

ESTIMACIÓN DEL CONSUMO DE ANCHO DE BANDA EN UN ENLACE PARA SERVICIOS EN TIEMPO REAL POR MEDIO DE MÉTODOS DE CLUSTERING DIFUSO.

Diego José Botia Valderrama, M.Sc¹, Member, IEEE.

Javier Fernando Botia Valderrama, M.Sc(c)², Member, IEEE {dbotia,javier.botia@udea.edu.co}

¹Grupo de Investigación en Telecomunicaciones Aplicadas-GITA. Universidad de Antioquia-UdeA

²Grupo de Investigación en Electrónica de Potencia, Automatización y Robótica-GEPAR.
Universidad de Antioquia-UdeA

Resumen: La información sobre el consumo de ancho de banda de un enlace de red es requerida para determinar su utilización y su disponibilidad. Diferentes aplicaciones de red pueden mejorar su desempeño con el conocimiento del Ancho de Banda Disponible (BWA) en la red, el cual se define como el menor BWA de todos los enlaces involucrados en una comunicación. En este artículo, se presenta la implementación de tres métodos de Clustering Difuso como SOM-Kmeans, FCM, y LAMDA para reducir la dimensión de los datos, y así obtener las clases y centros para ingresarlos como datos de entrenamiento a una red neuronal MLP. El resultado final muestra un excelente reconocimiento de patrones aplicados a dos casos de estudios, tráfico Periódico y Poisson, adquiridas por medio de la herramienta MGEN (Multigenerator). La congestión del enlace se simuló con tráfico cruzado del 30% y 60% de su capacidad.

Palabras Clave: Estimación de Ancho de Banda, Clustering Difuso, MLP, MGEN, tráfico cruzado.

Abstract: The information on the consumption of bandwidth in a network link is required to determine their use and availability. Different network applications can improve their performance with knowledge of the bandwidth available (BWA) in the network, which it is defined as the smallest BWA of all links involved in a communication. This paper shows the implementation of three Fuzzy Clustering methods such as SOM-Kmeans, FCM, and LAMDA to reduce the size of the data, and so to get the classes and centroids to enter them as training data to a neural network MLP. The final result shows a excellent pattern recognition applied to two case studies, Periodic and Poisson traffic, obtained them by means of the MGEN tool (Multigenerator). The link congestion was simulated with cross traffic by 30% and 60% of its capacity.

Index Terms: Bandwidth Estimation, Fuzzy Clustering, MLP, MGEN, cross traffic.

1. INTRODUCCIÓN

Recientemente, la estimación del ancho de banda disponible ha recibido considerable atención, debido a su uso potencial en diferentes aplicaciones de red (Guerrero y Labrador, 2006). Por ejemplo, las herramientas de administración de redes pueden determinar la utilización de los enlaces o monitorear y verificar niveles de Calidad de Servicio (QoS). Una caída atípica del ancho de banda disponible en la red puede asociarse a un ataque de negación de servicio tras ser analizada por mecanismos de seguridad en la red o por errores propios de la red. Estos inconvenientes conllevan a una congestión del enlace, pérdidas de paquetes, retardos indeseados, entre otros problemas. Por consiguiente, conducen a factores críticos en aplicaciones de audio y video. El Ancho de Banda Disponible (BWA) entre dos computadores en red, se define como el mínimo de todos los anchos de banda disponibles en los diferentes enlaces que conforman el camino de comunicación (Jain y Dovrolis, 2003). Para cada enlace, este valor se considera como la diferencia promedio entre la capacidad y la cantidad de tráfico que atraviesa el enlace en un periodo T . Este tráfico se conoce como Tráfico Cruzado. Multi Router Traffic Grapher (MRTG) es una herramienta de software libre que permite visualizar de manera gráfica la cantidad de tráfico cruzado en un enlace (Oetiker y Rand, 1998). Este software utiliza el protocolo SNMP (Simple Network Management Protocol) para leer el tráfico de los enrutadores y presentarlo en graficas embebidas sobre las páginas web que se consultan desde cualquier navegador. La visualización del tráfico se hace en periodos de cinco minutos, lo cual presenta una granularidad no aceptada para herramientas que requieran información de utilización del orden de segundos o incluso microsegundos. Un ejemplo de este caso es un protocolo de enrutamiento basado en métricas de congestión en la red. Una mayor granularidad se obtiene a través del uso de Sniffers como tcpdump (Jabcoson et al., 1998). Estas herramientas muestran la información de encabezado de los paquetes adquiridos a través de una tarjeta de red. Sin embargo, tcpdump no realiza ningún cálculo sobre cantidad de

