

Josep Sampé Domènech

CONGESTIÓN Y ENRUTAMIENTO EN REDES CUÁNTICAS

TREBALL FI DE MÀSTER

Dirigit pel Dr. Jordi Duch Gavladà

Màster en Seguretat Informàtica i Sistemes Intel·ligents



UNIVERSITAT ROVIRA I VIRGILI

Tarragona

2014

Resumen

Las redes de comunicación cuántica a baja escala son una realidad hoy en día, tanto a nivel de laboratorio como a nivel comercial, y cada vez se están haciendo más avances para que en un futuro no muy lejano podamos disfrutar de una Internet cuántica.

La principal aplicación, actualmente, de este tipo de redes es aprovechar las características que nos ofrece la mecánica cuántica para realizar una importante primitiva criptográfica: el intercambio de claves secretas. Así como la criptografía clásica basa su robustez en operaciones matemáticas no demostradas, la criptografía cuántica basa su robustez en leyes físicas inquebrantables, por lo que se presume mucho más segura.

En este trabajo pretendemos analizar cuál sería la dinámica en este tipo de redes a gran escala, siempre según su estructura subyacente. Para ello hemos definido varios modelos que capturan las propiedades básicas de una red de comunicación cuántica.

Además hemos definido un conjunto de redes creadas con diferentes modelos para poder estudiar la existencia de la transición de un estado de flujo libre a un estado de congestión en función de los parámetros de la red.

Finalmente hemos implementado varios algoritmos de enrutamiento para comprobar como estos afectan a la congestión del sistema y poder comparar su eficiencia en función de la red subyacente, tratando, siempre, de aumentar la zona de flujo libre.

Agraïments

En primer lloc m'agradaria agrair al Dr. Jordi Duch la orientació y suport que rebut durant els darreres sis mesos en la elaboració d'aquesta tesis, sense el qual això no agües estat possible. En segon lloc també m'agradaria agrair al Jesús, David i en especial a l'Alessio Cardillo el facilitar-me l'accés i configurar-me tot el necessari per a realitzar totes les simulacions que he fet en el transcurs d'aquesta tesis a un dels clústers ubicat a la Universitat de Zaragoza.

Per acabar, agraeixo a tota la meva família tot el suport que m'han donat durant tota la vida y que m'han ajudat ha arribar fins aquí.

Contenidos

Lista de tablas.....	VIII
Capítulo 1. Introducción.....	1
1.1 Objetivos	3
1.2 Estructura de la tesis	4
Capítulo 2. Background	6
2.1 Redes.....	6
2.1.1 Definición.....	7
2.1.2 Tipos de redes.....	8
2.1.3 Redes complejas.....	9
2.2 Routing.....	15
2.2.1 Selección del camino.....	15
2.2.2 Métodos de enrutamiento.....	16
2.2.3 Tipos de entrega de paquetes	18
2.2.4 Congestión	19
2.3 Mecánica cuántica.....	20
2.3.1 Qubit	20
2.3.2 Representación física	21
2.3.3 Entrelazamiento.....	22
2.3.4 Teleportación.....	23
2.3.5 Intercambio de entrelazamiento	25
Capítulo 3. Estado del arte.....	27
3.1 Redes y tráfico	27
3.2 Distribución de claves cuántica.....	29
3.3 Redes cuánticas.....	31

Capítulo 4. Propuesta.....	35
4.1 Redes.....	35
4.1.1 Redes Erdős–Rényi	36
4.1.2 Redes Barabási–Albert.....	37
4.1.3 Redes Watts-Strogatz	38
4.2 Modelo.....	40
4.2.1 Modelo 1	41
4.2.2 Modelo 1.1.....	42
4.2.3 Modelo 1.2.....	42
4.2.4 Análisis	43
4.3 Configuración y Simulación.....	44
4.3.1 Paralelización	46
4.4 Algoritmos de enrutamiento.....	47
4.4.1 Algoritmos deterministas o estáticos	47
4.4.2 Algoritmos adaptativos o dinámicos.....	49
Capítulo 5. Resultados.....	53
5.1 Congestión en el Modelo 1	53
5.1.1 Redes Erdős–Rényi	54
5.1.2 Redes Barabási–Albert.....	58
5.1.3 Redes Watts-Strogatz	62
5.2 Congestión en el Modelo 1.1	70
5.2.1 Redes Erdős–Rényi	70
5.2.2 Redes Barabási–Albert.....	76
5.2.3 Redes Watts-Strogatz	81
5.3 Congestión en el Modelo 1.2.....	93
5.3.1 Redes Erdős–Rényi	93
5.3.2 Redes Barabási–Albert.....	96
5.3.3 Redes Watts-Strogatz	98
Capítulo 6. Conclusiones	103
Bibliografía.....	107

Lista de tablas

Tabla 4.1. Ejemplo de tabla de enrutamiento en el algoritmo A3.	51
Tabla 4.2. Ejemplo de tabla de enrutamiento en el algoritmo A3.	52
Tabla 5.1 Grafico (derecha) y tabla de puntos críticos (izquierda) de las redes Erdős-Rényi sobre el algoritmo A1 con diferentes valores de m	54
Tabla 5.2. Grafico (derecha) y tabla de puntos críticos (izquierda) de las redes Erdős-Rényi con diferentes valores de m sobre el algoritmo A2.	55
Tabla 5.3. Grafico (derecha) y tabla de puntos críticos (izquierda) de las redes Erdős-Rényi sobre el algoritmo A3 con diferentes valores de m	56
Tabla 5.4 Grafico (derecha) y tabla de puntos críticos (izquierda) de las redes Barabási-Albert sobre el algoritmo A1 con diferentes valores de m	58
Tabla 5.5 Grafico (derecha) y tabla de puntos críticos (izquierda) de las redes Barabási-Albert sobre el algoritmo A3 con diferentes valores de m	59
Tabla 5.6 Grafico (derecha) y tabla de puntos críticos (izquierda) de las redes Barabási-Albert sobre el algoritmo A3 con diferentes valores de m	60
Tabla 5.7 Evolución del punto crítico en las redes Watts-Strogatz sobre el algoritmo A1 con diferentes valores de r en función del número de vecinos k	63
Tabla 5.8 Evolución del punto crítico en las redes Watts-Strogatz sobre el algoritmo A2 con diferentes valores de r en función del número de vecinos k	65
Tabla 5.9 Evolución del punto crítico en las redes Watts-Strogatz sobre el algoritmo A3 con diferentes valores de r en función del número de vecinos k	67

Tabla 5.10. Tabla comparativa y gráfico de puntos críticos en las redes Erdős–Rényi con diferentes valores de z sobre el Algoritmo A1, en función de la probabilidad de regenerar un link v .	71
Tabla 5.11. Tabla comparativa y gráfico de puntos críticos en las redes Erdős–Rényi con diferentes valores de z sobre el Algoritmo A2, en función de la probabilidad de regenerar un link v .	72
Tabla 5.12. Tabla comparativa y gráfico de puntos críticos en las redes Erdős–Rényi con diferentes valores de z sobre el algoritmo A3, en función de la probabilidad de regenerar un link v .	73
Tabla 5.13. Tabla comparativa de puntos críticos en las redes Barabási–Albert sobre el Algoritmo A1 con diferentes valores de m en función de la probabilidad de regenerar un link v .	77
Tabla 5.14. Tabla comparativa de puntos críticos en las redes Barabási–Albert sobre el algoritmo A2 con diferentes valores de m en función de la probabilidad de regenerar un link v .	78
Tabla 5.15. Tabla comparativa de puntos críticos en las redes Barabási–Albert sobre el algoritmo A3 en función de la probabilidad de regenerar un link v .	79
Tabla 5.16. Tabla comparativa de puntos críticos en las redes Watts-Strogatz sobre el Algoritmo A1 con $k = 4$ y $k = 8$ vecinos de media en función de los distintos valores de regeneración de enlace v .	83
Tabla 5.17. Gráficos de puntos críticos en las redes Watts-Strogatz sobre el Algoritmo A1 con $k = 4$ y $k = 8$ vecinos de media en función de los distintos valores de regeneración de link v .	84
Tabla 5.18. Tabla comparativa de puntos críticos en las redes Watts-Strogatz sobre el Algoritmo A2 con $k = 4$ y $k = 8$ vecinos de media en función de los distintos valores de regeneración de link v .	86
Tabla 5.19. Tabla comparativa de puntos críticos en las redes Watts-Strogatz sobre el algoritmo A3 con $k = 4$ y $k = 8$ vecinos de media en función de los distintos valores de regeneración de enlace v .	89

Capítulo 1

Introducción

Actualmente la capacidad de transmitir información y procesarla está limitada por las leyes físicas de los sistemas que se utilizan para tales procesos. Las propiedades de la mecánica cuántica nos permiten utilizar sistemas cuánticos, como por ejemplo fotones o electrones, para codificar información y, posteriormente, procesarla de una forma única y nunca antes vista. Entre estas propiedades podemos encontrar, por ejemplo, la superposición de estados cuánticos o la posibilidad de realizar un entrelazamiento cuántico entre un par de partículas.

Las redes de comunicación cuántica, o simplemente redes cuánticas, son redes de ordenadores que están conectados a través del uso de canales de comunicación cuánticos. Algunos ejemplos de estas redes, como el SECOQC [1] en Viena, han demostrado la viabilidad de este tipo de redes para establecer canales de comunicación seguros para la transferencia de información.

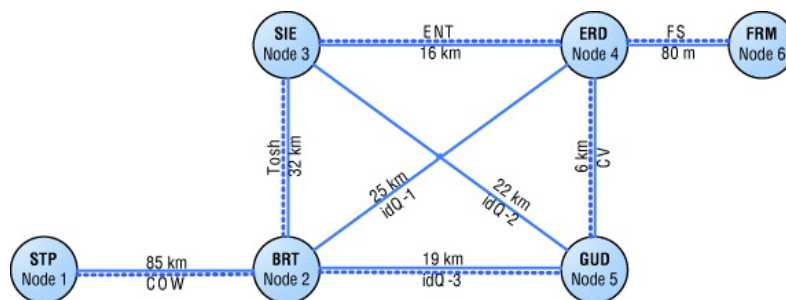


Figura 1.1. Esquema del proyecto SECOQC en Viena.

La principal aplicación de este tipo de redes ha sido la distribución de claves seguras para poder encriptar, posteriormente, las comunicaciones por un canal clásico. En la criptografía de clave pública tradicional la seguridad de las claves descansa en la complejidad computacional no demostrada de ciertas funciones matemáticas mientras que en la criptografía cuántica la seguridad descansa en los principios de la mecánica cuántica [5] que dictan que los estados no pueden ser identificados, copiados o divididos sin introducir perturbaciones, lo que hace que sea imposible que una tercera persona pueda hacer un ataque de “eavesdropping” y robar la clave, ya que sería detectado por los dos integrantes de la comunicación.

La comunicación cuántica se centra actualmente en la transmisión de información entre pares de equipos que tienen una conexión física directa, normalmente utilizando fotones de luz. Dicha información se envía a través de una fibra óptica, y pueden enviarse tanto qubits independientes como qubits entrelazados cuánticamente [7]. Actualmente el límite para estas transmisiones es de entre 100Km y 250Km [8, 48], dado que la probabilidad de transmisión decae exponencialmente con la distancia. Debido a esto, hace unos años se definió la idea del intercambio de entrelazamiento [2, 3, 4], donde un nodo de la red puede correlacionar dos de sus qubits que no tienen nada en común, creando así conexiones más largas como muestra el siguiente ejemplo:

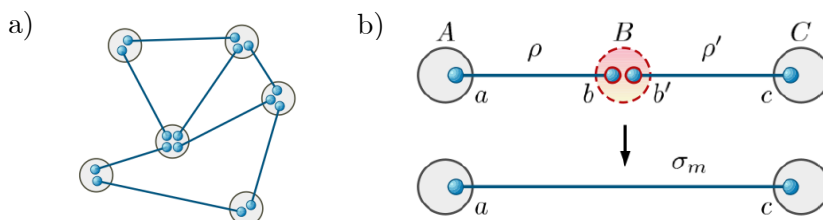


Figura 1.2. a) Red cuántica: Inicialmente los nodos comparten pares entrelazados de qubits (links). b) Intercambio de entrelazamiento: El nodo B realiza el intercambio de entrelazamiento creando una conexión directa entre los nodos A y C.

Esta tecnología abre la puerta a la creación de grandes redes de ordenadores cuánticos donde, en contraste de la comunicación clásica, la transferencia de información punto a punto es segura. Debido a la gran cantidad de nodos que podrían contener en estas redes, se hace necesario diseñar protocolos de enrutamiento destinados a definir cómo y dónde se realiza el intercambio de entrelazamiento entre los distintos nodos de la red para poder crear una comunicación de larga distancia.

1.1 Objetivos

En esta tesis vamos a considerar un nuevo enfoque mediante el modelado de las redes cuánticas como redes dinámicas, donde la información se almacena en los nodos y, usando las reglas de comunicación cuántica, en cada momento el sistema establecerá vínculos lógicos entre origen y destino para realizar la transferencia de información. Contamos con la posibilidad de intercambiar información a través de la teleportación cuántica y del entrelazamiento entre los nodos de la red para proponer un protocolo de comunicación cuántica que optimice la cantidad de información transferida para explorar las propiedades de congestión de estas redes.

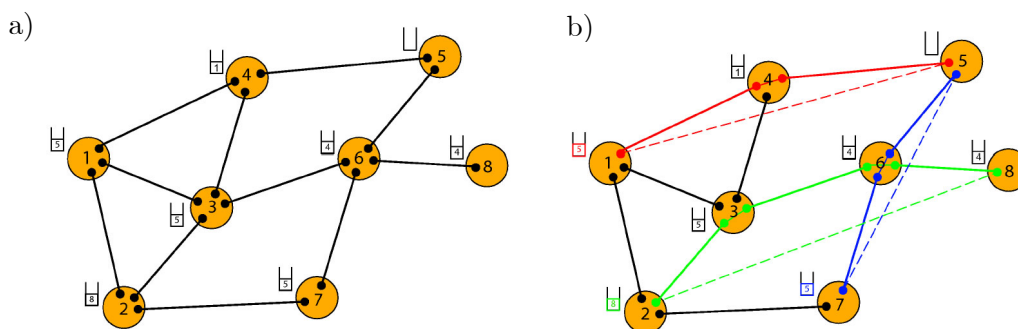


Figura 1.3. Ejemplo de un mecanismo de enrutamiento en una red cuántica. **a)** cada nodo de la red tiene una cola con paquetes que necesita ser enviado a un destino. **b)** Utilizando un algoritmo de routing, establecemos entrelazamientos cuánticos entre los diferentes nodos para conectar el número máximo de fuentes hacia destinos.

Para lograr esto, se llevarán a cabo varios pasos y objetivos parciales:

- Crear un modelo que capture las propiedades básicas de una red de comunicación cuántica.
- Generar un sistema dinámico para estudiar la distribución de la información en la parte superior de esta red, teniendo en cuenta la limitación de la capacidad de los canales de comunicación.
- Estudiar la existencia de las transiciones de congestión en función de los parámetros del sistema.
- Comparación de la eficacia de diferentes estrategias de enrutamiento en función de la topología de la red subyacente.

1.2 Estructura de la tesis

En el [capítulo 2](#) se hace un acercamiento a los antecedentes necesarios para comprender el núcleo de la tesis. Se explican los fundamentos de redes, los tipos de redes que podemos encontrar en el mundo real y los tipos de redes complejas que se han utilizado en el desarrollo del proyecto. También se explica algunos conceptos básicos sobre routing; en que consiste, que tipos hay, como se mesura la congestión, etc. Finalmente se hace una pequeña introducción a la Teoría Cuántica de la Información, explicando los temas claves utilizados en esta tesis, como son la teleportación, el entrelazamiento y el intercambio de entrelazamiento.

En el [capítulo 3](#) se revisa el estado del arte de los principales temas que se tratan en esta tesis; las redes y el tráfico, la distribución de claves cuántica y las redes cuánticas. Para ello se revisan algunos ‘papers’ importantes, tanto actuales como del pasado, en cada uno de los campos.

En el [capítulo 4](#) se explica nuestra propuesta, inicialmente se proponen las redes que utilizaremos en el transcurso del proyecto, creadas con distintos modelos de red. También se define un modelo que especifica las propiedades básicas de una red de comunicación cuántica. Finalmente se proponen algunos algoritmos de enrutamiento para estudiar cómo estos afectan a la congestión del sistema.

En el [capítulo 5](#) se muestran los resultados obtenidos de los diferentes experimentos realizados con cada uno de los algoritmos, sobre cada una de las redes propuestas en el capítulo 4.

El [capítulo 6](#) es para las conclusiones, en el veremos el conocimiento obtenido de esta tesis, así como las posibles ideas para un trabajo futuro.

Capítulo 2

Background

Vamos a iniciar la tesis dedicando un capítulo a conceptos básicos que se trataran durante el desarrollo de esta. Dado que no estudiamos un campo en concreto, esta tesis puede desglosarse en tres ramas de las cuales es necesario hacer un background para poder comprender los aspectos de estas. El objetivo no es hacer un análisis minucioso de cada una de ellas sino un acercamiento conceptual para comprender mejor todo lo realizado en la tesis. Por una parte tenemos las **redes**, por lo que es necesario comprender que son y qué tipos hay junto con algunas de sus características. Otra materia clave es el **enrutamiento**, igual que antes es necesario saber en qué consiste y como se clasifican sus protocolos. Finalmente tenemos la **mecánica cuántica** que estará detrás del funcionamiento de la red, ya que será la base del comportamiento del modelo y de los algoritmos de enrutamiento que estudiaremos.

2.1 Redes

En esta sección vamos a ver una pequeña definición de lo que son las redes junto con todas aquellas propiedades relevantes para el proyecto. También vamos a ver qué tipos de redes hay en el mundo según su clasificación analizando el tipo de nodo y los enlaces que las forman. Finalmente, se describirán los distintos modelos de redes que se utilizaran en esta tesis junto con sus principales características.

2.1.1 Definición

Una red, llamada muchas veces grafo [9] en la mayoría de literatura, está compuesta por **vértices** o **nodos** con conexiones entre ellos llamadas **aristas** o **arcos** que permiten representar relaciones entre los elementos de un conjunto.

Matemáticamente, una red o grafo G es un par ordenado $G = (V, E)$ donde V es un conjunto de vértices $\{v_0, v_1, v_2, \dots, v_n\}$ y E es un conjunto de aristas entre los vértices $E \subseteq V * V$, $V * V = \{(i, j) \mid i, j \in V\}$. Otros conceptos matemáticos relacionados con los grafos que consideraremos en esta tesis son los siguientes:

- Cada arista (v_i, v_j) se llama conexión dirigida del nodo v_i al nodo v_j . El grafo G se llama **no dirigido** si para cada pareja $(v_i, v_j) \in E$ también existe la pareja $(v_j, v_i) \in E$. De lo contrario, la red se denomina **dirigida**.
- Todos los nodos que estén conectados directamente a un nodo v_i , son los **vecinos** de v_i .
- El número k_i de vecinos del nodo v_i , es decir, el número de conexiones de v_i , se llama **grado de conectividad** de v_i .
- Un **camino** («path») es una secuencia de vértices, $\{v_0, v_1, v_2, \dots, v_n\}$ donde cada par de la secuencia es enlazado ($\forall i, i + 1, (v_i, v_{i+1}) \in E$). La longitud del camino es el número de nodos que están en la secuencia.
- En un grafo las aristas pueden tener un valor asociado E_w . Cuando este valor es 0 o 1, solo significa que la arista existe o no y por lo tanto lo llamaremos **grafo sin ponderar**. De lo contrario cuando $E_w \in \mathbb{R}$ se llamara **grafo ponderado**.
- Un grafo está **etiquetado** si sus vertices o aristas contienen una marca que los hace unívocamente distinguibles del resto.
- La **intermediación** («betweenness centrality») [10] es una medida que cuantifica la frecuencia o el número de veces que un nodo o arista actúa como un puente a lo largo del camino más corto entre otros dos nodos. Esta medida tiene una variante llamada

Intermediación de Newman [11] basada en considerar caminos aleatorios del grafo, y no exclusivamente los más cortos.

- La **centralidad de vector propio** («Eigenvector centrality») [12] mide la influencia de un nodo en una red. Los nodos que poseen un valor alto de esta medida de centralidad están conectados a muchos nodos que a su vez están bien conectados, también en este sentido; por lo tanto, son buenos candidatos para difundir información.

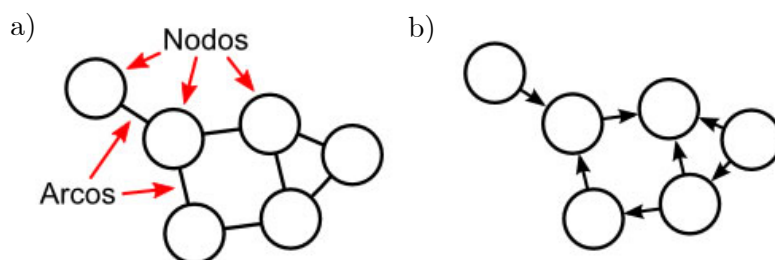


Figura 2.1. a) ejemplo de un grafo no dirigido y sin ponderar con nodos (puntos) y arcos (líneas). b) ejemplo de un grafo dirigido y sin ponderar con nodos (puntos) y arcos (flechas).

2.1.2 Tipos de redes

Podemos clasificar las redes reales de distintas formas; si analizamos el tipo de nodo y los enlaces que las forman entonces se forman los siguientes tipos de redes:

1. Redes sociales: Redes compuestas por individuos o grupos de individuos con patrones de contactos o interacciones entre ellos.
2. Redes tecnológicas: Redes como internet. También se incluyen las diseñadas para la distribución de electricidad, agua, gas, las redes de transporte, etc.
3. Redes de información: También denominadas redes de conocimiento. Un ejemplo sería las citas y co-citas de trabajos científicos. También se incluye la World Wide Web.
4. Redes biológicas: Algunos de los ejemplos de redes biológicas son las redes de reacciones metabólicas, las redes genéticas, los ecosistemas, cadenas tróficas y las redes neuronales o las vasculares.

2.1.3 Redes complejas

El tipo de redes utilizadas en el desarrollo de esta tesis son redes complejas. Las redes complejas [13, 14] son redes que poseen características no triviales, a diferencia de las redes simples, con patrones de conexión entre sus elementos que no son ni puramente regulares ni puramente aleatorios.

Algunas de estas características no triviales son las siguientes:

- Una cola pesada en la distribución de grado.
- Un alto coeficiente de agrupamiento («clustering»).
- Una estructura comunitaria a muchas escalas.
- Evidencias de una estructura jerarquizada.

La mayoría de redes sociales, tecnológicas, de información y biológicas mencionadas anteriormente cumplen estas características por lo que se definirían como redes complejas [15].

Dos clases bien conocidas y ampliamente estudiadas de estas redes son las redes libres de escala y las redes de mundo pequeño a pesar de que inicialmente todos los estudios parten de las redes aleatorias.

2.1.3.1 Redes aleatorias

En este tipo de redes, como su nombre indica, los nodos se conectan de forma aleatoria. La red aleatoria se obtiene a partir de un conjunto de N vértices aislados y de la adición de aristas entre ellos utilizando diversas estrategias, cada una de ellas proporciona un modelo de redes aleatorio.

Modelo Erdős–Rényi (ER)

El modelo Erdős–Rényi [16, 17, 18] es uno de los métodos utilizados para la generación de redes aleatorias. El modelo toma el nombre los matemáticos Paul Erdős y Alfréd Rényi.

El funcionamiento de este algoritmo es muy simple; Dado un número entero positivo n tal que $n \in \mathbb{N} - \{0\}$, y un valor de probabilidad $0 \leq p \leq 1$, se define el grafo $G(n, p)$ como un grafo no dirigido de n vértices cuyas aristas se eligen de la siguiente forma: Para todos los pares de vértices n_i, n_j se genera una arista (n_i, n_j) con probabilidad p .

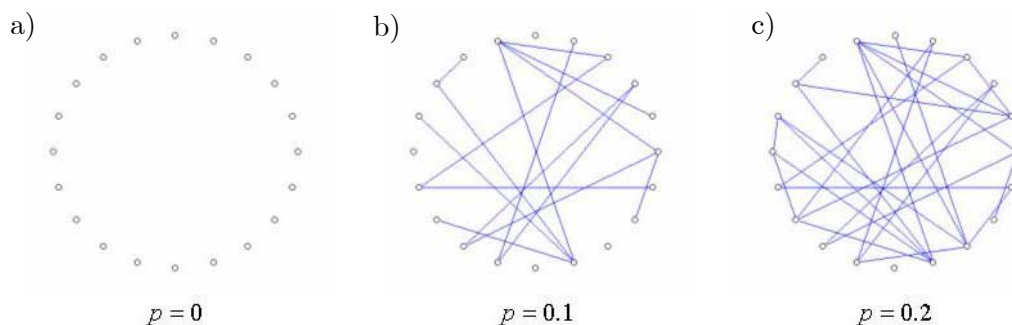


Figura 2.2. a) Ejemplo de una red de 20 nodos generada con el modelo Erdős-Rényi con probabilidad de generación de aristas $p = 0$. b) Ejemplo de una red de 20 nodos generada con el modelo Erdős-Rényi con probabilidad de generación de aristas $p = 0.1$. c) Ejemplo de una red de 20 nodos generada con el modelo Erdős-Rényi con probabilidad de generación de aristas $p = 0.2$.

En este tipo de redes la distribución de grado es una distribución de Poisson. En las redes con distribuciones poissonianas se concentra la probabilidad en torno a un valor de k (grado nodal) y decrece a una razón de $1/k!$ cuando se aleja del valor central.

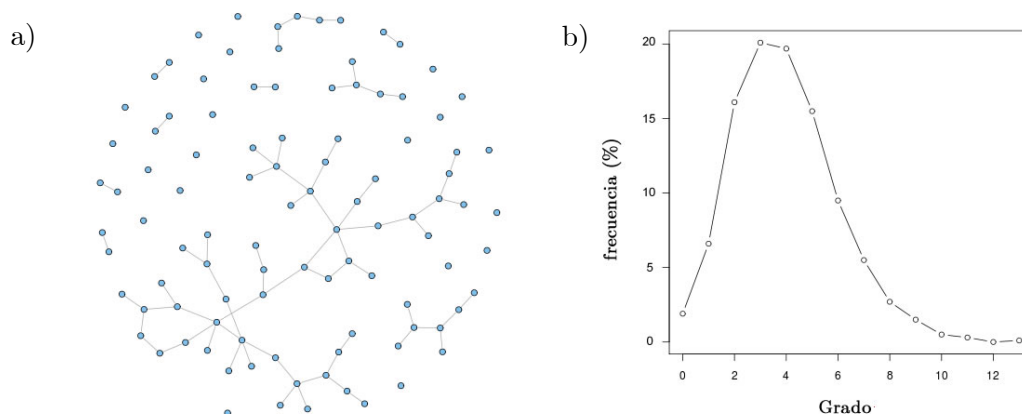


Figura 2.3. a) ejemplo de una red de 100 nodos generada con el modelo Erdős-Rényi con probabilidad de crear una arista $p = 0.015$. b) Distribución de grado de una red de 1000 nodos generada con el modelo Erdős-Rényi con la probabilidad de generar de aristas $p = 0.004$ (distribución de Poisson).

2.1.3.2 Redes libres de escala

Una de las propiedades más simples que se pueden medir directamente en una red es la distribución de grado. Como se ha descrito en la sección 2.1.3.1, una propiedad bien conocida para las redes ER es que la distribución de grado es una distribución de Poisson mientras que en las redes libres de escala esta regla no se cumple.

En este tipo de redes algunos nodos están altamente conectados, es decir, poseen un gran número de enlaces a otros nodos, aunque el grado de conexión de casi todos los nodos es bastante bajo. Esto quiere decir que la distribución de grado sigue una ley de potencia. Tenemos que tener en cuenta que los nodos con alto nivel de conectividad son críticos para el buen funcionamiento de la red ya que si se fallara uno de ellos se perdería gran parte de la conectividad entre los otros nodos.

Algunos ejemplos de este tipo de redes mediante los cuales se pueda entender bien este concepto son los siguientes:

- Las redes de comunicación como internet o la WWW o las redes de citas científicas.
- Las redes del crimen organizado, en los cuales unos cuantos "peces gordos" ordenan la actuación de muchos otros.
- Redes de correos electrónicos.

Modelo Barabási–Albert

Formalmente se denomina modelo Barabási-Albert [19] como un algoritmo para generar redes complejas libres de escala. El modelo toma el nombre de sus autores Albert-László Barabási y Réka Albert.

La red comienza con un conjunto de n_0 nodos conectados aleatoriamente. Debe notarse que $n_0 \geq 2$ y el grado de cada nodo en la red inicial debe ser al menos 1, dado que, sino, a medida que se añaden nuevos nodos a la red, el algoritmo haría que estos permanecieran desconectados completamente.

Este modelo utiliza una regla o mecanismo llamado conexión preferencial para conectar los nodos. Esto quiere decir que los nuevos nodos se enlazan

preferiblemente con los nodos más conectados. Formalmente la probabilidad p_i de que un nuevo nodo se conecte con i es:

$$p_i = \frac{k_i}{\sum_j k_j}$$

donde k_i es el grado del nodo i . Los nodos con gran cantidad de conexiones, denominados “hubs”, tienden a acumular rápidamente más enlaces, mientras que los que poseen pocos enlaces rara vez son el origen de nuevos enlaces.

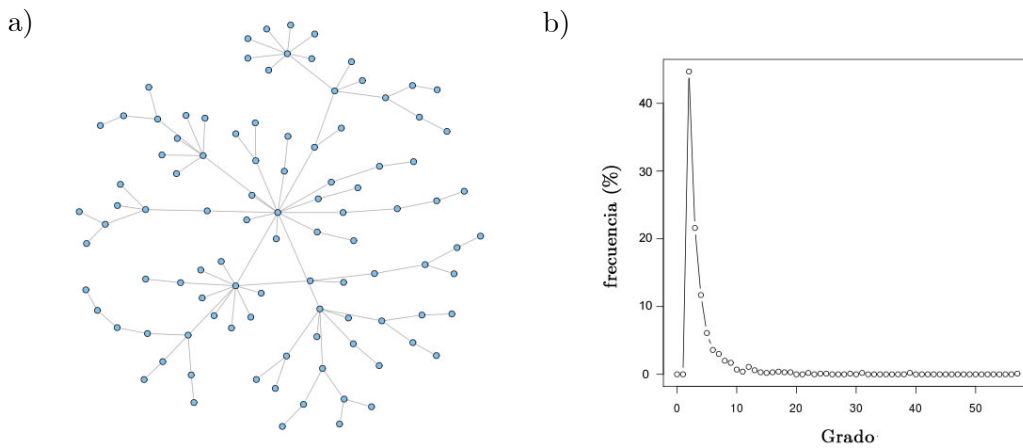


Figura 2.4. a) Ejemplo de una red libre de escala de 100 nodos generada con el modelo Barabási-Albert con $m = 1$. b) Distribución de grado de una red libre de escala de 1000 nodos generada con el modelo Barabási-Albert con $m = 2$.

Al generar la red con este modelo tenemos que introducir un valor $m \in \mathbb{N} - \{0\}$ el cual indica a cuantos nodos será enlazado el nuevo nodo en la red. Debe notarse que $m > 0$ y $m < N$, siendo N el número total de nodos que tenderá la red.

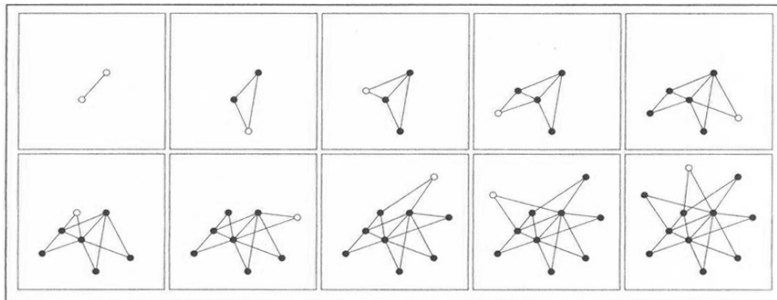


Figura 2.5. Ejemplo de la evolución en el momento de crear una red con el modelo Barabási-Albert donde $n = 11$, $n_0 = 2$ y $m = 2$.

2.1.3.3 Redes Small-World

En este tipo de redes la mayoría de nodos no son vecinos entre si y sin embargo cualquier nodo puede ser alcanzado por el nodo origen con un número relativamente corto de saltos entre ellos. Esto se consigue mediante “atajos” en la red.

Inicialmente disponemos de una red regular donde cada nodo está conectado con los k vecinos más cercanos. Añadiendo aristas entre nodos no vecinos conseguimos que el número de saltos entre dos nodos cualesquiera de la red sea más corto construyendo así una red de mundo pequeño.

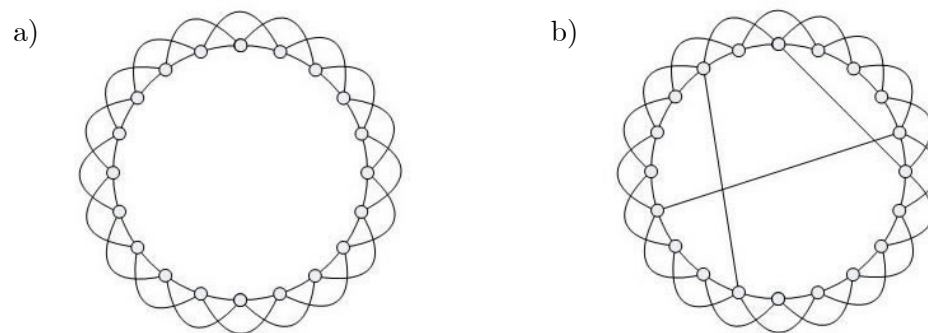


Figura 2.6. **a)** Red regular de 20 nodos con cuatro vecinos más cercanos ($k = 4$). **b)** Red de mundo pequeño de 20 nodos con cuatro vecinos más cercanos ($k = 4$) y tres enlaces puente.

Algunos ejemplos de este tipo de redes podrían ser:

- Las redes de transporte, tales como las redes de carreteras, las redes de trenes, etc.
- Redes de distribución de electricidad.

Modelo Watts-Strogatz

El modelo Watts-Strogatz [20] se utiliza para la construcción de algunas redes de mundo pequeño. El modelo toma el nombre de sus autores Duncan Watts y Steven Strogatz.

Genéricamente se trata de un modelo de generación de grafos aleatorios con distancias medias pequeñas y valores altos del coeficiente de agrupamiento.

El proceso de creación de la red es el siguiente:

Dado el número de nodos deseado N , el número de vecinos K y una probabilidad p , teniendo en cuenta $0 \leq p \leq 1$ y $N \gg K \gg \ln(N) \gg 1$, el modelo construye un grafo no dirigido de N nodos con $\frac{NK}{2}$ aristas de la siguiente forma:

1. Se construye un anillo regular con N nodos, cada uno conectado con K vecinos, $\frac{K}{2}$ en cada lado. (figura 2.6 a)
2. Para cada nodo n_i , se coge cada eje (n_i, n_j) con $i < j$, y se reconecta con probabilidad p . El reconectado se consigue reemplazando (n_i, n_j) con (n_i, n_l) donde l es escogido con probabilidad uniforme para todos los posibles valores, siempre evitando un auto-loop ($l \neq i$) y evitando duplicar un enlace. (figura 2.7 a)

La distribución de grado de estas redes en el caso del anillo es una función Delta de Dirac (figura 2.7 b), en caso de que $p \rightarrow 1$ la distribución de grado sería una distribución de Poisson (figura 2.3 b) dado que se ha convertido en una red aleatoria.

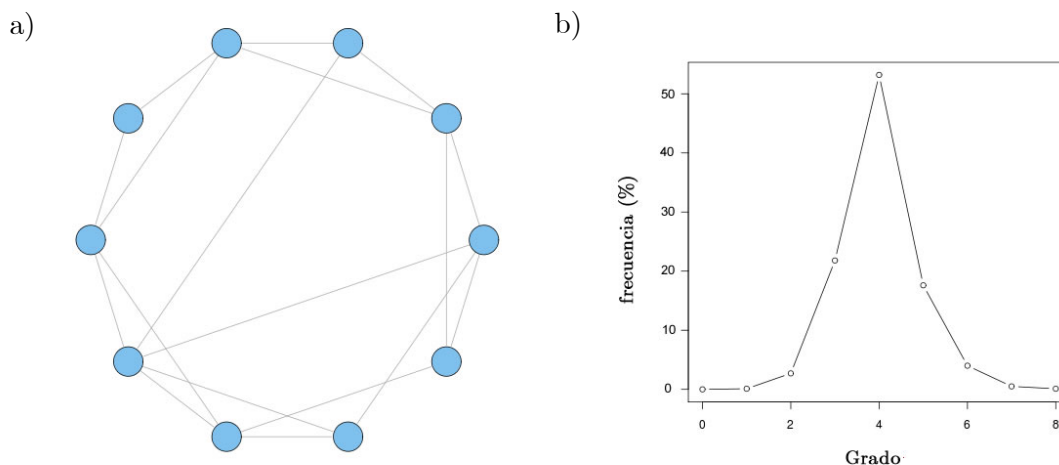


Figura 2.7. a) Ejemplo de una red de mundo pequeño con 10 nodos con $k = 2$ y $p = 0.1$ generada con el modelo Watts-Strogatz. b) Distribución de grado de una red de mundo pequeño de 1000 nodos con $k = 2$ y $p = 0.1$ generada con el modelo Watts-Strogatz.

2.2 Routing

Formalmente el routing o enrutamiento [21, 22] es el proceso de seleccionar los mejores caminos en una red. El enrutamiento se realiza en muchos tipos de redes como por ejemplo en las redes telefónicas (conmutación de circuitos), redes de datos electrónicos (conmutación de paquetes) y redes de transporte. En esta tesis nos centraremos en el enrutamiento en redes de datos electrónicos (conmutación de paquetes).

En las redes de datos electrónicos o conmutación de paquetes, el enrutamiento ordena el reenvío de paquetes desde su origen hasta su destino a través de nodos intermedios. Los nodos intermedios son habitualmente dispositivos de hardware de red tales como routers, puentes, pasarelas, cortafuegos o conmutadores. También se pueden usar ordenadores de propósito general para realizar el enrutamiento, aunque al no ser hardware especializado pueden sufrir de rendimiento limitado.

En el enrutamiento clásico, cuando se quiere enviar un mensaje desde un nodo, este saltará a uno de sus vecinos. El vecino puede ser elegido al azar o elegido de manera que la distancia (en hops) al destino del mensaje correspondiente se reduce en 1 hop. Cuando el mensaje alcanza el nuevo nodo este se pone al final de la cola, de modo que si está en la posición n –ésima, tendrá que esperar n pasos de tiempo hasta su siguiente salto. Este proceso se repite hasta que el paquete alcanza el nodo de destino. Evidentemente esto no sigue ningún tipo de protocolo por lo que en una red real esto sería poco eficiente y muy pocas veces los paquetes se enviarían por la mejor ruta posible. Debido a esto, generalmente el proceso de enrutamiento está basado en protocolos que utilizan tablas de enrutamiento que mantienen rutas hacia distintos puntos de la red. Por lo tanto, la construcción de tablas de enrutamiento, que se guardan en la memoria del router, es muy importante para el enrutamiento eficiente.

2.2.1 Selección del camino

Debido a que el enrutamiento trata de encontrar la mejor ruta posible, los algoritmos utilizan métricas [22] de la red para conseguirlo. Las métricas se calculan

mediante un algoritmo de enrutamiento y puede cubrir información como el ancho de banda, el retardo de la red, el número de saltos, el coste de la ruta, la carga, la fiabilidad y el coste de la comunicación entre otros.

