

Universidad de los Andes
Facultad de ingeniería
División de estudios de postgrado
Postgrado en Computación



X
QA249.5
T6

Sistema de comunicación autonómico basado en redes bayesianas y ontologías

Ing. José Francisco Torres Monzón

Tutor: Ph.D José L. Aguilar C.

www.bdigital.ula.ve

DONACION

Trabajo de grado presentado ante la ilustre Universidad de Los Andes como requisito parcial para optar al grado de Magister Scientiae en Computación

SERBIULA
Tullio Febres Cordero

Mérida, Mayo 2014

Resumen

En los sistemas de comunicación modernos es necesario desarrollar formas para optimizar los parámetros de comunicación. Para lograr este objetivo de forma flexible y adaptable, son necesarios nuevos paradigmas, como los vinculados a la computación autónoma, que establece, entre otras cosas, que los sistemas deben poder auto-configurarse automáticamente para mejorar su desempeño. En base a esto, en el presente trabajo se propone el desarrollo de un sistema de comunicación autónomo, el cual utiliza un sistema de toma de decisiones basado en una red bayesiana y una ontología como base de conocimiento, para tomar decisiones de reconfiguración en el sistema, de tal forma de optimizar y adaptar la plataforma comunicacional a las características del contexto. El sistema propuesto está inspirado en la arquitectura DAISY (Diagnosis for Adaptive Strategies in collaborative Systems), propuesto en el marco del proyecto ADREAM (Architectures for Dynamic Resilient Embedded Autonomous and Mobile systems), en el LAAS (Laboratory for Analysis and Architecture of Systems). La ontología provee el conocimiento acerca de los factores de desempeño y sus relaciones en las redes de comunicación, para configurar la estructura de la red bayesiana, mientras que esta última es el mecanismo de razonamiento estocástico usado por el sistema de toma de decisión. Para verificar el funcionamiento del mismo, se diseñaron varias pruebas sobre el simulador NS2. En las pruebas realizadas con el simulador se pudo observar una mejora en el rendimiento del sistema, entre un 85% y un 96%, para los escenarios considerados donde el sistema de decisión reconfiguro el sistema de comunicación de forma autónoma.

Contenido

Resumen.....	i
1. Introducción.....	1
1.1. Antecedentes.....	2
1.2. Objetivos.....	4
1.2.1. Objetivos Generales.....	4
1.2.2. Objetivos Específicos.....	4
1.3. Resultados esperados.....	4
1.4. Estructura del presente Trabajo.....	5
2. Marco teórico.....	6
2.1. Computación Autónoma.....	6
2.2. Sistemas de comunicaciones.....	9
2.2.1. Modelo de comunicación de Shannon y Weaver.....	9
2.2.2. Protocolos de comunicación.....	10
2.2.3. Sistemas de comunicación autonómicos.....	11
2.3. Redes Bayesianas.....	12
2.3.1. Conceptos Básicos.....	12
2.3.2. Inferencia en Redes Bayesianas.....	13
2.3.3. Aprendizaje en las redes bayesianas.....	14
2.3.4. Toma de decisiones en Redes Bayesianas y Diagrama de influencia.....	19
2.4. Ontologías.....	23
2.4.1. Ontología: conceptos básicos.....	23
2.4.4. Proceso de transformación de una ontología a una red bayesiana.....	29
3. Diseño del Sistema.....	33
3.1. Arquitectura propuesta para el sistema.....	33
3.1.1. Sistema de toma de decisión de reconfiguración.....	34
3.1.2. Sistema de reconocimiento de contexto.....	35
3.2. Ontología de dominio.....	35
3.2.1. Arquitectura propuesta para la transformación de la Ontología a la red bayesiana.....	38
3.2.2. Transformación de la Ontología propuesta en red bayesiana.....	39
3.2.3. Transformación de la red bayesiana en un diagrama de influencia.....	42
3.3. Entonación del Modelo de Decisión y aprendizaje inicial.....	44

4.	Pruebas experimentales del Sistema	49
4.1.	Diseño del protocolo experimental	49
4.1.1.	Sub-sistema de cálculo de desempeño.....	52
4.1.2.	Proceso de simulación	53
4.1.3.	Protocolo de experimentación.....	55
4.1.4.	Descripción del escenario de simulación.....	56
4.3.	Escenario cuya decisión es basada en las pérdidas.....	63
4.4.	Simulaciones y Resultados experimentales	64
4.4.1.	Resultados cuantitativos con decisión basada en el rendimiento	64
o	Resultados para la Diagrama de influencia 1	64
o	Resultados para la Diagrama de influencia 2	65
4.4.2.	Resultados cuantitativos de la simulación con decisiones en base a las pérdidas68	
4.4.3.	Resultados del proceso de aprendizaje.....	72
4.4.4.	Análisis de las variables clásicas de rendimiento en los sistemas comunicacionales para el escenario 1 de pérdidas	73
4.4.5.	Resultados cualitativos.....	76
5.	Conclusiones y Recomendaciones.....	79
5.1.1.	Conclusiones	79
5.1.2.	Recomendaciones	80
	Bibliografía	81
6.	Anexos.....	85
6.1.	Anexo de Simulaciones y Resultados experimentales.....	86
o	Resultados Escenario de simulación de pérdidas	86
o	Resultados Escenario de variación del Ancho de Banda.....	92
o	Resultados Escenario de variación del Tamaño de la Cola.....	96
o	Resultados Escenario de variación del Retardo	98
o	Resultados Escenario de variación de tipo y Tasa de transmisión de la conexión interferente.....	102
o	Resultados Escenario de simulación de pérdidas con decisión es base a las pérdidas.....	108
6.2.	Anexo de scripts y códigos de simulación y decisión.....	111
6.3.	Anexo de Simulaciones y Resultados experimentales.....	130

Índice de Figuras

Figura 2-1 Arquitectura de la Computación Autónoma (IBM, June 2005).....	7
Figura 2-2 Arquitectura de la Computación Autónoma (IBM, June 2005).....	8
Figura 2-3 Shannon y Weaver.....	9
Figura 2-4 Arquitectura para auto adaptación (Aguilar, Subias, Travé-Massuyés, & Zouaoui, GIIIS 2012).....	11
Figura 2-5 Comunicación Autónoma (Altamura, 2009).....	12
Figura 2-6 Tipos de propagación de probabilidades.....	13
Figura 2-7 nodos involucrados en el aprendizaje por conteo.....	17
Figura 2-8 Ejemplo de Diagrama de influencia.....	20
Figura 2-9 Red Bayesiana y Diagrama de influencia (Kjaerulff & Madsen, 2005).....	21
Figura 2-10 Ejemplo de Tabla CPT.....	23
Figura 2-11 Arquitectura dirigida por modelos (extraído de (Singh & Sood, December 2010)).....	25
Figura 2-12 Relaciones entre la meta ontología y las ontologías (Pittarello & Celentano).....	26
Figura 2-13 Ontología de Shannon y Weaver.....	27
Figura 2-14 Ontología para la capa de transporte (Exposito, 2010).....	28
Figura 2-15 Ontología del proyecto Hydra (Consortium, 2009).....	28
Figura 2-16 Ontología de protocolos de control basado en IETF, para el caso de contenidos multimedia.....	29
Figura 2-17 rdfs: subclassOf.....	30
Figura 2-18 owl: intersectionOf.....	30
Figura 2-19 owl: unionOf.....	31
Figura 2-20 owl: complementOf, owl: equivalentClass, owl: disjointWith.....	31
Figura 3-1 Arquitectura del sistema de comunicación autónomo.....	33
Figura 3-2 Sub-Sistema de decisión de reconfiguración.....	34
Figura 3-3 Ontología de Concepción de modelos de transporte de nueva generación.....	35
Figura 3-4 Ontología del proyecto Hydra.....	36
Figura 3-5 Ontología.....	36
Figura 3-6 Ontología.....	36
Figura 3-7 Modelo General.....	37
Figura 3-8 Ontología propuesta.....	38
Figura 3-9 Segmento de la ontología a implantar.....	38
Figura 3-11 Ontología reducida.....	39
Figura 3-10 Arquitectura para la transformación de la Ontología a la red bayesiana.....	39
Figura 3-12 Primitivas de clase.....	40
Figura 3-13 Subclases.....	40
Figura 3-14 Traducción estructural completa.....	41
Figura 3-15 Tablas CPT, con valores uniformes.....	42
Figura 3-16 Tablas CPT, luego del aprendizaje inicial con EM.....	42
Figura 3-17 Proceso de transformación de la red bayesiana en un diagrama de influencia.....	43
Figura 3-18 Ontología con nodo de decisión.....	44
Figura 3-19 Ontología transformada en Diagrama de influencia.....	44

Figura 3-20 Diagrama de influencia Utilizado para las simulaciones	45
Figura 3-21 Aprendizaje sobre la red Bayesiana	47
Figura 3-22 Tabla de utilidad.....	47
Figura 4-1 Sistema de decisión (Lazo cerrado de simulación).....	49
Figura 4-2 Resultados de la simulación.....	50
Figura 4-3 Normalización Retardo Nominal vs Retardo Teórico	51
Figura 4-4 Detalle de Rendimiento vs Tiempo de simulación	53
Figura 4-5 Detalle de Rendimiento vs Tiempo de simulación, con línea de tendencia.....	53
Figura 4-6 Grafico de rendimiento con y sin sistema autonómico.....	55
Figura 4-7 Grafico del Caudal Normalizado con y sin sistema autonómico	55
Figura 4-8 Escenario de simulación	57
Figura 4-9 Diagrama de influencia simple, Tipo 1.....	59
Figura 4-10 Diagrama de influencia de distribución de frecuencia para solo el rendimiento, Tipo 2.....	59
Figura 4-11 Escenario 1 Diagrama de influencia tipo 2 en base a tasa de pérdidas menores al 16 %.....	69
Figura 4-12 Escenario 2 Diagrama de influencia tipo 2 decisiones en base a tasa de pérdidas menor a 45%	71
Figura 4-13 Caudal del Escenario 2 Diagrama de influencia tipo 2 decisiones en base a tasa de pérdidas menor a 45%.....	71
Figura 4-14 Retardo con SA y sin SA, simulación de la Tabla 6-2 para un 16% de perdidas	74
Figura 4-15 Caudal con SA y sin SA simulación de la Tabla 6-2 para un 16% de perdidas	74
Figura 4-16 tasa de perdidas con SA y sin SA simulación de la Tabla 6-2 para un 16% de perdidas	75
Figura 6-1 Escenario 1 Diagrama de influencia tipo 1	87
Figura 6-2 Escenario 1 Diagrama de influencia tipo 2	88
Figura 6-3 Escenario 2 Diagrama de influencia tipo 1	89
Figura 6-4 Escenario 2 Diagrama de influencia tipo 2	91
Figura 6-5 Escenario de Diagrama de influencia tipo 1	92
Figura 6-6 Rendimiento Diagrama de influencia tipo 1	93
Figura 6-7 Caudal Diagrama de influencia tipo 1.....	93
Figura 6-8 Escenario de Diagrama de influencia tipo 2	94
Figura 6-9 Rendimiento Diagrama de influencia tipo 2	95
Figura 6-10 Caudal Diagrama de influencia tipo 2.....	95
Figura 6-11 Escenario de Diagrama de influencia tipo 1.....	96
Figura 6-12 Escenario de Diagrama de influencia tipo 2.....	97
Figura 6-13 Escenario de Diagrama de influencia tipo 1.....	98
Figura 6-14 Caudal de Diagrama de influencia tipo 1.....	99
Figura 6-15 Rendimiento de Diagrama de influencia tipo 1	99
Figura 6-16 Escenario de Diagrama de influencia tipo 2.....	100
Figura 6-17 Rendimiento Diagrama de influencia tipo 2	101
Figura 6-18 Caudal Diagrama de influencia tipo 2.....	101

Figura 6-19 Escenario de UDP de Diagrama de influencia tipo 1	102
Figura 6-20 Rendimiento de UDP con de Diagrama de influencia tipo 1	103
Figura 6-21 Caudal de UDP con de Diagrama de influencia tipo 1	103
Figura 6-22 Escenario de UDP de Diagrama de influencia tipo 2	104
Figura 6-23 Caudal de UDP con de Diagrama de influencia tipo 2	105
Figura 6-24 Rendimiento de UDP con de Diagrama de influencia tipo 2	105
Figura 6-25 Escenario de TCP de Diagrama de influencia tipo 1.....	106
Figura 6-26 Escenario de TCP de Diagrama de influencia tipo 2.....	107
Figura 6-27 Escenario 1 Diagrama de influencia tipo 2 en base a las pérdidas	109
Figura 6-28 Escenario 2 Diagrama de influencia tipo 2 decisiones en base a las pérdidas	110
Figura 6-29 Retardo con SA y sin SA simulación de la Tabla 6-2 para un 16% de perdidas	131
Figura 6-30 Caudal con SA y sin SA simulación de la Tabla 6-2 para un 16% de perdidas	131
Figura 6-31 tasa de perdidas con SA y sin SA simulación de la Tabla 6-2 para un 16% de perdidas	132
Figura 6-32 Rendimiento con SA y sin SA simulación de la Tabla 6-2 para un 16% de perdidas	132

www.bdigital.ula.ve

1. Introducción

En la actualidad, debido al gran número de dispositivos de interconexión, a las tecnologías de interconexión, y al desarrollo de nuevos sistemas de valor agregado sobre las plataformas de comunicación existentes, tenemos una diversidad de medios de comunicación disponibles. Por otro lado, cada día son más comunes las aplicaciones multimedia y las comunicaciones en tiempo real que requieren de mayores anchos de banda, menores latencias, menor variación en el tiempo de llegada de sus paquetes (jitter), mayor confiabilidad, y en línea general, de entornos sensibles a variaciones en el caudal (throughput). De esta manera, surge la necesidad de optimizar la utilización de los recursos actuales de las redes de comunicación. Una de las respuestas para lo anterior, es desarrollar sistemas para que los dispositivos de comunicación se auto-configuren automáticamente para mejorar su desempeño; es decir, se requiere del desarrollo de sistemas de comunicación autónomos los cuales mejoren el desempeño de la comunicación reconfigurándose.

En ese sentido, en este trabajo se propone desarrollar un sistema autónomo comunicacional que optimice su desempeño en función de lo que él conozca de su contexto (problemas de transmisión de datos, etc.) y de las aplicaciones ejecutándose sobre él (multimedias o no, con o sin calidad de servicio, etc.). El sistema autónomo comunicacional requiere de un sistema de toma de decisión que guíe el proceso de reconfiguración, como forma de adaptación a sus nuevas condiciones operacionales. En nuestro caso, el sistema de decisión autónomo requiere de una ontología que caracteriza/formaliza un sistema comunicacional, las tareas autónomas a realizar, etc. Además, a partir de esa ontología se desarrolla un modelo bayesiano de decisión, que se actualiza a través de un proceso de aprendizaje, para así poder adaptar su comportamiento en base a los cambios de su contexto operacional. Así, proponemos desarrollar un sistema autónomo de decisión basado en una ontología, a partir de la cual se concibe una red bayesiana como mecanismo de inferencia para la toma de decisión. La principal idea del uso de las redes bayesianas es debido a que las mismas permiten el manejo de la incertidumbre, típico en los sistemas de comunicaciones.

En particular, el sistema autónomo a desarrollar requiere de técnicas computacionales de toma de decisión basada en redes bayesianas, del paradigma de arquitecturas manejadas por ontologías, y de la computación autónoma, entre otros, como elementos teóricos de base. En ese sentido, los antecedentes del presente trabajo están basados en estos temas.

1.1. Antecedentes

El presente trabajo basa sus principios en: los sistemas de comunicación autonómicos, los sistemas de toma de decisiones inteligentes basados en ontologías y redes bayesianas, así como en los sistemas adaptativos de toma de decisión. Nuestra revisión bibliográfica se organizará de esa manera.

Los sistemas de comunicación autonómicos han sido desarrollados inspirados en la computación autonómica, con el objeto de permitir la autoconfiguración, autogestión, y auto-regulación de las infraestructuras de comunicaciones. Algunos trabajos en sistemas autonómicos de comunicación son presentados en (Dobson, Massacci, & Denazis, 2006), (Kephart & Chess, 2003) (Van, 2009). En cuanto a los tópicos más desarrollados en comunicación autonómica se encuentran: el desarrollo e implementación de un kit de comunicaciones autonómico (ACT), que provee a las aplicaciones de un sistema de comunicación que satisfaga sus requerimientos de calidad de servicio (QoS), propuesto en la Universidad de Toulouse III (Van, 2009). En ese trabajo se hace uso de las *cadena de Markov* para la toma de decisiones acerca de si el sistema se debe o no reconfigurar, y plantea teóricamente un modelo de *aprendizaje por reforzamiento* para optimizar el sistema de comunicaciones. Otro trabajo relacionado estrechamente con la comunicación autonómica es propuesto en (Qi, Zhang, Cao, Sun, & Tan, 2012), donde usan redes cognoscitivas y diagramas de influencia para tomar decisiones de cómo mejorar la comunicación. Para probar el concepto montaron un prototipo sobre el simulador NS2. Finalmente, un trabajo vinculado al problema de reconfiguración autonómica es propuesto en el proyecto DAISY (Diagnosis for Adaptive Strategies in collaborative Systems). En particular, en ese proyecto se propone una arquitectura de un sistema autonómico de comunicación, la cual se basa en la identificación de la situación actual en la que se encuentra un sistema de comunicación, para entonces invocar al proceso de auto-adaptación que garantiza que el sistema pueda seguir funcionando correctamente (Aguilar, Subias, Travé-Massuyés, & Zouaoui, GIIIS 2012).

En sistemas adaptativos de toma de decisión (decision-making of self-adaptive systems (SASs)) uno de los trabajos más relevantes es desarrollado en el INRIA de Paris (Bencomo & Belaggoun, 2013). En dicho trabajo se plantea una especificación metodológica, la cual plantea el uso de redes dinámicas de decisión basadas en redes bayesianas, las cuales se enriquecen con metas a corto plazo, para así mejorar su desempeño en procesos de toma de decisiones con incertidumbre.

En sistema de toma de decisiones inteligentes basados en ontologías, en la tesis doctoral de (Gómez-Montalvo, 2012), en el INSA-Toulouse, se plantea el uso de una ontología, basada en el paradigma *Arquitecturas dirigidas por ontologías* (ODA por sus siglas en inglés Ontology Driven Architecture), para la Gestión Autónoma de la Calidad de servicio en redes comunicacionales. En ese trabajo se hace uso de un marco de trabajo basado en ontologías, para desarrollar un sistema autonómico comunicacional. Dicha ontología les permite a los usuarios y aplicaciones describir sus requerimientos de QoS, para así

posibilitar el proceso autonómico del medio comunicacional, que le permita el aprovisionamiento de la QoS requerida.

Otros temas relacionados con el presente trabajo son las *redes bayesianas* (Ben-Gal, Faltin, & Kenett, 2007) (Tenenbaum, Griffiths, & Kemp, 2006) (Pearl, 1988), las cuales permiten representar la incertidumbre de las relaciones de dependencia entre las variables que describen un proceso de toma de decisión (en nuestro caso, vinculadas a los sistemas de comunicación), y el *aprendizaje en redes bayesianas* (Sucar, 2006) y (Pearl, 1988), el cual puede dividirse en dos tipos: paramétrico y estructural. Otros trabajos relacionados tienen que ver con la construcción de redes bayesianas a través de ontologías (Helspe & van der Gaag, 2002) (Rong, Zhongli, & Peng, 2006). En esos trabajos se plantean metodologías para la construcción de redes bayesianas por medio de ontologías. Por ejemplo, en (Rong, Zhongli, & Peng, 2006) se propone un marco de trabajo llamado BayesOWL, en el cual se especifica formalmente un método para la creación de redes bayesianas a partir de ontologías, que permite el manejo de incertidumbre sobre el dominio para el cual fue diseñado la ontología.

Finalmente, en cuanto al paradigma *MDA*, (Model Driven Architecture (MDA Guide Version 1.0.1, 2003)), el cual es un método de diseño de software que proporciona un conjunto de directrices basadas en modelos para la especificación de sistemas, en (Gómez-Montalvo, 2012) se propone el diseño e implementación de un sistema comunicacional basado en ese principio.

La principal contribuciones del presente trabajo es el uso de una ontología para construir una red bayesiana, para el proceso de toma de decisiones de reconfiguración en un sistema de comunicación autonómico. Con respecto a los trabajos previos, se utiliza una ontología basada en la fusión de varias ontologías de referencias de sistemas de comunicaciones, las cuales aportan una base de conocimiento para el sistema de toma de decisión. Con esa ontología se construye la red bayesiana, la cual hace que el sistema pueda lidiar con la incertidumbre típica de los sistemas de comunicación. La red bayesiana se utiliza para escoger la acción de reconfiguración del sistema que maximice el rendimiento de la red. Además, el sistema autónomo de toma de decisiones puede predecir con eficacia la tendencia de rendimiento de la red, de forma tal que pueda evitar su deterioro. Adicionalmente a esto, el sistema es capaz de aprender, de forma tal de actualizar los parámetros de la red bayesiana, para así poder reconfigurar la plataforma comunicacional según la información actual de ella. En adición a lo anterior, el sistema puede tomar decisiones con información parcial o total sobre el sistema de comunicación. El presente trabajo ofrece varias mejoras con respecto a los trabajos previos:

- Nuestro sistema no requiere acciones conjuntas entre el emisor y el receptor, solo el emisor toma las decisiones; en particular, todos los parámetros que se requieren para los caculos asociados con la comunicación (caudal, tasa de pérdidas, retardo) se pueden determinar observando la comunicación desde el punto de vista del emisor.
- Nuestro sistema mezcla las virtudes de muchos de los trabajos previos. Por ejemplo, el sistema propone un proceso adaptativo del mecanismo de toma de decisión (red

bayesiana) para mantenerlo actualizado con respecto a la información del contexto. También, el sistema se inspira en ODA para hacer una definición ontológica de la base de conocimiento a utilizar en el sistema de toma de decisión, la cual contiene los estándares de base usados en nuestra propuesta. Finalmente, el sistema autónomo que se propone termina fusionando en la red bayesiana el marco ontológico, y los procesos de análisis y planificación, para la reconfiguración del sistema comunicacional.

1.2. Objetivos

1.2.1. Objetivos Generales

Diseñar un sistema de toma de decisiones para un sistema de comunicaciones autónomo, el cual utilice el paradigma de arquitectura manejada por ontologías, para concebir un sistema bayesiano de decisión para el proceso de auto-configuración.

1.2.2. Objetivos Específicos

- Desarrollar una ontología para un sistema de comunicación autónomo.
- Diseñar un sistema de toma de decisión para un sistema autónomo, usando redes bayesianas basadas en la ontología de un sistema de comunicación autónomo, específicamente para el proceso de reconfiguración.
- Desarrollar un prototipo del sistema de comunicaciones autónomo, usando el simulador NS.
- Realizar un estudio de rendimiento del sistema de comunicaciones autónomo propuesto.

1.3. Resultados esperados

Los resultados esperados en el desarrollo del presente trabajo de investigación son:

- El diseño de un modelo arquitectónico de un sistema comunicacional autónomo, el cual estará basado en ontologías.
- Un sistema para toma de decisiones basado en una red bayesiana, concebida desde las ontologías del modelo arquitectónico, la cual se utilizará para escoger que tipo de reconfiguración se aplicará al sistema.
- Un mecanismo adaptativo del sistema de toma de decisión, basado en un algoritmo de aprendizaje
- Un prototipo para realizar pruebas y observar las posibles mejoras en el sistema.

El Modelo de toma de decisión para un entorno auto-reconfigurable se diseña usando una ontología, a partir de la cual se construirá una red de bayesiana para inferir las decisiones de reconfiguración a tomar en el sistema.

El mecanismo adaptativo basado en un algoritmo de aprendizaje actualiza la red bayesiana de toma de decisiones. Dicho mecanismo le permite al sistema aprender cuales son las acciones que conllevan una mejoría en su desempeño.

1.4. Estructura del presente Trabajo

Este trabajo está organizado de la siguiente manera: el Capitulo 1 hace una introducción general, y presenta los objetivos y antecedentes. El Capitulo 2 presenta el Marco teórico, en específico, qué es la computación autónoma, los sistemas de comunicación desde el punto de vista de un sistema autónomo, y las redes bayesianas para los procesos de toma de decisiones y como se construyen desde marcos ontológicos. El Capitulo 3 detalla el diseño del sistema de toma de decisión, en particular, presenta la arquitectura y ontología propuesta, así como el proceso de transformación de la ontología propuesta a la red bayesiana, para su uso en el sistema de toma de decisión. El Capitulo 4 detalla las pruebas experimentales, los escenarios de simulación. Así mismo, también se muestran y analizan los resultados arrojados por las pruebas realizadas. Finalmente, el Capitulo 5 presenta las conclusiones del trabajo y las recomendaciones del mismo.

www.bdigital.ula.ve

2. Marco teórico

2.1. *Computación Autónoma*

El paradigma de la computación autónoma (Autonomic Computing, AC) se inspira en el sistema nervioso autónomo del ser humano. Su objetivo fundamental es que los sistemas computacionales se puedan auto-gestionar. Un sistema autónomo toma las decisiones por su cuenta, utilizando políticas de alto nivel, para revisar y optimizar constantemente su situación, y adaptarse a las condiciones cambiantes de su ambiente (Kephart & Chess, 2003). La AC debe poseer al menos estas cuatro propiedades:

- Auto-configuración (Self-configuration): configuración automática de sus componentes;
- Auto-sanación (Self-healing): descubrimiento y corrección automática de errores
- Auto-optimización (Self-optimization): Control automático de los recursos, para garantizar el funcionamiento óptimo con respecto a los requisitos definidos;
- Auto-protección (Self-protection): identificación y protección proactiva contra ataques.

Adicionalmente a estas propiedades, algunos autores consideran otras más, entre las que se pueden mencionar (Parashar & Hariri, 2005), (FP7, 2012):

- Auto-consciencia (Self Awareness): un sistema o aplicación "se conoce a sí misma", y es consciente de su estado y sus comportamientos.
- Conscientes del Contexto (Context Aware): debe ser consciente de su entorno de ejecución, y ser capaz de reaccionar a los cambios en el medio ambiente.
- Abierto (Open): debe funcionar en un mundo heterogéneo, y debe ser portable a múltiples hardwares y arquitecturas.
- Anticipatoria (Anticipatory): el sistema debe ser capaz de anticiparse, en la medida de lo posible, a sus necesidades y comportamientos de su contexto, de manera proactiva.

La arquitectura de un sistema computacional autónomo ha sido descrita en (IBM, 2005), sus elementos le permiten manejar su comportamiento interno para genera las propiedades antes descritas (ver Figura 2-1 y Figura 2-2). Los componentes básicos (bloques) de la arquitectura se han definido de tal forma que la arquitectura sea abierta, para ser usada en ambientes heterogéneos.

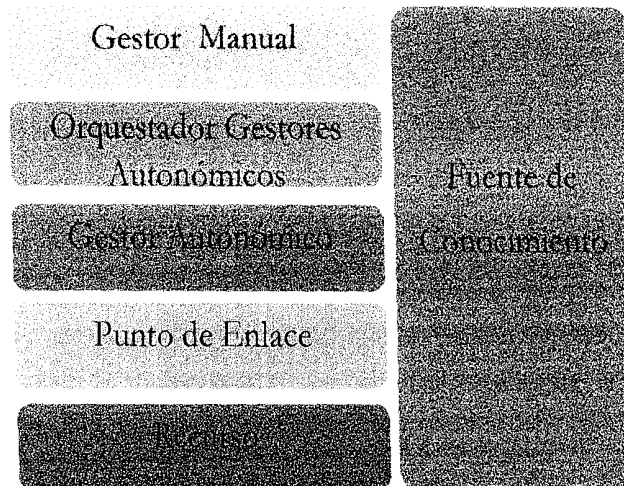


Figura 2-1 Arquitectura de la Computación Autónoma (IBM, June 2005)

Los componentes de esa arquitectura son:

- **Recurso Gestionado:** puede ser cualquier tipo de recursos (hardware o software) que puede ser gestionado. El recurso gestionado se controla a través de sus sensores y actuadores.
- **Puntos de Enlace:** Enlaza a los sensores y/o actuadores requeridos para la gestión de los recursos.
- **Gerente Autónomo:** Implementa los lazos de control inteligentes que automatizan las tareas de autorregulación/autogestión de las aplicaciones
- **Orquestador de los Gerentes Autónomos:** Debido a que un sistema autónomo puede contar con varios gerentes autónomos que necesitan trabajar en conjunto para garantizar el funcionamiento correcto de los recursos, este nivel proporciona el canal de comunicación para la coordinación entre ellos
- **Manejador Manual:** Permite a los humanos configurar los gerentes autónomos para realizar su tarea de autogestión, proveyendo para esto una interfaz hombre-máquina que permite conectar al hombre con el gerente autónomo.
- **Fuentes de Conocimiento:** Proporciona acceso a los conocimientos requeridos para la gestión autónoma del sistema. Los distintos componentes del gerente autónomo

En la Figura 2-2 se muestran los componentes del lazo de control de la arquitectura general de la AC, los cuales comparten una base de conocimiento en común, estos componentes son (IBM, 2005):

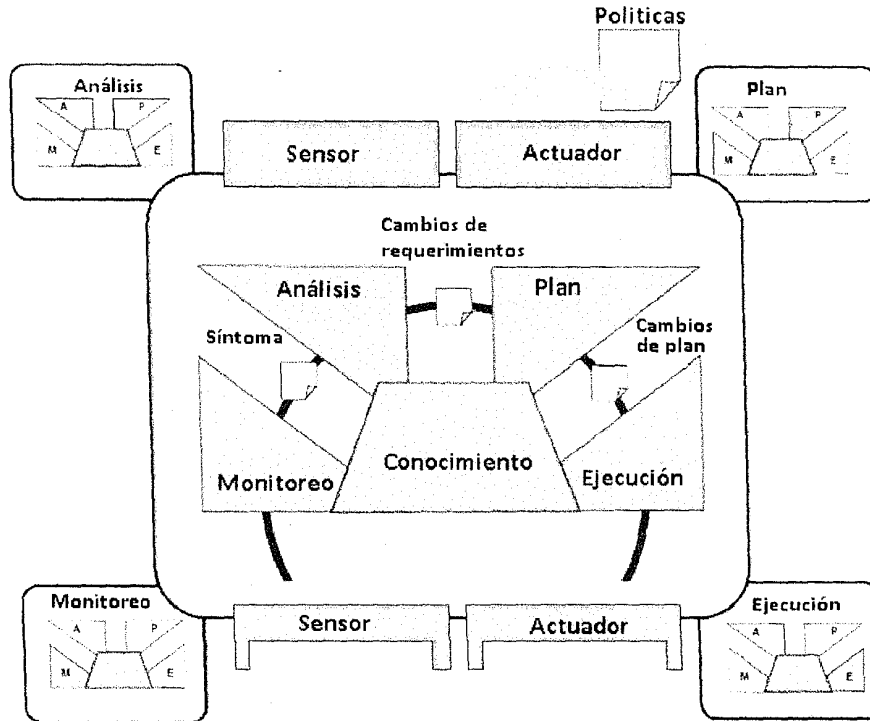


Figura 2-2 Arquitectura de la Computación Autónoma (IBM, June 2005)

- Componente de monitoreo: el cual prevé un mecanismo de recolección, agregación, filtrado y reporte de detalles de los recursos que se manejan.
- Componente de análisis: el cual suministra un mecanismo que correlaciona y modela situaciones complejas, para analizar que puede estar ocurriendo en el sistema. Este mecanismo permite al componente autónomo aprender acerca del ambiente, comprender la situación actual, inferir futuras situaciones, entre otras cosas.
- Componente de planificación: provee un mecanismo para construir y planear las acciones necesarias para cumplir las metas y objetivos del sistema autónomo.
- Componente de ejecución: define un mecanismo que ejecuta las acciones necesarias definidas en el componente de planificación, para cumplir con las metas y objetivos del sistema.

Estos cuatro componentes, junto con el componente de representación del conocimiento, forman el lazo de control. Todos estos componentes se comunican y colaboran unos con los otros, e intercambian conocimiento y datos (IBM, 2005). Particularmente, el componente de representación de conocimiento provee un repositorio de información y conocimiento.

2.2. Sistemas de comunicaciones

A continuación se describirá brevemente el modelo de comunicación de Shannon y Weaver, el cual es la base de la teoría moderna de los sistemas de comunicación. También se van a nombrar brevemente algunos de los protocolos utilizados en las redes de datos, los cuales se utilizarán posteriormente en las simulaciones, junto con una breve introducción a los sistemas de comunicación autónomos.

2.2.1. Modelo de comunicación de Shannon y Weaver

Se define como un sistema de comunicación a los componentes o subsistemas que permiten la transferencia y/o intercambio de información, entre varios entes. En línea general, los elementos de un sistema de comunicación son definidos en el modelo de comunicación de Shannon y Weaver, formulado a finales de los años 40, los cuales son (ver Figura 2-3):

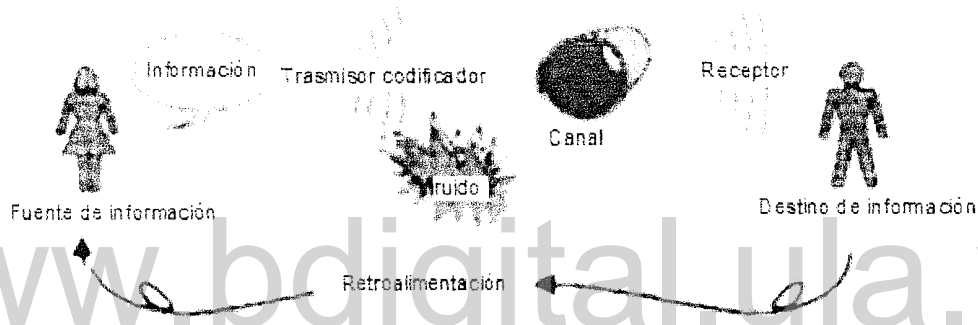


Figura 2-3 Shannon y Weaver

- Información (Message): es el mensaje que se desea entregar.
- Fuente de información (sender): es el ente que genera la información inicial que se desea transmitir.
- Trasmisor codificador (transmission): Convierte el mensaje a un formato adecuado para su transmisión y o codificación.
- Canal (Channel): o medio de comunicación, se define como el medio físico que une el transmisor y el receptor.
- Receptor o decodificador (reception): Decodifica o convierte el mensaje transmitido a un formato y /o codificación adecuado para el Destino.
- Destino de información (receiver): es el ente que recibe la información generada por la fuente de información.
- El ruido (noise): es un perturbador, que parasita en diverso grado la señal durante su transmisión.

- Retroalimentación (Feedback): la retroalimentación es la que le informa al emisor si el mensaje ha sido entregado (y en algunos casos, los términos de calidad con que fue entregado).

Desde el punto de vista de los sistemas de decisión, los sistemas de comunicaciones pueden ser vistos como una serie de elemento interactuado entre sí con una meta en común: transmitir un mensaje. Los mensajes pasan por un proceso de adecuación (es decir, se les hace una codificación, se les agregan elementos para distinguirlos de otros mensajes, se les puede segmentar, se les puede adicionar información adicional, etc.) para su transmisión, y luego son enviados a través de un canal, el cual es un punto de incertidumbre en el sistema, ya que las condiciones y características del mismo pueden variar con el tiempo. Adicionalmente, se pueden presentar señales de interferencia de múltiples naturalezas (ruido).

Por las anteriores razones, se deben tomar decisiones en los sistemas de comunicación. Una de las múltiples decisiones que se pueden hacer es la escogencia del protocolo de comunicación a usar, debido a que cada protocolo de comunicación ofrece diferentes características y ventajas con respecto a los otros, y se comporta diferente ante las distintas condiciones del canal de comunicación.

2.2.2. Protocolos de comunicación

En las redes modernas de comunicación se ha impuesto un modelo por capas, que ha provisto de un marco de trabajo para el desarrollo de servicios y protocolos para permitir, entre otras cosas, la comunicación en internet. Ese modelo es el modelo ISO/OSI (ISO, Organización Internacional de Normalización, 1994)) El mismo posee siete capas, las cuales son: la capa física, de enlace, de red, de transporte, de sesión, de presentación y de aplicación. Dentro de la capa de transporte, dos protocolos principalmente han sido definidos: TCP y UDP, cada uno creado pensando en satisfacer necesidades diferentes. Ahora bien, en la actualidad existen un buen número de nuevos protocolos, algunos derivados de los dos anteriores, para responder a las necesidades comunicacionales actuales. Uno de ellos es el SCTP, el cual trata de mezclar las principales ventajas de TCP y UDP. A continuación se nombran algunas de sus principales características de esos tres protocolos (Stallings, 2004):

- TCP: (Transmission Control Protocol): es un protocolo orientado a conexión, este protocolo se encarga de establecer circuitos virtuales para la comunicación entre dos sitios, proporciona un número de secuencia para el orden y control de pérdidas, confiabilidad, detección de errores y control de congestión.
- UDP: (User Datagram Protocol): es un protocolo que no está orientado a conexión sino a mensajes, entre las características principales de UDP es que no requiere del establecimiento de conexión (handshaking), por lo tanto puede enviar los datos en cualquier momento.
- SCTP: (Stream Control Transmission Protocol): fue diseñado para proveer algunas de las características de TCP y UDP, es decir, es orientado a mensajes como el UDP, y

garantiza confiabilidad, al igual que TCP, debido a que posee número de secuencia y control de congestión.

2.2.3. Sistemas de comunicación autónomos

La comunicación autónoma es una extensión del concepto de computación autónoma, el cual es aplicado a las redes de comunicación. En sí, la comunicación autónoma debe pensar, analizar, tomar decisiones y reaccionar, para mantener las comunicaciones. La comunicación autónoma, por lo tanto, se basa en el mismo principio que la computación autónoma, de tal manera de plantearse como metas auto-configurarse, auto-optimizarse, auto-protegerse y auto-sanarse, con la mínima intervención humana, para adaptarse a los cambios en las condiciones de funcionamiento o a su contexto.

Un ejemplo de arquitectura adaptativa de comunicación autónoma es DAISY (Aguilar, Subias, Travé-Massuyés, & Zouaoui, GIIS 2012), la cual se muestra en la Figura 2-4. En esta arquitectura, el proceso de reconfiguración toma en consideración distintas entradas, las mismas se enumeran a continuación:

- Las propiedades P_o de los distintos protocolos especificados en una ontología
- El contexto inicial $C_x(t_0)$
- Las propiedades requeridas por la aplicación P_a , y en particular, el contexto que requiere C_a
- El estado actual del contexto $C_x(t)$, y las modificaciones sugeridas al contexto $\hat{C}_x(t)$

La arquitectura DAISY se basa en un sistema de reconocimiento de contexto, el cual supervisa y evalúa el contexto del sistema de comunicación, para proponer los cambios sobre el mismo.

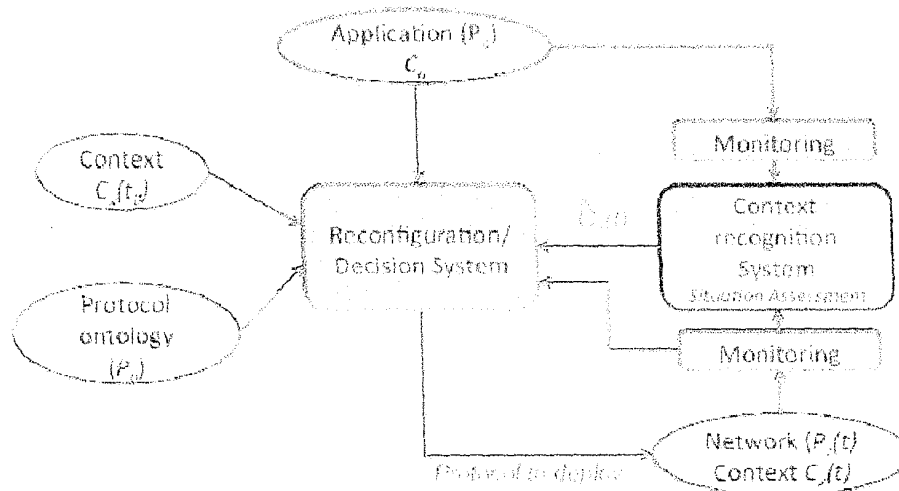


Figura 2-4 Arquitectura para auto adaptación (Aguilar, Subias, Travé-Massuyés, & Zouaoui, GIIS 2012)

La comunicación autónoma se le puede llamar también red autónoma, en caso de que se hable de comunicación en redes de datos. De hecho, la comunicación autónoma es un concepto más amplio que, además de los conceptos derivados de la computación autónoma, también se le pueden añadir algunos otros principios como el de manejo avanzado de la plataforma comunicacional en aspectos tales como: la virtualización de la red, la gestión especializada de ciertas aplicaciones sobre la red (telefonía, videoconferencia, etc.), los sistemas de apoyo a las comunicaciones (recuperación de desastres, etc.), la incorporación de la gestión de aplicaciones como servicios (SaaS), entre otros, como se pueden apreciar en la Figura 2-5

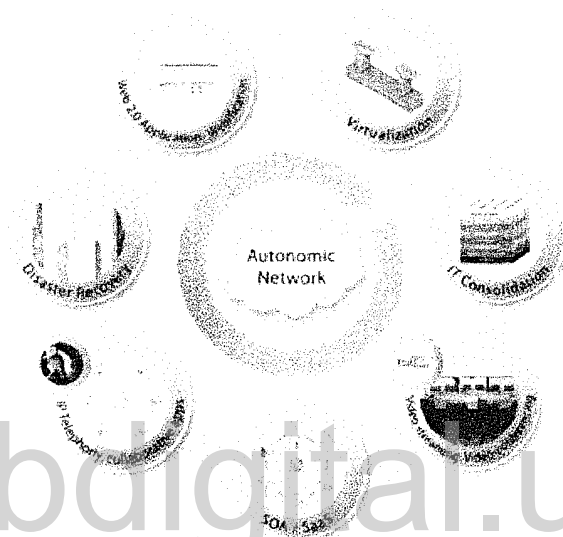


Figura 2-5 Comunicación Autónoma (Altamura, 2009)

2.3. *Redes Bayesianas*

A continuación se explicaran los tópicos básicos de las redes bayesianas que nos interesa desarrollar para el presente trabajo, entre ellos están: su proceso de inferencia y el problema de aprendizaje paramétrico y estructural.