tráfico cruzado en un periodo determinado. Debido a los problemas de enrutamiento y estabilidad del ancho de banda, una alternativa de solución son los métodos que ofrece la Inteligencia Computacional. En este artículo, se seleccionaron los métodos de Clustering Difuso para agrupar los datos en grupos o clases, y la red neuronal MLP (MultiLayer Perceptron) para tareas de reconocimiento de patrones. Utilizando tres métodos de Clustering Difuso, *K-Means* mejorado con el algoritmo de los *Mapas Auto organizados (SOM)*, el método *Fusca C-Means (FCM)*, y el método *LAMDA (Liaoning Algorithm Multivariable and Data Analysis)*, se clasificaron los datos utilizando las bases de datos de *testbed* del ancho de banda. Con la mejor clasificación obtenida, la matriz de los centros de las clases se ingresa como datos de entrenamiento a la red *MLP*. La adquisición de los datos históricos se obtiene por medio del tráfico generado por *MGEN* (Adamson y Gallavan, 1997) de tipo *Periódico y Poisson*.

El resto del artículo está organizado de la siguiente manera. En la sección II, se presentan los conceptos relevantes a la estimación de ancho de banda y métodos de Clustering Difuso. La herramienta *testbw* y sus modos de ejecución se presentan en la sección III. La sección IV se explica la adquisición de los datos usando la herramienta *MGEN*, el diseño del *testbed* para evaluar el desempeño de la estimación de ancho de banda y los resultados de la clasificación y reconocimiento de patrones del ancho de banda. Finalmente, la sección V se presenta las conclusiones y trabajos futuros.

2. MARCO CONCEPTUAL

A. Encapsulamiento TCP/IP

La arquitectura de Internet ubica los protocolos de comunicaciones en las capas de aplicación, transporte, red y de enlace. Como se observa en la Fig. 1, se muestra el proceso de encapsulamiento, que consiste en adicionar información correspondiente a la funcionalidad de cada capa. Esto implica que la información que originalmente se deseaba transmitir, va

a ser incrementada en 54 Bytes para el caso de paquetes TCP (*Protocolo de Control de Transporte*); estos Bytes se dividen en: 20 Bytes de encabezado TCP, 20 Bytes de encabezado IP (*Protocolo Internet*) y 14 Bytes de encabezado Ethernet. Si el paquete es UDP (*Protocolo de Datagramas de Usuario*), en lugar de 20 Bytes en la capa de transporte se tienen 8 Bytes para un total de 42 Bytes.

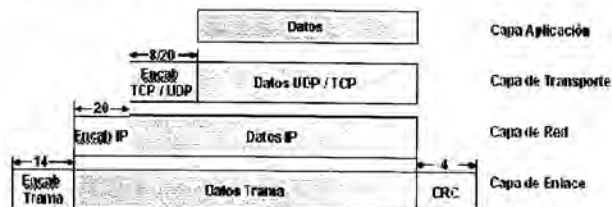


Fig. 1. Encapsulamiento de datos en TCP/IP

B. Generador de tráfico MGEN

La herramienta MGEN (Adamson y Gallavan, 1997) es uno de los generadores de tráfico más conocidos y útiles en el ámbito científico. Este software libre fue desarrollado por el Laboratorio de Investigación Naval de los Estados Unidos, y permite realizar mediciones y pruebas de rendimiento sobre una red, aplicando tráfico UDP/IP. La herramienta genera patrones de tráfico (*Periódico* y *Poisson*) en tiempo real, usados en una variedad de formas. Es posible usar archivos de scripts especiales para cargar dichos patrones en el tiempo. También permite emular patrones de tráfico en aplicaciones de tipo *unicast* o *multicast* usando los protocolos UDP/IP; además, soporta IPv6, y se integra en ambientes de simulación de redes como OPNET y ns-2. El log de datos de MGEN se usa para calcular tasas de pérdida de paquetes, estadísticas de rendimiento, retardos en la comunicación, entre otras. Actualmente, MGEN está en la versión 4.2 y tiene la ventaja de funcionar en variedad de sistemas operativos de red como sistemas basados en UNIX, MacOS X y plataformas Windows.

