

# ESTUDIO Y SIMULACIÓN DE ALGORITMOS PARA LA EVACUACIÓN DE PERSONAS EN SITUACIONES DE EMERGENCIA SOBRE UNA ESTRUCTURA SIMILAR AL RECTORADO DE LA ESPOL.

Manuel Santos-Navarrete, Ignacio Marin-Garcia  
Facultad de Ingeniería en Electricidad y Computación  
Escuela Superior Politécnica del Litoral (ESPOL)  
Campus Gustavo Galindo, Km 30.5 vía Perimetral  
Apartado 09-01-5863. Guayaquil-Ecuador  
mesantos@espol.edu.ec, imaringa@fiec.espol.edu.ec

## Resumen

*Este trabajo de investigación forma parte del proyecto KISHWAR y corresponde al estudio y simulación de una red que representará al edificio del Rectorado de la ESPOL sobre la cual se implementó un algoritmo que será encargado de recalculando rutas hacia las salidas más seguras durante una situación de emergencia y con esto, poder determinar cuál tiene un mejor rendimiento. Los algoritmos que fueron puestos a prueba en este estudio fueron: Dijkstra y Bellman-Ford. Con las simulaciones que realizamos, se pudieron obtener diversos datos que fueron determinantes a la hora de elegir un algoritmo por encima del otro. El algoritmo Dijkstra según nuestro análisis es ligeramente superior a Bellman-Ford en rendimiento. Consideramos que nuestro estudio fue un éxito ya que cumplimos con los objetivos planteados, y pudimos concluir cuál algoritmo tiene un mejor desempeño para el cálculo de rutas de evacuación y esperamos que este análisis sea tomado en cuenta durante la implementación del proyecto KISHWAR de la ESPOL.*

**Palabras Claves:** Algoritmo, Bellman-Ford, Dijkstra, Evacuación, Kishwar.

## Abstract

*This research is part of the Kishwar project and corresponds to the study and simulation of a network which represents the building of the Rectory of the ESPOL on which an algorithm that will be responsible for recalculating routes to safer exits during an emergency was implemented and with this, we were able to determine which performs better. We tested the following algorithms: Dijkstra and Bellman-Ford. With the simulations performed, we were able to gather data which were determining factors when choosing an algorithm over another. The performance of Dijkstra's algorithm according to our analysis is slightly superior than Bellman-Ford. We believe that our study was successful and we met the objectives, and we were able to conclude which algorithm has a better performance for the calculation of evacuation routes and we hope this analysis is taken into account during the implementation of the Kishwar project.*

**Keywords:** Algorithm, Bellman-Ford, Dijkstra, Evacuation, Kishwar.

## 1. Introducción

Hoy en día los algoritmos son piezas fundamentales en la implementación diversos sistemas tales como cálculo de rutas en los protocolos de encaminamiento, video juegos, o elección del mejor camino en Google Maps, pero en este estudio se van a poner a prueba ciertos algoritmos bajo condiciones un tanto inusuales para poder determinar cuál de ellos tiene el mejor rendimiento.

Este estudio tiene como punto de partida ser una base al momento de elegir un algoritmo de encaminamiento para el proyecto KISHWAR de la ESPOL. Este proyecto consiste en la implementación de un sistema de evacuación dinámica debido a que las actuales señaléticas estáticas dentro de los edificios que indican hacia donde se encuentra la salida más cercana no llegan a satisfacer las expectativas en el aspecto de cantidad de vidas salvadas. Por el contrario, este proyecto contempla la

implementación de una red de sensores y de señaléticas a base de LEDs para redirigir a las personas a las salidas de emergencia de una forma más eficiente. Como es de imaginar, pese a que solamente está contemplado que se implemente en el edificio del rectorado de la ESPOL, este sistema está basado en muchas tecnologías que deberán trabajar conjuntamente para poder desempeñarse de la forma esperada. Debido a esto, se subdividió el proyecto, en módulos más pequeños, uno de ellos es el encargado de la toma de decisiones en el mismo.

Esta toma de decisiones está basada en las diversas variables que la red de sensores va a entregar, pero que en definitiva se van a traducir a dos estados: disponible o no disponible. Teniendo en cuenta estos dos estados, se deberá proceder a calcular las rutas más óptimas hacia las salidas del edificio. En este punto es que entran en escena los algoritmos de enrutamiento, que serán los encargados de dicha toma de decisiones, pero siendo esta una topología muy

particular, ya que representarán los elementos que se encuentran en un edificio tales como: pasillos, puertas, escaleras, etc. es necesario evaluar cuál algoritmo tiene un mejor desempeño. Estos algoritmos serán puestos a prueba en fallas en la red que emulen o tengan un comportamiento muy similar a alguna situación de emergencia.