2.3.1. **Conceptos Básicos**

Las redes bayesianas (RB), también conocidas como red de creencias, pertenecen a la familia de modelos de grafos probabilísticos (GMs probabilistic graph models). Estas estructuras de grafos dirigidos, son utilizadas para representar el conocimiento incierto como un modelo probabilístico multivariado, que describe un conjunto de variables aleatorias y sus influencias causales (Ben-Gal, Faltin, & Kenett, 2007). En las RB, cada

nodo representa una variable aleatoria, y los arcos representan las dependencias probabilísticas entre las variables (la ausencia de los arcos indica independencia condicional). La incertidumbre representada en las RB es propensa a ser modificada en base a las observaciones sobre el sistema real que se está supervisando. Las RB son comúnmente usadas en estadística, sistemas expertos, aprendizaje de maquinas, e inteligencia artificial. (Ben-Gal, Faltin, & Kenett, 2007)

Las RB suelen ser divididas en dos niveles (o dimensiones); el nivel cualitativo, el cual describe el grafo dirigido acíclico (directed acyclic graph DAG); y el nivel cuantitativo o numérico, que define las probabilidades condicionales en los arcos (Sucheta, 2001). Las RB se distinguen de las Redes de Markov debido a que las RB son GMs dirigidos. Las RB a nivel cualitativo se pueden descomponer en tres tipos básicos, según el proceso de propagación de probabilidades: seriales, causales divergentes, o causales convergentes (Kajaerulff, 2005) (ver Figura 2-6).

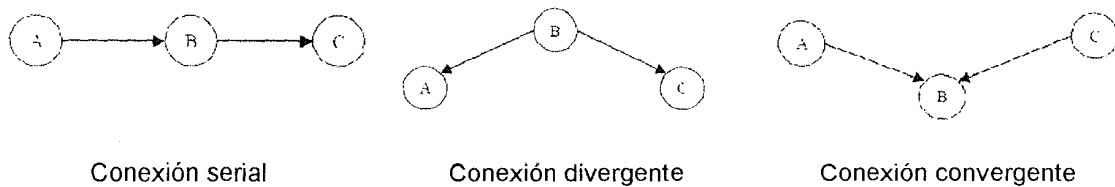


Figura 2-6 Tipos de propagación de probabilidades

Finalmente, una RB representa el conocimiento a través de las relaciones causales (estructura del grafo,) caracterizadas por tablas de probabilidades condicionales entre los nodos, por lo tanto, el conocimiento que representa la red está compuesto por los siguientes elementos (Kajaerulff, 2005):

- Un conjunto de nodos, que representan cada una de las variables del modelo.
- Un conjunto de enlaces o arcos entre aquellos nodos que tienen una relación causal.
- Una tabla de probabilidad condicional asociada a cada nodo, indicando la probabilidad de sus estados para cada combinación de los estados de sus padres.

2.3.2. Inferencia en Redes Bayesianas

El proceso de inferencia en las RB está fundamentado en el Teorema de Bayes (Kajaerulff, 2005). En las RB, la inferencia probabilística consiste en, dadas ciertas variables conocidas (evidencia), calcular la probabilidad posterior de las demás variables (desconocidas). Consecuentemente, este proceso se basa en calcular las probabilidades a posteriori de un conjunto de variables, después de obtener un conjunto de

observaciones desde el sistema supervisado. La inferencia en RB es un problema NP-Completo (Beltrá, 2002) y (Pearl, 1988)

El teorema de Bayes expresa la probabilidad condicional de un evento aleatorio h_i , dado x , en término de la distribución de probabilidad condicional del evento x dado h_i y la distribución de probabilidad marginal de sólo h_i (ver ecuación 1))

$$P(h_i|x) = \frac{P(x|h_i)P(h_i)}{P(x)} \quad (1)$$

Donde:

$P(h_i)$ Es la probabilidad a priori de h .

$P(x|h_i)$ Es la probabilidad de x dada la hipótesis h_i

$P(h_i|x)$ Es la probabilidad a posteriori de h_i dada la evidencia x .

En general, este modelo se puede extender considerando las restricciones T , de tal manera de determinar computacionalmente la hipótesis h acerca del problema que se está estudiando, dado los datos observados x las restricciones T , y las otras hipótesis sobre el problema (Tenenbaum, Griffiths, & Kemp, 2006) (ver ecuación 2)

$$P(h|x, T) = \frac{P(x|h, T)P(h|T)}{\sum_{h' \in H_T} P(x|h', T)P(h'|T)} \quad (2)$$

Donde el dominio T genera un espacio de hipótesis candidatas, denominadas h . La probabilidad $P(x|h, T)$ mide qué tan bien cada hipótesis predice los datos, y la probabilidad $P(h|T)$ manifiesta la verosimilitud de la hipótesis dado el conocimiento previo T . La probabilidad $P(h|x, T)$ es proporcional al producto de estos dos términos, que representan el grado de creencia en esta hipótesis, dada las restricciones previas T y lo observado en los datos x , dividido por la sumatoria del grado de creencia del aprendiz en cada hipótesis (Tenenbaum, Griffiths, & Kemp, 2006).

2.3.3. Aprendizaje en las redes bayesianas

El aprendizaje en las RB consiste en concebir un modelo de la estructura, y los parámetros asociados de la red, a partir de los datos del sistema a supervisar (Sucar, 2006) y (Pearl, 1988). De esta manera, el aprendizaje puede ser de dos formas (Sucar, 2006) :

- Aprendizaje estructural: Consiste en obtener la estructura o topología de la red.
- Aprendizaje paramétrico: Se basa en que, dada la estructura, se deben obtener las probabilidades asociadas.

○ *Aprendizaje Paramétrico*

Existen muchos métodos en la literatura para el Aprendizaje Paramétrico, sin embargo los más utilizados en la actualidad son (Ben-Gal, Faltin, & Kenett, 2007)((Norsys, 2010)):

- Aprendizaje por máxima verosimilitud
- Aprendizaje por método de maximización de la expectativa
- Aprendizaje por conteo

Aprendizaje por máxima verosimilitud (Maximum likelihood estimation MLE)

El estimador de máxima verosimilitud es un método clásico muy usado para esta tarea, partiendo del hecho de que se conoce la estructura y los datos son observables (Ben-Gal, Faltin, & Kenett, 2007). Es básicamente un método de estimación basado en la observación. El método lo que hace es que, dada la frecuencia de los datos, estima las probabilidades de la red.

El método consiste en encontrar el valor que maximice el logaritmo de la verosimilitud (L). Para este método se describe matemáticamente una red bayesiana de la siguiente forma (Friedman, Geiger, & Goldszmidt, 1997): Una red Bayesiana en un grafo acíclico dirigido que codifica una distribución de probabilidad conjunta sobre un conjunto de variables aleatorias U. Formalmente, la red bayesiana U es un par $B = (G, \theta)$. El primer componente G es el grafo acíclico en el cual sus vértices corresponden a las variables aleatorias X_1, \dots, X_n , y cuyos arcos representan dependencias directas entre las variables. El grafo G codifica las suposiciones de independencia de cada variable X_i . En cuanto la componente θ , representa el conjunto de parámetros que cuantifican la red. El mismo contiene un parámetro $\theta_{x_{ij}|\pi_{ij}} = P_B(x_{ij}|\pi_{ij})$ para cada posible valor x_{ij} de X_i el cual ocurre según la condición π_{ij} . Entonces, B define una única distribución de probabilidad conjunta (Friedman, Geiger, & Goldszmidt, 1997) (Ben-Gal, Faltin, & Kenett, 2007):

$$P_B(X_1, X_2, \dots, X_n) = \prod_{i=1}^n \prod_{j=1}^{k_i} \theta_{x_{ij}|\pi_{ij}} \quad (3)$$

Donde k_i es el número de valores posibles de la variable X_i . Este método requiere una colección de datos de entrenamiento compuesta por m casos, que en general se asumen independientes. Dada la colección de datos de entrenamiento $\Sigma = \{y_1 \dots y_m\}$, en donde cada y_l tiene definido un valor dado para cada X_i que se corresponde a uno de sus valores posibles x_{ij} , que ahora se llama en esa muestra x_{il} , tal que $y_l = (x_{l1} \dots x_{ln})^T$, y la colección de parámetros para cada variable $\Theta = (\theta_1 \dots \theta_n)^T$, donde θ_i es el vector de parámetros de la distribución condicional de la variable X_i , el logaritmo de la verosimilitud de la colección de datos de entrenamiento es (Ben-Gal, Faltin, & Kenett, 2007):

$$\log L(\theta, \Sigma) = \sum_{l=1}^m \sum_{i=1}^n \text{Log } P(x_{il}|\pi_{il}\theta_i) \quad (4)$$

Aprendizaje por método de maximización de la expectativa

El método de Aprendizaje por método de EM (EM, expectation maximization) es muy utilizado en el caso de existir nodos ocultos. Los nodos ocultos se definen

como los nodos donde no se conocen los valores que puede tener una variable. El algoritmo de este método es el siguiente (Sucar, 2006) (Ghahramani, 2003):

1. Iniciar los parámetros desconocidos (probabilidades condicionales) con valores aleatorios (o estimaciones de expertos). $\theta^{[t]} = \{\theta_1^{[t]}, \theta_2^{[t]}, \dots, \theta_c^{[t]}\}$
2. Utilizar los datos conocidos con los parámetros actuales $\theta_c^{[t]}$, para estimar los valores de las variables desconocidas (este paso es llamado también Paso E por la palabra estimación), donde t es el índice de la iteración, \mathcal{F} es la función auxiliar definida en la ecuación 5, $x = (x_1, x_2, \dots, x_n)$ son los datos observados, $z = (z_1, z_2, \dots, z_n)$ son las variables desconocidas, y q es una distribución sobre las variables z

$$\sum_z q(z|x) \log \frac{p(x,z|\theta)}{q(z|x)} \triangleq \mathcal{F}(q, \theta) \dots (5)$$

$$q_c^{[t+1]} \leftarrow \operatorname{argmax}_q \mathcal{F}(q, \theta_c^{[t]}) \quad (6)$$

3. Utilizar los valores estimados en el paso anterior $q_{[t+1]}$ para completar la tabla de datos. (llamado también Paso M por maximización).

$$\theta_c^{[t+1]} \leftarrow \operatorname{argmax}_\theta \mathcal{F}(q_c^{[t+1]}, \theta_c) \quad (7)$$

4. Repetir los pasos 2 y 4 hasta que no haya cambios significativos en las probabilidades.

Aprendizaje por conteo (Counting-Learning)

El aprendizaje por conteo es el método más simple y rápido que se puede utilizar, si no se poseen muchos datos faltantes o se tienen muchos datos inciertos. El método de conteo se puede utilizar para aprender o reforzar la tabla de probabilidad condicional (conditional probability tables CPT)

En el caso de aprendizaje, este método parte de una red que se encuentra en un estado de ignorancia total, en el cual todas las probabilidades son uniformes. El mecanismo permite aprender las probabilidades. El otro caso en que se utiliza este método es para reforzar el aprendizaje. En ambos casos se sigue el siguiente procedimiento (Norsys, 2010):

Para los nodos de la red en los cuales hay un hallazgo (por ejemplo, hay un hallazgo para un nodo (I_x) dado y sus padres (P_{x-z}), donde z es la cantidad de padres del nodo), se modifican los mismos (ver Figura 2-7), de forma tal que su experiencia y probabilidad individual sean modificadas (I_x) (se actualiza la probabilidad condicional desde sus padres al nodo (I_x)). La nueva experiencia $experiencia_{I_x}^n$ se obtiene al tomar el valor previo de $experiencia_{I_x}^{n-1}$, y sumar la

nueva creencia, para representar la nueva experiencia, como se muestra en la ecuación 8 (Norsys, 2010)

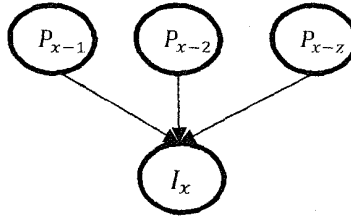


Figura 2-7 nodos involucrados en el aprendizaje por conteo

$$experiencia_{I_x}^n = experiencia_{I_x}^{n-1} + creencia \quad (8)$$

La creencia por lo general toma el valor de 1, pero la misma puede tener un valor mayor, como por ejemplo 2 si se presentan dos casos idénticos de una sola vez. Adicionalmente, puede tomar el valor de -1 si se desea que el caso se olvide (Norsys, 2010).

Seguidamente, dentro del vector de probabilidad, la misma es cambiada de $probabilidad_{I_x}^{n-1}$ a $probabilidad_{I_x}^n$, como se describe en la ecuación 9 (Norsys, 2010):

$$probabilidad_{I_x}^n = \frac{(probabilidad_{I_x}^{n-1} * experiencia_{I_x}^{n-1} + creencia)}{experiencia_{I_x}^n} \quad (9)$$

Las otras probabilidades de los hermanos de I_x en ese vector, cambian según la ecuación 9:

$$probabilidad_i^n = \frac{probabilidad_i^{n-1} * experiencia_{I_x}^{n-1}}{experiencia_{I_x}^n} \quad (10)$$

De esta forma, el nuevo vector queda normalizado de acuerdo al factor de $experiencia^n$

○ **Aprendizaje estructural**

El aprendizaje estructural consiste principalmente en obtener la estructura o topología de la red (DAG), es decir, hallar las relaciones de dependencia entre las diferentes variables. Estas relaciones pueden hallarse principalmente de dos formas; la primera consiste en obtener la estructura a partir de un experto, y la segunda consiste en utilizar minería de datos para obtener la estructura. En el segundo caso, y de acuerdo al tipo de estructura, podemos dividir los métodos en (Sucar, 2006):

- Aprendizaje de arboles
- Aprendizaje de poliarboles.

Para ambos casos de aprendizaje, de arboles o poliarboles, se plantea el problema como uno de optimización, y se debe obtener la estructura del árbol que más se aproxime a la distribución 'real'. Para la mayoría de los casos se usa el algoritmo de Chow-Liu (Ben-Gal, Faltin, & Kenett, 2007). El algoritmo de Chow-Liu asume que cada variable tiene a lo sumo una raíz (nodo padre), y se basa en aproximar una distribución de probabilidad a partir de un producto de probabilidades de segundo orden. En el algoritmo de Chow-Liu, la probabilidad conjunta de n variables se expresa por la ecuación:

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P(X_i | X_{j(i)}) \quad (11)$$

En donde $X_{j(i)}$ es el padre de X_i

Para obtener la estructura de la red se plantea la optimización de una medida de la diferencia de la información (DI) entre la distribución real P y la aproximada P^* , por medio de la ecuación 12. El objetivo es minimizar DI (Sucar, 2006)

$$DI(P, P^*) = \sum_x P(x) \text{Log} \frac{P(x)}{P^*(x)} \quad (12)$$

Se puede definir dicha diferencia en función de la información (I) mutua entre las variables, según la ecuación 13:

$$I(X_i, X_j) = \sum_{x_i, x_j} P(X_i, X_j) \text{Log} \frac{P(X_i, X_j)}{P(X_i)P(X_j)} \quad (13)$$

Por último, se encuentra el árbol óptimo mediante el siguiente algoritmo (Sucar, 2006) (Zaragoza, Sucar, & Morales, 2011):

1. Crear un grafo β vacío con n como el conjunto de vértices. η
2. Calcular la información mutua entre todos los pares de variables (En general). Para ello se utiliza la ecuación (13)
3. Ordenar las informaciones mutuas de mayor a menor.
4. Se agrega el arco con el mayor valor de la información mutua al grafo β .
5. Agregar el siguiente arco mientras no forme un ciclo, de lo contrario, desechar.
6. Repetir los pasos 3, 4 y 5 hasta que se cubran todas las variables η sean consideradas.

Para el caso de poliarboles se usa una variación del algoritmo de Chow-Liu. Se comienza con parte del esqueleto obtenido con el algoritmo de Chow-Liu, y se determina la dirección de los arcos utilizando pruebas de dependencia entre tripletas de variables, es decir, se realizan pruebas de independencias no solo entre dos variables, sino entre grupos de tres variables o tripletas. Dadas tres variables, existen tres posibles casos: arcos secuenciales $X \rightarrow Y \rightarrow Z$, arcos divergentes $X \leftarrow Y \rightarrow Z$, arcos convergentes $X \rightarrow Y \leftarrow Z$. En los dos primeros casos son distinguibles en base a las pruebas de independencia, Por ejemplo, X y Z son independientes o no de Y , lo que determina la

relación entre ellos dos. Sin embargo, para el caso de arcos convergentes las variables X y Z son marginalmente independientes (no logramos estar seguro de su relación). Se puede utilizar estas características, llamadas pruebas de dependencia de tripletas, para determinar las direcciones de los arcos entre estas tres variables, usando el siguiente algoritmo para el aprendizaje en poliarboles (Sucar, 2006):

1. Obtener el esqueleto utilizando el algoritmo de Chow y Liu.
2. Recorrer toda la red hasta encontrar una tripleta de nodos que sean convergentes ($X \rightarrow Y \leftarrow Z$), donde la variable a la que apuntan los arcos se le llama nodo multipadre Y .
3. A partir de un nodo multipadre, se determinan las direcciones de otros arcos utilizando la prueba de dependencia de tripletas, hasta donde sea posible (base causal).
4. Repetir los pasos 2 y 3 hasta que ya no se puedan descubrir mas direcciones.
5. Si quedan arcos sin direccionar, se debe utilizar semántica externa para obtener su dirección.

2.3.4. Toma de decisiones en Redes Bayesianas y Diagrama de influencia

La toma de decisiones en las redes bayesianas, se basa no solo en las probabilidades de un evento, sino también los posibles beneficios (utilidades) y los costo del mismo, para así poder determinar cómo afecta la posible decisión. La principal idea es poder encontrar un plan de acción que maximice los beneficios y minimice los costos. De esta forma, las redes bayesianas se han extendido para manejar la teoría de decisiones.

Para ello, la red bayesiana define diagramas de influencias, que es una versión extendida de ella. Los diagramas de influencia poseen tres tipos de nodos, los primeros corresponden a los nodos de las RB original (inciertos), los nodos correspondientes a las acciones a ser tomadas (decisión), y por último, los nodos correspondientes a la función de utilidad (ver Figura 2-8). El nodo de función de utilidades se representa con un rombo y las acciones como un cuadro.

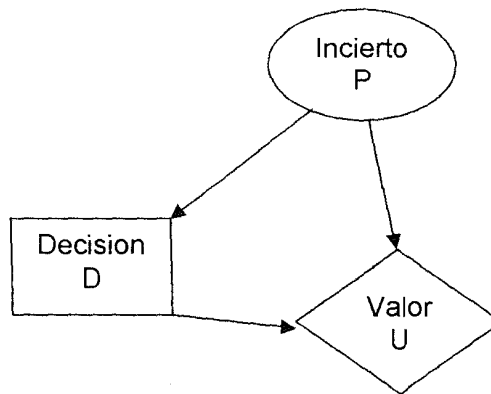


Figura 2-8 Ejemplo de Diagrama de influencia

Los diagramas de influencia (“influence diagram”, (Howard y Matheson 1981)) son utilizados para tomar decisiones que maximicen los beneficios de las acciones que se ejecutaran, basados en la utilidad esperada (MUE). Los diagramas de influencia son un medio eficaz para el modelado, la representación y la toma de decisiones bajo incertidumbre. En ese sentido, resolver un problema de decisión equivale a la determinación de la utilidad máxima esperada utilizando la estrategia elegida. La utilidad de ejecutar la acción a_i ($a_i \in A$, donde A son las alternativas a_1, a_2, \dots, a_m) bajo la hipótesis h_j es $\mu(a_i, h_j)$, donde $\mu(\)$ es la función de utilidad. De esta manera, la utilidad de ejecutar la acción a_i viene dada por la ecuación 15, en donde $P(\)$ representa la probabilidad condicional de H (conjunto de hipótesis h_1, h_2, \dots, h_n) dado ε , y ε es un conjunto de observaciones en la forma de evidencia en un momento dado (Kjaerulff & Madsen, 2005):

$$UE(a_i) = \sum_j \mu(a_i, h_j) P(h_j|\varepsilon) \quad (14)$$

La MUE se obtiene del máximo resultado de $UE(A)$, mediante la ecuación 14

$$a^* = \arg \max_{a_i \in A} EU(a_i) \quad (15)$$

Así, un diagrama de influencia puede ser considerado como una red bayesiana aumentada con las variables de decisión, las funciones de utilidad que especifican las preferencias en la toma de decisiones, y un orden de precedencia.

Específicamente, en la Figura 2-9.a se muestra el diagrama de influencia para el problema de encendido de un carro. Solo se toma por simplicidad que las posibles causas del problema son la falta de combustible y las bujías sucias. Adicionalmente, también se asume que además de los problemas de encendido, la única observación que se puede hacer es ver el nivel de combustible. La Figura 2-9.b muestra la estructura de la red aumentada, donde los vínculos del **Nivel de combustible** y de **Arranque** se enlazan a la

Acción, para indicar que los estados de nivel de combustible y arranque se requieren para tomar la decisión sobre la acción a realizar, según la función U (Kjaerulff & Madsen, 2005). Esta función U, a su vez, para poder ser evaluada, depende de la cantidad de **combustible** y de las **Bujías**.

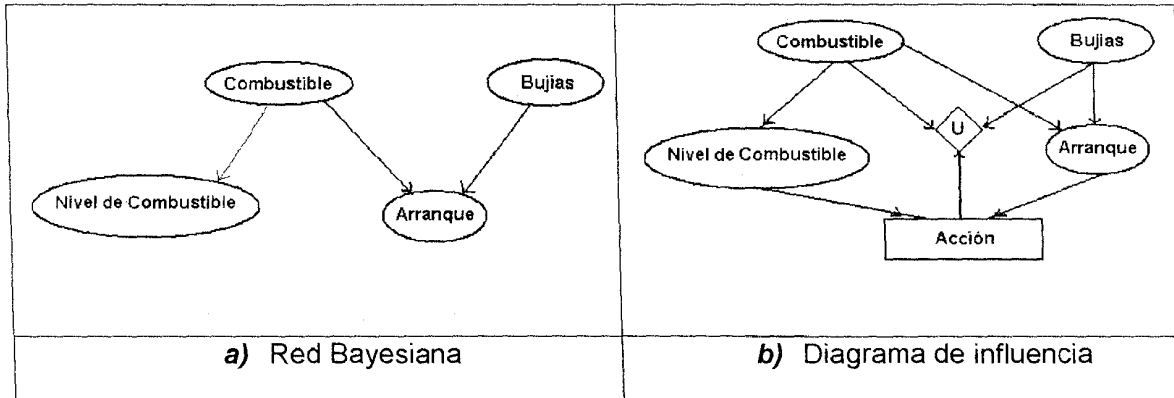


Figura 2-9 Red Bayesiana y Diagrama de influencia (Kjaerulff & Madsen, 2005)

En la Tabla 2-1 se muestra la función de utilidad U del ejemplo. En la tabla deben existir tres entradas para este ejemplo que son combustible, bujías, y la acción. En cuanto a las acciones posibles son: sin acción, agregar combustible y limpiar bujías. En la Tabla 2-1 se le asigna el valor de 1 a las condiciones de agregar combustible y/o limpiar bujías, en donde dichas acciones contribuyen a resolver el problema, y se asigna el valor 0 a las acciones que no contribuyen a la solución. Además, se agrega otra acción, que es no hacer nada. El 1 en esa acción significa el caso que especifica que no se debe hacer nada para resolver el problema (porque no hay problemas).

Combustible	Bujías	Acción		
		Sin acción	Agregar combustible	Limpiar bujías
No	Sucias	0	1	1
No	Limpias	0	1	0
Si	Sucias	0	0	1
Si	Limpias	1	0	0

Tabla 2-1 Función de utilidad (Kjaerulff & Madsen, 2005).

Esa tabla, a partir de las entradas, determina las acciones posibles en un momento dado (las que tengan el valor de uno). Si se quisiese cuantificar la función de utilidad, se le podrían colocar valores monetarios a las acciones de llenado de combustible y de limpiado de las bujías.

Formalmente, un diagrama de influencia es un grafo dirigido acíclico DAG que contiene N nodos representando las variables del problema de decisión $V = X \cup D \cup U$. Cada variable tiene su propio dominio $\text{dom}(v)$, y un conjunto de padres v_i denotado por π_{v_i} . Los tres tipos de variables de un diagrama de influencia son (Crowley, 2004):

- Nodos Inciertos, $x_i \in X$, representa las variables aleatorias del diagrama. Estas tiene asociada una tabla de probabilidades condicionales (CPT, Conditional Probability Tables), representando una distribución $P(x_i, \pi_{x_i})$ (los nodos inciertos del diagrama de influencia son una red bayesiana)
- Nodos de Decisión $d_i \in D$, representan las decisiones a tomar, las reglas de decisión δ_i representan el mapeado de los nodos padres a una decisión, $\delta_i(\pi_{d_i}) \in \text{dom}(d_i)$
- Nodos de Utilidad, denotados por $u_i \in U$, representan las funciones que resultan del mapeo de los nodos padres con una valor de utilidad $val_u(\pi_{v_i}) \in \text{dom}(u_i)$. ninguna de los nodos de utilidad tiene descendencia.

La CPT (tabla de probabilidades condicionales) representa la información cuantitativa de la red probabilística, la misma es representada como un número real en el intervalo $[0,1]$. La CPT es una simple tabla, la cual posee un valor numérico para cada posible combinación de un nodo padre con sus nodos hijos, que representa la probabilidad condicional de que ocurra el hijo dado que ocurre el padre (teorema de Bayes). Todas las probabilidades que salen de un nodo padre deben sumar la unidad "1" (Norsys, 2010). En la Figura 2-10 se puede observar una red bayesiana con su tabla CPT (Koski & Noble, 2009)

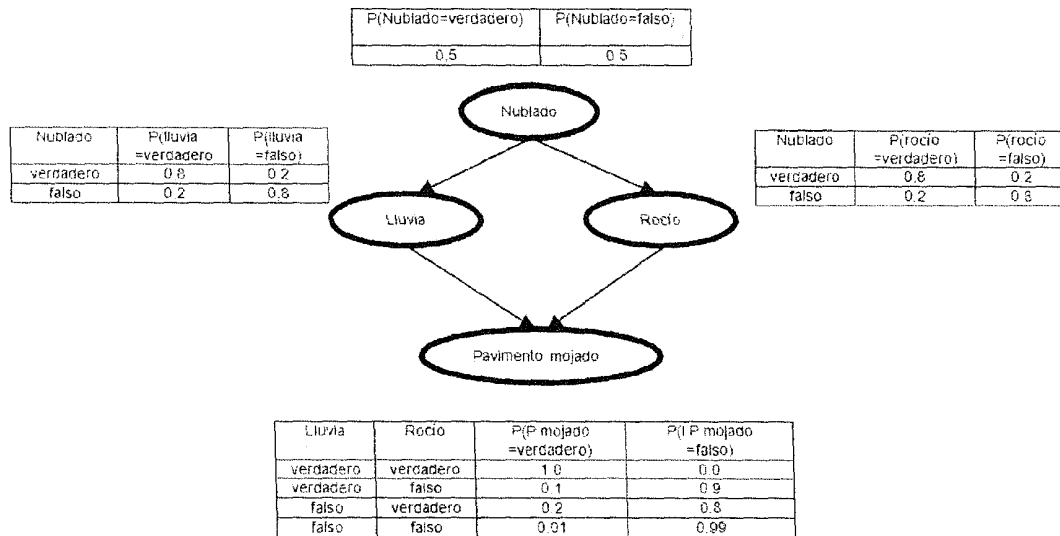


Figura 2-10 Ejemplo de Tabla CPT

En un diagrama de influencia se busca, en general, tomar la decisión óptima, que es la que maximiza el valor de utilidad. Cuando no existe una cuantificación (como el caso de la Tabla 2-1), sencillamente se limita a escoger las acciones posibles.

2.4. Ontologías

En nuestro trabajo, el componente de representación del conocimiento se representa mediante una ontología. Esa ontología se utilizara como base de conocimiento para construir la red bayesiana (Hudec & Fenz, 2009) (Helspe & van der Gaag, 2002). Para integrar esta ontología con el sistema autónomo se toman algunos conceptos del paradigma de arquitecturas dirigidas por ontologías (ODA); para ello se explican los conceptos pertinentes.

2.4.1. Ontología: conceptos básicos

La palabra Ontología proviene de la Filosofía, para explicar sistemáticamente la existencia. En un contexto computacional, una ontología es una representación formal de conocimiento, como una serie de conceptos relacionados entre sí, estos conceptos poseen un dominio específico (Gasevic, Djuric, & Devedzi, 2006). El uso de las ontologías permite definir un vocabulario en común, para así poder definir tipos, propiedades e interrelaciones entre los conceptos.

Los usos comunes de las ontologías se pueden agrupar principalmente en:

- Facilitar la comunicación entre diferentes entes.

- Compartir y reutilizar el conocimiento

Para nuestro caso, una ontología es un modelo que representa un dominio, el cual usamos para razonar sobre él. La ontología permite especificar formalmente a los objetos que existen en un dominio en particular, y como se relacionan entre ellos.

2.4.2. Ingeniería dirigida por modelos

La Ingeniería dirigida por modelos (Model Driven Engineering MDE) es un paradigma que se utiliza para el desarrollo de software usando modelos (Gómez-Montalvo, 2012) (Gasevic, Djuric, & Devedzi, 2006). El MDE contribuye a incrementar la productividad y la compatibilidad entre los sistemas debido al reusó de modelos estandarizados. El MDE plantea hacer ingeniería de modelos, que se refiere a una serie de enfoques de desarrollo que se basan en la utilización de software de modelado como una forma primaria de expresión. El MDE hace una abstracción del sistema real en un modelado basado en varias normas, tales como Facilidad Meta-Objetos (Meta Object Facility MOF), Lenguaje de especificación (Object Constraint Language OCL), y Lenguaje Unificado de Modelado (Unified Modeling Language UML). (Gómez-Montalvo, 2012).

La Arquitectura dirigida por modelos (“Model Driven Architecture”, MDA) es un campo de la MDE. Uno de los principales conceptos de MDA es separar la especificación de los sistemas de los detalles de su implementación en una determinada plataforma. MDA facilita la interoperabilidad en entornos heterogéneos, y proporciona una metodología para la integración de sistemas. Esto se logra mediante el uso de varias capas en el modelado definido por el MDA, las cuales son (MDA Guide Version 1.0.1, 2003):

- Modelo independiente computacional (computational-independent model CIM), el cual se centra en caracterizar el medio ambiente y los requisitos del sistema.
- Modelo independiente de la plataforma (platform-independent model PIM), este se centra en el funcionamiento de un sistema, sin considerar los detalles necesarios para su implementación en una plataforma determinada.
- Modelo específico a la plataforma (platform-specific model (PSM), extiende el modelo PIM considerando los detalles para implementar el sistema en una plataforma específica.

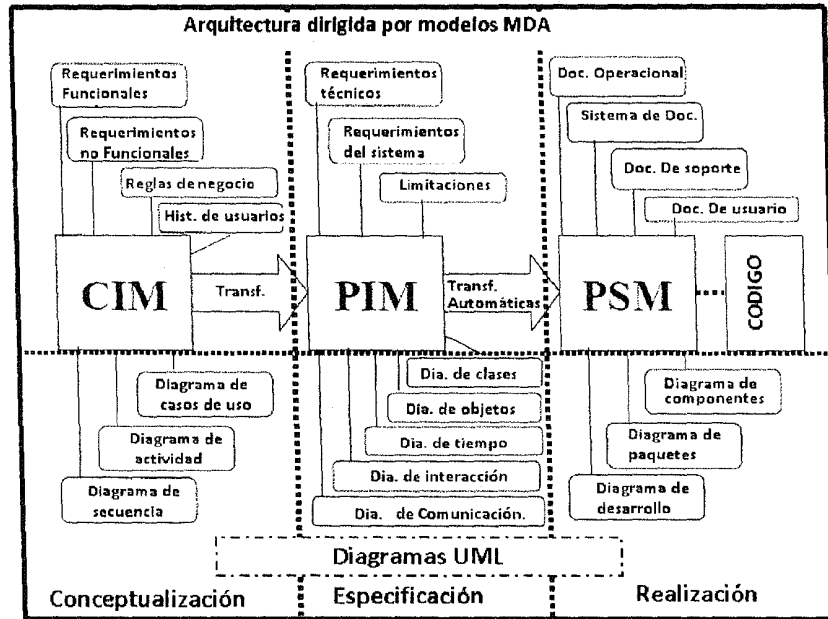


Figura 2-11 Arquitectura dirigida por modelos (extraído de (Singh & Sood, December 2010))

En la Figura 2-11 Se observan las tres capas de la arquitectura MDA, con sus respectivos procesos y transformaciones, así como los diagramas UML que usan. En general, la principal idea detrás de MDA es tratar de utilizar técnicas de transformación para convertir los modelos independientes de la plataforma, que especifican la operación de un sistema, en modelos específicos a una plataforma que indiquen los detalles de cómo usar las capacidades del sistema y de la plataforma para operar sobre la misma.

En los últimos años, ha sido propuesto por el consorcio W3C (World Wide Web Consortium W3C) una extensión del MDA basada en ontologías, denominado Arquitecturas Dirigidas por Ontologías (Ontology-Driven Architecture ODA). ODA emplea ontologías para construir modelos. ODA lo que nos indica es que se usan ontologías en las tres capas, para la conceptualización, la especificación y el diseño del sistema (ver figura 2-10).

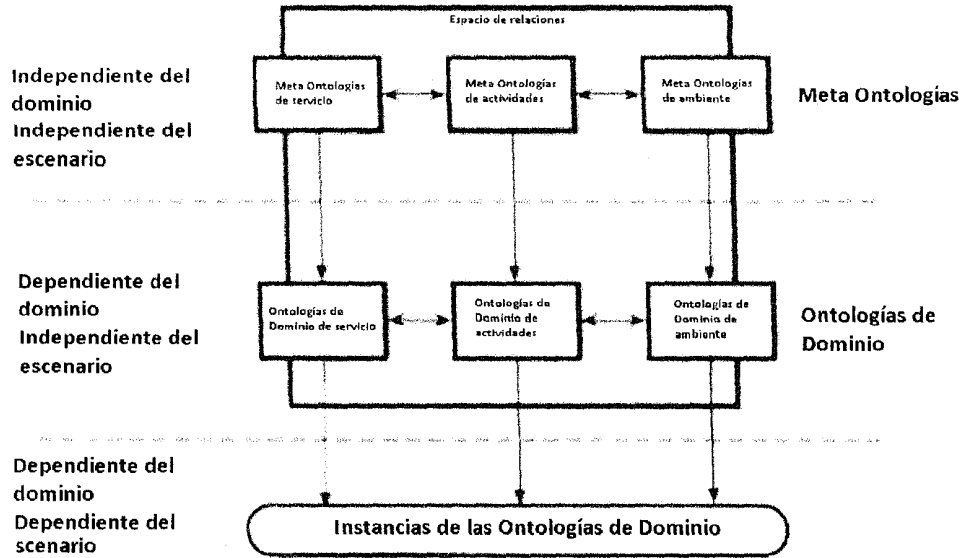


Figura 2-12 Relaciones entre la meta ontología y las ontologías (Pittarello & Celentano)

Así, en ODA las ontologías son usadas como modelos conceptuales para el desarrollo de los componentes del software, y como modelos de los componentes en tiempo de ejecución (Anandaraj, Dheepak, & Raja, 2011). En específico, las meta-ontologías representan las ontologías en un nivel superior (es decir, el nivel CIM en ODA), las cuales se utilizan como referencias ya que representan marcos conceptuales, estándares, etc.; las ontologías de dominio especifican todo el conocimiento requerido por el software a desarrollar; mientras que las instanciaciones son el uso de esas ontologías en la implantación del software.

En general, ODA combina técnicas de representación formal diseñadas para la web semántica, junto con prácticas utilizadas en ingeniería de software. El uso de ODA aporta muchas ventajas, entre las cuales podemos mencionar: lógica formal para la descripción de los componentes, razonamiento y búsqueda basada en ontologías, especificación del sistema usando ontologías para la posterior verificación de su consistencia (durante el desarrollo o en tiempo de ejecución), entre otras cosas.

2.4.3. Meta-Ontologías para los sistemas de comunicación

A continuación se presentan las distintas Meta-Ontologías utilizadas en el presente trabajo. Estas meta ontologías corresponden al nivel CIM (independiente del dominio), es decir a nivel de conceptualización.

La ontología de la Figura 2-13 está basada en el modelo de Shannon y Weaver, el cual es el modelo de comunicaciones usado. Este modelo involucra los conceptos de origen,

destino de la información, canal de comunicación, ruido sobre el canal de comunicación, transmisor, receptor, ruido y la información transmitida.

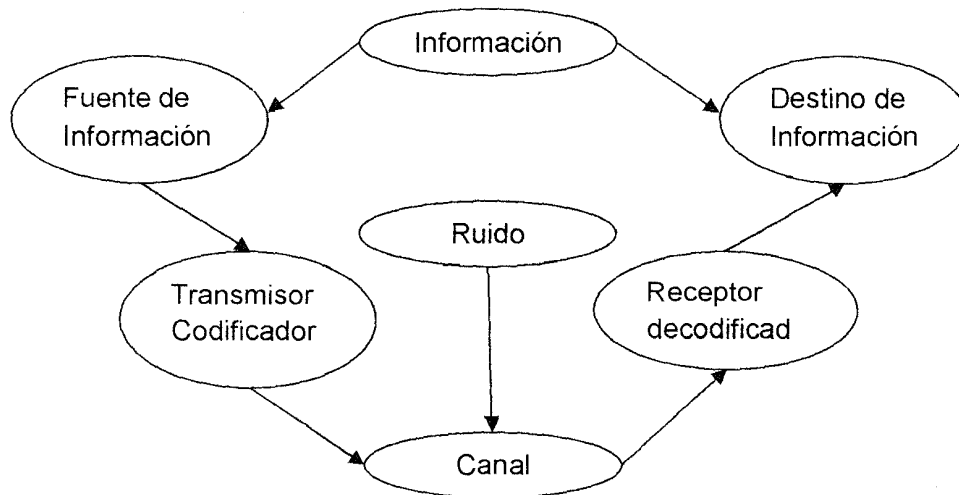


Figura 2-13 Ontología de Shannon y Weaver

A partir del modelo de Shannon y Weaver se definió la ontología de la para los sistema de comunicación. Sin embargo, en esta ontología no son tomados en cuenta de forma explícita algunos aspectos, los cuales son descritos por otras ontologías. Algunos de esos aspectos tienen que ver con los problemas de calidad de servicio, tipos de protocolos, etc., importantes en las redes modernas de comunicación. Esas ontologías se muestran en las Figura 2-14, Figura 2-15 y Figura 2-16

En la Figura 2-14 se representa una ontología sobre calidad de servicio (QoS) a nivel de la capa transporte, la misma integra los requerimientos de las aplicaciones, mecanismos, funciones y protocolos de la capa de transporte, entre otras cosas (Exposito, 2010). La Figura 2-15 define una ontología que representa los parámetros de calidad de servicio, con criterios de rendimiento como consumo de potencia, capacidad, etc. (Consortium, 2009). Por último, la Figura 2-16 muestra una ontología que caracteriza la gestión de los contenidos multimedia en los sistemas de comunicación (Exposito, Lamolle, & Gómez-Montalvo, 2010) Esta ontología se basa en protocolos de la IETF, e integra el control de sección, la descripción de la sección y el protocolo de transmisión

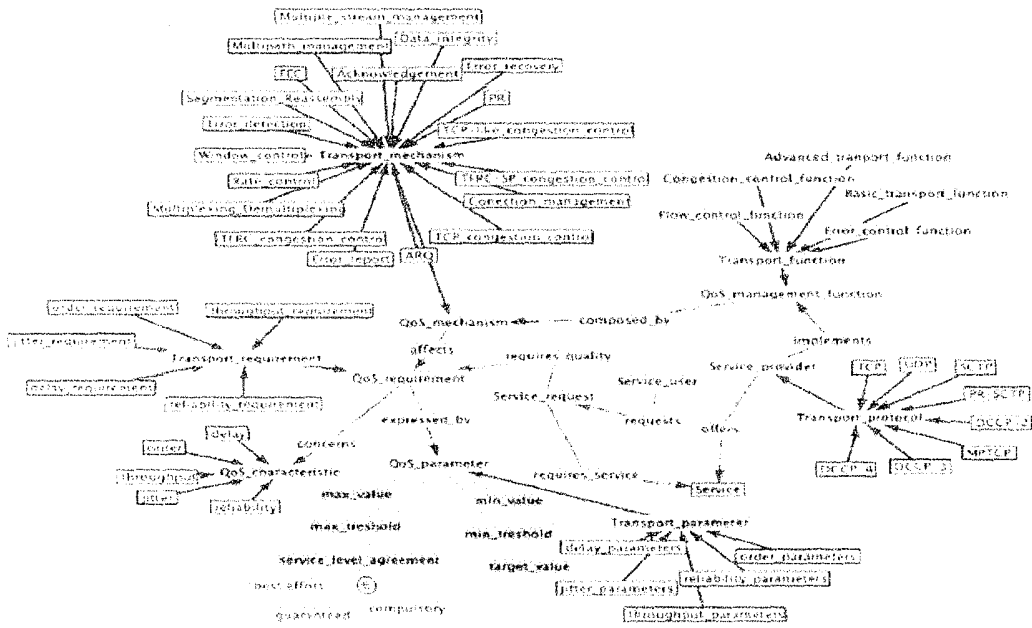


Figura 2-14 Ontología para la capa de transporte (Exposito, 2010)

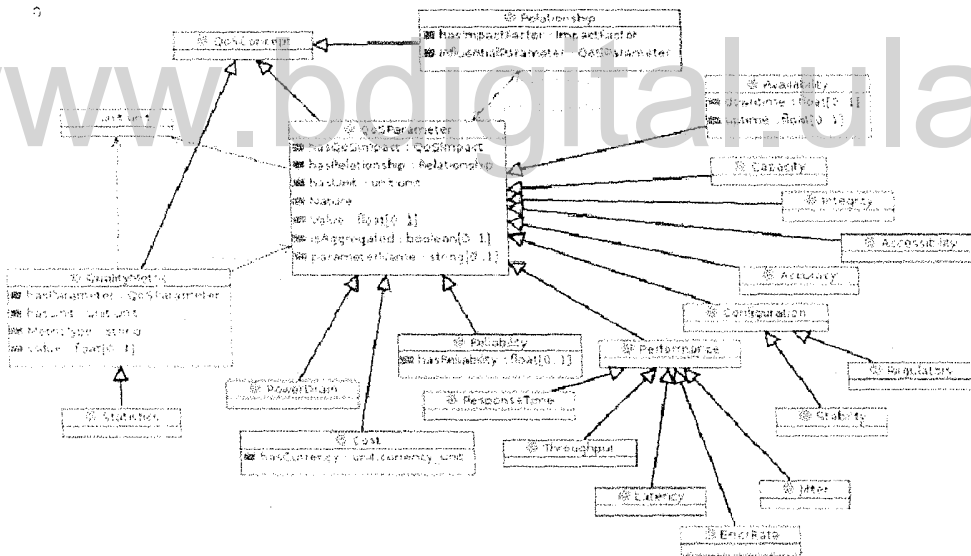


Figura 2-15 Ontología del proyecto Hydra (Consortium, 2009)

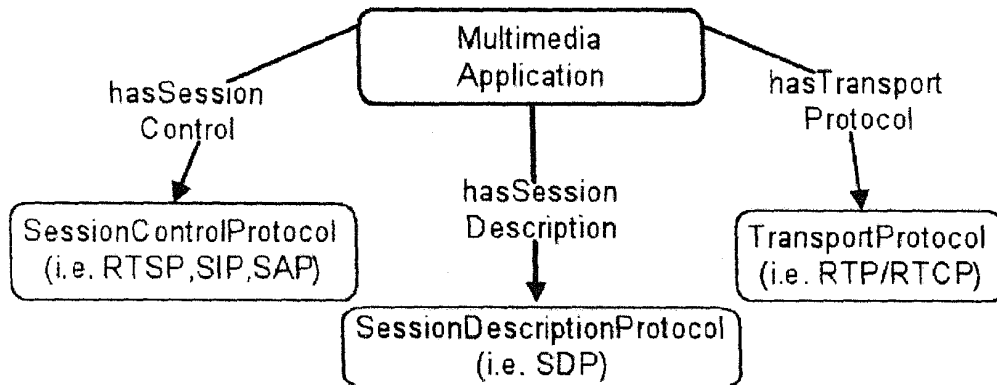


Figura 2-16 Ontología de protocolos de control basado en IETF, para el caso de contenidos multimedia

Todas esas ontologías conforman la capa CIM de nuestro sistema de toma de decisiones.

