

ALGORITMO DE ENRUTAMIENTO ATENTO A LA ENERGÍA PARA REDES INTRACHIP

ENERGY-AWARE ROUTING ALGORITHM FOR NETWORK ON CHIP

Arley Villa Salazar

Universidad de Antioquia, Colombia
arley.villa@udea.edu.co

Gustavo Patiño

Universidad de Antioquia, Colombia
adolfo.patino@udea.edu.co

Recepción: 22/octubre/2019

Aceptación: 23/noviembre/2019

Resumen

Las redes intrachip (*Network on Chip, NoC*) han sido reconocidas como una solución viable para resolver los desafíos del diseño de sistemas en chip (*Systems on Chip, SoC*), proporcionando una estructura de comunicación escalable y eficiente para un sistema multiprocesador. Sin embargo, esta solución tiene un alto consumo de energía. Para minimizar dicho consumo, es esencial diseñar un algoritmo de enrutamiento eficiente que permita la administración de energía mientras se maximiza el rendimiento.

Este documento propone un algoritmo de enrutamiento atento a la energía (llamado *EA-NoC*) que permite optimizar la energía dinámica al determinar la ruta óptima entre origen y destino, evitando el consumo innecesario. Los algoritmos de enrutamiento se simulan en una topología de malla 2D y se comparan con los algoritmos de enrutamiento implementados en el simulador Noxim. Los resultados experimentales muestran que el algoritmo propuesto, mejora en un 28% en promedio el consumo de potencia dinámica en comparación con los algoritmos existentes en la literatura.

Palabras Claves: Algoritmo de Enrutamiento, atento a la energía, multiprocesador, redes Intrachip, reducción de energía.

Abstract

Networks on Chip (NoC) have been recognized as a viable solution to solve the challenges of designing Systems on Chip (SoC), providing a scalable and efficient communication structure for multicore systems. However, this solution has high energy consumption. In order to minimize such consumption, it is essential to design an efficient routing algorithm which allows power management while maximizing throughput.

This paper proposes an energy aware routing algorithm (called EA-NoC) that allows optimizing this resource by determining the optimal route between source and destination, avoiding unnecessary energy consumption. The routing algorithms are simulated on a 2D-mesh topology, and compared to the State-of-the-Art routing algorithms implemented in the Noxim simulator. The experimental results show that the proposed algorithm performs 28 % better on average, in terms of dynamic power, compared to the existing algorithms in the literature.

Keywords: *Energy-aware, MPSoC, network on chip, reduction energy, routing algorithm.*

1. Introducción

La tendencia actual en los sistemas embebidos es aumentar el número de tareas, el uso de memoria, y la presencia de elementos de entrada y salida de datos, esto exige que se dispongan de sistemas computacionales con un poder de procesamiento cada vez mayor. Por esto, surgen los sistemas integrados de múltiples procesadores (*Multiprocessor System on Chip, MPSoC*) que permiten garantizar el aumento en el poder de procesamiento, dado que estos permiten dividir los costos de computación de aplicaciones entre los elementos que lo conforman [Borkar, 2007]. Esta tendencia parece no acabar y cada vez se aumenta el número de elementos de procesamiento, pasando en la última década de unos pocos a cientos [Nickolls, 2010], debido a esto se hace más compleja y difícil la implementación de las interconexiones directas y los buses compartidos requeridos para la comunicación de los dispositivos al interior de un *MPSoC*. El problema radica en que dichas interconexiones no son escalables y se vuelven ineficientes con el

aumento en el número de elementos de procesamiento [Tsai, 2012], debido al alto número de dispositivos que quieren acceder al bus de comunicaciones, lo que lleva a que el arbitraje de este ralentice la comunicación entre los elementos de la red, generando un consumo de potencia innecesario.

Para solucionar esta problemática surgen las redes *NoC*, orientadas a la conexión de dispositivos como memorias, registros, caché y procesadores al interior de un *MPSoC* [Benini, 2002]. Dichas redes pueden conectar cientos de dispositivos con una distancia de interconexión muy baja, permitiendo contrarrestar las limitaciones que presentan los *MPSoC* en arquitecturas basadas en buses de datos, tales como el tiempo de diseño y la baja escalabilidad. Es así que, en la última década, las *NoC* se han convertido en una solución flexible y de alto rendimiento para el problema de la interconexión de dispositivos al interior de un *MPSoC* [De Micheli, 2017]. Las características más importantes de un *NoC* son:

- La topología determina el orden físico y la disposición de los enlaces que conectan los nodos de la red.
- El control de flujo asigna recursos de comunicación para la transmisión segura y confiable de paquetes entre los núcleos.
- El Núcleo es el elemento de procesamiento.
- El algoritmo de enrutamiento determina la ruta mínima o menos congestionada que deben seguir los paquetes para evitar el consumo innecesario de energía [Patooghy, 2006].

Existen varios métodos y formas de clasificarlos. Una forma de clasificarlos es considerar donde se toman las decisiones de enrutamiento, pueden ser de origen (*Source Routing*, SR) o distribuido (*Distributed Routing*, DR).

En el enrutamiento de origen (SR), se calcula previamente la información básica sobre ruta a seguir en el *router* donde se generan los datos. En el enrutamiento distribuido (DR) cada *router* recibe el paquete y define la ruta a enviar [Mubeen, 2009].