C. Métodos de Clustering Difuso

Los métodos de clasificación ayudan a establecer

la similitud (por lo general, un criterio de distancia) de un individuo i a un grupo o clase j . La clasificación puede realizarse en forma *no-supervisada*, cuando se realiza un aprendizaje a partir de los datos pre-procesados para construir clases, desconociendo la salida esperada del agrupamiento, y *supervisada*, cuando realiza un reconocimiento de patrones, conociendo la salida esperada de la clasificación. En el caso de no tener información sobre el agrupamiento esperado de los datos, se habla de *Clustering*. La partición de los datos en el espacio es difusa, lo cual permite definir un grado de pertenencia de un individuo a una clase. Para efectos de la aplicación que se va a implementar, se explican tres métodos de Clustering Difuso. El método *K-Means* (Maqueen, 1967) es uno de los métodos de agrupamiento, cuyos parámetros utilizados para la clasificación son el número de clases, K , y los centros iniciales, v_o , que ayudan a construir las clases. La particularidad de este método se basa en encontrar K , que puede ser definido por un criterio del experto de los datos o usando una validación de clases. Al realizar esta asignación, cada elemento de los datos, $X = \{x_1, \dots, x_i\}$, es enviado al grupo o clase cuyo centro sea el más cercano, obteniéndose una representación geométrica en el espacio de datos. La partición de los datos es estrictamente "dura", lo cual significa que cada elemento pertenece o no a una clase. Como este método es iterativo, cada vez que se encuentra las clases se recalculan los centros y se realiza una nueva asignación hasta obtener una clasificación concreta y estable. Para encontrar la mejor clasificación de los datos, se define una función objetivo (1), J , para minimizar el error en la clasificación:

$$J = \sum_{i=1}^N \sum_{j=1}^K d_{ij,A}^2 \quad (1)$$

donde el término i es i -enésimo individuo, j es el j -enésima clase y N es el máximo número de individuos. La función de distancia, $d_{ij,A}$, que es la separación de los datos, x_i , y los centros asociados v_o , definida por una métrica de distancia en forma matricial, A , denotado por:

$$d_{ij,A}^2 = \|x_i - v_j\|_A^2 = (x_i - v_j)^T \cdot A \cdot (x_i - v_j) \quad (2)$$

Si A es la matriz identidad, es decir, $A = I$, se encuentra un criterio de **distancia Euclidiana**. Esta misma distancia es utilizada en el algoritmo de los Mapas Auto organizados (*SOM*) (Mojonen, 1995). Usando (2), se establece una función de mínima distancia, S_{ij} , entre x_i y v_j definida en (3):

$$G_{ij} = \arg \min d_{ij}(x_i - v_j) \quad (3)$$

Con (3), se puede minimizar el error cuadrático de la función J (1) de una forma más eficiente que ayuda a encontrar una clasificación que no está presente en el método *K-Means* tradicional. Lo anterior es la base para combinar el algoritmo *K-Means* y *SOM* o algoritmo *SOM-Keas* (Souza et al., 2009). El método *FCM* o *Fusca C-Means* (Bedel, 1981) es un algoritmo de agrupamiento basado en la teoría de la Lógica Difusa, que busca minimizar la distancia Euclidiana (2) entre los datos, x_i , y los centros, c_j , y a cada dato, en función de su cercanía al centro, y se le asigna un *grado de pertenencia*, $u_{ij} \rightarrow [0,1]$, que asocia un individuo i a la clase j . La función objetivo del *FCM* incluye el término u_{ij} , como se muestra en (4):

$$J = \sum_{i=1}^N \sum_{j=1}^K \mu_{ij}^m d_{ij}^2 \quad (4)$$

donde $m \in (1, \infty)$. es el término de fuzzificación. A diferencia del método *K-Means*, la función J permite maximizar o minimizar la distancia de los elementos con respecto a su centro, con lo cual cada elemento tiene un grado de pertenencia a una clase. Gracias a este criterio, permite obtener particiones en el espacio de datos más "suave", cuya forma geométrica en D -dimensiones son *Hiper-esferas*. El método utiliza los mismo parámetros iniciales que el método *K-Means*. El algoritmo se inicializa con un centro aleatorio y se recalculan los centros en η iteraciones y cada iteración actualiza una *Matriz de Grados de Pertenencia* (5), U , de dimensiones $N \times K$, donde incluye u_{ij} del individuo i a la clase j .