Entendemos por mejor ruta aquella que cumple las siguientes condiciones:

- Consigue mantener acotado el retardo entre pares de nodos de la red subyacente.
- Consigue ofrecer altas cadencias efectivas independientemente del retardo medio de tránsito.
- Permite ofrecer el menor costo. En general, el concepto de distancia o coste de un canal es una medida de la calidad del enlace basado en la métrica que se haya definido. En la práctica se utilizan varias métricas simultáneamente para decidir el mejor camino.

2.2.2 Métodos de enrutamiento

Disponemos de dos tipos de métodos de enrutamiento: Unos son los métodos estáticos, es decir, la tabla de enrutamiento no sufre ningún cambio con respecto al momento inicial $t(0)$. Los otros son métodos dinámicos, estos métodos sí varían la tabla de enrutamiento y la adaptan a las condiciones de la red con el tiempo.

2.2.2.1 Deterministas o estáticos

No tienen en cuenta el estado de la red subyacente al tomar las decisiones de enrutamiento. Las tablas de enrutamiento de los nodos se configuran manualmente y nunca son actualizadas por el algoritmo de enrutamiento. La única forma de alterarlas es cambiándolas, también, manualmente. Esto significa que no existe ningún tipo de adaptación en tiempo real a los cambios de las condiciones de la red.

El cálculo de la ruta óptima es también off-line por lo que no importa ni la complejidad del algoritmo ni el tiempo requerido para su convergencia. Un ejemplo de estos tipos de métodos es el algoritmo de Dijkstra (algoritmo de caminos mínimos) [21, 25, 27]. Estos algoritmos son rápidos y de diseño simple, sin embargo hay que tener en cuenta que son los que peores decisiones toman en general.

2.2.2.2 Adaptativos o dinámicos

Los métodos adaptativos o dinámicos [22, 24] pueden ser más tolerantes a cambios en la red subyacente tales como variaciones en el tráfico, incremento del retardo o fallos en la topología. El enrutamiento dinámico o adaptativo se puede clasificar, a su vez, en tres subcategorías dependiendo de dónde se tomen las decisiones y del origen de la información a intercambiar:

- Adaptativo centralizado: Todos los nodos de la red son iguales excepto un nodo central que es quien recoge la información de control y los datos de los demás nodos para calcular con ellos la tabla de enrutamiento. Un gran inconveniente de este método es que consume abundantes recursos de la propia red.
- Adaptativo distribuido: En este tipo de enrutamiento, el algoritmo se ejecuta de igual forma en todos los nodos de la red. Cada nodo recalcula continuamente la tabla de enrutamiento a partir de la información obtenida y de la información que contiene en su propia base de datos. Los algoritmos por vector de distancias y los de estado de enlace son los más utilizados en internet y perteneces a este método de enrutamiento.

Algoritmos por vector de distancias

Estos métodos utilizan el algoritmo de Bellman-Ford [25, 26]. Busca la ruta de menor coste por el método de búsqueda indirecta. El vector de distancias asociado al nodo de una red, es un paquete de control que contiene la distancia a los nodos de la red conocidos hasta el momento. Cada nodo envía a sus vecinos las distancias que conoce a través de este paquete. Los nodos vecinos examinan esta información y la comparan con la que ya tienen, actualizando su tabla de enrutamiento.

Algoritmos de estado de enlace

Este tipo de enrutamiento se basa en que cada nodo llegue a conocer la topología de la red y los costes asociados a los enlaces, para que a partir de estos datos, pueda obtener el árbol y la tabla de enrutamiento tras aplicar el algoritmo de coste mínimo (algoritmo de Dijkstra) al grafo de la red.

- Adaptativo aislado: Estos tipos de enrutamiento son altamente adaptables a los cambios en la red. Cada nodo actualiza sus tablas de enrutamiento mediante, únicamente, la información que poseen.

Un ejemplo de este tipo de enrutamiento sería el enrutamiento por inundación cuyo mecanismo consiste en reenviar cada paquete recibido con destino a otros nodos por todos los enlaces excepto por el que llegó. Aunque este tipo de enrutamiento genera demasiados paquetes y consume demasiado ancho de banda.

2.2.3 Tipos de entrega de paquetes

Los esquemas de enrutamiento se diferencian en cómo estos entregan los paquetes entre los diferentes nodos de la red, clasificándose en los siguientes grupos:

- Unicast: entrega un mensaje a un único nodo específico.
- Broadcast: entrega un mensaje de difusión a todos los nodos en la red.
- Multicast: entrega un mensaje a un grupo de nodos que han expresado su interés en recibir el mensaje.
- Anycast: entrega un mensaje a cualquiera de un grupo de nodos, típicamente el más cercano a la fuente.
- Geocast: de difusión geográfica entrega un mensaje a un área geográfica.

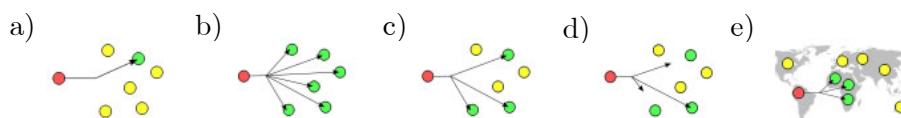


Figura 2.8. a) Unicast. b) Broadcast. c) Multicast. d) Anycast. e) Geocast.

Esta tesis se centra en los algoritmos de enrutamiento unicast.

2.2.4 Congestión

La congestión de redes [28, 29] es el fenómeno producido cuando a la red se le ofrece más tráfico del que puede procesar. Evidentemente la congestión de una red depende de la topología de la red subyacente así como de los protocolos de enrutamiento entre otras cosas.

El estado de congestión de una red puede medirse utilizando un típico parámetro de orden C , definido como [30]:

$$C = \lim_{t \rightarrow \infty} \frac{A(t + \tau) - A(t)}{\tau \cdot p \cdot N}$$

donde $A(t)$ es el número de mensajes de "activos" en la red, es decir, el número total de mensajes que se encuentran en las colas de los nodos, p es la probabilidad de generar un nuevo mensaje (paquete) y N es el número total de nodos de la red. Básicamente se toma una ventana temporal de duración τ pasos («steps») y se mide el número de activos al inicio ($A(t)$) y al final de la ventana ($A(t + \tau)$). Se hace la resta $A(t + \tau) - A(t)$ y se normaliza por el número de paquetes que se han inyectado en el sistema durante esta ventana temporal ($\tau \cdot p \cdot N$). De esta manera el parámetro es $C = 0$ si no hay congestión y $C > 0$ si hay congestión, siendo $C = 1$ su valor máximo debido a la normalización.

Mediante este parámetro se puede seguir la evolución $C(p)$, que normalmente produce una transición de fase en algún p_c separando la fase de **flujo libre** ($C = 0$) y el estado de **congestión** ($C > 0$). En un estado de **flujo libre** la red es capaz de equilibrar el flujo de ingresos de nuevos mensajes con la tasa de entrega. Sin embargo, cuando $p > p_c$ la cantidad de mensajes nuevos excede de los entregados y, por tanto, comienzan a acumularse en las colas de los nodos a una velocidad constante C .

2.3 Mecánica cuántica

Otro tema a tener en cuenta en esta tesis es la mecánica cuántica que hay detrás del funcionamiento de las redes cuánticas. La mecánica cuántica en si es un tema muy extenso por lo que en esta sección se explicaran únicamente aquellos aspectos relevantes para comprender el funcionamiento de las redes cuánticas que planteamos en este proyecto. Evidentemente no se entrará a explicar temas fisicos sino más bien a explicar algunos conceptos matemáticos y de funcionamiento esenciales.

2.3.1 Qubit

Para entender el funcionamiento de la física cuántica aplicada a la informática, primero hay que saber y entender que es un qubit [31]. Matemáticamente un qubit, denominado así por la simplificación de «quantum bit», es un estado cuántico en un espacio vectorial complejo bidimensional. Un qubit es la unidad mínima de información cuántica. Sus dos estados básicos se llaman, convencionalmente, $|0\rangle$ y $|1\rangle$ (ket cero y ket uno).

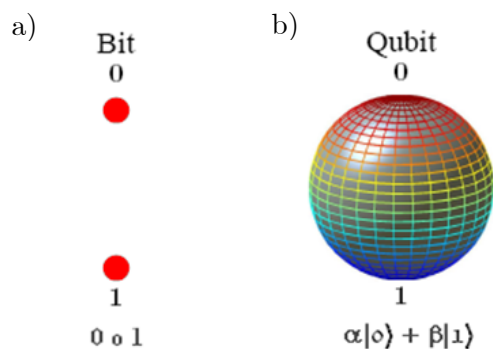


Figura 2.9. a) Ejemplo de un bit clásico con sus posibles valores (0 o 1). b) Ejemplo de un qubit presentado como una combinación lineal de sus dos estados ($|0\rangle$ y $|1\rangle$).

Dado que la definición puede resultar confusa si no se tienen conocimientos previos, lo más fácil para entender que es un qubit es compararlo con un bit clásico. Los dos estados básicos de un qubit $|0\rangle$ y $|1\rangle$ corresponden al 0 y 1 del bit clásico con la diferencia de que en un bit clásico, este solo puede tomar los valores 0 o 1 y un qubit puede encontrarse en un estado de superposición cuántica, esto significa que un qubit puede tener una combinación de esos dos estados ($\alpha|0\rangle + \beta|1\rangle$). En

resumen, un bit contiene un valor (0 o 1), y un qubit contiene, a la vez, ambos valores (0 y 1).

Como se ha dicho, un qubit se presenta como una superposición o combinación lineal de los estados básicos $|0\rangle$ y $|1\rangle$, y se representa de la siguiente forma:

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$$

donde α y β son números complejos que determinan, respectivamente, la probabilidad de obtener en una medida los resultados $|0\rangle$ y $|1\rangle$. Lógicamente la probabilidad total tiene que ser la unidad, por lo tanto α y β se deben relacionar por la siguiente ecuación:

$$|\alpha|^2 + |\beta|^2 = 1$$

Esta ecuación simplemente asegura que en la medición se obtiene un estado o el otro. Hay que tener en cuenta que debido a su naturaleza cuántica, cualquier medida del qubit altera inevitablemente su estado: se rompe la superposición y colapsa en aquel estado de base que ha resultado de la medida, y $\{\alpha, \beta\}$ se transforma irreversiblemente en $\{0,1\}$.

2.3.2 Representación física

Ahora que ya sabemos que es un qubit solo nos falta saber cómo se representa físicamente [34]. Hay varias formas de hacerlo por lo que solo se expondrán dos con el objetivo de entender el concepto:

- Fotón: Se utiliza la polarización de la luz para codificar el qubit. Un fotón con polarización horizontal equivale al estado $|0\rangle$ y un fotón con polarización vertical equivale al estado $|1\rangle$
- Electrón: En este caso se utiliza el espín (momento angular intrínseco) de un electrón para codificar el qubit, Un electrón con spin arriba equivale al estado $|0\rangle$ y un electrón con espín abajo equivale al estado $|1\rangle$

2.3.3 Entrelazamiento

Otra característica que diferencia un qubit de un bit clásico es que múltiples qubits pueden presentarse en un estado de entrelazamiento cuántico [32, 33].

El entrelazamiento es un fenómeno cuántico (en este caso no hay equivalente clásico) en el cual los estados cuánticos de dos o más objetos (en nuestro caso qubits) se deben describir mediante un estado único que involucra a todos los objetos del sistema. Para que dos qubits estén máximamente entrelazados debe de cumplir uno de los siguientes estados de Bell [34]:

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

$$|\Phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$$

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$$

$$|\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$$

Figura 2.10. Estados de Bell que representan dos qubits máximamente entrelazados.

Debido al entrelazamiento, dos partículas subatómicas permanecen inevitablemente relacionadas entre sí, si han sido generadas en un mismo proceso. Cuando una de las dos partículas sufre un cambio de estado, la otra lo sufre instantánea y automáticamente, independientemente de la distancia que las separe una de la otra en el momento de realizar la medición. Esto es así, la distancia no importa, estas dos partículas podrían encontrarse a escasos centímetros o en distintos puntos del universo y su comportamiento sería exactamente el mismo.

Esto se entiende mejor con un ejemplo. Para este ejemplo utilizaremos fotones donde su estado va determinado por la polarización tal y como se ha descrito en la sección 2.3.1.1.

Si se preparan una pareja de fotones máximamente entrelazados en el estado de Bell $|\Phi^+\rangle$, estos estarán polarizados a la vez horizontal y verticalmente. En el momento de medir uno de los fotones este colapsará en uno de sus dos estados, por

ejemplo en el estado de polarización horizontal ($|0\rangle$), influyendo instantáneamente en el otro fotón, sabiendo desde ahora y de antemano que este otro fotón también tendrá polarización horizontal ($|0\rangle$), pese a la imposibilidad de predecir, según los postulados de la mecánica cuántica, qué estado cuántico se observará.

Si hubiéramos creado la pareja de fotones máximamente entrelazados en el estado de Bell $|\Psi^+\rangle$ o $|\Psi^-\rangle$, cuando en un fotón obtuviésemos $|0\rangle$ en el otro obtendríamos $|1\rangle$ y viceversa.

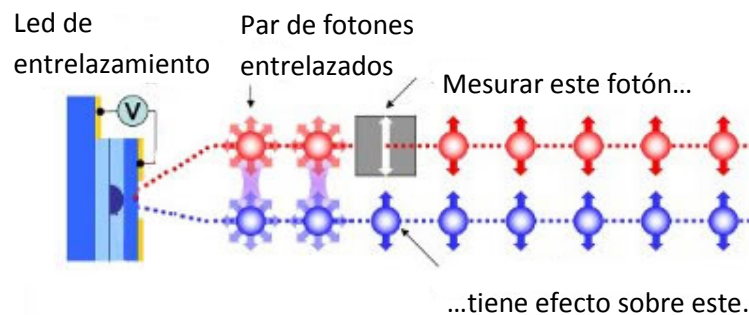


Figura 2.11. Ejemplo de cómo se comporta un par de fotones entrelazados. A partir del momento en que se mide uno de ellos, ambos fotones tendrán siempre la misma polarización.

2.3.4 Teleportación

La teleportación cuántica [6, 35] es una tecnología que consiste en enviar el estado cuántico de una partícula hacia otra, utilizando, para ello, un par de partículas entrelazadas y cierta información clásica. Debe quedar claro que no se teletransporta la partícula en sí, sino únicamente su estado cuántico, y que nunca se podrá enviar información a mayor velocidad que la de la luz debido a que el envío de información clásica se realiza por un canal clásico, por ejemplo internet.

El proceso de teleportación es el siguiente; imaginemos un sistema con 2 actores, en el caso de esta tesis suponemos que estos actores son nodos capaces de generar y enviar pares máximamente entrelazados y mantenerlos guardados hasta el momento de su uso.

1. El Nodo 1 prepara un par de partículas máximamente entrelazadas (A) en uno de los estados de Bell, por ejemplo:

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

2. El Nodo 1 se queda con la partícula $|A1\rangle$, envía la partícula $|A2\rangle$ al Nodo 2 y crea el qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ que será el que teletransportará hacia el Nodo 2.

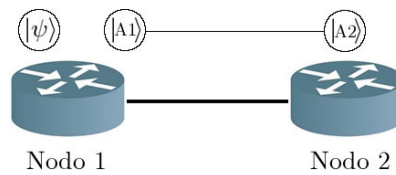


Figura 2.12. Estado de una red antes de realizar una teleportación, donde el Nodo 1 tiene dos qubits, uno de ellos entrelazado con otro en el Nodo 2.

3. El Nodo 1 aplica la compuerta cuántica [34, 36] CNOT a sus dos qubits.
4. El Nodo 1 aplica la compuerta cuántica Hadamard al primero de sus dos qubits ($|\psi\rangle$).
5. El Nodo 1 realiza una medición sobre ambos qubits y obtiene los dos bits clásicos $b_1 b_2$, que envía al Nodo 2 por un canal de comunicación clásico (comúnmente internet).
6. El Nodo 2 aplica la transformación $Z^{b_1} X^{b_2}$ sobre su qubit $|A2\rangle$, de acuerdo a los bits recibidos $b_1 b_2$ donde X es la matriz de Pauli σ_x y Z la matriz de Pauli σ_z . El resultado obtenido en su qubit será $|\psi\rangle$.

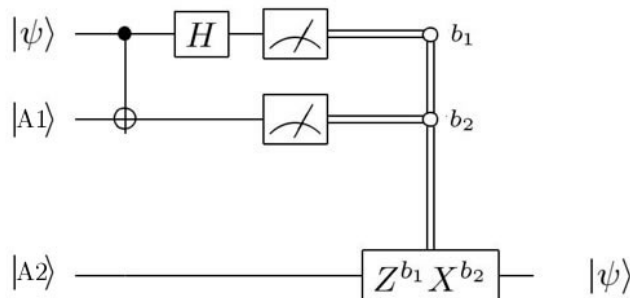


Figura 2.13. Esquema de teleportación cuántica.

2.3.5 Intercambio de entrelazamiento

El intercambio de entrelazamiento [2, 3, 4] consiste en entrelazar dos partículas que nunca han estado en contacto anteriormente. Esto se consigue de la siguiente forma:

Llamaremos $|A\rangle$ y $|B\rangle$ a los pares de partículas máximamente entrelazados generados por fuentes de luz independientes. Entonces $(|A1\rangle, |A2\rangle)$ es el par entrelazado $|A\rangle$ y $(|B1\rangle, |B2\rangle)$ es el par entrelazado $|B\rangle$. Como se ha dicho $|A\rangle$ y $|B\rangle$ no han tenido ningún contacto en el pasado por lo que cualquier medición sobre $|A1\rangle$ o $|A2\rangle$ no afectará de ningún modo sobre $|B1\rangle$ o $|B2\rangle$ y viceversa. Ahora imaginemos un sistema de 3 nodos, donde estas partículas se reparten de la siguiente forma; El Nodo 1 genera el par $|A\rangle$, se queda con $|A1\rangle$ y envía $|A2\rangle$ al Nodo 2. El Nodo 2 genera el par $|B\rangle$, se queda con $|B1\rangle$ y envía $|B2\rangle$ al Nodo 3:

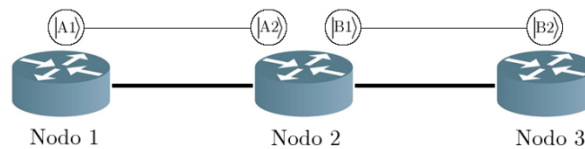


Figura 2.14. Pares de partículas máximamente entrelazadas y distribuidas entre varios nodos.

Ahora, con este esquema, queremos comunicar el Nodo 1 con el Nodo 3. En este momento es imposible ya que no existe ningún tipo de relación entre los pares $|A\rangle$ y $|B\rangle$, por lo tanto, el Nodo 2 tendrá que hacer un intercambio de entrelazamiento; El Nodo 2 realiza una medición proyectiva sobre sus dos partículas ($|A2\rangle$ y $|B1\rangle$) en una base de Bell (figura 2.11) con un analizador de estado de Bell (una medición de Bell es una operación de entrelazamiento), y luego comunica el resultado mediante un canal clásico al Nodo 3. Tal y como se ha descrito en la sección 2.3.3, el estado cuántico de la partícula $|A2\rangle$ se envía por teleportación a la partícula $|B2\rangle$. Ahora las partículas $|A1\rangle$ y $|B2\rangle$ están máximamente entrelazadas creando un canal de comunicación cuántico desde el Nodo 1 al Nodo 3. Debe notarse que al realizar una medición sobre las partículas ($|A2\rangle$ y $|B1\rangle$), estas dejan de existir, por lo tanto, como se muestra en la figura 2.16, ya no existe ningún canal (partículas entrelazadas) entre el Nodo 2 y los Nodos 1 y 3.

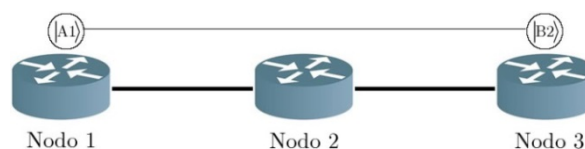


Figura 2.15. Partículas entrelazadas después de un intercambio de entrelazamiento.

Capítulo 3

Estado del arte

3.1 Redes y tráfico

Desde hace varias décadas, el estudio de las propiedades topológicas y dinámicas de las redes complejas ha recibido una gran cantidad de interés. Parte de este interés proviene del intento de comprender la topología y el comportamiento de las redes, tales como las redes sociales y las redes tecnológicas, entre otras.

En cuanto a las redes tecnológicas, como por ejemplo internet, se nos plantean una nueva e importante cuestión, la congestión. Así es, la congestión que se produce en la red cuando hay cierto tráfico. Muchas investigaciones se han centrado en intentar encontrar topologías de red óptimas para reducir esta congestión, entre ellas en [37], Roger Guimerà et al. se plantearon una cuestión; si dado un algoritmo de búsqueda que utilice solamente información local y un número fijo de nodos y enlaces, ¿cuál sería la topología óptima para el proceso de búsqueda? En el análisis consideraron una situación general en la que la red tiene que hacer frente a varios problemas de búsqueda paralelos, lo que plantea, simultáneamente, problemas congestión. En el estudio definen que, obviamente, la topología óptima para un solo problema de búsqueda sería una red de estrella, donde todos los nodos estén conectados a un nodo central. Este tipo de estructuras es ‘*barata*’ de montar

en términos de enlaces y eficiente en términos de búsqueda, ya que el número de pasos (costo) para encontrar cualquier otro nodo sería siempre 2, independiente del tamaño de la red. Sin embargo, esta estructura de estrella es muy ineficiente cuando hay muchas búsquedas en paralelo, debido a la limitación del nodo central. Por lo tanto, concluyen que la topología de red óptima para un gran número de búsquedas paralelas es una red homogénea donde todos los nodos tienen esencialmente el mismo grado, intermediación, etc.

Medidas de centralidad, como la intermediación, de los nodos en una red contiene información muy valiosa cuando múltiples búsquedas se realizan en paralelo y se vuelve relevante considerar la congestión. Muchos estudios demuestran que las redes reales poseen topologías de conexión libres de escala (ver sección 2.1.3.2), en las redes reales, el estado de la densidad del tráfico surge de una interacción entre la dinámica del flujo y la estructura de red subyacente.

En [38] se investiga la relación entre las medidas de centralidad y la densidad del tráfico para algunos modelos de redes libres de escala, además de cómo la velocidad de la dinámica se ve afectada por la estructura de la red subyacente. La libre circulación del tráfico en una red compleja es clave para su funcionamiento normal y eficiente, en [39] presentan unos modelos para abordar como la topología de una red compleja influye en la dinámica del flujo de tráfico, teniendo en cuenta la topología de la red, la tasa de generación de información, y la capacidad de procesamiento de información de nodos individuales. En el estudio se encuentran con que hay una tasa crítica de generación de información (punto crítico), por debajo del cual el tráfico en la red es libre, pero por encima del cual se produce congestión. También concluyen, entre otras cosas, que las redes libres de escala y las redes aleatorias son más tolerantes a la congestión.

La congestión en las redes es un tema ampliamente estudiado, en [40] se define un modelo de flujo de tráfico en redes complejas con el fin de estudiar el equilibrio entre las estrategias de enrutamiento basadas en la topología y las basadas en el tráfico, llegando a concluir que el control del tráfico mejora el rendimiento global, ampliando la zona de flujo libre (sólo en redes heterogéneas), además de que el control de tráfico presenta efectos no lineales y puede desencadenar la aparición de una fase congestionada de manera discontinua.

La congestión en las redes a menudo viene de repente y provoca que el sistema deje de funcionar, esto hace necesario el desarrollo de técnicas de enrutamiento que mejoren la eficiencia del transporte de información para tratar de evitarla. En [41] en lugar de utilizar una estrategia de enrutamiento de caminos mínimos, proponen un algoritmo de enrutamiento generalizado para encontrar la llamada ruta eficiente. Esta ruta eficiente considera la posible congestión en los nodos a lo largo los diferentes caminos. Puesto que los nodos con el mayor grado son muy susceptibles a la congestión, una forma eficaz de mejorar la congestión es redistribuir la carga de tráfico de los nodos centrales a otros nodos no centrales. Los resultados de los estudios indican que con este procedimiento se mejora el procesamiento de tráfico hasta 10 veces. Otras estrategias incorporan información local sobre el tráfico a través de un parámetro ajustable [42] para explorar cómo el rendimiento medio de la red depende de la habilidad del protocolo de enrutamiento para desviar el tráfico a través de caminos que no sean los más cortos.

3.2 Distribución de claves cuántica

Hoy en día las comunicaciones seguras son cada vez más importantes para los comunicadores con el fin de no ser interceptadas por intrusos. La criptografía cuántica [5] promete revolucionar el problema de la distribución de claves en el sistema criptográfico, proporcionando un canal de comunicación seguro entre dos partes. La criptografía cuántica, al igual que en la criptografía clásica, el objetivo siempre ha sido garantizar la absoluta confidencialidad de la información que se desea transmitir. Para lograr esto, se necesita, previamente, hacer una distribución de claves segura.

La distribución de clave cuántica es una tecnología que utiliza las propiedades de la mecánica cuántica para realizar una importante primitiva criptográfica: la distribución de claves. A diferencia de las técnicas utilizadas en la criptografía tradicional "clásica", donde la seguridad se basa en la complejidad computacional no demostrada de ciertas funciones matemáticas, la seguridad de QKD se puede probar formalmente. Las claves secretas establecidas mediante la distribución de claves cuántica son información teóricamente segura, lo que implica

que cualquier adversario tratando de espiar (eavesdropping) no puede obtener ninguna información sobre las claves.

Los canales de comunicación necesarios para realizar la distribución de claves cuántica consisten en un canal óptico, con el que se intercambian estados cuánticos bien controlados de luz (fotones), y un canal clásico que se utiliza para la señalización durante los intercambios cuánticos y para la fase de reconciliación de la clave. En este sentido, la distribución de claves cuántica es por naturaleza una tecnología de telecomunicaciones.

En la actualidad hay unos cuantos protocolos para realizar la distribución de claves cuántica, los primeros que se desarrollaron y los más conocidos son el protocolo BB84 [43] y el E91 [44]. Siguiendo estos dos como modelo inicial, se han propuesto muchos más con el fin de incrementar la seguridad [45] ante posibles ataques.

El protocolo BB84 se basa en el envío de fotones de luz polarizados mediante un canal cuántico, por ejemplo de fibra óptica. Además, como se ha dicho, se necesita un canal clásico, como por ejemplo internet, para el envío de información para que ambos puedan construir la clave secreta compartida. Ninguno de los dos canales tiene que ser seguro ya que un ataque (eavesdropping) sobre estos no proporcionaría ninguna información útil a una tercera persona.

En cuanto al protocolo E91, este se basa en enviar fotones entrelazados. El esquema es similar al BB84 pero en este se necesita una fuente de fotones entrelazados. La fuente puede estar tanto en el emisor, en el receptor, o en cualquier otro lugar, lo importante es que un fotón del par entrelazado llegue al nodo origen de la comunicación y el otro llegue al nodo destino. Debido a la correlación que existe entre los fotones (supongamos que máximamente entrelazados), cuando los usuarios hagan una medición sobre su fotón, siempre obtendrán el mismo resultado (o diferente dependiendo del estado de Bell en el que haya sido generado). La seguridad viene garantizada porque antes de la medición es imposible predecir qué estado se obtendrá ($|0\rangle$ o $|1\rangle$).

3.3 Redes cuánticas

Las redes cuánticas descienden del estudio de la criptografía cuántica. Ampliar el rango de distribución de claves cuánticas (QKD) a distancias muy largas, y que permita el intercambio de claves secretas entre múltiples usuarios requiere del desarrollo de una infraestructura de red que conecte múltiples enlaces QKD individuales. Los enlaces QKD están adaptados al intercambio de claves punto a punto entre los nodos del canal cuántico. Las transmisiones en dichos enlaces están limitadas, actualmente, a unos pocos centenares de kilómetros [8, 48].

Una red de distribución de claves cuántica es una infraestructura compuesta por enlaces QKD, es decir, pares de dispositivos QKD unidos por un canal cuántico y un canal de comunicación clásico (p. ej. internet). Estos enlaces se utilizan, conjuntamente, para conectar múltiples nodos distantes. Sobre la base de estos recursos y el uso de protocolos apropiados, esta infraestructura puede permitir la distribución incondicionalmente segura de claves secretas simétricas entre cualquier par de usuarios legítimos de la red.

En la actualidad todos los bloques necesarios para la construcción de redes cuánticas ya se han demostrado, y de hecho, las redes cuánticas a pequeña escala ya son una realidad. Una de las primeras implementaciones es la red cuántica de DARPA [49] operativa desde el año 2003, que conecta la universidad de Harvard, la universidad de Boston y BBN [50]. Esta red se compone, actualmente, de un total de 10 nodos y soporta tanto enlaces de fibra óptica como enlaces en el espacio libre. Otros ejemplos de redes cuánticas son la SECOCQ en Viena [1], que consta de 6 nodos conectados por enlaces de entre 6 y 85 kilómetros (ver figura 1.1), y la SwissQuantum en Ginebra [51], compuesta de 3 nodos conectados por fibra óptica, instalada en la zona metropolitana de Ginebra en marzo de 2009 y cuyo propósito principal fue validar la fiabilidad y la robustez de la distribución de claves cuántica en operación continua en una red durante un largo período de tiempo.

Las redes cuánticas o redes de distribución de claves cuántica se pueden clasificar en dos grupos generales [52]:

- Las redes que crean un canal cuántico punto a punto entre dos usuarios.

- Las redes que requieren del transporte de la clave a través de muchos nodos intermedios (repetidores) de confianza.

En el primer grupo, encontramos las redes en las que se aplica una función óptica clásica, como la conmutación, a nivel de nodo, en las señales cuánticas enviadas sobre un canal cuántico. Este enfoque permite QKD multiusuario, pero no puede ser utilizado para ampliar la distancia de distribución de la clave. Otra clase de este tipo de redes son las redes QKD basadas en *repetidores cuánticos*. Los repetidores cuánticos [46, 47] pueden crear un canal cuántico punto a punto mediante la distribución del entrelazamiento [2] entre dos usuarios de la red, combatiendo, así, las pérdidas en la propagación que afectan a las señales cuánticas cuando viajan por un canal cuántico. La implementación de repetidores cuánticos, sin embargo, requiere operaciones cuánticas complejas y memorias cuánticas [53], cuya realización sigue siendo un reto experimental.

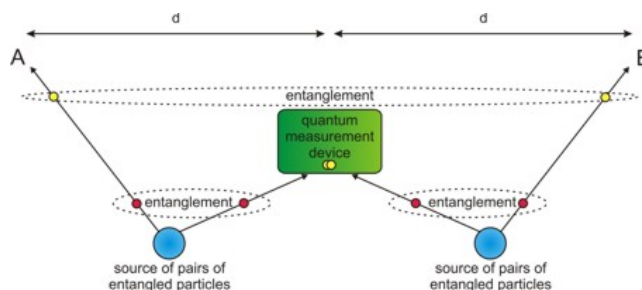


Figura 3.1. Repetidor cuántico.

Los repetidores cuánticos se basan en el intercambio de entrelazamiento entre dos partículas que no han tenido contacto previo. Recientemente se ha logrado realizar un intercambio de entrelazamiento entre dos pares de fotones máximamente entrelazados, previa distribución de uno de ellos mediante un canal cuántico de larga distancia (~ 145 km) en el espacio libre [54]. Esto se ha hecho, según el artículo, bajo la influencia de condiciones ambientales muy exigentes, de hecho, con condiciones más desfavorables que las que se esperaría encontrar en un enlace

cuántico vía satélite. Esto demuestra la viabilidad de repetidores cuánticos, en un futuro internet cuántico, tanto en tierra como en el espacio.

Otro experimento reciente ha logrado teletransportar el estado cuántico (polarización) de un fotón a una memoria cuántica de estado sólido (elemento clave en los repetidores y, por lo tanto, en las redes cuánticas de larga distancia), cuyos fotones entrelazados se han distribuido mediante un canal de fibra óptica común de 12.5 km [55].

La combinación de la teleportación cuántica con las memorias cuánticas provee de esquemas altamente escalables para las redes cuánticas. Con estos experimentos explicado aquí y muchos otros más ya se ha demostrado el correcto funcionamiento de todos los ingredientes necesarios para crear redes cuánticas punto a punto con repetidores, incluso con la fibra óptica común que ya hay distribuida actualmente por todo el mundo, ya sea para el intercambio de claves o para una comunicación cuántica distinta. Actualmente el problema reside en el tiempo que las memorias cuánticas pueden almacenar el estado de una partícula. En el experimento anterior este tiempo es de unos 50ns, aunque en un experimento con electrones, en lugar de fotones, se ha logrado alcanzar los 39 minutos [56].

El otro tipo de redes de distribución de claves cuánticas son las basadas en *repetidores de confianza*, de hecho, todas las redes explicadas anteriormente [1, 49, 51] siguen este modelo. En estas redes, los nodos actúan como relés de confianza que almacenan localmente las claves generadas con la de distribución de claves cuántica en memorias clásicas, y luego utilizan estas claves para realizar la distribución de claves a larga distancia entre dos nodos de la red. Por lo tanto, las redes de distribución de claves cuántica basadas en *repetidores de confianza* no requieren nodos equipados con memorias cuánticas; sólo requieren dispositivos de distribución de claves cuántica y memoria clásica, así como unidades de procesamiento colocadas dentro de lugares seguros. Estas características permiten que este tipo de redes se puedan implementar con las tecnologías disponibles en la actualidad.

El enrutamiento en redes QKD basadas en *repetidor de confianza* se realiza según el siguiente principio general: En primer lugar, se generan las claves locales a través de los enlaces QKD y se almacenan en los nodos que se colocan en ambos extremos de cada enlace. La distribución de claves global se realiza sobre un camino QKD, es decir, una cadena unidimensional de relés de confianza conectados por enlaces QKD, estableciendo una conexión entre dos nodos finales. Las claves secretas se remiten, salto a salto, a lo largo de los enlaces QKD que forman el camino. De este modo las claves se envían de forma segura, punto a punto, siempre que los nodos intermedios sean de confianza.

Capítulo 4

Propuesta

En esta sección se describe nuestra propuesta para el estudio de la congestión en las redes cuánticas. Para ello, primero hemos definido un conjunto de redes creadas con los modelos Erdős–Rényi, Barabási–Albert y Watts–Strogatz con distintas configuraciones dado que se desea comprobar cuáles de estas configuraciones son más propensas a la congestión y cuáles no.

En segundo lugar vamos a definir varios modelos de red cuántica, obviamente, nuestros modelos descuidan algunos ingredientes tecnológicos, pero tiene como objetivo la incorporación de nuevas características que las tecnologías de comunicación cuántica ofrecen.

Finalmente definiremos algunos protocolos de enrutamiento para estudiar cómo estos afectan a la congestión en las distintas redes.

4.1 Redes

A continuación se describirán los tipos de redes utilizados en esta tesis para el análisis de la congestión, junto con sus respectivos parámetros. Estas redes están definidas en la sección [2.1.3](#). Todas ellas contienen un total de 300 nodos.

4.1.1 Redes Erdős–Rényi

Como se ha descrito en la sección 2.1.3.1, el modelo Erdős–Rényi tiene un parámetro z que indica la probabilidad de crear un enlace entre dos nodos.

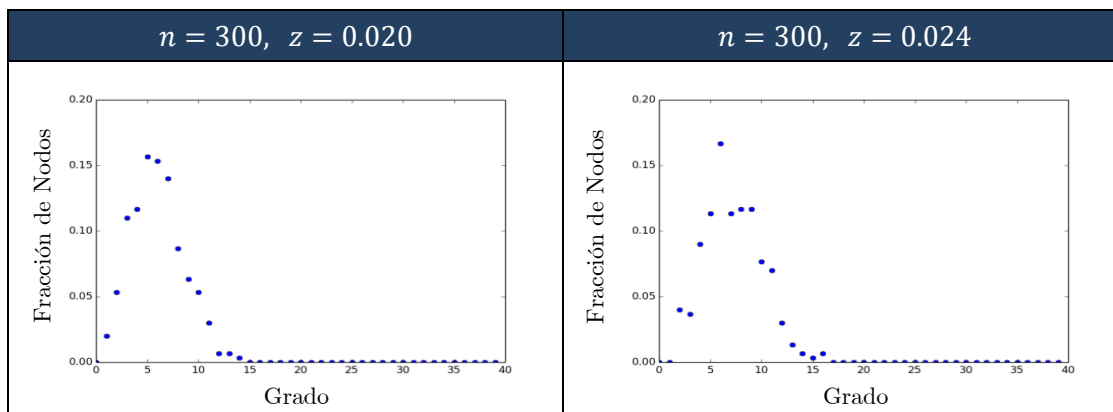


Figura 4.1. Distribución de grado en una red de 300 nodos creada con el modelo Erdős–Rényi, con $z=0.02$ (derecha) y $z=0.024$ (izquierda)

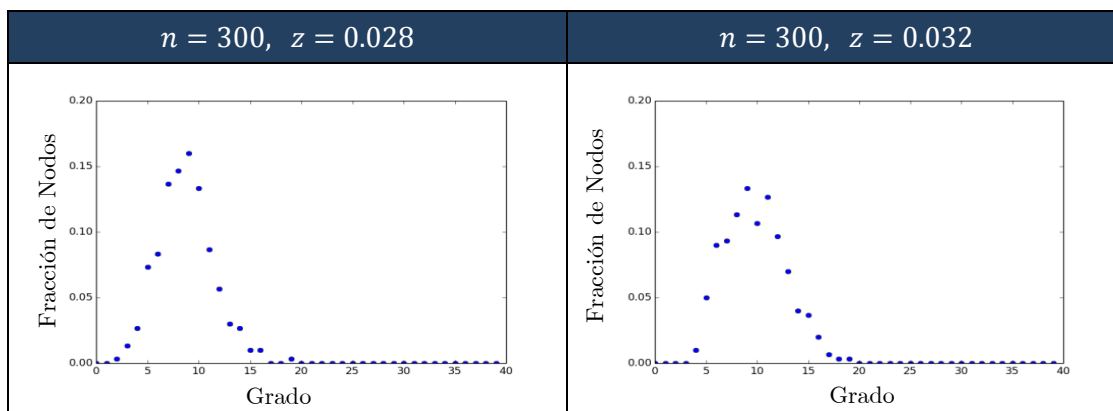


Figura 4.2. Distribución de grado en una red de 300 nodos creada con el modelo Erdős–Rényi, con $z=0.028$ (derecha) y $z = 0.032$ (izquierda)

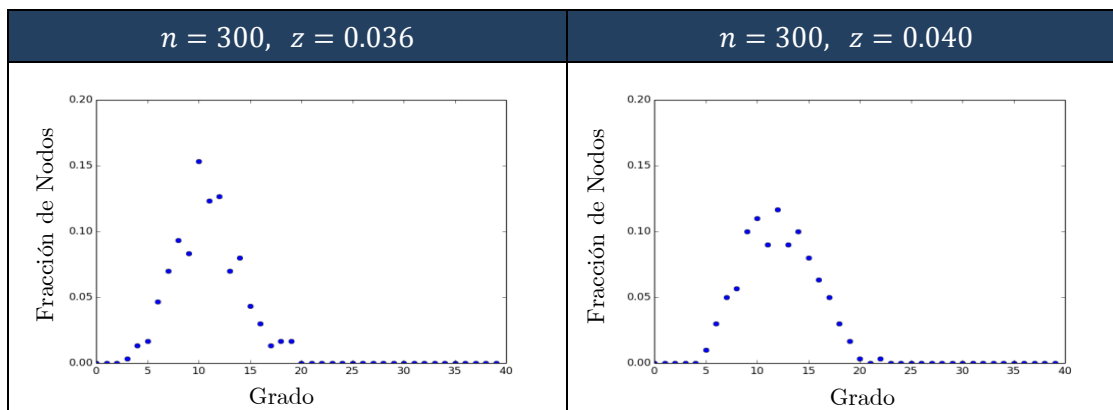


Figura 4.3. Distribución de grado en una red de 300 nodos creada con el modelo Erdős–Rényi con, $z=0.036$ (derecha) y $z = 0.040$ (izquierda)

4.1.2 Redes Barabási–Albert

Como se ha descrito en la sección 2.1.3.2, el modelo Barabási–Albert tiene un parámetro m que indica a cuantos nodos se conectará el nuevo nodo al crear la red.