Aunque las *NoC* resuelven en parte los problemas generados con el aumento de los núcleos, esta solución tiene un alto costo energético. Estudios previos han

estimado que entre el 10% y el 40% del consumo total de energía de un MPSoC se debe a la NoC [Bell, 2008], [Zhan, 2015].

Este problema ha adquirido especial relevancia en la fabricación de MPSoC debido a su impacto en factores físicos, económicos y ambientales. Por lo tanto, lograr una reducción efectiva del consumo de energía es un objetivo en este tipo de sistemas. La ecologización de las tecnologías y protocolos de red conocida como *Green Networking* [Chiaraviglio, 2015], [Bolla, 2014] surgió para abordar esta preocupación y tiene una fuerte influencia en los diseños de electrónica y específicamente en el campo de las redes de datos.

Este documento propone un algoritmo de enrutamiento donde la energía se optimiza aplicando un proceso de enrutamiento periódico, dado que este, toma la decisión al interior del MPSoC de cómo se enrutan los paquetes a través de los *routers*, decidiendo cual es la ruta óptima, evitando así el consumo innecesario de energía. Por lo tanto, la ruta seguida por el paquete (*flit*) jugará un papel clave en la determinación del consumo de energía de la NoC.

Trabajos Relacionados

[Wu, 2017], propone CRA, un novedoso diseño multinúcleo NoC (*Multiple Network-on-Chip*) con algoritmos de enrutamiento distintos para diferentes subredes, integrando una política de programación de paquetes y energía que detecta la congestión. CRA puede lograr un bajo consumo de energía sin degradar el rendimiento al variar la utilización de la red.

[Das, 2017], propone una metodología de diseño predictivo inspirada en el aprendizaje automático para arquitecturas de múltiples núcleos, confiable y energéticamente eficiente, habilitada por la integración 3D. Para esto, plantea un algoritmo de asignación de enlace vertical de repuesto (*spare-vertical link, sVL*) computacionalmente eficiente basado en una formulación de búsqueda de espacio de estado. Los resultados muestran que el algoritmo de asignación de sVL propuesto puede mejorar significativamente la confiabilidad, así como la vida útil.

[Kullu, 2019], propone un método basado en algoritmos genéticos, que genera una población inicial basada en la topología anillo y produce mejoras en topologías

irregulares en términos de consumo de energía a través de operadores genéticos. La función objetivo del método propuesto es minimizar el consumo de energía resultante de la comunicación de la red.

[Rahaman, 2019], propone un algoritmo de enrutamiento tolerante a fallos, adaptable y sin interbloqueo para NoC denominado *F-Route-NoC-Mesh* (FRNM), el cual ignora cualquier canal virtual en regiones convexas ortogonales, equilibrando el tráfico de la red asumiendo un límite de bloque virtual defectuoso y enrutando paquetes a través de este límite virtual. Utilizando el algoritmo propuesto, se desarrolla un enfoque basado en un modelo de bloque de fallas, obteniendo mejoras de la latencia, rendimiento y el consumo de energía.

Kullu et al., proponen soluciones basadas en algoritmos genéticos, si bien estos han demostrado ser efectivos en el enrutamiento estático en sistemas diseñados para realizar un conjunto fijo de tareas o donde se conoce el patrón de comunicación, estos son costosos en términos de tiempo y recursos, además se pueden presentar casos en los cuales dependiendo de los parámetros que se utilicen para la evaluación, el algoritmo podría no llegar a converger en una solución óptima.

En [Rahaman, 2019], proponen un algoritmo de enrutamiento de tipo distribuido por lo que la decisión de enrutamiento se toma según la información del encabezado, esto genera un consumo de energía alto debido al procesamiento adicional en cada uno de los núcleos, por lo que las soluciones de este tipo implican incremento en el consumo de la NoC.

Wu y Das et al., proponen soluciones que implican modificaciones físicas de los componentes de la red, ya sea en la forma de conexión de los enlaces o en el tipo de enrutadores utilizados o en los transistores con los que se diseñan los elementos que componen la red.

2. Métodos

Esta sección presenta el modelo de energía que permitirá evaluar y optimizar el algoritmo propuesto para la aplicación de enrutamiento NoC. EA-NoC busca la ruta que minimice el consumo de energía en la comunicación para la topología tipo malla. El consumo de energía de una red es proporcional al número de transiciones

de bits para transmitir un paquete (*flit*). Para evaluar el consumo de la arquitectura NoC, se usará el siguiente modelo esquematizado de la siguiente forma:

El consumo de energía de un circuito es la integral de la potencia disipada por este a lo largo del tiempo, generalmente para una tarea u operación determinada. De esta forma, si un circuito disipa menos potencia durante un mismo tiempo de operación, consumirá menos energía en ese intervalo como se muestra en la ecuación 1.

$$E_T = \int P dt \quad (1)$$

Donde P es la potencia disipada por el circuito y es definida como la multiplicación entre el voltaje y la corriente $P = v(t) \cdot i(t)$, donde $i(t)$ es la corriente instantánea proporcionada por la fuente de alimentación, y $v(t)$ es la tensión de alimentación instantánea. La disipación de potencia en circuitos construidos con tecnología CMOS (*Complementary metal-oxide-semiconductor*), puede dividirse en dos componentes principales: dinámica y estática, ecuación 2.

