

ESTADO DEL ARTE DE SCHEDULING ALGORITHMS (ALGORITMOS DE PROGRAMACIÓN DE ENVÍO)

Juan David Flórez Charry

Ingeniero de Control, MSc (c). en Ciencias de la Información y las Comunicaciones.
jdflorezc@correo.udistrital.edu.co

David Murillo Núñez

Ingeniero Electronico, MSc (c). en Ciencias de la Información y las Comunicaciones.
damurillon@correo.udistrital.edu.co

Cesar Hernández

Ingeniero Electronico, MSc. en Ciencias de la Información y las Comunicaciones, Especialización en redes Telemáticas y MSc. en Economía, docente de planta de la Universidad Distrital Francisco José de Caldas.
cahernandezs@udistrital.edu.co

Tipo: Artículo de investigación

Fecha de Recepción: Marzo 29 de 2011

Fecha de Aceptación: Mayo 10 de 2011

STATE-OF-THE-ART SCHEDULING ALGORITHMS

Abstract

This article presents state-of-the-art scheduling algorithms, which are employed to determine the method for a network device to forward packets to the transmission medium. Such algorithms are important in achieving Quality of Service on convergent networks. A good understanding of these algorithms is crucial when studying the traffic phenomena and also when fulfilling the requirements associated to different types of networks (e.g. Ethernet, TDM, WiMAX, Wireless, etc).

Keywords: Scheduling Algorithms, QoS, queue, queue size, prioritization, flujo, traffic.

Resumen

El presente artículo presenta un estado del arte de los llamados algoritmos de programación de envío, o en inglés "scheduling algorithms" los cuales determinan el método en que un dispositivo de red envía paquetes al medio de transmisión. Estos algoritmos son importantes para la consecución de calidad de servicio en redes convergentes y la correcta comprensión de ellos es determinante para entender los fenómenos de tráfico y cumplir los requerimientos dados en los diferentes tipos de redes (Ethernet, TDM, WiMAX, Wireless, etc).

Palabras clave: Scheduling Algorithms, QoS, colas, tamaño de cola, priorización, flujo, tráfico.

1. INTRODUCCIÓN

En redes de datos, en las cuales fluyen múltiples tipos de tráfico a través de un medio común se hace necesario definir mecanismos de clasificación de tráfico, con el fin de brindar un trato diferente a cada tipo de flujo en una red. A las técnicas que definen

cómo tratar los múltiples tipos de tráfico a lo largo de la red se les conoce como calidad de servicio (QoS). Los equipos de red, ya sean switches, routers o servidores deben ser programados para poder cumplir con las políticas de QoS establecidas en la red para los flujos de tráfico y para lograrlo deben poner en el medio los paquetes de

acuerdo a diferentes prioridades. Para el manejo correcto de esta situación se define lo que se denominan scheduling algorithms los cuales indican en qué orden poner paquetes en el medio dependiendo de la clase de tráfico a la cual pertenezcan. En el presente paper lo que se hará es brindar una pequeña definición de los scheduling algorithms más importantes y los estudios recientes realizados sobre cada uno de ellos para establecer el estado del arte de los mismos.

2. FIRST COME FIRST SERVED (FCFS)

Es la forma más sencilla de poner paquetes en el medio; tal como lo indica su nombre trata simplemente de atender los paquetes en el mismo orden que llegan, es decir, se define una única cola de atención y el primero en llegar será el primero en atenderse. Al tratar los paquetes de esta forma no existe QoS ya que el primer paquete en llegar podría ser de baja prioridad y el último en llegar el de más alta prioridad, y siguiendo este algoritmo se atendería primero el paquete de baja prioridad generando un alto retardo en el paquete de alta prioridad ya que lo deja de último en la cola de atención [1].

No se recomienda utilizarlo en entornos que requieran calidad de servicio, debido a que este algoritmo no puede ofrecer ventaja alguna, sin embargo, tampoco debe menospreciarse completamente debido a que es tan sencillo de implementar, que puede ser utilizado en sistemas que no tengan requerimientos específicos de QoS, lo cual ocurre en problemas del día a día tales como pedir una cita médica, comprar comida en un restaurante, pagar los artículos en el supermercado, hacer reservas en un hotel o programación de aterrizaje de aviones [2] en un aeropuerto.