2.4.4. Proceso de transformación de una ontología a una red bayesiana

Existen múltiples trabajos de investigación que abordan la transformación de una ontología en una red bayesiana, entre ellos podemos mencionar los siguientes trabajos de (Helspe & van der Gaag, 2002), (Hudec & Fenz, 2009) y (Rong, Zhongli, & Peng, 2006). Nosotros en particular, en este trabajo usamos el marco de trabajo BayesOWL.

○ *Introducción a BayesOWL*

El marco de trabajo BayesOWL (Rong, Zhongli, & Peng, 2006) provee una serie de reglas y procedimientos para la transformación directa de una ontología descrita en OWL en una red Bayesiana. BayesOWL permite:

- Representar información de OWL en probabilidades
- Traducción de ontologías OWL a RB
 - Reglas para construir la estructura de la RB desde tripletas OWL
 - Ayuda para la adición de nodos especiales para las relaciones lógicas
 - Construcción de tablas CPT
 - Incorporar la información de probabilidad local

○ *Traducción estructural*

Para la traducción estructural de una ontología descrita en OWL en una Red Bayesiana se usan reglas. El principio general de las reglas de conversión es que todas las clases son transformadas en nodos en la red bayesiana (estas clases se especifican como sujetos y objetos en la tripleta RDF), y un arco es dibujado entre dos nodos solo si los dos nodos están relacionados por un predicado descrito en OWL, con dirección de superclase a subclase (Rong, Zhongli, & Peng, 2006). Adicionalmente se crea una clase especial de nodos, L-Nodos, también llamados nodos de control, los cuales se crean para facilitar el modelado de las relaciones lógicas entre las clases descritas en OWL. Los nodos de control contienen conexiones convergentes desde cada uno de los nodos conceptos involucrados en esa relación lógica. Existen cinco clases de nodos de control,

correspondiente a las relaciones lógicas “and” (owl:intersectionOf), “or” (owl:unionOf), “not” (owl:complementOf), “disjoint” (owl:disjointWith), y “same as” (owl:equivalentClass).

Las reglas de BayesOWL plantean cinco pasos para la **traducción estructural**, descritos a continuación (Rong, Zhongli, & Peng, 2006):

1. Cada primitiva o concepto clase C definido, es mapeado en variables de dos estados (verdadero o falso) en la red bayesiana.
2. El constructor “rdfs:subClassOf” es modelado por un arco directo del nodo de la clase padre (parent superclass) a cada nodo de la subclase hijo. Por ejemplo, un concepto de clase C definido con una clase del súper-concepto C_i ($i = 1, \dots, n$) por “rdfs:subClassOf” es mapeado en una subred en la RB traducida con una conexión que converge de cada C_i con C, como se muestra en la Figura 2-17

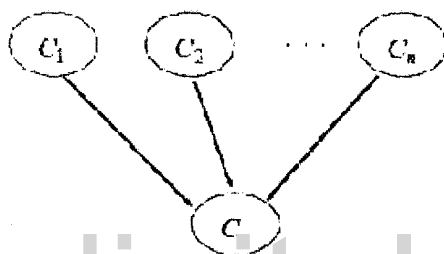


Figura 2-17 rdfs: subClassOf

3. Una clase concepto C definida por la operación de intersección de las clases C_i ($i = 1, \dots, n$) usando el constructor “owl:intersectionOf” es mapeado en una subred (ver Figura 2-18), en la traducción en la RB, con una conexión convergente de cada C_i a C, y una conexión convergente de C y cada C_i a un nodo llamado LNodeIntersection.

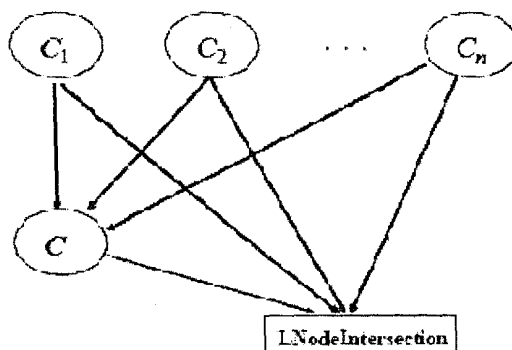


Figura 2-18 owl: intersectionOf

- Una clase concepto C definida por la operación Unión (`owl:unionOf`) de las clases C_i ($i=1, \dots, n$) es mapeada en una subred (ver Figura 2-19), en la traducción de la RB, con una conexión convergente de C a cada C_i , y una conexión convergente de cada C_i a un nodo llamado `LNodeUnion`.

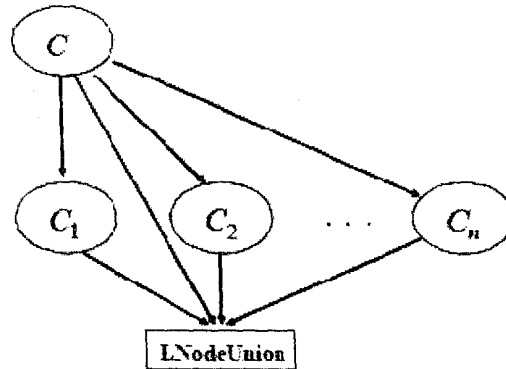


Figura 2-19 owl: unionOf

- Si dos conceptos de clase C_1 y C_2 están relacionadas por alguna relación como complemento (`owl:complementOf`), equivalente (`owl:equivalentClass`), o disjuntos (`owl:disjointWith`), entonces se crea un nodo de control en la red bayesiana, y se crean enlaces dirigidos de C_1 y C_2 a ese nodo.



Figura 2-20 owl:complementOf, owl:equivalentClass, owl:disjointWith

Basado en las reglas previamente definidas, la RB traducida contendrá dos clases de nodo, los nodos de clase regular para los conceptos, y los L-nodos, los cuales son nodos puente asociados a relaciones lógicas. Por medio del uso de los L-Nodos, la relación "rdfs:subClassOf" es diferenciada de las otras relaciones lógicas. Así, a partir de esas relaciones se construyen las tablas CPT del nodo C .

La comprobación de la valides de esas reglas es presentada en (Rong, Zhongli, & Peng, 2006). La implementación actual del algoritmo propuesto en BayesOWL solo funciona para ontologías taxonómicas

- **Construcción de las tablas CPT**

Para construir las CPTs de la Red Bayesiana se deben tomar en consideración los dos tipos de nodos: los nodos lógicos (o de control, llamados también L-nodos) y los nodos de concepto. A continuación se especifican los pasos para dicha transformación (Rong, Zhongli, & Peng, 2006):

- Tabla CPT para L-nodos (son cinco tipos): se especifican de la siguiente forma:
 - LNodeComplement: la relación complemento entre C_1 y C_2 se calcula como LNodeComplement verdadero si $c_1 \bar{c}_2 \vee \bar{c}_1 c_2$, de lo contrario es falso.
 - LNodeDisjoint: la relación disjuncto entre C_1 y C_2 se calcula como: LNodeDisjoint verdadero si $c_1 \bar{c}_2 \vee \bar{c}_1 c_2 \vee \bar{c}_1 \bar{c}_2$, de lo contrario es falso.
 - LNodeEquivalent: la relación equivalente entre C_1 y C_2 se calcula como: LNodeEquivalent verdadero si $c_1 c_2 \vee \bar{c}_1 \bar{c}_2$, de lo contrario es falso.
 - LNodeIntersection: la relación C que intercepta a C_1 y C_2 se calcula como: LNodeIntersection verdadero si $c_1 c_2 \vee \bar{c}_1 \bar{c}_2 \vee \bar{c}_1 c_2 \vee c_1 \bar{c}_2$, de lo contrario es falso.
- Tabla CPT para los nodos de concepto
 - Para calcular la tabla CPT, es decir calcular $P(C|\pi_c)$ para cada variable C de cada nodo concepto en la estructura (DAG), en donde π_c es el conjunto de todos los padres de C , se procede de la siguiente manera:
 - Determinar probabilidad marginal $P(C)$ o
 - Determinar la probabilidad condicional $P(C|\pi_{c_i})$, en donde $\pi_{c_i} \subseteq \pi_c$, (Rong, Zhongli, & Peng, 2006). La distribución de probabilidad para una entrada C indica el grado de inclusión de C en sus padres, y debería ser usada para llenar la CPT, de tal forma que sea consistente con las restricciones probabilísticas correspondientes para el caso, estas restricciones pueden ser tomadas de la ontología y/o aprendidas de los datos experimentales (Rong, Zhongli, & Peng, 2006).

3. Diseño del Sistema

En este capítulo se detallan todos los elementos que se utilizan para el diseño del sistema de toma de decisiones para el sistema autónomo de comunicaciones

3.1. *Arquitectura propuesta para el sistema*

La arquitectura propuesta para la implementación se basa en la arquitectura DAISY mostrada en la figura 2.-4. En particular, en este trabajo se rehacen el sistema de reconocimiento del contexto y el sistema de decisión de reconfiguración (ver Figura 3-2). Este ultimo sistema es una separación que se hace en el sistema de reconfiguración de la figura 2-4, el cual está compuesto por dos partes, uno que toma la decisión de reconfigurar y qué hacer en particular, y otro que ajusta el protocolo de la capa de transporte en el emisor, de tal forma de optimizar los criterios de rendimiento de la comunicación (es decir, aumentar el caudal, disminuir la latencia, disminuir los errores, entre otros). Para lograr este objetivo, se deben obtener las trazas del sistema comunicacional supervisado, y calcular a partir de ellas todas medidas de rendimiento relacionados con la calidad de servicio (Quality of Service QoS). Esto se realiza en la parte de monitoreo de la Figura 3-1

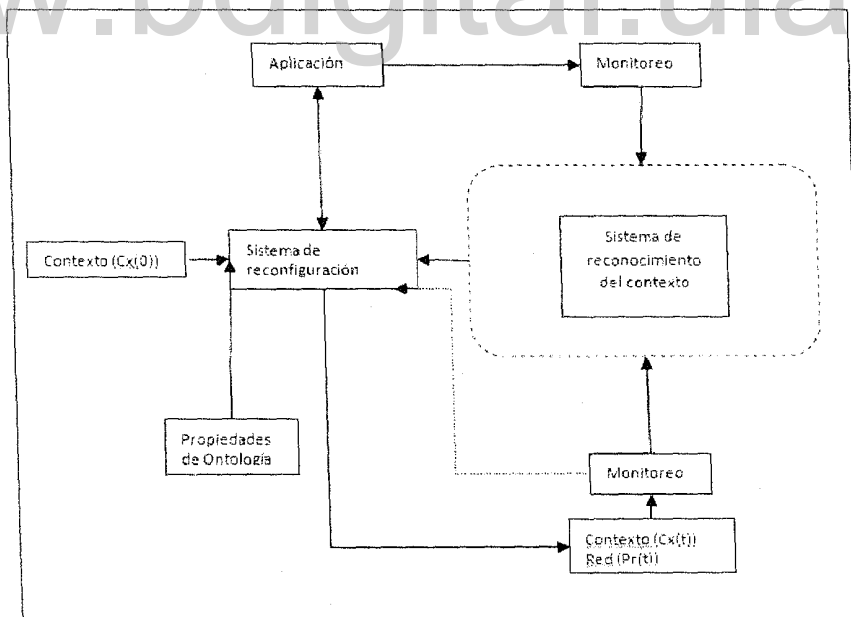


Figura 3-1 Arquitectura del sistema de comunicación autónómico

3.1.1. Sistema de toma de decisión de reconfiguración

Este sistema forma parte del Sistema de reconfiguración de la figura 3-1, el cual está compuesto por dos componentes: el sub-sistema de decisión de reconfiguración (toma la decisión de reconfigurar y qué hacer en particular), y el sub-sistema de ejecución de la reconfiguración (ajusta el protocolo de la capa de transporte en el emisor).

El sub-sistema de decisión de reconfiguración (ver Figura 3-2) se basa principalmente en un diagrama de influencia, como elemento principal para la toma de decisiones. El mismo es alimentado por los parámetros observados del sistema comunicacional supervisado (estos parámetros son: caudal, retardo, tasa de pérdidas, entre otros), los cuales son obtenidos mediante el análisis de las trazas de la simulación (subsistema de cálculo del desempeño). Dichos parámetros también contribuyen a actualizar el diagrama de influencia mediante el aprendizaje clásico (aprendizaje por conteo), de esta forma el sistema se va actualizando en base a las distintas experiencias que contribuyen positivamente con el sistema comunicación. Los resultados de la simulación son analizados en busca de problemas en la comunicación. Por otro lado, el diagrama de influencia toma la decisión si debe haber o no una reconfiguración en el sistema de comunicación.

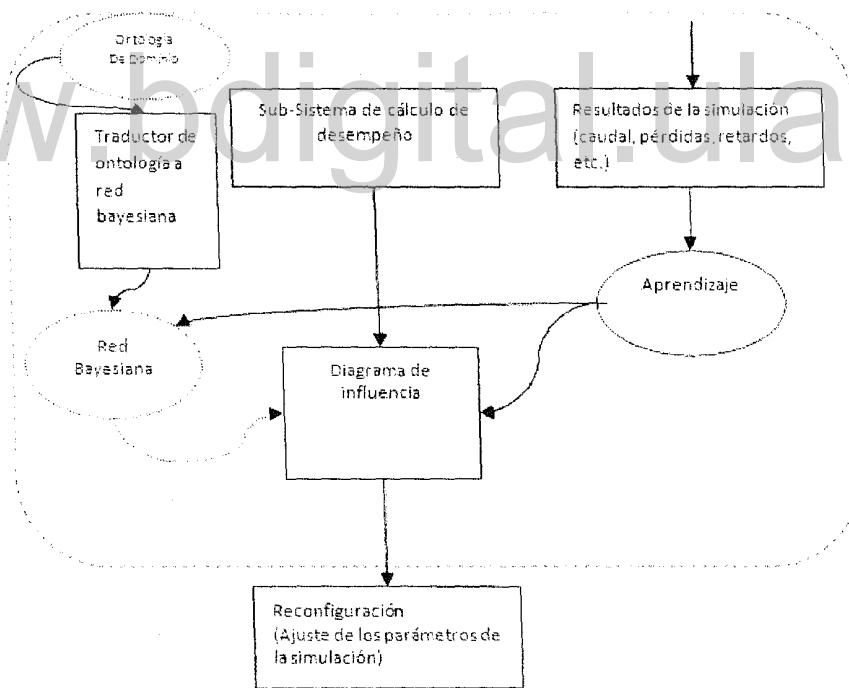


Figura 3-2 Sub-Sistema de decisión de reconfiguración

3.1.2. Sistema de reconocimiento de contexto

Básicamente, este sistema está compuesto por la ontología de dominio y la red bayesiana. Este subsistema monitorea el sistema de comunicación, y con el conocimiento que tiene instancia las reglas respectivas contenidas en la ontología, que se llevan a la red bayesiana, que caracterizan el contexto actual. La caracterización ontológica es mostrada en la sub-sección siguiente

3.2. Ontología de dominio

Las ontologías de dominio son las ontologías que se derivan de las Meta ontologías, que son específicas a nuestro sistema de comunicación autónomo. Las Meta ontologías fueron descritas en la sección 2.4.3, estas definen el nivel CIM (paradigma ODA), para caracterizar todos los estándares que se usan. Ellas ahora se usan para conformar una sola ontología que cumpla con el siguiente requisito: caracterizar el sistema de comunicación autónoma. Para ello, vamos a ir analizando cada meta-ontología. Primeramente, es solo necesario considerar una parte de dos de las meta-ontologías presentadas en el capítulo anterior, las cuales se muestran en un cuadro en las Figura 3-4 y Figura 3-3, respectivamente

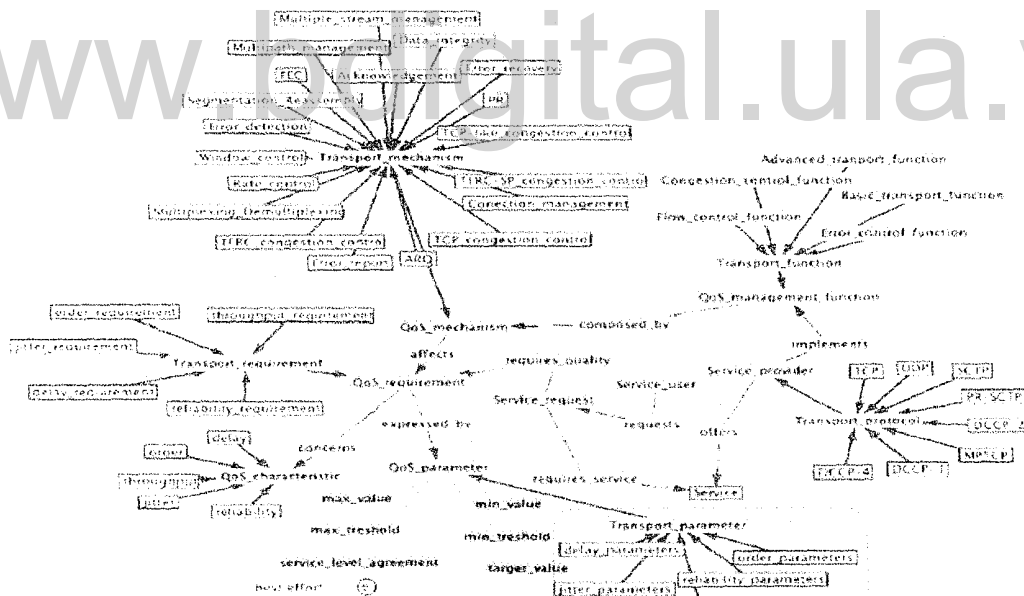


Figura 3-3 Ontología de Concepción de modelos de transporte de nueva generación

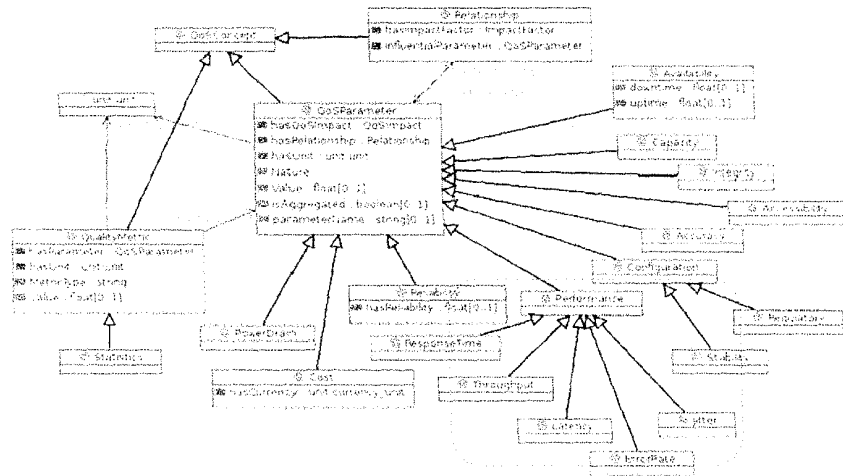


Figura 3-4 Ontología del proyecto Hydra

Así, los elementos relevantes de las meta-ontologías a considerar en nuestro trabajo se muestran en las Figura 3-5 y Figura 3-6. Estos fragmentos caracterizan los parámetros de calidad de servicio, y su relación ontológica con el rendimiento en general de la comunicación en las redes de comunicación. Los parámetros de calidad de servicio describen (modelan) las características básicas del canal de comunicación, ya que por ejemplo dan cuenta del retardo (delay o latency), del caudal (Throughput), entre otras cosas.

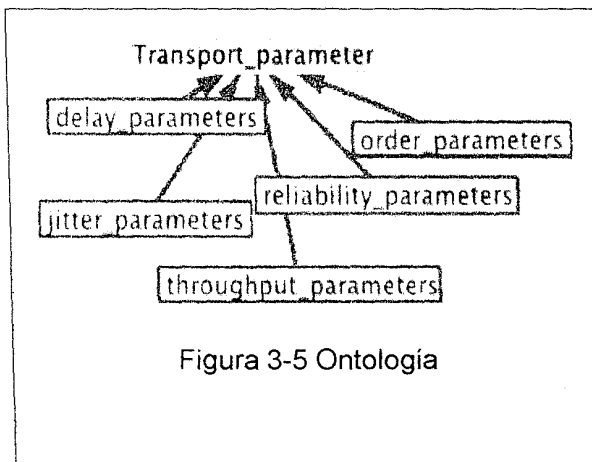


Figura 3-5 Ontología

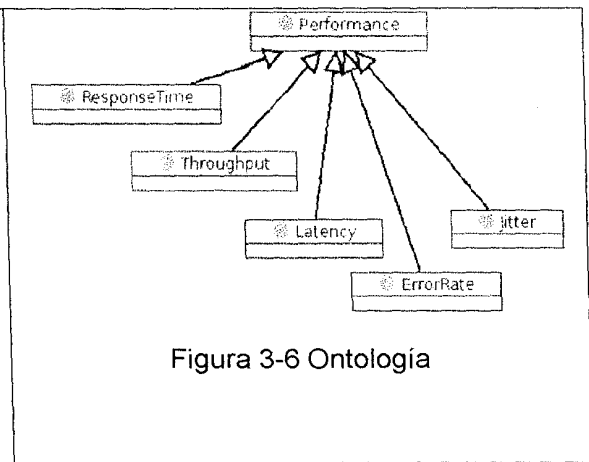


Figura 3-6 Ontología

También, requerimos la meta-ontología que describe el canal, mostrada en la figura 2-13, a la cual se le añade la naturaleza de la información a transmitir (ver la ontología de la

Figura 2-16), y los parámetros de rendimiento antes descritos en las Figura 3-5 y Figura 3-6 (ver Figura 3-7).

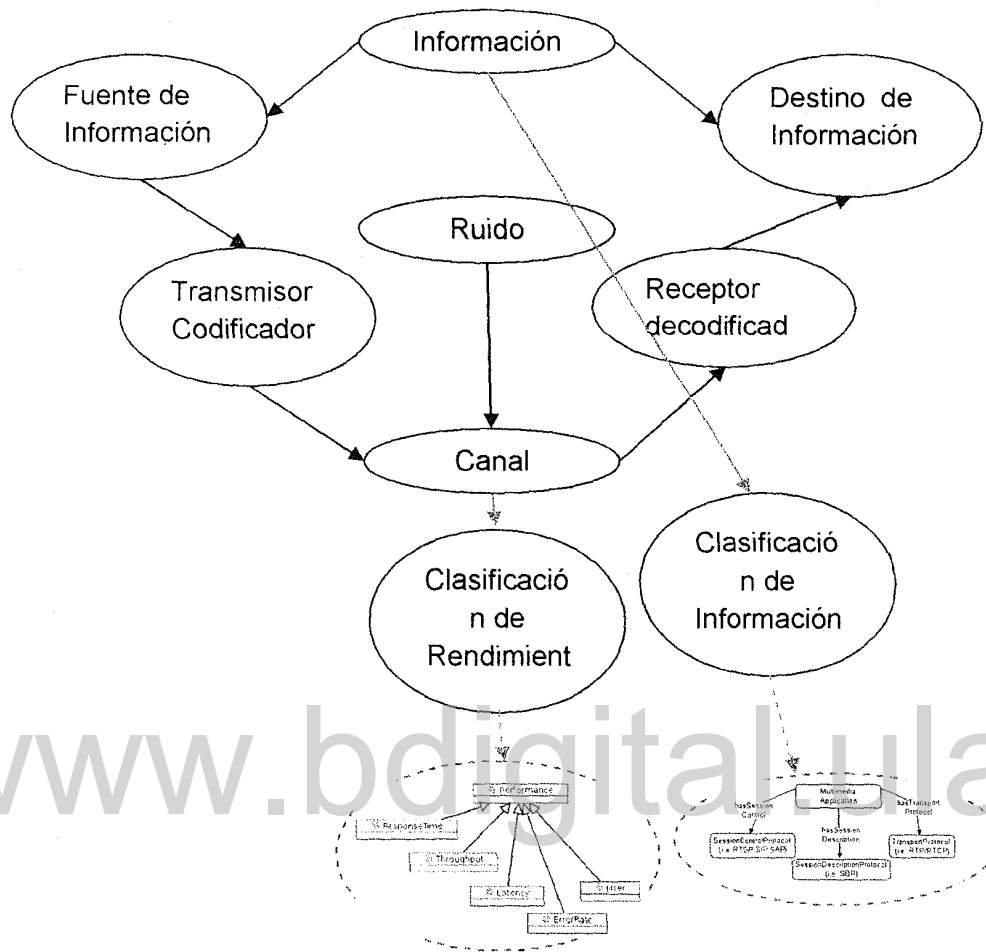


Figura 3-7 Modelo General

La ontología fusionada en formato reducido se muestra en la Figura 3-8, la misma representa un compendio de todos los elementos utilizados en las redes de comunicación autónomas, dando cuenta de los elementos involucrados en el proceso de toma de decisión.

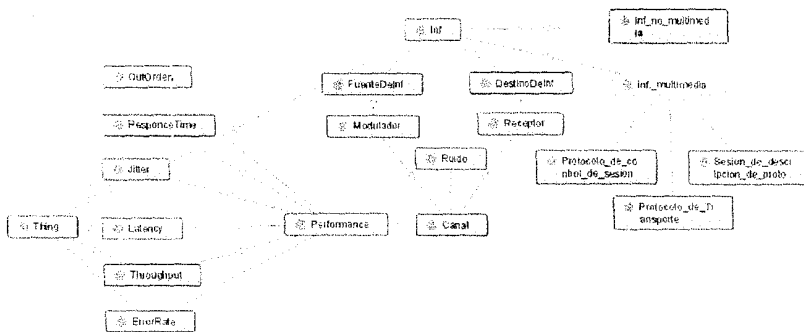


Figura 3-8 Ontología propuesta

La ontología de la Figura 3-8 representa la ontología propuesta para el caso de implementación ideal; sin embargo, debido a las limitaciones impuestas por los softwares usados para realizar las pruebas y simulaciones (NS2 y Netica), se toma solo una fracción de la misma, mostrada en la Figura 3-9 (cuadro punteado).

En cuanto a las limitaciones de esa ontología reducida, su impacto no será apreciable desde el punto de vista de las simulaciones, ya que NS2 no simula conexiones de tipo multimedia de forma real. Además, los otros parámetros como el tiempo de respuesta y la variación del retardo no son críticos, ya que son solo validos para las aplicaciones de tipo multimedia. En conclusión, la fracción de la ontología propuesta permite hacer la prueba de conceptos, ella permite estudiar la factibilidad del diseño y su implementación.

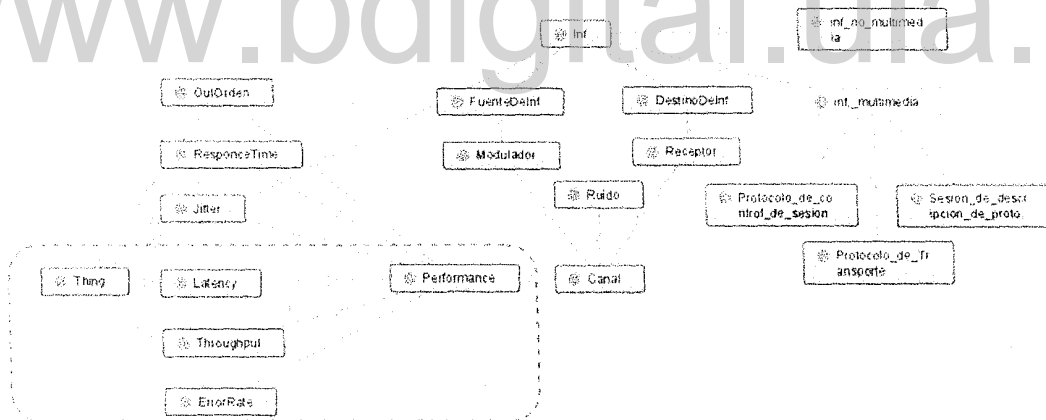


Figura 3-9 Segmento de la ontología a implantar

3.2.1. Arquitectura propuesta para la transformación de la Ontología a la red bayesiana

Para que la ontología propuesta sea útil en el contexto propuesto, la misma debe ser transformada en una red bayesiana. Para esto, la ontología debe estar descrita en OWL,

por lo que se utiliza la herramienta Protege, y después se invoca al sistema de traducción (transformación) estructural usado (BayesOWL 1.0, Ver Figura 3-10). Para esto se utilizó una implementación escrita en Java del algoritmo BayesOWL (Rong, Zhongli, & Peng, 2006).

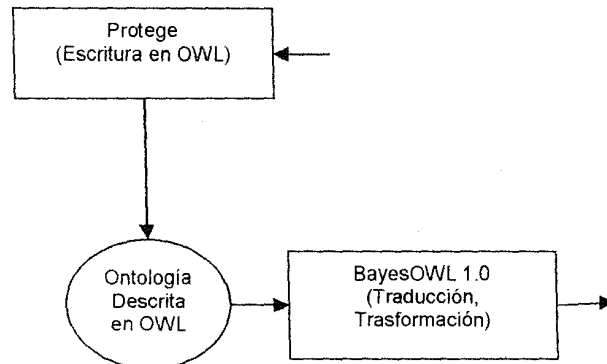


Figura 3-10 Arquitectura para la transformación de la Ontología a la red bayesiana

3.2.2. Transformación de la Ontología propuesta en red bayesiana

Para mostrar el proceso de transformación, se considera el fragmento de la ontología demarcado en la Figura 3-9. A continuación se muestran cada uno de los pasos del algoritmo:

Se comienza con la ontología mostrada en la Figura 3-11, la cual en adelante será la ontología a ser usada,



Figura 3-11 Ontología reducida

- **Traducción estructural de la ontología reducida**

A continuación se muestra la aplicación detallada de cada uno de los pasos descritos en BayesOWL, para transformar la ontología de la Figura 3-11.

➤ Paso 1

Cada primitiva o concepto Clase C definido, es mapeado en variables de dos estados (verdadero o falso) en la red bayesiana.

```
<owl:Class rdf:ID="Throughput"/>
<owl:Class rdf:ID="Latency"/>
<owl:Class rdf:ID="ErrorRate"/>
```

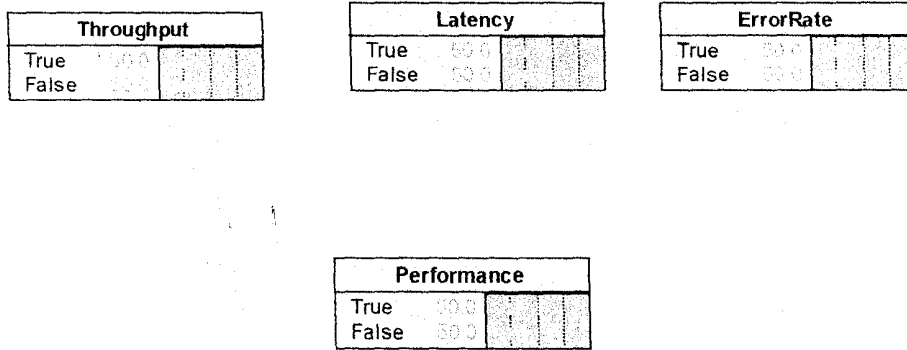


Figura 3-12 Primitivas de clase

➤ Paso 2

El constructor "rdfs:subClassOf" es modelado por un arco directo del nodo de la clase padre (parent superclass) a cada nodo de la subclase hijo. Para ello usamos la descripción en OWL:

```
<owl:Class rdf:ID="Performance">
  <rdfs:subClassOf rdf:resource="#Throughput"/>
  <rdfs:subClassOf rdf:resource="#Latency"/>
  <rdfs:subClassOf rdf:resource="#ErrorRate"/>
</owl:Class>
```

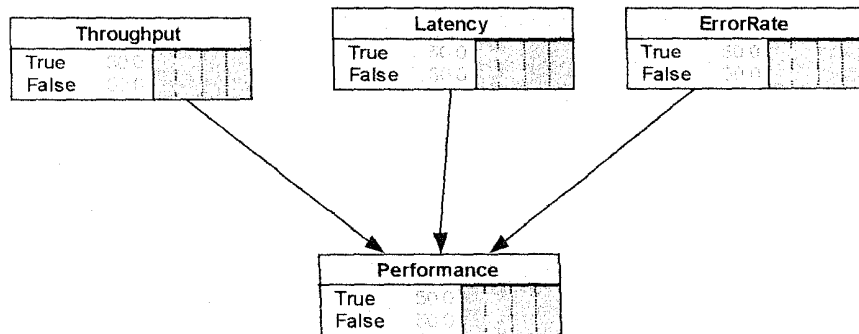


Figura 3-13 Subclases

Los pasos para crear los nodos de operaciones “owl:intersectionOf”, “owl:unionOf”, “owl:complementOf”, “owl:equivalent-Class”, “(owl:disjointWith),” no aplican, porque la ontología no posee esos subconceptos.

Esta red bayesiana representa la transformación de la ontología de la Figura 3-11, y la misma muestra como la variable performance (rendimiento), es afectada por el caudal “Throughput”, la latencia “latency” y por la tasa de errores “ErrorRate”

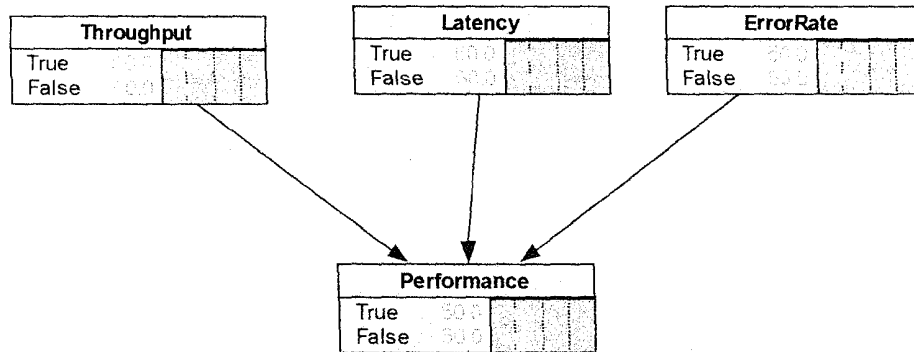


Figura 3-14 Traducción estructural completa

Una vez concluida la transformación de la ontología (ver Figura 3-14), se pasa a definir las tablas CPTs. Para construir las tablas iniciales CPTs, se siguen los siguientes pasos:

➤ Paso 1

Según el método de transformación BayesOWL, las tablas CPTs se pueden construir a partir de datos experimentales o de las restricciones ontológicas, para este caso se calcularon a partir de datos experimentales.

➤ Paso 2

Se realizan múltiples simulaciones experimentales, en dichas simulaciones se obtuvieron datos de todas las variables.

➤ Paso 3

Una vez obtenidos los datos experimentales, los mismos se usaron para calcular las CPTs por medio del aprendizaje por método de maximización de la expectativa. Para aplicar este método, se inicializaron las variables con valores aleatorios. En nuestro caso, todos los valores se distribuyeron de forma uniforme (ver Figura 3-15)

Node: Rend			Apply OK											
Chance			Reset Close											
Delay	Th	Loss	0 to 10	10 to 20	20 to 30	30 to 40	40 to 50	50 to 60	60 to 70	70 to 80	80 to 90	90 to 100		
0.75 to 1	0.5 to 0.75	0.75 to 1	10	10	10	10	10	10	10	10	10	10		
0.75 to 1	0.75 to 1	0 to 0.25	10	10	10	10	10	10	10	10	10	10		
0.75 to 1	0.75 to 1	0.25 to 0.5	10	10	10	10	10	10	10	10	10	10		
0.75 to 1	0.75 to 1	0.5 to 0.75	10	10	10	10	10	10	10	10	10	10		
0.75 to 1	0.75 to 1	0.75 to 1	10	10	10	10	10	10	10	10	10	10		

Figura 3-15 Tablas CPT, con valores uniformes

➤ Paso 4

Los datos experimentales se usaron como los parámetros actuales $\theta_c^{[t]}$, para estimar las variables desconocidas en el paso E, usando las ecuaciones 5 y 6. Luego se pasa al paso M, donde se usa la ecuación 7 para calcular los valores de la tabla. Se repiten los pasos E y M hasta que se establecen los valores estimados. Al terminar el aprendizaje, se modifican los valores iniciales en el CPTs por los provistos por esta fase (ver Figura 3-16).

Node: Rend			Apply OK											
Chance			Reset Close											
Delay	Th	Loss	0 to 10	10 to 20	20 to 30	30 to 40	40 to 50	50 to 60	60 to 70	70 to 80	80 to 90	90 to 100		
0.75 to 1	0 to 0.25	0 to 0.25	1.15e-6	1.15e-6	1.15e-6	1.15e-6	1.15e-6	25.977	78.872	0.845	1.15e-6	1.15e-6		
0.75 to 1	0 to 0.25	0.25 to 0.5	5.02e-4	5.02e-4	5.02e-4	5.02e-4	5.02e-4	89.955	5.02e-4	5.02e-4	5.02e-4	5.02e-4		
0.75 to 1	0 to 0.25	0.5 to 0.75	10	10	10	10	10	10	10	10	10	10		
0.75 to 1	0 to 0.25	0.75 to 1	10	10	10	10	10	10	10	10	10	10		
0.75 to 1	0.25 to 0.5	0 to 0.25	9.17e-6	9.17e-6	9.17e-6	9.17e-6	9.17e-6	9.17e-6	40.367	59.633	9.17e-6	9.17e-6		

Figura 3-16 Tablas CPT, luego del aprendizaje inicial con EM

3.2.3. Transformación de la red bayesiana en un diagrama de influencia

Ahora la Red Bayesiana se debe transformar en un diagrama de influencia para poder utilizar el criterio de máxima utilidad esperada, y así facilitar la toma de decisión en el sistema. En el presente trabajo, la idea es que el sistema decida qué tipo de reconfiguración debe ser ejecutada para maximizar el rendimiento de la comunicación. A continuación se muestra el flujo que se sigue para lograr dicha transformación (ver Figura 3-17).

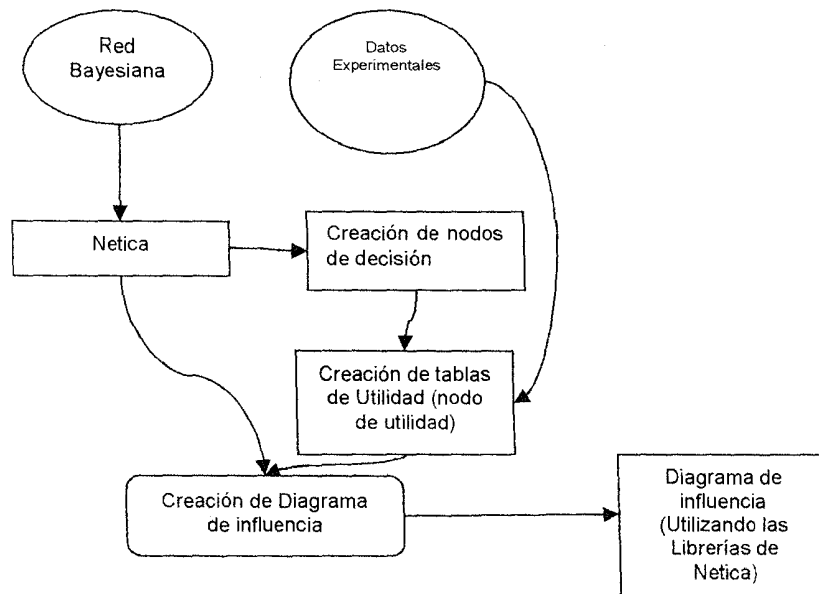


Figura 3-17 Proceso de transformación de la red bayesiana en un diagrama de influencia

En el nodo de decisión se deben colocar las alternativas de cambio que se presentan. Debido a las limitaciones del simulador usado (NS2), en dicho nodo se pueden utilizar los siguientes tipos de TCP:

- Reno
- New Reno
- Sack1
- Vegas
- Fack
- Linux
- TCP o Tahoe

Para este proceso se comienza con la ontología de la Figura 3-14 y se le agregan los nodos de decisión, como se observa en la Figura 3-18. En nuestro caso hay un solo nodo de decisión, en este nodo de decisión estarán codificados todos los posibles cambio que se puedan realizar según el contexto en que se va a utilizar.