$$P = P_d + P_s \quad (2)$$

La potencia dinámica P_d es la potencia consumida cuando el dispositivo está activo, es decir, cuando las señales cambian de valor. En los dispositivos CMOS, la energía dinámica (Ecuación 3) se consume principalmente debido a:

- La carga y descarga de las capacitancias.
- La corriente de cortocircuito

$$P_d = P_{sw} + P_{sc} \quad (3)$$

Potencia Conmutación (P_{sw}) es el resultado de la carga y descarga de las capacitancias del circuito, y al utilizar tecnologías de hasta 100 nm, es la parte de disipación de potencia que más contribuye al consumo total de un circuito. Sin embargo, para tecnologías modernas esta parte de disipación de potencia puede ser menor que la potencia estática [Reehal, 2012]. En un circuito síncrono, la frecuencia de carga y descarga de las capacitancias se refleja directamente en la potencia conmutación, en la ecuación 4 se muestra dicho comportamiento.

$$P_{sw} = \alpha \cdot f_{clk} \cdot C_L \cdot V_{dd}^2 \quad (4)$$

En la ecuación 4, α corresponde a la capacidad de conmutación del total de puertos lógicos del circuito; f_{clk} corresponde a la frecuencia de reloj del circuito; C_L corresponde a la carga capacitiva total del circuito; V_{dd} corresponde a la tensión de alimentación.

Potencia Cortocircuito (P_{sc}) está asociada a cada transición en la entrada de un puerto lógico, cuando tanto el NMOS como el PMOS conducen simultáneamente durante un corto periodo de tiempo. Durante este lapso, existe una corriente de cortocircuito que fluye directamente de la fuente de alimentación V_{dd} a la tierra (GND), no participando en la carga o descarga de la capacitancia del circuito C_L [Reehal, 2012].

Potencia Estática (P_s) El consumo de energía estática en los dispositivos CMOS se debe principalmente a las corrientes de fuga (*leakage current*) en los transistores y en las uniones, estas generan:

- Fugas por debajo del umbral a través de los transistores de apagado.
- Fugas en la compuerta a través del dieléctrico.
- Fugas en la unión (juntura) de la fuente/drenaje.

Para tecnologías con un ancho de canal superior a 180 nm, la potencia estática puede ser despreciada, pues es muy inferior a la potencia dinámica. En las tecnologías actuales, este hecho no ocurre, pues la potencia estática es inversamente proporcional al tamaño de los transistores [Reehal, 2012].

Modelo de Energía para Transmitir un *Flit*

La información intercambiada entre los componentes de un *MPSoC* se organiza en forma de mensajes, que circulan en la NoC a través de los *routers*, partiendo del nodo origen al destino de la comunicación. En general, un mensaje tiene tres partes: encabezado (*header*), carga útil (*payload*) y terminador (*tail*).

El encabezado incluye información de enrutamiento y control, utilizada por los *routers* para propagar el mensaje hacia el nodo destino de la comunicación. El

terminador incluye información utilizada para la detección de errores y para la señalización del final del mensaje. Ambos forman un encapsulado alrededor de la carga útil del mensaje. Los mensajes se dividen en paquetes para la transmisión. Un paquete mantiene una estructura similar a la de un mensaje (*header*, *payload* y *tail*) y es la unidad de información más pequeña que contiene detalles sobre el enrutamiento y la secuencia de la información.

El paquete está constituido por unidades menores denominadas *flits* (*flow control units*). Un *flit* es la menor unidad de información sobre la cual se realiza el control de flujo. Como se caracteriza previamente en [Wang, 2002], la energía consumida al transmitir un *flit* está dada por ecuación 5.

$$\begin{aligned} E_{flit} &= (E_{wrt} + E_{rd} + E_{arb} + E_{xb} + E_{link}).H \\ &= (E_{buf} + E_{wrt} + E_{rd} + E_{xb} + E_{link}).H \end{aligned} \quad (5)$$

En la ecuación 5, E_{wrt} es la energía disipada al escribir un *flit* en el *buffer* de entrada, E_{rd} es la energía disipada al leer un *flit* del *buffer* de entrada, $E_{buf} = E_{wrt} + E_{rd}$, es la energía del *buffer*, E_{arb} es la energía de arbitraje, E_{xb} es la energía promedio de *crossbar* transversal, E_{link} es la energía del enlace transversal promedio y H es la cantidad de saltos atravesados por el *flit*. La ecuación 5 se puede reformular como ecuación 6.

$$E_{flit} = E_R \cdot H + E_{wire} \cdot D \quad (6)$$

En la ecuación 6, $E_R = E_{buf} + E_{wrt} + E_{rd} + E_{xb}$ es la energía promedio del enrutador, E_{wire} es la energía promedio de transmisión de enlace por unidad de longitud, asumiendo repetidores colocados de forma óptima y D es la distancia Manhattan entre el origen y el destino. En este trabajo, para obtener la eficiencia energética de la NoC, se utilizará el simulador Noxim basado en SystemC [Catania, 2016], donde los parámetros de entrada se pueden cambiar a través de una interfaz de línea de comando. La eficiencia energética utilizada por este simulador se basa en el modelo de potencia Orión, un simulador diseñado por el departamento de ingeniería eléctrica de la Universidad de Princeton, capaz de proporcionar características de potencia y rendimiento, para permitir intercambios rápidos de potencia a nivel de arquitectura [Wang, 2002].