Para el caso de telecomunicaciones esto puede aplicarse por ejemplo en [3], en donde se plantea el uso del principio de FCFS para disminuir el tiempo de recuperación

de una red con enlaces redundantes, cuando hay cambios de topología, es decir, cuando un enlace presenta una caída o cuando se reestablece tras una caída. En este caso los autores proponen aplicar el concepto de FCFS al protocolo de enrutamiento y así minimizar el tiempo que toma el envío del cambio de estado de un nodo a los demás. Esto es particularmente útil en redes ópticas ya que dadas las altísimas velocidades el QoS en ocasiones pasa a un segundo plano y dada precisamente la sencillez del algoritmo, ayuda a disminuir los tiempos de espera. Adicionalmente si se tiene una mayor cantidad de enlaces que hagan que la topología sea en malla (característica primordial en redes de core), los experimentos realizados en este trabajo demuestran que incluso si hay una mayor cantidad de nodos, el tiempo de los recálculos del algoritmo de enrutamiento disminuye. Lo anterior queda expresado en la Fig. 1.

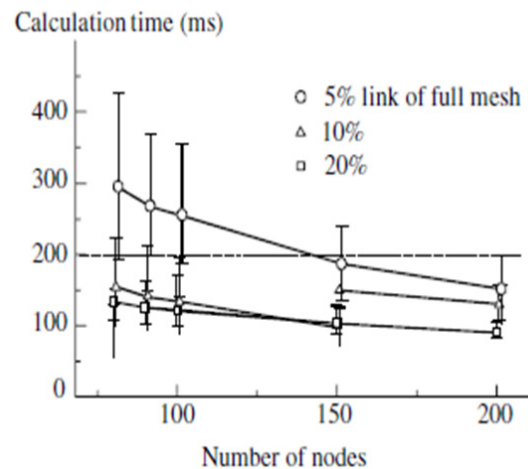


Fig. 1. Número de nodos vs tiempo de cálculo [3].

Adicionalmente han realizado varios estudios en los que se realizan comparaciones entre FCFS y otros Scheduling Algorithms para medir la eficiencia de unos versus otros como en el caso de [20] y [21].

3. ROUND ROBIN (RR)

Este algoritmo fue desarrollado para cu-

brir algunas de las desventajas de FCFS y consiste en definir no una cola sino varias colas y se ubican los paquetes en diferentes colas dependiendo de cómo estén marcados. Luego se hace una ronda en la que se atiende el primer paquete de cada cola. Esto genera mayor justicia en la atención de los paquetes ya que no todos pertenecerán a la misma cola sino que cada uno estará ubicado en una diferente. Hay que tener en cuenta que el método de atención para cada cola es FCFS, y por ello siendo un algoritmo sencillo de implementar y dependiendo del número de colas definidas puede ser un algoritmo de poca flexibilidad [1].

La mayor desventaja de este algoritmo es el hecho de que no se puede determinar la longitud de cada paquete en las diferentes colas entonces a veces se podría atender en rondas, paquetes de 64 Kb en una cola y paquetes de 1500 Kb en otra haciendo que el concepto de justicia en cada ronda se pierda completamente.

De acuerdo a lo planteado por Yuan y Duan en [4] el algoritmo Round Robin se comportará de una manera justa a largo plazo, sin embargo, destacan su importancia debido a su simplicidad por lo cual lo toman como una alternativa a tener en cuenta. Para corregir el problema de la falta de justicia a corto plazo, los autores plantean una variación a la que denominan Fair Round Robin (FRR).

Esta alternativa consiste en asignar un peso específico a cada cola y de esta forma intentar distribuir la información no por paquetes sino por cantidad de bytes lo cual brindará al usuario una sensación de justicia, en comparación al round robin tradicional. Por otro lado Hao Yu [5] propone una forma novedosa de utilizar round robin en escenarios de multicast con lo cual las colas ya no aplican para tráfico unicast sino que crea colas multicast y les da un tratamiento tal que le permite incrementar aun mas las técnicas de multicast para aplicaciones como TV sobre IP o radio. Nuevamente se

aprovecha la simplicidad del algoritmo para hacerlo una alternativa válida.

Adicionalmente existen varios estudios más aunque casi todos se enfocan en aplicativos que operan con el algoritmo de round robin mas no en mejoras o nuevas propuestas del algoritmo en si; algunos de estos estudios se presentan en [22], [23] y [24].

4. STRICT PRIORITY (SP)

En este algoritmo se tienen varias colas al igual que en RR, sin embargo ya no se realizará una ronda atendiendo un paquete de cada cola sino que también se le asignarán prioridades a cada cola, y habiendo hecho esto, el algoritmo solo atenderá la cola de mayor prioridad hasta que esta esté vacía y luego atenderá a la segunda de mayor prioridad siempre y cuando la de mayor prioridad continúe vacía; de igual forma solo se atenderá la tercera cola de mayor importancia solo mientras la de mayor importancia y la segunda estén vacías y así sucesivamente con las demás [1].