$$U = \begin{pmatrix} \mu_{1,1} & \mu_{1,2} & \dots & \mu_{1,j} \\ \mu_{2,1} & \mu_{2,2} & \dots & \mu_{2,j} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{i,1} & \mu_{i,2} & \dots & \mu_{i,j} \end{pmatrix} \quad (5)$$

El método *LAMDA* (Piera et al., 1989) no requiere conocer el número de clases, porque permite actualizar clases y trabajar con datos cuantitativos o cualitativos. Este método estima el grado de adaptación de cada dato a cada clase. Esta adaptación se emplea en un sentido posibilista. El método analiza el *Grado de Adecuación Marginal (MAD)* de cada descriptor x_i a cada clase j y luego, a partir de una ponderación de *T-Norma* (β) y *S-Norma* (γ), se obtiene el grado de adecuación de cada dato a cada clase denominado *Grado de Adecuación Global (GAD)* (Piera y Aguilar, 1991), como se presenta en (6):

$$GAD(\bar{x}_i | K_j) = \alpha \gamma [MAD(\bar{x}_1 | K_j), \dots, MAD(\bar{x}_d | K_j)] + (1 - \alpha) \beta [MAD(\bar{x}_1 | K_j), \dots, MAD(\bar{x}_d | K_j)] \quad (6)$$

donde α es el grado de exigencia, $\forall \alpha \rightarrow [0,1]$ $\forall \alpha \rightarrow [0,1]$ y d es la máxima cantidad de descriptores de los datos. Para la adecuación de las clases se emplea la *NIC* o *Clase No Informativa*, que establece automáticamente un umbral mínimo a superar, para determinar la pertenencia de un elemento a una clase. En caso que no se supere el grado de pertenencia de la *NIC*, significa que no hay una adecuación importante a ninguna de las clases actuales, y por consiguiente, se crea una nueva clase con la información de la *NIC* y del dato que se está analizando. La Fig. 2 ilustra el esquema general del método:

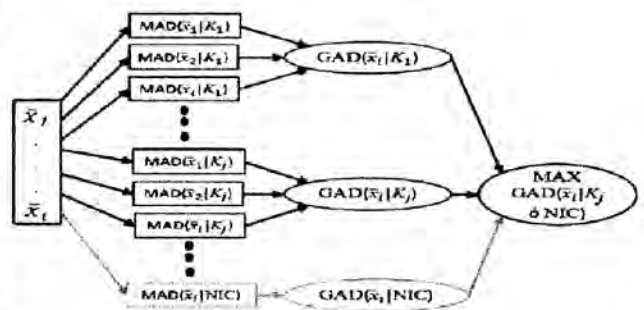


Fig 2. Esquema general del método *LAMDA*.

3. TESTBW

La herramienta de estimación de ancho de banda que permite adquirir los datos para utilizar la Red Neuronal se denomina *testbw*, desarrollada por los autores de este artículo. La herramienta genera estimaciones en tiempo

real de la cantidad de tráfico cruzado, que se captura en un periodo de tiempo T (7) dado por la diferencia entre los tiempos t de captura de dos paquetes consecutivos n y $n+1$:

$$T = t(n+1) - t(n) \quad (7)$$

El tiempo o *timestamp* en el momento de registrar la captura del paquete se presenta en el primer bit del paquete, y permite calcular el ancho de banda (BW) basado en la cantidad de bits (*size*) del primero de dos paquetes capturados. El

$$BW = \frac{size(n)}{T} \quad (8)$$

En la Fig. 3, se presenta la salida de la herramienta que genera el tiempo de captura de un paquete, el número de bytes del paquete y el ancho de banda (bits/seg) estimado previo a la captura del siguiente paquete. Nótese que la granularidad de las estimaciones es del orden de milisegundos.

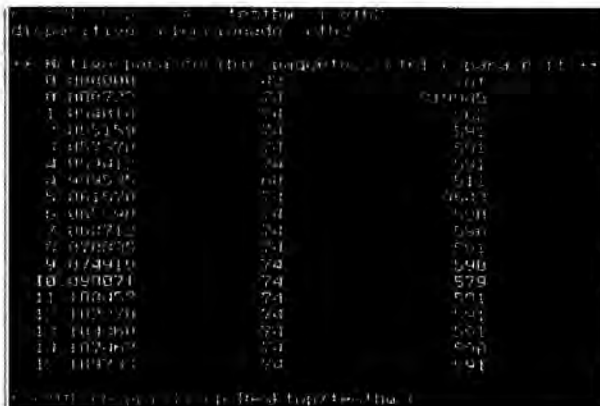


Fig. 3. Ejemplo de salida de *testbw* al calcular ancho de banda.