Teniendo en cuenta nuestros objetivos, limitantes, entre otras cuestiones, se optó por realizar todas las pruebas correspondientes en un entorno simulado, utilizando un simulador de redes muy general, ya que no requerimos comportamientos específicos de un enrutador, conmutador u otro equipo de red.

## 2. Diseño e implementación

El simulador que se escogió para realizar este estudio, se llama Network Simulator versión 2 (NS2), es un simulador de redes a nivel general de código abierto y libre distribución. Está orientado a la investigación de varios protocolos en todas las capas del modelo OSI, tanto para redes cableadas, inalámbricas y móviles. El NS2 es un simulador basado en eventos, que comprende de dos lenguajes de programación: C++ (lenguaje de programación Orientado a Objetos) que es el núcleo del mismo ya que define su comportamiento interno y OTcl que es el encargado de establecer y configurar los parámetros de la topología actual y los eventos que ocurrirán en ella.

El edificio que iba a ser representado como una topología lógica fue el rectorado de la Escuela Superior Politécnica del Litoral (ESPOL), consta de cuatro pisos: el sótano, la planta baja, el primer piso y el segundo piso. Los nodos, en nuestra topología, no contienen una cola de procesamiento, por el contrario, las colas se encuentran ubicadas en los enlaces, de lo que se encarga el nodo es de procesar un paquete a la vez y calcular las rutas a todos los nodos de la red. Los enlaces, son utilizados para conectar nodos, ya que solamente se han considerado el uso de enlaces punto a punto, y entre sus funciones se encuentran la de agendar eventos, basándose atributos que posee estos objetos. Los enlaces tienen la función de agendar dos eventos: la de la partida de un paquete del nodo transmisor y la del arribo del paquete al nodo del siguiente salto; el primer evento se refiere al tiempo de transmisión de un paquete desde el punto de vista del nodo que lo envía (ver ecuación 1); el segundo evento que se debe agendar, corresponde al tiempo en el que el paquete arriba al nodo del siguiente (ver ecuación 2).

$$t_{Tx} = \frac{PktSize}{BW} \quad (1)$$

$$t_{Rx} = t_{Tx} + \text{retardo} \quad (2)$$

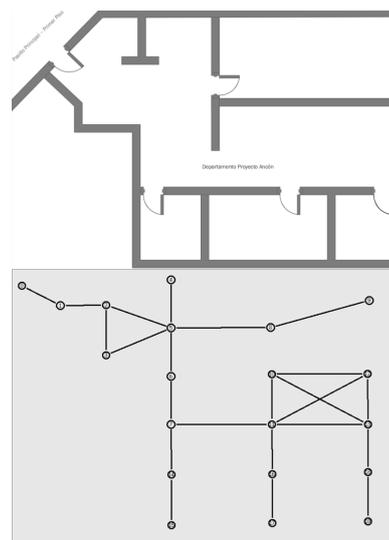


Figura 1. Representación lógica de una oficina

En la Figura 1, podemos ver la representación lógica de la Unidad del Proyecto Ancón ubicada en el primer piso del Rectorado de la ESPOL. Como se puede observar, la topología guarda la forma de la disposición física de dicho departamento. A simple vista, probablemente no se notaría una diferencia significativa entre un pasillo y una puerta, pero la diferencia radica en la configuración que tienen los enlaces que representan a una puerta y un pasillo. Entre los parámetros establecidos se encuentran la capacidad de las colas de procesamiento, que en las puertas su capacidad es menor a la de un pasillo; el ancho de banda, que en las puertas es mayor ya que está relacionado con los tiempos de transmisión de los paquetes y que en las puertas es menor debido a que se requiere menos tiempo atravesar la misma, en comparación con un pasillo, etc.

Los paquetes representaron a las personas dentro del edificio por lo que si dentro de nuestras simulaciones un paquete queda atrapado, representará un vida perdida en la realidad. El tipo de tráfico utilizado fue el de Tasa de Bits Constante (CBR por sus siglas en inglés) ya que corresponde a un tipo de paquete estándar, en el que se puede configurar el tamaño del paquete y la cantidad máxima que transmitirá, por lo que es ideal para el entorno que queríamos recrear. Se decidió crear dos tipos de paquetes, cada uno con su propio tamaño, los cuales representan a personas sin dificultad para trasladarse (tamaño menor) y a personas con alguna dificultad en su traslado (tamaño mayor), ya que de esta forma existen paquetes que atraviesan más rápido los enlaces que otros, tal como sucede en la realidad. Para poder transmitir los paquetes, es necesario contar con un protocolo de capa de Transporte (según el modelo OSI) y puesto que no requerimos acuses de recibos y una extrema fiabilidad de este tipo de protocolos, se optó por utilizar UDP ya que es un protocolo que no es

orientado a la conexión y es más ligero en términos del tamaño de su encabezado.