La desventaja de este algoritmo es que los paquetes en la cola de baja prioridad podrían quedarse por siempre en la cola si llegan continuamente paquetes a las colas de alta prioridad, ya que el algoritmo siempre les dará prioridad a las colas de alta prioridad. De ahí que se incluya la palabra "estricto" para este algoritmo.

Uno de los avances más recientes es el que se plantea en [6], el cual propone una forma adaptativa de este algoritmo para aplicar en redes WiMAX.

Esta adaptabilidad permite enfrentar la desventaja enunciada anteriormente, es decir, le quita un poco de rigidez al algoritmo ya que la elección de la cola más importante se hace dependiendo del canal que tenga menos congestión o menos interferencia en un momento dado, a partir de lo definido en el estándar 802.16d.

En este mismo trabajo se hizo un compara-

tivo del comportamiento de este algoritmo contra el WFQ (se hablará más adelante de éste) y se encontró que el delay general es más bajo utilizando esta nueva propuesta tal y como se ve en la Fig. 2, teniendo en cuenta que la línea con puntos representa el nuevo algoritmo y la línea punteada el WFQ.

Adicionalmente en [25], [26], [27] y [28] se desarrollan estudios de aplicaciones específicas a nivel de red aunque conservando siempre la misma definición original del esquema de SP.

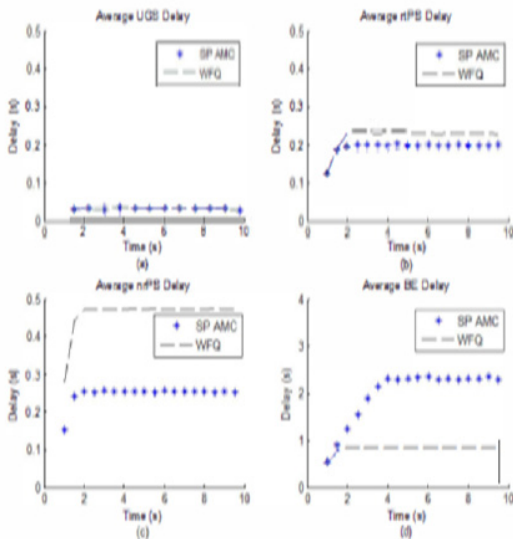


Fig. 2. Comparación de retardos entre SP adaptativo vs WFQ [6].

5. EARLIEST DEADLINE FIRST (EDF)

Este algoritmo lo que hace es asignar un límite de tiempo a cada paquete y asegura que el envío se realizará antes de que se cumpla dicho tiempo, enviando el paquete que está más cerca del tiempo límite en cada ciclo [1].

Asumiendo que el momento de llegada de un paquete j que hace parte de un flujo i es a_{ij} y que el retardo máximo permisible para el flujo i es d_i , la idea es que el paquete no esté en cola por más de $a_{ij} + d_i$.

Se plantea una condición de suficiencia para que el algoritmo sea válido y se puedan enviar los paquetes antes de llegar al tiempo límite. Esta condición se presenta a continuación en Ec. (1)

$$\sum_i A_i(t - d_i) + l_{\max} I_{(d_{\min} \leq t \leq d_{\max})} \leq Ct \quad (1)$$

En donde $A_i(t)$ representa la cantidad de tráfico del flujo i entrando a la cola en un intervalo de tiempo t , C la capacidad del enlace de salida, l_{\max} el tamaño máximo posible del paquete considerando todos flujos, $d_{\min} = \min_i\{d_i\}$ y $d_{\max} = \max_i\{d_i\}$.

Dentro de los estudios más recientes se tiene el planteado en [7] en el cual se combina las técnicas EDF con RR generando así lo que denominan los autores EDF-RR y lo implementan en un sistema en tiempo real como lo es un switch ATM con celdas de tamaño definido. Lo que se busca es sacar provecho de las ventajas de cada uno de los algoritmos, es decir, de la facilidad de implementación y de la sencillez de RR combinado con la capacidad de atender en un tiempo límite determinado que ofrece EDF.

Los resultados finales de implementar esta técnica resultan tanto en un retardo promedio menor así como un tamaño promedio de cola menor en comparación con la implementación de cada algoritmo por separado tal y como se muestra en la Fig. 3.

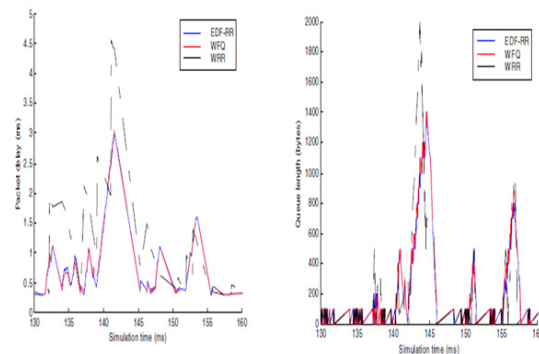


Fig. 3. Comparación entre EDF-RR vs RR [7].