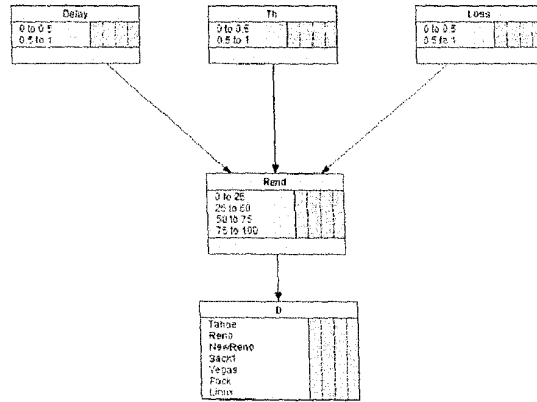


Figura 3-18 Ontología con nodo de decisión

Una vez creados los nodos de decisión, se procede a crear la función de utilidad (tabla de utilidad). Esta tabla se mostrara como un nodo (nodo de utilidad) con dependencia de todos los nodos con los cuales esa tabla de utilidad tenga relación (ver Figura 3-19). En la sección 3.3 explicamos cómo se calculan esos valores de la tabla de utilidad.

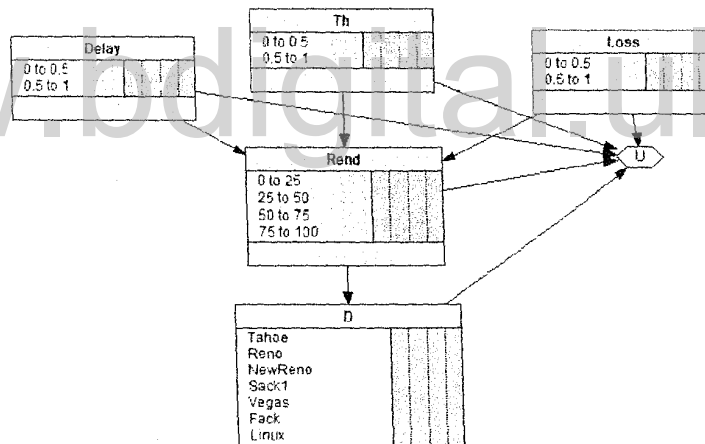


Figura 3-19 Ontología transformada en Diagrama de influencia

3.3. **Entonación del Modelo de Decisión y aprendizaje inicial**

Una vez obtenido el diagrama de influencia (Figura 3-19), se procede a caracterizar el nodo de utilidad con los datos experimentales obtenidos de múltiples simulación para diferentes valores de parámetros (retardo, capacidad del canal, y tasa de pérdidas).

En la Figura 3-20 se muestra el diagrama de influencia utilizado para las simulaciones del sistema de comunicación con NS2. En la Figura 3-20 se muestra que los valores para las variable retardo (Delay), Caudal (Th), rendimiento (Rend) y tasa de pérdidas (Loss) se

dividen en cuatro rangos de valores, esto es debido a el tipo de dispersión de la data experimental, que no era fácilmente describible con menos rangos. Estos rangos se asignan por medio de una distribución de frecuencia. Esta distribución de frecuencia depende de las simulaciones, y pueden cambiar. De hecho, en las pruebas del sistema se utilizaron dos diagramas de influencia con rangos diferentes para la variable de rendimiento, para probar la influencia de la escogencia de los mismos en el resultado final.

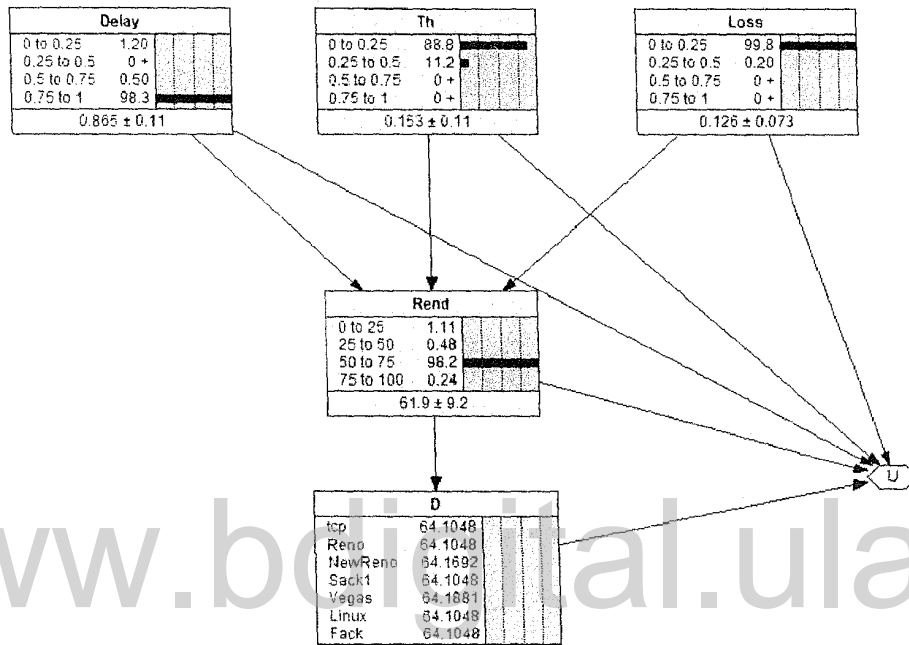


Figura 3-20 Diagrama de influencia Utilizado para las simulaciones

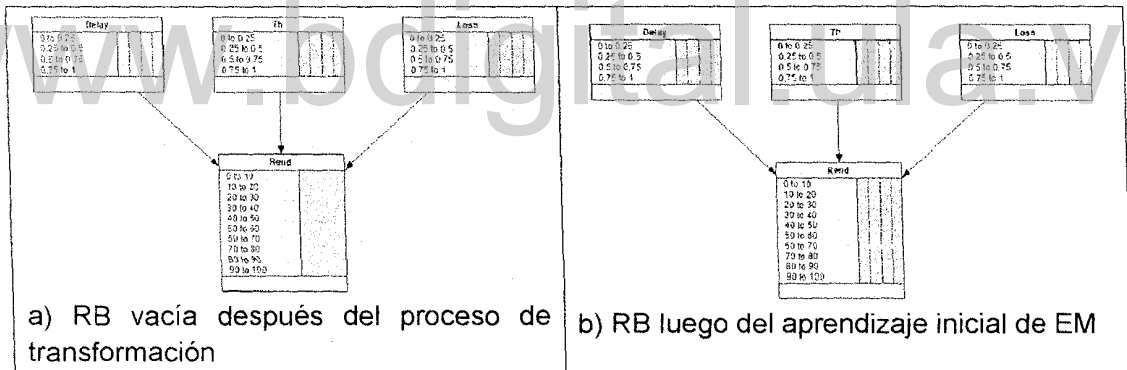
La tabla de utilidad tiene las siguientes características: es definida por el nodo de utilidad U del diagrama de influencia de la Figura 3-20, el cual depende, en este caso, de todas las variables del diagrama de influencia (retardo (Delay), Caudal (Th), rendimiento (Rend) y tasa de pérdidas (Loss)) y del nodo de decisión (D). Esa dependencia se observa en la Figura 3-20, y esta descrita por medio de las flechas que llegan al nodo U. Debido a la dependencia del nodo U con todas las variables, su valor vendrá dado dependiendo de las mismas, como se observa en las tablas de la Figura 3-22 (más adelante se describe su cálculo).

El proceso de aprendizaje inicial se realizó usando la técnica EM (maximización de la expectativa) sobre la red bayesiana; luego de este proceso inicial, el sistema continúa el aprendizaje mediante el método por conteo. El aprendizaje se realiza cada vez que se ejecuta una decisión de reconfiguración. Este proceso refuerza la red bayesiana, de forma tal que las decisiones que incrementan el rendimiento del sistema ayuden también a mejorar el sistema de decisión. Este proceso consiste en actualizar la red bayesiana (Ver Figura 3-21) y la tabla de utilidad, para que el sistema pueda mejorar su rendimiento. La

figura 3-21.a muestra la red bayesiana inicialmente, la figura 3-21.b la red bayesiana después del aprendizaje inicial usando EM (como se indico antes), y la figura 3-21.c es la red bayesiana después de los procesos de aprendizaje por conteo (después de cada simulación).

En la Figura 3-22 se puede observar la tabla de utilidad antes y después del proceso de aprendizaje, la misma no está llena en su totalidad debido a que la misma se va llenando cuando se van presentando en el sistema los aspectos específicos descritos en la tabla (por ejemplo, se uso el protocolo Fack, con un retardo (Delay) de 0.75-1, un caudal (Th) de 0-0.25, una perdida (Loss) de 0-0.25 y un rendimiento (Rend) de 50-60, obteniéndose una utilidad (U) de 57.518). En la Figura 3-22.a se puede observa la tabla de utilidad al principio del proceso de aprendizaje, cuando está vacía, y en la Figura 3-22.b se muestra la tabla de utilidad generada por el proceso de aprendizaje. La utilidad se calcula mediante la ecuación 16 (los detalles de esta ecuación serán explicados con detenimiento en el punto 4.1), usando los datos observados en el sistema de las variables retardo (Delay), caudal (Th), perdida (Loss), y en particular del Rendimiento (Rend), que también depende de todos los anteriores. Así, el modelo de toma de decisión busca alcanzar la máxima utilidad esperada en el rendimiento:

$$U = MAX(Rend) \quad (16)$$



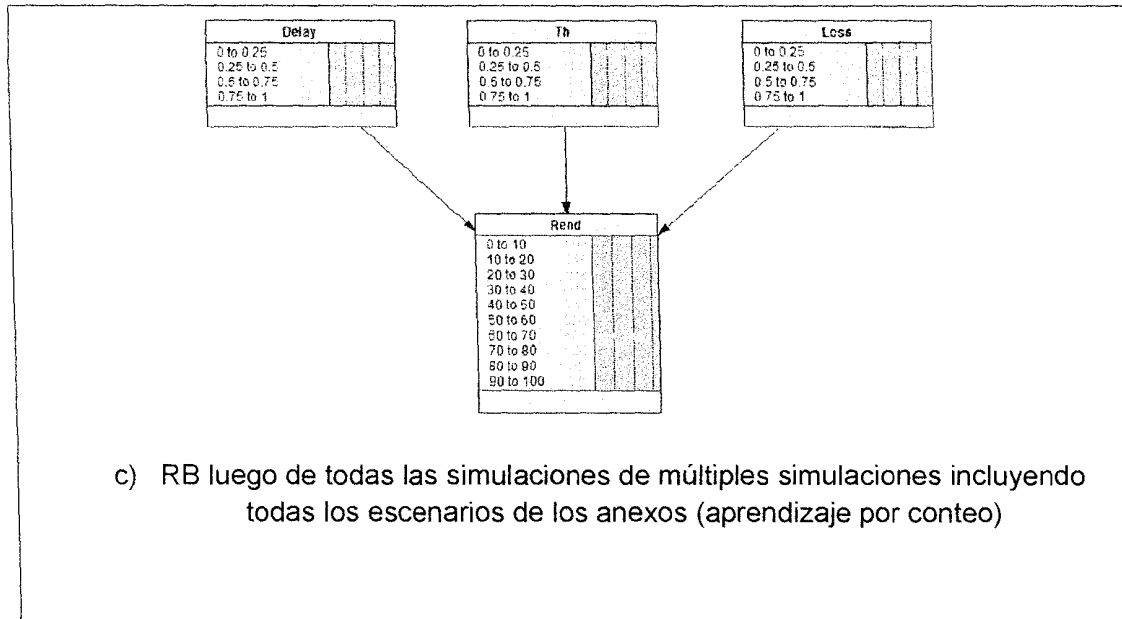


Figura 3-21 Aprendizaje sobre la red Bayesiana

D	Delay	Th	Loss	Rend	U	D	Delay	Th	Loss	Rend	U
Fack	0.75 to 1	0 to 0.25	0 to 0.25	30 to 40		Fack	0.75 to 1	0 to 0.25	0 to 0.25	30 to 40	
Fack	0.75 to 1	0 to 0.25	0 to 0.25	40 to 50		Fack	0.75 to 1	0 to 0.25	0 to 0.25	40 to 50	
Fack	0.75 to 1	0 to 0.25	0 to 0.25	50 to 60		Fack	0.75 to 1	0 to 0.25	0 to 0.25	50 to 60	57.512
Fack	0.75 to 1	0 to 0.25	0 to 0.25	60 to 70		Fack	0.75 to 1	0 to 0.25	0 to 0.25	60 to 70	58.52
Fack	0.75 to 1	0 to 0.25	0 to 0.25	70 to 80		Fack	0.75 to 1	0 to 0.25	0 to 0.25	70 to 80	71.322
Fack	0.75 to 1	0 to 0.25	0 to 0.25	80 to 90		Fack	0.75 to 1	0 to 0.25	0 to 0.25	80 to 90	
Fack	0.75 to 1	0 to 0.25	0 to 0.25	90 to 100		Fack	0.75 to 1	0 to 0.25	0 to 0.25	90 to 100	
Fack	0.75 to 1	0 to 0.25	0.25 to 0.5	0 to 10		Fack	0.75 to 1	0 to 0.25	0.25 to 0.5	0 to 10	
Fack	0.75 to 1	0 to 0.25	0.25 to 0.5	10 to 20		Fack	0.75 to 1	0 to 0.25	0.25 to 0.5	10 to 20	
Fack	0.75 to 1	0 to 0.25	0.25 to 0.5	20 to 30		Fack	0.75 to 1	0 to 0.25	0.25 to 0.5	20 to 30	
Fack	0.75 to 1	0 to 0.25	0.25 to 0.5	30 to 40		Fack	0.75 to 1	0 to 0.25	0.25 to 0.5	30 to 40	39.325
Fack	0.75 to 1	0 to 0.25	0.25 to 0.5	40 to 50		Fack	0.75 to 1	0 to 0.25	0.25 to 0.5	40 to 50	41.723
Fack	0.75 to 1	0 to 0.25	0.25 to 0.5	50 to 60		Fack	0.75 to 1	0 to 0.25	0.25 to 0.5	50 to 60	51.426
Fack	0.75 to 1	0 to 0.25	0.25 to 0.5	60 to 70		Fack	0.75 to 1	0 to 0.25	0.25 to 0.5	60 to 70	
Fack	0.75 to 1	0 to 0.25	0.25 to 0.5	70 to 80		Fack	0.75 to 1	0 to 0.25	0.25 to 0.5	70 to 80	
Fack	0.75 to 1	0 to 0.25	0.25 to 0.5	80 to 90		Fack	0.75 to 1	0 to 0.25	0.25 to 0.5	80 to 90	
Fack	0.75 to 1	0 to 0.25	0.25 to 0.5	90 to 100		Fack	0.75 to 1	0 to 0.25	0.25 to 0.5	90 to 100	
Fack	0.75 to 1	0 to 0.25	0.5 to 0.75	0 to 10		Fack	0.75 to 1	0 to 0.25	0.5 to 0.75	0 to 10	
Fack	0.75 to 1	0 to 0.25	0.5 to 0.75	10 to 20		Fack	0.75 to 1	0 to 0.25	0.5 to 0.75	10 to 20	19.424
Fack	0.75 to 1	0 to 0.25	0.5 to 0.75	20 to 30		Fack	0.75 to 1	0 to 0.25	0.5 to 0.75	20 to 30	29.552
Fack	0.75 to 1	0 to 0.25	0.5 to 0.75	30 to 40		Fack	0.75 to 1	0 to 0.25	0.5 to 0.75	30 to 40	33.653
Fack	0.75 to 1	0 to 0.25	0.5 to 0.75	40 to 50		Fack	0.75 to 1	0 to 0.25	0.5 to 0.75	40 to 50	
Fack	0.75 to 1	0 to 0.25	0.5 to 0.75	50 to 60		Fack	0.75 to 1	0 to 0.25	0.5 to 0.75	50 to 60	
Fack	0.75 to 1	0 to 0.25	0.5 to 0.75	60 to 70		Fack	0.75 to 1	0 to 0.25	0.5 to 0.75	60 to 70	
Fack	0.75 to 1	0 to 0.25	0.5 to 0.75	70 to 80		Fack	0.75 to 1	0 to 0.25	0.5 to 0.75	70 to 80	
Fack	0.75 to 1	0 to 0.25	0.5 to 0.75	80 to 90		Fack	0.75 to 1	0 to 0.25	0.5 to 0.75	80 to 90	
Fack	0.75 to 1	0 to 0.25	0.5 to 0.75	90 to 100		Fack	0.75 to 1	0 to 0.25	0.5 to 0.75	90 to 100	
Fack	0.75 to 1	0 to 0.25	0.75 to 1	0 to 10		Fack	0.75 to 1	0 to 0.25	0.75 to 1	0 to 10	0
Fack	0.75 to 1	0 to 0.25	0.75 to 1	10 to 20		Fack	0.75 to 1	0 to 0.25	0.75 to 1	10 to 20	17.125
Fack	0.75 to 1	0 to 0.25	0.75 to 1	20 to 30		Fack	0.75 to 1	0 to 0.25	0.75 to 1	20 to 30	

a) Tabla de utilidad antes del proceso de aprendizaje

b) Tabla de utilidad durante el proceso de aprendizaje

Figura 3-22 Tabla de utilidad

Mientras que el sistema aprende en las etapas iniciales, el mismo no toma ninguna decisión. Al concluir esta etapa, el sistema aprende y toma decisiones en base a la tabla

de utilidad, escogiendo la mayor utilidad posible en la tabla para una entrada dada. Por ejemplo, en la Tabla 3-1 se observa el aprendizaje para una entrada de la función de utilidad en el cual el valor previo era de 52.455, y luego se actualiza a un valor de 55.501. Para el caso de la actualización de la red bayesiana, las CPTs cambian de forma similar a como se observó en las Figura 3-15 y 3-16 (se va actualizando la CPT para cada nodo).

Aprendizaje	Valor de la utilidad U	Entrada de la Tabla				
		Proto D	Delay	Th	Loss	Rend
Valor Previo	52.455	Linux	0.5 to 0.75	0.25 to 0.5	0 to 0.25	50 to 60
Valor posterior	55.501	Linux	0.5 to 0.75	0.25 to 0.5	0 to 0.25	50 to 60

Tabla 3-1 Ejemplo de actualización de una entrada de la tabla de utilidad

www.bdigital.ula.ve

4. Pruebas experimentales del Sistema

Para verificar el comportamiento del sistema de decisión construido a partir de la ontología propuesta, se utiliza la herramienta de simulación NS2 (Qi, Zhang, Cao, Sun, & Tan, 2012), la cual es un simulador de redes a eventos discretos, utilizado principalmente para investigación y enseñanza. El NS2 es utilizado en gran medida debido a que es un simulador de código abierto, adicionalmente a esto, el NS2 posee cantidad de protocolos y una gran comunidad de usuarios.

En adición al simulador de redes NS2, para implementar el sistema de decisión del sistema de comunicaciones autónomo se utilizan las librerías especializadas (APIs) para redes de creencias (redes bayesianas) y diagramas de influencia de la empresa NORSYS (en particular, el software netica, ver figura 4-1) (Norsys, 2010). Estas librerías permiten usar las redes bayesianas desde los scripts de simulación, para la toma de decisión de reconfiguración.

4.1. Diseño del protocolo experimental

Para ejecutar las simulaciones con NS2, se implementó un script llamado paramétrico (script de simulación), el cual cambia los parámetros de la simulación del NS2 (ver. Figura 4-1) en función a las decisiones tomadas por el sistema de decisión. El sistema de decisión usa como entrada el análisis de las trazas de simulación, de forma tal de determinar el contexto para proponer variaciones en los parámetros del sistema de comunicación. Una vez obtenida las trazas de la simulación, son analizadas mediante un script que determina los resultados de la simulación: caudal, tasa de pérdidas y retardo (ese proceso de cálculo se describe con más detalle más en los próximos párrafos). Con esos resultados, el sistema de decisión determina si se debe ejecutar un cambio en la simulación para mejorar la comunicación. Después, el simulador se ejecuta nuevamente con los cambios de reconfiguración y recomienza el proceso.

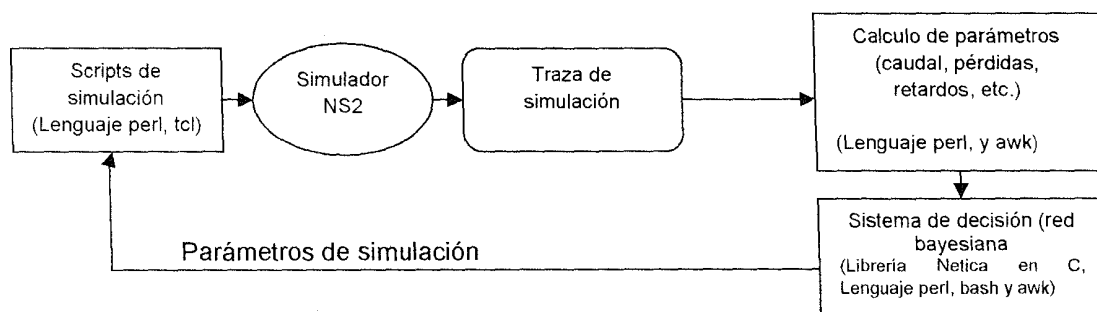


Figura 4-1 Sistema de decisión (Lazo cerrado de simulación)

Como se comento, para calcular el caudal, la tasa de pérdidas y el retardo, se utiliza un script (ver Anexo 6.2), que toma en consideración la cantidad de paquetes que se reciben de forma exitosa, que se envían, que se descartan, y los tiempo en que fueron enviados y recibidos, para así poder realizar los cálculos pertinentes para determinar esos tres parámetros. La Figura 4-2 muestra un ejemplo de esos resultados dados por el script, para un lapso de tiempo dado.

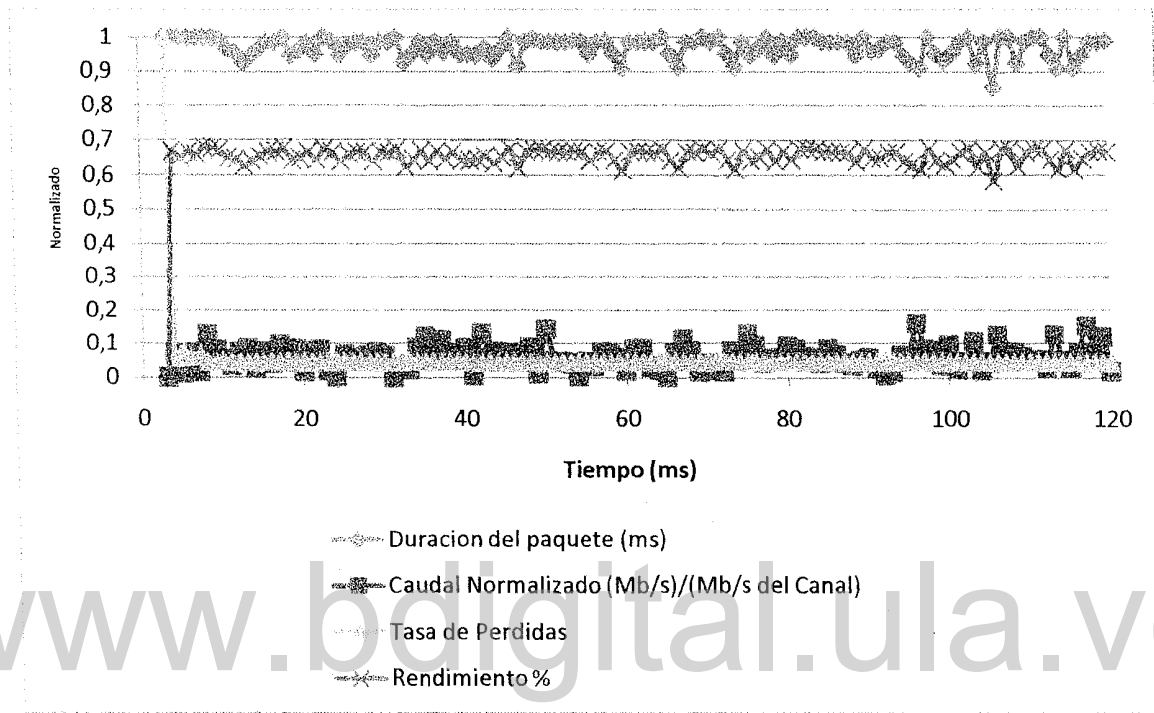


Figura 4-2 Resultados de la simulación

Ahora bien, usando esos tres parámetros se propone utilizar la siguiente ecuación, basada en una propuesta en (Van, 2009), inspirada en los parámetros de una comunicación ideal, llamada Rendimiento (%):

$$Rendimiento = \sqrt{\frac{RetardoR^2 + CaudalN^2}{2}} (1 - Tperdidas) \quad (17)$$

En la ecuación 17 de rendimiento, se toma en consideración que si el caudal relativo *CaudalN* (descrito en la ecuación 21) junto con el Retardo Relativo *RetardoR* (descrito en la ecuación 18) son nominales (es decir que su valor numérico es la unidad "1"), y el porcentaje de pérdidas *Tperdidas* (descrito en la ecuación 22) es cero por ciento (es decir, que no existen pérdidas), el valor numérico del rendimiento es 100%. Un valor de 100% de rendimiento significa que desde el punto de vista de la formula la comunicación es ideal. La fórmula del rendimiento se implemento pensando en obtener una media de

todos los criterios de rendimiento definidos en nuestra ontología reducida. En el Anexo 6.3, se amplían algunos detalles de la fórmula de rendimiento.

El RetardoR es el retardo relativo, normalizado linealmente con un retardo teórico (Rt), de 3 veces el retardo teórico. Se toma ese número debido a que un cálculo exacto del retardo, o del tiempo de viaje RTT (round trip time), es difícil de calcular de forma simple (Tamas & Saridor, 1999) (ver Figura 4-3):

$$\text{RetardoR} = \frac{-1}{2 \cdot \text{RetardoT}} * ((\text{Duracion de paquete en la red}) - \text{RetardoT}) + 1 \quad (18)$$

$$\text{Donde, RetardoT} = \frac{\text{Retardo de Canal (ms)}}{1000} + \frac{\text{Tamaño de paquete (bytes)}}{\text{Ancho de Banda del Canal (MB)} \cdot \frac{1048576}{8}} \quad (19)$$

El RetardoT (retardo teórico) es el tiempo que le toma a un paquete recorrer el canal de comunicación en estudio (en los nodos donde el canal puede presentar ruido, interferencia, y retardos debido a colas).

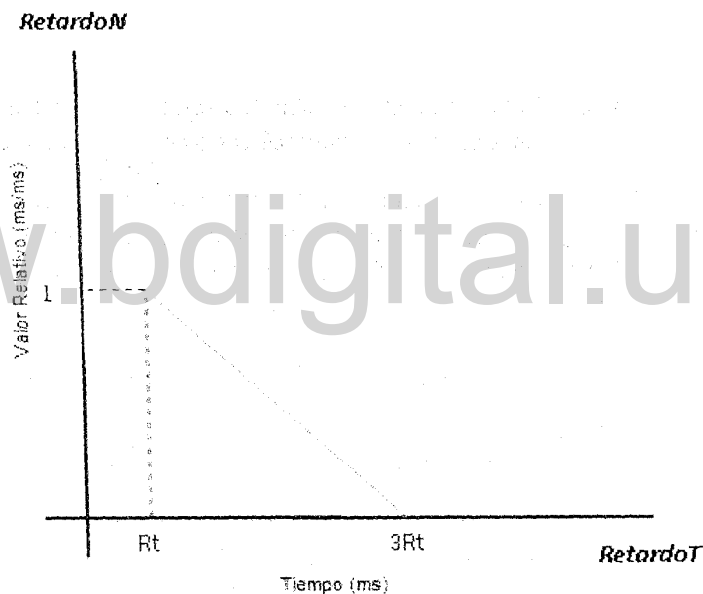


Figura 4-3 Normalización Retardo Nominal vs Retardo Teórico

El CaudalN es el Caudal Normalizado al máximo que se puede obtener en la simulación debido al ancho de banda mínimo del trayecto, el cual representa el máximo caudal que puede alcanzar la comunicación deseada. El caudal se determina como el número de paquetes enviados en una conexión entre el tiempo. El caudal se determina como la cantidad de datos enviados en una conexión entre el tiempo (Van, 2009)

$$\text{Caudal} = \frac{\text{cantidad de datos enviados en la conexión deseada}}{\text{tiempo}} \quad (20)$$

$$CaudalN = \frac{Caudal}{Caudal.Max} \quad (21)$$

Tpérdidas es la tasa de pérdidas, es la relación entre el número de paquetes perdidos en la conexión deseada y el número total de paquetes enviados en la conexión deseada (Van, 2009)

$$Tperdidas = \frac{Numero\ de\ paquetes\ perdidos\ en\ la\ conexion\ deseada}{Numero\ de\ paquetes\ enviados\ en\ la\ conexion\ deseada} \quad (22)$$

El sistema de decisión tratara de obtener el máximo rendimiento en todos los casos.

En el sub-sistema de toma de decisión mostrado en la Figura 3-2, se observa el modulo llamado sub-sistema de cálculo de desempeño, el cual observa como varia el desempeño del sistema, y decide cuándo debe consultarse el sistema de decisión.

4.1.1. Sub-sistema de cálculo de desempeño

El sub-sistema de cálculo de desempeño invoca al diagrama de influencia cuando determina que se debe tomar una decisión, que es cuando hay una degradación en el rendimiento. Veamos cómo se realiza ese proceso.

En las Figura 4-4 y Figura 4-5 se pueden observar con mayor grado de detalle el rendimiento. En particular, en la Figura 4-5 se observa que es la primera oportunidad que el rendimiento cae, y sigue una linea de tendencia a la baja. El sistema de cálculo de desempeño analiza los puntos de la gráfica, en busca de una serie de tres valores seguidos en los cuales si se calcula su pendiente entre todas las combinaciones posibles de estos tres puntos, todas tendran valores negativos. Al encontrarse esta serie de tres puntos, se determina que se debe invocar al sistema de desición (se detecta la caída en el rendimiento), el cual usa el diagrama de influencia para determinar si debe o no reconfigurar el sistema.

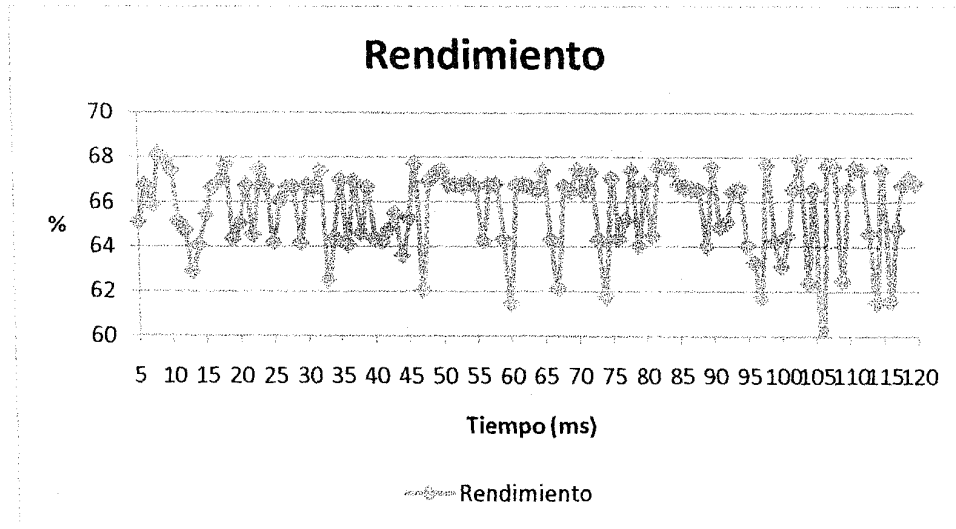


Figura 4-4 Detalle de Rendimiento vs Tiempo de simulación

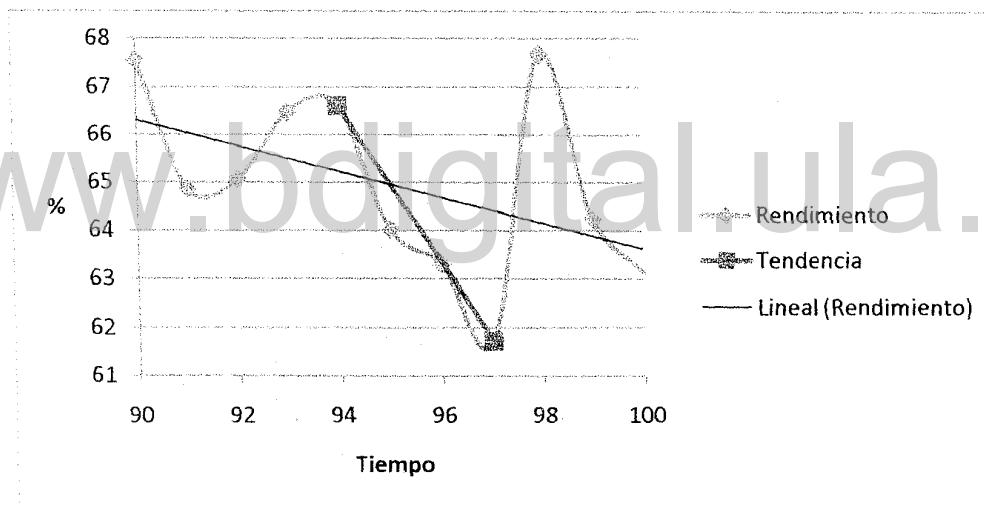


Figura 4-5 Detalle de Rendimiento vs Tiempo de simulación, con línea de tendencia

4.1.2. Proceso de simulación

Debido a las limitaciones en la forma como se construyó el simulador NS2, no se pueden hacer variaciones de los parámetros en tiempo real. Por esa razón, en nuestra plataforma experimental el sistema de decisión toma la decisión de reconfigurar fuera de línea. Además, por eso también se lanza una nueva simulación con la nueva configuración originada por la toma de decisión de nuestro sistema, y se compara con la simulación inicial. De esta forma, se puede determinar si la decisión tomada por el sistema de decisión incrementa el desempeño. Un ejemplo de comparación, para un escenario con

tasa de pérdida del 25%, de la simulación con y sin cambios, se puede observar en la Tabla 4-1. En la misma, los primeros valores de rendimiento promedio y caudal promedio son menores que para los valores de rendimiento promedio y caudal promedio una vez propuesto los cambios por el sistema de decisión, el cual, para este caso decidió cambiar en el tiempo de 24,001147 ms. de TCP/Linux a TCP/Sack1. Las graficas de rendimiento y caudal para la simulación de la Tabla 4-1, se pueden observar en las Figura 4-6 y Figura 4-7, respectivamente.

Tasa de errores %	Rendimiento promedio % (sin sistema autónómico)	Caudal Normalizado (sin sistema autónómico)	Numero de cambios propuestos	Rendimiento promedio % (con sistema autónómico)	Caudal Normalizado (con sistema autónómico)	cambios propuestos: (Tiempo ms /tipo de TCP)
25	38,2617	0,00356373	1	43,9269	0,00517973	24,001147/ (cambia de TCP/Linux a TCP/Sack1)

Tabla 4-1 Ejemplo del resultado de una simulación

En la Figura 4-6 se puede observar el resultado de dos simulaciones: en la primera serie (Rendimiento sin SA) se observa la grafica del rendimiento sin sistema autónómico (SA), y en la segunda serie (Rendimiento con SA) se observa el resultado del rendimiento cuando SA tomo la decisión de reconfigurar en el tiempo de 24,001147 ms (cambia de TCP/Linux a TCP/Sack1). Se observa un incremento en el rendimiento promedio de 38,26% a 43,92%. Es importante resaltar que para la serie del Rendimiento sin SA se pueden observar que en ocasiones supera a la grafica de rendimiento con SA, por lo que la estrategia de comparación en este tipo de casos debe ser el promedio ponderado de cada curva. Con respecto a la Figura 4-7, se observa el Caudal con y sin SA, para la misma simulación de la Figura 4-6. Allí claramente se observa que las dos graficas permanecen con el mismo comportamiento hasta llegar a los 24 ms, en el segundo 24 nuestro sistema observa la condición de cambio (descrita anteriormente), y decide realizar un cambio de TCP/Linux a TCP/Sack1, por lo que de allí en adelante ambas graficas varían, siendo claramente mejor el caudal del SA a simple vista.

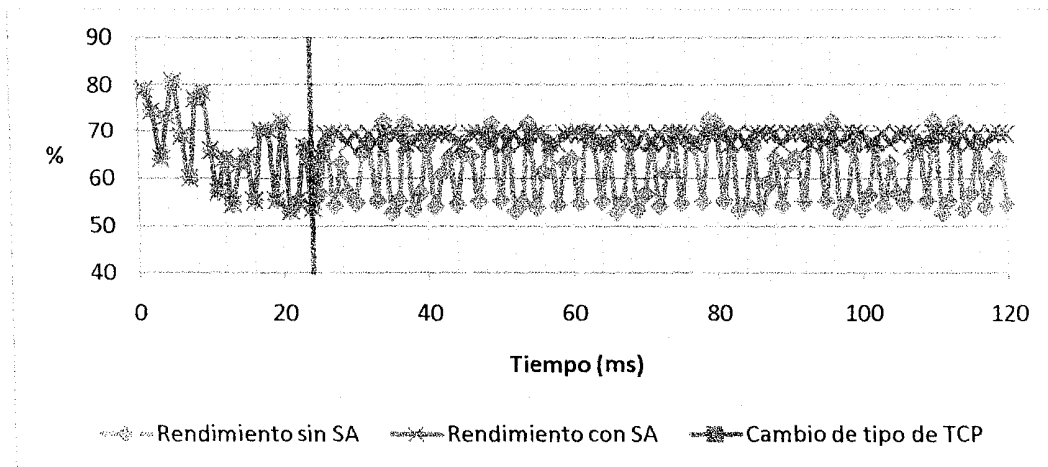


Figura 4-6 Grafico de rendimiento con y sin sistema autónómico

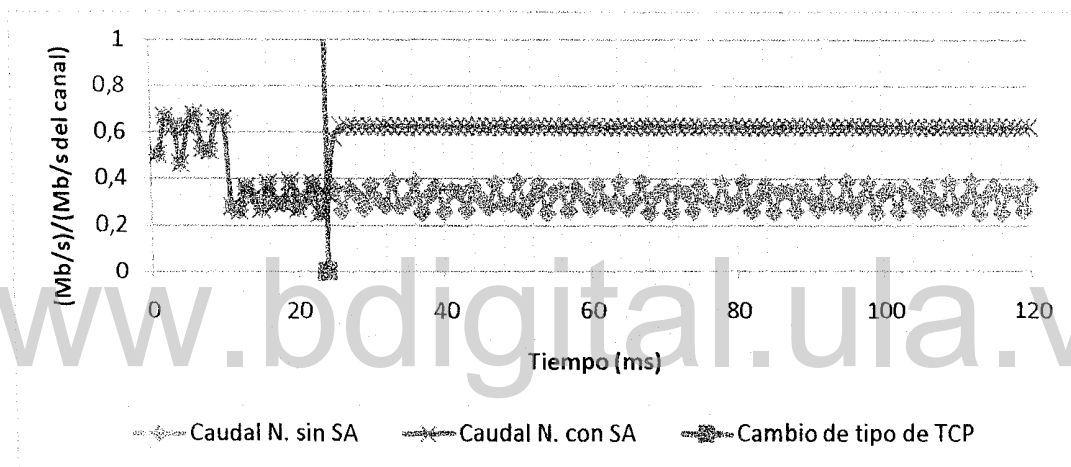


Figura 4-7 Grafico del Caudal Normalizado con y sin sistema autónómico

4.1.3. Protocolo de experimentación

Para verificar si el sistema de decisión mejora la comunicación, y además probar el mecanismo de aprendizaje, se diseñaron una serie de simulaciones, divididas en dos grandes grupos:

- Simulación con cambios en base al rendimiento: a su vez, estas simulaciones están divididas en dos categorías:
 - Variación de los parámetros relacionados con el canal de comunicación: Tasa de Pérdidas, Ancho de Banda del canal, Cola en los nodos intermedios, y retardo.
 - Variación de la tasa y el tipo de transferencia de la comunicación interferente
- Simulación con cambios en base a la tasa de pérdidas

A continuación se muestra la Tabla 4-2, en donde se indican los parámetros del canal de comunicación que se van a variar en las simulaciones, sus dominios, y la justificación del por qué se van a variar durante las simulaciones.

Parámetro	Dominio	Justificación
Retardo	1-100 ms	la variación de este parámetro es importante, porque causa problemas en conexiones que requieren retardo controlado, como por ejemplo conexiones de voz
Tasa de pérdidas en el enlace	0-50%	Es el porcentaje de paquetes que se pierden en el canal durante la transmisión. La tasa de pérdida afecta de forma distinta a cada uno de los tipos de protocolos y sus tipos
Ancho de banda	10-100 Mb	Es la capacidad, en cantidad de datos, de transporte del canal. El ancho de banda es un parámetro crítico, ya que determina en algunos casos si el canal es usable o no para cierto tipo de comunicaciones
Tamaño de la cola (número de paquetes)	0-20	Es la máxima capacidad de paquetes que puede retener un nodo antes de colocarlo en el canal para su transmisión, debido a que el canal no está libre. Es un parámetro solo observable en ciertos tipos de sistemas de comunicaciones, como por ejemplo en redes de computadoras.

Tabla 4-2 Tabla resumen de los parámetros de simulación

4.1.4. Descripción del escenario de simulación

En el escenario descrito en la Figura 4-8 se muestra la topología de red a ser usada en las simulaciones, El canal en el cual se varían los parámetros es el enlace entre el nodo 2 y 3 (enlace 2-3), la conexión a optimizar (comunicación de interés) es la que se realiza entre los nodos 0 y 4 (usa también el enlace 2-3). También existe una conexión interferente entre los nodos 1 y 5, la cual comparte el enlace 2-3. Los enlaces 0-2 y 3-4; no poseen pérdidas, ni colas ni conexiones adicionales, adicionalmente a esto, sus velocidades son siempre mayor a la del enlace 2-3, lo mismo sucede para los enlaces 1-2; y 3-5.