### 3. Pruebas realizadas

El primer tipo de pruebas que se realizaron fueron las que correspondían a seleccionar aleatoriamente los nodos que quedarían inhabilitados en toda la topología. Con esto nos referimos a que de forma totalmente aleatoria se eligieron cuáles eran los nodos que iban a fallar, ya que contamos con 936 nodos que corresponden a la topología, más diez nodos entre los cuales se encuentran los nueve relacionados a todas las puertas de salidas que posee el edificio y un nodo final que es al cual todos los paquetes deberían llegar para ser considerados que se encuentran a salvo. Tratando de realizar una analogía con la estructura real, un temblor o un terremoto tiene un comportamiento similar ya que varias zonas de la estructura pueden quedar intransitables durante dicho evento.

Además se realizaron otros tipos de pruebas que representan a otros tipos de catástrofes, como por ejemplo el caso de un incendio o alguna posible fuga de gas. Debido a esto, se procedió a implementar dichos eventos; en el caso de los incendios y fugas de gas se procedió a escoger un nodo que represente a una sección del edificio en la que sea probable que en algún momento este sea el foco de un evento similar (focos diferentes por evento) y cada cierto tiempo que se vayan deshabilitando los nodos adyacentes al nodo foco y pasado el mismo intervalo de tiempo, ocurra lo mismo con los nodos vecinos a estos, continuando con la misma idea por seis o siete intervalos de tiempo. La diferencia entre ambos eventos, a más de los diferentes focos, radica en el intervalo de tiempo de propagación, ya que en una fuga de gas es ligeramente más rápida que en un incendio.

Los datos más importantes que se pueden determinar debido a las pruebas realizadas, es en torno a los paquetes/personas que son salvados bajo las instrucciones un algoritmo dado. Entre los datos que son arrojados gracias a los paquetes salvados, es la cantidad de paquetes totales salvados en promedio, así como también la cantidad de paquetes salvados pequeños y más grandes que, como se explicó previamente, representan a las personas sin discapacidad y con capacidades disminuidas, respectivamente. Además se puede conocer desde donde iniciaron su recorrido los paquetes que se salvaron, y también cual puerta de salida cruzaron para poder desalojar el edificio. En el caso de la cantidad promedio de personas atrapadas en el edificio, se puede conocer en que lugares del rectorado quedaron atrapadas.

Es necesario considerar el tiempo de acción de los algoritmos ya que el tiempo en el que un edificio debe ser evacuado durante una situación de emergencia es

finito y no puede prolongarse indefinidamente puesto que conforme pasen los minutos la eficacia de la evacuación va mermando. Es por esto que se ha establecido un tiempo quince minutos como tiempo máximo en que la estructura deberá ser evacuada en su totalidad y pasado este tiempo los paquetes que aún se encuentren en la estructura, podrán ser considerados como perdidos. El tiempo de evacuación es un parámetro a tener muy en cuenta dado que es fundamental en situaciones de emergencia poder desalojar a las personas lo más rápido posible y para esto, el algoritmo de enrutamiento para encontrar rutas de evacuación debe reaccionar de forma veloz y recuperarse de inmediato ante una red que va cambiando conforme pasa el tiempo.

### 4. Resultados

Partiremos nuestro análisis con los resultados obtenidos de las simulaciones utilizando el algoritmo de Dijkstra. Como se puede observar en la Tabla 1, el porcentaje promedio de personas que logran ser evacuadas es del 75.56% con una confiabilidad del 95%. De forma similar, con el algoritmo Bellman-Ford se tiene que los resultados son similares a los obtenidos con Dijkstra, como se observa en la Tabla 1, con respecto al porcentaje de personas que logran ser evacuadas, ya que se puede hablar de un porcentaje promedio del 75.92% con una confiabilidad del 95%, por consiguiente es muy pequeño el porcentaje de personas que no lograron salir del edificio. Su rendimiento en cuanto a números al parecer es ligeramente superior a los de Dijkstra, sin embargo hay que tener en cuenta que durante las simulaciones, le tomo mucho mas tiempo completarlas, lo cual a la hora de implementarlo, podría traducirse en una sobrecarga de recursos del equipo en donde se vaya a implementar.