Por otra parte los autores de [8] evalúan el comportamiento de EDF en un entorno inalámbrico 802.11. La conclusión de este trabajo lleva a que si bien el algoritmo es utilizable en estos entornos, es necesario plantearlo de una manera distribuida, es decir, se tiene que analizar cada una de las estaciones que pretenden tener acceso al medio y dicho análisis se hizo utilizando cadenas de Markov.

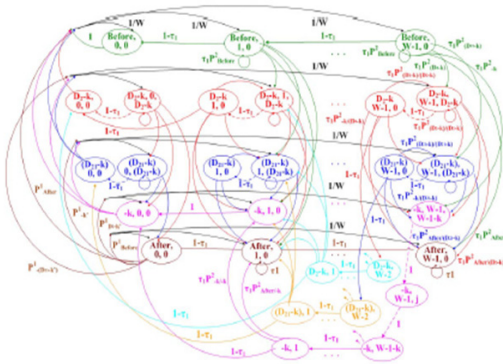


Fig. 4. Cadena de Markov para una estación que requiere acceso a un entorno inalámbrico [8].

Suponiendo que tan solo hay 2 estaciones la cadena de Markov resultante para una de ellas se muestra en la Fig. 4.

Al detallar esta figura resulta evidente la complejidad que conlleva este análisis lo cual dejará como resultado final una difícil implementación la cual posiblemente no tendrá resultados óptimos en tiempo real. Por último, se han realizado estudios que demuestran que el uso de EDF en redes puede mejorar desempeños. Ejemplos de estos estudios se muestran en [29], [30] y [31].

6. GENERALIZED PROCESSOR SHARING (GPS)

Este algoritmo es más avanzado que los anteriores y consiste en generar varias colas y asignar un peso a cada tipo de tráfico el cual está en cada cola y de acuerdo a la capacidad total del enlace enviar determinada cantidad de información de cada cola

dependiendo del peso de la misma. Así por ejemplo, suponga que en una cola 1 hay 400 Kbytes de información y en otra (cola 2) 200 Kbytes de información. Suponga además que la cola 1 tiene un peso equivalente al doble de la cola 2. En este orden de ideas, si se tuviera un enlace de 64 Kbps se enviarían 42 Kbytes de la cola 1 mientras que se envían apenas 21 Kbytes de la cola 2, y de esta forma se evidenciará la prioridad de una información sobre la otra. Se asume en este caso que el tráfico es infinitamente divisible. Esto hace que en la práctica sea imposible de implementar, sin embargo, para estudios teóricos ha sido de mucha utilidad.

En [1] se hace un planteamiento más formal según el cual existen N flujos, y a cada flujo i se le asigna un peso w_i . Además, se define la función $S_i(\tau, t)$ que expresa la cantidad de tráfico del flujo i servida en un intervalo entre τ y t . Según lo anterior un proceso de servicio GPS se define como se muestra en la Ec. (2)

$$\frac{S_i(x, t)}{S_j(x, t)} \geq \frac{w_i}{w_j}, j = 1, 2, \dots, N \quad (2)$$

Al atrasar el flujo i se garantizará una rata mínima dada en la Ec. (3), en donde C es la capacidad del enlace de salida:

$$g_i = \frac{w_i}{\sum_j w_j} C \quad (3)$$

Un detalle a tener en cuenta, es el hecho de que los pesos de cada flujo (w_i) siempre son constantes y de acuerdo a lo planteado en [9] esto es causal de retrasar innecesariamente algunas sesiones ya que el flujo es dinámico y por ende no siempre las colas tendrán la misma longitud. Para atacar ésta desventaja los autores proponen una variación a la que denominaron Output Rate-Controlled GPS (ORC-GPS).

En esta variación lo que hicieron fue controlar la rata de cada flujo de tráfico utilizando la técnica leaky bucket para de esta

forma hacer que el comportamiento del algoritmo no sea tan estricta como en su definición inicial sino que pueda variar dependiendo de la cantidad de tráfico y la rata de los diferentes flujos en un enlace. Para demostrar lo anterior se hacen experimentos los cuales se arrojan resultados mostrados en la Fig. 5 en donde se demuestra las mejoras de ORC-GPS vs GPS, en términos de los retardos contra la cantidad de tráfico en el sistema.

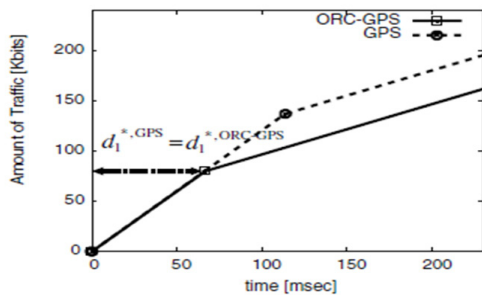


Fig. 5. Cantidad de tráfico vs retardo en ms [9].