Propuesta de Optimización de Enrutamiento

Hasta hace poco, el consumo de energía de un *MPSoC* era un problema de segundo orden en el diseño de este tipo de dispositivos, después de problemas como el costo, el área y el tiempo de diseño. Hoy en día, para la mayoría de los diseños basados en *MPSoC*, el presupuesto de energía es uno de los objetivos de diseño más importantes del proyecto. Superar este presupuesto puede ser fatal, dado que puede causar una confiabilidad inaceptable, un pobre desempeño debido a un consumo excesivo de energía, no cumplir con la vida útil o requerir una batería excesiva. Este tipo de problemas empeoran para dispositivos de tecnología de 90 nm o inferiores. Por ejemplo, la potencia de fuga se ha convertido en una parte importante de la potencia total del *MPSoC*; alcanzando casi el 40% en tecnologías de 65nm. Reducir la potencia siempre que sea posible es casi esencial para cualquier diseño que incluya una *NoC* [Reehal, 2012].

El consumo de energía estático se debe a fugas y corrientes de cortocircuito, esto quiere decir, que se debe a la forma física como están construidas las componentes del *MPSoC*. El consumo dinámico de energía se deriva de la actividad de conmutación, es decir, las transiciones de bits tal como se indicó en las ecuaciones (4) y (5). Las interconexiones consumen la mayor parte del poder dinámico en los *MPSoC* modernos. Por ejemplo, los estudios previos muestran que los enlaces de interconexión consumen hasta el 60% de la potencia dinámica en la *NoC* [Banerjee, 2007], y de esta, más del 60% de la potencia dinámica en un microprocesador moderno [Berman, 2010].

Dado que el algoritmo propuesto busca la ruta que minimice el consumo de energía en la comunicación, solo tendrá en cuenta la energía dinámica disipada por la *NoC*, dado que la energía estática se refiere a la forma física, y usando la ecuación ⁽⁶⁾, en la que se indica el consumo energético que requiere un *flit* para llegar del origen al destino.

La red se representará como un grafo dirigido, $G = (N, L)$ que se muestra en la figura 1, donde N representa el conjunto de nodos (y_i), que modelan los dispositivos de interconexión y L el conjunto de arcos que modelan los enlaces de comunicación (x_{ij}) para la *NoC*.

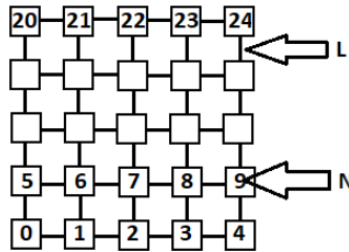


Figura 1 Grafo dirigido $G = (N, L)$, N conjunto de nodos y L el conjunto de enlaces.

La ecuación 7 describe el consumo total de energía para los nodos y los enlaces de la red de comunicación.

$$E_T = \sum_{i=1}^N \sum_{j=1}^N x_{ij} \cdot E_{wire_{ij}} + \sum_i y_i \cdot E_{R_i} \quad (7)$$

En la ecuación 7, $E_{wire_{ij}}$ y E_{R_i} son el consumo de energía del enlace i a j , y el nodo i respectivamente.

La carga impuesta en la red está definida por la matriz de tráfico para cada par de nodos de entrada y salida (s, d) respectivamente. r_{rd} : Tráfico de s a d , f_{ij}^{sd} : Tráfico sobre cualquier enlace (i, j) , a través de los nodos de tránsito (s, d) . La ecuación 8 establece las restricciones de conservación de flujo.

$$\sum_{j=1}^N f_{ij}^{sd} - \sum_{j=1}^N f_{ji}^{sd} = \begin{cases} r_{sd} \quad \forall (s, d), & i = s \\ -r_{sd} \quad \forall (s, d), & i = d \\ 0 \quad \forall (s, d), & i \neq s, d \end{cases} \quad (8)$$

En ecuación 8, cuando i es igual a s , tenemos un nodo de entrada, cuando i es igual a d , tenemos un nodo de salida. De lo contrario tenemos un nodo de destino. La ecuación 9 evalúa la cantidad de tráfico enrutando por enlace.

$$f_{ij} = \sum_{s=1}^N \sum_{d=1}^N f_{ij}^{sd} \quad \forall (i, j) \quad (9)$$

Para preservar la calidad del servicio (QoS) no se debe utilizar el enlace en exceso. En otras palabras, el nivel requerido de desempeño estará garantizado, pero utilizando una cantidad de recursos que se dimensionan sobre la demanda de tráfico real, en lugar de la demanda máxima [Bianzino, 2010]. Por este motivo, se definen los siguientes conjuntos de restricciones:

$$f_{ij} \leq \alpha c_{ij} x_{ij} \quad \forall (i, j) \in L \quad (10)$$

En ecuación 10, α es un valor arbitrario que se considera seguro para la red. Finalmente, un nodo solo se puede desactivar si todos los enlaces de entrada y salida están desactivados, para cualquier elemento de la red bajo el conjunto de restricciones mediante ecuación 11.