Tabla 1. Resultados obtenidos

| EVENTO               | ALGORITMO    | T PROM [min] | %SALVADOS PROM |
|----------------------|--------------|--------------|----------------|
| Terremoto            | Dijkstra     | 12,21        | 74,60%         |
| Incendio VRec        | Dijkstra     | 12,28        | 76,40%         |
| Incendio Suministros | Dijkstra     | 12,25        | 75,80%         |
| Fuga Gas Cocina      | Dijkstra     | 12,18        | 75,00%         |
| Fuga Gas Rectorado   | Dijkstra     | 12,20        | 76,00%         |
| Terremoto            | Bellman-Ford | 12,28        | 79,33%         |
| Incendio VR          | Bellman-Ford | 12,29        | 75,00%         |
| Incendio Suministros | Bellman-Ford | 12,15        | 75,00%         |
| Fuga Gas Cocina      | Bellman-Ford | 12,15        | 74,00%         |
| Fuga Gas Rectorado   | Bellman-Ford | 12,36        | 75,00%         |

El tiempo que se tomó cada algoritmo para poder evacuar el edificio de la mejor forma posible es muy importante al momento de decidimos a la hora de implementar un algoritmo en particular. Pero al final de las simulaciones realizadas, resulta interesante darnos cuenta que con ningún algoritmo se excedió del tiempo que inicialmente habíamos establecido como máximo para evacuar la estructura, que era de quince minutos. En el caso de Dijkstra, le tomó en promedio 12.22 minutos (12'13'') evacuar el edificio;

por otra parte, con Bellman-Ford la evacuación tomó en promedio 12.26 minutos (12'16").

## 5. Conclusiones

Como conclusiones a este trabajo de investigación se tiene lo siguiente:

1. Estadísticamente el algoritmo Dijkstra tiene un mejor porcentaje promedio de personas evacuadas en todos los escenarios, a excepción del que representaba a un terremoto, en comparación con Bellman-Ford.

2. Aunque ambos algoritmos hacen posible la evacuación del edificio en un tiempo inferior a los 15 minutos que se estipuló como tiempo límite, el algoritmo Dijkstra lo hizo en un tiempo promedio menor en comparación al tiempo promedio requerido por Bellman-Ford para desalojar la estructura, sin embargo, son estadísticamente similares.

3. Dado que la desviación estándar de los datos es inferior al 5% de la media en todos los casos, se puede inferir que la confiabilidad de las pruebas con Bellman-Ford y Dijkstra es estadísticamente igual.

4. Pese a que depende de la robustez del equipo en donde se implementará, igualmente hay que considerar que Bellman-Ford requirió mucho más tiempo para completar las simulaciones, por lo que podemos deducir que tiene una mayor carga de procesamiento, en comparación a Dijkstra.

## 6. Agradecimientos

Agradezco a todas las personas que aportaron activamente con esta investigación tal es el caso del personal de la Unidad de Planificación de Infraestructura Física que proporcionó los planos del Rectorado de la ESPOL, además de todo el personal que se encontraba dentro del edificio durante la recolección de datos, ya que sin la predisposición que mostraron para aportar a esta investigación, hubiese sido muy difícil llevarla a cabo.

## 7. Referencias

- [1] Pauls, J. L., "The movement of people in buildings and design solutions for means of egress", Fire Technology, Vol.20, No.1, 1984
- [2] Casadesús, S.; Garriga, F., "Optimal time of building evacuation considering evacuation routes", EJOR, Vol.192, Issue 2, 2009
- [3] Mejía, R; Bernal, J.; López, D., "Análisis de tipos de Gestores de Colas en NS-2.", Revista TIA, Octubre 15 de 2013
- [4] Gento, M.; López B.; Posada M., "Simulación del desalojo de edificios. Situación en España: E.T.S. Ingenieros Industriales (Universidad de Valladolid)", V Congreso de Ingeniería de Organización, 2003.

- [5] Lovas, G., "On performance measures for evacuation systems", European Journal of Operational Research, Vol.85, Issue 2,199
- [6] Lu, Q.; Huannng, Y.; Shekha S., "Evacuation Planning: A Capacity Constrained Routing Approach", [http://www.cc.ntut.edu.tw/~cmliu/SD/SD\\_F06g/papers/SN2\\_lu\\_george\\_shekhar.pdf](http://www.cc.ntut.edu.tw/~cmliu/SD/SD_F06g/papers/SN2_lu_george_shekhar.pdf), 2005
- [7] Bellman, Richard, "On a routing problem", Quarterly of Applied Mathematics, 1958
- [8] Dijkstra, E. W., "A note on two problems in connexion with graphs", Numerische Mathematik 1, 1959
- [9] Rodríguez, G., "Proyecto Getaway: Nuevos sistemas de evacuación en emergencias", <http://prevenblog.com/proyecto-getaway-nuevos-sistemas-de-evacuacion-en-emergencias/>, Marzo 24 de 2014
- [10] Avendaño, A., "Instalando y ejecutando NS2", <http://aveoctavo.blogspot.com/2013/03/laboratorio-de-tele-ns2.html>, Marzo 4 de 2013