Posteriormente, los mismos autores de [9] reaparecen en [10] para proponer una mejora a su propio algoritmo planteado anteriormente; esta mejora consiste en hacer que los ajustes necesarios para que planteamientos hechos en [9] apliquen no solo para un nodo sino para varios nodos como se muestra en la Fig. 6, y se mide el retardo del tráfico extremo a extremo en una red de prueba.

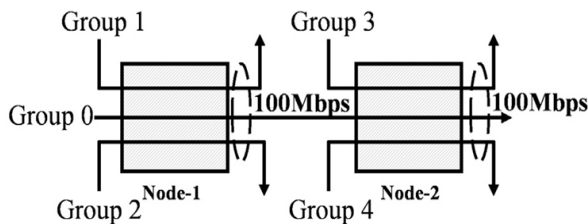


Fig. 6. Red de 2 nodos [10].

Otros estudios como el realizado en [11] tratan el siguiente problema: dado que se asume que el tráfico es infinitamente divisible, se tendría entonces partes de flujo tan pequeñas que requerirían un ancho de banda de 0 por lo cual es necesario hacer una

aproximación práctica a fin de que pueda ubicarse un ancho de banda real para cada tipo de tráfico.

El resultado de este estudio son 3 propiedades las cuales garantizarán la ubicación de un único ancho de banda para cada flujo. Estas 3 propiedades son:

1. Si ninguna de las colas está vacía entonces el ancho de banda total requerido está dado por la Ec. (4):

$$\sum_{i=1}^n r_i(t) = C \quad (4)$$

En donde $r_i(t)$ es el ancho de banda reservado para el flujo i en el tiempo t .

2. Si el tipo de tráfico i se retrasa en el intervalo (τ, t) entonces para cualquier j :

$$\frac{S_i(x, t)}{S_j(x, t)} \geq \frac{c_i}{c_j} \quad (5)$$

En donde

$$S_i(x, t) = \int_x^t r_i(u) du \quad (6)$$

Representa la cantidad de tráfico de la clase i servido en el intervalo $[\tau, t)$. Es de notar que $S_i(\tau, t)$ es continua.

3. $r_i(t) \leq a_i(t)$ para todo i en donde la cola i esté vacía, en donde $a_i(t)$ representa la rata de arribo de tráfico de clase i en el tiempo t .

Finalmente, se tienen algunos estudios más, aplicados en redes [32], [33], [34] y [35] en los cuales se aplica el concepto de GPS para lograr eficiencia en el desempeño.

7. PACKET-BY-PACKET GENERALIZED PROCESSOR SHARING (PGPS)

Este algoritmo presenta el escenario práctico que más se asemeja al GPS ideal, y puede ser implementado en redes de conmu-

tación por paquetes ya que aquí ya no se asume que el tráfico es infinitamente divisible ya que los paquetes en realidad no lo son [1].

Para saber realmente cuanto se acerca PGPS al GPS ideal lo que se hace es encontrar el peor de los casos de PGPS. En este caso (el peor de los casos) se cumplen dos teoremas:

1. Para todo paquete p , sean G_p y F_p los tiempos de salida del paquete p bajo el esquema PGPS y GPS respectivamente. Entonces:

$$G_p - F_p \leq \frac{L_{\max}}{C} \quad (7)$$

En donde L_{\max} es la longitud máxima de un paquete y C la capacidad del enlace de salida.

2. Para todos los tiempos τ y todos los flujos i :

$$S_i(0, \tau) - S'_i(0, \tau) \leq L_{\max} \quad (8)$$

En donde $S_i(a, b)$ y $S'_i(a, b)$ son la cantidad de flujo i servidos en el intervalo $[a, b]$.

En [12] se presenta un esquema basado en PGPS para redes móviles llamado MIMO PGPS, el cual consiste en proveer acceso a los canales conservando el trabajo. Esta técnica adicionalmente compensa el delay para cada usuario y el retraso en los usuarios.

Los resultados de este algoritmo, en comparación a otros se muestran en las Fig. 7 y 8.