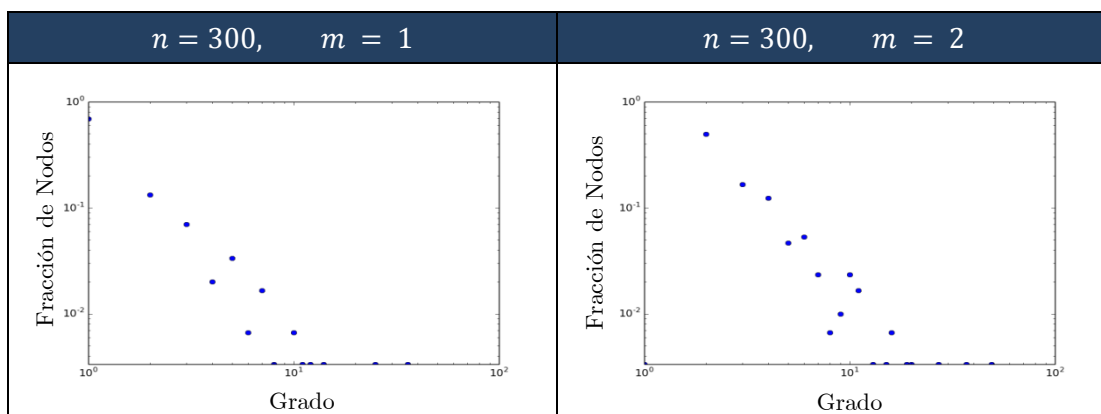


Figura 4.4. Distribución de grado en una red de 300 nodos creada con el modelo Barabási–Albert, con $m=1$ (derecha) y $m = 2$ (izquierda)

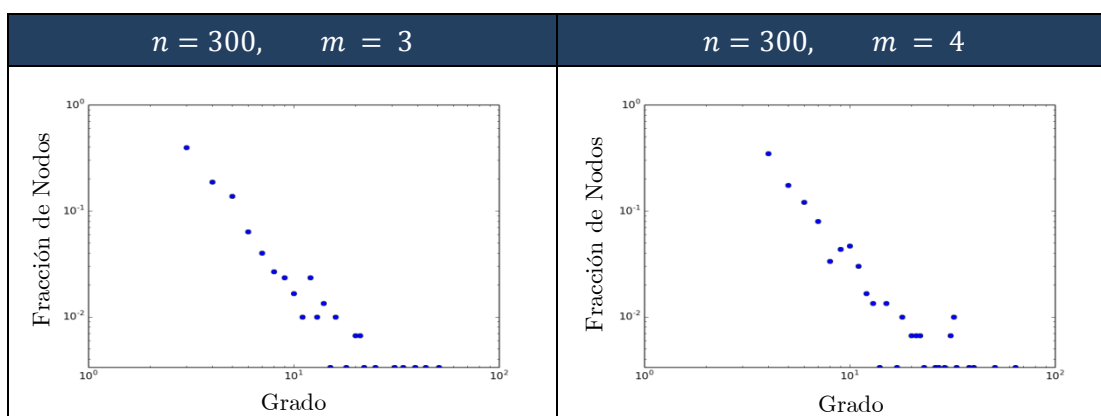


Figura 4.5. Distribución de grado en una red de 300 nodos creada con el modelo Barabási–Albert, con $m=3$ (derecha) y $m = 4$ (izquierda)

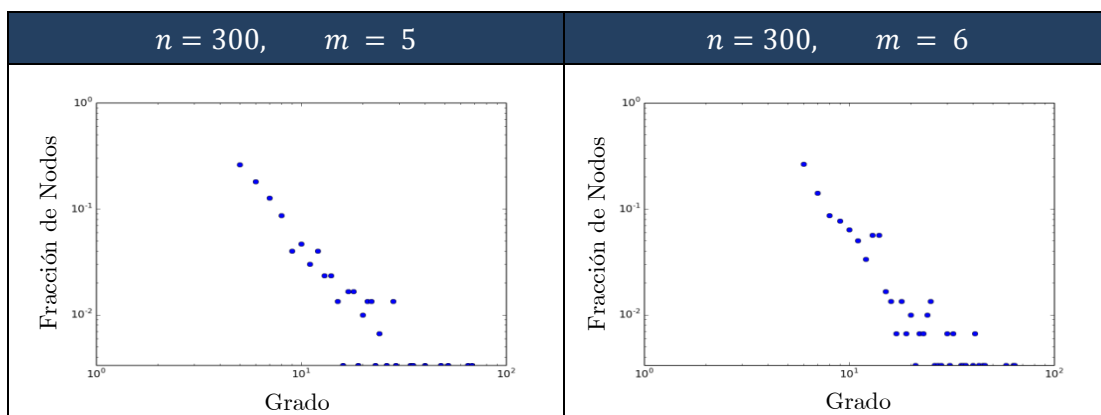


Figura 4.6. Distribución de grado en una red de 300 nodos creada con el modelo Barabási–Albert, con $m=5$ (derecha) y $m = 6$ (izquierda)

4.1.3 Redes Watts-Strogatz

Como se ha descrito en la sección 2.1.3.3, el modelo Watts-Strogatz tiene un parámetro k (vecinos) u un parámetro p (probabilidad de reconexión).

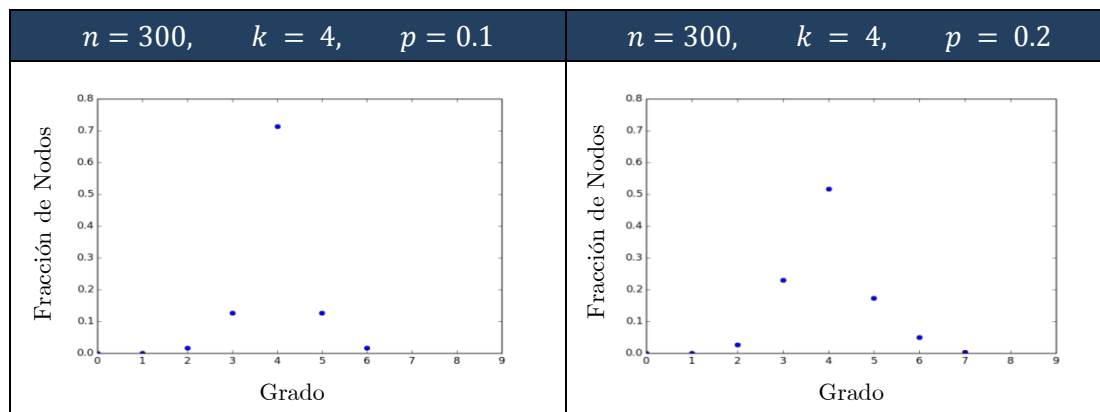


Figura 4.7. Distribución de grado en una red de 300 nodos creada con el modelo Watts-Strogatz, con $k=4$ y $p=0.1$ (derecha) y $k=4$ y $p=0.2$ (izquierda)

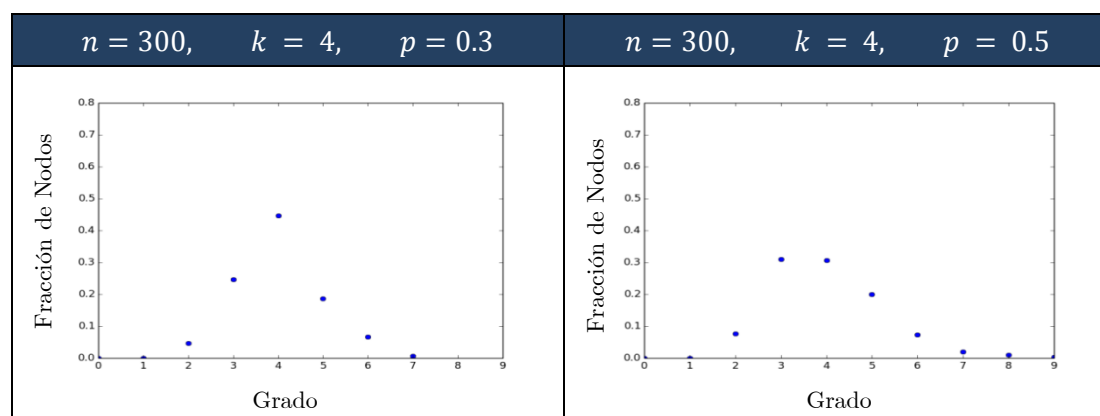


Figura 4.8. Distribución de grado en una red de 300 nodos creada con el modelo Watts-Strogatz, con $k=4$ y $p=0.3$ (derecha) y $k=4$ y $p=0.6$ (izquierda)

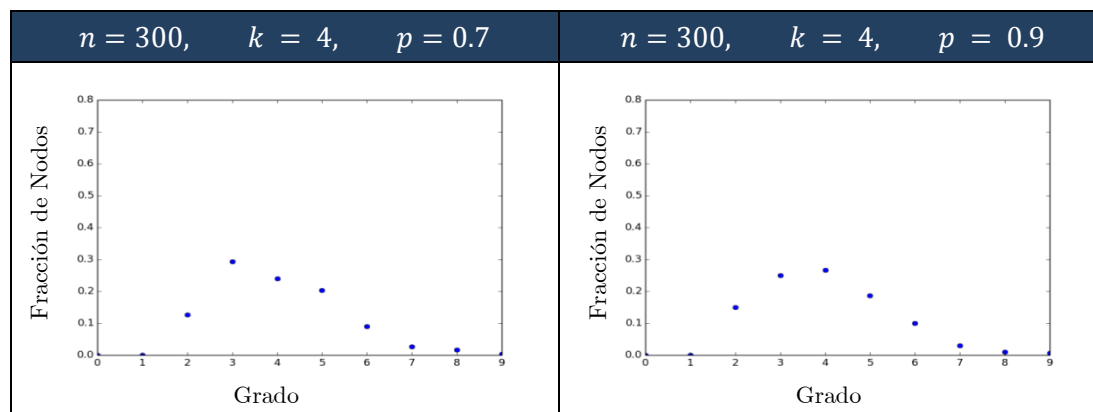


Figura 4.9. Distribución de grado en una red de 300 nodos creada con el modelo Watts-Strogatz, con $k=4$ y $p=0.7$ (derecha) y $k=4$ y $p=0.9$ (izquierda)

Redes Watts-Strogatz con 8 vecinos de media:

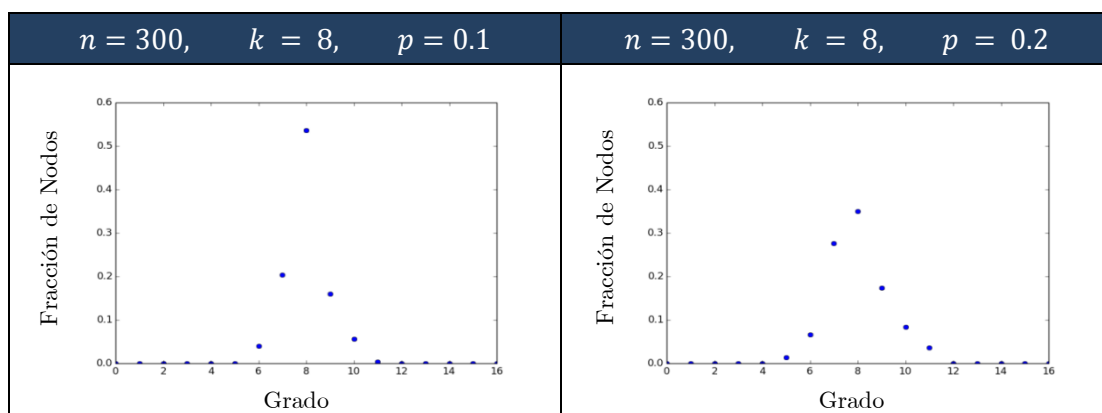


Figura 4.10. Distribución de grado en una red de 300 nodos creada con el modelo Watts-Strogatz, con $k=8$ y $p=0.1$ (derecha) y $k=8$ y $p=0.2$ (izquierda)

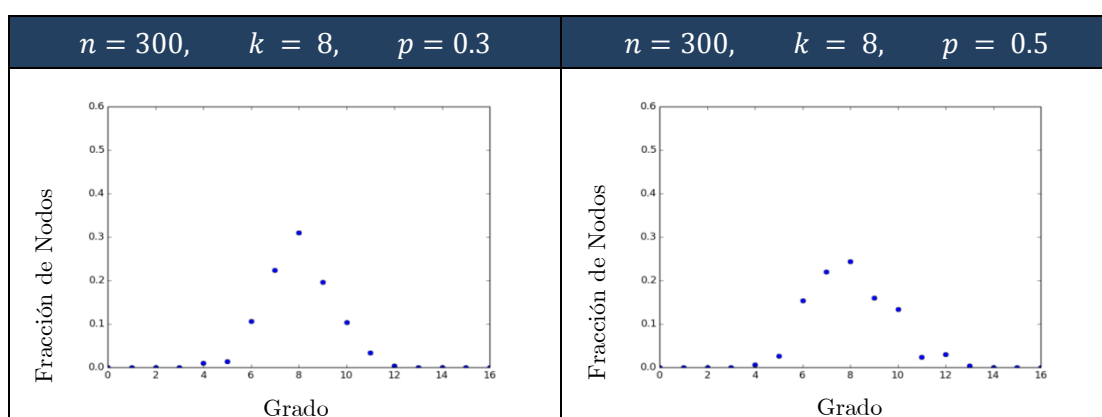


Figura 4.11. Distribución de grado en una red de 300 nodos creada con el modelo Watts-Strogatz, con $k=8$ y $p=0.3$ (derecha) y $k=8$ y $p=0.5$ (izquierda)

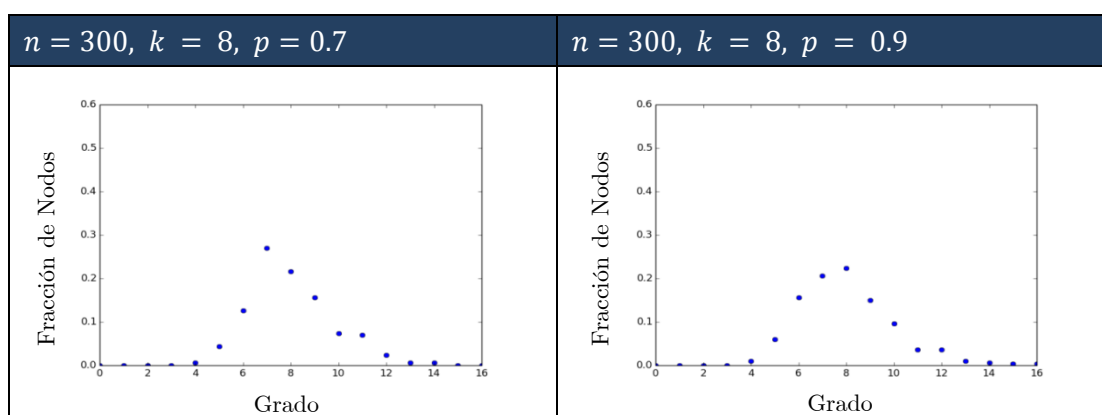


Figura 4.12. Distribución de grado en una red de 300 nodos creada con el modelo Watts-Strogatz, con $k=8$ y $p=0.7$ (derecha) y $k=8$ y $p=0.9$ (izquierda)

4.2 Modelo

En nuestro protocolo para establecer nodos entrelazados el ingrediente más importante que queremos explotar es la posibilidad de establecer canales de comunicación cuánticos utilizando el entrelazamiento (ver sección 2.3.2) y las operaciones de intercambio de entrelazamiento (ver sección 2.3.4) a través de una ruta de acceso física. Este entrelazamiento se puede utilizar para, por ejemplo el intercambio de claves simétricas o para intercambiar otro tipo de información, momento en que se explotará la teleportación (ver sección 2.3.3).

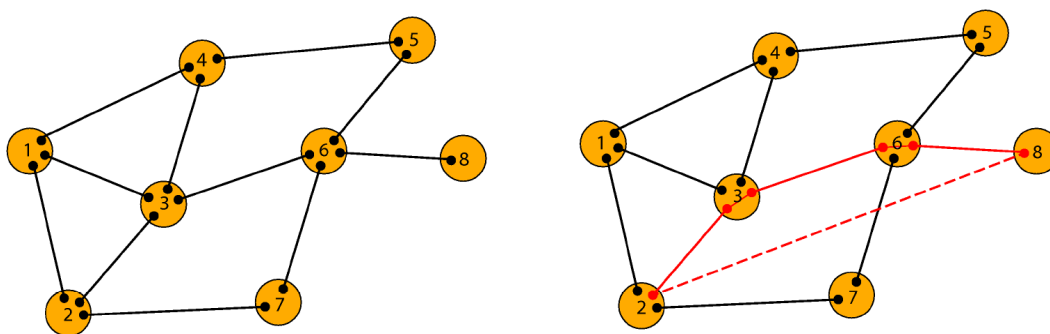


Figura 4.13. a) Red de comunicación física compuesta por $n = 8$ nodos. Cada nodo i está entrelazado con cada uno de sus vecinos k_i b) Canal de comunicación cuántico (línea roja discontinua) que conecta los nodos 2 y 8. Este camino se construye mediante la conexión del camino $\{2, 3, 6, 8\}$, realizando dos operaciones de intercambio de entrelazamiento dentro de los nodos 3 y 6.

Al hacer esta tarea vamos a asumir que:

- Contamos con una red física que consiste en nodos conectados con enlaces a sus vecinos. Un ejemplo de una red física se da en la [figura 4.13 a](#).
- Cada nodo puede estar máximamente entrelazado con todos sus vecinos. De hecho, cada pareja de vecinos puede iniciarse en cada paso de tiempo como máximamente entrelazados (depende del modelo).
- Entenderemos por canal de comunicación cuántico al camino entrelazado que conecta dos nodos mediante las operaciones de intercambio de entrelazamiento (ver [figura 4.13 b](#)).

4.2.1 Modelo 1

A continuación vamos a esbozar la dinámica general de un problema de comunicación cuántica en redes complejas. El tiempo se divide en instantes de tiempo de duración similar, es decir, en pasos de tiempo. En cada paso de tiempo tenemos estas etapas:

- i. Inicialmente cada enlace contiene $q = 1$ pares máximamente entrelazados entre sus nodos.
- ii. Con una probabilidad ρ se crea un nuevo mensaje (o paquete de información) en cada uno de los N nodos de la red (de modo que, en promedio, se crean ρN nuevos mensajes en cada paso de tiempo). Cada mensaje apunta a un nodo de destino que se elige al azar entre los restantes ($N - 1$) nodos del sistema. Los nuevos mensajes se encuentran al final de la cola de cada nodo. La cola de un nodo es la lista de los mensajes que se enviarán desde el.
- iii. Cada nodo intenta enviar el primer mensaje que hay en su cola. El orden en el que se comprobaran los nodos y el camino a utilizar para llegar al destino lo determinará el algoritmo de enrutamiento (los posibles algoritmos se describen a continuación, en la sección 4.4). Digamos que el destino está separado por n saltos desde el origen.
- iv. Entonces se debe realizar una operación de intercambio de entrelazamiento en los $n - 1$ nodos intermedios para crear un camino entrelazado de longitud n .
- v. Debido al entrelazamiento conectamos el origen con el destino, creando un canal de comunicación cuántico instantáneo. En este proceso se destruyen todos los enlaces intermedios (ver sección 2.3.4), por lo que estos ya no se podrán utilizar más durante este paso de tiempo ($q = q - 1$ en cada enlace intermedio).
- vi. Podría ser que un nodo con algún paquete para enviar no encontrase disponible un camino hacia el nodo destino debido a que hay enlaces que ya no se pueden usar por lo explicado en la etapa anterior, por lo que en este caso el paquete esperaría en la cola.

- vii. Cuando se ha comprobado todos los nodos y se han establecido todos los canales de comunicación cuánticos, los mensajes “afortunados” son enviados a sus destinos a través de teleportación.
- viii. Cuando un mensaje llega a su destino desaparece de la red.
- ix. Al final de cada paso de tiempo y después de que se hayan enviado todos los paquetes se regenera el entrelazamiento entre los vecinos.

Definido el modelo principal vamos a proponer dos modelos alternativos. Estos incluirán pequeñas modificaciones definiendo un comportamiento alternativo.

4.2.2 Modelo 1.1

En el primer modelo alternativo vamos a introducir un nuevo parámetro; este parámetro es el ratio de regeneración (v) del par máximamente entrelazado entre los nodos vecinos, es decir, la probabilidad de regenerar el entrelazamiento entre vecinos en cada paso de tiempo. En este caso solo se modifica la etapa [ix](#). Dicha etapa ahora será la siguiente:

- ix. Al final de cada paso de tiempo y después de que se hayan enviado todos los paquetes se comprueban todos los enlaces para ver cuales tienen $q = 0$ pares máximamente entrelazados entre sus nodos. En este caso y con una probabilidad de v se reestablecerá el enlace entrelazando sus nodos. En el caso de que debido a $v < 1$ no se reestableciera el enlace, este quedaría en 0 pares máximamente entrelazados.

4.2.3 Modelo 1.2

En el segundo modelo alternativo vamos a introducir otro parámetro además del parámetro v definido en el [modelo 1.1](#); este parámetro (q_{max}) indicará cuantos pares máximamente entrelazados puede haber como máximo entre cada par de nodos vecinos. En este caso se modificaran las etapas [i](#) y [ix](#). Dichas etapas ahora serán las siguientes:

i. Inicialmente cada enlace contiene $q = q_{max}$ pares máximamente entrelazados entre sus nodos.

ix. Al final de cada paso de tiempo y después de que se hayan enviado todos los paquetes se comprueban todos los enlaces para ver cuales tienen $q < q_{max}$ pares máximamente entrelazados entre sus nodos. En este caso y con una probabilidad de v , se reestablecerá el enlace entrelazando sus nodos. En el caso de que debido a $v < 1$ no se reestableciera el enlace, este quedaría en $q = q - 1$ pares máximamente entrelazados hasta llegar a $q = 0$.

4.2.4 Análisis

Nótese que la diferencia significativa con los protocolos clásicos es que la raíz de la congestión no se deberá a la congestión producida en los nodos altamente conectados (debido a los cuellos de botella), sino a la falta de disponibilidad de enlaces (debido al intercambio de entrelazamiento) que obligará a los mensajes a esperar en las colas.

En la [figura 4.14](#) se describe las primeras etapas de un paso de tiempo utilizando un algoritmo de enrutamiento de caminos mínimos con el [modelo 1](#):

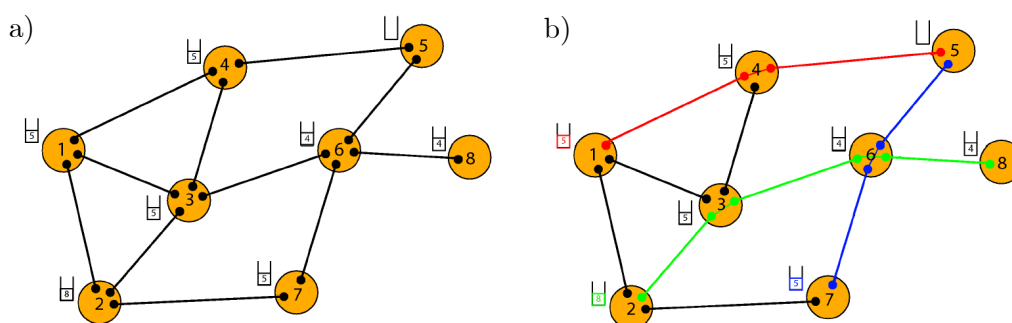


Figura 4.14. **a)** Red física de comunicación cuántica, los enlaces representan pares máximamente entrelazados entre nodos vecinos. En cada nodo hay una cola con paquetes que se tienen que enviar. Cada paquete indica el nodo destino. **b)** Dado un algoritmo de enrutamiento de caminos mínimos se establecen los caminos que se utilizarán para enviar los paquetes. En este caso los nodos se comprueban en orden de su numeración (1, 2, 3, 4, 5, 6, 7, 8, 9). Hay paquetes que no tienen camino debido a que otro nodo ya lo ha ocupado.

En la [figura 4.15](#) se describe las últimas etapas de un paso de tiempo utilizando un algoritmo de enrutamiento de caminos mínimos con el [modelo 1](#):

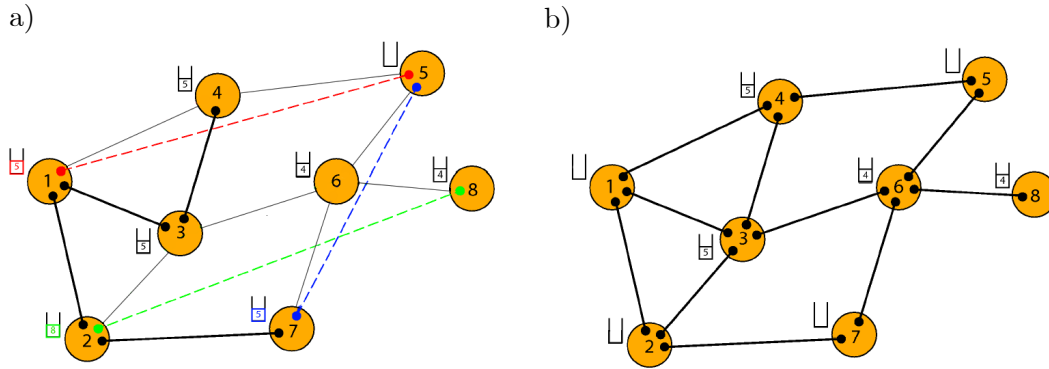


Figura 4.15. **a)** Se realiza el intercambio de entrelazamiento en los nodos intermedios (4, 3 y 6) creando un canal de comunicación cuántico instantáneo entre el nodo origen y el nodo destino. Los enlaces intermedios (pares de fotones máximamente entrelazados entre vecinos) desaparecen debido al intercambio de entrelazamiento, aunque entre ellos sigue habiendo un canal de comunicación físico (internet clásico, fibra óptica, etc.). **b)** Los paquetes ya han sido teletransportados, por lo que se eliminan de la red y se reestablecen los pares máximamente entrelazados entre los nodos vecinos. En este caso la probabilidad de reestablecer el canal es $v = 1$, es decir, se reestablecen todos los enlaces que había antes de hacer el intercambio de entrelazamiento.

4.3 Configuración y Simulación

Descrito el funcionamiento del modelo, ahora vamos a describir el funcionamiento de la simulación y otras características que hay que tener en cuenta:

- Al iniciar la simulación se carga una de las redes definidas en la sección [4.1](#).
- En una simulación se comprueban p probabilidades de generación de paquete distintas.
- En el [modelo 1.1](#) se comprueban k probabilidades de regeneración de links distintas.
- Dada una probabilidad de generación de paquetes p , se realizaran t pasos de tiempo para, después, poder calcular la congestión producida tal y como se ha descrito en la sección [2.2.4](#).

El pseudocódigo general de la simulación es el que sigue:

```

(1) red ← cargar_red(leer(nombre_red))
(2) p ← leer(p)
(3) t ← leer(pasos)
(4) q_max ← leer(q_max)
(5) v_ini ← leer(v_ini)
(6) v_fin ← leer(v_fin)
(7) W ← leer(w)
(8) escala ← (v_fin - v_ini)/(W-1)
(9)
(10) para i=0 hasta p hacer
(11)     pg = (i+1)/p # probabilidad de generar paquetes
(12)     para j=0 hasta k hacer
(13)         v = (j*escala) + v_ini # probabilidad de regenerar un link
(14)         ejecutar_simulación(red,pg,v,q_max, t)
(15)
(16) función ejecutar_simulacion(red,pg,qr,q_max, t)
(17)     para i=0 hasta t hacer
(18)         numero_paquetes_generados[i] = generar_paquetes(red,pg)
(19)         para cada nodo en red.nodos() hacer
(20)             camino ← algoritmo_enrutamiento(red,nodo)
(21)             si hay camino entonces
(22)                 intercambio_entrelazamiento(camino,red)
(23)                 lista_nodos_afortunados.añadir(nodo)
(24)                 enviar_primer_paquete_cola(lista_nodos_afortunados)
(25)                 paquetes_en_cola[i] = paquetes_en_cola()
(26)                 regenerar_links_cuanticos(qr,q_max)
(27)     retorna numero_paquetes_generados, paquetes_en_cola

```

El software que realiza la simulación se ha desarrollado en Python de tal manera que todas las variables se puedan introducir por parámetro antes de ejecutar la simulación (función *leer()* en el pseudocódigo).

Para entender el pseudocódigo hay que tener en cuenta lo siguiente:

- La variable p indica el número de probabilidades de generación de paquetes que queremos comprobar, es decir, si p vale 50, en (11) se generarán dinámicamente 50 probabilidades diferentes desde 0 hasta 1. En este caso generará: 0.02, 0.04, 0.06,..., 1.00 (es decir, de 0 a 1 cada 0.02). Evidentemente la probabilidad 0.00 no se comprueba dado que, lógicamente, si no hay paquetes tampoco habrá congestión.

- La variable q_max indica el número máximo de pares máximamente entrelazados que habrá en el inicio de la simulación entre los nodos vecinos tal que $q_max \in \mathbb{N} - \{0\}$.
- La variable v_ini indica la probabilidad inicial de regenerar un link que se comprobará.
- La variable v_fin indica la probabilidad final de regenerar un link que se comprobará.
- La variable W indica cuantas probabilidades de regenerar un link se generarán entre v_ini y v_fin . Es decir, si v_ini vale 0.5, v_fin vale 1 y W vale 3, en (13) se generaran dinámicamente las probabilidades ($v=0.5$, $v=0.75$, $v=1$).

4.3.1 Paralelización

Debido a la gran carga computacional que representa realizar la simulación en las redes que hemos definido, se ha hecho un análisis del código secuencial para poder paralelizar la simulación de tal manera que podamos aprovechar todas las CPU's de la máquina que realice tal tarea, reduciendo los tiempos de ejecución.

El análisis de paralización consiste en determinar la existencia de dependencia de datos entre las distintas iteraciones de los bucles que hay en el código. El análisis se realiza desde el bucle más interno, hasta el bucle más externo:

- i) Como podemos ver en el pseudocódigo, el bucle (19) no se puede paralelizar dado que hay una dependencia de datos entre los distintos nodos a la hora de seleccionar un camino, ya que, como se ha explicado, cuando un nodo utiliza un camino, nadie más en ese paso de tiempo puede utilizar-lo.
- ii) Si ahora nos fijamos en el bucle (17), este tampoco puede ser paralelizado debido a que hay una dependencia de datos en (25), dado que las colas de los nodos van evolucionando en cada paso de tiempo.

- iii) Si analizamos los dos bucles principales (10) (12), podemos ver que no existe ninguna dependencia de datos entre las distintas ejecuciones de:

(14) `ejecutar_simulación(red,pg,qr,q_max,pasos)`

por lo tanto se paralelizará la línea (14) de tal forma que cada `ejecutar_simulación()` se ejecute en una CPU distinta.

Para llevar a cabo dicha tarea de paralelización, las ejecuciones de todas las simulaciones realizadas en esta tesis se han realizado en un clúster de 32 nodos (o CPU's). Al final de todas las ejecuciones de `ejecutar_simulación()` la simulación recopila todos los datos para combinarlos y guardarlos debidamente en formato "json" para el posterior análisis.

4.4 Algoritmos de enrutamiento

A continuación se propondrán varios algoritmos de enrutamiento para poder estudiar las transiciones de congestión en función de los parámetros del sistema y para poder comparar la eficacia de las diferentes estrategias de enrutamiento en función de la topología de la red subyacente.

4.4.1 Algoritmos deterministas o estáticos

Estos algoritmos no tienen en cuenta la red subyacente y tampoco evolucionan con el tiempo (ver sección 2.2.2.1).

4.4.1.1 Algoritmo A1

El primer algoritmo que comprobaremos será el algoritmo de Dijkstra o algoritmo de caminos mínimos. Este algoritmo, como su nombre indica, determina el camino más corto entre un nodo origen y un nodo destino. Los pasos que se seguirán serán los siguientes:

- i. Se comprobará si hay paquetes en la cola de los nodos de forma aleatoria hasta recorrer todos los nodos en cada paso de tiempo.

- ii. Dado un paquete, con el algoritmo de Dijkstra se comprobará si hay algún camino mínimo hasta el nodo destino.
- iii. Si hay camino disponible, se procede a enviar el paquete.
- iv. Si el camino está ocupado, el paquete esperará en la cola. Como se ha descrito más arriba, cuando un nodo encuentra un camino para enviar un paquete, este camino ya no podrá ser utilizado más por ningún otro nodo dentro de un mismo paso de tiempo.

4.4.1.2 Algoritmo A2

El segundo algoritmo que implementaremos será una modificación del A1. En este caso también utilizaremos el algoritmo de caminos mínimos, la diferencia estará en que cuando un enlace este ocupado, este será como si no existiera para el resto de nodos, de tal forma que los caminos mínimos pueden variar dentro de un mismo paso de tiempo. Para llevar a cabo el enrutamiento, en cada paso de tiempo se seguirán los siguientes pasos:

- v. Se comprobará si hay paquetes en la cola de los nodos de forma aleatoria hasta recorrer todos los nodos en cada paso de tiempo.
- vi. Dado un paquete, con el algoritmo de Dijkstra se comprobará si hay algún camino mínimo hasta el nodo destino.
- vii. Si hay camino disponible, se procede a enviar el paquete.

Igual que en el algoritmo A1, cuando un nodo encuentra un camino para enviar un paquete, este camino ya no podrá ser utilizado más durante ese paso de tiempo, por lo tanto, en este caso, se supondrá que ese camino ya no existe para el resto de nodos, haciendo que los caminos mínimos entre nodos puedan variar dependiendo de cuando se comprueben.

Con el siguiente ejemplo se entiende mejor:

- i. Inicialmente algunos nodos tienen un paquete para enviar ([figura 4.16 a](#)).

- ii. Supongamos que ya se han comprobado los nodos (1, 7, 2, 8, 6, 3) de forma aleatoria, por lo que solo nos queda comprobar el nodo 4. (figura 4.16 b).
- iii. Si tenemos en cuenta la red física, el camino más corto entre el nodo 4 y 1 es {4,1}, pero como se ha comentado, una característica que se ha definido con este algoritmo es que cuando un camino se está utilizando es como si ya no existiera para el resto de nodos, por lo tanto el nodo 4 encontrará que el camino más corto hasta el nodo 1 es {4, 3, 1}, pudiendo, así, enviar el paquete.

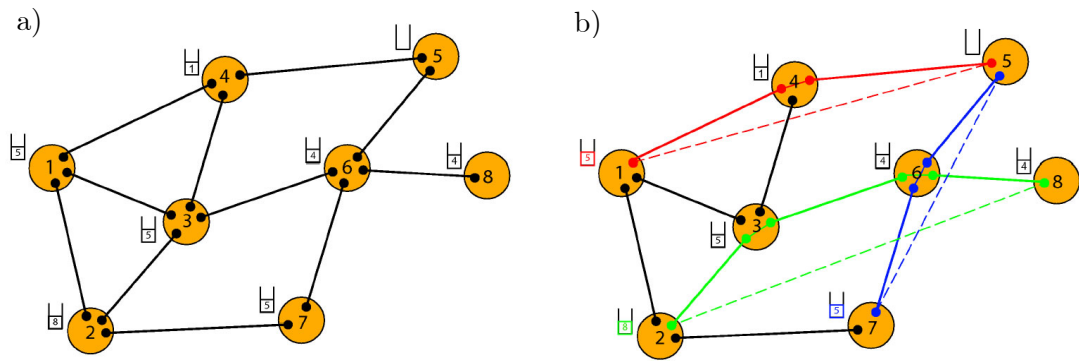


Figura 4.16. a) Red física de comunicación cuántica, los enlaces representan pares máximamente entrelazados entre nodos vecinos. En cada nodo hay una cola con paquetes que se tienen que enviar. Cada paquete indica el nodo destino. b) Ejemplo de enrutamiento con algoritmo de caminos mínimos donde los nodos 1, 2, 7 han encontrado camino para enviar un paquete, creando un enlace cuántico instantáneo entre ellos y el nodo destino.

4.4.2 Algoritmos adaptativos o dinámicos

Estos algoritmos sí que tienen en cuenta la red subyacente y evolucionan con el tiempo, adaptándose a los cambios de la red (ver sección 2.2.2.2).

4.4.2.1 Algoritmo A3

El algoritmo que planteamos ahora será del tipo *adaptativo centralizado*, en el que todos los nodos de la red son iguales excepto un nodo central que es el que toma las

decisiones. En este caso el nodo central decidirá el orden de envío de los paquetes, intentando, en cada paso de tiempo, que se envíen el máximo número de paquetes posible. En este algoritmo también utilizaremos el algoritmo de Dijkstra para encontrar los caminos mínimos.

Para decidir que paquetes se envían primero se tendrán en cuenta una métrica de red, en concreto la *betweenness centrality* (ver sección 2.1.1) de los enlaces de la red. Para llevar a cabo el enrutamiento, en cada paso de tiempo se seguirán los siguientes pasos:

- viii. Un nodo central analiza todos los nodos para ver que paquetes se tienen que enviar (como máximo 1 de cada nodo).
- ix. El nodo central elabora una tabla de enrutamiento. Dicha tabla constará de una clave y un valor. La clave será el camino desde el nodo origen hasta el nodo destino y el valor será la suma de la *betweenness* de todos los enlaces que conformen el camino. En dicha tabla se añadirán todos los caminos mínimos que haya entre el nodo origen hasta el nodo destino con su respectiva suma de *betweenness*.
- x. El nodo central ordena las rutas, poniendo en primer lugar aquellas que tengan una *betweenness* más baja (se ordena de menos a más).
- xi. Una vez ordenada la tabla de enrutamiento se procede a enviar los paquetes en el orden establecido. Eliminando de la red los enlaces utilizados.
- xii. Cuando se ha comprobado toda la tabla se vuelve al paso *viii*. Este proceso se repite tantas veces como se necesario en cada paso de tiempo (algoritmo recursivo) hasta que en el nodo central se genere una tabla de enrutamiento vacía. Evidentemente, cada vez que en el mismo paso de tiempo se genera una nueva tabla de enrutamiento se obvian todos aquellos nodos que ya han enviado su paquete.

El nodo central podrá generar una tabla vacía por dos motivos:

- Todos los nodos han enviado su paquete.

- Los enlaces que quedan en la red no conectan los nodos necesarios para enviar los paquetes de aquellos nodos que no han podido efectuar el envío.

Se ha decidido elaborar un algoritmo recursivo debido a que cuando en el nodo central se genera la tabla de enrutamiento, en ese momento todos los enlaces están disponibles, cosa que cambia conforme se van enviando paquetes, por lo que cuantos más paquetes se envíen menos caminos habrá y, por lo tanto, menos paquetes se podrán enviar a partir de ese momento utilizando esa tabla.

Al hacer el algoritmo recursivo, el nodo central vuelve a comprobar todos los caminos de los paquetes que faltan por enviar. Los caminos mínimos por un mismo paquete van variando cada vez que se genera la tabla debido a que cuando se envía un paquete, el camino utilizado desaparece de la red (igual que en el [algoritmo A2](#)). El algoritmo se entiende mejor con un ejemplo, supongamos la siguiente tabla de enrutamiento ya ordenada, donde hay 4 paquetes para enviar (25, 17, 7, 10):

	Camino	Suma de betweenness
1	{1,5,25}	0.15
2	{5,9,17}	0.19
3	{1,3,25}	0.23
4	{0,2,5,25,7}	0.41
5	{3,5,9,10}	0.49
6	{0,2,4,19,7}	0.51

Tabla 4.1. Ejemplo de tabla de enrutamiento en el algoritmo A3.