Para la evaluación del *testbed* se construyó el diseño mostrado en la Fig. 4. Esta red controlada consta de tres máquinas *Linux*, de las cuales dos son usadas para transmitir y recibir tráfico cruzado a través de un enrutador (*sender* y *receiver*). En estas dos máquinas se instaló el generador de tráfico *MGEN* versión 4.2.

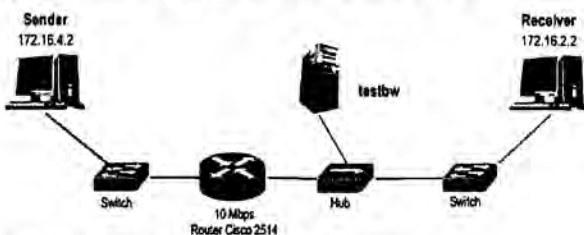


Fig. 4. *Testbed* de evaluación de *testbw*

La tercera maquina tiene la herramienta que monitorea el enlace de salida del enrutador y realiza los cálculos de tráfico cruzado en tiempo real. Las maquinas poseen sistema operativo *Linux Knoppix 5.0.1*, basado en *Debian*. Además, se realizaron pruebas utilizando la *distribución BackTrack y Ubuntu*.

4. RESULTADOS DE LA ESTIMACIÓN DE ANCHO DE BANDA DISPONIBLE

Mediante el *Testbed* desarrollado, se procedió a generar tráfico desde *MGEN* (de tipo Periódico y *Poisson*) con 30% (aproximadamente 381 paquetes/seg) y luego con 60% (aproximadamente 742 paquetes/seg) del total del canal. El Ancho de Banda utilizado es 10 Mbps (canal disponible de la red LAN probada). Para el 30% del canal, se envió un tráfico promedio de 2.993.136 *Bits/seg* ó 3 *Mbps* y en el caso de 60% del canal, se envió un tráfico de 5.829.152 *Bits/seg* ó 5.8 *Mbps*. En la Fig. 5, se ilustra los resultados generados por la tasa de tráfico *Periódico* de 30% durante un tiempo de 10 seg, con un total de 1891 paquetes capturados. Resultados similares se obtuvieron para 60% del canal utilizando tráfico de tipo *Periódico*. En la Fig. 6, se muestra el tráfico *Poisson* ocupando el 30% del canal, con un comportamiento de los datos dispersos y la mayor cantidad de datos se encuentra aproximadamente entre 1 Mbps y 3Mbps.

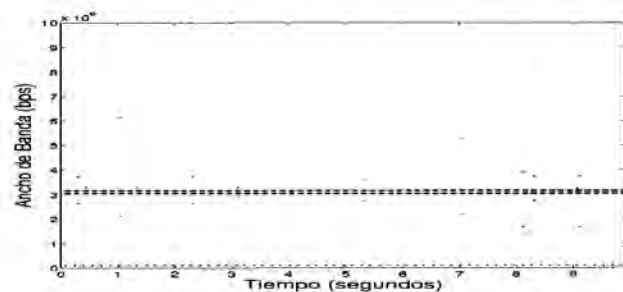


Fig 5. Ancho de Banda calculado al 30% del canal, usando distribución *Periódica*.

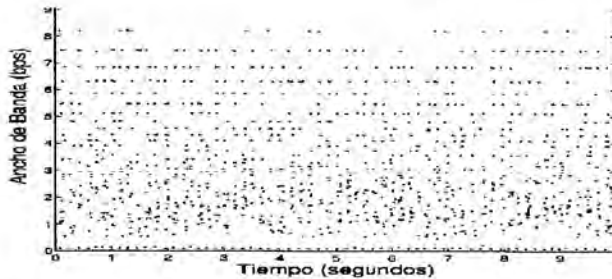


Fig 6. Ancho de Banda calculado al 30% del canal, usando distribución *Poisson*.