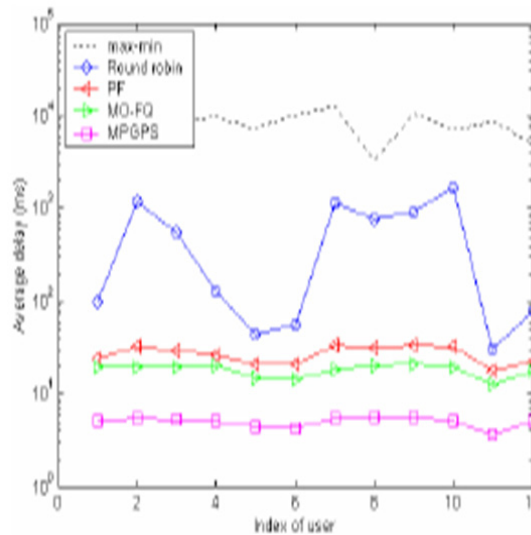


Fig. 7. Retardo promedio de cada usuario con una carga de 84.34 Mbps [12].

Para concluir esta parte, cabe apuntar que se realizaron estudios especificados en [36], [37] y [38], en los cuales se mide el comportamiento de varios tipos de redes que implementan el concepto de PGPS.

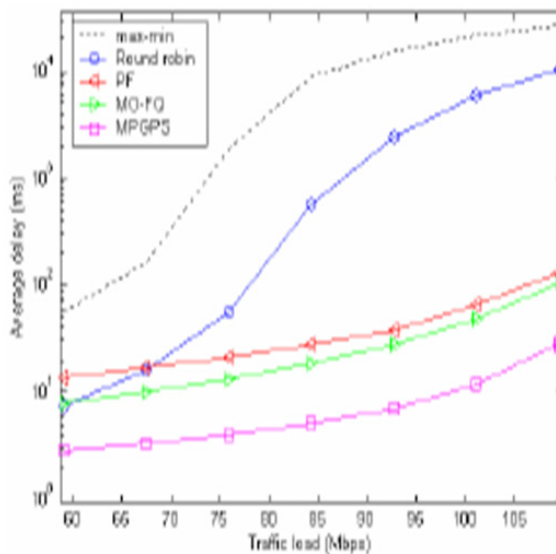


Fig. 8. Retardo promedio de los usuarios bajo diferentes cargas de tráfico [12].

8. IDEALIZED WIRELESS FAIR QUEUEING (IWFQ)

Este es uno de los primeros algoritmos que se utilizó como método de acceso a las redes inalámbricas [1]. Su sistema de referencia es el modelo inalámbrico WFQ, y se dice que es ideal porque se basa en dos supuestos claves ideales [13]:

Cada flujo sabe si puede transmitir correctamente en el slot actual, esto implica que cada flujo conoce el estado de la red al momento de transmitir.

Que existe o se tiene un protocolo MAC perfecto.

La diferencia entre IWFQ y WFQ, se enfoca en el primer supuesto, y se trata que IWFQ no va a transmitir si no encuentra un slot en buen estado para la transmisión si esto no sucede busca el siguiente paquete con el menor tiempo virtual y este es elegido, y se reinicia el proceso hasta que encuentre un slot en buen estado. El algoritmo funciona de la siguiente manera:

1. Se simula un flujo que se encuentre libre de error en cualquier instante de tiempo. El tiempo virtual para el algoritmo es el mismo tiempo seleccionado para para el flujo anterior en ese mismo instante.
2. Cada paquete que llega es etiquetado de la misma forma que en WFQ [14]: un paquete que llegue con un número de secuencia de llegada a tiempo se le asigna dos etiquetas, una de inicio y una de finalización.
3. Se realizan las acciones del "Scheduling loop", entre ellas esta reajustar las etiquetas de cada flujo, a cada flujo poner la etiqueta de servicio igual a la etiqueta del paquete head-of-line (HOL) del flujo.

A un flujo que se le haya negado el servicio debido a errores del canal, se le garantizara servicio, aunque el canal no cambie de estado mientras no se modifiquen las etiquetas del flujo. Se dice que un flujo es lagging,

o in sync cuando el tamaño de la cola es más pequeño que, más grande que o igual al tamaño de la cola del sistema de referencia. Para evitar una cantidad ilimitada de compensación que consuma otros enlaces en buen estado, la cantidad total de lag que será compensado entre todos los flujos lagging estará limitada por la cantidad de bits B , definidos en el sistema de referencia.

Por otra parte se tienen los trabajos realizados en [39], [40], [41], [42] y [43] en los cuales se aplica a situaciones, el concepto de IWFQ.

9. CHANNEL CONDITION INDEPENDENT PACKET FAIR QUEUEING (CIF-Q).

El algoritmo channel condition independent Packet Fair Queueing (CIF-Q), también utiliza un algoritmo justo y libre de error como sistema de referencia [1]. En [15] utilizan como sistema de referencia Start-time Fair Queueing (SFQ), aunque teóricamente se puede utilizar cualquier otro algoritmo de los bien conocidos como WFQ, SCFQ, WF22Q, EEVDF.