En la [tabla 4.1](#), se procederá a enviar el primer paquete (25) que va del nodo 1 al nodo 25, este se enviara sin problema dado que es el primero. El segundo paquete (5) también se envía sin problema ya que ningún enlace contenido en el camino ha sido utilizado. Al comprobar la tercera entrada de la tabla vemos que el paquete 25 ya se ha enviado, por lo tanto se salta. En la cuarta entrada hay un camino para enviar el paquete 7, si nos fijamos, el enlace {5,25} ya se ha utilizado por la primera entrada por lo tanto este camino no se puede utilizar para enviar el paquete 7. En

la quinta entrada pasa lo mismo, el enlace $\{5,9\}$ ya se ha utilizado por lo tanto tampoco se puede enviar el paquete 10. Finalmente, en la sexta entrada, vemos que el paquete 7 no se ha enviado porque el camino de la entrada 4 ya no existía, por lo tanto, al estar disponibles todos los enlaces contenidos en el camino $\{0,2,4,19,7\}$ se procede a enviar el paquete 7.

Es en este momento cuando recursivamente se vuelve a iniciar el proceso; Solo nos falta por enviar, si es posible, el paquete 10. En este caso el nodo central encuentra otro camino mínimo desde el nodo 3 hasta el nodo 10 (en este caso la longitud del camino mínimo es de 5, en vez de 4, ya que el camino de 4 ha desaparecido de la red), por lo tanto crea la siguiente tabla:

	Camino	Suma de betweenness
1	$\{3,8,14,23,10\}$	0.65

Tabla 4.2. Ejemplo de tabla de enrutamiento en el algoritmo A3.

En este caso ningún enlace contenido en el camino ha sido utilizado, por lo tanto se puede enviar el paquete 10. Como ya se ha recorrido toda la tabla, el nodo central vuelve a iniciar todo el proceso creando de nuevo la tabla, al no quedar paquetes para enviar, la tabla estará vacía y se pasará al siguiente paso donde todos los enlaces volverán a estar disponibles (dependiendo del modelo, claro)

Capítulo 5

Resultados

En este capítulo vamos a mostrar los resultados obtenidos de las distintas simulaciones realizadas a las redes definidas en 4.1. Además vamos a comparar, sobre cada modelo definido en 4.2, cómo evoluciona la congestión en función de los parámetros de la red, centrándonos en encontrar el punto crítico P_c , es decir, el punto en el que la red pasa de estar en un estado de flujo libre a un estado de congestión, y viendo en qué medida se van consumiendo los enlaces de la red. Finalmente se compararan los algoritmos de enrutamiento definidos en 4.4, para ver cuáles son más eficientes y porqué.

En todos los experimentos se ha definido $t = 1000$. Para calcular la congestión se ha definido una ventana temporal sobre los resultados de 200 pasos de duración, desde el paso 800 hasta el paso número 1000. Otra cosa a tener en cuenta es que se ha definido un umbral de 0.005 en la congestión, es decir, a partir de este valor se considerará que ya hay congestión en la red.

5.1 Congestión en el Modelo 1

Iniciaremos el análisis sobre el modelo estándar definido en 4.2.1. En este caso el único parámetro utilizado en todas las simulaciones es t , que indica el número de pasos que se producirán.

5.1.1 Redes Erdős–Rényi

5.1.1.1 Algoritmo A1

En la [figura 5.1 izqda.](#) podemos comparar cómo evoluciona la congestión en función de los parámetros de la red sobre algoritmo A1. En este caso se produce congestión en todas las redes. En la [figura 5.1 drcha.](#) se puede observar cómo las redes consumen los enlaces disponibles conforme evoluciona ρ , hasta llegar al ρ_c , donde el número de enlaces utilizados se queda fijo a un punto dependiendo de la red.

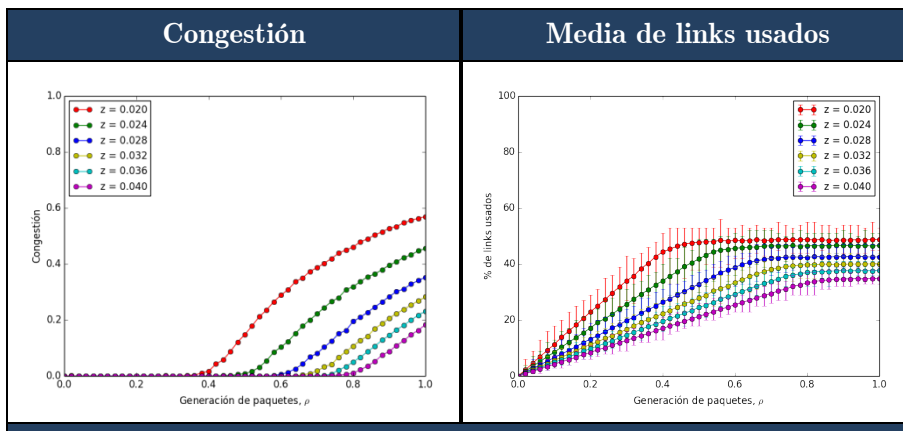


Figura 5.1. (Evolución de la congestión (**izquierda**) y media de enlaces utilizados (%) (**derecha**) en las redes Erdős–Rényi sobre el algoritmo A1 con diferentes valores de z .

En la [tabla 5.1.](#) se muestran los puntos críticos de la distintas redes. Podemos ver como la diferencia ($\rho_c(z) - \rho_c(z - 1)$) entre los ρ_c va decreciendo en función del parámetro z , describiendo una ligera curva.

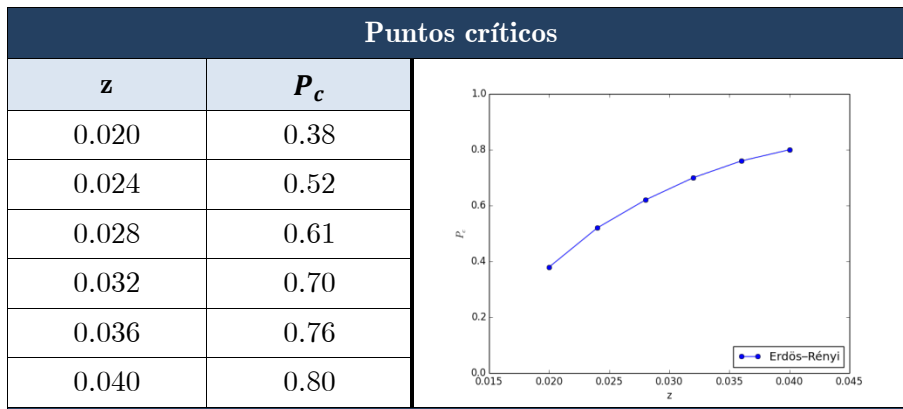


Tabla 5.1 Gráfico (**derecha**) y tabla de puntos críticos (**izquierda**) de las redes Erdős–Rényi sobre el algoritmo A1 con diferentes valores de m .

5.1.1.2 Algoritmo A2

En la [figura 5.2 izqda.](#) podemos comparar cómo evoluciona la congestión en función de los parámetros de la red sobre algoritmo A2. En este caso solo se produce congestión en las redes con probabilidad de generar un enlace entre dos nodos $z < 0.032$. En la [figura 5.2 drcha.](#) se puede ver cómo las redes consumen los enlaces disponibles conforme evoluciona ρ , hasta llegar al ρ_c , donde el número de enlaces utilizados se queda fijo a más o menos un 80%.

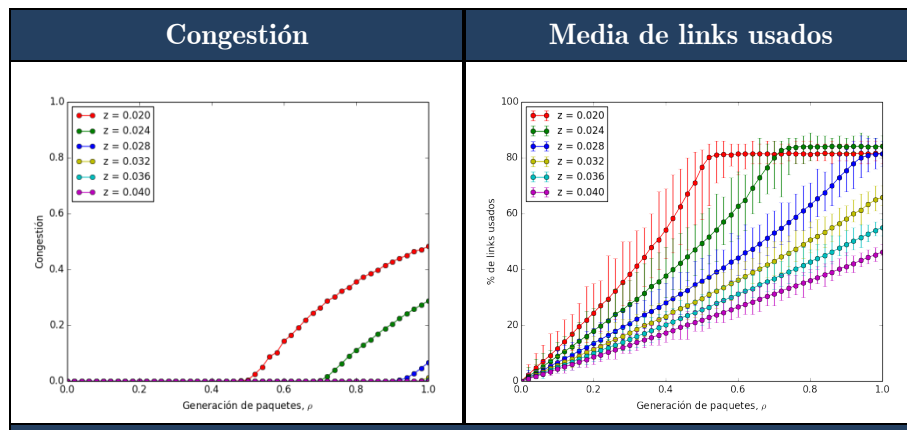


Figura 5.2. Evolución de la congestión (**izquierda**) y media de enlaces utilizados (%) (**derecha**) en las redes Erdős–Rényi con diferentes valores de z sobre el algoritmo A2.

A partir de la [figura 5.2](#) podemos extraer los puntos críticos de las distintas redes, estos se muestran en la [tabla 5.2](#). Podemos ver como el ρ_c crece linealmente en función del parámetro z (~ 0.20), hasta llegar a 1 donde, después, ya no se produce congestión.

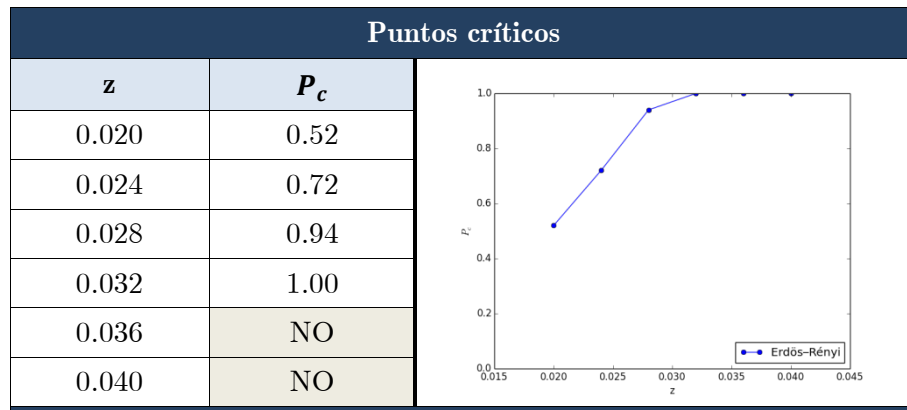


Tabla 5.2. Grafico (**derecha**) y tabla de puntos críticos (**izquierda**) de las redes Erdős–Rényi con diferentes valores de m sobre el algoritmo A2.

5.1.1.3 Algoritmo A3

En la [figura 5.3 izqda.](#) podemos comparar cómo evoluciona la congestión en función de los parámetros de la red sobre algoritmo A3. En este caso también se produce congestión en las redes con probabilidad de generar un enlace entre dos nodos $z < 0.032$. En la [figura 5.3 drcha.](#) Se puede ver cómo las redes se van quedando sin enlaces disponibles conforme evoluciona ρ , hasta llegar al ρ_c , donde, también en este algoritmo, el número de enlaces utilizados se queda fijo a más o menos un 80%.

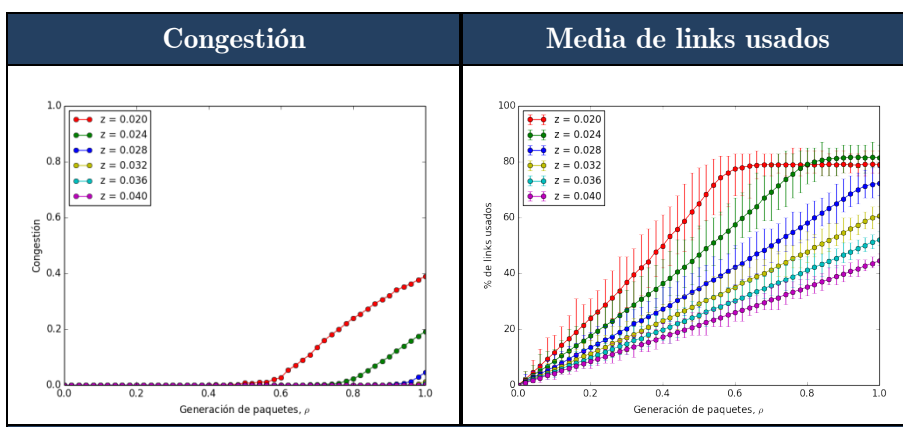


Figura 5.3. Evolución de la congestión (**izquierda**) y media de enlaces utilizados (%) (**derecha**) en las redes Erdős–Rényi sobre el algoritmo A3 con diferentes valores de z .

En la [tabla 5.3](#) se muestran los puntos críticos de las redes. Podemos ver como el ρ_c crece casi linealmente en función del parámetro z , en este caso tiene tendencia a disminuir el aumento conforme se va aproximando a 1, donde, a partir de $z = 0.36$ ya no hay ρ_c , por lo tanto, para cualquier $z > 0.36$ ya no se produciría congestión.

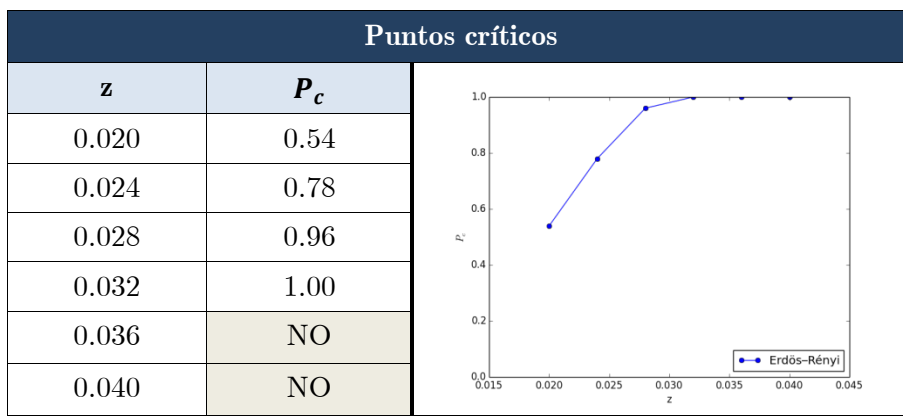


Tabla 5.3. Grafico (**derecha**) y tabla de puntos críticos (**izquierda**) de las redes Erdős–Rényi sobre el algoritmo A3 con diferentes valores de m .

5.1.1.4 Comparativa de algoritmos

En esta sección vamos a comparar los tres algoritmos de enrutamiento que hemos propuesto y estudiado. En la [figura 5.4 izqda.](#) se muestra la comparativa de la congestión en la red Erdős-Rényi con $z = 0.02$. Esta figura muestra como efectivamente, en este tipo de redes la congestión en $\rho = 1$ ha disminuido, aproximadamente, un 0.09 entre el algoritmo A1 y el algoritmo A2 y también un 0.09 entre el algoritmo A2 y el algoritmo A3. Para valores de $\rho < 1$ la diferencia es un poco más grande (~ 0.15).

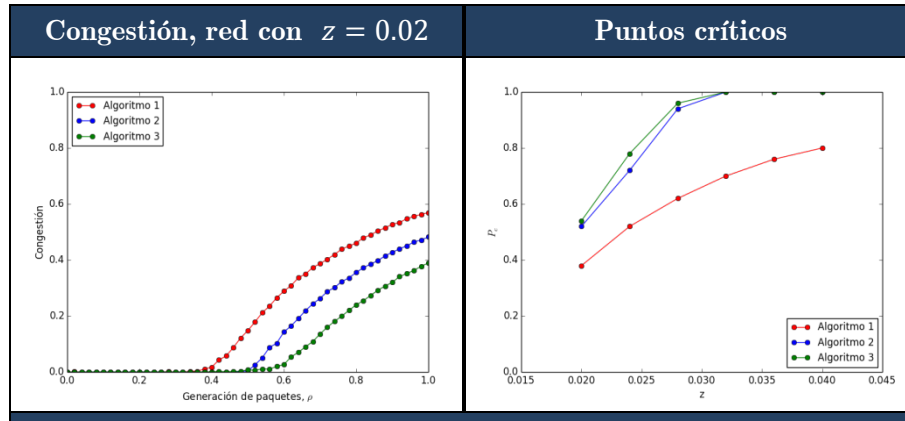


Figura 5.4. (**izquierda**) Comparativa de los distintos algoritmos de enrutamiento en la evolución de la congestión en la red Erdős-Rényi con $z = 0.02$ (**derecha**) Comparativa de los distintos algoritmos de enrutamiento en la evolución de los puntos críticos en todas las redes Erdős-Rényi.

Respecto a los puntos críticos, la [figura 5.4 drcha.](#) muestra como en el algoritmo A1 la diferencia entre los puntos críticos ($\rho_c(z) - \rho_c(z - 1)$) va decreciendo en función del parámetro z describiendo una ligera curva, mientras que en el algoritmo A2 y algoritmo A3 la diferencia es prácticamente la misma, creciendo linealmente hasta llegar a $\rho_c = 1$. La diferencia de ρ_c en las mismas redes entre el algoritmo A1 y los otros dos es bastante grande (de ~ 0.15 a ~ 0.30), pero entre el algoritmo A2 y el algoritmo A3 es muy pequeña (~ 0.02).

Con estos datos podemos afirmar que el algoritmo A3 es el más eficiente en cuanto a evitar la congestión. En términos de punto crítico el algoritmo A3 está igualado con el algoritmo A2, siendo el A1 el menos eficiente tanto en congestión como en ρ_c .

5.1.2 Redes Barabási–Albert

5.1.2.1 Algoritmo A1

En la [figura 5.5 izqda.](#) podemos comparar cómo evoluciona la congestión en función de los parámetros de la red (m) sobre algoritmo A1. En este caso también se produce congestión en todas las redes. En cuanto al consumo de links, vemos como la red con $m = 1$, solamente consumiendo un 5% de los links ya genera congestión, esto se debe a que en esta red la mayoría de nodos tienen pocos vecinos, por lo tanto, enviar un paquete consumirá links con alto betweenness desconectando gran parte de la red. Para el resto de redes permanecen sin consumir entre el 55% y el 65% de los enlaces, según la red.

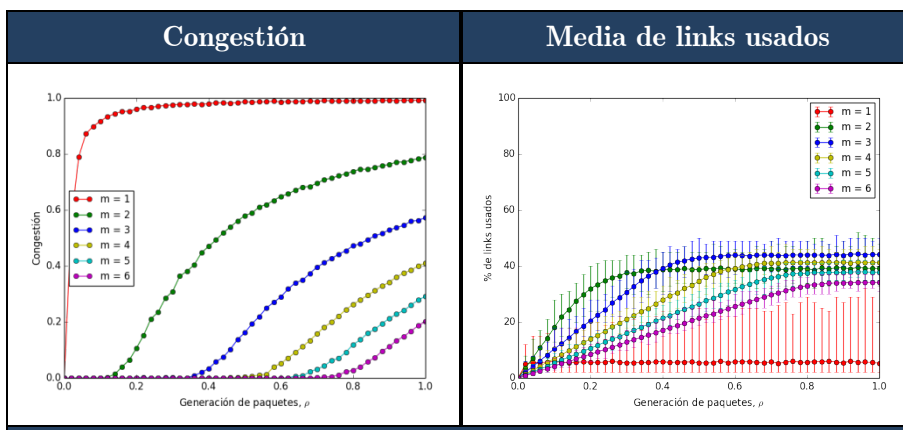


Figura 5.5. Evolución de la congestión (**izquierda**) y media de enlaces utilizados (%) (**derecha**) en las redes Barabási–Albert con diferentes valores de m sobre el algoritmo A1.

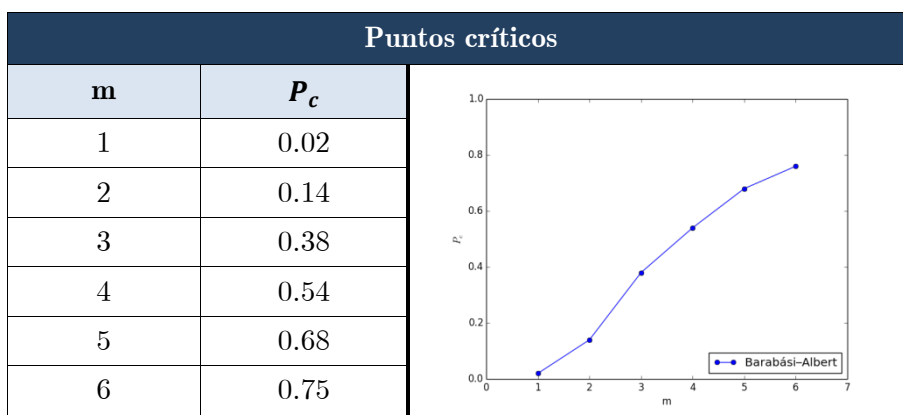


Tabla 5.4 Gráfico (**derecha**) y tabla de puntos críticos (**izquierda**) de las redes Barabási–Albert sobre el algoritmo A1 con diferentes valores de m .

5.1.2.2 Algoritmo A2

En la [figura 5.6 izquierda](#). podemos comparar cómo evoluciona la congestión en función de los parámetros de la red (m) sobre Algoritmo A2. En este caso también se produce congestión en las redes con $m < 6$. Vemos que la congestión se reduce un 30% en cada incremento de m . En cuanto al consumo de links, menos en la red con $m = 1$, el resto de redes consumen entre el 75% y 85% de los enlaces antes de generar congestión.

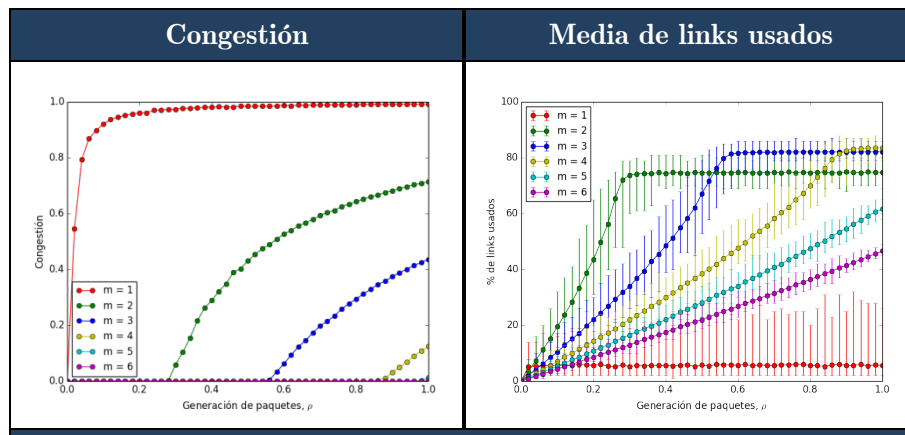


Figura 5.6. Evolución de la congestión (**izquierda**) y media de enlaces utilizados (%) (**derecha**) en las redes Barabási-Albert sobre el algoritmo A2 con diferentes valores de m .

Extrayendo los ρ_c de la [figura 5.6](#), vemos como el aumento del ρ_c crece linealmente en función de cada incremento de m , este incremento es de ~ 0.30 hasta llegar a 1. Por lo tanto para cualquier $m > 5$ no se generará congestión

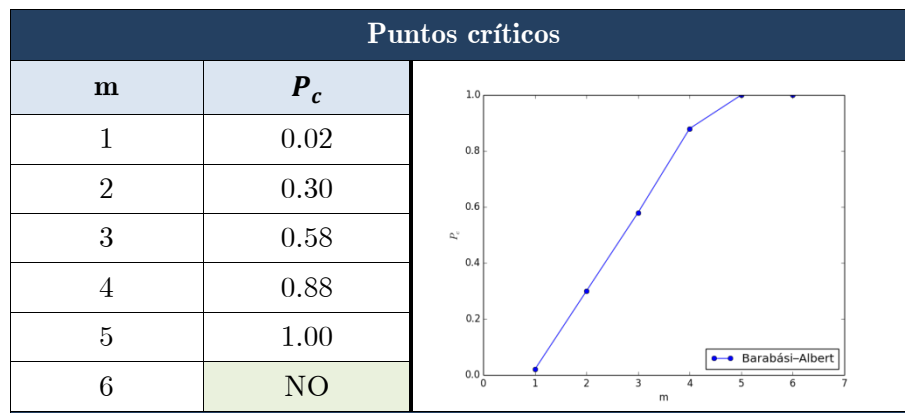


Tabla 5.5 Grafico (**derecha**) y tabla de puntos críticos (**izquierda**) de las redes Barabási-Albert sobre el algoritmo A3 con diferentes valores de m .

5.1.2.3 Algoritmo A3

En la [figura 5.7 izqda.](#) podemos comparar cómo evoluciona la congestión en función de los parámetros de la red sobre algoritmo A3. En este caso también se produce congestión en las redes con $m < 6$. Con este algoritmo, se reduce la congestión en un 35% cuando pasamos de $m = 1$ a $m = 2$, un 30% de $m = 2$ a $m = 3$ y un 25% de $m = 3$ a $m = 4$. El consume de enlaces es igual que en el caso anterior, todas las redes consumen entre el 70% y el 80% de enlaces.

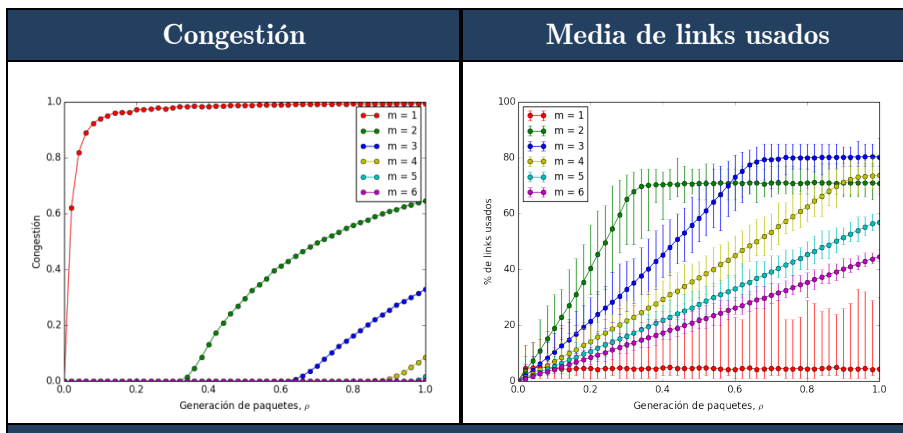


Figura 5.7. Evolución de la congestión (**izquierda**) y media de enlaces utilizados (%) (**derecha**) en las redes Barabási–Albert con diferentes valores de m sobre el algoritmo A3.

Extrayendo los ρ_c de la [figura 5.7](#), vemos como el aumento del ρ_c en función de cada incremento de m es lineal (~ 0.30) hasta llegar a 1.

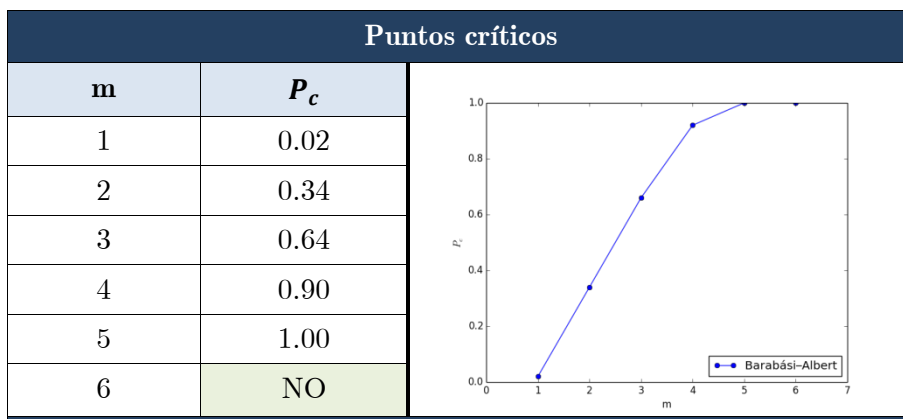


Tabla 5.6 Gráfico (**derecha**) y tabla de puntos críticos (**izquierda**) de las redes Barabási–Albert sobre el algoritmo A3 con diferentes valores de m .

5.1.2.4 Comparativa de algoritmos

En esta sección vamos a comparar los tres algoritmos de enrutamiento que hemos propuesto y estudiado. En la [figura 5.8 izqda.](#) se muestra la comparativa de la congestión en la red Barabási–Albert con $m = 2$. Esta figura muestra como efectivamente, en este tipo de redes la congestión ha disminuido. Para valores de $\rho \rightarrow 1$ esta disminución es más o menos igual que en las redes Erdős–Rényi, aproximadamente un 0.09 entre el algoritmo A1 y el A2 y entre el algoritmo A2 y el A3, pero para valores de $\rho < 1$, la diferencia es bastante más grande (0.2).

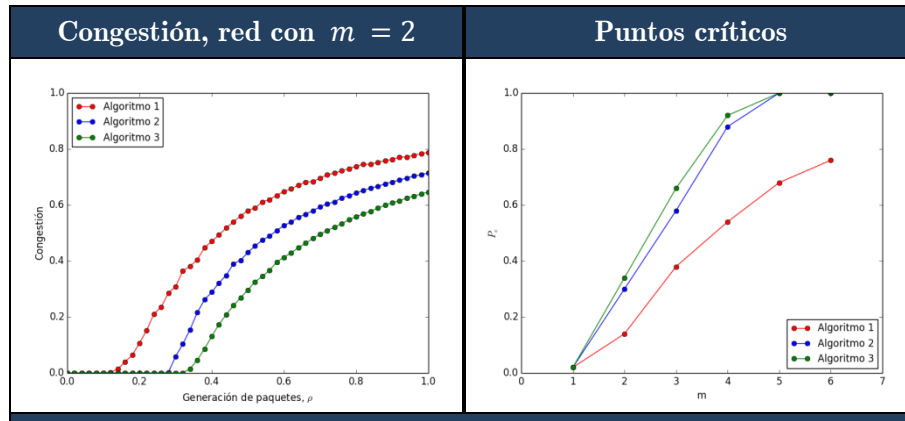


Figura 5.8. (izquierda) Comparativa de los distintos algoritmos de enrutamiento en la evolución de la congestión sobre la red Barabási–Albert con $m = 2$ (*derecha*) Comparativa de los distintos algoritmos de enrutamiento en la evolución de los puntos críticos sobre todas las redes Barabási–Albert.

Respecto a los puntos críticos, la [figura 5.8 drcha.](#) muestra como en el algoritmo A1 la diferencia entre los puntos críticos, ρ_c , $(\rho_c(z) - \rho_c(z - 1))$ va decreciendo en función del parámetro z describiendo una ligera curvatura, mientras que en el algoritmo A2 y algoritmo A3 la diferencia es prácticamente la misma, creciendo linealmente hasta llegar a $\rho_c = 1$. Podemos ver como en la red con $m = 1$, el algoritmo de enrutamiento no afecta a la congestión, produciendo el mismo ρ_c para todos los algoritmos. Para el resto de redes, la diferencia de ρ_c entre el algoritmo A1 y los otros dos es bastante grande (de ~ 0.15 a ~ 0.30), pero entre el algoritmo A2 y el A3 es muy pequeña (de ~ 0.02 a ~ 0.10). Con estos datos podemos afirmar que el algoritmo A3 es el más eficiente en cuanto a evitar la congestión y a aumentar el punto crítico.

5.1.3 Redes Watts-Strogatz

5.1.3.1 Algoritmo A1

En la [figura 5.9 izqda.](#) podemos comparar cómo evoluciona la congestión en función de los parámetros de la red (r) sobre Algoritmo A1, con $k = 4$ vecinos de media. En este caso se produce congestión en todas las redes. Vemos como la diferencia de la congestión va decreciendo conforme se va incrementando r , siendo casi igual para todas las redes con $r \geq 0.5$. En cuanto al consumo de enlaces, en la [figura 5.9 drcha.](#) vemos como cuanto más grande sea r , más enlaces se consumen, pero en ningún caso consume más del 50% de los enlaces de la red.

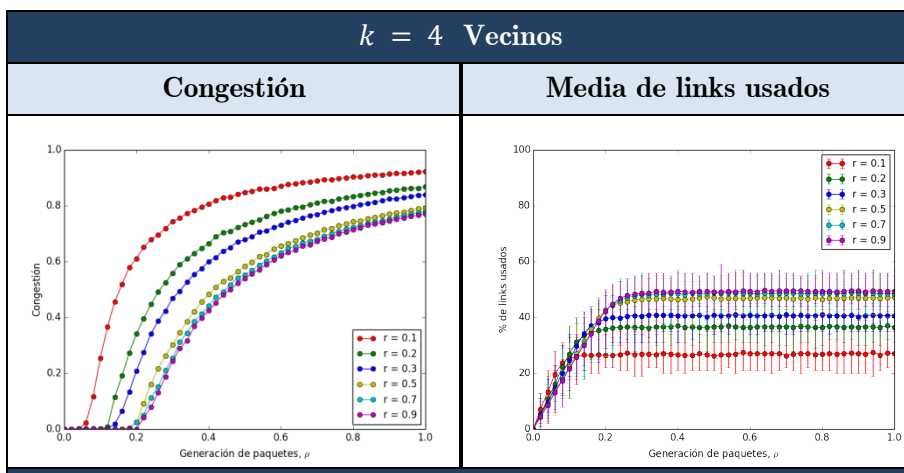


Figura 5.9. Evolución de la congestión (**izquierda**) y media de enlaces utilizados (%) (**derecha**) en las redes Watts-Strogatz con diferentes valores de m sobre el algoritmo A1 y con $k = 4$ vecinos de media.

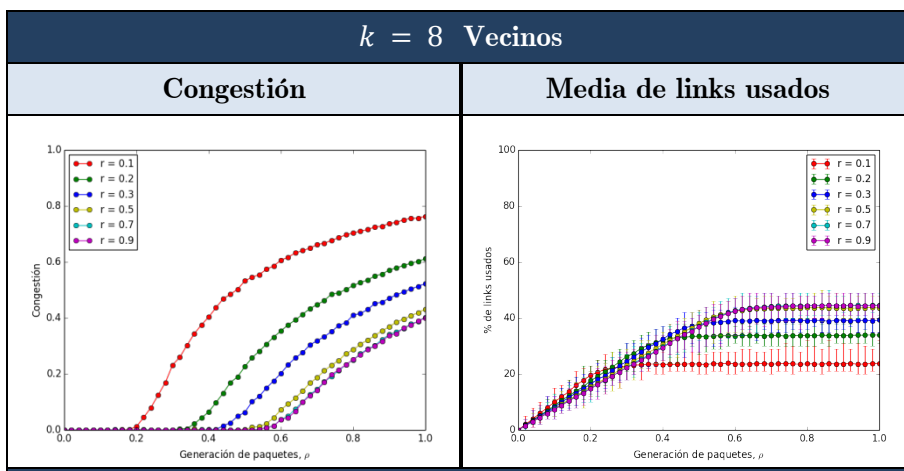


Figura 5.10. Evolución de la congestión (**izquierda**) y media de enlaces utilizados (%) (**derecha**) en las redes Watts-Strogatz con diferentes valores de m sobre el algoritmo A1 y con $k = 8$ vecinos de media.

En la [figura 5.10 izqda.](#) podemos comparar cómo evoluciona la congestión en función de los parámetros de la red (r) sobre algoritmo A1, ahora con $k = 8$ vecinos de media. En este caso vemos que se ha reducido la congestión en todas las redes, igual que en el caso anterior, conforme se va incrementando r , siendo casi igual para todas las redes con $r \geq 0.5$. Por lo que respecta al consumo de enlaces podemos ver como el consumo crece homogéneamente en todas las redes hasta llegar al punto crítico, donde el consumo se mantiene fijo.

Vemos en la [tabla 5.7](#) los puntos críticos extraídos de las [figuras 5.9](#) y [5.10](#). En el caso de $k = 4$, estos apenas varían 0.08, manteniendo unos puntos críticos bastante bajos en cuanto a la probabilidad de generación de paquetes ρ . En el caso de $k = 8$ los puntos críticos están bastante más equidistantes, de todas formas, igual que antes, cuando $r > 0.5$, el ρ_c ya se mantiene estable a 0.58. Hay que tener en cuenta que las redes con $k = 8$ tienen el doble de enlaces que las redes con $k = 4$, con el mismo número de nodos, por lo tanto, es bastante obvio que se haya reducido la congestión y que hayan aumentado los puntos críticos.

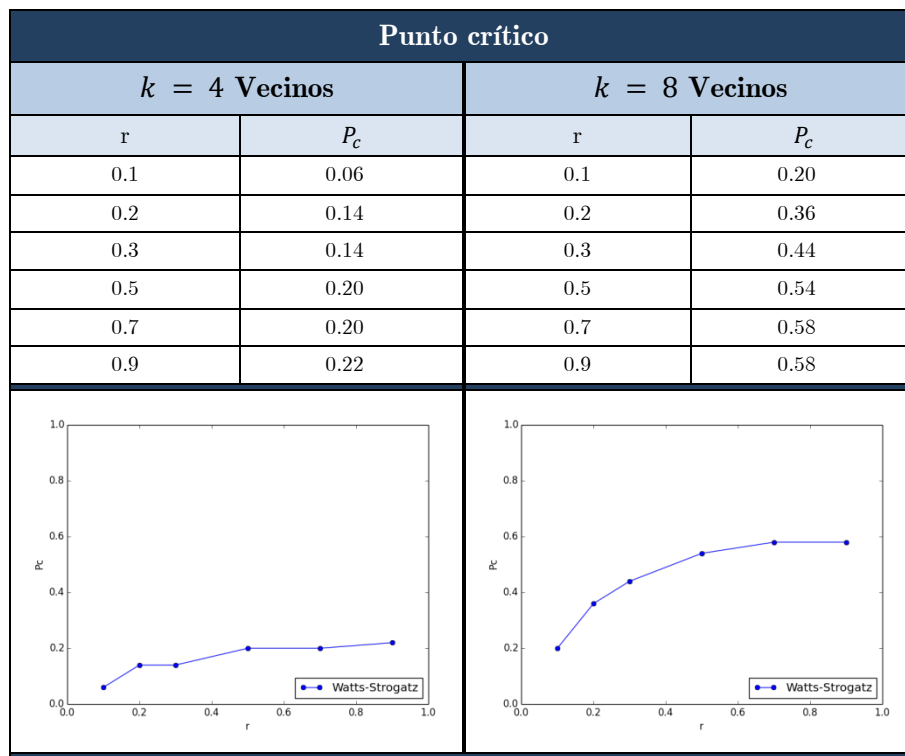


Tabla 5.7 Evolución del punto crítico en las redes Watts-Strogatz sobre el algoritmo A1 con diferentes valores de r en función del número de vecinos k .

5.1.3.2 Algoritmo A2

En la [figura 5.11 izqda.](#) podemos comparar cómo evoluciona la congestión en función de los parámetros de la red (r) sobre algoritmo A2, con $k = 4$ vecinos de media. En este caso se produce congestión en todas las redes. Vemos como la diferencia de la congestión va decreciendo conforme se va incrementando r , siendo casi igual para todas las redes con $r \geq 0.5$. En cuanto al consumo de links, vemos que como cuanto más grande sea r , más links consume. Ahora el mínimo de links utilizados es del 50% para $r = 0.1$ y del 75% para cualquier $r \geq 0.5$, siendo para $r = 0.2$ y $r = 0.3$ aproximadamente un 60%

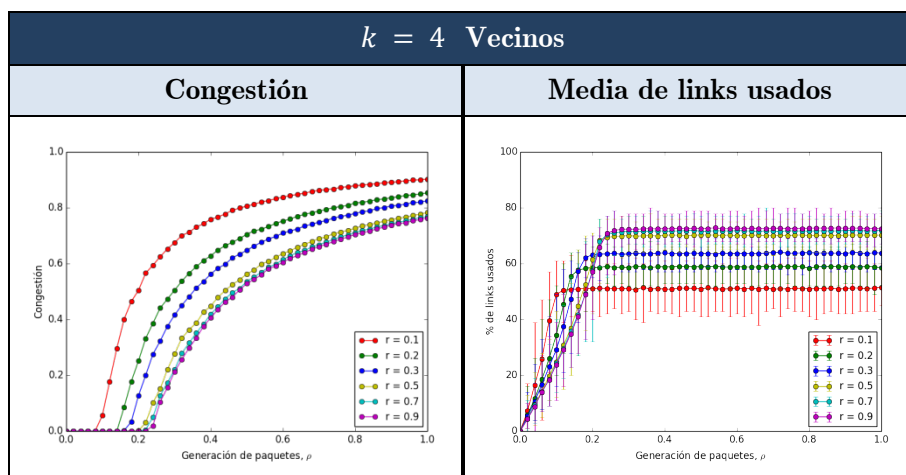


Figura 5.11. Evolución de la congestión (**izquierda**) y media de enlaces utilizados (%) (**derecha**) en las redes Watts-Strogatz con diferentes valores de m sobre el algoritmo A2 y con $k = 4$ vecinos de media.