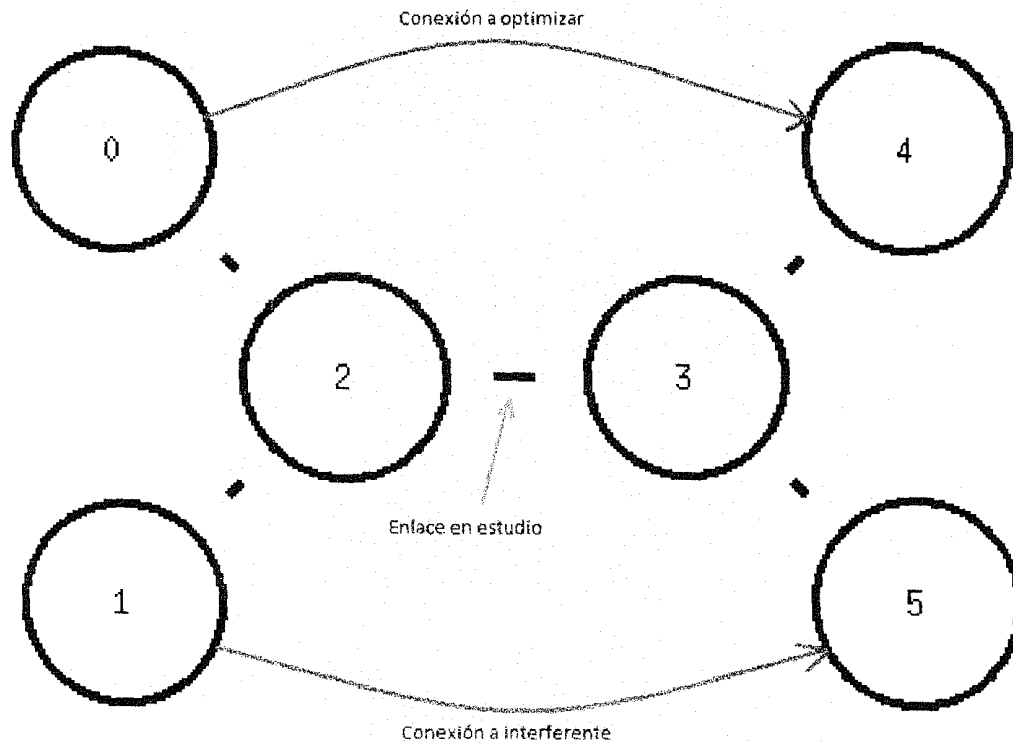


Figura 4-8 Escenario de simulación

En los escenarios de simulación propuestos solo se optimiza una conexión (originada en el nodo 0 y terminada en el nodo 4) al mismo tiempo, tal que cualquier otra comunicación que comparta el canal es una señal interferente. En todos los escenarios de simulación, durante los primeros 10 ms, la conexión deseada enfrenta solo las condiciones del canal como dificultad, y a partir de los 10 ms existe una conexión interferente para simular condiciones reales donde otra conexión causa un ruido o interferencia adicional sobre la conexión deseada.

En la conexión interferente se pueden variar de forma manual los mismos parámetros que en la conexión deseada. En enlace entre el nodo 2 y 3 es el punto donde se estudian las trazas. En ese enlace se van a modificar los parámetros del canal, entre simulaciones, mostrados en la Tabla 4-2

Para determinar cual será el tipo de TCP para usar como tipo inicial, se realizaron simulaciones con cada uno de ellos para determinar inicialmente su comportamiento (ver tabla 4-3). Se utilizó un solo tipo de TCP como conexión inicial, simulando el comportamiento actual de la mayoría de los sistemas, que en línea general solo usan un tipo de conexión.

Las condiciones para estas simulaciones fueron

- Tiempo de simulación de: 120 ms
- Arranque de conexión deseada a los 0.1 ms

- Arranque de conexión interferente en 10 ms
- Tasa de error 10%
- Caudal de la conexión interferente 3.75 Mb/s

Tipo de TCP	Sub índice	Rendimiento %	Caudal normalizado* (Throughput)
TCP (Tahoe)	0	56,7054	0,0579904
Reno	1	56,5239	0,0450805
New Reno	2	56,8098	0,0626219
Sack1	3	56,8967	0,0271509
Vegas	4	59,8442	0,04678
Fack	5	57,0529	0,0260693
Linux	6	59,1134	0,0277957

Tabla 4-3 Rendimiento y Caudal de los distintos tipo de TCP inicialmente

Se tomo como clase de protocolo inicial el tipo Linux, debido a que tiene un alto rendimiento (ver tabla 4-3). El caudal máximo teórico normalizado es de aproximadamente 0.0625 luego de que la conexión interferente está presente. En la Tabla 4-3, la columna llamada sub-índice indica cómo se llaman por número cada uno de los tipos de TCP

Para los escenarios de simulación, además de las condiciones para las simulaciones definidas antes, se usaron las siguientes características:

- Topología (ver Figura 4-8)
- Tipo de TCP inicial: Linux
- Agente de recepción TCPSink
- Aplicación sobre TCP: ftp
- Tipo de cola "DropTail"
- Topología "dumbbell"

Para todos los escenarios se realizaron simulaciones con dos diferentes diagramas de influencia, cada uno de estos diagramas difiere del anterior en el numero de rangos que se usan para describir la variable rendimiento. A continuación se muestran, se les llama, respectivamente, tipo 1 (Figura 4-10) y tipo 2 (Figura 4-10). La principal razón de tomar dos diagramas de influencia es debido a que se desea probar que al tener un mayor número de estados de rendimiento el sistema puede distinguir un mayor número de situaciones, por lo que podrá tomar en teoría mejores acciones.

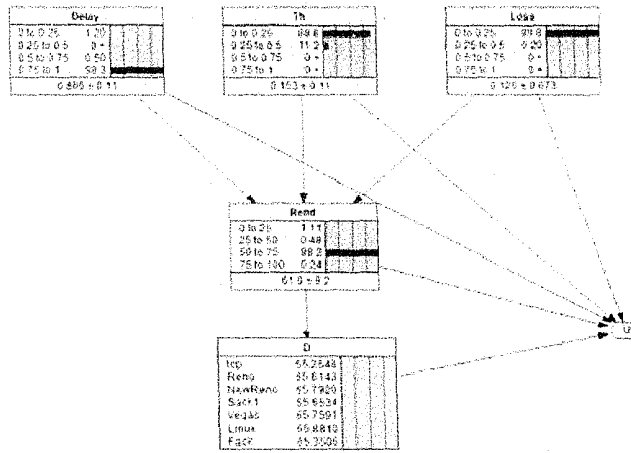


Figura 4-9 Diagrama de influencia simple, Tipo 1

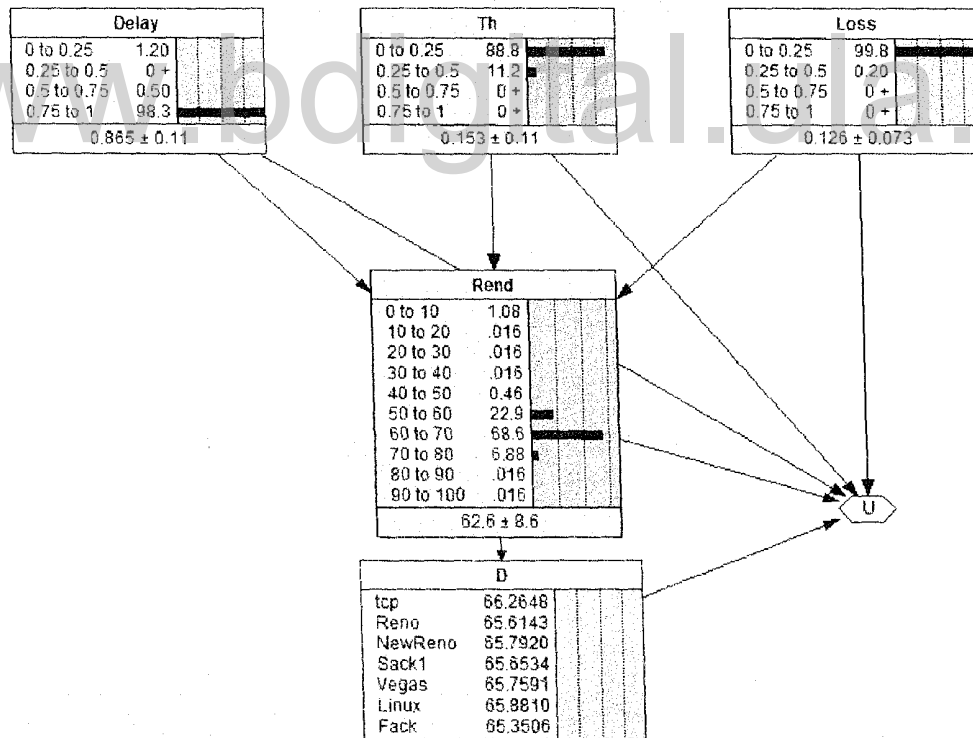


Figura 4-10 Diagrama de influencia de distribución de frecuencia para solo el rendimiento, Tipo 2

4.2. Escenario de simulación con decisión basada al rendimiento

En los siguientes escenarios se plantea que el sistema decida si debe ejecutar los cambios en base al comportamiento del rendimiento

- **Escenario de pérdida de paquetes**

En este escenario se van a realizar simulaciones con diferentes tasas de pérdidas, una variación leve (menor o igual al 16%) en el escenario 1 y una variación fuerte (menor o igual al 45%) en el escenario 2; de esta forma se pretende evaluar si el sistema de decisión puede mejorar la comunicación a diferentes tasas de pérdida.

Escenario 1	
Parámetro	Valor
Cola en el nodo 2:	10
Retardo de en el enlace 2-3:	10 ms
Tipo de conexión interferente / tasa de transmisión	UDP / 3.75 Mb/s
Valor inicial de la Tasa de pérdidas	0 %
Paso de la variación de la Tasa de pérdidas	2%
Valor final de la Tasa de pérdidas	16%

Escenario 2	
Parámetro	Valor
Cola en el nodo 2:	10
Retardo de en el enlace 2-3:	10 ms
Tipo de conexión interferente / tasa de transmisión	UDP / 3.75 Mb/s
Valor inicial de la Tasa de pérdidas	0 %
Paso de la variación de la Tasa de pérdidas	5%
Valor final de la Tasa de pérdidas	45%

- **Escenario de variación del Ancho de Banda**

En este escenario se va a variar el ancho de banda del canal, para determinar si el sistema se comporta de igual forma para escenarios con o sin restricciones de ancho de banda.

Parámetro	Valor
Cola en el nodo 2:	10
Retardo de en el enlace 2-3:	10 ms
Tipo de conexión interferente / tasa de transmisión	UDP / 6.75 Mb/s
Tasa de pérdidas	10%
Valor inicial del ancho de Banda (enlace 2-3)	10 Mb/s
Paso de la variación del ancho de Banda	10 Mb/s
Valor final del ancho de Banda	100 Mb/s

- **Escenario de variación de la Cola**

Este escenario permitirá determinar si el tamaño de la cola puede beneficiar o no a la toma de decisión del sistema. El sistema debería ajustarse al cambio dinámico de los mismos.

www.bdigital.ula.ve

Escenario 1	
Parámetro	Valor
Ancho de banda del enlace 2-3	10 Mb/s
Retardo de en el enlace 2-3:	10 ms
Tipo de conexión interferente / tasa de transmisión	UDP / 3.75 Mb/s
Tasa de pérdidas	10%
Valor inicial de la cola	0
Paso de la variación de la cola	2
Valor final de la cola	20

- **Escenario de variación del Retardo**

Este escenario de variación del retardo mostrará si el sistema de decisión puede o no manejar de forma adecuada distintos retardo. El retardo, por lo general, se traduce en distancia (distancia física o lógica) entre el nodo receptor y el emisor.

Parámetro	Valor
Ancho de banda del enlace 2-3	10 Mb/s
Cola en el nodo 2:	10
Tipo de conexión interferente / tasa de transmisión	UDP / 3.75 Mb/s
Tasa de pérdidas	10%
Valor inicial del retardo (enlace 2-3)	10 ms
Paso de la variación del retardo	10 ms
Valor final del retardo	100 ms

- *Escenario de variación del tipo y Tasa de transmisión de la conexión interferente*

Debido a que cada protocolo reacciona de forma diferente ante la congestión y tasa de pérdidas causadas por el canal y las otras conexiones que comparten el medio, es importante ver si la toma de decisión se adecua a diferentes tipos de conexión interferente, ya que cada tipo va a causar una interferencia con características diferentes.

Escenario 1	
Parámetro	Valor
Ancho de banda del enlace 2-3	10 Mb/s
Cola en el nodo 2:	10
Tasa de pérdidas	10%
Valor del retardo	10 ms
Tipo de conexión interferente	UDP
Valor inicial tasa de transmisión conexión interferente	2 Mb/s
Paso de la tasa de transmisión conexión interferente	2 Mb/s
Valor final de la tasa de transmisión conexión interferente	8 Mb/s

Escenario 2	
Parámetro	Valor
Ancho de banda del enlace 2-3	10 Mb/s
Cola en el nodo 2:	10
Tasa de pérdidas	10%
Valor del retardo	10 ms
Tipo de conexión interferente	TCP
Valor inicial de la Tasa de pérdidas	0 %
Paso de la variación de la Tasa de pérdidas	5%%
Valor final de la Tasa de pérdidas	15%

4.3. Escenario cuya decisión es basada en las pérdidas

En este escenario se van a realizar simulaciones con diferentes tasas de pérdidas, desde una variación leve (menor o igual al 16%) para el escenario 1 a una variación fuerte (menor o igual al 45%) para el escenario 2. De esta forma se pretende evaluar si el sistema de decisión puede mejorar la comunicación a diferentes tasas de pérdida. A diferencia de los escenarios anteriores, en donde el sistema basa su decisión en base al rendimiento, en estos escenarios el sistema tomara la decisión en base solamente a la tasa de pérdidas observada. Es importante resaltar que la tasa de pérdidas del canal es diferente a la tasa de pérdidas de la simulación, debido a que no solo se producen pérdidas en el canal, sino también debido a la cola (en la cola la principal causa de pérdidas es debido a la conexión interferente)

Escenario 1	
Parámetro	Valor
Cola en el nodo 2:	10
Retardo de en el enlace 2-3:	10 ms
Tipo de conexión interferente / tasa de transmisión	UDP / 3.75 Mb/s
Valor inicial de la Tasa de pérdidas	0 %
Paso de la variación de la Tasa de pérdidas	2%
Valor final de la Tasa de pérdidas	16%

Escenario 2	
Parámetro	Valor
Cola en el nodo 2:	10
Retardo de en el enlace 2-3:	10 ms
Tipo de conexión interferente / tasa de transmisión	UDP / 3.75 Mb/s
Valor inicial de la Tasa de pérdidas	0 %
Paso de la variación de la Tasa de pérdidas	5%
Valor final de la Tasa de pérdidas	45%

4.4. Simulaciones y Resultados experimentales

A continuación se presentan los resultados de los distintos escenarios de simulación descritos anteriormente, cabe destacar que solo se muestran las tablas resumen de todos los escenarios, y que las tablas de resultados de cada escenario de simulación esta en los anexos.

4.4.1. Resultados cuantitativos con decisión basada en el rendimiento

Los resultados están divididos por el tipo de diagrama de influencia que se uso para obtener los resultados:

- **Resultados para la Diagrama de influencia 1**

Los resultados obtenidos para la Diagrama de influencia 1 se pueden apreciar en la Tabla 4-4, en la misma se muestra que para este caso el sistema de decisión autonómico (SA) planteo hacer cambios el 47,4 % de los casos. En los escenarios simulados descritos en la Tabla 4-4, para los casos donde el SA plantea un cambio, el rendimiento mejoro en un 92,5 % y el caudal se incremento en un 85,2 %. El promedio de incremento de rendimiento es de 1,6% y el promedio de incremento de caudal es de 62,6% (estos números se obtiene de promediar todos los valores de rendimiento y caudal de las tablas del anexo 6.1 para el diagrama de influencia tipo 1). De la Tabla 4-4 podemos observar que en el escenario de tasa de perdida1 el sistema ejecuta pocos cambio (2/9) pero dichos cambios incrementan el caudal y el rendimiento al mismo tiempo. Esa situación no se repite para el escenario de tasa de perdida 2 (ver Tabla 6-3), y se debe a que el valor de rendimiento es muy bajo para altas tasas de perdida. y en el caso del caudal a que al hacerse un cambio del tipo de protocolo se produce un periodo donde no hay paquetes enviados de forma efectiva debido a la alta tasa de pérdidas y al cambio de protocolo. Es importante resaltar que se habla de altas tasas de pérdidas, con porcentajes de pérdidas en el canal mayores al 40%, de hecho, para las condiciones de simulación que se utilizaron para estos experimentos con tasas del 50%, no es posible transmitir data, ya que debido a esta tasa, más las pérdidas introducidas por la conexión interferente, se hace prácticamente imposible la comunicación. Para el caso de la variación de ancho de banda el sistema se adapto sin problemas, mejorando en todas las ocasiones el rendimiento y el caudal (ver Tabla 6-5). Otro escenario de simulación con condiciones particulares es el de tamaño de la cola, en el mismo, el sistema propone solamente un cambio en el que la comunicación mejora. Sin embargo, en el resto de los escenarios el sistema no propone cambio. En la Tabla 6-7 se puede observar que para las condiciones utilizadas en las simulaciones no hay diferencia en el rendimiento o en el caudal luego

que el tamaño de la cola supera el valor de 2, por lo que para los efectos de nuestro sistema de decisión este parámetro prácticamente irrelevante. Para los resultados obtenidos para el escenario de la variación del retardo (ver Tabla 6-9) el sistema mejoro al mismo tiempo el caudal y el rendimiento. En cuanto al escenario donde la conexión interferente es UDP el sistema mejoro ambas variable (Tabla 6-11), a diferencia del escenario con TCP (Tabla 6-13), donde el rendimiento no mejoro debido a una sutil diferencia numérica entre 57,78 y 57,16 (ver Tabla 6-13 con tasa de perdidas 0,15). Sin embargo, en esa simulación el caudal se incremento de 0,014 a 0,025 s (ver Tabla 6-13 con tasa de perdidas 0,15), que representa una mejoría cercana al 80%.

Escenario (Tipo 1)	Numero de simulaciones	Numero de simulaciones con cambios	Numero de simulaciones con incremento de rendimiento	Numero de simulaciones con incremento de Caudal	Porcentaje de simulaciones con incremento de rendimiento	Numero de simulaciones con incremento de Caudal	%cambios propuestos
tasa de perdida Escenario 1	9	2	2	2	100,0%	100,0%	22,2%
tasa de perdida Escenario 2	10	8	7	4	87,5%	50,0%	80,0%
Ancho de Banda Escenario 1	10	4	4	4	100,0%	100,0%	40,0%
Cola Escenario 1	10	1	1	1	100,0%	100,0%	10,0%
Retardo Escenario 1	10	6	6	6	100,0%	100,0%	60,0%
conexión interferente UDP	4	3	3	3	100,0%	100,0%	75,0%
conexión interferente TCP	4	3	2	3	66,7%	100,0%	75,0%
Totales	57	27	25	23	92,6%	85,2%	47,4%

Tabla 4-4 Tabla resumen de los resultados de las simulaciones con diagrama de influencia 1

○ **Resultados para la Diagrama de influencia 2**

Para las simulaciones realizadas con el Diagrama de influencia 2 se muestran los resultados obtenidos de forma resumida en la Tabla 4-5. El SA planteo hacer cambios en el 43,9% de los casos simulados. En el 96% (24/25) donde el SA planteo cambios, se incrementaron conjuntamente el caudal y el rendimiento, por lo que efectivamente se puede afirmar que el SA toma decisiones que mejoran de forma efectiva la comunicación. En comparación con los resultados obtenidos para el diagrama de influencia tipo 1, se obtuvo un mejora en el incremento del rendimiento de 92,6% a 96% para el caso del rendimiento, y un incremento del 85,2 al 96% para el caudal. Los resultados son mejores que en el caso anterior debido a que el sistema tiene una mayor riqueza a la hora de distinguir entre los distintos niveles del rendimiento (logra mejor caracterizar el contexto).

En la Tabla 4-5 se observa algo que no ocurría en la Tabla 4-4, es que para todos los escenarios en donde se mejoro el rendimiento también lo hizo el caudal, por lo que se puede concluir rápidamente que a un mayor número de estados en el sistema mayor es su efectividad. Sin embargo, el único precio a pagar por esta mejoría es que los scripts de clasificación toman mayor tiempo, y la tabla de utilidad crece considerablemente de 1792 entradas a 4480 entradas, lo cual a pesar de todo no es muy significativo para los entes involucrados en la comunicación, debido a las grandes capacidades de memoria y procesamiento que existen en la actualidad. Si se analiza el por qué de la mejoría en los parámetros de la comunicación, es debido a que el diagrama de influencia caracteriza los distintos protocolos de comunicación en función a sus características de rendimiento, y de los parámetros presentes en el diagrama de influencia (caudal, tasa de pérdidas y retardo). También es importante resaltar que en ocasiones el sistema no toma decisiones debido al funcionamiento propio del sub-sistema de cálculo de desempeño, el cual, por cierto, se podría implementar de muchas otras formas (como por ejemplo, tomando promedios y desviaciones estándar entre los puntos del desempeño, o calculando muestras de la derivada de la envolvente (envelope function) del desempeño).

Escenario (Tipo 2)	Numero de simulaciones	Numero de simulaciones con cambios	Numero de simulaciones con incremento de rendimiento	Numero de simulaciones con incremento de Caudal	Porcentaje de simulaciones con incremento de rendimiento	Numero de simulaciones con incremento de Caudal	%cambios propuestos
Tasa de perdida Escenario 1	9	2	2	2	100,0%	100,0%	22,2%
Tasa de perdida Escenario 2	10	6	5	5	83,3%	83,3%	60,0%
Ancho de Banda Escenario 1	10	4	4	4	100,0%	100,0%	40,0%
Cola Escenario 1	10	1	1	1	100,0%	100,0%	10,0%
Retardo Escenario 1	10	6	6	6	100,0%	100,0%	60,0%
conexión interferente UDP	4	3	3	3	100,0%	100,0%	75,0%
conexión interferente TCP	4	3	3	3	100,0%	100,0%	75,0%
Totales	57	25	24	24	96,0%	96,0%	43,9%

Tabla 4-5 Tabla resumen de los resultados de las simulaciones con diagrama de influencia 2

En cuanto al promedio de las mejoras en los dos tipos de diagramas de influencias, el rendimiento promedio es de 1,95% y el caudal promedio es de 66,35% para el caso actual, y para el caso anterior los valores eran de 1,6% y 62,6%, respectivamente (ver Tabla 4-6). Para los cálculos de la tabla de la Tabla 4-6 se tomaron los datos de todas las tablas del anexo 6. En el caso, por ejemplo, del caudal promedio con diagrama de influencia 1, se tomo el promedio de todas las tablas con esta condición, y se promediaron

todos los datos en la columna del Caudal promedio, tanto para con y sin SA. De esta forma se obtuvieron los datos de la Tabla 4-6.

	Caudal promedio con SA	Rendimiento promedio con SA	Producto Caudal x Rendimiento
Diagrama de influencia 1	62,60%	1,60%	0,0100
Diagrama de influencia 2	66,35%	1,95%	0,0129

Tabla 4-6 Comparación de resultados de acuerdo a el tipo de diagrama de influencia

Así, el que se desempeña mejor es el diagrama de influencia 2. Fue exitoso en incrementar el rendimiento y el caudal de forma conjunta en el 96% de los casos propuestos (ver tabla 4-5); adicionalmente, es el que tiene mejor relación caudal por rendimiento en promedio (ver Tabla 4-6).

www.bdigital.ula.ve

4.4.2. Resultados cuantitativos de la simulación con decisiones en base a las pérdidas

Para este escenario se utiliza solo el Diagrama de influencia 2, que es la que obtuvo mejores resultados en las pruebas anteriormente realizadas. A continuación se muestran los resultados obtenidos de las simulaciones.

En este escenario se está evaluando como responde el sistema cuando solo conoce la tasa de pérdidas, obtenida a partir de la traza de la simulación, es decir, el sistema no funciona prediciendo el estado general (criterio de rendimiento).

- **Escenario 1 Diagrama de influencia tipo 2 y tasa de pérdidas menores al 16 %.**

En la Tabla 4-7 se muestran los resultados para la variación de pérdidas en porcentajes menores al 16 %. Para este caso, el sistema mejora el rendimiento y el caudal en forma conjunta en un 100% de las veces que el sistema planteo un cambio. En la Figura 4-11 se muestra en la grafica que el rendimiento (Rend) y el caudal (Thn) con SA está por encima del rendimiento y del caudal para las simulaciones sin SA, lo cual indica que el sistema funciona con mejor desempeño cuando el SA está en uso. Para ese escenario, el rendimiento promedio se incremento en un 3,7%, y el caudal promedio se incremento en un 114 %.

Tasa de errores %	Rendimiento promedio % (sin sistema autónomico)	Caudal Normalizado (sin sistema autónomico)	Numero de cambios propuestos	Rendimiento promedio % (con sistema autónomico)	Caudal promedio Normalizado (con sistema autónomico)	cambios propuestos (Tiempo ms /tipo de TCP)
0	61,9497	0,342438	0			
2	65,0132	0,112563	1	68,2821	0,217631	27,000613 TCP/Linux -> TCP/Tahoe
4	64,0111	0,0593216	1	66,0921	0,14723	15,000613 TCP/Linux -> TCP/Tahoe
6	62,7275	0,0454619	1	64,5466	0,104063	15,000613 TCP/Linux -> TCP/Tahoe
8	60,7534	0,0335296	1	63,0999	0,0735979	33,00168 TCP/Linux -> TCP/Tahoe
10	59,1134	0,0277957	1	61,3737	0,0542469	27,000613 TCP/Linux -> TCP/Tahoe
12	58,6851	0,0229077	1	59,9828	0,0511893	18,00008 TCP/Linux -> TCP/Tahoe
14	56,9253	0,0167509	1	57,7487	0,029904	39,000613 TCP/Linux -> TCP/Tahoe
16	53,2205	0,00969973	1	57,2679	0,0219237	54,00008 TCP/Linux -> TCP/Tahoe
Veces que se incremento rendimiento						8/8
Veces que se incremento caudal						8/8

Tabla 4-7 Escenario 1 Diagrama de influencia tipo 2 en base a tasa de pérdidas menores al 16 %.

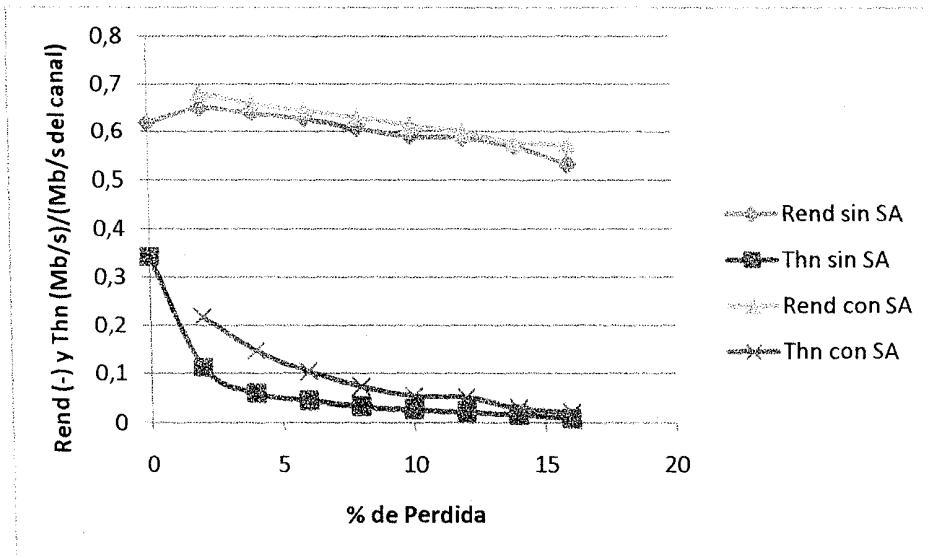


Figura 4-11 Escenario 1 Diagrama de influencia tipo 2 en base a tasa de pérdidas menores al 16 %.

- **Escenario 2 Diagrama de influencia tipo 2 y tasa de pérdidas menores al 45 %.**

En la Tabla 4-8 se pueden observar los resultados del escenario 2 para una Tasa de pérdidas que oscila entre el 0% y el 45% en la misma. El SA introdujo cambios para todas las tasas de pérdidas entre 5% y 30% y para 45% de pérdida, lo que se traduce en que introdujo cambios en casi el 75% de los casos. En todos los cambios introducidos el sistema mejoro el rendimiento y el caudal. El rendimiento promedio se incremento en un 127 % y el caudal promedio se incremento en un 119%. (Ver Tabla 4-8, si el porcentaje de incremento del rendimiento es de 227%, se dice que el incremento es de 127%, de igual forma se hace para el caudal promedio) El importante incremento en el rendimiento promedio se debe a que para el experimento con una tasa de pérdida de 45% (muy importante), el sistema logro incrementar el rendimiento de 1,32 a 10,91 al escoger un adecuado protocolo para solventar ese problema. En general, la razón de la mejora de los resultados es porque el tipo de TCP que el sistema escoge se comporta mejor en las condiciones del experimento, que el tipo que el sistema tenia inicialmente. Esto en si es una de las fortalezas del sistema propuesto, y consiste en que el sistema tiene caracterizado el desempeño de cada tipo de TCP en base a las condiciones de la red, y por lo tanto, puede escoger cual es el que mejor se adecua para cada situación. También, es importante resaltar que el sistema puede plantear varios cambios, como por ejemplo en la Tabla 4-8 para una tasa de perdidas, en donde el sistema plantea dos cambios: a los 45,00168 ms plantea cambiar de TCP/Linux -> TCP/Sack1, y luego, plantea cambiar a los 57,00168 ms de TCP/Sack1 -> TCP/Tahoe. A partir de esos dos cambios, el caudal y el rendimiento mejoran considerablemente. La habilidad de que el sistema pueda plantear varios cambios demuestra la flexibilidad del mismo, y demuestra

que el sistema puede efectuar cambios sin importar cuál es el tipo de TCP que se estaba usando previamente.

Tasa de errores %	Rendimiento promedio % (sin sistema autonómico)	Caudal Normalizado (sin sistema autonómico)	Numero de cambios propuestos	Rendimiento promedio % (con sistema autonómico)	Caudal Normalizado (con sistema autonómico)	cambios propuestos (Tiempo ms /tipo de TCP)	Mejora % en Rendimiento (con respecto al valor sin SA)	Mejora % en Caudal (con respecto al valor sin SA)	
0	61,9497	0,342438	0						
5	63,5108	0,0517296	1	65,6532	0,120953	15,000613 TCP/Linux -> TCP/Tahoe	103,37	233,82	
10	59,1134	0,0277957	1	61,3737	0,0542469	27,000613 TCP/Linux -> TCP/Tahoe	103,82	195,16	
15	56,4405	0,0148373	0						
20	50,6641	0,009984	1	52,0231	0,015032	18,00008 TCP/Linux -> TCP/Tahoe	102,68	150,56	
25	38,2617	0,00356373	2	43,1804	0,00637947	45,00168 TCP/Linux -> TCP/Sack1 57,00168 TCP/Sack1 -> TCP/Tahoe	112,86	179,01	
30	32,0556	0,00115093	1	37,5587	0,00330773	18,00008 TCP/Linux -> TCP/Tahoe	117,17	287,40	
35	24,5275	0,0002496	0						
40	18,8261	0,00067253	0						
45	1,32631	0,00017333	1	10,9184	0,00047173 3	63,000613 TCP/Linux -> TCP/Tahoe	823,22	272,16	
Promedio								227,19	219,68
Veces que se incremento rendimiento						7/7			
Veces que se incremento caudal						7/7			

Tabla 4-8 Escenario 2 Diagrama de influencia tipo 2 decisión en base a tasa de pérdidas menor a 45%

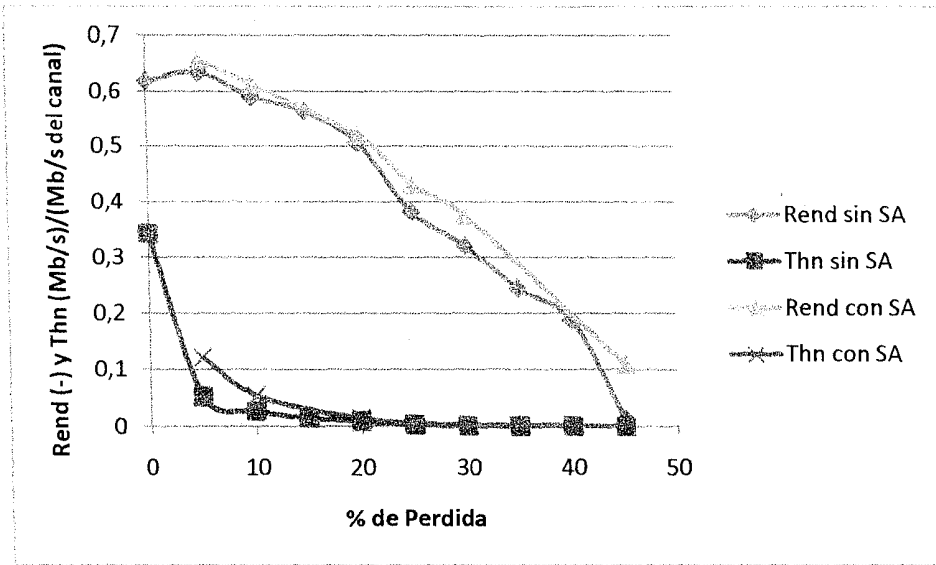


Figura 4-12 Escenario 2 Diagrama de influencia tipo 2 decisiones en base a tasa de pérdidas menor a 45%

En la Figura 4-12 se muestran los valores obtenidos de las simulaciones. En la figura 4-12 la curva de SA está por encima de sin SA; sin embargo, debido a la escala no se puede apreciar con facilidad lo mismo para el Caudal (Thn), por lo que hacemos un zoom a la parte del Caudal, como se observa en la Figura 4-13. En la Figura 4-13 se aprecia con facilidad que el caudal con sistema SA supera al caudal sin SA, en especial, para valores bajos de pérdidas (<25%).

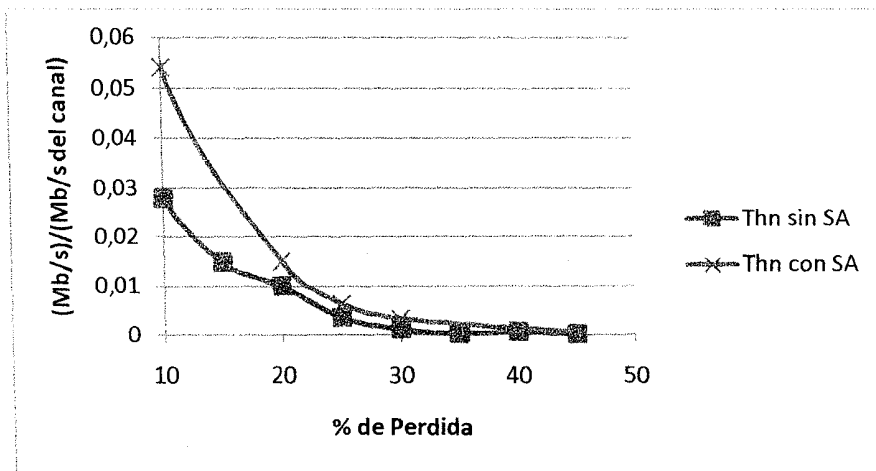


Figura 4-13 Caudal del Escenario 2 Diagrama de influencia tipo 2 decisiones en base a tasa de pérdidas menor a 45%

4.4.3. Resultados del proceso de aprendizaje

Un ejemplo del proceso de aprendizaje en la red bayesiana se puede observar en la Figura 3-21. En la Figura 3-21.a se muestra la RB con los valores iniciales en CPT (las probabilidades están distribuidas de forma uniforme, lo cual indica que el sistema está en un estado de total incertidumbre); en la Figura 3-21.b se muestra la RB con los nuevos valores en CPT (los cuales son adquiridos usando el proceso de aprendizaje de maximización de las expectativas y datos de múltiples simulaciones con cambios en el canal de forma aleatoria); y por último, la Figura 3-21.c se muestra la RB una vez concluidos todos los escenarios de simulación propuestos (aprendizaje por conteo).

Una muestra de la calidad del aprendizaje sobre la utilidad se muestra en la Tabla 4-9. En la misma se aprecia que el Valor de U (Utilidad) mejora según el sistema va aprendiendo. En esa tabla, cada fila es una simulación diferente. Por ejemplo, en la primera fila de la Tabla 4-9 el valor de utilidad comienza en 0, cuando el sistema no ha comenzado el aprendizaje. Al finalizar el primer periodo de aprendizaje el valor es de 52.455 (aprendizaje por EM), luego de ese periodo el sistema empieza a tomar decisiones y sigue aprendiendo (aprendizaje por conteo). Al finalizar las simulaciones (segundo periodo de aprendizaje), el valor de U toma el valor de 55.501. En todos los casos de la tabla 4-9, vemos como el proceso de aprendizaje permite mejorar la utilidad (U) permanentemente, por lo que es un componente fundamental de nuestro sistema autónomo.

Valor inicial U	Valor luego de periodo de entrenamiento (aprendizaje por EM)	Valor al finalizar de las simulaciones U (Aprendizaje por Conteo)	Condiciones de las Variables				
			Proto	Delay	Th	Loss	Rend
0	52.455	55.501	Linux	0.5 to 0.75	.25 to 0.50	0 to 0.25	50 to 60
0	47.065	47.363	Linux	0.5 to 0.75	0.5 to 0.75	0 to 0.25	50 to 60
0	50.569	50.926	Linux	0.75 to 1	0 to 0.25	0.25 to 0.5	50 to 60
0	51.035	53.422	Linux	0.75 to 1	0.5 to 0.75	0 to 0.25	60 to 70

Tabla 4-9 Muestra del aprendizaje sobre la tabla de utilidad

4.4.4. Análisis de las variables clásicas de rendimiento en los sistemas comunicacionales para el escenario 1 de pérdidas

Vamos a estudiar uno de los casos considerados en nuestros experimentos, el escenario 1 de simulación de pérdidas con el diagrama de influencia tipo 2, para una pérdida de 16% (Tabla 4-10). En esta simulación, se propone un cambio de TCP/Linux a TCP/Tahoe a los 42.00008 ms

Tasa de errores %	Rendimiento promedio % (sin sistema autónomico)	Caudal Normalizado (sin sistema autónomico)	Numero de cambios propuestos	Rendimiento promedio % (con sistema autónomico)	Caudal Normalizado (con sistema autónomico)	cambios propuestos (Tiempo ms /tipo de TCP)
16	53,2205	0,00969973	1	57,0934	0,0249813	42,00008 TCP/Linux -> TCP/Tahoe

Tabla 4-10 Escenario 1, tasa de errores del 16 con diagrama de influencia tipo 2

En el análisis de todas las variables, solo se tomara en consideración los datos obtenidos luego del cambio de protocolo, ya que antes del mismo no existe diferencia en el análisis con SA y sin SA.

A simple vista en la Figura 4-14 no se puede apreciar en cuál de los dos escenarios existe menor retardo, por lo que una forma simple de analizar estos gráficos es realizando un análisis estadísticos simple de los datos (ver Tabla 4-11). En esa tabla se puede observar que el sistema sin SA tiene mayor retardo que el sistema con SA, lo cual afirma que el cambio propuesto es beneficioso. Sin embargo, la desviación es menor sin SA, lo cual, dependiendo de la aplicación, puede o no ser perjudicial. Una gran variación del retardo es crítica para aplicaciones con datos de voz (por ejemplo, VoIP). Por último, el retardo acumulado (es decir la suma de todos los retardos, ver resultado sumatoria de la Tabla 4-11) es mayor para el sistema sin SA. Por lo tanto, para este ejemplo el sistema tiene menor retardo (77,314 vs 77,604) con el cambio propuesto (es decir con SA), con la única desventaja que la variación del retardo es mayor en un 15%.

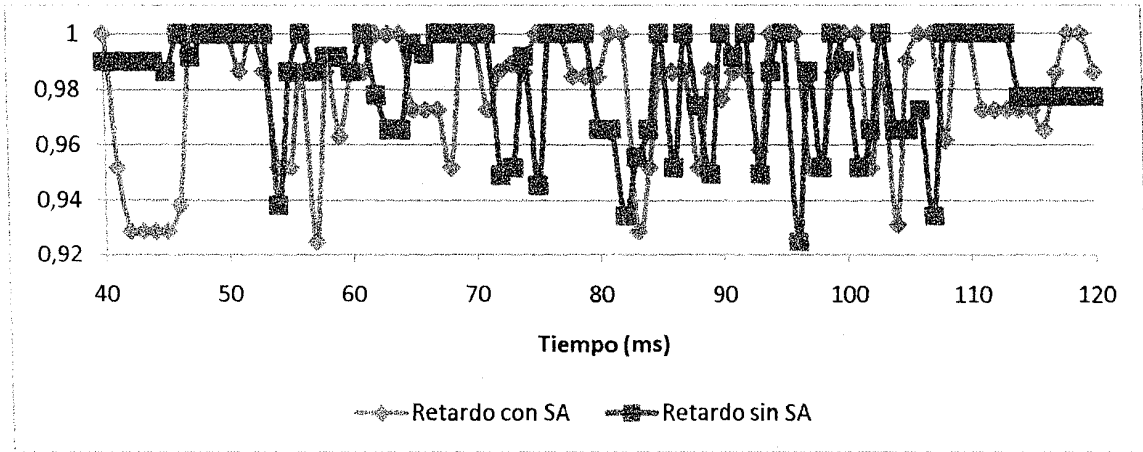


Figura 4-14 Retardo con SA y sin SA, simulación de la Tabla 6-2 para un 16% de pérdidas

	Retardo Promedio	Desviación estándar	sumatoria
Retardo sin SA	0,982	0,020	77,604
Retardo con SA	0,979	0,023	77,314

Tabla 4-11 Tabla comparación del retardo

Con respecto al caudal, el mismo es mayor, como se puede observar en la Figura 4-15 y en la Tabla 4-12. El único aspecto negativo con respecto al caudal con SA, es que tiene una mayor desviación. Sin embargo, la cantidad de datos transferidos es mucho mayor (0,99 vs 2,34), lo cual se traduce en $2,34/0,99=2,36$ veces más datos), compensando ampliamente el problema de la desviación.

