rubino, G., Menchaca-Mendez, R. & Menchaca-Mendez, R. (2018), 'An efficient resource allocation scheme for VoD services over window-based P2P networks', *Multimed. Tools Appl.* **77**(23), 31427–31445.

Tosun, U. (2005), Hidden markov models to analyze user behaviour in network traffic, Technical report, Citeseer.

Wagner, A. & Plattner, B. (2002), Peer-to-peer systems as attack platform for distributed denial-of-service, *in* 'ACM SACT Workshop, Washington, DC, USA', Citeseer.

Wararkar, P., Kapil, N., Rehani, V., Mehra, Y. & Bhatnagar, Y. (2016), Resolving Problems Based on Peer to Peer Network Security Issue's, *in* 'Phys. Procedia', Vol. 78, Elsevier B.V., pp. 652–659.

Washbourne, L. (2015), 'A survey of P2P network security', pp. 1–12.
URL: *http://arxiv.org/abs/1504.01358*