Los *timestamps* capturados (sin considerar el tamaño del paquete, porque es un valor fijo en toda la captura), se utilizaron como datos de entrada para la Red Neuronal. El formato de cada *timestamp* es *segundos.milisegundos*. La salida de la Red Neuronal va a estar representada por el Ancho de Banda Disponible (*BWD*) calculado en Bps (*bits por segundo*), como se muestra en (9):

$$BWD = f(\vec{w}, T) \quad (9)$$

donde \vec{w} es el vector de pesos de la Red Neuronal y T es el vector de tiempo o periodo de captura del *BWD*. Los datos adquiridos, cada uno con dos descriptores (*Tiempo* y *Ancho de Banda*), se muestran en la tabla 1:

Nombre	Tamaño
Periódico 30%	1891 x 2
Periódico 60%	3701 x 2
Poisson 30%	1864 x 2
Poisson 60%	3800 x 2

Tabla 1. Datos Adquiridos del *BWD*.

Los datos de tipo Periódico 30% y 60% se usaron para entrenar directamente la Red Neuronal *MLP* y no fue necesario hacer *Clustering Difuso*, porque no presentaban una alta dispersión. El método *Train Feed-Forward Network with Propagation (trainbp)* fue utilizado para el entrenamiento de la red con los datos de Periódico 30%, y el método *Levenberg Marquardt (trainlm)* fue utilizado para los datos de Periódico 60%. Estos algoritmos de entrenamiento presentaron los mejores resultados alcanzados, donde el 50% de los datos se usaron para entrenamiento y el otro 50% para validación o prueba. La tabla

2 muestran las estructuras de la Red Neuronal *MLP* donde se obtuvo una buena convergencia en el entrenamiento y una buena generalización de la red.

Tipo Tráfico	Capa Entrada	Capa Oculta	Capa de Salida
30% Periódico	40 Tansig	20 Tansig	1 Purelin
60% Periódico	24 Tansig	12 Tansig	1 Purelin

Tabla 2. Estructura de *MLP* para Tráfico Periódico

Las Fig. 7 y 8 muestran los resultados de la validación de la *MLP*, donde los puntos azules representan la salida de la *MLP* y los puntos rojos representan los datos originales. Se puede apreciar una buena aproximación al ancho de banda que es de 3 Mbps y 6 Mbps.



Fig 7. Validación de resultados para Periódico 30%.

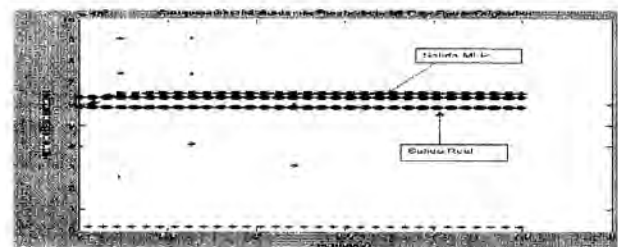


Fig 8. Validación de resultados para Periódico 60%.

Para el caso de los datos de *Poisson* 30% y 60%, se debió implementar una clasificación con la finalidad de reducir la dimensión de los datos. La anterior estrategia se implementó debido a la alta dispersión que encontrada. La idea principal es encontrar la mejor clasificación (validada por el experto en los datos), para lo cual se empleó la matriz de los centros y se realizó una normalización de estos datos para ingresarlos como entrada a la *MLP* (Ver Fig. 9). La tabla 3 muestra los parámetros implementados para

los métodos *SOM-KMeans* (para *Poisson* 30%), *FCM* y *LAMDA* (para *Poisson* 60%), donde se obtuvieron las mejores clasificaciones validadas por el experto.

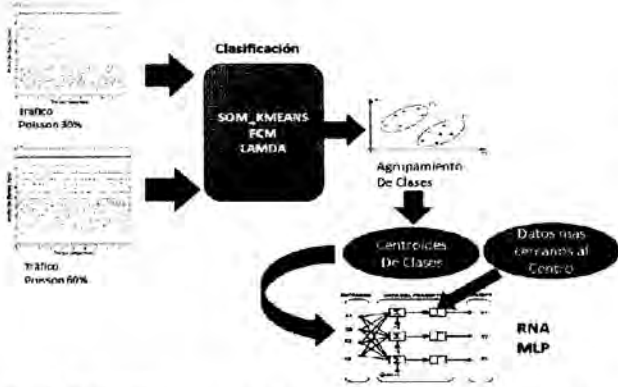


Fig 9. Esquema general para los datos de *Poisson*, usando *Clustering Difuso*.