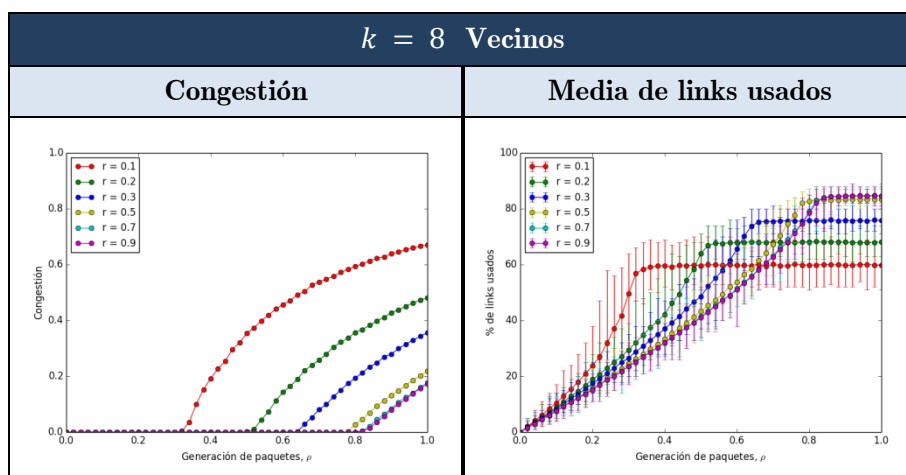


Figura 5.12. Evolución de la congestión (**izquierda**) y media de links sin usar (%) (**derecha**) en las redes Watts-Strogatz con diferentes valores de m sobre el algoritmo A2 y con $k = 4$ vecinos de media.

En la [figura 5.12 izqda.](#) podemos comparar cómo evoluciona la congestión en función de los parámetros de la red (r) sobre algoritmo A2, ahora con $k = 8$ vecinos de media. En este caso se ha reducido la congestión significativamente, siendo esta muy baja para todas las redes con $r \geq 0.5$. Por lo que respecta al consumo de enlaces ([figura 5.12 drcha.](#)), este crece bastante rápido en la red con $r = 0.1$ y más suavemente cuando $r \rightarrow 1$, esto se debe a que la red con $r = 0.1$ tiene pocos enlaces ‘puente’ por lo que tendrá que consumir muchísimos más enlaces para enviar un paquete, en cambio, cuanto más grande sea r , más enlaces ‘puente’ habrá, y, por lo tanto, más corto será el camino. En este caso el límite de consumo de enlaces se mantiene al 85% para todas las redes con $r \geq 0.5$. Para el resto de redes, el consumo va del 60% al 80% según la red.

En cuanto a los puntos críticos ([tabla 5.8](#)), vemos que cuando $k = 4$, estos apenas varían 0.04, manteniendo unos puntos críticos bastante bajos en cuanto a ρ . En el caso de $k = 8$ los puntos críticos están bastante más equidistantes. A pesar de esto, en las redes con $r > 0.5$, el ρ_c se mantiene estable a 0.84.

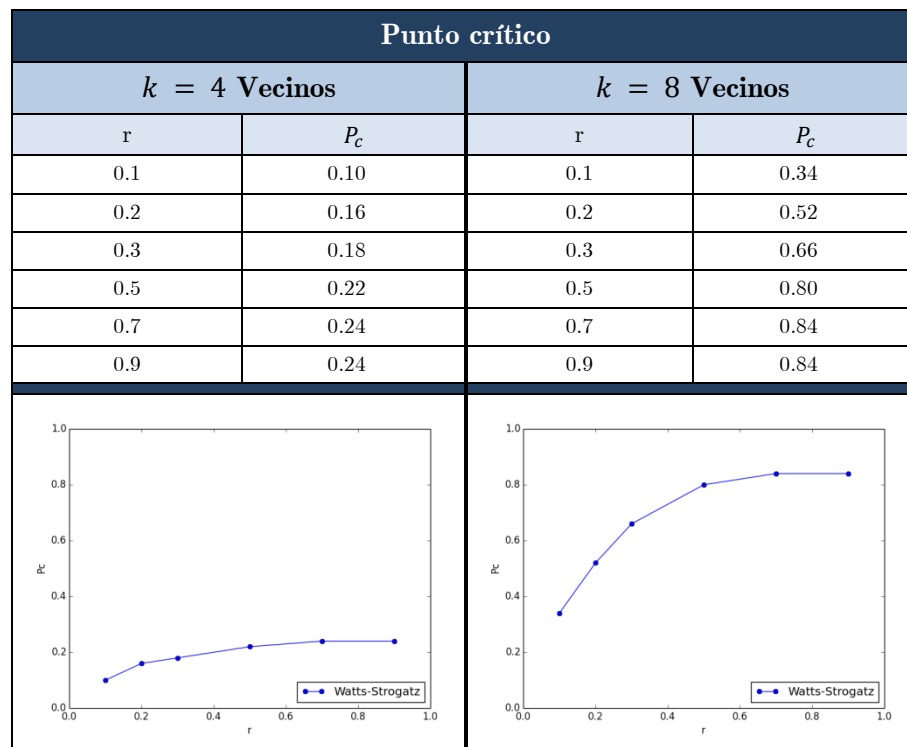


Tabla 5.8 Evolución del punto crítico en las redes Watts-Strogatz sobre el algoritmo A2 con diferentes valores de r en función del número de vecinos k .

5.1.3.3 Algoritmo A3

En la [figura 5.13 izqda.](#) podemos comparar cómo evoluciona la congestión en función de los parámetros de la red (r) sobre Algoritmo A3, con $k = 4$ vecinos de media. En este caso se produce congestión, también, en todas las redes. Vemos como la diferencia de la congestión va decreciendo conforme se va incrementando r , siendo casi igual para todas las redes con $r \geq 0.5$. En cuanto al consumo de links, vemos en la [figura 5.13 drcha.](#) que cuanto más grande sea r , más links consume antes de entrar en fase de congestión. Ahora el mínimo de links utilizados es de 45% para $r = 0.1$, 55% para $r = 0.2$ y de aproximadamente el 70% para $r \geq 0.5$.

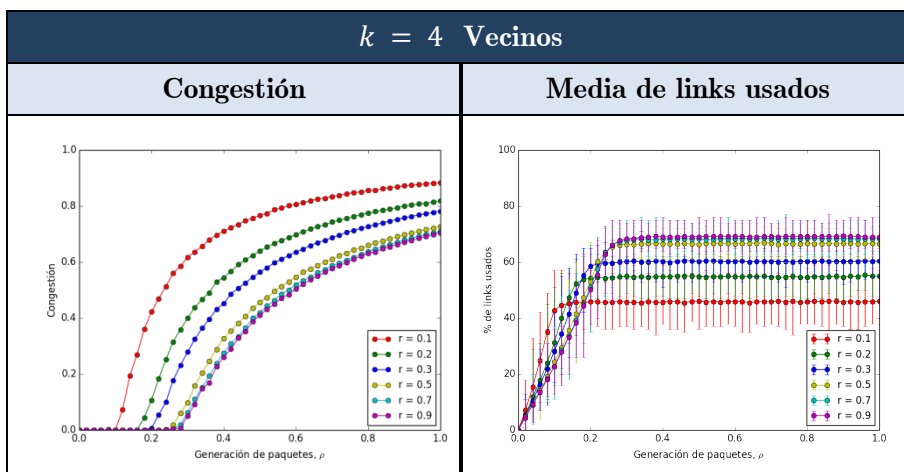


Figura 5.13. Evolución de la congestión (**izquierda**) y media de enlaces utilizados (%) (**derecha**) en las redes Watts-Strogatz con diferentes valores de m sobre el algoritmo A3 y con $k = 4$ vecinos de media.

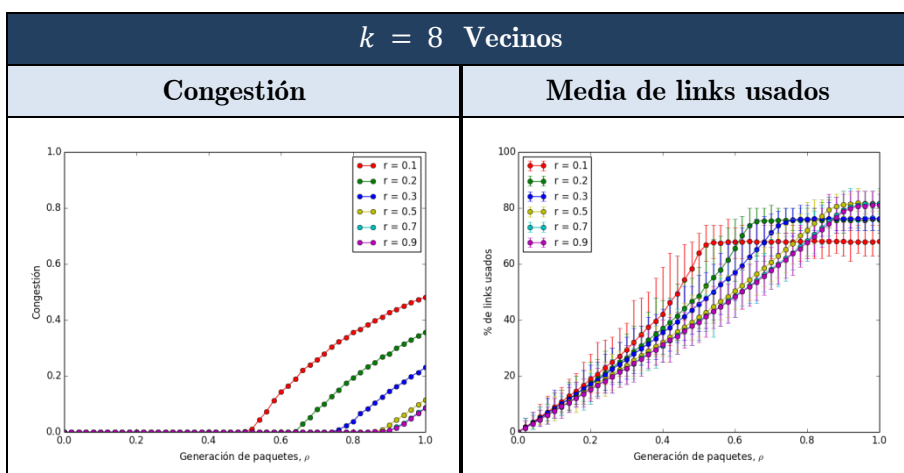


Figura 5.14. Evolución de la congestión (**izquierda**) y media de links sin usar (%) (**derecha**) en las redes Watts-Strogatz con diferentes valores de m sobre el algoritmo A3 y con $k = 4$ vecinos de media.

En la [figura 5.14 izqda.](#) podemos comparar cómo evoluciona la congestión en función de los parámetros de la red (r) sobre Algoritmo A3, ahora con $k = 8$ vecinos de media. En este caso se ha reducido la congestión significativamente, siendo esta muy baja para todas las redes con $r \geq 0.5$. Por lo que respecta a el consumo de enlaces ([figura 5.14 drcha.](#)), la red con $r = 0.1$ tiene pocos enlaces ‘puente’ por lo que tendrá que consumir muchos más enlaces para enviar un paquete, en cambio, cuanto más grande sea r , más enlaces ‘puente’ habrá, y, por lo tanto, más corto será el camino. En este caso el límite de consumo de enlaces se mantiene al 85% para todas las redes con $r \geq 0.5$. Para las redes con $r = 0.1$ y $r = 0.2$ el límite es del 67 %, mientras que para la red con $r = 0.3$ es del 75%.

En la [tabla 5.9](#) se muestran los puntos críticos, vemos que cuando $k = 4$, estos apenas varían, aproximadamente, 0.05, creciendo muy ligeramente en función de r y manteniendo unos puntos críticos bastante bajos. En el caso de $k = 8$ los puntos críticos están bastante más equidistantes entre las distintas redes. A pesar de esto, en las redes con $r > 0.5$, el ρ_c se mantiene estable a 0.92.

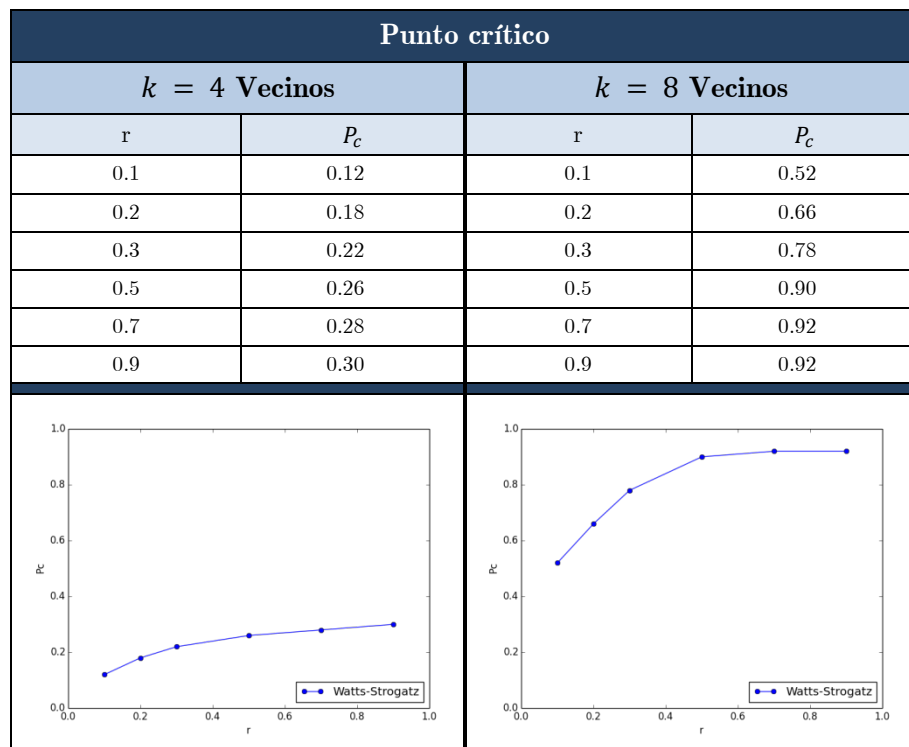


Tabla 5.9 Evolución del punto crítico en las redes Watts-Strogatz sobre el algoritmo A3 con diferentes valores de r en función del número de vecinos k .

5.1.3.4 Comparativa de algoritmos

En esta sección vamos a comparar los tres algoritmos de enrutamiento que hemos propuesto y estudiado sobre las redes Watts-Strogatz. En la [figura 5.15 izqda.](#) se muestra la comparativa de la congestión en la red Watts-Strogatz con $r = 0.2$ y $k = 4$. Esta figura muestra como en este tipo de redes la congestión ha disminuido. Esta disminución es mucho más baja que en el caso de las redes Erdős-Rényi y Barabási-Albert. La optimización desciende cuando $\rho \rightarrow 1$, siendo, para $\rho = 1$ de, aproximadamente, un 0.02 entre el algoritmo A1 y el A2 y entre el algoritmo A2 y el A3. Para valores bajos de ρ la diferencia entre los algoritmos es mucho más grande (~ 0.10).

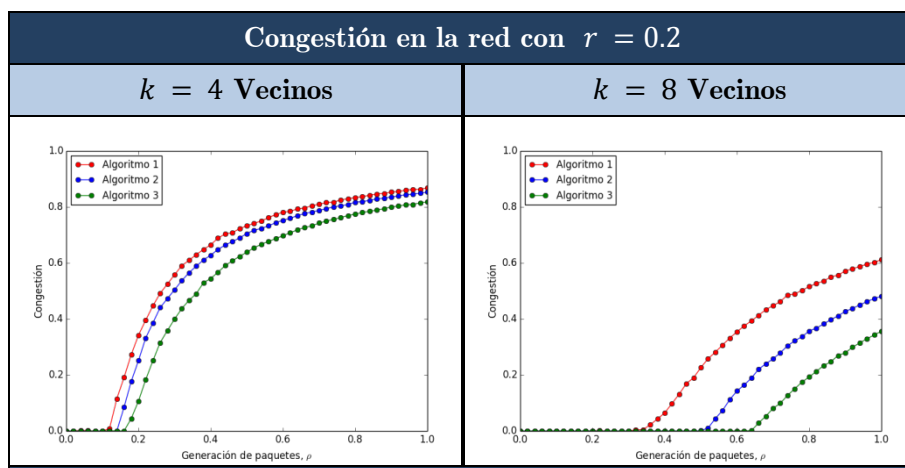


Figura 5.15. Comparativa de los distintos algoritmos de enrutamiento en la evolución de la congestión sobre la red Watts-Strogatz con $r = 0.2$, con $k = 4$ (**izquierda**) y $k = 8$ (**derecha**) vecinos de media en la distribución de grado.

En la [figura 5.15 drcha.](#) se muestra la comparativa de la congestión en la red Watts-Strogatz con $r = 0.2$ y $k = 8$. En este caso la congestión ha disminuido significativamente con el A3, hasta 0.40 con respecto del A1 y un 0.20 con respecto del algoritmo A2, en $\rho = 0.64$. Cuando $\rho \rightarrow 1$ esta disminución se va reduciendo, pero de todas formas sigue siendo bastante significativa. Recordamos en este punto que la red con $k = 8$ tiene el doble de enlaces que la red con $k = 4$, por lo tanto, se

evidencia que, en este tipo de redes, cuantos más enlaces disponga la red, más eficiente serán los algoritmos A2 y A3 sobre el A1 respectivamente.

En la [figura 5.16 izqda.](#) se muestra cómo evolucionan los puntos críticos en las redes con $k = 4$. Vemos como los tres algoritmos de enrutamiento tienen la misma tendencia en la evolución con respecto a r . En este caso la diferencia entre ellos es de entre 0.02 a 0.05, por lo que podemos decir que se produce poca optimización en cuanto al punto crítico., aun así, el Algoritmo A3 es el mejor en cuanto a aumentar los puntos críticos

En el caso de $k = 8$ ([figura 5.16 drcha.](#)), vemos como el algoritmo A2 y algoritmo A3 tienen la misma tendencia, creciendo con una ligera curva hasta llegar a la red con $r = 0.5$, donde se quedan fijos a 0.84 y 0.92 respectivamente. Con el algoritmo A1 vemos que los puntos críticos se quedan fijos en 0.58 para las redes con $r > 0.5$.

En general vemos como el algoritmo A3 es el que mejor optimiza los puntos críticos, llegándolo a aumentar en la red con $r = 5$ y $k = 8$ en 0.36 con respecto del Algoritmo A1. El algoritmo A2 también los optimiza considerablemente con respecto del algoritmo A1. De esta forma, igual que en el caso de la congestión, cuantos más enlaces tenga la red, más será la optimización de puntos críticos.

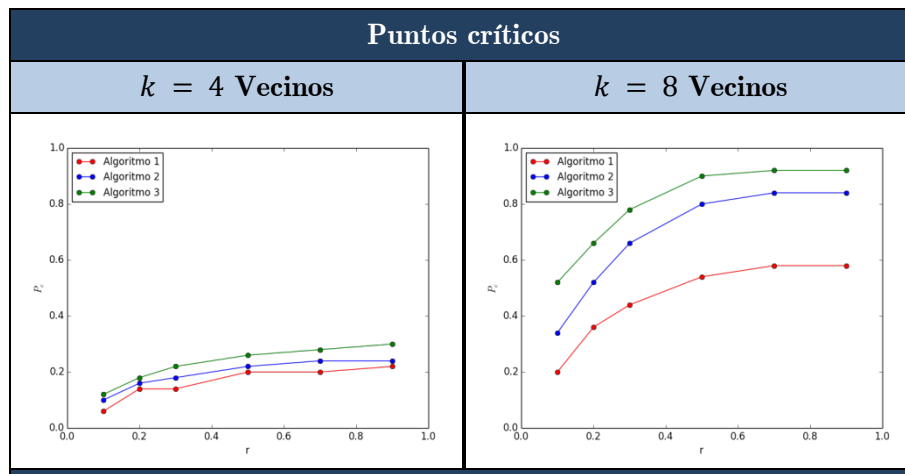


Figura 5.16. Comparativa de los distintos algoritmos de enrutamiento en la evolución de los puntos críticos sobre todas las redes Watts-Strogatz con $k = 4$ (**izquierda**) y $k = 8$ (**derecha**) de media en la distribución de grado.

5.2 Congestión en el Modelo 1.1

En este caso vamos a introducir el parámetro de regeneración (ν) definido en el [modelo 1.1](#). Este parámetro nos indica la probabilidad de regenerar el link máximamente entrelazado ente los vecinos de la red. Todas las simulaciones tendrás los siguientes parámetros en común: $\mathbf{v_ini} = 0.2$, $\mathbf{v_fin} = 0.8$, $\mathbf{W} = 4$ y $\mathbf{t} = 1000$ pasos, el resto de parámetros se especificará en cada una de las subsecciones. En esta ocasión se omite la probabilidad $\nu = 1$ dado que los resultados obtenidos en [5.1](#) son, realmente, como si hubiéramos definido esta variable a 1. A pesar de esto, $\nu = 1$ si se tendrá en cuenta en la tabla de comparación de puntos críticos.

Como hemos en el capítulo anterior, para los algoritmos A2 y A3 la tendencia en el consumo de enlaces es exactamente la mismo para cualquier ν , solo cambia la pendiente que describe hasta el momento en el que se produce la congestión. En el caso del algoritmo A1 para cada incremento de ν , también se incrementa el número de enlaces utilizados hasta llegar al ρ_c , aunque todas las redes acaban consumiendo, aproximadamente, el mismo número de enlace. Este comportamiento se repite en todos los modelos de red, así que a partir de ahora se omitirán los gráficos de utilización de enlaces.

Los resultados obtenidos se dispondrán en función de la variable ν para comprobar como esta afecta a la congestión de las redes. Además, al final de cada modelo de red se mostrará una comparativa entre los distintos algoritmos de enrutamiento.

5.2.1 Redes Erdős–Rényi

5.2.1.1 Algoritmo A1

En la [figura 5.17](#) podemos comparar cómo evoluciona la congestión en función de los parámetros de la red (z) y en función de la probabilidad de regenerar el enlace (ν) sobre el Algoritmo A1. Vemos como el parámetro ν influye, como era lógico, en la congestión de la red. Todas las redes se comportan de forma igual a este parámetro. Referente a los puntos críticos, estos crecen casi linealmente en función de z para cualquier ν .

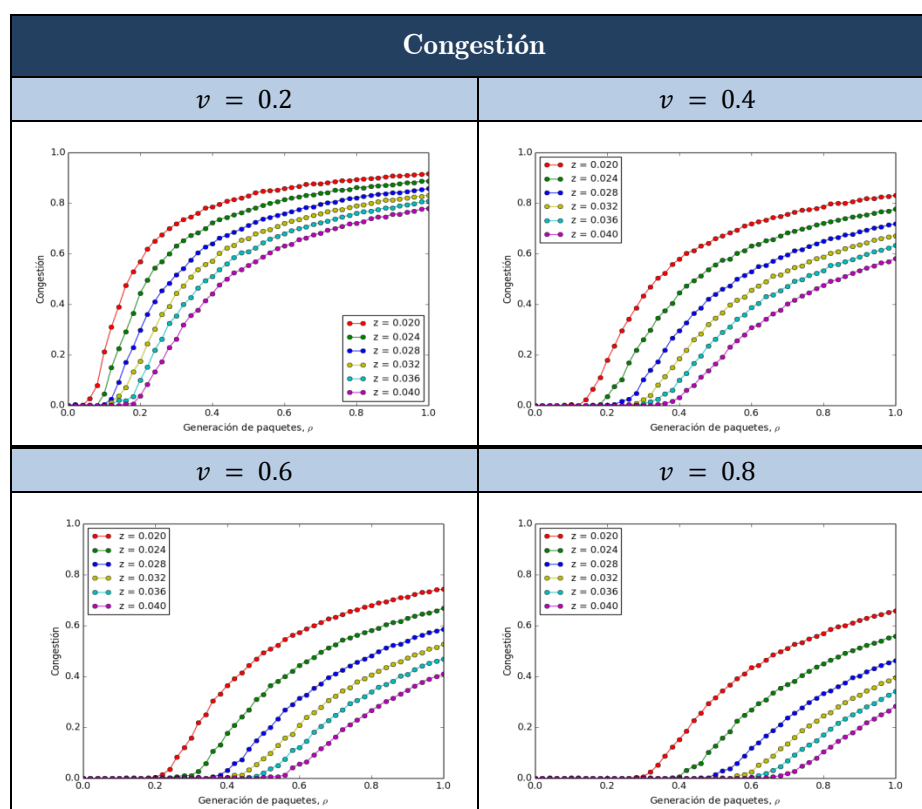


Figura 5.17. Evolución de la congestión en las redes Erdős-Rényi con diferentes valores de z sobre el Algoritmo A1, en función de la probabilidad de regenerar un link v .

En cuanto a los puntos críticos crecen linealmente en función de z , siendo para valores altos de v , el crecimiento es más pronunciado.

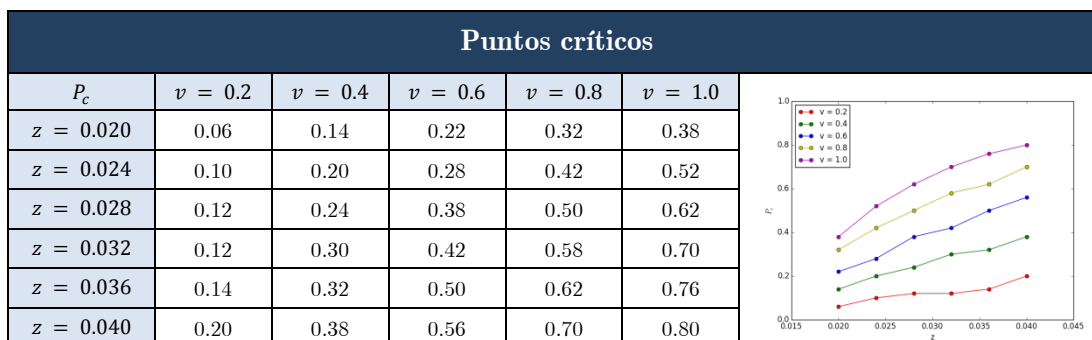


Tabla 5.10. Tabla comparativa y gráfico de puntos críticos en las redes Erdős-Rényi con diferentes valores de z sobre el Algoritmo A1, en función de la probabilidad de regenerar un link v .

5.2.1.2 Algoritmo A2

En la figura 5.18 podemos comparar cómo evoluciona la congestión en función de los parámetros de la red (z) y en función de la probabilidad de regenerar el enlace (v)

sobre el Algoritmo A2. La congestión disminuye conforme $v \rightarrow 1$, para cada incremento más significativamente.

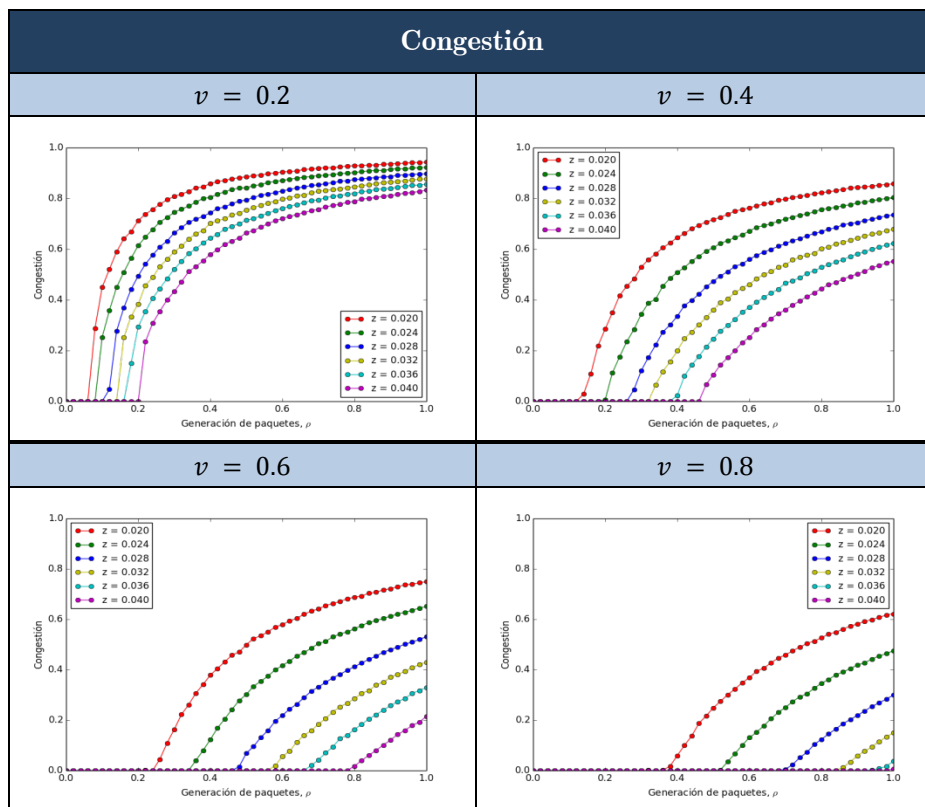


Figura 5.18. Evolución de la congestión (derecha) y media de enlaces utilizados (izquierda) en las redes Erdős-Rényi con diferentes valores de z sobre el Algoritmo A2, en función de v .

En cuanto a los puntos críticos (tabla 5.11), estos crecen linealmente en función de z , siendo para valores altos de v , el crecimiento es más pronunciado (hay más diferencia entre $\rho_c(z)$ y $\rho_c(z + 1)$).

Puntos críticos					
P_c	$v = 0.2$	$v = 0.4$	$v = 0.6$	$v = 0.8$	$v = 1.0$
$z = 0.020$	0.08	0.14	0.26	0.38	0.52
$z = 0.024$	0.10	0.20	0.36	0.80	0.72
$z = 0.028$	0.12	0.28	0.48	0.72	0.94
$z = 0.032$	0.16	0.34	0.58	0.86	1.00
$z = 0.036$	0.18	0.40	0.68	0.96	NO
$z = 0.040$	0.22	0.48	0.80	1.0	NO

Tabla 5.11. Tabla comparativa y gráfico de puntos críticos en las redes Erdős-Rényi con diferentes valores de z sobre el Algoritmo A2, en función de la probabilidad de regenerar un link v .

5.2.1.3 Algoritmo A3

En la [figura 5.19](#) podemos comparar cómo evoluciona la congestión en función de los parámetros de la red (z) y en función de la probabilidad de regenerar el enlace (v) sobre el Algoritmo A3. La congestión disminuye conforme $v \rightarrow 1$.

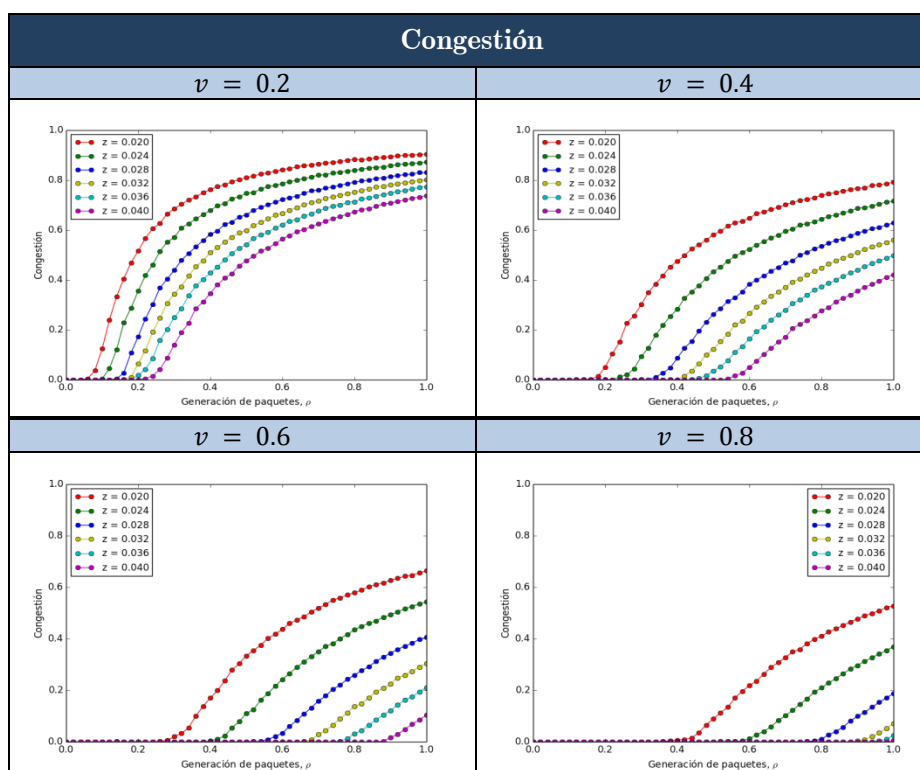


Figura 5.19. Evolución de la congestión en las redes Erdős-Rényi con diferentes valores de z sobre el algoritmo A3 y en función de v .

En cuanto a los puntos críticos ([tabla 5.12](#)), estos crecen linealmente en función de z . Conforme $v \rightarrow 1$, el crecimiento es más pronunciado (hay más diferencia entre $\rho_c(z)$ y $\rho_c(z + 1)$) hasta llegar a 1.

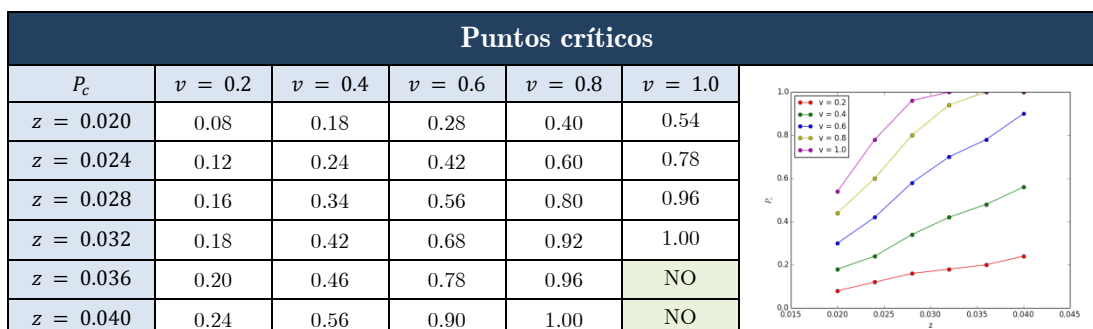


Tabla 5.12. Tabla comparativa y gráfico de puntos críticos en las redes Erdős-Rényi con diferentes valores de z sobre el algoritmo A3, en función de la probabilidad de regenerar un link v .

5.2.1.4 Comparativa de algoritmos

En esta sección vamos a comparar los tres algoritmos de enrutamiento que hemos propuesto y estudiado sobre las redes Erdős–Rényi. En la [figura 5.20](#) se muestra la comparativa de la congestión en la red Erdős–Rényi con $z = 0.02$ en función de v . Al contrario de lo que hemos visto hasta ahora, vemos que el algoritmo A2 es menos eficiente que el algoritmo A1 para valores de $v < 0.6$, e en el caso de $v = 0.6$ es igual, y a partir de valores de $v > 0.6$ ya empieza a ser más eficiente el algoritmo A2 que el algoritmo A1. Con respecto al algoritmo A3, este es más eficiente para todas los valores de $v > 0.2$ ya que con $v = 0.2$ es prácticamente igual que el algoritmo A1.

En general, el Algoritmo A3 reduce más la congestión cuando $v \rightarrow 1$ y el Algoritmo A2 es menos eficiente que el Algoritmo A1 cuando $v \rightarrow 0$.

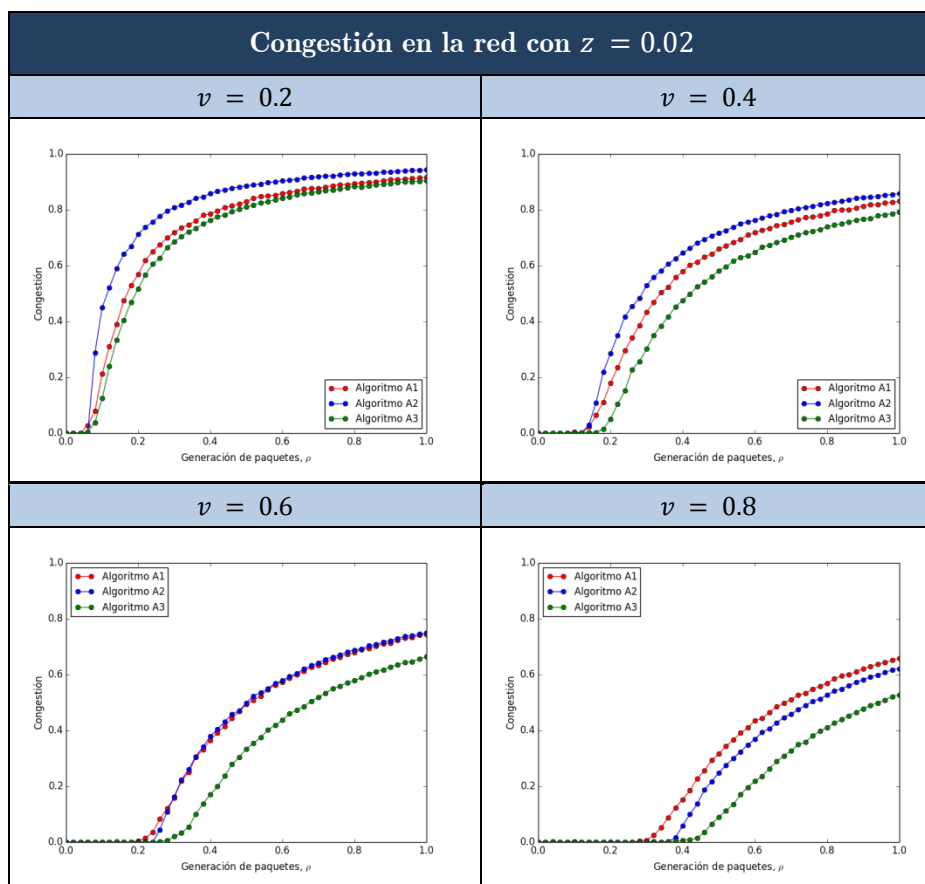


Figura 5.20. Evolución de la congestión en la red Erdős–Rényi con $z = 0.02$ con los distintos algoritmos de enrutamiento en función de v .

La [figura 5.21](#) muestra los puntos críticos de todas las redes Erdős–Rényi, en función del parámetro ν . Podemos ver como para valores bajos de ν apenas hay diferencia entre los tres algoritmos. A pesar de que el Algoritmo A2 es menos eficiente que el algoritmo A1 para valores de $\nu < 0.6$ vemos como los puntos críticos son más altos (mejores) que en el algoritmo A1. Cuando $\nu \rightarrow 1$ vemos cómo se van distanciando cada vez más el algoritmo A2 y algoritmo A3 del algoritmo A1, manteniéndose este último siempre bastante bajo. Los algoritmos A2 y A3 evolucionan de manera prácticamente igual, aunque entre los dos hay una diferencia de, aproximadamente, 0.1.

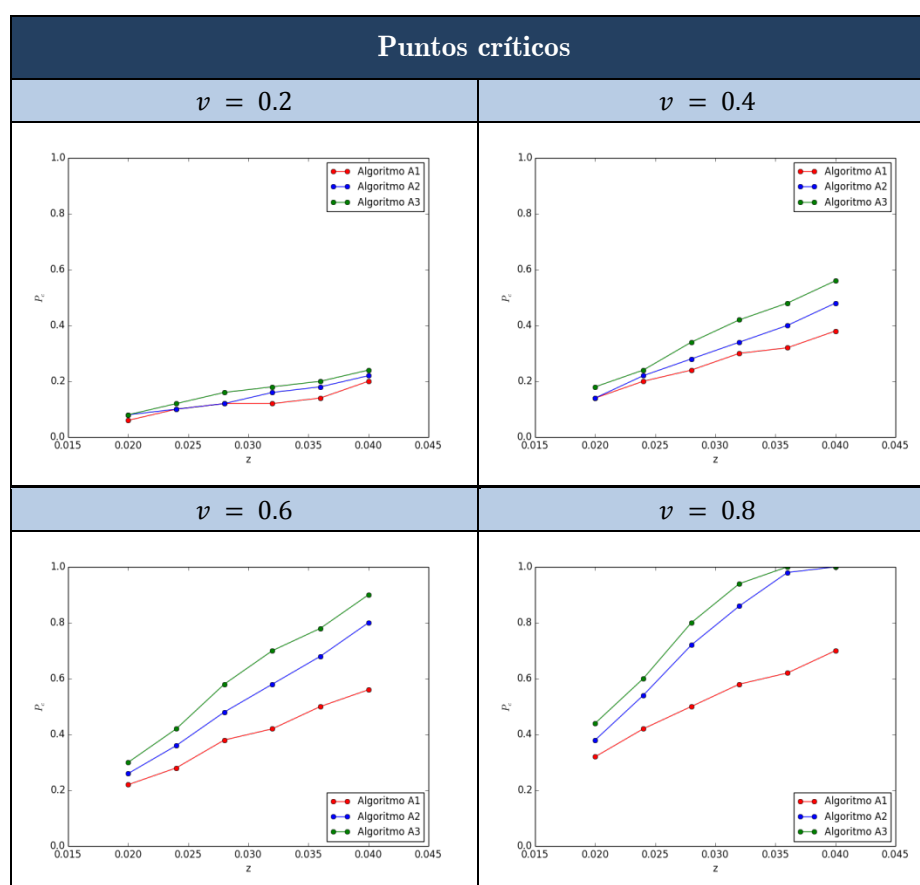


Figura 5.21. Evolución de los puntos críticos en las redes Erdős–Rényi con los distintos algoritmos de enrutamiento en función de ν .