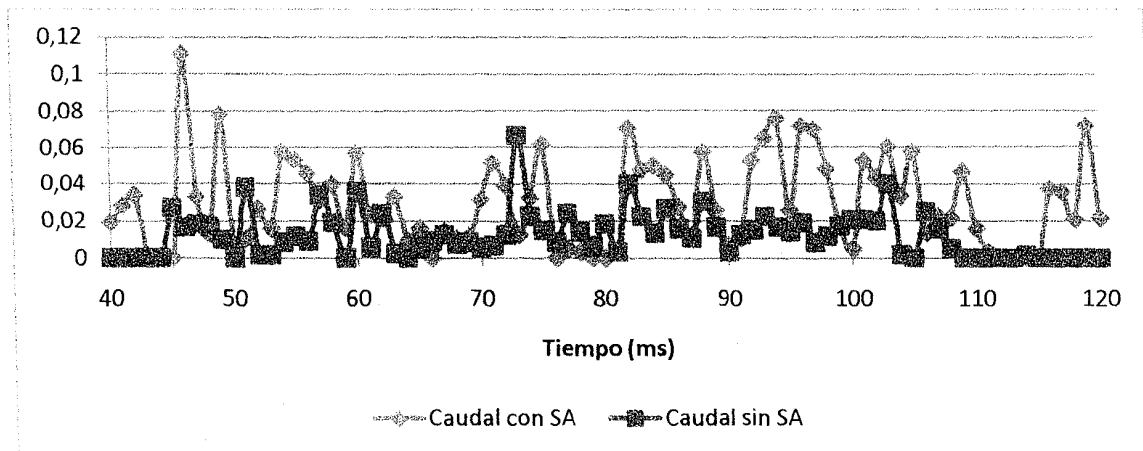


Figura 4-15 Caudal con SA y sin SA simulación de la Tabla 6-2 para un 16% de pérdidas

	Caudal Promedio	Desviación estándar del Caudal	Sumatoria
Caudal sin SA	0,013	0,012	0,996736
Caudal con SA	0,030	0,025	2,344608

Tabla 4-12 Tabla comparación del Caudal

La tasa de pérdidas con SA es notablemente menor que sin SA, esto es debido a las características propias del protocolo TCP/Tahoe (con SA) con respecto a TCP/Linux (sin SA). Esa característica el sistema la explota debido a su capacidad de aprendizaje (Ver Figura 4-16), que le sirve en el momento de tomar decisiones.

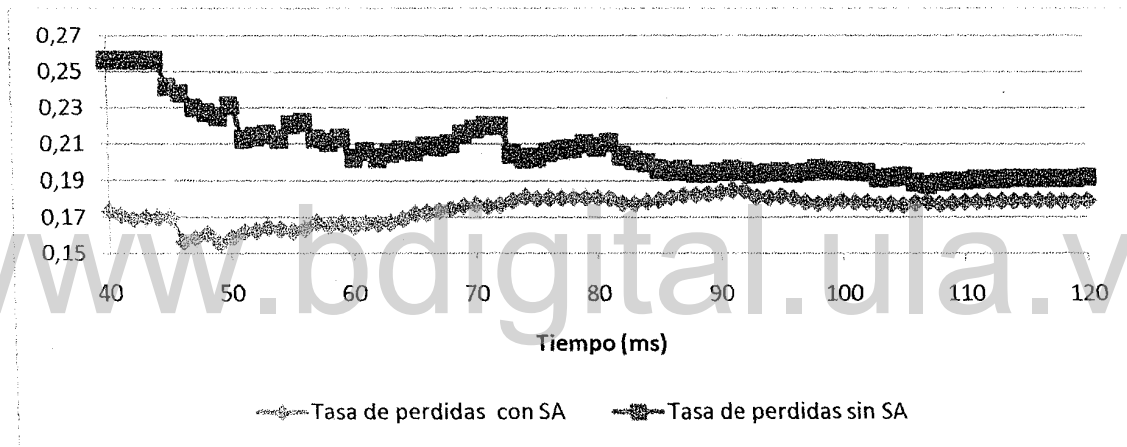


Figura 4-16 tasa de pérdidas con SA y sin SA simulación de la Tabla 6-2 para un 16% de pérdidas

4.4.5. Resultados cualitativos

En esta sección se resaltan la mayoría de las cualidades de los sistemas de comunicaciones autonómicas consultadas, y se comparan con la propuesta en el presente trabajo de investigación.

Trabajo	Sistema de comunicación autonómico basado en redes bayesianas y ontologías (El presente trabajo)	Une approche pour la composition autonome de services de communication orientés QoS Application aux protocoles de transport configurables (Van, 2009)	Cognitive Networks Decision-making Approach Based on Influence Diagram (Qi, Zhang, Cao, Sun, & Tan, 2012)	Algunas cualidades de interés de un sistema autonómico
Atributo	PT	VAN	RA	
Base de conocimiento	Ontología/ bayesiana/Diagrama de influencia	Cadena de Markov	Red bayesiana/Diagrama de influencia	Este atributo es fundamental, por la posibilidad de usar conocimiento en el proceso de toma de decisión (razonar). En los casos de red bayesiana y diagrama de influencia, el sistema es más rico en su proceso de adaptación. En este punto ganan los sistemas con RB/Diagrama de influencia. Cumplen PT y RA
Capacidad predictiva	Si, a través de la Red bayesiana	No posee	Si, a través de la Red bayesiana	Un sistema sin capacidad predictiva necesita conocer todas las variables para tomar una decisión, por lo tanto esto es una desventaja. Cumplen: PT y RA
Evaluación de Desempeño general	Comparación entre el sistema sin control del sistema autonómico y con sistema autonómico, y Metas de Calidad de Servicio	Metas de Calidad de servicio QoS	Comparación entre el sistema sin control del sistema autonómico y con sistema autonómico	Si el sistema tiene más formas de evaluar su desempeño, tiene una mayor posibilidad de tomar una correcta decisión, es más flexible, etc. Cumple: PT
Mecanismo de acción	Cambio de protocolo de transporte en el emisor	Cambio de protocolo de transporte en el emisor	Cambios en el enrutados (DiffServ architecture)	Las implementaciones distribuidas son mucho más difíciles de coordinar, por tanto, entre menos entes estén involucrados en el sistema es más fácil la implementación Cumplen PT y VAN
Métricas	Distancia normalizada y/o cualquiera de las variables (Ejemplo rendimiento, caudal,	Distancia normalizada	Relación señal ruido pico (Peak Signal-to-Noise Ratio PSNR).	Entre mayor sea el numero de variables que el sistema tome en consideración para calcular el desempeño de mismo, la métrica será mejor.

	error etc.)				Cumplen PT y VAN
Mecanismo adaptativo	Aprendizaje sobre la RB y actualización de la tabla de utilidad	No tiene mecanismo de aprendizaje definido	Aprendizaje sobre la RB y actualización de la tabla de utilidad		Un sistema autónomo sin capacidad adaptativa, incumple con una de las cualidades básicas que lo describen. Cumplen: PT y RA
Implementa un lazo cerrado (MAPE)	Lazo cerrado de retroalimentación, sobre su diseño e implementación basado en la arquitectura DAISY	Lazo de control cerrado para ATP	Lazo cerrado basado en la Arquitectura FOC-ALE		Todo sistema de control autónomo debe contener lazos de retroalimentación para que el sistema pueda ser supervisado y controlado al mismo tiempo. Cumplen: PT, VAN y RA
Se auto-gestiona	El sistema se auto-gestiona, en base a las variables observadas por el sistema	Se auto-gestiona por medio de un autómata	Se auto gestiona por el diseño de la Arquitectura FOC-ALE		Una de las principales características que debe poseer un SA, es la capacidad de auto-gestionarse, para adaptarse, optimizarse, etc. Cumplen: PT, VAN y RA
Mecanismo de implementación actual	Simulado con NS2, Topología "dumbbell"	Emulado con Linux Network Emulator (NetEm)	Simulado con NS2. Topología "dumbbell"		Los simuladores son una buena aproximación inicial, para el desarrollo temprano de mecanismos adaptativo. Sin embargo, los entornos emulados son más cercanos a los mecanismos de despliegue en entorno reales. Ganador VAN
Posible métodos de implementación del sistema	En agentes sobre sistema de laboratorio (sin piggybacking), una vez implementado sin las limitaciones de los simuladores se puede implantar en un entorno real, como un agente o servicio en el sistema.	Implementado en laboratorio. Requiere "piggybacking" campos adicionales para su implementación, emisor/receptor para ATP, no especifica cómo se puede aplicar fuera del entorno de prueba.	No especificada		Al plantear un nuevo sistema siempre es adecuado tener metas de implementación en entornos reales desde sus primeras etapas de desarrollo. Cumplen PT y VAN

Tabla 4-13 Tabla comparativa de distintos trabajos de Comunicación Autónoma

En la Tabla 4-13, se pueden apreciar algunos de los criterios que son deseables considerar al caracterizar los sistemas de comunicación autónoma. En la misma se resaltan cuales de los trabajos comparados cumplen con los criterios establecidos, siendo el presente trabajo el que más cumple con los mismos. En algunos de los atributos evaluados varios de los trabajos cumplen, sin embargo, ninguno cumplió con todos los atributos. Por ejemplo, el presente trabajo PT presenta una debilidad comparado con

otro de los trabajos, la cual es que está implementado en un simulador. Pero, prácticamente es el único atributo que no cumple, y tiene que ver más con el ambiente de prueba que con una característica de un entorno autonómico. En el caso del sistema propuesto por VAN, el mismo tiene un grave problema por no presentar un mecanismo de aprendizaje (su capacidad adaptativa se reduce enormemente, por lo que su proceso de retroalimentación es muy débil). En cuanto a RA, el mismo requiere un mecanismo de acción distribuido, lo cual representa una gran dificultad para su despliegue (implementación) comparado con los otros trabajos. Además, en ese enfoque las métricas usadas son muy limitadas, y deben ser mejoradas para que su proceso de toma de decisión sea más eficiente.

www.bdigital.ula.ve

5. Conclusiones y Recomendaciones

5.1.1. Conclusiones

En el presente trabajo el principal objetivo era el desarrollo de un sistema de comunicación autonómico basado en redes bayesianas y ontologías, el cual se inspira en la arquitectura DAISY, Esto se cumplió a cabalidad, con las únicas limitaciones impuestas por el sistema de simulación utilizado para las pruebas de funcionamiento. Para poder cumplir con el objetivo principal se desarrollo una ontología, la cual sirvió como base para construir una red bayesiana, que forma parte esencial del sistema de toma de decisiones.

En cuanto la meta de realizar pruebas con el SA desarrollado, las mismas fueron satisfactorias, mejorando la comunicación en línea general. En el caso del caudal, se logro una mejoría promedio del 66%; dicha mejoría solo requiere de la interacción con el emisor, es decir con la fuente de comunicación, por lo que su implementación es sencilla en tal sentido. Adicionalmente a esto, se probaron las capacidades del sistema en múltiples escenarios donde los parámetros típicos del canal del sistema de configuración fueron variados, probando así las capacidades de adaptación y de aprendizaje del sistema.

El sistema que se desarrollo posee, entre otras, la capacidad de auto-optimizarse, la cual se refleja principalmente en el proceso de aprendizaje que se realiza sobre la red bayesiana y sobre la tabla de utilidad. Ambos componentes se modifican conforme la experiencia del SA.

Para completar el trabajo no solo se realizaron pruebas cuantitativas, que aunque son muy útiles, no dan cuenta de algunas ventajas que el sistema ofrece con respecto a trabajos previos similares. En esta comparación se puede apreciar que el SA que se implemento cumple con muchas de las categorías evaluadas, siendo este el que más cumple con los criterios allí establecidos. Esos criterios tienen la cualidad de definir aspectos fundamentales en un SA.

Otra ventaja de nuestro sistema es el bajo costo computacional del mismo, ya que ninguno de los algoritmos utilizado tiene costo computacional mayor que lineal N , lo cual contribuye de forma positiva, en su posible implementación.

En el presente trabajo se encontraron múltiples dificultades técnicas, las cuales fueron superadas, en algunos casos, mediante extensivas revisiones a los antecedentes del mismo, algunas incluso abrieron puertas para seguir investigando en el desarrollo de SA. Adicionalmente se propuso una métrica, llamada rendimiento, la cual fue muy útil ya que la misma representa de alguna forma un compendio de todos los parámetros de interés. Otro aspecto interesante acerca del presente trabajo es que el sistema, al partir de una ontología de parámetros de calidad de servicio, le da gran robustez conceptual al mismo.

5.1.2. Recomendaciones

Se recomienda abordar algunos de tópicos que no se desarrollaron por completo en el presente trabajo. Entre estos tópicos se encuentran:

- La implementación del sistema de toma de decisión en un agente, el cual pueda tomar acciones de cambio sobre un sistema operativo real (como por ejemplo en Linux), para poder así continuar con el siguiente paso natural de esta investigación, que es probarla en un ambiente Emulado, siendo un candidato muy fuerte para esto el emulador (NetEm).
- En esa posible implementación, se puede también tomar como punto de partida lo ontología completa, de forma tal que la toma de decisión se enriquezca aun más. El uso de toda la ontología completa no solo permitirá optimizar el tráfico debido a las características del canal, sino que también tomará en consideración diferentes tipos de información a comunicar, y sus posibles requerimientos de calidad de servicio.
- Otro de los aspectos tiene que ver con que una vez probado el concepto en un sistema real, lo más conveniente sería desarrollar librerías propias específicas para sustituir lo hecho con las librerías de Netica.

www.bdigital.ula.ve

Bibliografía

Aguilar, J. (2014). <http://www.ing.ula.ve/~aguilar/>. Recuperado el 2 de 2 de 2014, de <http://www.ing.ula.ve/~aguilar/actividad-docente/IA/transparencias/clase91IA.pdf>

Aguilar, J., Subias, A., Travé-Massuyés, L., & Zouaoui, K. (GIIS 2012). "Situation Assessment in Autonomous Systems". *Proceedings of the 4th Global Information Infrastructure and Networking Symposium , IEEE Communication Society* .

Alhir, S. S. (2003). Understanding the Model Driven Architecture (MDA). *Methods & Tools Vol 11 No 3* , 17-24.

Altamura, G. (13 de 07 de 2009). *searchnetworking.it*. Recuperado el 21 de 01 de 2014, de *searchnetworking.it*: http://searchnetworking.techtarget.it/celare-la-complessita-della-rete-con-l-autonomic-networking/0,1254,19_ART_100616,00.html

Anandaraj, Dheepak, & Raja. (2011). Study of Ontology in Software Modelling Process and Life Cycle. *International Journal of Research and Reviews in Software Engineering, Vol. 1, No. 1* , 13-17.

Beltrá, F. R. (2002). *Metodos para obtener conocimiento utilizando redes bayesianas y procesos de aprendizaje con algoritmos evolutivos*. Sevilla: Univesidad de Sevilla.

Bencomo, N., & Belaggoun, A. (2013). Supporting Decision-Making for Self-Adaptive Systems: From Goal Models to Dynamic Decision Networks. *Lecture Notes in Computer Science Volume 7830* , pp 221-236.

Ben-Gal, I. F., Faltin, F., & Kenett, R. (2007). Bayesian Networks. En *Encyclopedia of Statistics in Quality & Reliability*. Wiley & Sons.

Consortium, H. (2009). *D4.10 Quality-of-Service Enabled Hydra Middleware*.

Crowley, M. (2004). *Evaluating Influence Diagrams Where we've been and where we're going*. British Columbia: University of British Columbia.

Dobson, Massacci, & Denazis. (2006). A Survey of Autonomic Communications. *ACM Transactions on Autonomous and Adaptive Systems Vol. 1, No. 2* , 223–259.

Exposito, E. (2010). *Méthodologie modèles et paradigmes pour la conception d'une couche transport de nouvelle génération*. Toulouse: LAAS-CNRS.

Exposito, E., Lamolle, M., & Gómez-Montalvo, J. (2010). Introducing an Ontology Driven Architecture for Distributed. *EKNOW '10 Proceedings of the 2010 Second International Conference on Information, Process, and Knowledge Management* , 7-12.

FP7, E. C. (2012). *Awareness: Self-Awareness in Autonomic Systems*. Recuperado el 21 de 02 de 2012, de Awareness is a Future and Emerging Technologies Proactive Initiative funded by the European Commission under FP7: <http://www.aware-project.eu/>

Friedman, n., Geiger, D., & Goldszmidt, M. (1997). Bayesian Network Classifiers. En *Machine Learning* (págs. 131-163). Berkeley: 7 Kluwer Academic Publishers.

Gasevic, D., Djuric, D., & Devedzi, V. (2006). *Model Driven Architecture and Ontology Development*. berlin: springer.

Ghahramani, Z. (2003). The Handbook of Brain Theory and Neural Networks. En M. Arbid, *Graphical models: parameter learning* (págs. 486-490). Boston: MIT Press .

Gómez-Montalvo, J. R. (2012). *PhD Thesis MODA: A Multimedia Ontology-Driven Architecture for Autonomic Quality of Service Management in Home Networks*. Toulouse: LAAS-CNRS.

Helspe, E. M., & van der Gaag, L. C. (2002). Building Bayesian Networks through Ontologies. *Proceedings of the 15th European Conference on Artificial Intelligence* , 680–684.

Hudec, M., & Fenz, S. (2009). Ontology-based generation of Bayesian networks. *International Conference on Complex, Intelligent and Software Intensive Systems* (págs. 712-717). Fukuoka, Japan: 10.1109/CISIS.2009.33.

IBM. (2005). *An architectural blueprint for autonomic computing*. New York: IBM.

IETF. (Octubre de 1989). *Internet Engineering Task Force*. Recuperado el 2 de 2 de 14, de <https://tools.ietf.org/html/rfc1122>

ISO, Organización Internacional de Normalización. (1994). *Open system interconnection basic reference model/ISO/IEC 7498-1*. ginebra: zuiza.

Kjaerulff, U. (2005). *Probabilistic Networks — An Introduction to Bayesian Networks and Influence Diagrams*. Aalborg: Springer.

Kephart, J., & Chess, D. (2003). The vision of autonomic computing. *IEEE Computer Society 0018-9162/03* , 41-50.

Kjaerulff, U. B., & Madsen, A. L. (2005). *Bayesian Networks and Influence Diagrams*. Aalborg Denmark: springer.com.

Koski, T., & Noble, J. M. (2009). *Bayesian Network, An Introduction*. Estocolmo: Wiley.

MDA Guide Version 1.0.1. (2003).

Norsys, S. C. (2010). *Netica API Programmer's Library Reference Manual Version 4.18*. Vancouver, BC, Canada: Norsys Software Corp.

Parashar, & Hariri. (2005). Autonomic Computing: An Overview. *BanÅtre et al. LNCS, Springer Verlag* , 247-259.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufman.

Pittarello, F., & Celentano, A. *Deployment of Multimodal Services: an Ontology Driven Architecture*. Trieste, Italy: IEEE International Conference on Pervasive Services ISBN 9781424413256 267-274.

Qi, J., Zhang, S., Cao, L., Sun, Y., & Tan, L. (2012). Cognitive Networks Autonomic Decision-making Approach Based on Influence Diagram. *JOURNAL OF NETWORKS* , VOL. 7, NO. 5.

Rong, P., Zhongli, D., & Peng, Y. (2006). A Bayesian Network Approach to Ontology Mapping. Baltimore, Maryland, USA.

Rong, P., Zhongli, D., & Peng, Y. (2006). BayesOWL: Uncertainty modeling in Semantic Web ontologies. *Soft Computing in Ontologies and Semantic Web. Studies in Fuzziness and Soft Computing* , vol 204.

Shachter, R. D. (1986). Evaluating Influence Diagrams. *Operations Research*, Vol. 34, No. 6. , 871-882.

Shachter, R. D. (1989). Evidence absorption and propagation through evidence reversals. *the Fifth Workshop on Uncertainty in Artificial Intelligence*, (págs. 303-308).

Shachter, R. D. (1988). Probabilistic inference and influence diagrams. *Operations Research* , 89-604.

Simon, D., Denazis, S., & Fernández, A. (December 2006.). A Survey of Autonomic Communications. *ACM Transactions on Autonomous and Adaptive Systems* , Vol. 1, No. 2.

Singh, Y., & Sood, M. (December 2010). The Impact of the Computational Independent Model for Enterprise Information System Development. *International Journal of Computer Applications Volume 11– No.8* , 21-26.

Stallings, W. (2004). *Redes e Internet de Alta Velocidad Rendimiento y Calidad de Servicio*. Madrid: Pearson.

Sucar, L. E. (2006). Redes Bayesianas. En B. S. ARAUJO, *Aprendizaje Automático: conceptos básicos y avanzados* (págs. 77-100). Pearson Educación.

Sucheta, N. P. (2001). A Bayesian network approach to making inferences in causal maps. *European Journal of Operational Research 128 issue 3* , 479-298.

Tamas, E., & Sandor, M. (1999). On the Distribution of Round-trip Delays in TCP/IP Networks. *Local Computer Networks. LCN '99. Conference on* .

Tenenbaum, J., Griffiths, T., & Kemp, C. (2006). Theory-based Bayesian models of inductive learning and reasoning. *TRENDS in Cognitive Sciences* , Vol.10 No.7 309-318.

Van, N. (2009). *PhD Thesis: Une approche pour la composition autonome de services de communication orientés QoS. Application aux protocoles de transport configurables*. Toulouse: Mathématiques Informatique Télécommunications de Toulouse (MITT).

Zaragoza, J. H., Sucar, E., & Morales, E. F. (2011). A Two-Step Method to Learn Multidimensional Bayesian Network Classifiers Based on Mutual Information Measures. *Proceedings of the Twenty-Fourth International Florida Artificial Intelligence Research Society Conference*.

www.bdigital.ula.ve

6. Anexos

www.bdigital.ula.ve

6.1. Anexo de Simulaciones y Resultados experimentales

A continuación se presentan los resultados de los distintos escenarios de simulación descritos anteriormente, es muy importante resaltar que en todas las tablas de este anexo el caudal esta normalizado con el caudal ancho de banda del enlace 2-3 (Figura 4-8). El mismo puede variar según el escenario de simulación.

En las graficas del presente anexo el Rendimiento (Rend) no tiene unidades, debido a que esta normalizado, y el caudal (Thn) esta de igual forma normalizado (Mb/s)/(Mb/s del canal) según el ancho de banda del enlace 2-3 (Figura 4-8) y como se dijo anteriormente el mismo puede variar según el escenario de simulación. Las graficas corresponden a los resultados de las simulaciones y reflejan visualmente los valores de la tabla correspondiente a casa escenario de simulación.

○ Resultados Escenario de simulación de pérdidas

➤ Escenario 1

• Escenario 1 Diagrama de influencia tipo 1

Tasa de errores %	Rendimiento promedio % (sin sistema autónómico)	Caudal Promedio Normalizado (sin sistema autónómico)	Numero de cambios propuestos	Rendimiento promedio % (con sistema autónómico)	Caudal Promedio Normalizado (con sistema autónómico)	cambios propuestos (Tiempo ms /tipo de TCP)
0	61,9497	0,342438	0			
2	65,0132	0,112563	0			
4	64,0111	0,0593216	1	65,0971	0,0815227	96,000162 TCP/Linux -> TCP/Tahoe
6	62,7275	0,0454619	0			
8	60,7534	0,0335296	0			
10	59,1134	0,0277957	0			
12	58,6851	0,0229077	0			
14	56,9253	0,0167509	0			
16	53,2205	0,00969973	1	57,0934	0,0249813	42,00008 TCP/Linux -> TCP/Tahoe
Tasa de incrementos (rendimiento)						2/2
Tasa de incrementos (caudal)						2/2

Tabla 6-1 Escenario 1 Diagrama de influencia tipo 1

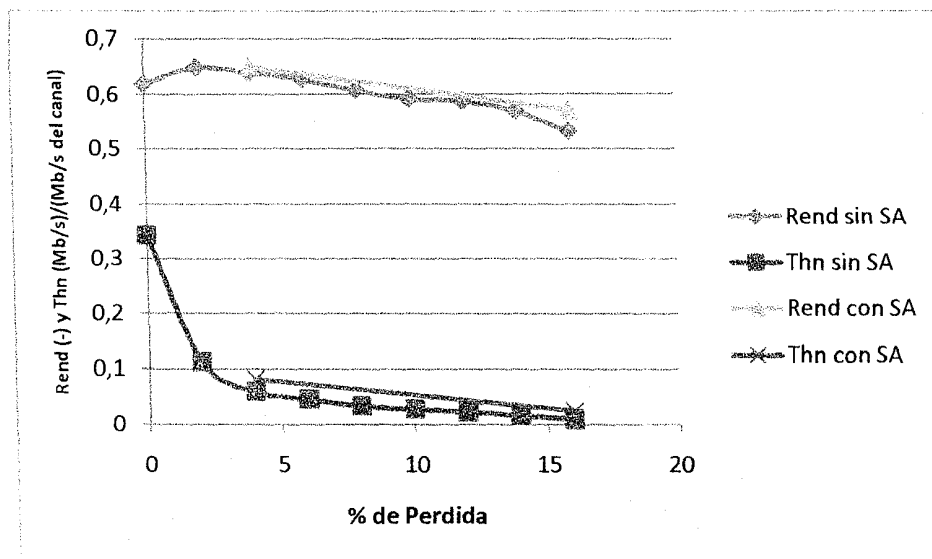


Figura 6-1 Escenario 1 Diagrama de influencia tipo 1

• **Escenario 1 Diagrama de influencia tipo 2**

Tasa de errores %	Rendimiento promedio % (sin sistema autonómico)	Caudal Promedio Normalizado (sin sistema autonómico)	Numero de cambios propuestos	Rendimiento promedio % (con sistema autonómico)	Caudal Promedio Normalizado (con sistema autonómico)	cambios propuestos (Tiempo ms /tipo de TCP)
0	61,9497	0,342438	0			
2	65,0132	0,112563	0			
4	64,0111	0,0593216	1	65,0971	0,0815227	96,000162 TCP/Linux -> TCP/Tahoe
6	62,7275	0,0454619	0			
8	60,7534	0,0335296	0			
10	59,1134	0,0277957	0			
12	58,6851	0,0229077	0			
14	56,9253	0,0167509	0			
16	53,2205	0,00969973	1	57,0934	0,0249813	42,00008 TCP/Linux -> TCP/Tahoe
Tasa de incrementos (rendimiento)						2/2
Tasa de incrementos (caudal)						2/2

Tabla 6-2 Escenario 1 Diagrama de influencia tipo 2

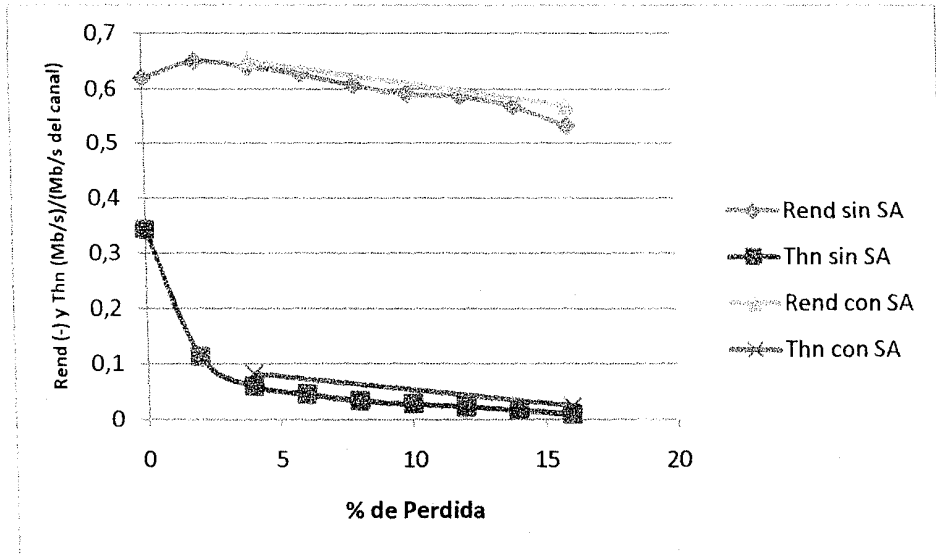


Figura 6-2 Escenario 1 Diagrama de influencia tipo 2

www.bdigital.ula.ve

➤ Escenario 2

• **Escenario 2 Diagrama de influencia tipo 1**

Tasa de errores %	Rendimiento promedio % (sin sistema autonómico)	Caudal Promedio Normalizado (sin sistema autonómico)	Numero de cambios propuestos	Rendimiento promedio % (con sistema autonómico)	Caudal Promedio Normalizado (con sistema autonómico)	cambios propuestos (Tiempo ms /tipo de TCP)
0	61,9497	0,342438	0			
5	63,5108	0,0517296	1	65,0384	0,0904528	66,00008 TCP/Linux -> TCP/Tahoe
10	59,1134	0,0277957	0			
15	56,4405	0,0148373	1	57,2243	0,0165157	105,00168 TCP/Linux -> TCP/Tahoe
20	50,6641	0,009984	1	53,2129	0,00825813	90,00008 TCP/Linux -> TCP/Tahoe
25	38,2617	0,00356373	1	45,5196	0,00643467	24,001147 TCP/Linux -> TCP/Tahoe
30	32,0556	0,00115093	1	34,9002	0,00102667	30,00008 0
35	24,5275	0,0002496	2	31,9479	0,00300267	39,000613 TCP/Linux -> TCP/Tahoe 93,00168 TCP/Tahoe -> TCP/Vegas
40	18,8261	0,00067253	1	2,1213	0,000104533	84,001147 TCP/Linux -> TCP/Tahoe
45	1,32631	0,00017333	1	14,8591	0,0005896	39,000613 TCP/Linux -> TCP/Tahoe
Tasa de incrementos (rendimiento)						7/8
Tasa de incrementos (caudal)						4/8

Tabla 6-3 Escenario 2 Diagrama de influencia tipo 1

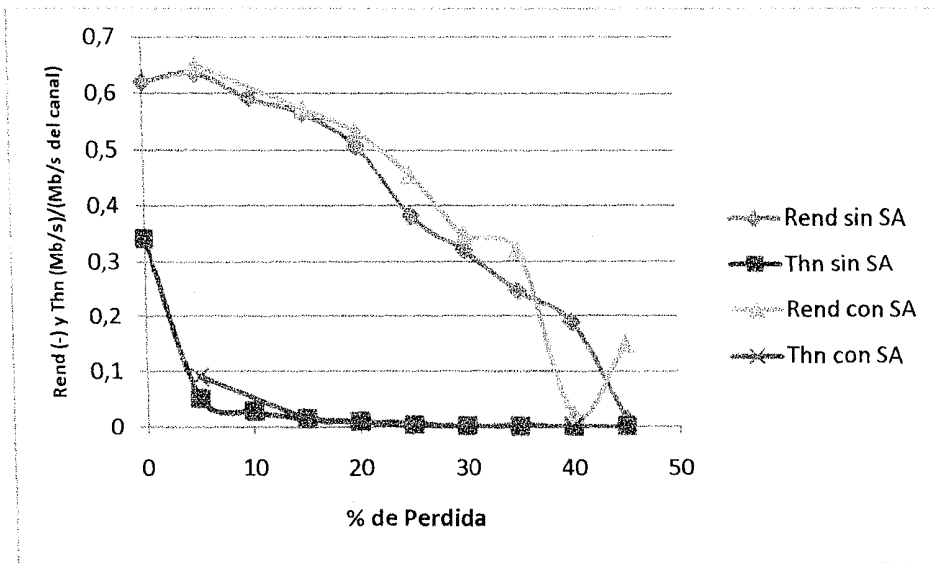


Figura 6-3 Escenario 2 Diagrama de influencia tipo 1

• **Escenario 2 Diagrama de influencia tipo 2**

Tasa de errores %	Rendimiento promedio % (sin sistema autonómico)	Caudal Promedio Normalizado (sin sistema autonómico)	Numero de cambios propuestos	Rendimiento promedio % (con sistema autonómico)	Caudal Promedio Normalizado (con sistema autonómico)	cambios propuestos (Tiempo ms /tipo de TCP)
0	61,9497	0,342438	0			
5	63,5108	0,0517296	1	65,0384	0,0904528	66,00008 TCP/Linux -> TCP/Tahoe
10	59,1134	0,0277957	0			
15	56,4405	0,0148373	1	57,2243	0,0165157	105,00168 TCP/Linux -> TCP/Tahoe
20	50,6641	0,009984	1	53,2129	0,00825813	90,00008 TCP/Linux -> TCP/Tahoe
25	38,2617	0,00356373	1	43,9269	0,00517973	24,001147 TCP/Linux -> TCP/Sack1
30	32,0556	0,00115093	1	36,6095	0,00185867	30,00008 TCP/Linux -> TCP/Sack1
35	24,5275	0,0002496	1	11,5869	0,0006176	39,000613 TCP/Linux -> TCP/Sack1
40	18,8261	0,00067253	0			
45	1,32631	0,00017333	0			
Tasa de incrementos (rendimiento)						5/6
Tasa de incrementos (caudal)						5/6

Tabla 6-4 Escenario 2 Diagrama de influencia tipo 2

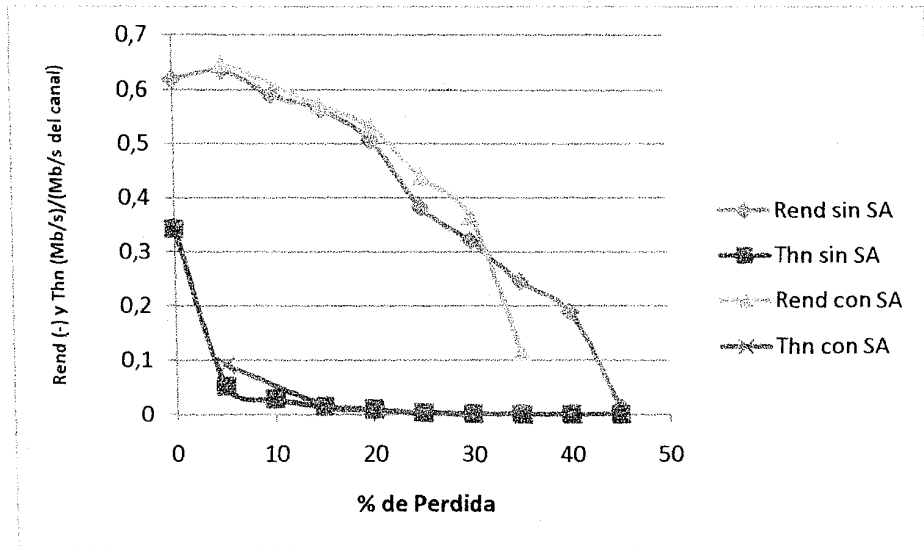


Figura 6-4 Escenario 2 Diagrama de influencia tipo 2

www.bdigital.ula.ve

○ **Resultados Escenario de variación del Ancho de Banda**

➤ Escenario de variación del Ancho de Banda

• **Escenario de Diagrama de influencia tipo 1**

Ancho de Banda (MB)	Rendimiento promedio % (sin sistema autonómico)	Caudal Promedio Normalizado (sin sistema autonómico)	Numero de cambios propuestos	Rendimiento promedio % (con sistema autonómico)	Caudal Promedio Normalizado (con sistema autonómico)	cambios propuestos (Tiempo ms /tipo de TCP)
10	58,6769	0,0239131	0			
20	60,1767	0,00894747	1	61,3531	0,0158672	108,0004 TCP/Linux -> TCP/Tahoe
30	60,5127	0,00893013	0			
40	59,648	0,00535773	1	61,891	0,0138945	30,00008 TCP/Linux -> TCP/Tahoe
50	60,8595	0,00540661	0			
60	60,8154	0,00474009	0			
70	61,058	0,00409859	0			
80	60,2977	0,00306973	1	61,7796	0,0061456	42,00008 TCP/Linux -> TCP/Tahoe
90	60,4566	0,00310844	1	62,2254	0,00546815	51,000376 TCP/Linux -> TCP/Tahoe
100	60,4931	0,00255355	0			
Tasa de incrementos (rendimiento)						4/4
Tasa de incrementos (caudal)						4/4

Tabla 6-5 Escenario de Diagrama de influencia tipo 1

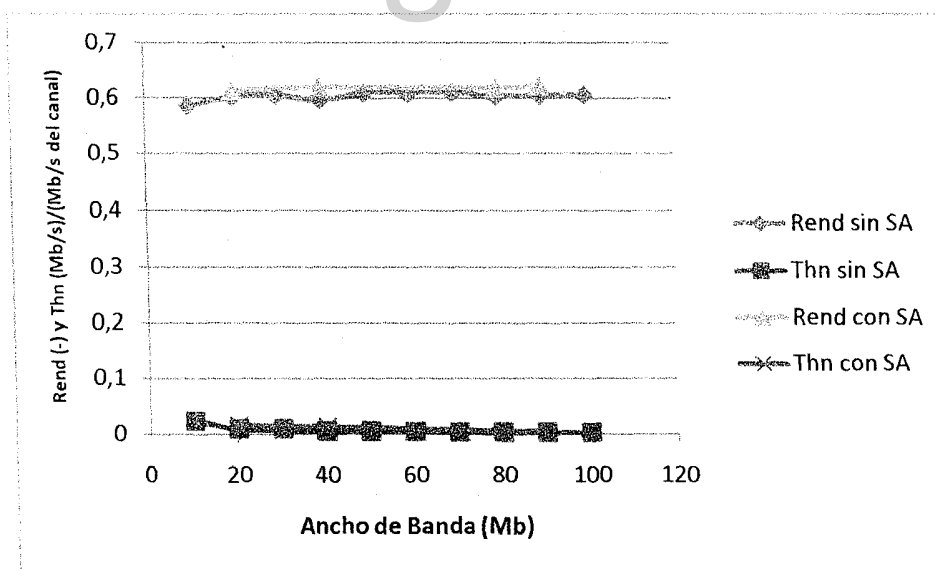


Figura 6-5 Escenario de Diagrama de influencia tipo 1

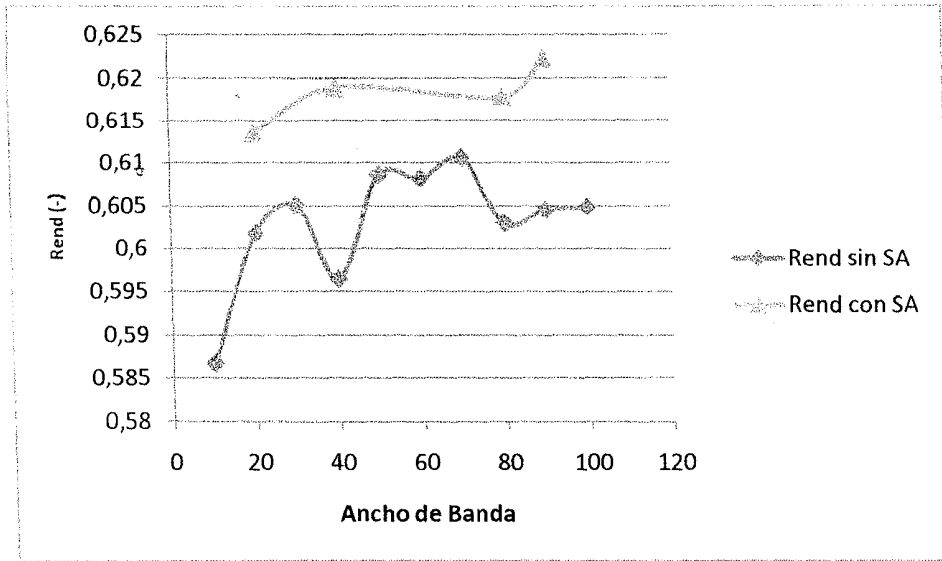


Figura 6-6 Rendimiento Diagrama de influencia tipo 1

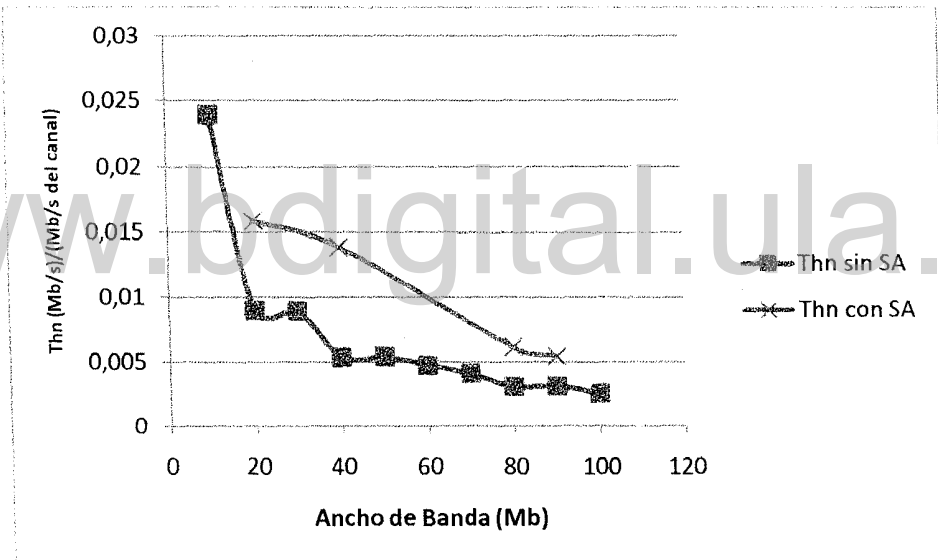


Figura 6-7 Caudal Diagrama de influencia tipo 1

• **Escenario de Diagrama de influencia tipo 2**

Ancho de Banda (MB)	Rendimiento promedio % (sin sistema autonómico)	Caudal Promedio Normalizado (sin sistema autonómico)	Numero de cambios propuestos	Rendimiento promedio % (con sistema autonómico)	Caudal Promedio Normalizado (con sistema autonómico)	cambios propuestos (Tiempo ms /tipo de TCP)
10	58,6769	0,0239131	0			
20	60,1767	0,00894747	1	61,3531	0,0158672	108,0004 TCP/Linux -> TCP/Tahoe
30	60,5127	0,00893013	0			
40	59,648	0,00535773	1	61,891	0,0138945	30,00008 TCP/Linux -> TCP/Tahoe
50	60,8595	0,00540661	0			
60	60,8154	0,00474009	0			
70	61,058	0,00409859	0			
80	60,2977	0,00306973	1	61,7796	0,0061456	42,00008 TCP/Linux -> TCP/Tahoe
90	60,4566	0,00310844	1	62,2254	0,00546815	51,000376 TCP/Linux -> TCP/Tahoe
100	60,4931	0,00255355	0			
Tasa de incrementos (rendimiento)						4/4
Tasa de incrementos (caudal)						4/4