Algoritmo	Parámetros				
	SOM-KMeans	Método	Clases	Error	Iteraciones
Batch		18	10^{-18}	1000	1
FCM	Clases	Factor of Fuzzification	Error	Iteraciones	Vol.
	18	2.0	10^{-18}	1000	1
LAMDA	Método	Exigencia	Iteraciones	Probabilidad	Conectividad
	Self-Learning	0.7	1	Gauss1	Min-Max

Tabla 3. Parámetros de Clasificación para SOM-Kmeans, FCM y LAMDA.

La Fig. 10 muestra la clasificación con SOM-KMeans para *Poisson* 30%, cuyo resultado fue mucho mejor con respecto a la clasificación obtenida con *FCM* o *LAMDA*, de acuerdo al criterio del experto. Usando el 50% de los datos para entrenamiento de la *MLP* y la matriz de los centros, se calculan una nueva distancia Euclidiana. Con la matriz de distancias, se calculan las menores distancias por cada clase (datos de entrada a la *MLP*) y luego, se encuentran los datos más cercanos a los centros (datos de salida de la *MLP*). Al validar con los datos, se encuentra una buena optimización del ancho de banda (Ver Fig. 11), usando 1 neurona en la capa de entrada, 4 neuronas de capa oculta, 1 neurona de salida y el algoritmo de entrenamiento *trainlm*.

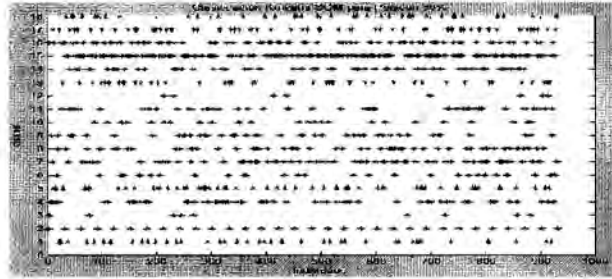


Fig 10. Clasificación con SOM-KMeans *Poisson* 30%.

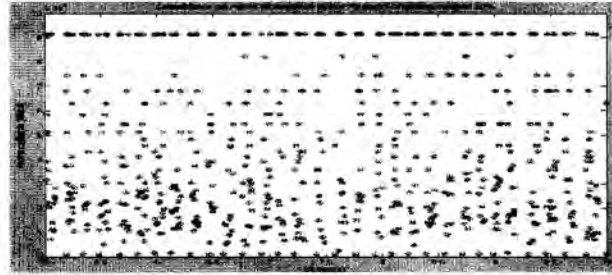


Fig 11. Validación con la MLP *Poisson* 30%.

La misma estrategia se implementó para los datos de *Poisson* 60%, pero la mejor clasificación se encontró usando los métodos *FCM* (Ver Fig. 12) y *LAMDA* (Ver Fig. 13). La validación con nuevos datos (Ver Fig. 14 y 15) entregó buenos resultados aplicando ambos métodos, usando 1 neurona en la capa de entrada, 10 neuronas en la capa oculta, 1 neurona de salida y el algoritmo de entrenamiento *trainlm*.

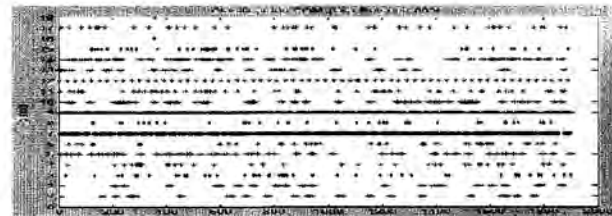


Fig 12. Clasificación con FCM *Poisson* 60%.

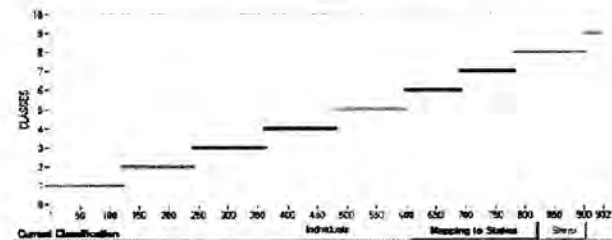


Fig 13. Clasificación con LAMDA *Poisson* 60%.

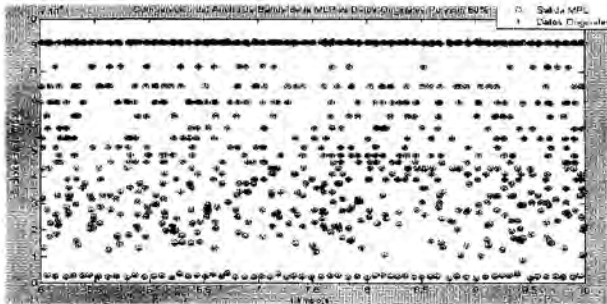


Fig 14. Validación con la MLP Poisson 60% (FCM).