$$\sum_{j=1}^N x_{ij} + \sum_{j=1}^N x_{ji} \leq M y_i \quad \forall (i, j) \quad (11)$$

En ecuación 11 M es un número "grande"(big) que se usa para forzar a la variable x_{ij} (enlace) a tomar el valor 1 cuando tiene una carga mayor que cero y 0 cuando la carga es cero, lo que minimiza el consumo total de energía presentado en la ecuación 9 y satisface todas las restricciones mencionadas. La solución de este problema de optimización se llevará a cabo mediante el método de programación lineal de enteros mixtos (*Mixed Integer Linear Programming, MILP*), con variables binarias (x_{ij}). Este método se utiliza en problemas de optimización, donde las variables del modelo pueden ser un conjunto de enteros y reales, además, las restricciones y funciones son lineales [Achuthan, 1991].

Algoritmo de Enrutamiento EA-NoC

En esta sección, presentamos el algoritmo de enrutamiento de fuente, dinámico y adaptativo (*adaptive dynamic and source*) EA-NoC, que permite determinar el mínimo consumo de energía dinámica entre los nodos y enlaces para una determinada ruta S expresada en la ecuación 7. Por medio de la ecuación 8, se determinará si un nodo es de tránsito o destino. Finalmente, la ecuación 11 nos permitirá determinar la ruta seguida por los paquetes entre el origen y el destino de acuerdo con la selección de un puerto de salida. Posteriormente, se explicará cómo se usa este algoritmo en la reducción del consumo de energía dinámica en una NoC. En la figura 2, se muestra el pseudocódigo de la implementación del algoritmo EA-NoC. Dicho algoritmo determina la ruta que deben tomar los paquetes entre origen y destino, consumiendo la menor potencia dinámica en el proceso. Por lo tanto, es un algoritmo de costo mínimo. Si el nodo de origen también es el nodo de

destino (pasos 1, 2), el costo es cero y la dirección de enrutamiento es el puerto local (paso 5). El costo se calcula como el consumo que se requiere para llevar el paquete al nodo vecino a través del enlace (x_{ij}), agregando los costos locales de cada enrutamiento (y_i) (pasos 6, 7), tal como lo expresa la ecuación 7. El costo máximo y el costo mínimo se obtienen al tomar el valor máximo y mínimo, respectivamente, de todas las rutas posibles al destino a través de una ruta (pasos 8, 9). Finalmente, el algoritmo inserta la ruta óptima en el encabezado del paquete, este indica a los enrutadores de tránsito el puerto de salida (paso 10).

Algoritmo 1: Pseudocódigo Algoritmo EA-NoC	
1: Read $Coor_{current} = (Current_{id})$	push_back_energy_east
2: Read $Coor_{destination} = (Dst_{id})$	goto 5
3: Start	
4: Compare ($Current_{id}, Dst_{id}$)	7: else
5: if ($Current_{id} = Dst_{id}$)	get_neighbors_id
push_back_direction_local	get_energy_neighbors_id
push_back_energy_zero	if (direction_west = not_valid)
goto 8	push_back_direction_north
6: else if ($Current_{id} > Dst_{id}$)	push_back_power_north
get_neighbors_id	goto 5
get_energy_neighbors_id	else if (direction_north = not_valid)
if (energy_south = not_valid)	push_back_direction_west
push_back_direction_east	push_back_power_west
push_back_energy_east	goto 5
goto 5	else
else if (energy_east = not_valid)	if (direction_west)
push_back_direction_south	push_back_direction_west
push_back_energy_south	push_back_power_west
goto 5	goto 5
else	else if (direction_north)
if (energy_east < energy_south)	push_back_direction_north
push_back_direction_south	push_back_power_north
push_back_energy_south	goto 5
goto 5	else
else if (energy_east > energy_south)	push_back_direction_west
push_back_direction_east	push_back_power_west
(Continuación)	goto 5
push_back_energy_east	8: Calculate the maximum power of the routes
goto 5	9: Calculate route with lower cost
else	10: Send package direction
push_back_direction_east	11: End

Figura 2 Pseudocódigo Algoritmo EA-NoC.

Flujo de Análisis de Información

El algoritmo de enrutamiento obtiene la potencia de los nodos y enlaces de la red utilizando *Orion*, estos se utilizarán como parámetros de entrada de enrutamiento,

que se actualizarán cada 1000 ciclos de reloj, lo que permitirá que el algoritmo de enrutamiento sea periódico. Cuando se implementa en Noxim, determina la mejor ruta para enviar paquetes entre origen y destino, adaptándose a los parámetros de consumo de la red. El flujo de información se muestra en la figura 3.

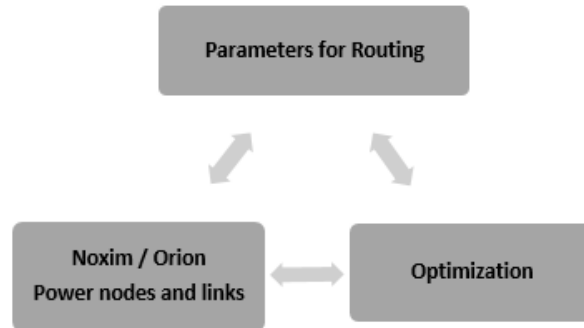


Figura 3 Análisis del flujo de información.