Tabla 6-6 Escenario de Diagrama de influencia tipo 2

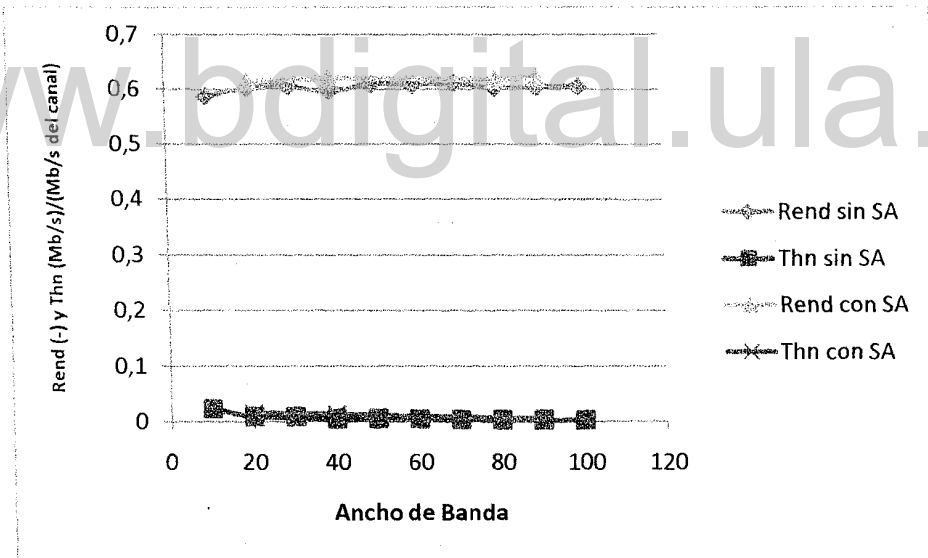


Figura 6-8 Escenario de Diagrama de influencia tipo 2

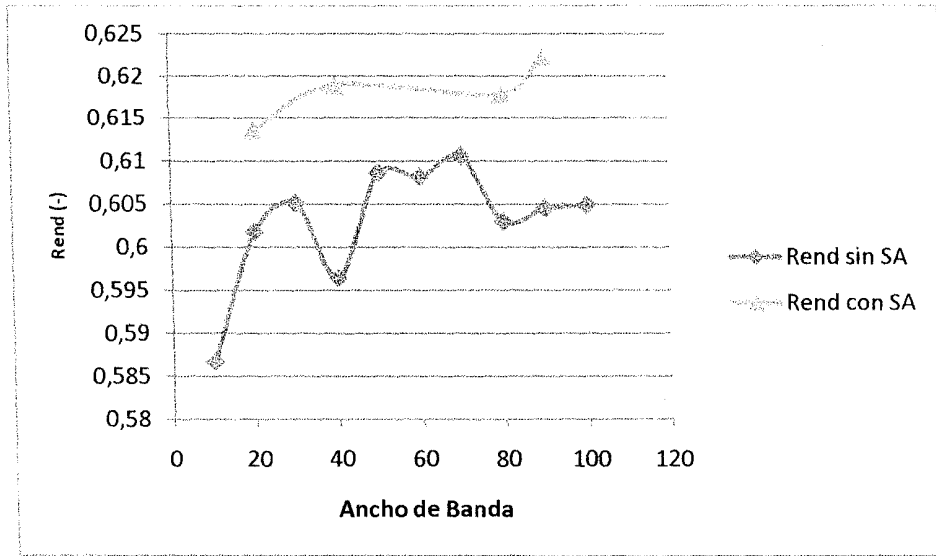


Figura 6-9 Rendimiento Diagrama de influencia tipo 2

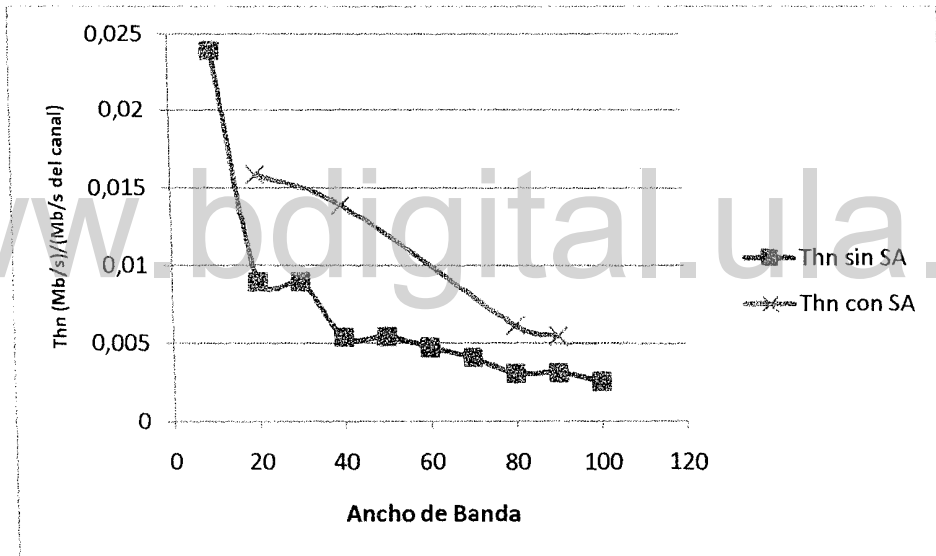


Figura 6-10 Caudal Diagrama de influencia tipo 2

○ **Resultados Escenario de variación del Tamaño de la Cola**

➤ Escenario de variación del Tamaño de la Cola

• **Escenario de Diagrama de influencia tipo 1**

Tamaño de la cola	Rendimiento promedio % (sin sistema autonómico)	Caudal Promedio Normalizado (sin sistema autonómico)	Numero de cambios propuestos	Rendimiento promedio % (con sistema autonómico)	Caudal Promedio Normalizado (con sistema autonómico)	cambios propuestos (Tiempo ms /tipo de TCP)
2	49,9429	0,00760587	1	53,9739	0,0144981	21.000969 TCP/Linux -> TCP/Tahoe
4	58,5493	0,0243568	0			
6	58,6769	0,0239131	0			
8	58,6769	0,0239131	0			
10	58,6769	0,0239131	0			
12	58,6769	0,0239131	0			
14	58,6769	0,0239131	0			
16	58,6769	0,0239131	0			
18	58,6769	0,0239131	0			
20	58,6769	0,0239131	0			
Tasa de incrementos (rendimiento)						1/1
Tasa de incrementos (caudal)						1/1

Tabla 6-7 Escenario de Diagrama de influencia tipo 1

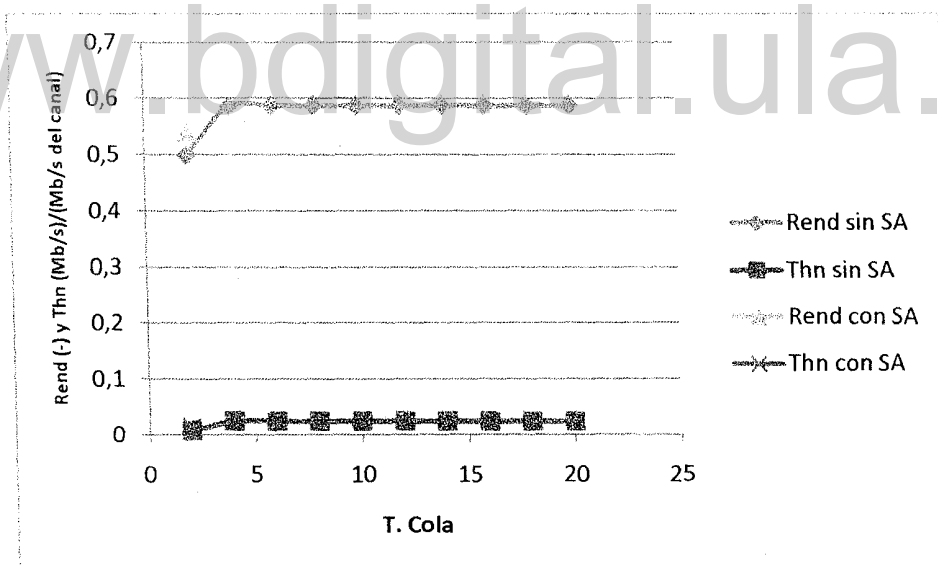


Figura 6-11 Escenario de Diagrama de influencia tipo 1

• **Escenario de Diagrama de influencia tipo 2**

Tamaño de la cola	Rendimiento promedio % (sin sistema autonómico)	Caudal Promedio Normalizado (sin sistema autonómico)	Numero de cambios propuestos	Rendimiento promedio % (con sistema autonómico)	Caudal Promedio Normalizado (con sistema autonómico)	cambios propuestos (Tiempo ms /tipo de TCP)
2	49,9429	0,00760587	1	53,9739	0,0144981	21.000969 TCP/Linux > TCP/Tahoe
4	58,5493	0,0243568	0			
6	58,6769	0,0239131	0			
8	58,6769	0,0239131	0			
10	58,6769	0,0239131	0			
12	58,6769	0,0239131	0			
14	58,6769	0,0239131	0			
16	58,6769	0,0239131	0			
18	58,6769	0,0239131	0			
20	58,6769	0,0239131	0			
Tasa de incrementos (rendimiento)						1/1
Tasa de incrementos (caudal)						1/1

Tabla 6-8 Escenario de Diagrama de influencia tipo 2

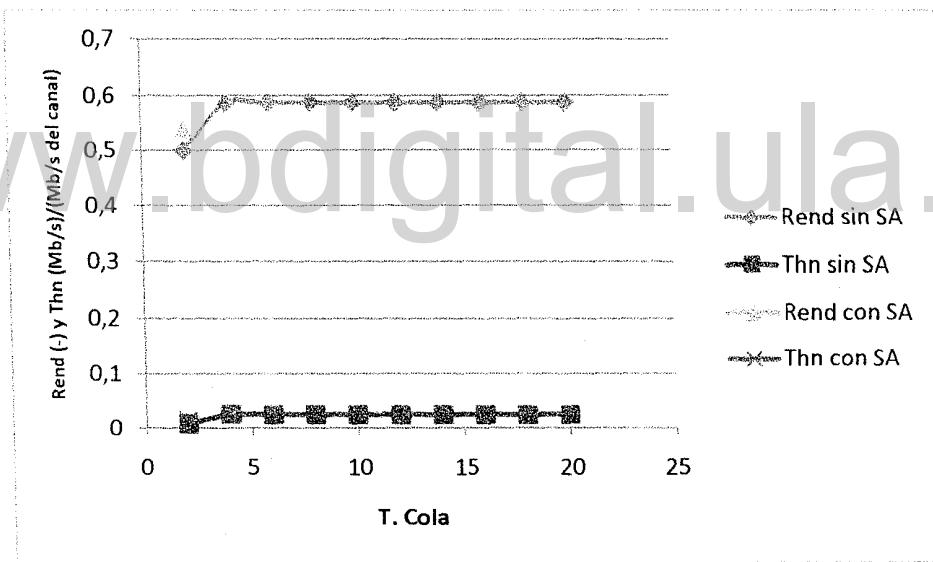


Figura 6-12 Escenario de Diagrama de influencia tipo 2

○ **Resultados Escenario de variación del Retardo**

➤ Escenario de variación del Retardo

• **Escenario de Diagrama de influencia tipo 1**

Retardo (ms)	Rendimiento promedio % (sin sistema autónomico)	Caudal Promedio Normalizado (sin sistema autónomico)	Numero de cambios propuestos	Rendimiento promedio % (con sistema autónomico)	Caudal Promedio Normalizado (con sistema autónomico)	cambios propuestos (Tiempo ms /tipo de TCP)
10	59,1134	0,0277957	0			
20	61,0569	0,0237536	1	62,016	0,0286629	78,00008 TCP/Linux → TCP/Tahoe
30	59,9804	0,0163557	0			
40	60,2567	0,0145669	0			
50	60,0637	0,0125147	1	61,8208	0,0162661	72,00008 TCP/Linux → TCP/Tahoe
60	60,9882	0,0114955	1	63,3226	0,0137701	72,00008 TCP/Linux → TCP/Tahoe
70	59,8615	0,00983147	1	61,8477	0,0120645	90,00008 TCP/Linux → TCP/Tahoe
80	59,4149	0,00855573	0			
90	61,051	0,0095056	1	62,4068	0,00980427	90,00008 TCP/Linux → TCP/Tahoe
100	59,0666	0,00693333	1	62,4835	0,00951307	84,001147 TCP/Linux → TCP/Tahoe
Tasa de incrementos (rendimiento)						6/6
Tasa de incrementos (caudal)						6/6

Tabla 6-9 Escenario de Diagrama de influencia tipo 1

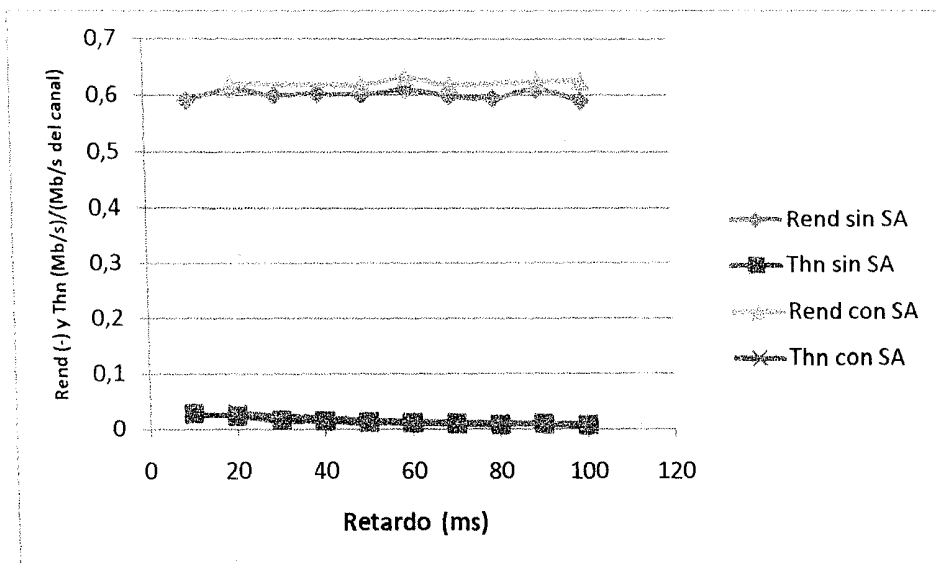


Figura 6-13 Escenario de Diagrama de influencia tipo 1

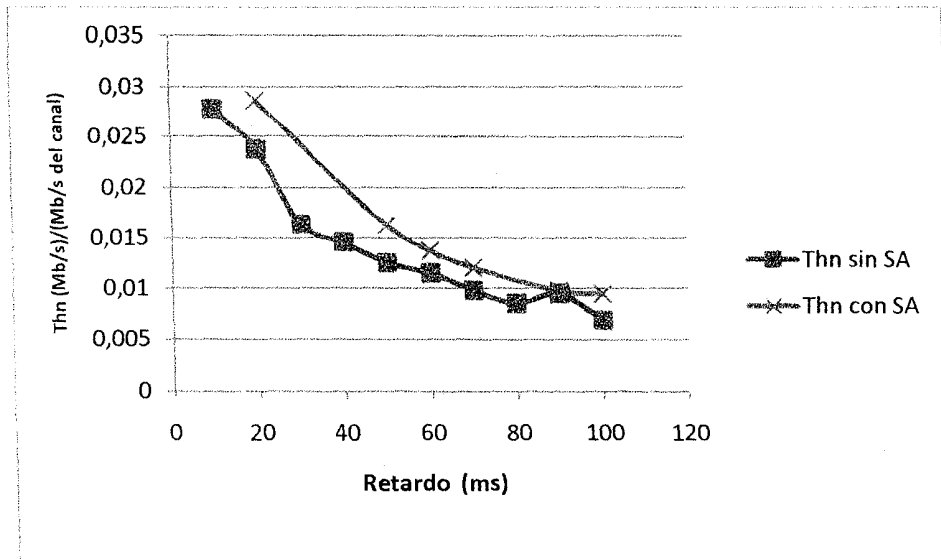


Figura 6-14 Caudal de Diagrama de influencia tipo 1

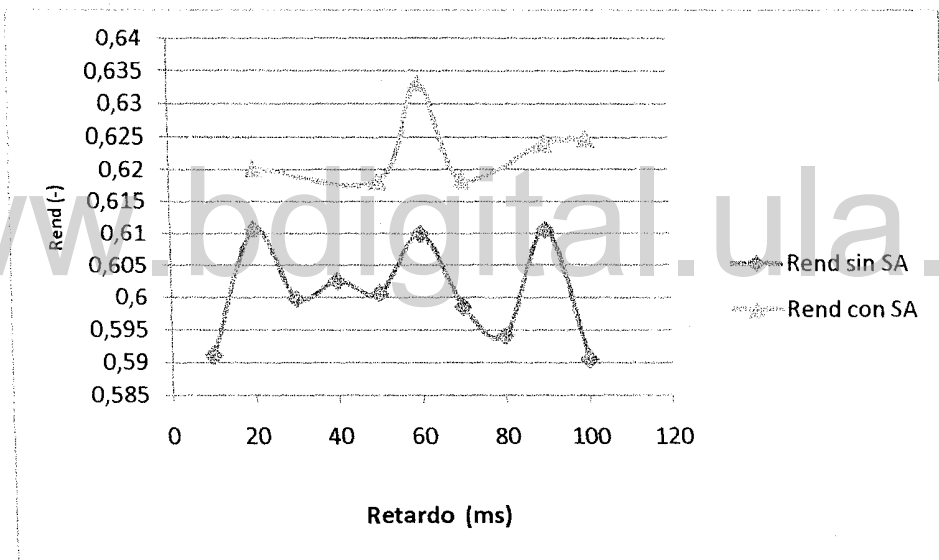


Figura 6-15 Rendimiento de Diagrama de influencia tipo 1

• **Escenario de Diagrama de influencia tipo 2**

Retardo (ms)	Rendimiento promedio % (sin sistema autonómico)	Caudal Promedio Normalizado (sin sistema autonómico)	Numero de cambios propuestos	Rendimiento promedio % (con sistema autonómico)	Caudal Promedio Normalizado (con sistema autonómico)	cambios propuestos (Tiempo ms /tipo de TCP)
10	59,1134	0,0277957	0			
20	61,0569	0,0237536	1	62,016	0,0286629	78,00008 TCP/Linux -> TCP/Tahoe
30	59,9804	0,0163557	0			
40	60,2567	0,0145669	0			
50	60,0637	0,0125147	1	61,8208	0,0162661	72,00008 TCP/Linux -> TCP/Tahoe
60	60,9882	0,0114955	1	63,3226	0,0137701	72,00008 TCP/Linux -> TCP/Tahoe
70	59,8615	0,00983147	1	61,8477	0,0120645	90,00008 TCP/Linux -> TCP/Tahoe
80	59,4149	0,00855573	0			
90	61,051	0,0095056	1	62,4068	0,00980427	90,00008 TCP/Linux -> TCP/Tahoe
100	59,0666	0,00693333	1	62,4835	0,00951307	84,001147 TCP/Linux -> TCP/Tahoe
Tasa de incrementos (rendimiento)						6/6
Tasa de incrementos (caudal)						6/6

Tabla 6-10 Escenario de Diagrama de influencia tipo 2

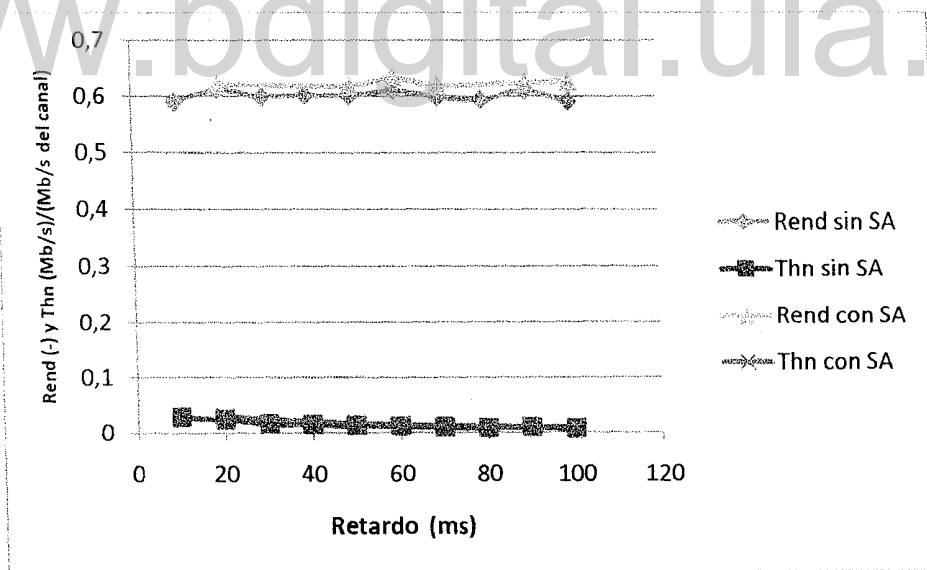


Figura 6-16 Escenario de Diagrama de influencia tipo 2

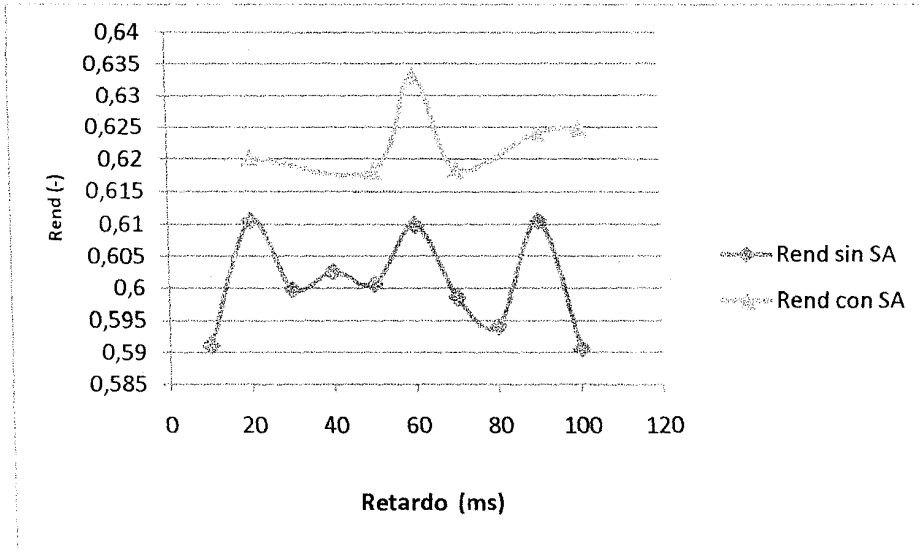


Figura 6-17 Rendimiento Diagrama de influencia tipo 2

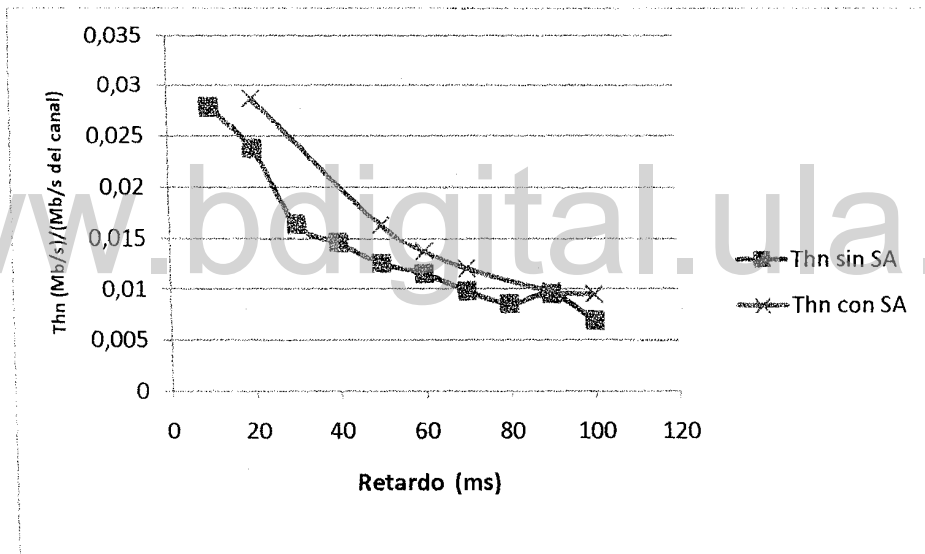


Figura 6-18 Caudal Diagrama de influencia tipo 2

- **Resultados Escenario de variación de tipo y Tasa de transmisión de la conexión interferente**

➤ Escenario de variación de tipo y Tasa de transmisión de la conexión interferente

• **Escenario de UDP de Diagrama de influencia tipo 1**

Tasa de UDP Mb/s	Rendimiento promedio % (sin sistema autonómico)	Caudal Promedio Normalizado (sin sistema autonómico)	Numero de cambios propuestos	Rendimiento promedio % (con sistema autonómico)	Caudal Promedio Normalizado (con sistema autonómico)	cambios propuestos (Tiempo ms /tipo de TCP)
2	59,2618	0,0248629	1	61,6034	0,0466133	69,00008 TCP/Linux -> TCP/Tahoe
4	58,6073	0,0205504	1	61,5137	0,0520075	24,00008 TCP/Linux -> TCP/Tahoe
6	59,0475	0,0273173	1	60,0351	0,0388411	72,00008 TCP/Linux -> TCP/Tahoe
8	58,6828	0,0273867	0			
Tasa de incrementos (rendimiento)						3/3
Tasa de incrementos (caudal)						3/3

Tabla 6-11 Escenario de UDP de Diagrama de influencia tipo 1

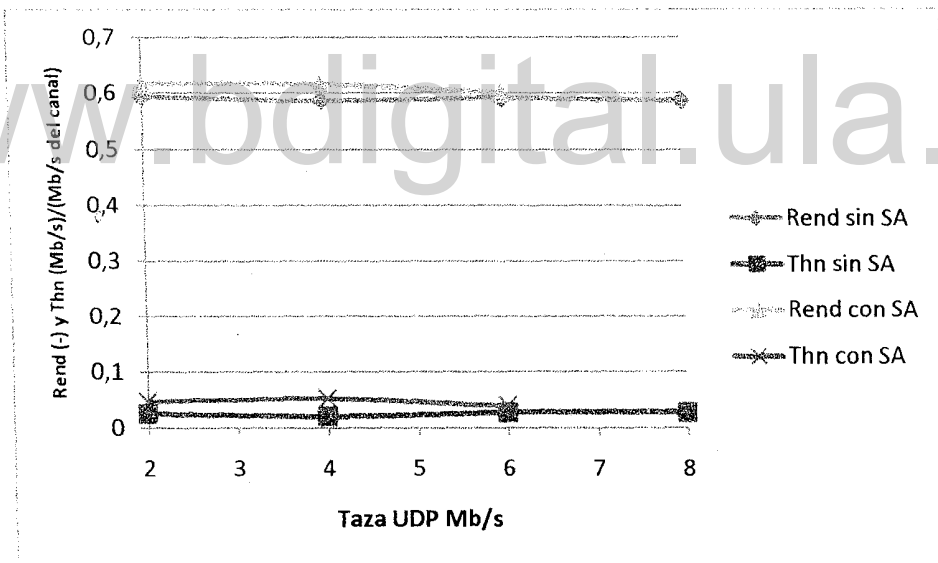


Figura 6-19 Escenario de UDP de Diagrama de influencia tipo 1

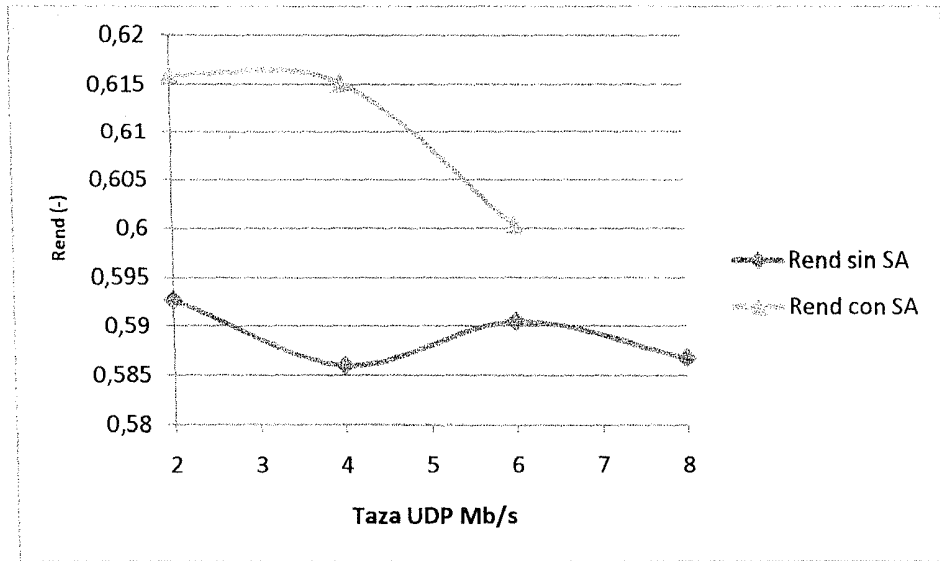


Figura 6-20 Rendimiento de UDP con de Diagrama de influencia tipo 1

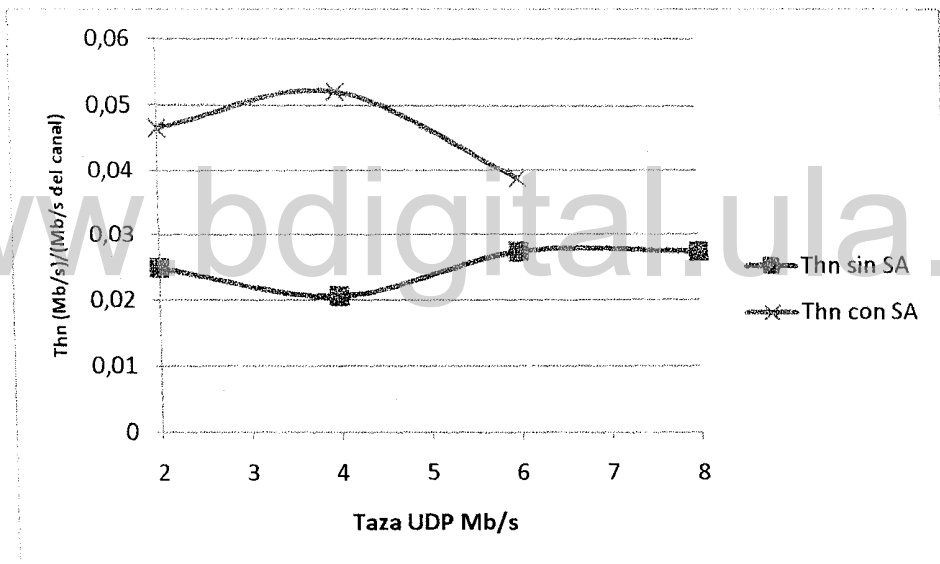


Figura 6-21 Caudal de UDP con de Diagrama de influencia tipo 1

• **Escenario de UDP de Diagrama de influencia tipo 2**

Tasa de UDP Mb/s	Rendimiento promedio % (sin sistema autonómico)	Caudal Promedio Normalizado (sin sistema autonómico)	Numero de cambios propuestos	Rendimiento promedio % (con sistema autonómico)	Caudal Promedio Normalizado (con sistema autonómico)	cambios propuestos (Tiempo ms /tipo de TCP)
2	59,2618	0,0248629	1	61,6034	0,0466133	69,00008 TCP/Linux -> TCP/Tahoe
4	58,6073	0,0205504	1	61,5137	0,0520075	24,00008 TCP/Linux -> TCP/Tahoe
6	59,0475	0,0273173	1	60,0351	0,0388411	72,00008 TCP/Linux -> TCP/Tahoe
8	58,6828	0,0273867	0			
Tasa de incrementos (rendimiento)						3/3
Tasa de incrementos (caudal)						3/3

Tabla 6-12 Escenario de UDP de Diagrama de influencia tipo 2

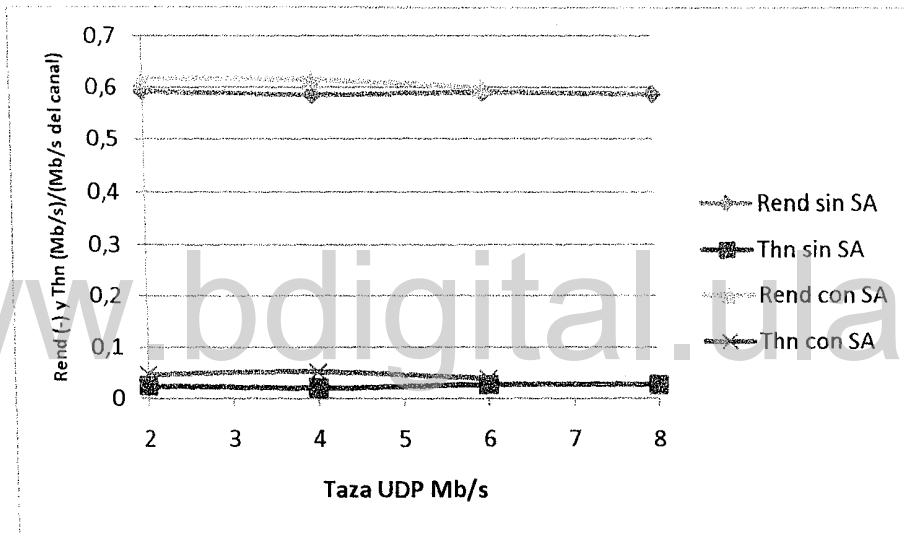


Figura 6-22 Escenario de UDP de Diagrama de influencia tipo 2

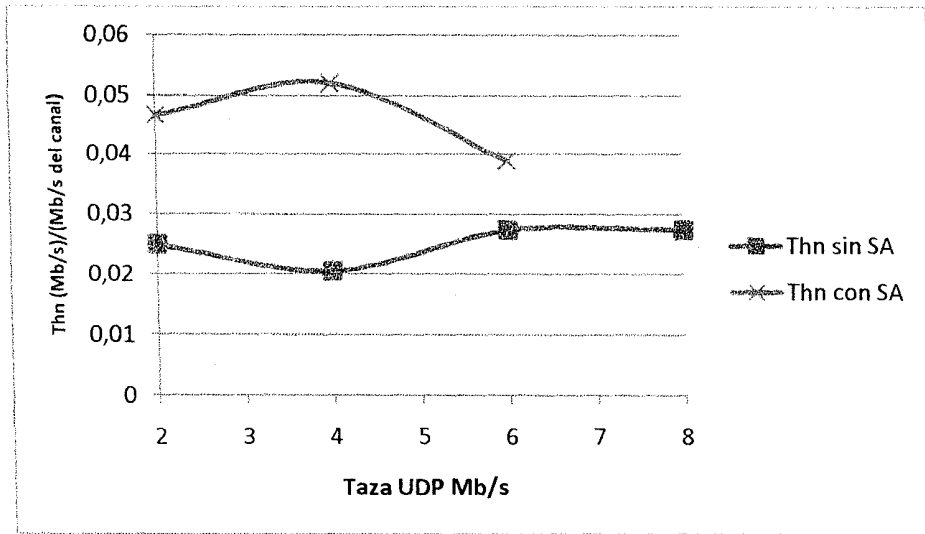


Figura 6-23 Caudal de UDP con de Diagrama de influencia tipo 2

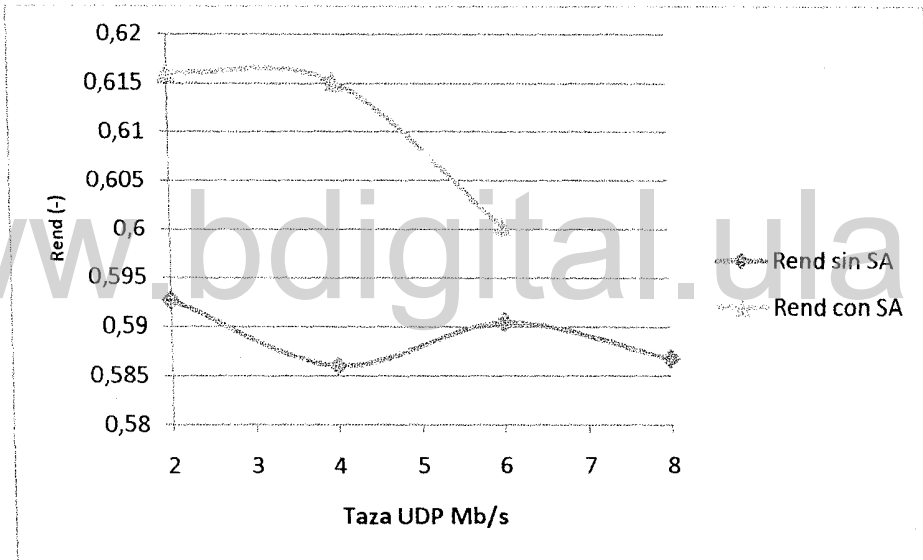


Figura 6-24 Rendimiento de UDP con de Diagrama de influencia tipo 2

• **Escenario de TCP de Diagrama de influencia tipo 1**

Error %	Rendimiento promedio % (sin sistema autonómico)	Caudal Promedio Normalizado (sin sistema autonómico)	Numero de cambios propuestos	Rendimiento promedio % (con sistema autonómico)	Caudal Promedio Normalizado (con sistema autonómico)	cambios propuestos (Tiempo ms /tipo de TCP)
0,00	65,817	0,449232	1	69,7786	0,61715	18,000317 TCP/Linux → TCP/Tahoe
0,05	65,6949	0,0539624	0			
0,10	61,4724	0,0234557	1	61,8316	0,0485131	45,002966

0,15	57,7861	0,0144771	1	57,1623	0,0256192	TCP/Linux -> TCP/Tahoe 45,019926
Tasa de incrementos (rendimiento)						TCP/Linux -> TCP/Tahoe
Tasa de incrementos (caudal)						2/3
						3/3

Tabla 6-13 Escenario de TCP de Diagrama de influencia tipo 1

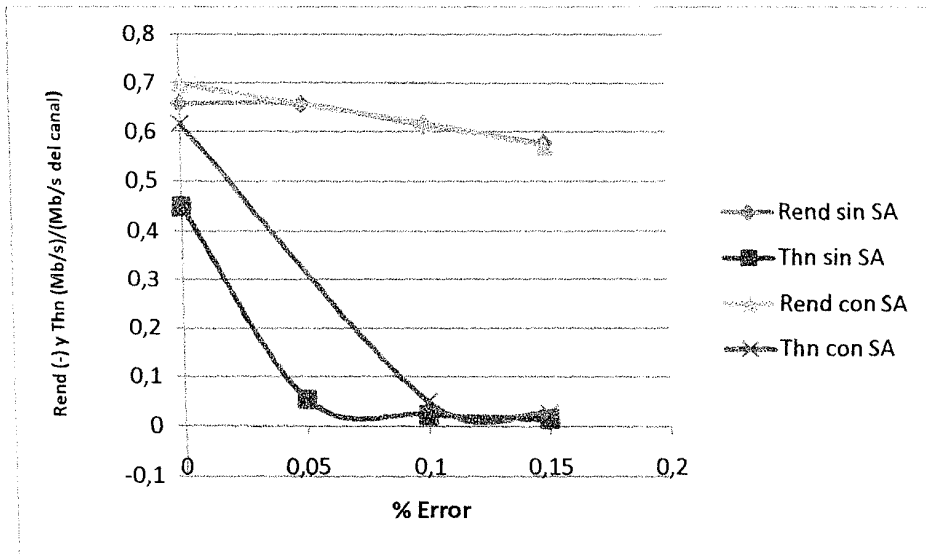


Figura 6-25 Escenario de TCP de Diagrama de influencia tipo 1

• **Escenario de TCP de Diagrama de influencia tipo 2**

Error %	Rendimiento promedio % (sin sistema autonómico)	Caudal Promedio Normalizado (sin sistema autonómico)	Numero de cambios propuestos	Rendimiento promedio % (con sistema autonómico)	Caudal Promedio Normalizado (con sistema autonómico)	cambios propuestos (Tiempo ms /tipo de TCP)
0,00	65,817	0,449232	1	69,7786	0,61715	18,000317 TCP/Linux -> TCP/Tahoe
0,05	65,6949	0,0539624	0			
0,10	61,4724	0,0234557	1	61,8316	0,0485131	45,002966 TCP/Linux -> TCP/Tahoe
0,15	57,7861	0,0144771	1	57,1623	0,0256192	45,019926 TCP/Linux -> TCP/Tahoe
Tasa de incrementos (rendimiento)						2/3
Tasa de incrementos (caudal)						3/3

Tabla 6-14 Escenario de TCP de Diagrama de influencia tipo 2

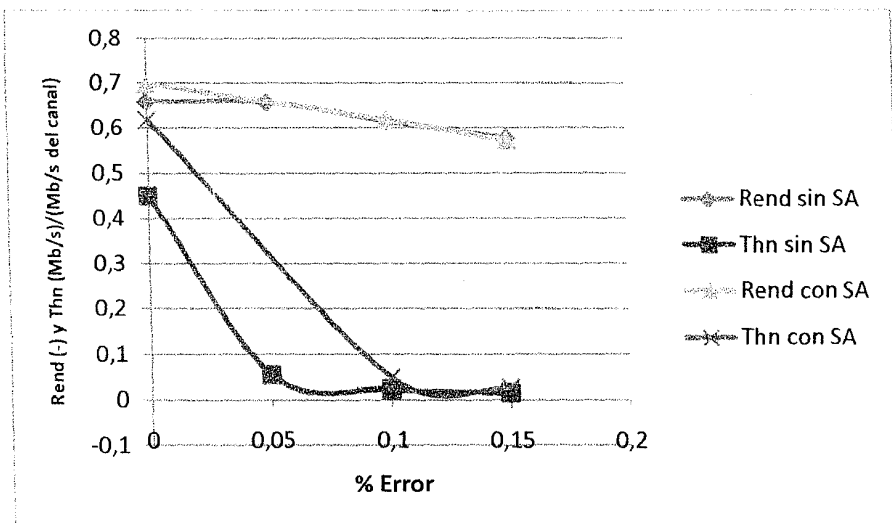


Figura 6-26 Escenario de TCP de Diagrama de influencia tipo 2

www.bdigital.ula.ve

- **Resultados Escenario de simulación de pérdidas con decisión es base a las pérdidas**

- **Escenario 1 Diagrama de influencia tipo 2**

Tasa de errores %	Rendimiento promedio % (sin sistema autónomico)	Caudal Promedio Normalizado (sin sistema autónomico)	Numero de cambios propuestos	Rendimiento promedio % (con sistema autónomico)	Caudal Promedio Normalizado (con sistema autónomico)	cambios propuestos (Tiempo ms /tipo de TCP)
0	61,9497	0,342438	0			
2	65,0132	0,112563	1	68,2821	0,217631	27,000613 TCP/Linux -> TCP/Tahoe
4	64,0111	0,0593216	1	66,0921	0,14723	15,000613 TCP/Linux -> TCP/Tahoe
6	62,7275	0,0454619	1	64,5466	0,104063	15,000613 TCP/Linux -> TCP/Tahoe
8	60,7534	0,0335296	1	63,0999	0,0735979	33,00168 TCP/Linux -> TCP/Tahoe
10	59,1134	0,0277957	1	61,3737	0,0542469	27,000613 TCP/Linux -> TCP/Tahoe
12	58,6851	0,0229077	1	59,9828	0,0511893	18,00008 TCP/Linux -> TCP/Tahoe
14	56,9253	0,0167509	1	57,7487	0,029904	39,000613 TCP/Linux -> TCP/Tahoe
16	53,2205	0,00969973	1	57,2679	0,0219237	54,00008 TCP/Linux -> TCP/Tahoe
Tasa de incrementos (rendimiento)						8/8
Tasa de incrementos (caudal)						8/8