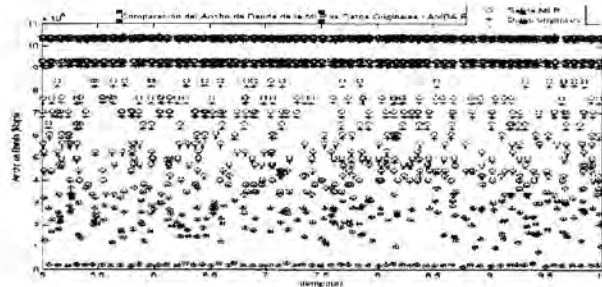


Fig 15. Validación con la MLP Poisson 60% (LAMDA).

5. CONCLUSIONES

Los métodos de Clustering Difuso permiten encontrar una clasificación no-supervisada, lo cual, al calcular la mínima distancia de cada clase y los datos más cercanos al centro de cada clase, permite reducir el tamaño de la estructura de la Red Neuronal MLP. Por consiguiente, el aprendizaje de los datos es más rápido con una óptima generalización de los datos. Este método es recomendable para casos con datos muy dispersos y difíciles de entrenar en una MLP. Esta propuesta permite optimizar el Ancho de Banda Disponible sin importar la alta dispersión de los datos, ayudando a garantizar al usuario de una red la reserva de los recursos de la misma, de acuerdo al tipo de tráfico que vaya a emplear. Se recomienda profundizar en el estudio de las mediciones de anchos de banda, aplicando teoría de colas, realizar mediciones de Anchos de Banda por medio de redes VPN y/o redes WAN, mejorar la función de captura de los *timesteps* de los paquetes enviados para el tráfico con patrón *Poisson*, aplicar modificaciones a la herramienta para la medición de anchos de banda sobre redes WLAN (Utilizando IEEE 803.11), generar soporte hacia el Protocolo IPv6, adaptar la herramienta a un entorno gráfico utilizando el lenguaje de

programación JAVA, por medio del API JPCAP. Lo anterior permitirá aprovechar más las ventajas que ofrecen los métodos de *Clustering Difuso*. Además, se espera aplicar Algoritmos Genéticos para optimizar el Ancho de Banda Disponible.

REFERENCIAS

- Adamson, B. and Gallavan, S. (1997). MGEN. <http://cs.itd.nrl.navy.mil/work/mgen/index.php>
- Bezdek, J.C. (1981). Pattern recognition with fuzzy objective function algorithms. *Plenum Publishing Corporation*, New York, USA.
- Carstens, T. (2001). Programming with pcap, <http://www.tcpdump.org/pcap.htm>.
- Guerrero, C.D. and Labrador, M. A. (2006). Experimental and Analytical Evaluation of Available Bandwidth Estimation Tools. In: *Proceedings of the IEEE Local Computer Networks*, pp. 710-717.
- Jacobson, V. Leres, C. and McCanne, S. (1989). *Tcpdump*.
- Jain, M. and Dovrolis, C. (2003). End-to-end Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput. *IEEE/ACM Transactions on Networking*, vol. 11, pp. 537-549.
- Kohonen, T. (1995). *The Self-Organizing Maps*. Springer - Verlag, Berlin, Germany.
- MacQueen, J. (1967). Some methods for classification and analysis of multivariable observation. *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, University of California, Berkeley, USA, vol. 1, pp. 281-287.
- Oetiker, T. and Rand, D. (1998). MRTG - Multi Router Traffic Grapher. <http://oss.oetiker.ch/mrtg/cast/index.en.html>.
- Piera, N., Deroches, P. and Aguilar.Martin, J. (1989). LAMDA: An Incremental Conceptual Clustering System. Report 89420, LAAS-CNRS, Toulouse, France.
- Piera, N. and Aguilar-Martin, J. (1991). Controlling Selectivity in Non-Standard Pattern Recognition Algorithms. *IEEE Transactions on Systems, man and Cybernetics*, vol. 21, N°1, pp. 71-82.
- Souza, J.R., Ludermir T.B. and Almeida, L.M. (2009). A Two Stage Clustering Method Combining Self-Organizing Maps and Ant K-Means. *ICANN 2009*, pp. 485-494.