En general, como en el caso de la congestión, el algoritmo A3 es el más eficiente de los tres en este tipo de redes.

5.2.2 Redes Barabási–Albert

5.2.2.1 Algoritmo A1

En la [figura 5.22](#) podemos comparar cómo evoluciona la congestión en función de los parámetros de la red (m) y en función de la probabilidad de regenerar el enlace (v) sobre el Algoritmo A1. En este caso la congestión también disminuye conforme $v \rightarrow 1$, salvo la red con $m = 1$, que mantiene la misma congestión sea cual sea v . Para valores bajos de v y de ρ , hay mucha más diferencia en la congestión entre las distintas redes que para valores altos de estos parámetros.

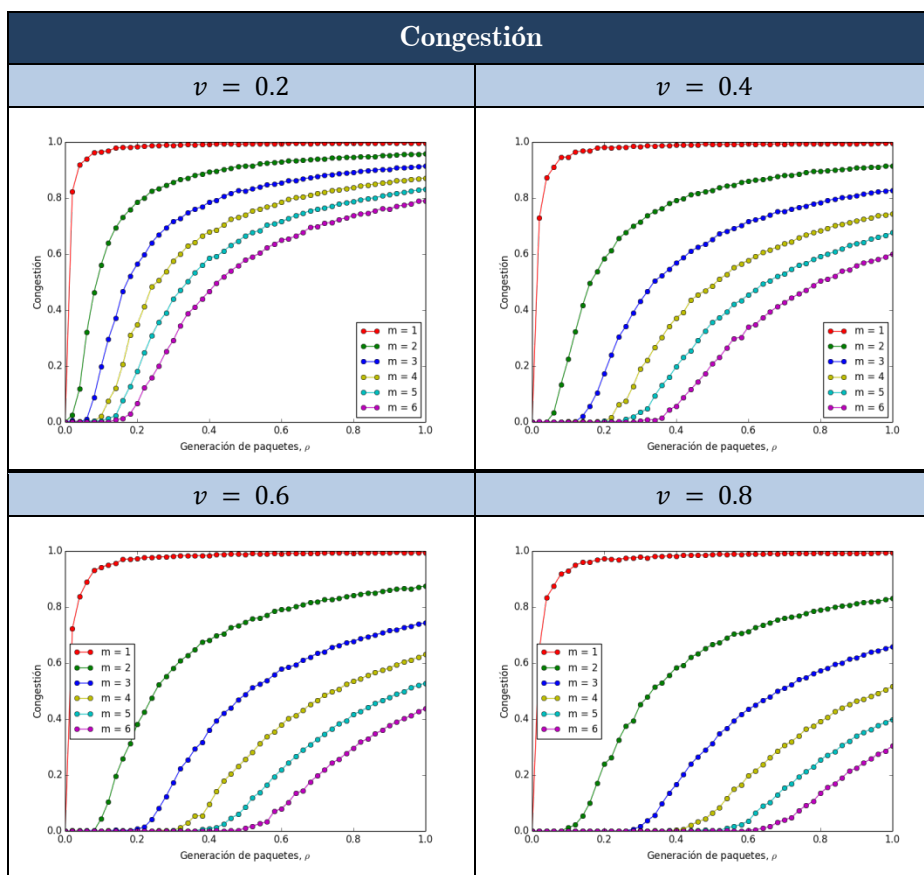


Figura 5.22. Evolución de la congestión en las redes Barabási–Albert sobre el Algoritmo A1 con diferentes valores de m en función de v .

En cuanto a los puntos críticos ([tabla 5.13](#)), vemos como para todas las v se mantiene el punto crítico en la red con $m = 2$. Para el resto de valores, cuanto más grande sea v más equidistantes están los puntos críticos (hay más diferencia entre $\rho_c(z)$ y $\rho_c(z + 1)$). En este caso los puntos críticos tienen tendencia a crecer prácticamente linealmente en función de z .

Puntos críticos					
P_c	$v = 0.2$	$v = 0.4$	$v = 0.6$	$v = 0.8$	$v = 1.0$
$m = 1$	0.02	0.02	0.02	0.02	0.02
$m = 2$	0.02	0.06	0.10	0.10	0.14
$m = 4$	0.02	0.14	0.20	0.28	0.38
$m = 6$	0.10	0.22	0.32	0.42	0.54
$m = 8$	0.12	0.26	0.38	0.54	0.68
$m = 10$	0.14	0.32	0.50	0.76	0.76

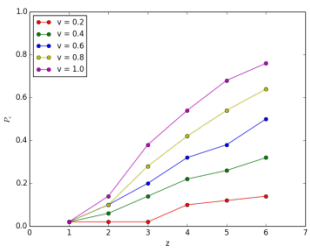


Tabla 5.13. Tabla comparativa de puntos críticos en las redes Barabási–Albert sobre el Algoritmo A1 con diferentes valores de m en función de la probabilidad de regenerar un link v .

5.2.2.2 Algoritmo A2

En la figura 5.23 podemos comparar cómo evoluciona la congestión en función de los parámetros de la red (m) y en función de la probabilidad de regenerar el enlace (v) sobre el algoritmo A2. En este caso la congestión se reduce cada vez más conforme $v \rightarrow 1$. Cuando $v \rightarrow 0$, la mayor optimización viene dada por valores bajos de ρ .

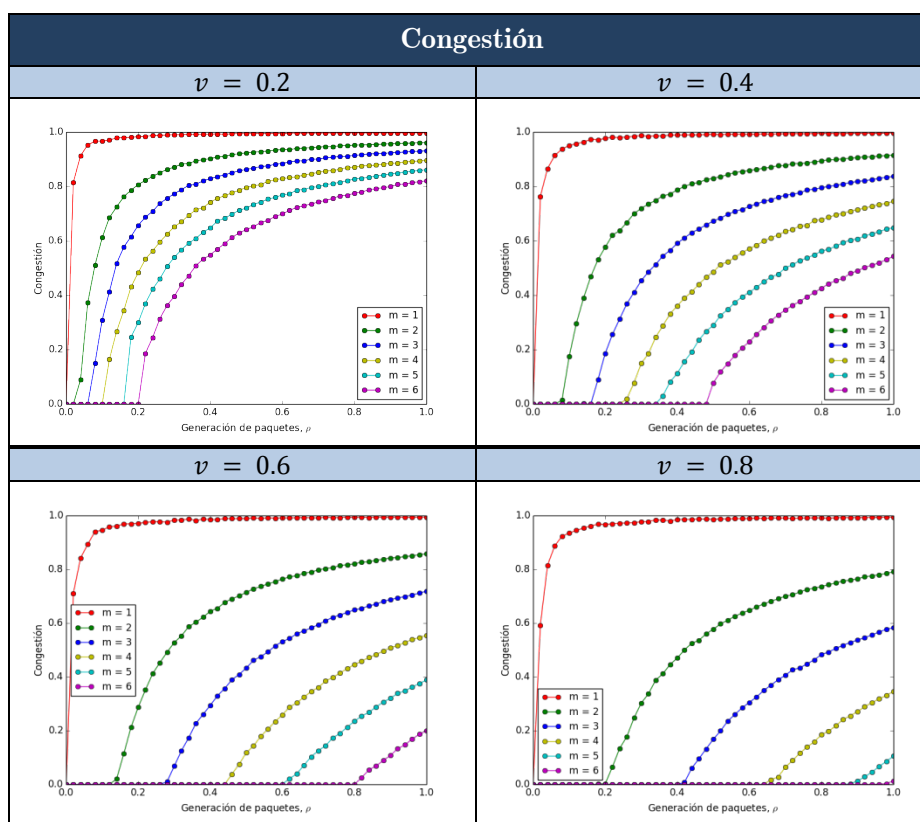


Figura 5.23. Evolución de la congestión en las redes Barabási–Albert sobre el algoritmo A2 con diferentes valores de m en función de la probabilidad de regenerar un enlace v .

En cuanto a los puntos críticos (tabla 5.14), estos crecen linealmente en función de z . Para valores bajos de v apenas hay diferencia entre los puntos críticos de las distintas redes, pero conforme $v \rightarrow 1$, el crecimiento es más pronunciado y, por lo tanto, hay más diferencia entre $\rho_c(z)$ y $\rho_c(z + 1)$. Vemos como la red con $m = 6$, en $v = 1$ no genera congestión. Por lo tanto se puede afirmar que para todas las redes con $m > 6$ en $v = 1$ no generarán congestión

Puntos críticos					
P_c	$v = 0.2$	$v = 0.4$	$v = 0.6$	$v = 0.8$	$v = 1.0$
$m = 1$	0.02	0.02	0.02	0.02	0.02
$m = 2$	0.04	0.08	0.14	0.22	0.30
$m = 3$	0.08	0.18	0.28	0.42	0.58
$m = 4$	0.12	0.26	0.46	0.66	0.88
$m = 5$	0.18	0.36	0.62	0.90	1.00
$m = 6$	0.22	0.50	0.82	1.00	NO

Tabla 5.14. Tabla comparativa de puntos críticos en las redes Barabási–Albert sobre el algoritmo A2 con diferentes valores de m en función de la probabilidad de regenerar un link v .

5.2.2.3 Algoritmo A3

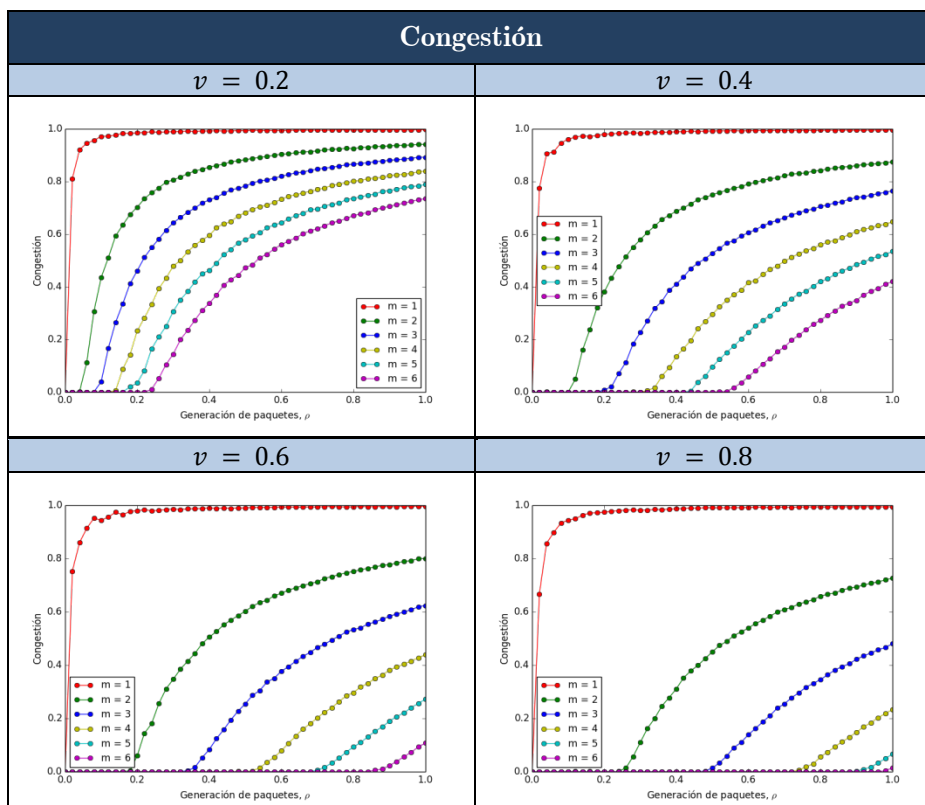


Figura 5.24. Evolución de la congestión en las redes Barabási–Albert sobre el algoritmo A3 con diferentes valores de m en función de la probabilidad de regenerar un enlace v .

En la [figura 5.24](#) podemos comparar cómo evoluciona la congestión en función de los parámetros de la red (m) y en función de la probabilidad de regenerar el enlace (v) sobre el algoritmo A2. En este caso la congestión se reduce cada vez más conforme $v \rightarrow 1$, menos en la red con $m = 1$ donde se mantiene igual sea cual sea el valor de v .

En la [tabla 5.15](#) se muestran los puntos críticos extraídos de la [figura 5.24](#). Como podemos ver estos crecen linealmente en función de z . Para valores bajos de v hay poca diferencia entre los puntos críticos de las distintas redes, pero conforme $v \rightarrow 1$, el crecimiento es más pronunciado y, por lo tanto, hay más diferencia entre $\rho_c(z)$ y $\rho_c(z + 1)$. En este caso la red con $m = 6$ y $v = 1$ no tiene punto crítico, por lo que cuando se regeneren en cada paso de tiempo todos los enlaces, esta nunca producirá congestión.

Puntos críticos					
P_c	$v = 0.2$	$v = 0.4$	$v = 0.6$	$v = 0.8$	$v = 1.0$
$m = 1$	0.02	0.02	0.02	0.02	0.02
$m = 2$	0.06	0.12	0.20	0.26	0.34
$m = 3$	0.10	0.20	0.36	0.50	0.64
$m = 4$	0.14	0.32	0.54	0.74	0.90
$m = 5$	0.18	0.46	0.70	0.92	1.00
$m = 6$	0.24	0.56	0.86	1.0	NO

Tabla 5.15. Tabla comparativa de puntos críticos en las redes Barabási–Albert sobre el algoritmo A3 en función de la probabilidad de regenerar un link v .

5.2.2.4 Comparativa de algoritmos

En esta sección vamos a comprar los tres algoritmos de enrutamiento que hemos propuesto y estudiado sobre las redes Barabási–Albert. En la [figura 5.25](#) se muestra la comparativa de la congestión en la red Barabási–Albert con $m = 2$ en función de v . Igual que en el modelo de red Erdős–Rényi, vemos que el algoritmo A2 es menos eficiente que el algoritmo A1, en este caso para valores de $v < 0.4$. Para $v = 0.4$ es igual, y a partir de valores de $v > 0.4$ ya empieza a ser más eficiente el

algoritmo A2 que el algoritmo A1. Con respecto al algoritmo A3, este es más eficiente para todas los valores de v .

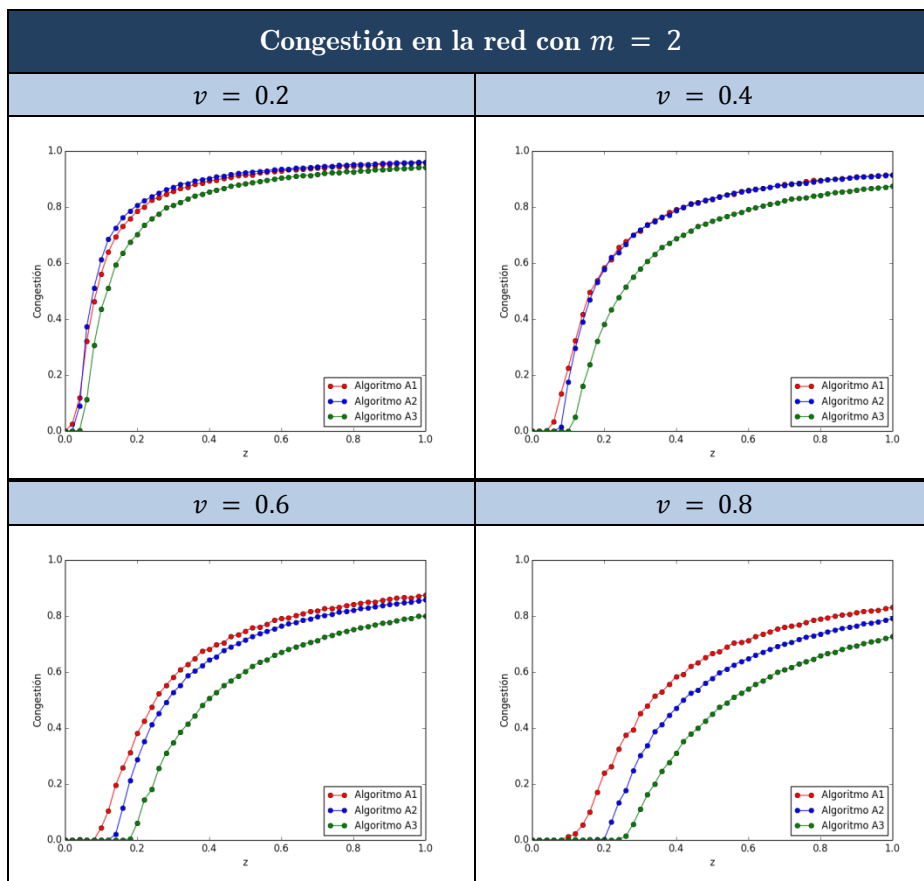


Figura 5.25. Comparativa de los distintos algoritmos de enrutamiento en la evolución de la congestión en la red Barabási-Albert con $m = 2$.

En general, el algoritmo A3 reduce más la congestión cuando $v \rightarrow 1$ y el algoritmo A2 es poco eficiente cuando $v \rightarrow 0$. La curvatura de la gráfica nos indica que la diferencia en la mejora entre los distintos algoritmos va disminuyendo, por lo tanto, para valores bajos de ρ el algoritmo A3 es más eficiente que cuando $\rho \rightarrow 1$.

En cuanto a los puntos críticos, vemos en la figura 5.26 cómo evolucionan en función de v . Tenemos que para valores bajos de v ($v \leq 2$), los puntos críticos son prácticamente iguales para todas las redes en los algoritmos algoritmo A2 y algoritmo A3, estos recen casi paralelamente en función de m , sea cual sea v . Igual

que en las redes Erdős–Rényi, el algoritmo A1 es el peor en cuanto a los puntos críticos, siendo el algoritmo A3 el mejor a no mucha distancia del algoritmo A2.

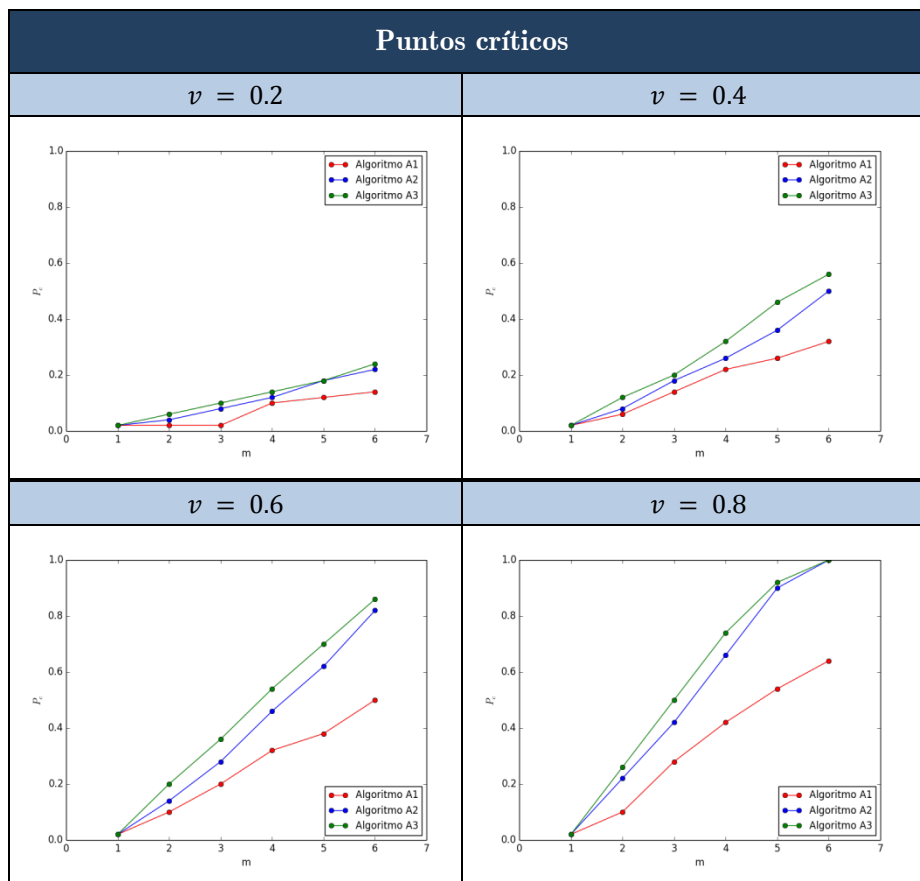


Figura 5.26. Evolución de los puntos críticos en las redes Barabási–Albert con los distintos algoritmos de enrutamiento en función de v .

5.2.3 Redes Watts-Strogatz

5.2.3.1 Algoritmo A1

En la figura 5.27 podemos comparar cómo evoluciona la congestión en función de los parámetros de la red, r y $k = 4$, y en función de la probabilidad de regenerar el enlace, v , sobre el algoritmo A1. En este tipo de modelo de red la congestión se reduce cada vez más conforme $v \rightarrow 1$, excepto en las redes con $r \geq 5$, donde la congestión es prácticamente igual sea cual sea v .

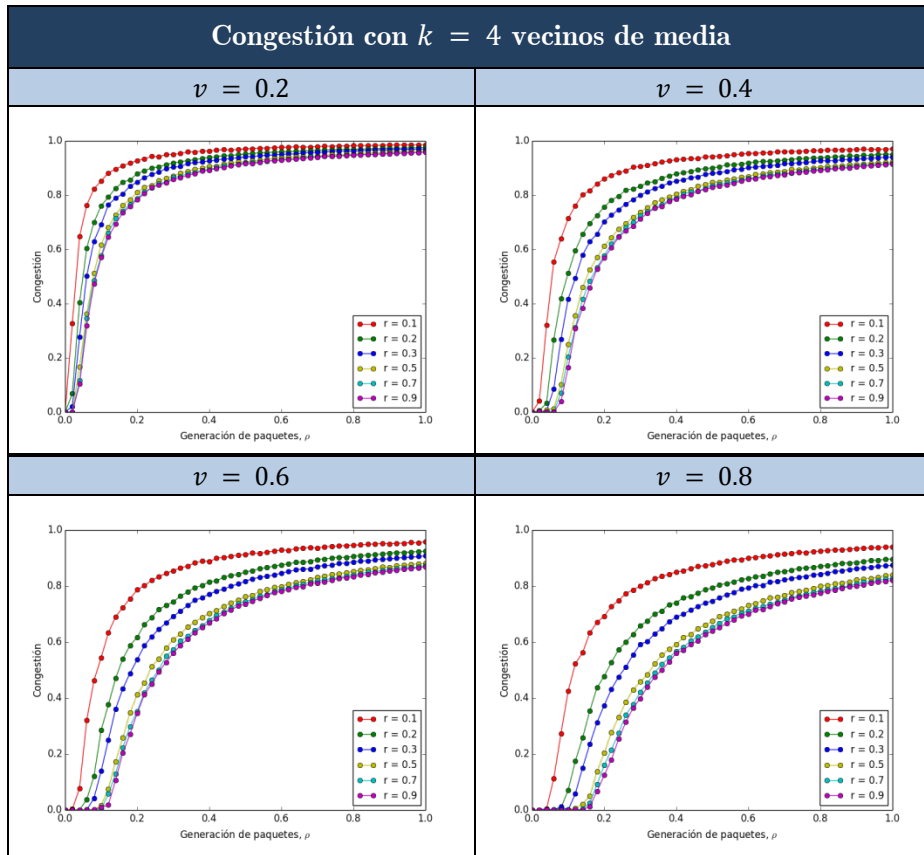


Figura 5.27. Evolución de la congestión en las redes Watts-Strogatz sobre el algoritmo A1 con $k = 4$ vecinos de media y con diferentes valores de r en función de la probabilidad v de regenerar un link.

En cuanto a $k = 8$, las figura 5.28 y 5.29 muestran cómo evoluciona la congestión en las distintas redes en función de v . Vemos que, a diferencia de $k = 4$, conforme $\rho \rightarrow 1$ hay más diferencia en la congestión entre las distintas redes, aunque para las redes con $r \geq 5$ la congestión sigue siendo prácticamente igual sea cual sea v .

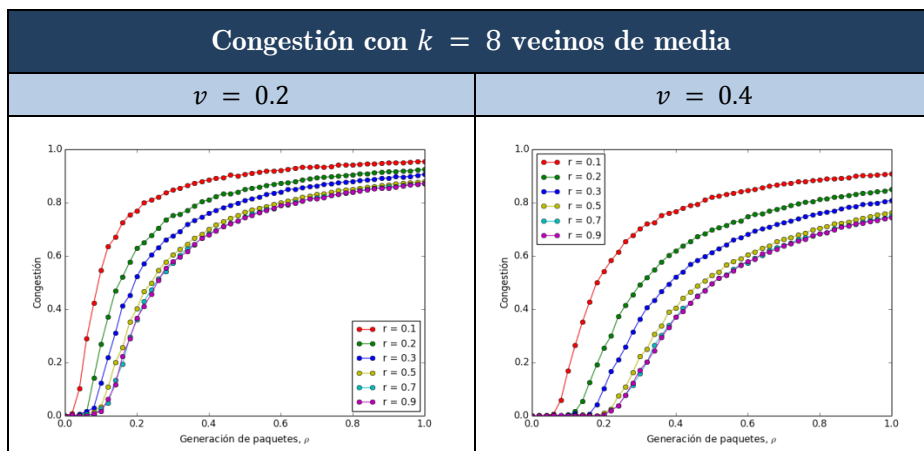


Figura 5.28. Evolución de la congestión en las redes Watts-Strogatz sobre el Algoritmo A1 con $k = 8$ vecinos de media y con diferentes valores de r en función de la probabilidad v de regenerar un link.

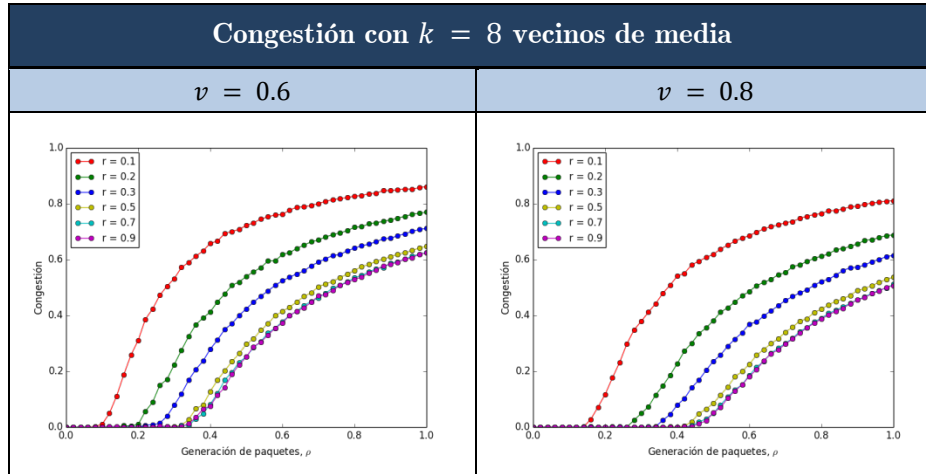


Figura 5.29. Evolución de la congestión en las redes Watts-Strogatz sobre el Algoritmo A1 con $k = 8$ vecinos de media y con diferentes valores de r en función de la probabilidad v de regenerar un link.

Vemos en la tabla 5.16 como efectivamente, para las redes con $r \geq 5$ la congestión es prácticamente igual sea cual sea v . En el caso de $k = 4$, los puntos críticos son prácticamente los mismos para todas las redes. En las redes con $k = 8$, para valores de $v \rightarrow 0$ hay poca diferencia en el ρ_c , pero para valores de $v \rightarrow 1$ hay más diferencia en el ρ_c entre las redes con $r < 5$.

Puntos críticos; Redes con $k = 4$ vecinos de media					
P_c	$v = 0.2$	$v = 0.4$	$v = 0.6$	$v = 0.8$	$v = 1.0$
$r = 0.1$	0.02	0.02	0.04	0.06	0.1
$r = 0.2$	0.02	0.04	0.06	0.10	0.16
$r = 0.3$	0.02	0.06	0.08	0.08	0.18
$r = 0.5$	0.04	0.06	0.10	0.14	0.22
$r = 0.7$	0.04	0.06	0.12	0.16	0.24
$r = 0.9$	0.04	0.08	0.12	0.16	0.24

Puntos críticos; Redes con $k = 8$ vecinos de media					
P_c	$v = 0.2$	$v = 0.4$	$v = 0.6$	$v = 0.8$	$v = 1.0$
$r = 0.1$	0.04	0.08	0.10	0.16	0.22
$r = 0.2$	0.06	0.12	0.22	0.28	0.36
$r = 0.3$	0.06	0.18	0.26	0.36	0.46
$r = 0.5$	0.08	0.20	0.34	0.44	0.56
$r = 0.7$	0.10	0.22	0.36	0.46	0.60
$r = 0.9$	0.10	0.22	0.34	0.46	0.60

Tabla 5.16. Tabla comparativa de puntos críticos en las redes Watts-Strogatz sobre el Algoritmo A1 con $k = 4$ y $k = 8$ vecinos de media en función de los distintos valores de regeneración de enlace v .

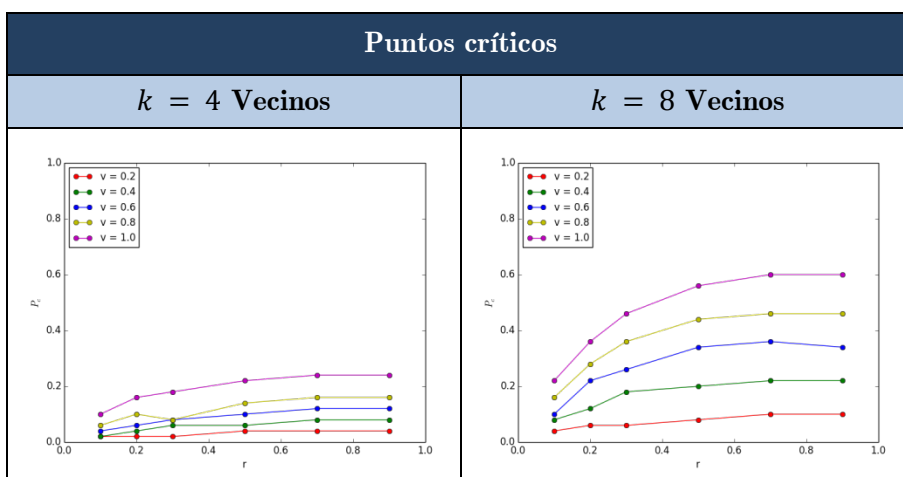


Tabla 5.17. Gráficos de puntos críticos en las redes Watts-Strogatz sobre el Algoritmo A1 con $k = 4$ y $k = 8$ vecinos de media en función de los distintos valores de regeneración de link v .

5.2.3.2 Algoritmo A2

En la [figura 5.30](#) y [5.31](#) podemos comparar cómo evoluciona la congestión en función de los parámetros de la red, r y $k = 4$, y en función de la probabilidad de regenerar el enlace, v , sobre el Algoritmo A2. En este tipo de modelo de red la congestión es prácticamente la misma para valores bajos de v , aunque para todas las redes con $r \geq 5$ la congestión es exactamente igual sea cual sea v . Conforme $v \rightarrow 1$ se va reduciendo la congestión, sobre todo para valores bajos de ρ , aunque cuando $\rho \rightarrow 1$, la reducción es mínima. El consumo de enlaces esta entre el 30% y el 50% en función de la red, igual que en la [figura 5.11](#).

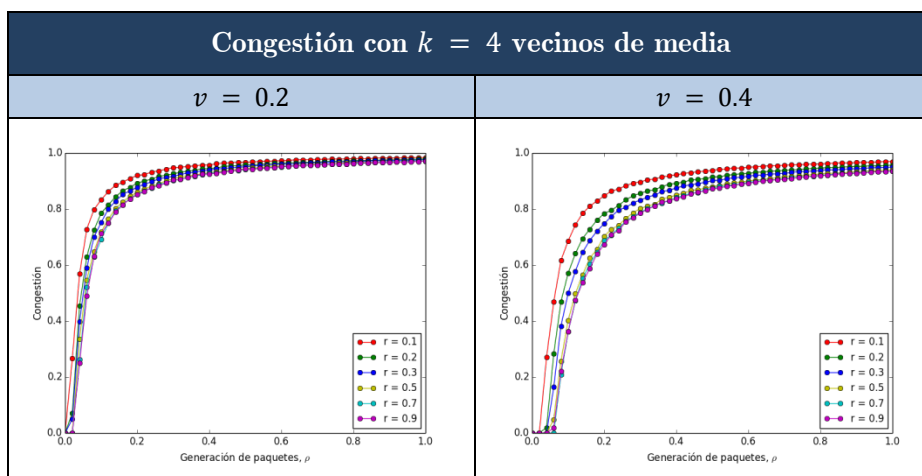


Figura 5.30. Evolución de la congestión en las redes Watts-Strogatz sobre el Algoritmo A2 con $k = 4$ vecinos de media y con diferentes valores de r en función de la probabilidad $v = 0.2$ y $v = 0.4$.

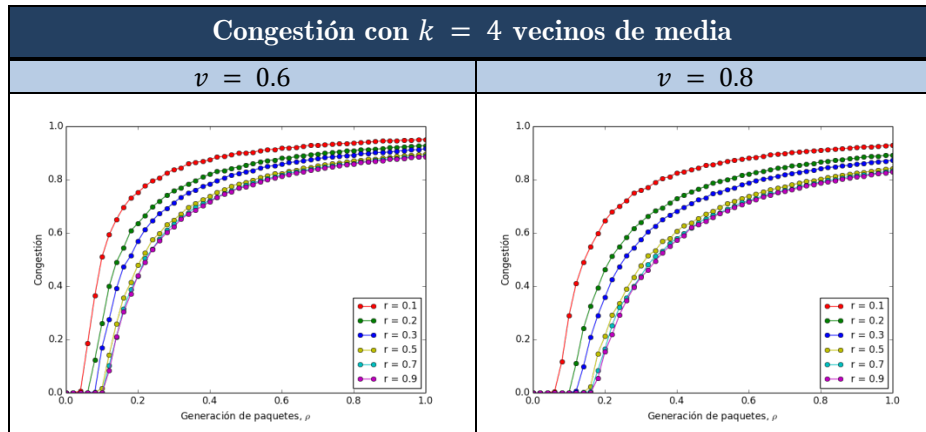


Figura 5.31. Evolución de la congestión en las redes Watts-Strogatz sobre el Algoritmo A2 con $k = 4$ vecinos de media y con diferentes valores de r en función de la probabilidad $v = 0.6$ y $v = 0.8$.

En el caso de $k = 8$, la figura 5.32 muestra cómo evoluciona la congestión en las distintas redes en función de v . Vemos que, a diferencia de $k = 4$, conforme $\rho \rightarrow 1$ hay más diferencia en la congestión entre las distintas redes, aunque para las redes con $r \geq 5$ la congestión es prácticamente igual sea cual sea v .

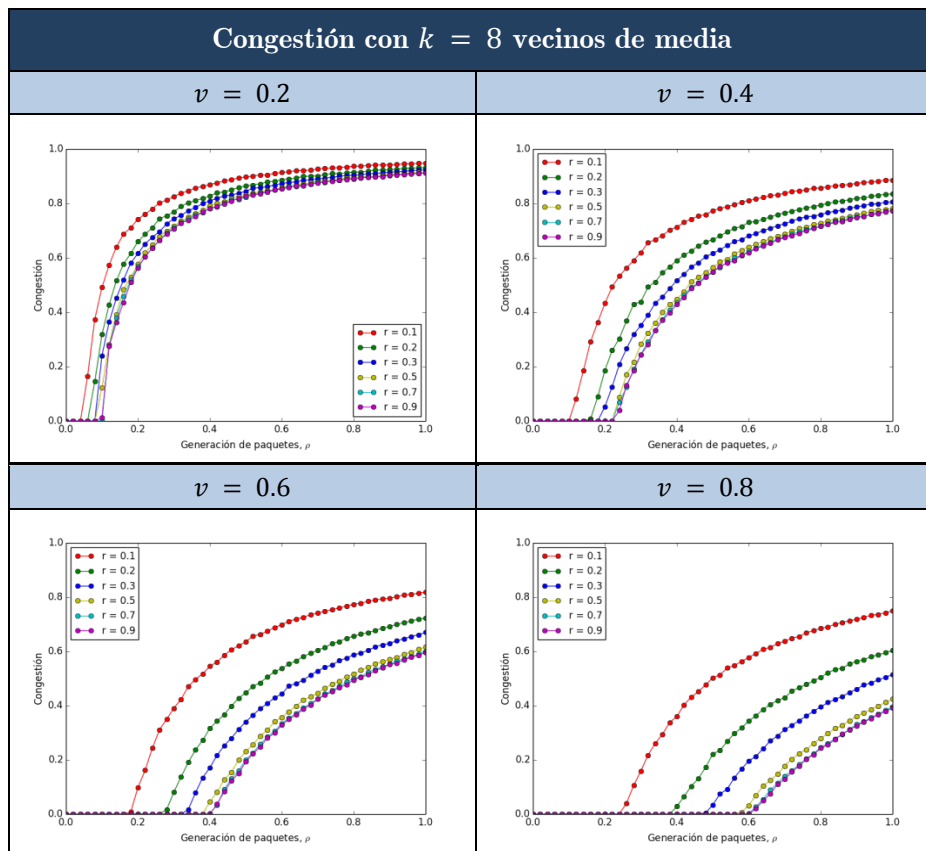


Figura 5.32. Evolución de la congestión en las redes Watts-Strogatz sobre el Algoritmo A2 con $k = 8$ vecinos de media y con diferentes valores de r en función de la probabilidad v de regenerar un link.

Vemos en la [tabla 5.18](#) como efectivamente, para las redes con $r \geq 5$ la congestión es prácticamente igual sea cual sea v . En el caso de $k = 4$, los puntos críticos son prácticamente los mismos para todas las redes (hay un pequeño incremento cuando incrementa v). En las redes con $k = 8$, para valores de $v \rightarrow 0$ hay poca diferencia en el ρ_c , pero para valores de $v \rightarrow 1$ hay más diferencia en el ρ_c entre las redes con $r < 5$.