3. Resultados

Esta sección presenta los resultados de la simulación del algoritmo atento a la energía (EA-NoC), implementado en la plataforma de simulación Noxim. Para ello se utilizó una topología tipo malla 10X10, todas las NoC implementadas trabajarán con *flits* de 32 bits y buffer de entrada de 4 posiciones. Las simulaciones se realizaron con una inyección de paquetes (PIR) de tamaño 0.1, 0.2, 0.4, 0.5, 0.6, 0.8 y 1 (flits/cycle), un canal virtual, separación router a router 1.0 mm. y distribución de tráfico sintético, en la tabla 1 se muestra un resumen de los parámetros usados.

Tabla 1 Parámetros de simulación NoC.

Parámetro	Valor Usado
Tipo de red	Malla
Dimensión en X, Y de la red	10 X 10
Tamaño del <i>flit</i>	32 bits
Canales Virtuales	1
Longitud del enlace <i>router a router</i>	1 mm
Periodo de reloj (<i>ps</i>)	1000
Tiempo de simulación (en número de ciclos)	11000

Para verificar el correcto funcionamiento del algoritmo de enrutamiento, se eligió un patrón de tráfico de matriz transpuesta, donde el nodo con coordenadas $a_{n-1}, a_{n-2}, \dots, a_1, a_0$ se comunica con el nodo $a_{\frac{n}{2}-1}, \dots, a_0, a_{n-1}, \dots, a_{\frac{n}{2}}$.

Donde a_0 es la coordenada $x = 0, y = 0$, a_1 es la coordenada $x = 0, y = 0, \dots$, y así sucesivamente.

La figura 4 muestra el análisis de la energía dinámica consumida por la red de malla descrita anteriormente, al implementar el patrón de tráfico matriz transpuesta, para los algoritmos de enrutamiento EA-NoC, North Last, Odd Even, West First y XY.

Debido a que el consumo de energía dinámica del algoritmo XY es mucho mayor que el de otros algoritmos analizados, este debe eliminarse para observar el comportamiento del algoritmo implementado en comparación con los disponibles en la literatura, dicho análisis se muestra en la figura 5.

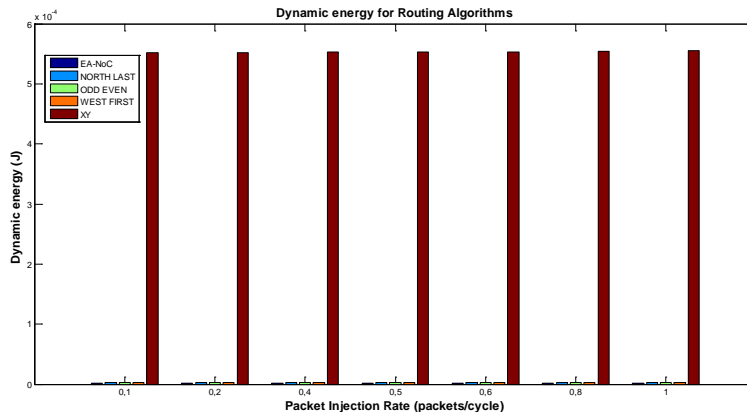


Figura 4 Consumo de energía dinámica.

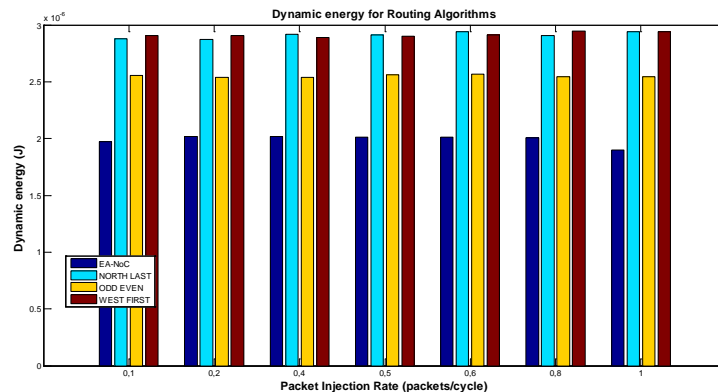


Figura 5 Consumo de energía dinámica sin XY.

La figura 5 muestra el consumo dinámico de energía para siete tasas de inyección de paquetes (PIR). Esta permite analizar el desempeño energético del algoritmo EA-NoC, y se observa una mejora en términos de consumo hasta del 28% en promedio de los algoritmos *North Last*, *West First* y *Odd Even*.

En la figura 6, se muestra el rendimiento de la red para una inyección de paquetes (PIR). Debido que EA-NoC es un algoritmo de costo mínimo, genera pérdidas en el rendimiento asociadas al tiempo que este tarda en determinar cuál es la ruta óptima para enviar el paquete, a diferencia de los algoritmos de rotación como *North Last*, *Odd Even*, *West First*. Esta pérdida representa el 11% en promedio, con respecto al algoritmo XY, se obtienen mejoras del 6%.