Tabla 6-15 Escenario 1 Diagrama de influencia tipo 2 en base a las pérdidas

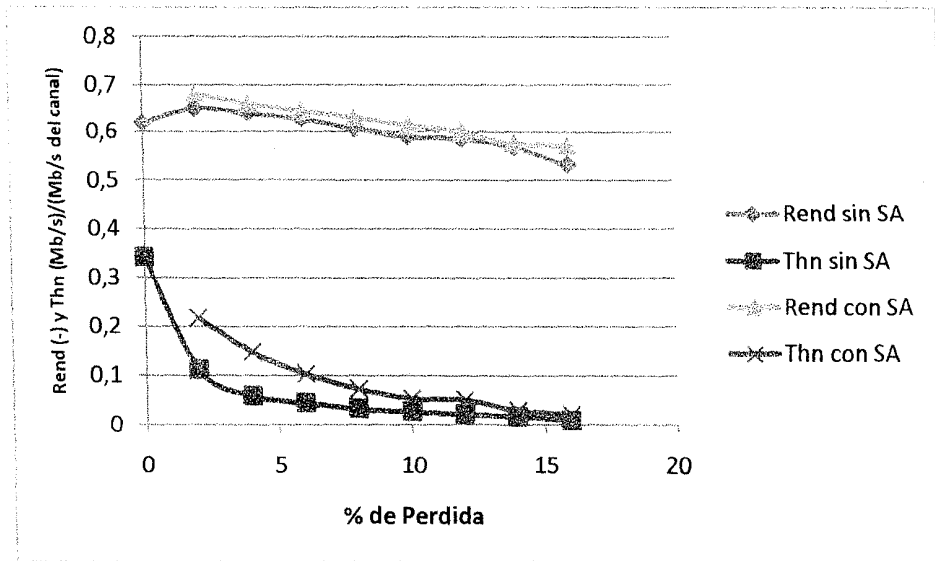


Figura 6-27 Escenario 1 Diagrama de influencia tipo 2 en base a las pérdidas

• **Escenario 2 Diagrama de influencia tipo 2**

Tasa de errores %	Rendimiento promedio % (sin sistema autónómico)	Caudal Promedio Normalizado (sin sistema autónómico)	Número de cambios propuestos	Rendimiento promedio % (con sistema autónómico)	Caudal Promedio Normalizado (con sistema autónómico)	cambios propuestos (Tiempo ms /tipo de TCP)
0	61,9497	0,342438	0			
5	63,5108	0,0517296	1	65,6532	0,120953	15,000613 TCP/Linux -> TCP/Tahoe
10	59,1134	0,0277957	1	61,3737	0,0542469	27,000613 TCP/Linux -> TCP/Tahoe
15	56,4405	0,0148373	0			
20	50,6641	0,009984	1	52,0231	0,015032	18,00008 TCP/Linux -> TCP/Tahoe
25	38,2617	0,00356373	2	43,1804	0,00637947	45,00168 TCP/Linux -> TCP/Sack1 57,00168 TCP/Sack1 -> TCP/Tahoe
30	32,0556	0,00115093	1	37,5587	0,00330773	18,00008 TCP/Linux -> TCP/Tahoe
35	24,5275	0,0002496	0			
40	18,8261	0,00067253	0			
45	1,32631	0,00017333	1	10,9184	0,000471733	63,000613 TCP/Linux -> TCP/Tahoe
Tasa de incrementos (rendimiento)						6/6
Tasa de incrementos (caudal)						6/6

Tabla 6-16 Escenario 2 Diagrama de influencia tipo 2 decisión en base a las pérdidas

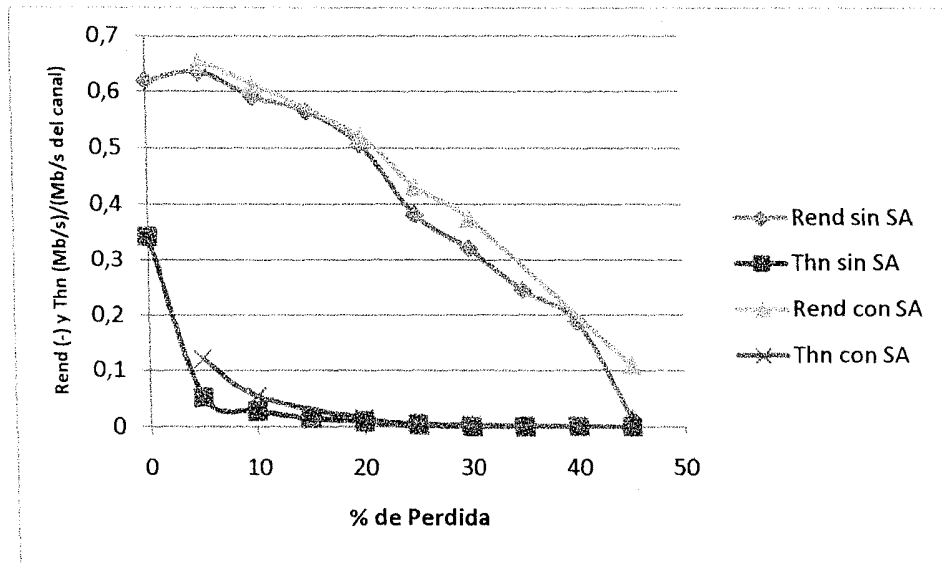


Figura 6-28 Escenario 2 Diagrama de influencia tipo 2 decisiones en base a las pérdidas

www.bdigital.ula.ve


```

print "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX \n";
print "err = $error \n";

my $BWMb=$BW . "Mb";
my $DLms=$DL . "ms";

my $cle = qx(echo XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX >> quality.out );
my $simp = qx(echo error rate = $error >> quality.out );
my $arrstp="";
my $aux=0;
our $simulacion="";
#####3
our $auxx=0;
our $simu=0;
for ($simu = 0; $simu <= 2; $simu++) {

$t[0]=0;
$p[0]=4;

if($auxx == 0){
if($simu == 0){
if($debu == 1){print "cero \n";}
$simulacion = qx(ns simulacion.tcl $error $QS $BWMb $DLms);
$simp = qx(echo $simu ns simulacion.tcl $error $QS $BWMb $DLms >> quality.out );
print "$simu I1 ns simulacion.tcl $error $QS $BWMb $DLms \n";
}else {
if($debu == 1){print "otro \n";}
$arrstp=$arrstp . " " . $t[$simu] . " " . $p[$simu];
$simulacion = qx(ns simulacion.tcl $error $QS $BWMb $DLms $arrstp);
$simp = qx(echo $simu ns simulacion.tcl $error $QS $BWMb $DLms $arrstp >> quality.out );
print "$simu I1 ns simulacion.tcl $error $QS $BWMb $DLms $arrstp \n";
if($debu == 1){print "otro \n";}
}
print "Simulation Output is: $simulacion\n";
my $predgra = qx(./pred-gra.sh $BW $DL);
if($simu == 0){my $simp2 = qx(perf tabla-nor.pl simulacion.tcl vp d-tcp-udp-v4.dne );}
$aux=$simu +1;
$t[$aux] = qx(./closesloop.sh | cut -d " " -f1 | tr -d " " | tr -d "\n");
$p[$aux] = qx(./closesloop.sh | cut -d " " -f2 | tr -d " " | tr -d "\n");
if(($t[$aux] eq "")){ $auxx=1; $t[$aux]=0;}
if(($p[$aux] eq "")){ $auxx=1; $p[$aux]=0;}

if($debu == 1){print "tact $t[$simu] tnext $t[$aux] pact $p[$simu] pnext $p[$aux] \n";}
if ($simu > 0){
if( $t[$aux] eq $t[$simu] ){ $auxx=1; if($debu == 1){print "c1 \n";}}
if( $t[$aux] == $t[$simu] ){ $auxx=1; if($debu == 1){print "c2 \n";}}
if( $t[$aux] < $t[$simu] ){ $auxx=1; if($debu == 1){print "c3 \n";}}
if( $p[$aux] eq $p[$simu] ){ $auxx=1; if($debu == 1){print "c4 \n";}}
if( $p[$aux] == $p[$simu] ){ $auxx=1; if($debu == 1){print "c5 \n";}}
if($debu == 1){print "auxx= $auxx \n";}
}
}
} # for simu
#####3
} # for error
#sed -e 's/\./g'
$out = qx(./coma.sh >> cquality.out);
exit(0);
}

```

6.2.2. Scripts de simulación en tcl

Nombre	simulacion.tcl
Lenguaje de programación	tcl
Dependencias (scrips)	proto-change-h.tcl
Entradas	% de error del enlace 2-3 Tamaño de la cola Ancho de banda del enlace en estudio Retardo del enlace en estudio Tiempo de cambio / Tipo de TCP Tiempo de cambio / Tipo de TCP .. .
Salidas	Por consola (a) out.tr out.nam
Propósito	Simulación ns2

```
////////////////////////////////////  
source proto-change-h.tcl
```

```
# primer argumento es % de error  
set err [lindex $argv 0]  
set qs [lindex $argv 1]  
set bw [lindex $argv 2]  
set del [lindex $argv 3]  
  
set scbr 3.75mb  
set iproto 6
```

```
#0 Agent/TCP  
#1 Agent/TCP/Reno  
#2 Agent/TCP/Newreno  
#3 Agent/TCP/Sack1  
#4 Agent/TCP/Vegas  
#5 Agent/TCP/Fack  
#6 Agent/TCP/Linux
```

```
#####  
global t_time  
global t_proto
```

```
if {$argc > 4} {  
set lv [expr "($argc - 4)/2"]  
set in 1  
set inn 4  
puts "Number of proto change= $lv\n"  
for {set u 0} {$u < $lv} {incr u} {  
set t_time($in) [lindex $argv $inn]  
set t_proto($in) [lindex $argv [expr "$inn +1"]]  
incr in  
set inn [expr "$inn+2"]  
}  
}
```

```
#####
```

```

#Create a simulator object
set ns [new Simulator]

##### sim param #####3
#Define different colors for data flows (for NAM)
$ns color 1 Blue
$ns color 2 Red

#Open the NAM trace file
set nf [open out.nam w]
$ns namtrace-all $nf

#Open the Trace files
#set file1 [open outfile.tr w]
#set winfile [open outWinFile w]

#Define a 'finish' procedure
proc finish {} {
    global ns nf tracef file1 winfile
    $ns flush-trace
    #Close the NAM trace file
    close $nf
    close $tracef
    #    close $file1
    #    close $winfile
    #Execute NAM on the trace file
    #    exec nam out.nam &
    exit 0
}

##### topology #####
#Create four nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]

#Create links between the nodes
$ns duplex-link $n0 $n2 100Mb 0ms DropTail
$ns duplex-link $n1 $n2 100Mb 0ms DropTail
$ns duplex-link $n2 $n3 $bw1 $del DropTail
# $ns duplex-link $n2 $n3 10Mb 1ms DropTail
$ns duplex-link $n3 $n4 100Mb 0ms DropTail
$ns duplex-link $n3 $n5 100Mb 0ms DropTail

set tracef [open out.tr w]
$ns trace-queue $n2 $n3 $tracef

#Set Queue Size of link (n2-n3) to 10
$ns queue-limit $n2 $n3 $qs
#$ns queue-limit $n2 $n3 10

#Give node position (for NAM)
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n2 $n3 orient right
$ns duplex-link-op $n3 $n4 orient right-up
$ns duplex-link-op $n3 $n5 orient right-down

#Monitor the queue for link (n2-n3). (for NAM)
$ns duplex-link-op $n2 $n3 queuePos 0.5

```

```

#error model
set loss_module [new ErrorModel]
$loss_module set rate_ $err
$loss_module ranvar [new RandomVariable/Uniform]
$loss_module drop-target [new Agent/Null]
$ns lossmodel $loss_module $n2 $n3

##### Def Conn #####
#####
#Setup a UDP connection
set udp [new Agent/UDP]
$ns attach-agent $n1 $udp
set null [new Agent/Null]
$ns attach-agent $n3 $null
$ns connect $udp $null
$udp set fid_ 2
#Setup a CBR over UDP connection
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
$cbr set type_ CBR
$cbr set packet_size_ 1000
$cbr set rate_ $scbr
$cbr set random_ false
#####

#\\
ftp_traffic 0 $n0 $n4 $proto 1
#ftp_traffic 0 $n0 $n4 4 1
#ftptraffic conx-number $n0 $n4 proto (1 star or 0 stop)
#\\

#Schedule events for the CBR and FTP agents
$ns at 0.1 "$ftp_conn(0) start"
$ns at 10.0 "$cbr start"

if {$argc > 4} {
  set ib 1
  for {set a 0} {$a < $lv} {incr a} {
    puts "time $ib = $t_time($ib)"
    puts "proto $ib = $t_proto($ib)"
    ftp_traffic $ib $n0 $n4 $t_proto($ib) 1
    incr ib
  }
}

if {$argc > 4} {
  set ib 1
  for {set a 0} {$a < $lv} {incr a} {
    if {$a == 0} {
      $ns at $t_time($ib) "$ftp_conn(0) stop ; ftp_traffic 0 $n0 $n4 0 0 "
      $ns at $t_time($ib) "$ftp_conn($ib) start "
    } else {
      set id [expr "$ib-1"]
      $ns at $t_time($ib) "$ftp_conn($id) stop ; ftp_traffic $id $n0 $n4 $t_proto($id) 0 "
      $ns at $t_time($ib) "$ftp_conn($ib) start "
    }
    incr ib
  }
}

$ns at 120.0 "finish"
#Run the simulation
$ns run
\\

```

6.2.3. Scripts para cambio de tipo de conexión

Nombre	proto-change-h.tcl
Lenguaje de programación	tcl
Dependencias (scrips)	
Entradas	Numero de conexión Tiempo de cambio / Tipo de TCP Arranque / parada
Salidas	
Propósito	Simulación cambio de conexión

```

proto-change-h.tcl
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
proc ftp_traffic { nca node0 node1 tproto be } {
global ns
global ftp_conn
global sink
global tcp_conn

set i $nca

if { $be == 1 } {
#puts "conex"
if { $tproto == 0 } {set tcp_conn($i) [new Agent/TCP] ;puts "TCP"}
if { $tproto == 1 } {set tcp_conn($i) [new Agent/TCP/Reno] ;puts "Reno" }
if { $tproto == 2 } {set tcp_conn($i) [new Agent/TCP/Newreno] ;puts "NewReno"}
if { $tproto == 3 } {set tcp_conn($i) [new Agent/TCP/Sack1] ;puts "Sack1"}
if { $tproto == 4 } {set tcp_conn($i) [new Agent/TCP/Vegas] ;puts "Vegas"}
if { $tproto == 5 } {set tcp_conn($i) [new Agent/TCP/Fack] ;puts "Fack"}
if { $tproto == 6 } {set tcp_conn($i) [new Agent/TCP/Linux] ;puts "Linux"}

#Setup a TCP connection
$tcp_conn($i) set class_2
$ns attach-agent $node0 $tcp_conn($i)

set sink($i) [new Agent/TCPSink]
$ns attach-agent $node1 $sink($i)
$ns connect $tcp_conn($i) $sink($i)
$tcp_conn($i) set fid_1

#Setup a FTP over TCP connection
set ftp_conn($i) [new Application/FTP]
$ftp_conn($i) attach-agent $tcp_conn($i)
$ftp_conn($i) set type_FTP
}; # bebegin

if { $be == 0 } {
#puts "sin conex"
$ns detach-agent $node0 $tcp_conn($i)
$ns detach-agent $node1 $sink($i)
}; # beend

}; # end proc ftp_traffic

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```

6.2.4. Scripts para cálculo de parámetros de simulación

Nombre	analisis-tr.pl
Lenguaje de programación	Perl
Dependencias (scritps)	ninguno
Entradas	Archivo de traza (out.tr) Nodo origen Nodo destino Granularidad Ancho de banda del enlace en estudio Retardo del enlace en estudio
Salidas	Por consola (a) tiempo duración-del-paquete Caudal-normalizado Tasa-de-pérdidas Rendimiento% Caudal
Propósito	Calculo de Resultados de la simulacion

```

//////////////////////////////////// cat analisis-tr.pl////////////////////////////////////
# type: perl all-fr-protosd.pl <trace file> <src node> <dst node> <granularity> <BW> <delay> > output file
# perl all-fr-protosd.pl out.tr 0.X 4.Y 1 12 10 (el sistema lee 0.1 0.2 0.3 etc y 4.0 4.1 4.2 etc)
# + 23.255333 2 3 cbr 1000 ----- 2 1.0 3.1 214 5193

```

```

use strict;
use warnings;

```

```
#####
```

```

my $end=0;
my @endtime=0;
my $loss=0;
my $rend=0;
my $start=0;
my @starttime=0;
my $thn=0;
my $throughput=0;
my @x=0;
my $x8=0;
my $x9=0;
#####

```

```

my $infile=$ARGV[0];
my $fonode=substr($ARGV[1],0,1);
my $tonode=substr($ARGV[2],0,1);
my $granularity=$ARGV[3];
my $lbw=$ARGV[4]; # 20.5 (Mb)
my $linkd=$ARGV[5]; # 6 (ms)
my $ndelay=$linkd/1000+1040/($lbw*1000000/8);

```

```

#by granularity parameter in seconds
my $sum=0;
my $send=0;
my $drop=0;
my $clock=0;
my $highestpacketid=0;
my $starttime=0;
my $packetduration=0;
my $pdid=0;

```

```

open (DATA,"<$infile")
|| die "Can't open $infile $!";

```

```

while (<DATA>) {
    @x = split(' ');

#my $str = substr($x[8],0,1);
#print STDOUT "OK $str\n";
$x8 = substr($x[8],0,1);
$x9 = substr($x[9],0,1);

#print "fonode $fonode $x8 tonode $tonode $x9 \n";

#####
if ($x[0] eq '+' && $x8 eq $fonode && $x9 eq $tonode ){starttime[$x[11]] = $x[1];}
if ($x[0] eq 'r' && $x8 eq $fonode && $x9 eq $tonode ) {$endtime[$x[11]] = $x[1];}
if ($x[0] eq 'd' && $x8 eq $fonode && $x9 eq $tonode ) { $starttime[$x[11]] = -1;}
#####

if ($x[1]-$clock <= $granularity)
{ ##if1

if ($x[0] eq 'r' && $x8 eq $fonode && $x9 eq $tonode )
{
    $sum=$sum+$x[5];
    $pdid=$x[11];
}

if ($x[0] eq '+' && $x8 eq $fonode && $x9 eq $tonode )
{
    $send=$send+1;
    #print STDOUT "XX send $send\n";
}

if ($x[0] eq 'd' && $x8 eq $fonode && $x9 eq $tonode )
{
    $drop=$drop+1;
    #print STDOUT "XX drop $drop\n";
}

if ($drop > 0 && $send > 0){if ($send > $drop){$loss=$drop/$send;}else{$loss=1}
}
else {$loss=0;}

} ##if1

else
{ ## else 1
###
$start = $starttime[$pdid];
$end = $endtime[$pdid];
if ( $start ne '-1' && $pdid > 0 ){
    $packetduration = ((-1)/(2*$ndelay))*(($end - $start)-$ndelay)+1;
    if ($packetduration > 1 ){ $packetduration = 1}
}

if ($drop > 0 && $send > 0){if ($send > $drop){$loss=$drop/$send;}else{$loss=1}
}
else {$loss=0;}

    $throughput=0.000008*$sum/$granularity;
    $thn=0.000008*$sum/($granularity*$lbw);

    $rend=(sqrt(($thn*$thn + $packetduration*$packetduration )/2)*(1 - $loss))*100;
    if ($rend > 100) {$rend = 100}
}
}

```

```

print STDOUT "$x[1] $packetduration $thn $loss $rend $throughput\n";
$clock=$clock+$granularity;
$sum=0;
###
} ## else1
}
# print last

###
$start = $starttime[$pdid];
$end = $endtime[$pdid];
if ( $start ne '-1' && $pdid > 0 ){
$packetduration = ((-1)/(2*$ndelay))*(($end - $start)-$ndelay)+1;
if ($packetduration > 1 ){ $packetduration = 1}
}

$throughput=0.000008*$sum/$granularity;
$thn=0.000008*$sum/($granularity*$lbw);

$rend=(sqrt(($thn*$thn + $packetduration*$packetduration )/2)*(1 - $loss ))*100;
if ($rend > 100) {$rend = 100}

if ($drop > 0 && $send > 0){if ($send > $drop){$loss=$drop/$send;}else{$loss=1}
}
else {$loss=0;}

print STDOUT "$x[1] $packetduration $thn $loss $rend $throughput\n";
$clock=$clock+$granularity;
$sum=0;
###

```

```

close DATA;
exit(0);
////////////////////////////////////

```

www.bdigital.ula.ve

6.2.5. Scripts de desglose de los resultados

Nombre	pred-gra.sh	
Lenguaje de programación	Bash	
Dependencias (scrips)	analisis-tr.pl avg.awk tablav3.pl	
Entradas	Ancho de banda del enlace en estudio Retardo del enlace en estudio	
Salidas	gdelay	Tiempo duración-del-paquete
	gth	Tiempo Caudal-normalizado
	gloss	Tiempo Tasa-de-pérdidas
	grend	Tiempo Rendimiento%
	gth-no-not	Tiempo Caudal
	quality.out	Rendimiento Caudal
	vp	perl tablav3.pl gsal
Propósito	Desglose de los resultados de la simulación y graficar los mismos	

```

//////////////////////////////////// pred-gra.sh //////////////////////////////////////
#!/bin/bash!

```

```

ARGV1=$1 # not used do NOT delete
ARGV2=$2 # BW
ARGV3=$3 # delay
#Tiempo thn packetduration loss rend

```

```

if [ -f a ] ; then rm a; fi
if [ -f gth ] ; then rm gth; fi
if [ -f ga ] ; then rm ga; fi
if [ -f gt ] ; then rm gt; fi
if [ -f gdelay ] ; then rm gdelay; fi
if [ -f gth ] ; then rm gth; fi
if [ -f gloss ] ; then rm gloss; fi
if [ -f grend ] ; then rm grend; fi
if [ -f glearn ] ; then rm glearn; fi
if [ -f gsal ] ; then rm gsal; fi
if [ -f vp ] ; then rm vp; fi
if [ -f gth-no-not ] ; then rm gth-no-not; fi
if [ -f Data.cas ] ; then rm Data.cas; fi
if [ -f Data0.cas ] ; then rm Data0.cas; fi

```

```

perl analisis-tr.pl out.tr 0.0 4.0 1 $1 $2 >a
cat a | cut -d " " -f 1,2 > gdelay
cat a | cut -d " " -f 1,3 > gth
cat a | cut -d " " -f 1,4 > gloss
cat a | cut -d " " -f 1,5 > grend
cat a | cut -d " " -f 1,6 > gth-no-not
cat a | cut -d " " -f 2,3,4,5 >> Data0.cas

```

```

cat -n Data0.cas > Data1.cas
echo "IDnum Delay Th Loss Rend" > Data.cas
cat Data1.cas >> Data.cas
if [ -f Data0.cas ]; then rm Data0.cas; fi
if [ -f Data1.cas ]; then rm Data1.cas; fi

cat a | cut -d " " -f 2,3,4,5 >> DataF.cas

if [ "$ARGV3" = "-g" ]; then
xgraph gth & xgraph gdelay & xgraph gloss & xgraph grend &
fi

cat grend | cut -d " " -f2 > ga
cat gth | cut -d " " -f2 > gt

cat a | cut -d " " -f 1,2,3,4 > glearn
cat a | cut -d " " -f 2,3,4,5 >> gsal

ren=`awk -f avg.awk ga`
#echo rend $ren >> q
th=`awk -f avg.awk gt`
#echo thput $th >> q
echo rend $ren thput $th >> quality.out

cat a | cut -d " " -f 2,3,4,5 > gsal
#perl tablav2.pl gsal | grep vp >> vp
perl tablav3.pl gsal | grep vp >> vp
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```

www.bdigital.ula.ve

6.2.6. Scripts Categorización de los resultados de salida

Nombre	Tablav3.pl
Lenguaje de programación	perl
Dependencias (scripts)	
Entradas	(gsal) duración-del-paquete Caudal-normalizado Tasa-de-pérdidas Rendimiento%
Salidas	Consola
Propósito	Categorización de los resultados de salida

```

////////////////////////////////////
# type: perl tabla.pl <trace file> > output file
#perl tablav3.pl gsal
$infile=$ARGV[0];

#by granularity parameter in seconds
$sum=1;

    open (DATA,"<$infile")
    || die "Can't open $infile $!";

    while (<DATA>) {
        @x = split(' ');
        #####
        if ($x[0] != 0){
            for ($er = 0; $er <= 3 ; $er=$er + 1) {

                if ($er != 3){
                    if (($x[$er] >= 0) && ($x[$er] <= 0.25) ){${s[$er]}[$sum]="0 to 0.25";}
                    if (($x[$er] > 0.25) && ($x[$er] <= 0.5) ){${s[$er]}[$sum]="0.25 to 0.5";}
                    if (($x[$er] > 0.5) && ($x[$er] <= 0.75) ){${s[$er]}[$sum]="0.5 to 0.75";}
                    if (($x[$er] > 0.75) ){${s[$er]}[$sum]="0.75 to 1"; }
                } else {
                    if (($x[$er] >= 0) && ($x[$er] <= 25) ){${s[$er]}[$sum]="0 to 25";}
                    if (($x[$er] > 25) && ($x[$er] <= 50) ){${s[$er]}[$sum]="25 to 50";}
                    if (($x[$er] > 50) && ($x[$er] <= 75) ){${s[$er]}[$sum]="50 to 75";}
                    if (($x[$er] > 75) ){${s[$er]}[$sum]="75 to 100"; }
                    ${s[4]}[$sum]=$x[$er];
                }
            }
            $sum++;
        } #endif
    }

    for ($eq = 1; $eq <= ($sum - 1) ; $eq=$eq + 1) {
        for ($er = 0; $er <= 3 ; $er=$er + 1) {
            $arrstp[$eq]=$arrstp[$eq] . " " . ${s[$er]}[$eq];
        }
    }

    for ($eq = 1; $eq <= ($sum - 1) ; $eq=$eq + 1) {
        for ($eqq = ($eq + 1); $eqq <= ($sum - 1) ; $eqq=$eqq + 1) {
            if (($arrstp[$eq] eq $arrstp[$eqq] ) && ($arrstp[$eqq] ne "" )){
                $acm[$eq]= $acm[$eq] + ${s[4]}[$eq]; $acc[$eq]=$acc[$eq]+1;
            }
        }
    }

```


6.2.8. Scripts de cambio de parámetros

Nombre	change.pl
Lenguaje de programación	Perl.sh
Dependencias (scrips)	
Entradas	(grend) Tiempo Rendimiento%
Salidas	Consola Tiempo-de-cambio tendencia Cambio
Propósito	Calcular el tiempo en el cual se debe tomar la decisión de cambio

```

////////////////////////////////////
# type: perl change.pl <trace file> > output file

```

```
$infile=$ARGV[0];
```

```
#by granularity parameter in seconds
```

```

my $sum=0;
my $lines = 0;
my $av= 0;
my $il = 0;
my $dd = 0;
my $ddc = 0;
my $dm = 0;
my $dmc = 0;
my @vetc =0 ;
my @veta =0 ;
my $veta[0]=0;

```

```

##### wc -l
open(DATA, $infile) or die "Can't open `infile': $!";
while (<DATA>) {
    $lines = $lines + 1;
    @x = split(' ');
    $sum=$sum+$x[1];
}
close FILE;
$av=$sum/$lines;
##### wc -l

```

```

open (DATA,"$infile")
|| die "Can't open $infile $!";
while (<DATA>) {
    @x = split(' ');
#####

```

```

#####
my $comp=3;
if ($dd < $comp){
    #print "node dd= $dd\n";
    $dd=$dd+1;
    $su=$su+$x[1];
    $vetc[$dd]=$x[1];
    if ($dd == $comp){
        $ag1=$su/$comp;
        $dd=0;
        $su=0;
        $ddc=$ddc+1;
        $m1=$vetc[2]-$vetc[1];
        $m2=$vetc[3]-$vetc[2];

```


6.2.9. Código del programa de toma de decisión

Nombre	my-dec-rend
Lenguaje de programación	C con librerías de Netica
Dependencias (scrips)	
Entradas	Diagrama de influencia tiempo de cambio rendimiento
Salidas	Tiempo-de-cambio tipo-de-TCP
Propósito	Usar el diagrama de influencia para la toma de decisión

```

/////////////////////////////////////////////////////////////////
#include <stdio.h>
#include <stdlib.h>
#include "Netica.h"
#include "NeticaEx.h"

#define CHKERR {if (GetError_ns (env, ERROR_ERR, NULL)) goto error;}

environ_ns* env;

void fun (node_bn *dd, float ties)
{
  int fss, indm;
  const util_bn* utilss;
  float maxx;

  utilss = GetNodeExpectedUtils_bn (dd);
  for (fss = 0; fss < GetNodeNumberStates_bn (dd); ++fss){
    if(fss==0){
      maxx=utilss[fss];
      indm=fss;
    }
    else{
      if(utilss[fss] > maxx ){maxx=utilss[fss]; indm=fss; }
    }
  }
  printf("XXX %lf %s = %lf\n",ties,GetNodeStateName_bn (dd,indm), maxx);
} //end void

int main (int argc, char *argv[]){
  net_bn* net = NULL;
  node_bn *delay, *th, *loss, *rend, *d;
  state_bn fs, decision;
  const util_bn* utils;
  char mesg[MESG_LEN_ns];
  int res;
  report_ns* err;
  double a, b;
  int ww;
  char filed;

  env = NewNeticaEnviron_ns (NULL, NULL, NULL);
  res = InitNetica2_bn (env, mesg);
  printf ("%s\n", mesg);
  if (res < 0) exit (-1);

```

```

filed=argv[1];
a=atof(argv[2]);
b=atof(argv[3]);

net = ReadNet_bn ( NewFileStream_ns (argv[1], env, NULL, NO_VISUAL_INFO);

    delay = NodeNamed_bn ("Delay", net);
    th     = NodeNamed_bn ("Th", net);
    loss  = NodeNamed_bn ("Loss", net);
    rend  = NodeNamed_bn ("Rend", net);
    d     = NodeNamed_bn ("D", net);

CompileNet_bn (net);

utils = GetNodeExpectedUtils_bn (d); // returns expected utilities, given current findings
CHKERR

////////////////////////////////////
EnterNodeValue_bn (rend,b);
fun (d,b);
    CHKERR
////////////////////////////////////

RetractNetFindings_bn (net);
GetNodeExpectedUtils_bn (d);

end:
DeleteNet_bn (net);
res= CloseNetica_bn (env, mesg);
printf ("%s\n", mesg);
return (res < 0 ? -1 : 0);

error:
err = GetError_ns (env, ERROR_ERR, NULL);
fprintf (stderr, "MakeDecision: Error %d %s\n",
    ErrorNumber_ns (err), ErrorMessage_ns (err));
goto end;
}

```

```

////////////////////////////////////

```

6.2.10. Script de reforzamiento de la tabla de utilidades

Nombre	tabla-nor.pl
Lenguaje de programación	Perl
Dependencias (scripts)	
Entradas	script de simulación archivo de categorización (vp) Diagrama de influencia
Salidas	Diagrama de influencia con la tabla de función de utilidad
Propósito	Reforzar la tabla de utilidades

```

////////////////////////////////////
# type: perl tabla-nor.pl <simulation-file> <in-file> <output-file>
# use strict;
# use warnings;
my $simfile=$ARGV[0];
my $infile=$ARGV[1];
my $outfile=$ARGV[2];
my $psim=$ARGV[3];
#by granularity parameter in seconds
my $sum=1;
our $proto="x";
my @x=0;
if($psim == 0){$proto=tcp;}
if($psim == 1){$proto=Reno;}
if($psim == 2){$proto=NewReno;}
if($psim == 3){$proto=Sack1;}
if($psim == 4){$proto=Vegas;}
if($psim == 5){$proto=Fack;}
if($psim == 6){$proto=Linux;}
print "PROTO Output: $psim $proto\n";
    open (DATA,"<$infile")
        || die "Can't open $infile $!";

    while (<DATA> ) {
        @x = split(' ');

my $mul=substr($x[12],3,6)*1;
my $sal1="$x[0] $x[1] $x[2] $x[3] $x[4] $x[5] $x[6] $x[7] $x[8] $x[9] $x[10] $x[11]";
my $sal2="$mul, // $proto $sal1";
print "$sal2\n";
#####
my $ion2 = qx(sed -i "s%\@undef, // $proto $sal1%$sal2%g" $outfile);

my $actv = qx(cat $outfile | grep -w "$proto $sal1" | cut -d "," -f1 | tr -d "\t" | tr -d " " | tr -d "\n");
#}
if($actv < $mul){
my $sal1="$x[0] $x[1] $x[2] $x[3] $x[4] $x[5] $x[6] $x[7] $x[8] $x[9] $x[10] $x[11]";
my $sal2="$mul, // $proto $sal1";
print "act $actv, // $proto $sal1\n";
print "new $sal2\n";
my $ion2 = qx(sed -i "s%$actv, // $proto $sal1%$sal2%g" $outfile);
}
#####
} # while
    close DATA;
    exit(0);
////////////////////////////////////

```

6.3. Anexo de Simulaciones y Resultados experimentales

En la ecuación 23 se puede observar la ecuación utilizada para calcular el Rendimiento:

$$\text{Rendimiento} = \sqrt{\frac{\text{Retardo}R^2 + \text{Caudal}N^2}{2}} (1 - T_{\text{perdidas}}) \quad (23)$$

Para el escenario 1 de simulación de pérdidas con el diagrama de influencia tipo 2 para una pérdida de 16% (Tabla 6-2), para esta simulación el cambio de TCP/Linux a TCP/Tahoe se realizo a los 42.00008 ms

	Sin SA	Con SA	Mejora con respecto al sistema sin SA
Retardo promedio	0,95978697	0,98163274	Desfavorable
Caudal Promedio	0,00969973	0,02498133	Favorable
% de perdidas Promedio	0,21037776	0,17789105	Favorable
Rendimiento promedio	53,220531	57,0933793	Favorable

Tabla 6-18 Parámetros de salida de la simulación de la Tabla 6-2 para un 16% de perdidas

En la Tabla 6-18 se puede apreciar que cuando se compara el sistema con y sin sistema autónomo, todos los parámetros mejoran con excepción del retardo el cual se incrementa de forma desfavorable, también es importante resaltar que por ejemplo la tasa de perdidas nominal de la simulación de del 16% y que para el caso sin SA la tasa promedio es de 21% y con SA es de 17,7%, en particular la tasa de perdidas es mayor debido a la cola en el nodo 2 que produce pérdidas debido a la conexión interferente. En las figuras de la Figura 6-29 a la Figura 6-32 se pueden apreciar el retardo, el caudal, la tasa de perdidas y por último el rendimiento, lado a lado con y sin SA, en línea general se puede observar durante algunos momento el sistema sin SA es mejor que con SA, pero al final un promedio calculado para cada serie arroja que los resultados observados en la Tabla 6-18

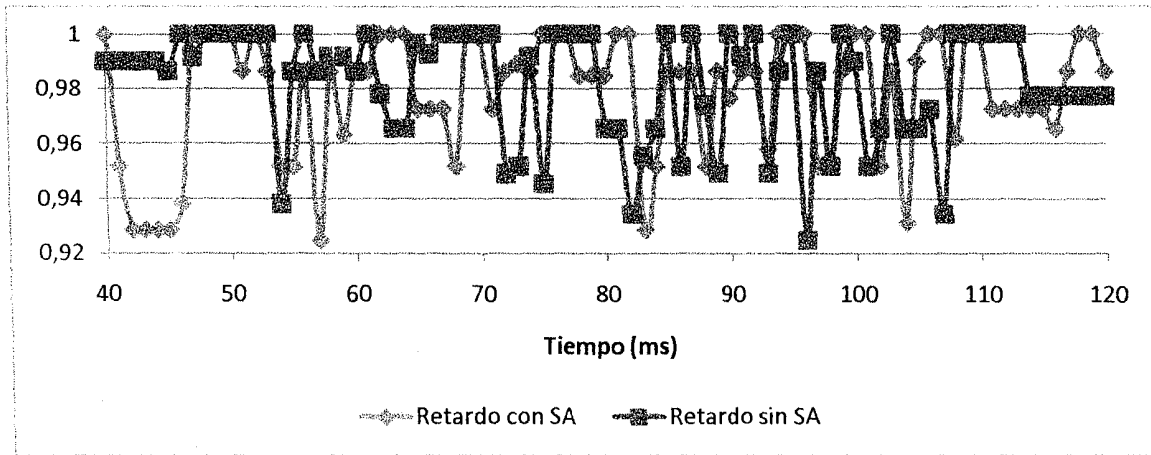


Figura 6-29 Retardo con SA y sin SA simulación de la Tabla 6-2 para un 16% de pérdidas

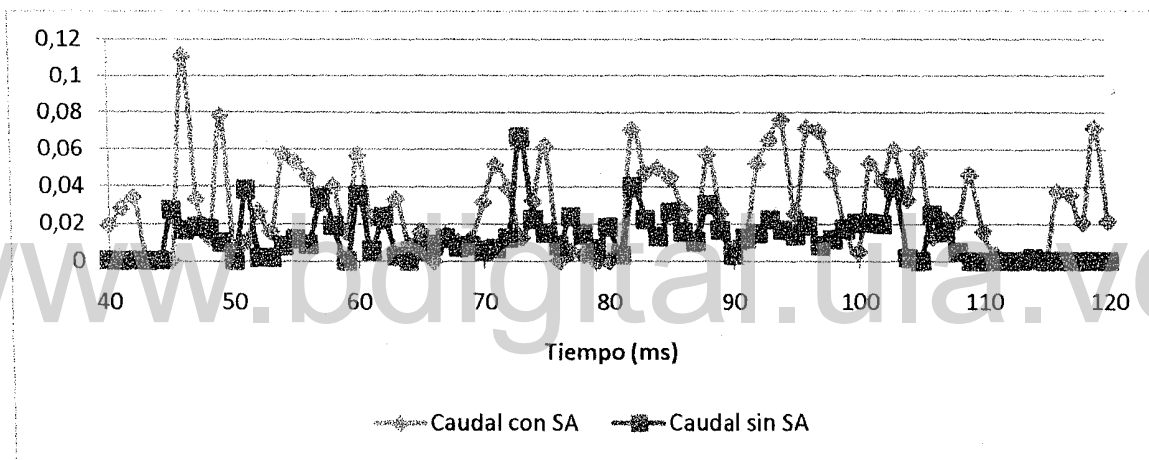


Figura 6-30 Caudal con SA y sin SA simulación de la Tabla 6-2 para un 16% de pérdidas

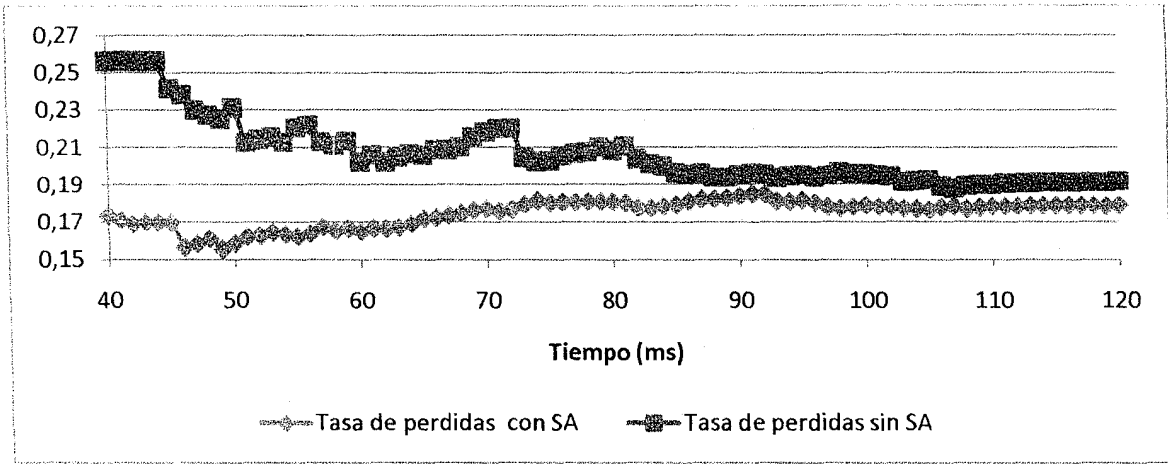


Figura 6-31 tasa de perdidas con SA y sin SA simulación de la Tabla 6-2 para un 16% de perdidas

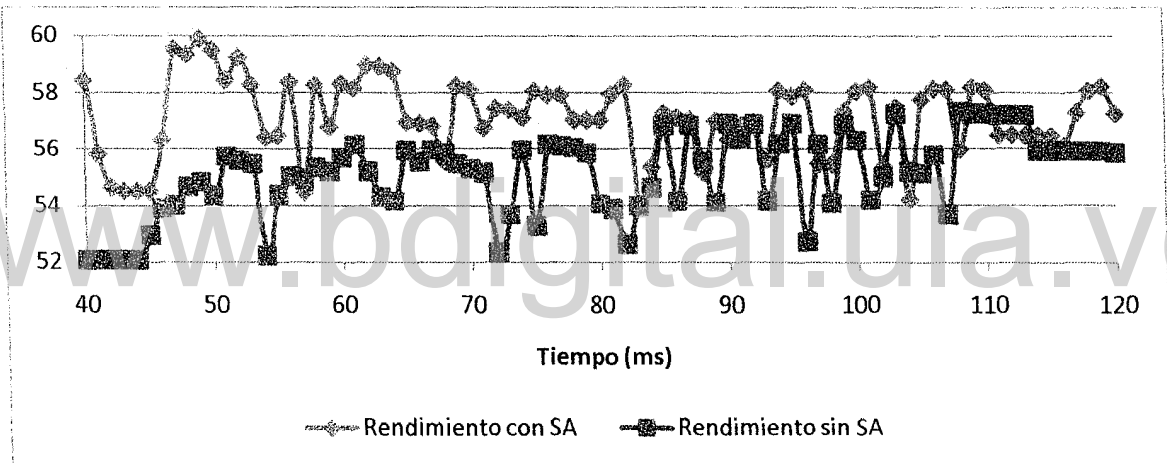


Figura 6-32 Rendimiento con SA y sin SA simulación de la Tabla 6-2 para un 16% de perdidas