Puntos críticos; Redes con $k = 4$ vecinos de media					
P_c	$v = 0.2$	$v = 0.4$	$v = 0.6$	$v = 0.8$	$v = 1.0$
$r = 0.1$	0.02	0.04	0.06	0.08	0.10
$r = 0.2$	0.02	0.04	0.08	0.12	0.16
$r = 0.3$	0.02	0.06	0.10	0.14	0.18
$r = 0.5$	0.04	0.06	0.10	0.16	0.22
$r = 0.7$	0.04	0.08	0.12	0.18	0.24
$r = 0.9$	0.04	0.06	0.12	0.18	0.24

Puntos críticos; Redes con $k = 8$ vecinos de media					
P_c	$v = 0.2$	$v = 0.4$	$v = 0.6$	$v = 0.8$	$v = 1.0$
$r = 0.1$	0.06	0.12	0.20	0.26	0.34
$r = 0.2$	0.08	0.18	0.28	0.40	0.52
$r = 0.3$	0.10	0.20	0.34	0.50	0.66
$r = 0.5$	0.10	0.24	0.40	0.60	0.80
$r = 0.7$	0.12	0.24	0.42	0.62	0.84
$r = 0.9$	0.10	0.24	0.42	0.62	0.84

$k = 4$ Vecinos	$k = 8$ Vecinos
-----------------	-----------------

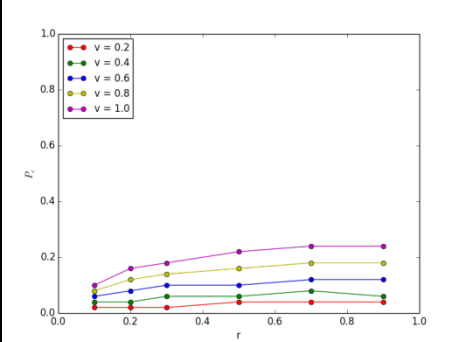
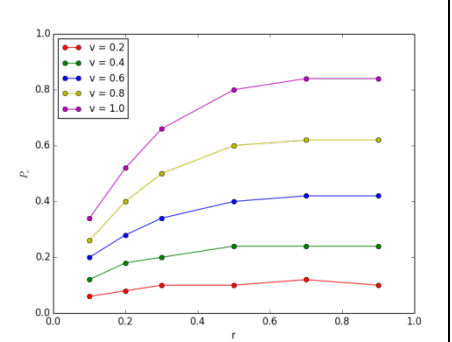



Tabla 5.18. Tabla comparativa de puntos críticos en las redes Watts-Strogatz sobre el Algoritmo A2 con $k = 4$ y $k = 8$ vecinos de media en función de los distintos valores de regeneración de link v .

5.2.3.3 Algoritmo A3

En la [figura 5.33](#) podemos comparar cómo evoluciona la congestión en función de los

parámetros de la red, r y $k = 4$, y en función de la probabilidad de regenerar el enlace, v , sobre el Algoritmo A3. En este tipo de modelo de red la congestión es prácticamente la misma para valores bajos de v y cuando $\rho \rightarrow 1$, aunque para todas las redes con $r \geq 5$ la congestión es prácticamente igual sea cual sea v y ρ . Conforme $v \rightarrow 1$ se va reduciendo la congestión en mayor o menor grado dependiendo de la red, sobre todo para valores bajos de ρ , cuando $\rho \rightarrow 1$, la reducción es más baja.

El consumo de enlaces esta entre el 35% y el 70% en función de la red, el límite del 70 % es para todas las redes con $r \geq 5$, igual que en la [figura 5.13](#).

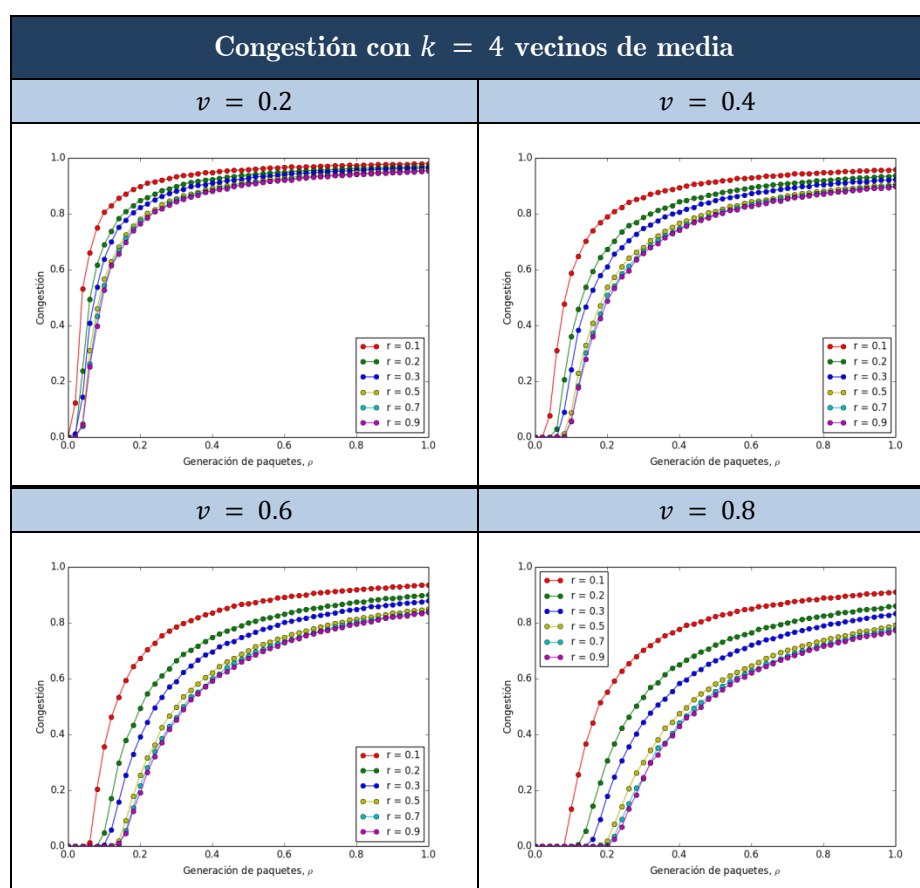


Figura 5.33. Evolución de la congestión en las redes Watts-Strogatz sobre el Algoritmo A3 con $k = 4$ vecinos de media y con diferentes valores de r en función de la probabilidad v de regenerar un enlace.

En el caso de $k = 8$, la [figura 5.34](#) muestra cómo evoluciona la congestión en las distintas redes en función de v . Vemos que, a diferencia de $k = 4$, conforme $\rho \rightarrow 1$ hay más diferencia en la congestión entre las distintas redes, aunque para las redes con $r \geq 5$ la congestión es prácticamente igual sea cual sea v .

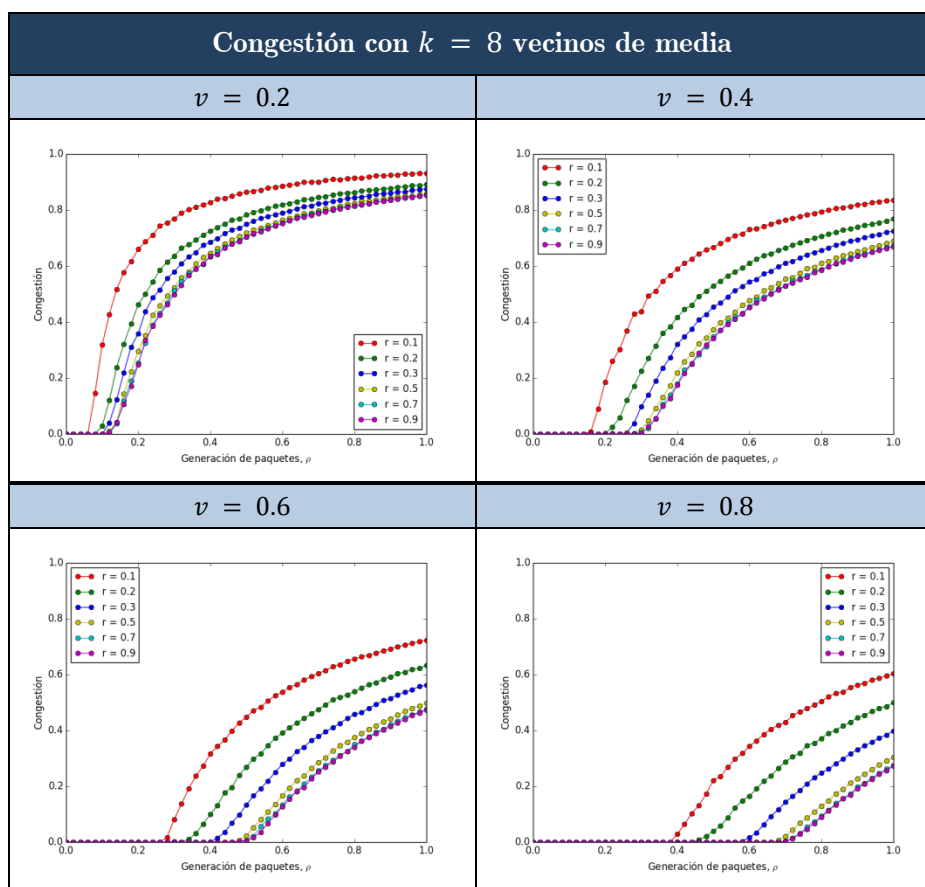


Figura 5.34. Evolución de la congestión en las redes Watts-Strogatz sobre el algoritmo A3 con $k = 8$ vecinos de media y con diferentes valores de r en función de la probabilidad v de regenerar un link.

En la [tabla 5.19](#) podemos ver cómo evolucionan los puntos críticos sobre el algoritmo A3 las redes Watts-Strogatz. Tanto para $k = 4$ como para $k = 8$, en las redes con $r > 5$ los puntos críticos son prácticamente igual sea cual sea v . En las redes con $r < 5$, en el caso de $k = 4$ los puntos críticos incrementan poco tanto de red en red, como en función de v sobre la misma red, en cambio, para $k = 8$, el incremento es más grande en las dos situaciones.

Puntos críticos; Redes con $k = 4$ vecinos de media					
P_c	$v = 0.2$	$v = 0.4$	$v = 0.6$	$v = 0.8$	$v = 1.0$
$r = 0.1$	0.02	0.04	0.06	0.10	0.10
$r = 0.2$	0.04	0.06	0.10	0.14	0.16
$r = 0.3$	0.02	0.08	0.12	0.16	0.18
$r = 0.5$	0.04	0.08	0.14	0.20	0.22
$r = 0.7$	0.04	0.10	0.16	0.22	0.24
$r = 0.9$	0.04	0.10	0.14	0.22	0.24

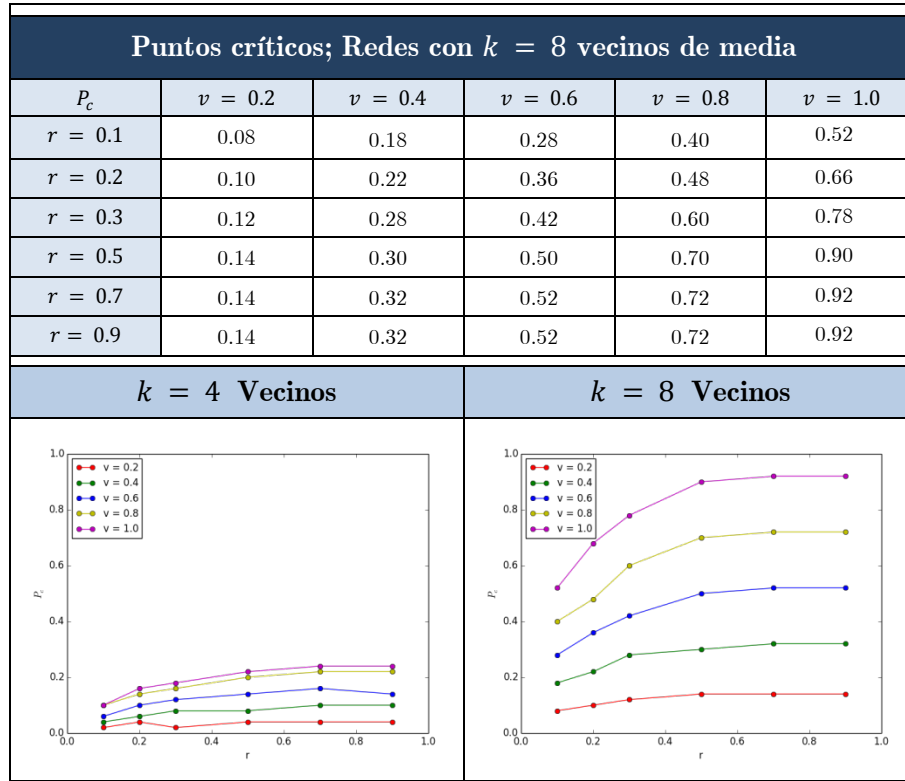


Tabla 5.19. Tabla comparativa de puntos críticos en las redes Watts-Strogatz sobre el algoritmo A3 con $k = 4$ $k = 8$ vecinos de media en función de los distintos valores de regeneración de enlace v .

5.2.3.4 Comparativa de algoritmos

En esta sección vamos a comprar los tres algoritmos de enrutamiento que hemos propuesto y estudiado sobre las redes Watts-Strogatz.

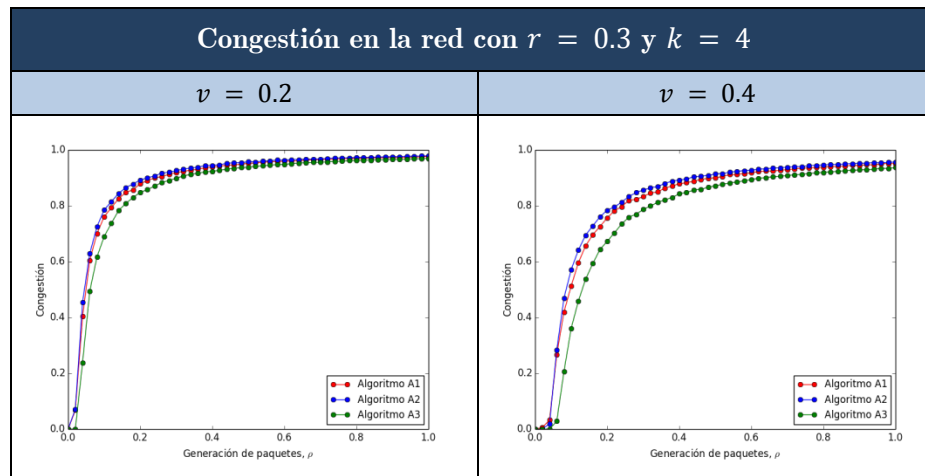


Figura 5.35. Comparativa de los distintos algoritmos de enrutamiento en la evolución de la congestión en la red Watts-Strogatz con $r = 0.3$ y $k = 4$, y con $v = 0.2$ y $v = 0.4$.

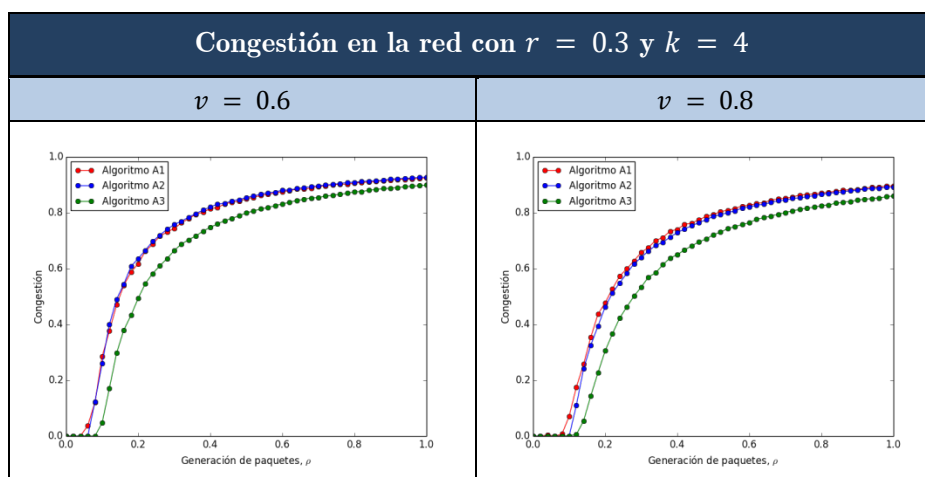


Figura 5.36. Comparativa de los distintos algoritmos de enrutamiento en la evolución de la congestión en la red Watts-Strogatz con $r = 0.3$ y $k = 4$, y con $v = 0.6$ y $v = 0.8$.

En la figura 5.35 y 5.36 se muestra la comparativa de la congestión en la red Watts-Strogatz con $r = 0.3$ y con $k = 4$ en función de v . Igual que en el modelo de red Erdős-Rényi, vemos que el Algoritmo A2 es menos eficiente que el Algoritmo A1, en este caso para valores de $v \leq 0.6$. Para $v = 0.6$ es igual, y a partir de valores de $v > 0.6$ ya empieza a ser más eficiente el Algoritmo A2. Con respecto al Algoritmo A3, este es más eficiente para todas los valores de $v > 0.5$, para el resto de valores esta igualada a los otros dos algoritmos.

En cuanto a los puntos críticos, para valores de $v < 0.5$, todos se encuentran prácticamente en el mismo lugar, apenas hay evolución.

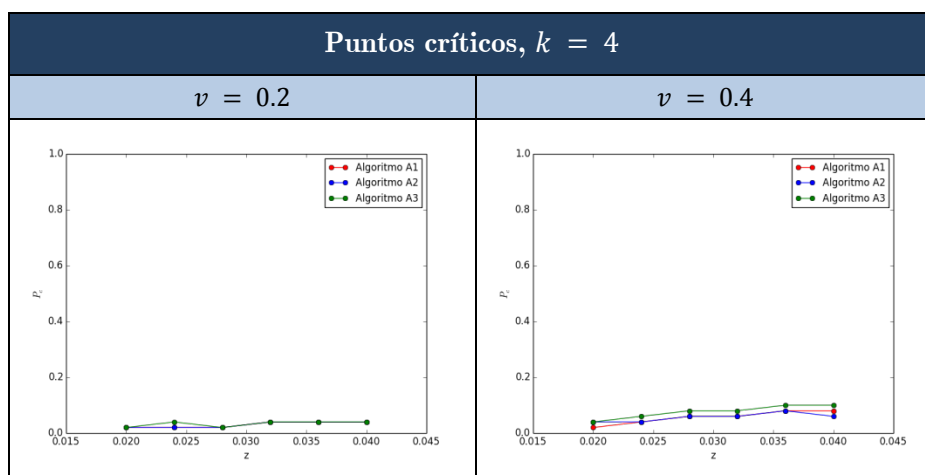


Figura 5.37. Evolución de los puntos críticos en las redes Watts-Strogatz con los distintos algoritmos de enrutamiento, con $k = 4$ y en función de $v = 0.2$ y $v = 0.4$.

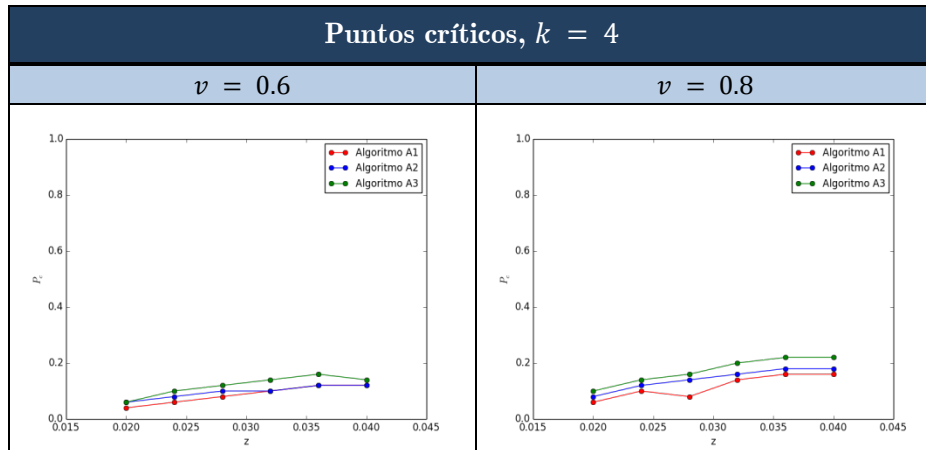


Figura 5.38. Evolución de los puntos críticos en las redes Watts-Strogatz con los distintos algoritmos de enrutamiento, con $k = 4$ y en función de $v = 0.6$ y $v = 0.8$.

En la figura 5.39 se muestra la comparativa de la congestión en la red Watts-Strogatz con $r = 0.3$ y con $k = 8$ en función de v . En este caso el algoritmo A2 es peor que el A1 en valores de $v < 0.2$. El algoritmo A3 reduce más la congestión en todos los casos.

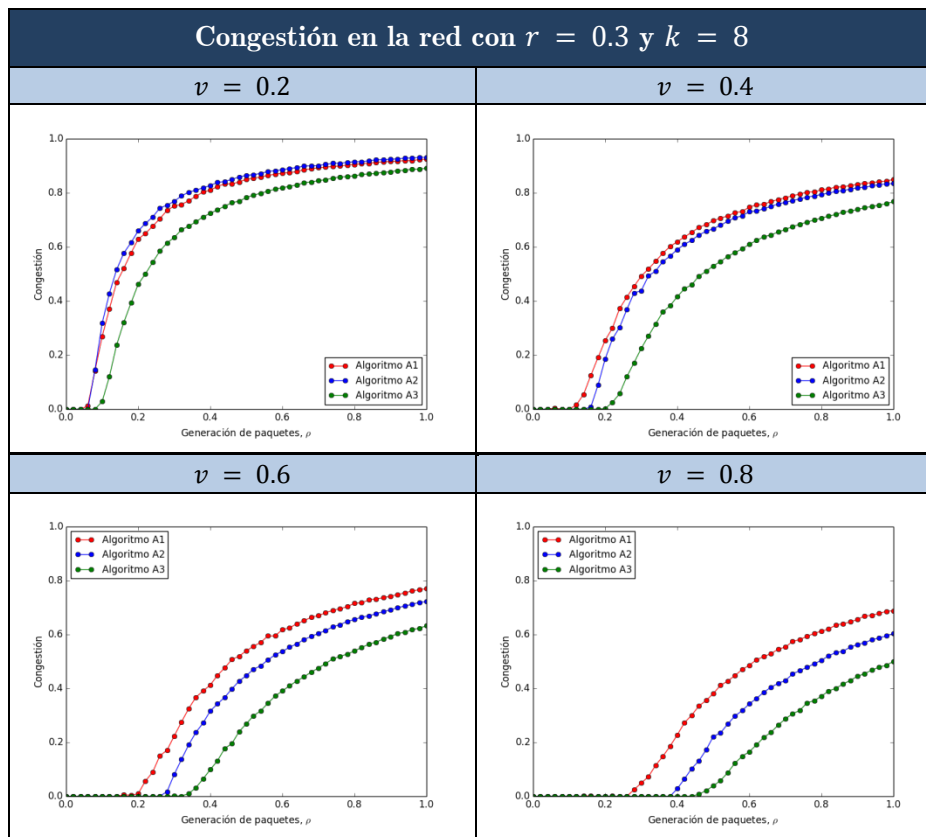


Figura 5.39. Comparativa de los distintos algoritmos de enrutamiento en la evolución de la congestión en la red Watts-Strogatz con $r = 0.3$ y con $k = 8$.

La [figura 5.40](#) muestra los puntos críticos en las redes Watts-Strogatz con $k = 8$, vemos como a diferencia de las redes con $k = 4$, en este caso a partir de valores de $\nu > 0.2$ al algoritmo A3 marca la diferencia, aumentando los puntos críticos en todas las redes, desmarcándose de los otros cuando $\nu \rightarrow 1$, marcando una diferencia con el A1 de hasta 0.30. El algoritmo A2 también es más eficiente cuando $\nu > 0.2$ que el A1, aunque no tanto como el A3, aproximadamente la mitad.

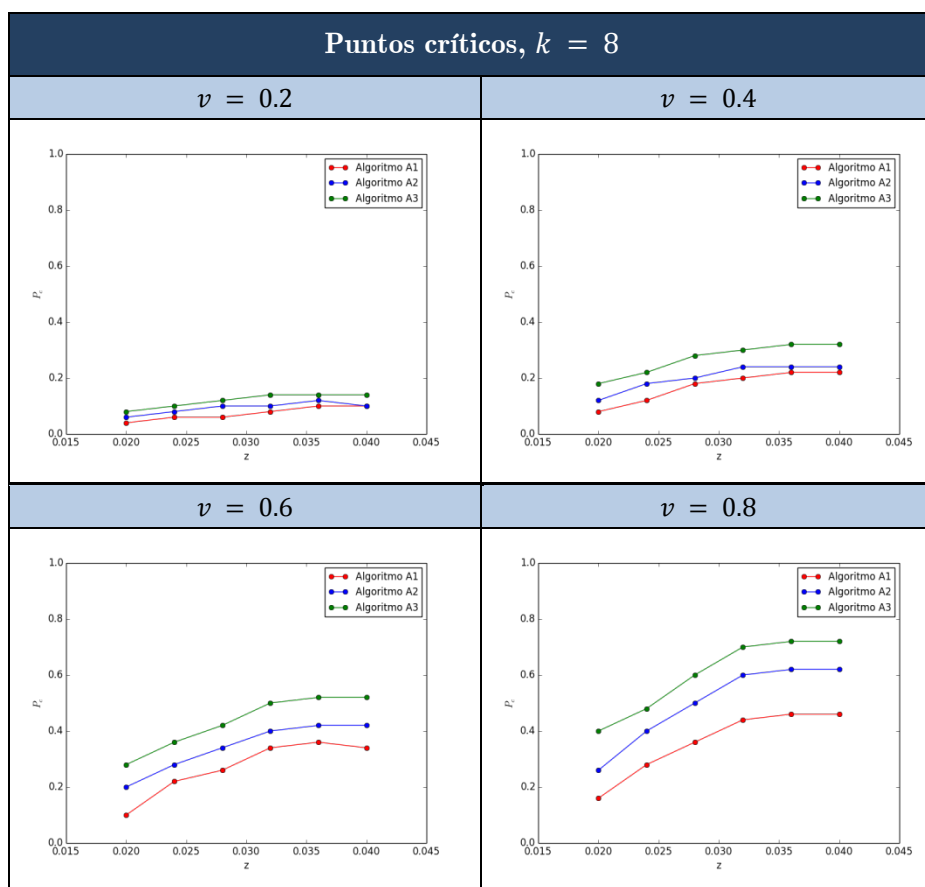


Figura 5.40. Evolución de los puntos críticos en las redes Watts-Strogatz con los distintos algoritmos de enrutamiento, con $k = 8$ y en función de ν .

5.3 Congestión en el Modelo 1.2

En este caso vamos a introducir el parámetro (q_{max}) definido en el [modelo 1.2](#). Este parámetro nos indica el número máximo de pares máximamente entrelazados que habrá como máximo entre los nodos vecinos de la red. Todas las simulaciones tendrás los siguientes parámetros en común: $v_{ini} = 0.2$, $v_{fin} = 0.8$, $w = 4$ y $t = 1000$ pasos, el resto de parámetros se especificará en cada una de las subsecciones. En esta ocasión también se omite la probabilidad $v = 1.0$ dado que con ella, q_{max} no tendría sentido ya que en cada paso de tiempo se regenerarían todos los links, por lo tanto, siempre obtendríamos los mismos resultados que en [5.1](#).

Ya hemos visto en las secciones [5.1](#) y [5.2](#) como se comportan las distintas redes en función de sus parámetros y en función de sus parámetros aplicándole v , por lo tanto en esta sección solo se comprobará como afecta q_{max} a los puntos críticos sobre cada algoritmo de enrutamiento, discutiendo los resultados al final de cada modelo de red. En esta ocasión también se dispondrán los resultados en función de v , para ver cómo se comportan los distintos valores de q_{max} sobre esta.

5.3.1 Redes Erdős–Rényi

5.3.1.1 Algoritmo A1

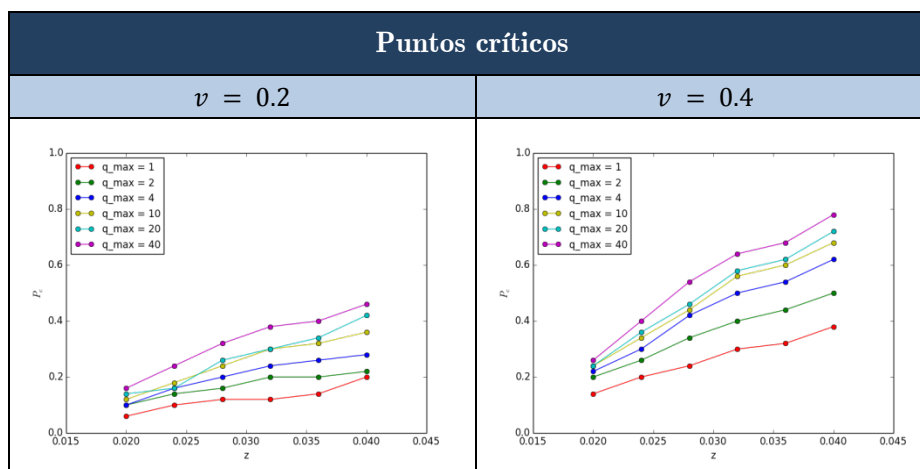


Figura 5.41. Puntos críticos en las redes Erdős–Rényi con diferentes q_{max} sobre el algoritmo A1, en función de la probabilidad de regenerar un enlace $v = 0.2$ y $v = 0.4$.

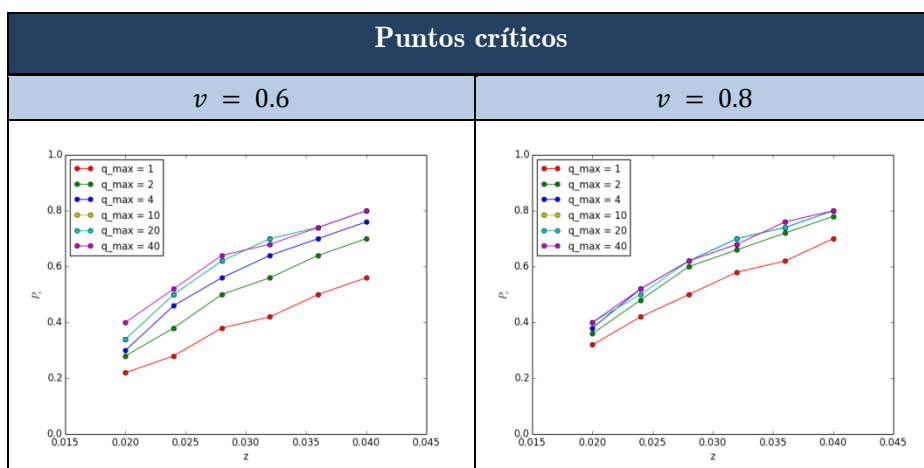


Figura 5.42. Puntos críticos en las redes Erdős-Rényi con diferentes q_{max} sobre el algoritmo A1, en función de la probabilidad de regenerar un enlace $v = 0.6$ y $v = 0.8$.

5.3.1.2 Algoritmo A2

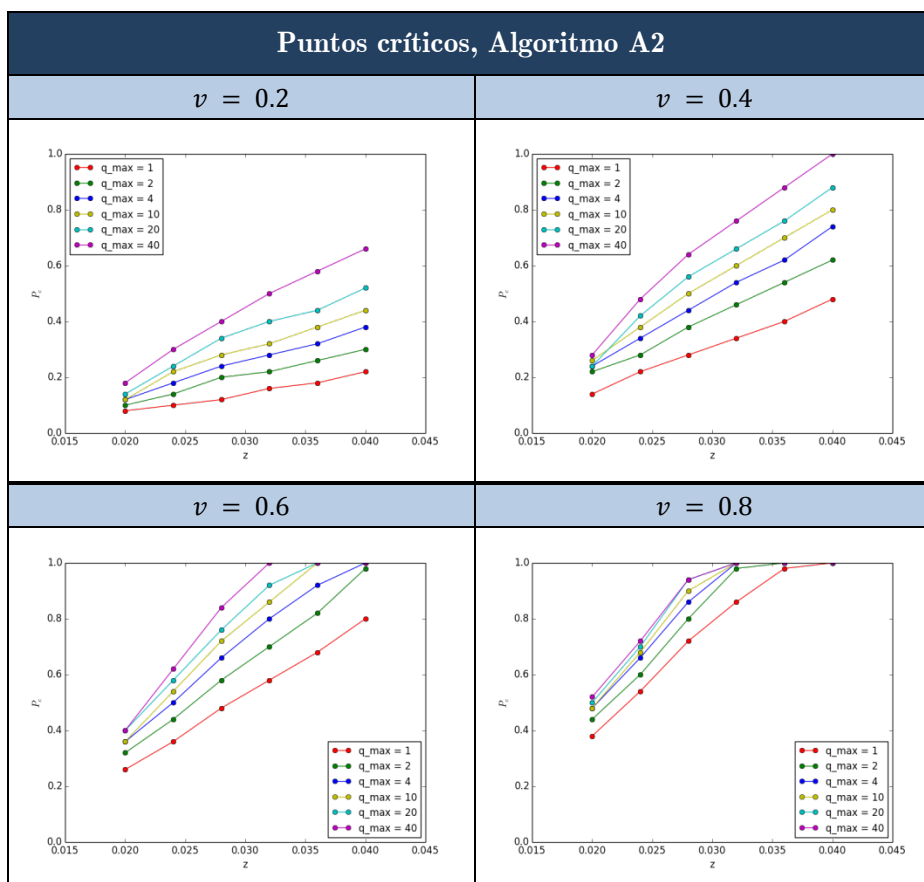


Figura 5.43. Puntos críticos en las redes Erdős-Rényi con diferentes q_{max} sobre el algoritmo A2, en función de la probabilidad de regenerar un enlace v .

5.3.1.3 Algoritmo A3

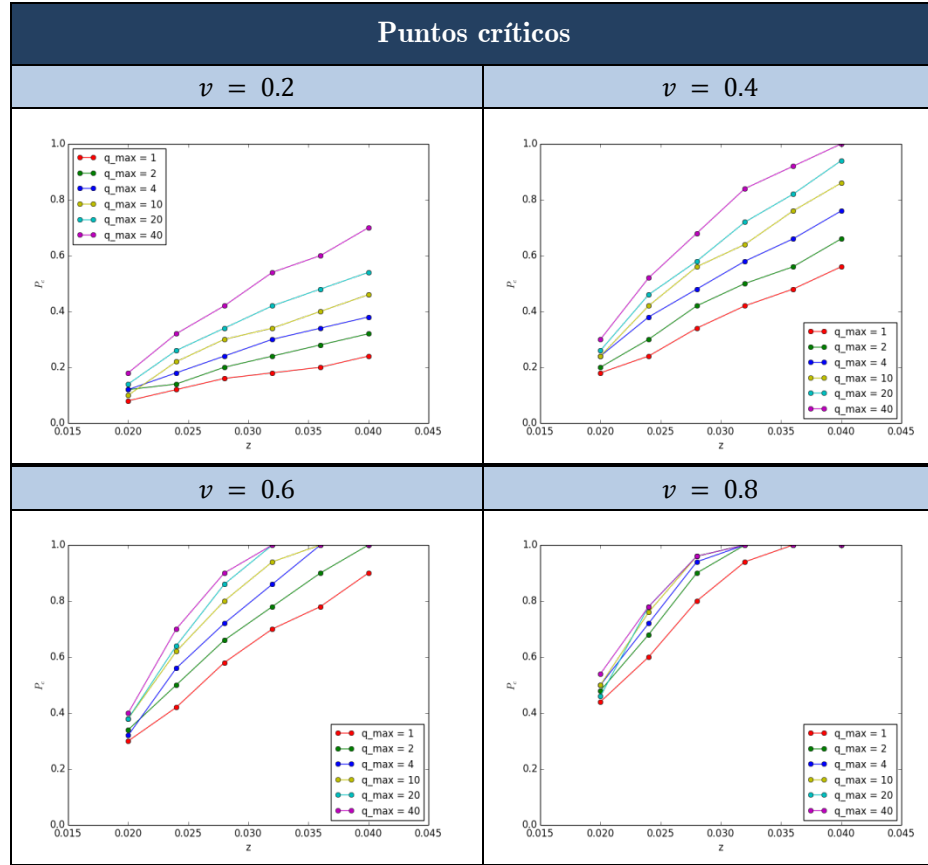


Figura 5.44. Puntos críticos en las redes Erdős–Rényi con diferentes q_{max} sobre el algoritmo A3, en función de la probabilidad de regenerar un enlace v .

5.3.1.4 Discusión

En las distintas figuras de esta sección (5.3.1) podemos ver cómo afecta la variable q_{max} sobre las redes creadas con el modelo Erdős–Rényi. Vemos en todos los algoritmos como esta variable se vuelve más importante, en cuanto a aumentar la zona de flujo libre de la red, cuando $q_{max} \rightarrow 0$. Conforme q_{max} se va aproximando a 1, esta variable se va volviendo cada vez más irrelevante. El crecimiento, como podemos comprobar es lineal en función de z , por lo que cuantos más enlaces tenga una red, mas diferencia habrá entre los distintos puntos críticos.

Si comparamos los tres algoritmos vemos como el algoritmo A1 sigue siendo el menos eficiente. Básicamente la diferencia entre los distintos algoritmos es la misma que la descrita en la sección 5.2.1.4.

5.3.2 Redes Barabási–Albert

5.3.2.1 Algoritmo A1

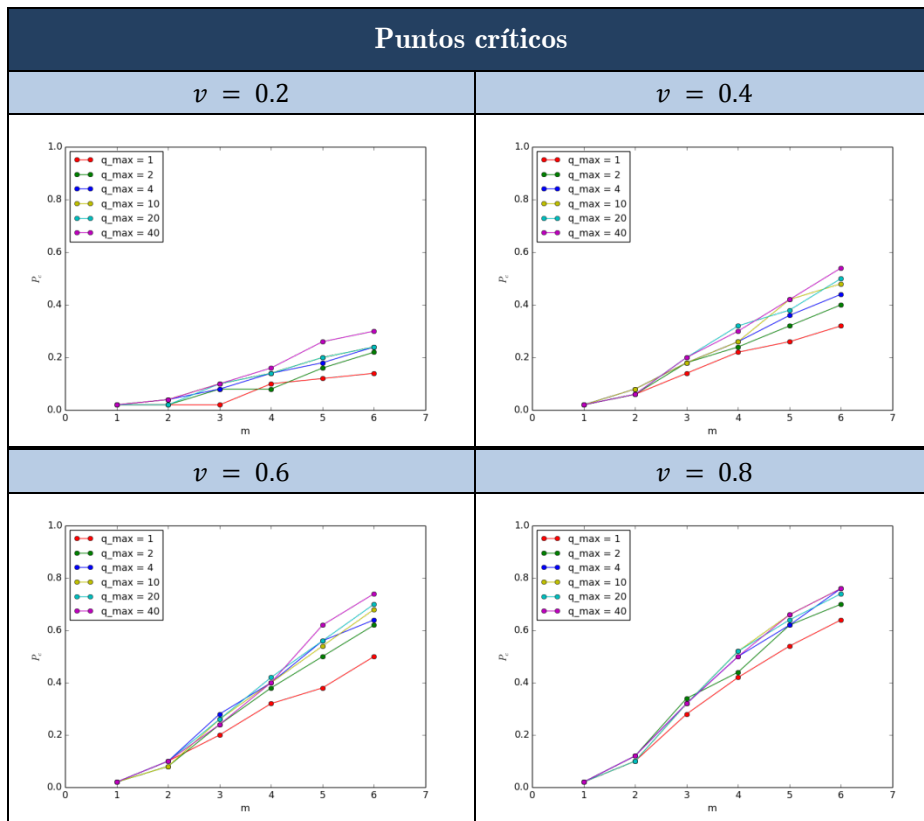


Figura 5.45. Puntos críticos en las redes Barabási–Albert con diferentes q_{max} sobre el algoritmo A1, en función de la probabilidad de regenerar un enlace v .

5.3.2.2 Algoritmo A2

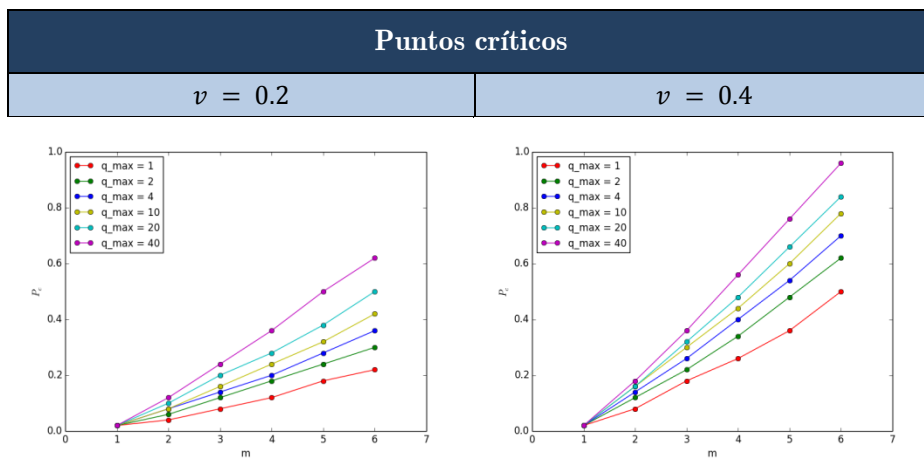


Figura 5.46. Puntos críticos en las redes Barabási–Albert con diferentes q_{max} sobre el algoritmo A2, en función de la probabilidad de regenerar un enlace $v = 0.2$ y $v = 0.4$.