Entonces, para llevar una cuenta del comportamiento del servicio en un sistema con errores, se asocia a cada sistema S un sistema de referencia S_r [15], de la misma forma que se hace en WFQ, un sistema o flujo se califica según el la referencia en lagging, leading, o satisfied, de esta forma una sesión está en estado leading cuando recibe más servicio que la referencia S_r , esta en lagging cuando recibe menos servicio y en satisfied cuando recibe la misma cantidad. Existen dos temas que se deben tener en cuenta al momento de realizar las funciones de scheduling, la primera es que las decisiones de scheduling son tomadas S_r , y no en S , más detalladamente la sesión que tiene el tiempo virtual más pequeño es la seleccionada para ser usada en S . Lo segundo, no importa cual sesión este sirviendo en S , en S_r se asume que el paquete que se está sirviendo pertenece a esa sesión y los

tiempos virtuales son actualizados acordes a la sesión.

- Los tiempos virtuales de las sesiones no mantiene rastro del servicio obtenido de la sesión recibida.
- Para asegurar la justicia del algoritmo se utiliza un parámetro adicional llamado Lag.
- Una sesión en estado leading, no se le permite abandonar hasta que no pierda este estado.
- Para enfrentar el problema en el cual todas las sesiones presenten error, se introduce el concepto de compensación forzada [15].

La implementación de este algoritmo es más sencilla comparado con IWFQ, debido a que este algoritmo no utiliza un emulación de la sesión como en IWFQ [13].

Algunas aplicaciones de este algoritmo se pueden encontrar en los trabajos realizados en [44] y [45].

10. IMPROVED CHANNEL STATE DEPENDENT PACKET SCHEDULING (I-CS-DPS)

Este es un algoritmo para el acceso inalámbrico que utiliza una versión mejorada del Deficit Round Robin (DRR). En la siguiente figura se muestra la estructura de la red, en la cual se describe el modelo Adaptative – QoS, para asegurar la calidad del servicio.

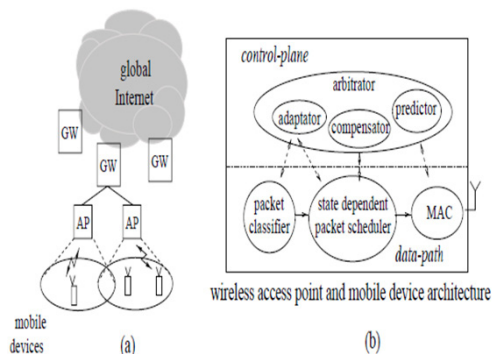


Fig. 9. Havanna Framework [16].

En la Fig. 9 (a), se observa la estructura de

la red en la cual se utilizan varios gateways para acceder a internet, que se encuentran interconectados a los Access Point (AP), que son los usados por los usuarios finales para poder ingresar a la red. En la Fig. 9 (b), se encuentra el plano de control de los AP, sobre el cual se implementó el modelo de calidad, que tiene como principal característica el uso de tres entidades nombradas como adaptador, compensador y predictor, todos estos bajo el mando del Arbitrator o juez, que es el que coordina toda la operación, enviando y recibiendo los mensajes a las tres entidades. Antes de que un paquete se transmita el juez solicita al predictor que realice un análisis sobre la red para poder conocer su estado y decidir si se puede transmitir o no, si no se puede transmitir se almacena el paquete y se invoca al compensador para que maneje el flujo de “créditos”, si el buffer de almacenamiento de paquetes está a punto de sobrecargarse se invoca al adaptador que se encarga de modificar las características de los paquetes cambiándoles prioridades para poder transmitirlos [16].

El compensador es la entidad que utiliza el algoritmo modificado de DRR, para “acreditar” y “compensar” las posibles injusticias que se presenten debido al mal estado de los canales que sufren los móviles al acceder a la red. La transmisión es controlada por los siguientes parámetros: quantum (Q) y un déficit counter (DC), donde Q, lleva el conteo del número de cuotas de bytes dados para el flujo en cada ronda, y mientras que el DC lleva el rastro de la historia de transmisión de “crédito” por cada ronda [17]. Una ronda se define como la visita del programador o “scheduler” por todas las colas aunque sea una vez. Al principio de cada ronda el Q se agrega al DC por cada flujo que exista en el sistema, después de esto el programador recorre cada flujo y compara si el DC es mayor al tamaño del paquete de cabecera para poder transmitir, si es mayor verifica el canal para poder transmitir, si no es mayor no transmite y acumula el DC para la siguiente ronda, si el

canal está en buen estado y el DC es mayor transmite y descuenta el valor del DC. Con el DRR modificado se tiene un valor nuevo que es el contador de compensación CC, que lleva la cuenta de cuantos Bytes no pudieron ser transmitidos por ronda debido a problemas de estado del enlace, entonces cuando un flujo no transmite por problemas del enlace el valor Q se le resta a DC y se le suma a CC como compensación, en la siguiente ronda cuando se vuelva a calcular el valor de DC se le adiciona un valor de compensación $\alpha \cdot CC$ donde α es un valor entre 0 y 1 [1], [16].