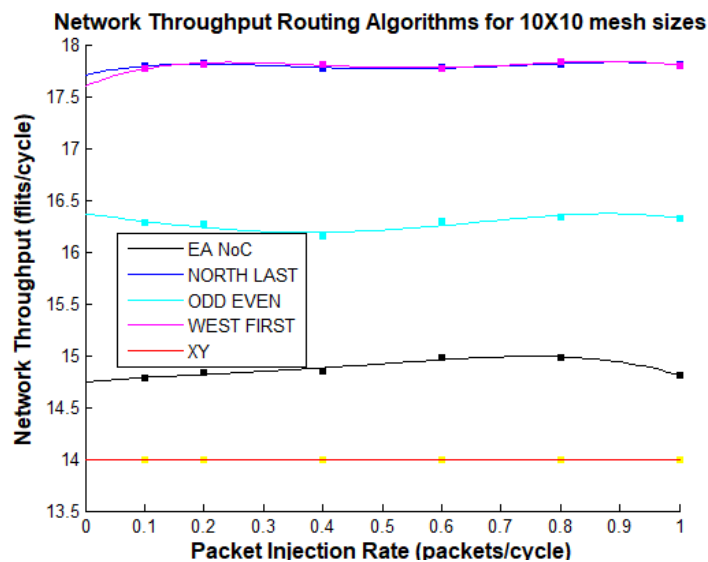


Figura 6 Rendimiento de la red mesh.

La figura 7 muestra la latencia de la red para diferentes tasas de inyección de paquetes (PIR), y como se indicó anteriormente, EA-NoC es un algoritmo de costo mínimo y presentará retardos al determinar la ruta óptima. Tales retrasos representan una pérdida del 2% en promedio.

En la figura 8, se muestra el consumo total de la red. Esta permite observar que los costos de procesamiento adicionales que genera el algoritmo *EA-NoC*, no generan un impacto negativo en el consumo de energía total, obteniendo ahorros del 5% en promedio y en el caso del algoritmo XY superiores al 90%.

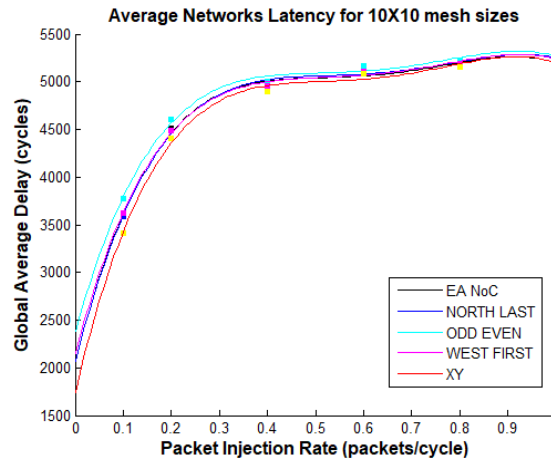


Figura 7 Latencia de la red mesh.

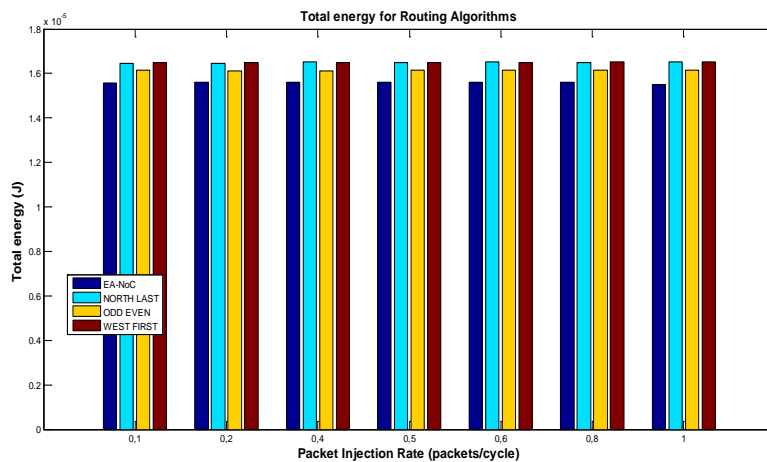


Figura 8 Consumo de total de energía de red sin el algoritmo XY.

4. Discusión y Conclusiones

Los algoritmos adaptativos y distribuidos son complejos y demandan mucha lógica, generando un consumo de energía innecesario. Por lo tanto, no son adecuados para redes intrachip. Por esta razón, se propone un algoritmo parcialmente adaptativo, utilizando el enrutamiento de origen cuyo objetivo es obtener un menor consumo de energía dinámica.

Debido a la disminución constante en el tamaño del MPSoC y al aumento en el número de elementos de procesamiento, se incrementa el consumo de energía, esto hace necesario implementar algoritmos de enrutamiento atentos a la energía como una solución viable para reducir dicho consumo.

Los resultados experimentales del presente trabajo, muestran que el algoritmo *EA-NoC* mejora en un 28% el consumo de energía dinámica en comparación con los algoritmos existentes en la literatura.

Se podría lograr un mayor rendimiento modificando el algoritmo e ingresando métricas adicionales (como la latencia, el rendimiento, la utilización del buffer, etc.), lo que en general, proporcionaría un estado global del sistema de comunicación que obtendría mayores ahorros, pero debe analizarse los costos en Latencia y Throughput que esto acarrearía.