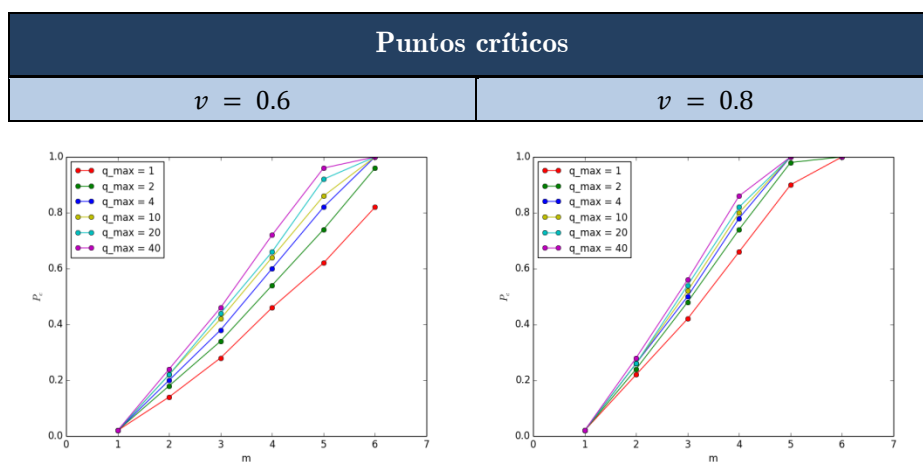


Figura 5.47. Puntos críticos en las redes Barabási–Albert con diferentes q_{max} sobre el algoritmo A2, en función de la probabilidad de regenerar un enlace $v = 0.6$ y $v = 0.8$.

5.3.2.3 Algoritmo A3

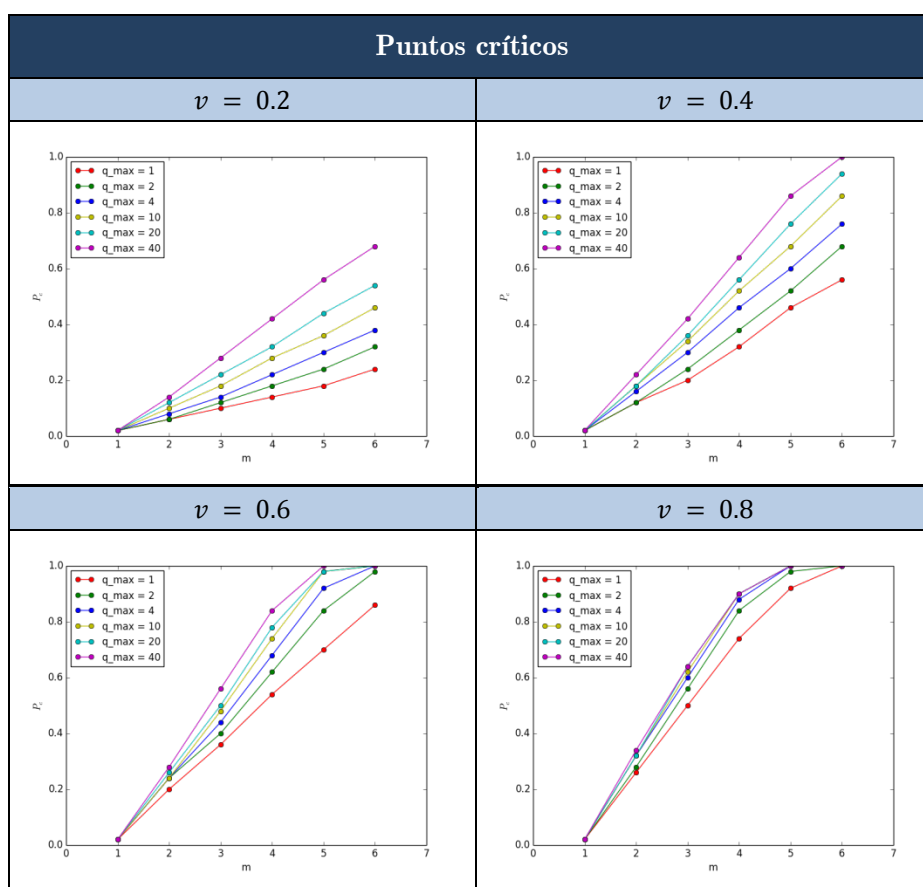


Figura 5.48. Puntos críticos en las redes Barabási–Albert con diferentes q_{max} sobre el algoritmo A3 en función de la probabilidad de regenerar un enlace v .

5.3.2.4 Discusión

En las distintas figuras de esta sección (5.3.2) podemos ver cómo afecta la variable q_{max} sobre las redes creadas con el modelo Barabási–Albert. En este caso, para el algoritmo A1 vemos como solo hay optimización para valores altos de m , para el resto de redes los puntos críticos son los mimos. En el resto de algoritmos, el punto crítico crece de forma lineal, cada vez más en función de m . Igual que antes, cuando $v \rightarrow 0$ esta variable se vuelve más importante, en cuanto a aumentar la zona de flujo libre de la red. Conforme $v \rightarrow 1$, esta variable se va volviendo cada vez más irrelevante.

5.3.3 Redes Watts-Strogatz

5.3.3.1 Algoritmo A1

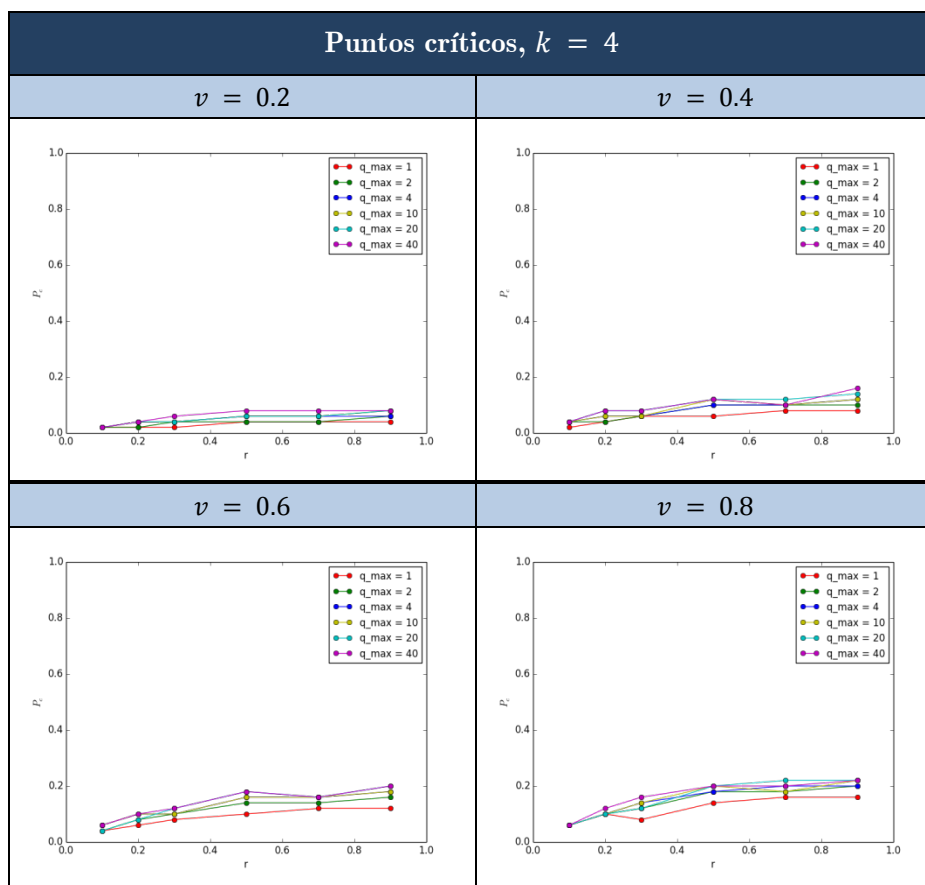


Figura 5.49. Puntos críticos en las redes Watts-Strogatz con $k=4$ sobre el algoritmo A1, en función de la probabilidad de regenerar un enlace v y con diferentes valores de q_{max} .

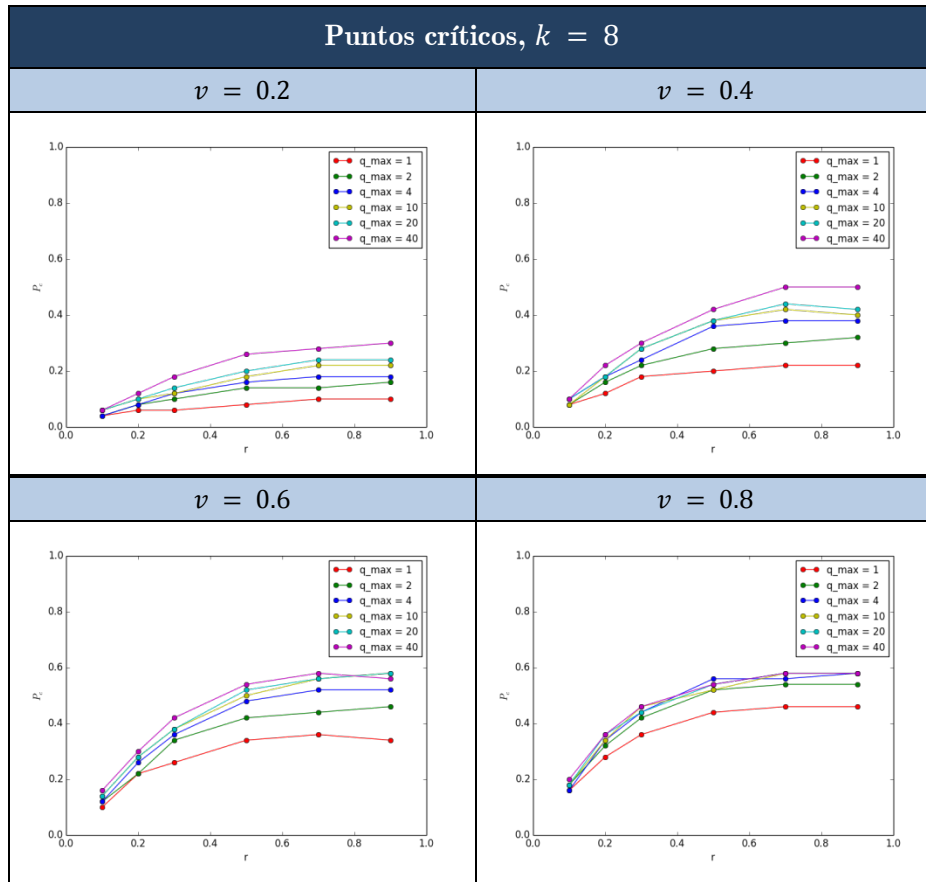


Figura 5.50. Puntos críticos en las redes Watts-Strogatz con $k=8$ sobre el algoritmo A1, en función de la probabilidad de regenerar un enlace v y con diferentes valores de q_{\max} .

5.3.3.2 Algoritmo A2

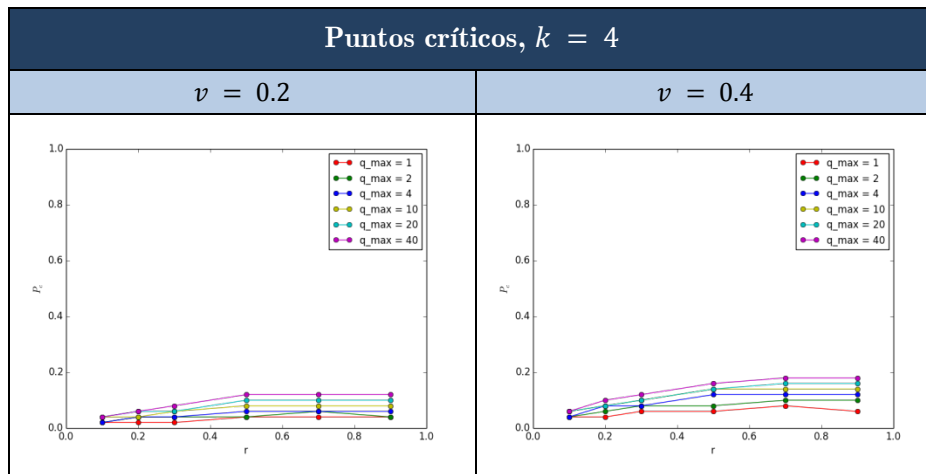


Figura 5.51. Puntos críticos en las redes Watts-Strogatz con $k=4$ sobre el algoritmo A2, en función de la probabilidad de regenerar un enlace $v = 0.2$ y $v = 0.4$ y con diferentes valores de q_{\max} .

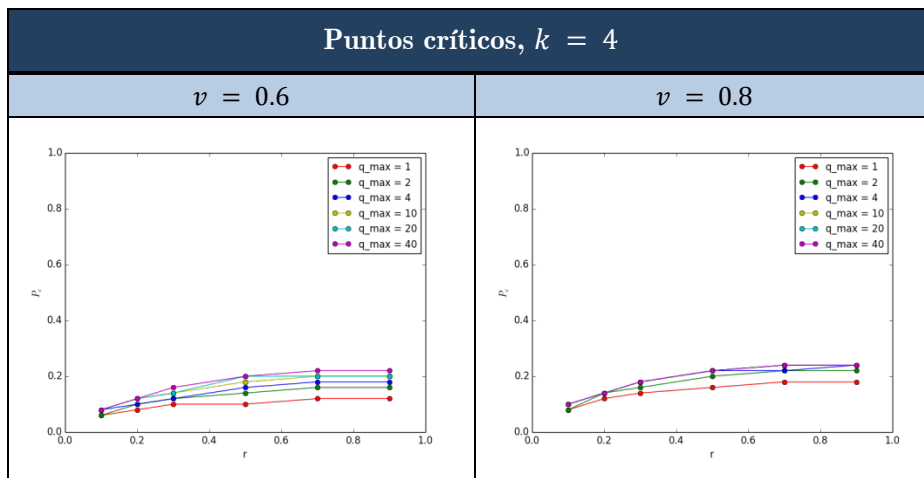


Figura 5.52. Puntos críticos en las redes Watts-Strogatz con $k=4$ sobre el algoritmo A2, en función de la probabilidad de regenerar un enlace $v = 0.6$ y $v = 0.8$ y con diferentes valores de q_{\max} .

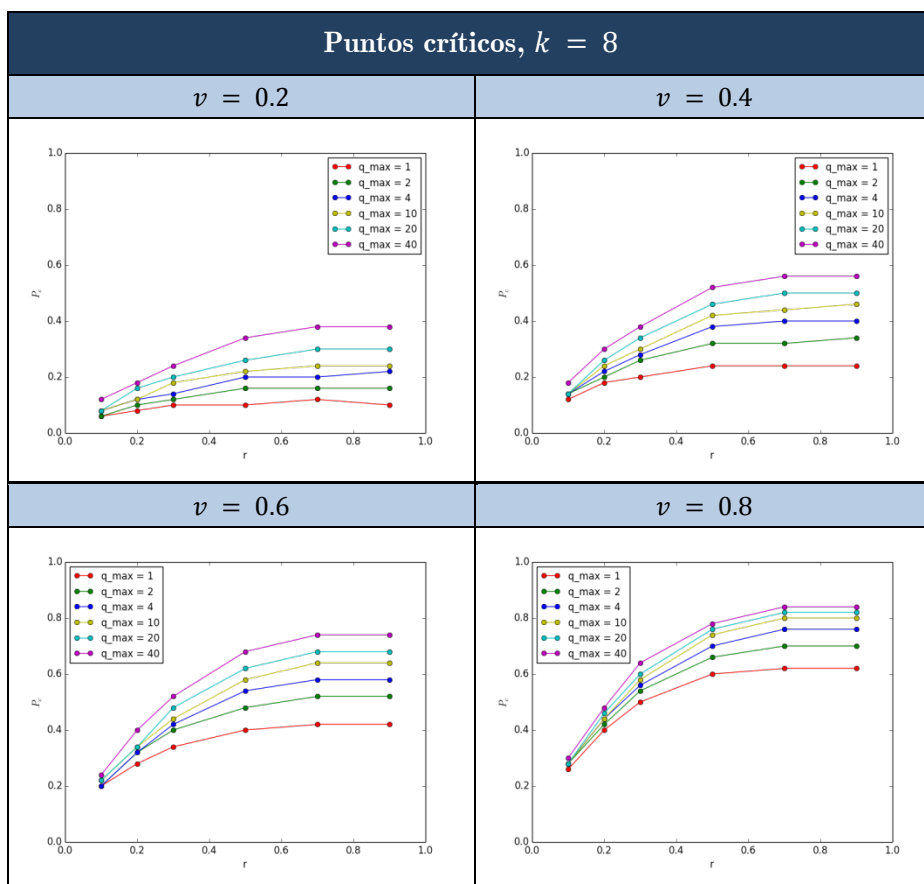


Figura 5.53. Puntos críticos en las redes Watts-Strogatz con $k=8$ sobre el algoritmo A2, en función de la probabilidad de regenerar un enlace v y con diferentes valores de q_{\max} .

5.3.3.3 Algoritmo A3

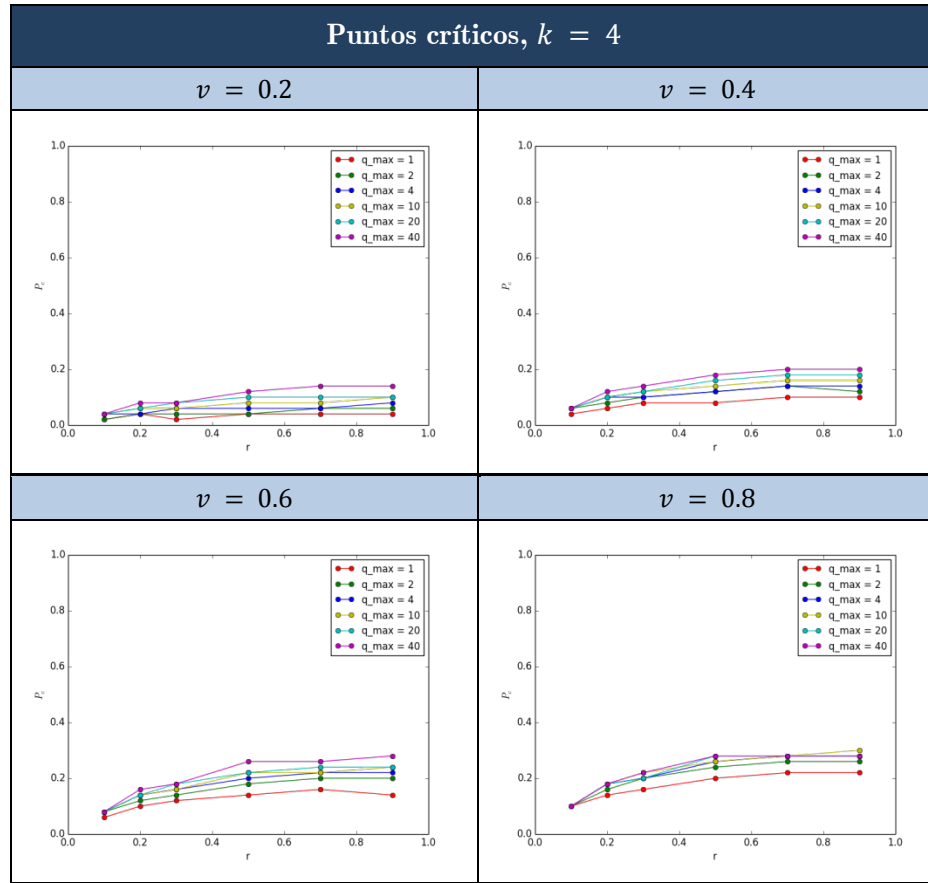


Figura 5.54. Puntos críticos en las redes Watts-Strogatz con $k=4$ sobre el algoritmo A3, en función de la probabilidad de regenerar un enlace $v = 0.6$ y $v = 0.8$ y con diferentes valores de q_{max} .

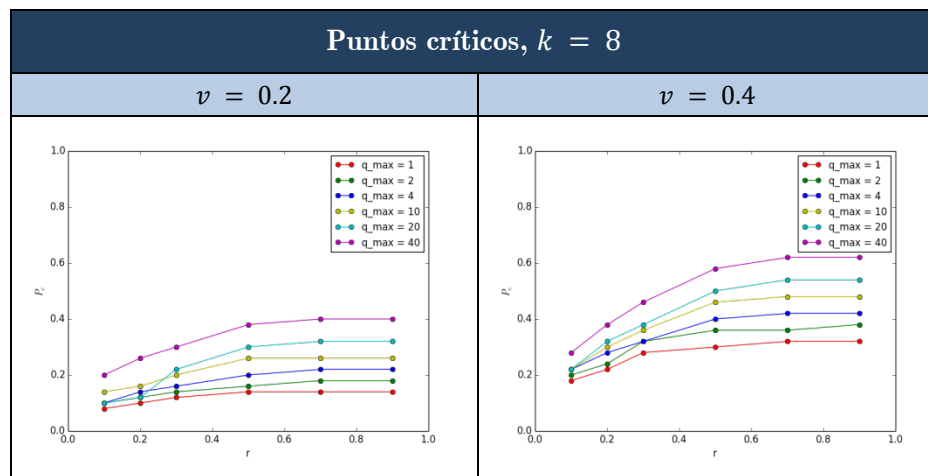


Figura 5.55. Puntos críticos en las redes Watts-Strogatz con $k=8$ sobre el algoritmo A3, en función de la probabilidad de regenerar un enlace $v = 0.2$ y $v = 0.4$ y con diferentes valores de q_{max} .

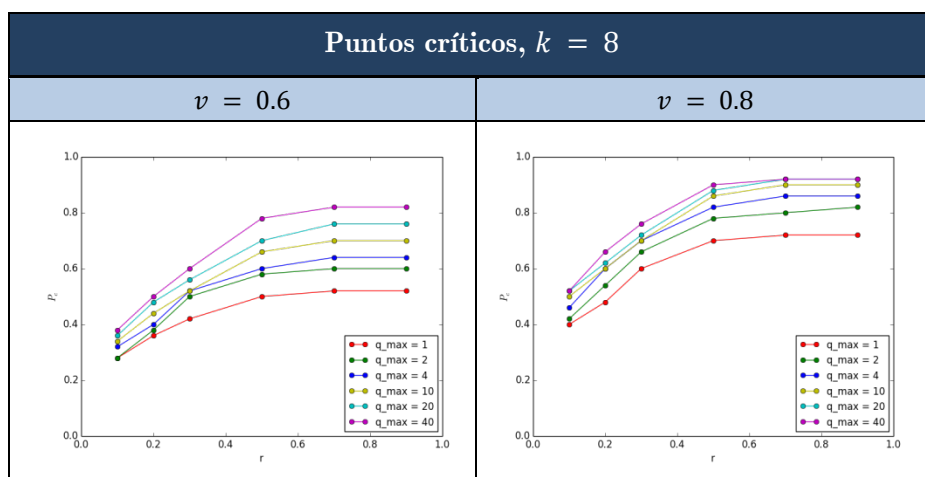


Figura 5.56. Puntos críticos en las redes Watts-Strogatz con $k=8$ sobre el algoritmo A3, en función de la probabilidad de regenerar un enlace $v = 0.6$ y $v = 0.8$ y con diferentes valores de q_{max} .

5.3.3.4 Discusión

En las distintas figuras de esta sección (5.3.3) podemos ver cómo afecta la variable q_{max} sobre las redes creadas con el modelo Watts-Strogatz. En este caso, para las redes con $k = 4$ apenas se produce optimización en ninguno de los algoritmos. De lo contrario, en las redes con $k = 8$, igual que antes, cuando $v \rightarrow 0$ esta variable se vuelve más importante, en cuanto a aumentar la zona de flujo libre de la red, esta optimización es proporcional a q_{max} . En este caso cuando $v \rightarrow 1$, la variable aún tiene sus efectos, pero, obviamente, cuando llega a 1 no tendrá ninguno.

Capítulo 6

Conclusiones

El estudio de la congestión y de los algoritmos de enrutamiento sobre redes cuánticas es un tema muy novedoso y que nunca se había hecho antes. Esto se debe a que la mayor parte de estas redes están implementadas, aún, a pequeña escala y con muy pocos nodos, por lo que todas las transmisiones que se realizan están controladas para garantizar el buen funcionamiento de la red. Actualmente la aplicación principal de este tipo de redes es la distribución de claves simétricas aprovechando las leyes que nos ofrece la mecánica cuántica, aunque no se descarta en un futuro la transmisión de paquetes de datos con otro tipo de información.

Debido a que no existen trabajos relacionados específicamente a este campo, en esta tesis hemos tratado de dar una visión global del estado del arte de los tres temas principales que engloba nuestro trabajo; la congestión y dinámica en las redes, las redes cuánticas y la distribución de claves cuántica. En segundo lugar hemos diseñado un modelo cuyo propósito principal es el envío de información punto a punto de forma segura, aprovechando características cuánticas como son el entrelazamiento y la teleportación. Evidentemente hemos obviado algunos aspectos tecnológicos en cuanto a las redes cuánticas, ya que algunos de ellos se encuentran en fase experimental, como podría ser el caso de los repetidores cuánticos, memorias cuánticas, etc... A pesar de esto, día a día se van haciendo grandes avances en este campo, por lo que nuestro trabajo pretende hacer uso de las últimas innovaciones y descubrimientos.

Hemos visto en el [capítulo 3](#) como la congestión y la dinámica en las redes es un tema ampliamente estudiado en las últimas décadas, muchos investigadores han propuesto algoritmos de enrutamiento para tratar de hacer las redes más eficientes en cuanto al envío de paquetes y así reducir la congestión, desgraciadamente, muchos de estos algoritmos se basan en redistribuir la carga de información entre los distintos nodos de la red por lo que no son aplicables a nuestro modelo ya que se perdería la esencia principal: el envío de información punto a punto. Debido a esto, en la sección [4.4](#) hemos planteado tres algoritmos de enrutamiento que mantenga este principio general y así poder estudiar la congestión en las redes cuánticas. El primero de ellos se basa en la utilización de los caminos mínimos entre los nodos de la red, el segundo, además de utilizar caminos mínimos, también utiliza caminos alternativos más largos. El tercero es un algoritmo centralizado que utiliza métricas de red para controlar el tráfico y decidir el orden de envío de paquetes. Diremos que un algoritmo es eficiente cuando puede realizar el mayor número de envíos paralelos posibles en cada paso de tiempo.

Estudios han demostrado que muchas de las redes que nos rodean son de índole compleja, por lo que este trabajo se ha centrado en el estudio de este tipo de redes. Para hacerlo se han utilizado los tres modelos de red más comunes; el Erdős–Rényi (redes aleatorias), el Barabási–Albert (redes libres de escala) y el modelo Watts-Strogatz (Small-World), para estudiar un total de 42 redes distintas.

Los resultados nos muestran como un algoritmo que considere, además de los caminos mínimos, otros caminos alternativos más largos (algoritmos [A2](#) y [A3](#)), es mucho más eficiente en cualquiera de las redes definidas siempre y cuando los enlaces entrelazados se regeneren en cada paso de tiempo, tal y como hemos definido en el [modelo 1](#). Además, con el algoritmo [A3](#) también se demuestra como un algoritmo centralizado que además de considerar caminos alternativos a los mínimos, también utilice métricas de red, como por ejemplo (y en nuestro caso) la intermediación de los enlaces, resulta ser mucho más eficiente que cualquier otro. En definitiva, el control del tráfico mejora el rendimiento global, ampliando la zona de flujo libre, aunque a veces ha introducido una fase congestionada fija muy baja.

Dado que un sistema real difícilmente va a ser perfecto, decidimos considerar la regeneración de los enlaces entrelazados de forma probabilística planteando el [modelo 1.1](#), donde añadimos un parámetro de probabilidad v . Teniendo en cuenta

este parámetro hemos comprobado como aumenta la congestión, reduciendo considerablemente la zona de flujo libre cuando $v \rightarrow 0$. Además, también cuando $v \rightarrow 0$, hemos podido comprobar como en todos los modelos de red el algoritmo A2 es menos eficiente que el A1, por lo que considerar caminos alternativos a los mínimos no es tan buena idea cuando hay pocos enlaces disponibles. En todo caso, un algoritmo centralizado que controle el tráfico de la red (algoritmo A3) es el más eficiente sea cual sea esta probabilidad de regeneración de un enlace, aunque para valores bajos de v ($v \leq 0.2$) la mejora sea bastante reducida.

Aquí es donde entra el tercer y último parámetro que consideramos en nuestro modelo: q_{max} , definiendo lo que es el [modelo 1.2](#). Este parámetro indica el número máximo de pares máximamente entrelazados que puede haber entre dos nodos vecinos (hasta ahora habíamos considerado que este parámetro era 1). Con este parámetro se han realizado alrededor de 150 simulaciones entre todas las redes, y los resultados nos muestran que es bueno tener más enlaces entre vecinos cuando $v \rightarrow 0$. Esto lo que hace es que durante q_{max} pasos de tiempo como mínimo la red funcione como si $v = 1$, aumentando, de esta forma, la zona de flujo libre. En cambio, a media que $v \rightarrow 1$, este parámetro se vuelve prácticamente irrelevante (no es necesario tener 40 enlaces si en cada paso de tiempo se regeneran todos y solo puedo utilizar 1).

Hemos visto en [estado del arte](#) que, en las redes convencionales, las redes con forma de estrella son poco eficientes para realizar muchas operaciones en paralelo debido a que el nodo central produce mucha congestión, y que las redes que mejor evitan la congestión son aquellas en las que todos los nodos tienen la misma distribución de grado, intermediación, etc. En nuestro caso, ya que la congestión viene dada por la falta de enlaces y no por la sobrecarga de un nodo en concreto, una red de estrella no sería la peor para un número alto de operaciones en paralelo.

Después de analizar los datos se ha visto que dada una red con la el mismo número de nodos y el mismo número de enlaces, la estructura más eficiente es una red libre de escala, siempre y cuando el grado de sus nodos sea más grande que 1, seguida por una red aleatoria, siendo la Small-World la menos eficiente.

La peor estructura de todas es una red libre de escala donde la mayoría de nodos tienen grado 1, ya que su estructura estaría muy jerarquizada provocando

que enviar un paquete a un nodo lejano desconecte gran parte de la red. A esta red le seguirían, como peores redes en cuanto a evitar la congestión, todas aquellas redes Small-World con pocos vecinos ($k < 6$).

En general las redes aleatorias y las libres de escala son las más óptimas en cuanto a reducir la congestión en las redes cuánticas. Como la congestión viene dada por la falta de enlaces, cuantos más enlaces tengan estas redes, mejor, aunque en las redes reales esto no es siempre posible.

Para un trabajo futuro quedaría analizar más algoritmos de enrutamiento, talvez algunos que tengan en cuenta otras métricas de la red, además de la intermediación, y estudiar si sería posible aumentar aún más la zona de flujo libre. También sería interesante añadir otras modificaciones al modelo, como por ejemplo limitar el número de saltos o añadir cola con prioridades en los nodos. El algoritmo A3 es el más eficiente de todos, pero cabría comprobar si debido al control de tráfico, este algoritmo genera congestión en un nodo en concreto debido a un camino muy largo que nunca puede resolverse.

Bibliografía

- [1] **The SECOQC quantum key distribution network in Vienna**, M Peev et al., New J. Phys. 11 075001 (2009).
- [2] **Distribution of entanglement in large-scale quantum networks**, S.Perseguers, G.J. Lapeyre, D. Cavalcanti, M. Lewenstein and A. Ací, Rep.Prog.Phys. 76 096001 (2013).
- [3] **“Event-ready-detectors” Bell experiment via entanglement swapping**, M. Żukowski, A. Zeilinger, M. A. Horne, and A. K. Ekert, Phys. Rev. Lett. 71, 4287 (1993).
- [4] **Entanglement Between Photons that have Never Coexisted**, E. Megidish, A. Halevy, T. Shacham, T. Dvir, L. Dovrat, and H. S. Eisenberg, Phys. Rev. Lett. 110, 210403 (2013).
- [5] **Quantum Cryptography**, Nicolas Gisin, Grégoire Ribordy, Wolfgang Tittel, Hugo Zbinden, Rev. Mod. Phys. 74,145 (2002).
- [6] **Teleporting an unknown quantum state via dual classical and Einstein–Podolsky–Rosen channels**, Charles H. Bennett, Gilles Brassard, Claude Crépeau, Richard Jozsa, Asher Peres, and William K. Wootters, Rev. Lett. 70 1895–9 (1993).
- [7] **Quantum cryptography with and without entanglement**, Nicolas Gisin, Nicolas Brunner (2003).
- [8] **High-fidelity transmission of polarization encoded qubits from an entangled source over 100 km of fiber**, H. Hubel, M. R. Vanner, T. Lederer,

- B.Blauensteiner, T. Lorunser, A. Poppe, and A. Zeilinger. *Opt. Express*, 15(12):7853–7862 (2008).
- [9] **Graph Theory**, V. K Balakrishnan. McGraw-Hill. ISBN 0-07-005489-4 (1997).
- [10] **Centrality in valued graphs: A measure of betweenness based on network flow**, Linton C. Freeman , Stephen P. Borgatti , Douglas R. White (1991).
- [11] **A measure of betweenness centrality based on random walks**, M. E. J. Newman (2003).
- [12] **Justification and Application of Eigenvector Centrality**, Leo Spizzirri (2011).
- [13] **The structure and function of complex networks**, M. E. J. Newman (2003).
- [14] **Structure and traffic on complex networks**, Jordi Duch (2008).
- [15] **Complex Networks: Structure, Robustness and Function**, R. Cohen, S. Havlin, Cambridge University Press. (2010).
- [16] **On Random Graphs. I.**, P. Erdős; A. Rényi, (1959).
- [17] **On the Evolution of Random Graphs**, P. Erdős, A Rényi (1960).
- [18] **On the strength of connectedness of a random graph**, P. Erdős, A Rényi (1961).
- [19] **Emergence of Scaling in Random Networks**, Albert-László Barabási, Réka Albert (1999).
- [20] **Collective dynamics of 'small-world' networks**, Duncan J. Watts & Steven H. Strogatz (1998).
- [21] **Network Routing: Algorithms, Protocols, and Architectures**, Medhi, Deepankar and Ramasamy, Karthikeyan (2007)
- [22] **Routing Basics**, Cisco.
- [23] **Dynamic Routing Protocols**, CISCO Press (2001).
- [24] **On Self Adaptive Routing in Dynamic Environments**, Haiyong Xie, Lili Qiu, Yang Richard, Yang and Yin Zhang (2004).

-
- [25] **Introduction to Algorithms**, Thomas H. Cormen , Charles E. Leiserson , Ronald L. Rivest , Clifford Stein (2009).
- [26] **On a routing problem**, Richard Bellman (1958).
- [27] **A Note on Two Problems in Connexion with Graphs**, E. W Dijkstra (1959).
- [28] **Optimal traffic networks**, Marc Barthélemy and Alessandro Flammini (2006).
- [29] **Congestion in different topologies of traffic networks**, J. J. Wu¹, Z. Y. Gao¹, H. J. Sun, and H. J. Huang (2006).
- [30] **Traffic Dynamics on Complex Networks: A Survey**, Shengyong Chen, Wei Huang, Carlo Cattani and Giuseppe Altieri (2011).
- [31] **Quantum encoding**, Benjamin Schumacher, Phys. Rev. A 51, 2738 (1995).
- [32] **Discussion of Probability Relations between Separated Systems**, E. Schrödinger (1935).
- [33] **Quantum Entanglement and Information**, Stanford EOPH (2001).
- [34] **Physical Qubits**, C/CS/Phys C191, Berkeley (2009).
- [35] **Teleportation**, C/CS/Phys C191, Berkeley (2009).
- [36] **Explorations in Quantum Computing, Chapter 2: Quantum gates**, Colin P. Williams, ed. Springer (2011).
- [37] **Optimal Network Topologies for Local Search with Congestion**, R. Guimerà, A. Díaz-Guilera, F. Vega-Redondo, A. Cabrales, and A. Arenas. (2002).
- [38] **Congestion and centrality in traffic flow on complex networks**, P. Holme (2003).
- [39] **Onset of traffic congestion in complex networks**, Liang Zhao, Ying-Cheng Lai, Kwangho Park, and Nong Ye (2005).
- [40] **Congestion phenomena on complex networks**, D. Martino et al. (2009).
- [41] **Efficient routing on complex networks**, Gang Yan, Tao Zhou, Bo Hu, Zhong-Qian Fu, and Bing-Hong Wang (2006).

- [42] **Dynamics of jamming transitions in complex networks**, P. Echenique, J. Gómez-Gardeñes and Y. Moreno (2005).
- [43] **Quantum Cryptography: public key distribution and coin tossing**, Charles Bennett and Gilles Brassard. (1984).
- [44] **Quantum Cryptography Based on Bell's theorem**, Artur K. Ekert (1991).
- [45] **The Security of Practical Quantum Key Distribution**, Valerio Scarani, Helle Bechmann-Pasquinucci , Nicolas J. Cerf , Miloslav Dusek, N. Lutkenhaus, Momtchil Peev (2009).
- [46] [QuReP, Quantum Repeaters for long distance Fibre-Based Quantum Communication.](#)
- [47] **Quantum repeaters: the role of imperfect local operations in quantum communication.** H. J. Briegel, W. Dur, J. I. Cirac & P. Zoller, Phys. Rev. Lett. 89, 5932–5935 (1998).
- [48] **High rate, long-distance quantum key distribution over 250 km of ultra low loss fibres**, D. Stucki, N. Walenta, F. Vannel, R. T. Thew, N. Gisin, S Gray, H. Zbinden, C. R. Towery and S. Ten, New J. Phys. 11 075003 (2009).
- [49] **The DARPA Quantum Network**, Chip Elliott (2004).
- [50] [BBN Technologies.](#)
- [51] [SwissQuantum.](#)
- [52] **Security of Trusted Repeater Quantum Key Distribution Networks**, Louis Salvail, Momtchil Peev, Eleni Diamanti, Romain Alleaume, Norbert Lutkenhaus, Thomas Laenger (2009).
- [53] **Quantum Memories A Review based on the European Integrated Project “Qubit Applications (QAP)”**, Eur. Phys. J. D 58, 1-22 (2010).
- [54] **Teleportation of entanglement over 143 km**, Thomas Herbst, Thomas Scheidl, Matthias Fink, Johannes Handsteiner, Bernhard Wittmann, Rupert Ursin, and Anton Zeilinger, (2014).

-
- [55] **Quantum teleportation from a telecom-wavelength photon to a solid-state quantum memory**, F. Bussières, C. Clausen, A. Tiranov, B. Korzh, Varun B. Verma, S. Woo Nam, F. Marsili, A. Ferrier, P. Goldner, H. Herrmann, C. Silberhorn, W. Sohler, M. Afzelius, y N. Gisin, (2014).
- [56] **Room-Temperature Quantum Bit Storage Exceeding 39 Minutes Using Ionized Donors in Silicon-28**, K. Saeedi, S. Simmons, Jeff Z. Salvail, Phillip Dluhy, H. Riemann, Nikolai V. Abrosimov, P. Becker, Hans-Joachim Pohl, John J. L. Morton, Mike L. W. Thewalt, (2014).



Congestión y enrutamiento en redes cuánticas by [Sampé Domènech, Josep Duch Gavalrà, Jordi](#) is licensed under a [Creative Commons Reconocimiento-NoComercial-SinObraDerivada 4.0 Internacional License](#).

Puede hallar permisos más allá de los concedidos con esta licencia en <http://creativecommons.org/licenses/by-nc-nd/4.0/deed.ca>