5. Bibliografía y Referencias

- [1] Achuthan, NR; Caccetta, L: Integer linear programming formulation for a vehicle routing problem. En: *European Journal of Operational Research* 52, Nr. 1, p.86-89, 1991.
- [2] Banerjee, Arnab; Mullins, Robert; Moore, Simon: A power and energy exploration of network-on-chip architectures. En: *First International Symposium on Networkson-Chip (NOCS'07) IEEE*, p. 163-172, 2007.
- [3] Bell, Shane; Edwards, Bruce; Amann, John; Conlin, Rich; Joyce, Kevin; Leung, Vince; MacKay, John; Reef, Mike; Boa, Leeway; Brown, John [u. a.]: Tile64-processor: A 64-core SoC with mesh interconnect. En: *2008 IEEE International Solid-State Circuits Conference-Digest of Technical Papers IEEE*, p. 88-598, 2008.
- [4] Benini, Luca; De Micheli, Giovanni: Networks on chips: A new SoC paradigm. En: *computer* 35, Nr. 1, p. 70-78, 2002.
- [5] Berman, Amit: *Power Reduction Techniques for Networks-on-Chip*, 2010.
- [6] Bianzino, Aruna P.; Chaudet, Claude; Larroca, Federico; Rossi, Dario; Rougier, Jean-Louis: Energy-aware routing: A reality check. En: *2010 IEEE Globecom Workshops IEEE*, p. 1422-1427, 2010.
- [7] Bolla, Raffaele; Bruschi, Roberto; Carrega, Alessandro; Davoli, Franco: Green networking with packet processing engines: Modeling and optimization. En: *IEEE/ACM Transactions on Networking (TON)* 22, Nr. 1, p. 110-123, 2014.

- [8] Borkar, Shekhar: Thousand core chips a technology perspective. 44th ACM/IEEE Design Automation Conference IEEE, 2007, p. 746-749, 2007.
- [9] Catania, Vincenzo; Mineo, Andrea; Monteleone, Salvatore; Palesi, Maurizio; Patti, Davide: Cycle-accurate network on chip simulation with noxim. En: ACM Transactions on Modeling and Computer Simulation (TOMACS) 27, Nr.1, p4, 2016.
- [10] Chiaraviglio, Luca; Wiatr, Pawel; Monti, Paolo; Chen, Jiajia; Lorincz, Josip; Idzikowski, Filip; Listanti, Marco; Wosinska, Lena: Is green networking beneficial in terms of device lifetime? En: IEEE Communications Magazine 53, Nr. 5, p. 232-240, 2015.
- [11] Das, Sourav; Doppa, Janardhan R.; Pande, Partha P.; Chakrabarty, Krishnendu: Design-space exploration and optimization of an energy-efficient and reliable 3-D smallworld network-on-chip. En: IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 36, Nr. 5, p. 719-732, 2017.
- [12] De Micheli, Giovanni; Benini, Luca: Networks on chips: 15 years later. En: Computer, Nr. 5, p. 10-11, 2017.
- [13] Kullu, Pinar; Tosun, Suleyman: Energy-aware and fault-tolerant custom topology design method for network-on-chips. En: Nano Communication Networks 19, p. 54-66, 2019.
- [14] Mubeen, Saad. Evaluation of source routing for mesh topology network on chip platforms. 2009.
- [15] Nickolls, John; Dally, William J.: The GPU computing era. En: IEEE micro 30, Nr. 2, p. 56-69, 2010.
- [16] Patooghy, Ahmad; Sarbazi-Azad, Hamid: Performance comparison of partially adaptive routing algorithms. En: 20th International Conference on Advanced Information Networking and Applications-Volume 1 (AINA'06) Vol. 2 IEEE, p. 5-pp, 2006.
- [17] Reehal, Gursharan K.: Designing Low Power and High Performance Network-on-Chip Communication Architectures for Nanometer SoCs, The Ohio State University, Tesis de Grado, 2012.

- [18] Rahaman, Munshi M.; Ghosal, Prasun; Das, Tuhin S.: Latency, Throughput and Power Aware Adaptive NoC Routing on Orthogonal Convex Faulty Region. En: *Journal of Circuits, Systems and Computers* 28, Nr. 04, p. 1950055, 2019.
- [19] Tsai, Wen-Chung; Lan, Ying-Cherng; Hu, Yu-Hen; Chen, Sao-Jie: Networks on chips: structure and design methodologies. En: *Journal of Electrical and Computer Engineering* 2012, p. 2, 2012.
- [20] Wang, Hang-Sheng; Zhu, Xinping; Peh, Li-Shiuan; Malik, Sharad: Orion: a power performance simulator for interconnection networks. En: *Proceedings of the 35th annual ACM/IEEE international symposium on Microarchitecture* IEEE Computer Society Press, p. 294-305, 2002.
- [21] Wu, Ji; Dong, Dezun; Wang, Li: NoC power optimization using combined routing algorithms. En: *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS) IEEE*, p. 299-304, 2017.
- [22] Zhan, Jia; Ouyang, Jin; Ge, Fen; Zhao, Jishen; Xie, Yuan: DimNoC: A dim silicon approach towards power-efficient on-chip network. En: *2015 52nd ACM / EDAC / IEEE Design Automation Conference (DAC) IEEE*, p. 1-6, 2015.