Entre los estudios realizados se destacan los presentados en [46] y [47] los cuales tratan de aplicaciones puntuales de este algoritmo.

11. PROPORTIONAL FAIR (PF)

Dos de los algoritmos mas usados en los últimos años, en las redes celulares son: el Maximum Carrier - To - Interference Ratio (Max CIR) y el Proportional Fair (PF) [1]. Max CIR busca maximizar la capacidad del sistema permitiendo el acceso o atendiendo a los flujos en los cuales el canal se encuentra en buenas condiciones o tiene una calidad de servicio alta [17]. PF intenta maximizar el grado de justicia a través de los canales seleccionando aquellos que tienen la mejor calidad de canal relativa [18]. Sin embargo, estudios reciente han mostrado que PF da cierta prioridad a los canales con una alta variación de calidad [19].

PF está definido como se muestra a continuación:

Un "Scheduling" o programación P, es justa si y solo si, para cualquier programación S, satisface lo siguiente:

$$\sum_{i \in U} \frac{R_i^{(S)} - R_i^{(P)}}{R_i^{(P)}} \leq 0 \quad (9)$$

Donde U es la configuración del usuario y $R_i(S)$ es el promedio de uso de un usuario

del esquema S. También se sabe que la asignación de P es la suma promedio de todas las tasas de un usuario como se observa en la Ec. (10):

$$P = \operatorname{argmax}_s \sum_{i \in U} \log R_i^{(s)} \quad (10)$$

Considerando N usuarios en el sistema, que usan el mismo canal inalámbrico, las condiciones del canal varían para cada usuario. La calidad del canal se determina cuando el usuario realiza mediciones al canal y se las envía al PF scheduler que se encuentra ubicado en la estación base, si la realimentación de la base se demora un tiempo relativamente menor al tiempo de envío de información del usuario se asume que el canal se encuentra en buenas condiciones. El algoritmo PF se propuso después de ver los inconvenientes que presentaba CDMA al hacer un uso injusto del canal, favoreciendo a los usuarios que se encontraban cerca a las estación base, como se mencionó anteriormente PF busca hacer que el sistema sea más justo sin degradar las características del sistema [1], esto ha hecho que este algoritmo reciba mucha atención y sea uno de los que más se usan ahora. El PF puede lograr diversidad multiusuario, debido a que el Scheduler puede detectar los cambios de calidad en los canales de los usuarios y asignar a aquellos que tengan el rango de calidad más alto, en otras palabras el esquema de PF es un esquema basado en los estados de los canales [1].

Para terminar se tienen algunos trabajos sobre aplicaciones de este algoritmo presentadas en [48], [49] y [50] las cuales reflejan sus ventajas.

12. CONCLUSIONES

Tras haber visto varios algoritmos de programación de envío se puede afirmar lo siguiente:

- Cada algoritmo tiene sus propias ventajas y desventajas, las cuales van desde complejidad de implementación hasta

- variedad de esquemas de QoS que pueden tratar.
- Los scheduling algorithms van evolucionando de acuerdo a las nuevas necesidades identificadas de una manera progresiva, es decir, no se genera un nuevo paradigma sino que se toma un algoritmo ya desarrollado y se le hacen pequeñas modificaciones a fin de cubrir la nueva necesidad o superar la desventaja identificada.
 - Los nuevos algoritmos o sus respectivas variaciones surgen cuando aparece la necesidad por parte de las demandas de tráfico sobre la red; en este orden de ideas, dado que la cantidad de usuarios y aplicaciones de red crece a diario, continuamente la demanda aumenta y por ello es de esperar que sigan apareciendo variaciones o nuevos estudios sobre cómo utilizar las técnicas de envío de paquetes al medio.
 - La complejidad de cada algoritmo afecta el tiempo de respuesta y dado que en estos sistemas se requieren respuestas en tiempo real, es importante saber que las nuevas implementaciones deben intentar ser lo más sencillas posibles.
 - El medio y tipo de tecnología de acceso utilizada influye también para el algoritmo de envío ya que no se presentan las mismas condiciones en entornos Ethernet, TDMA o inalámbricos y esto hace necesaria la implementación de algoritmos diferentes.
 - Se encuentra que unagran parte de referencias son de autores de Asiaoriental por lo cual se evidencia que la mayor parte de investigación en estos temas se centra en esa parte del mundo.

Referencias Bibliográficas

- [1] Tsung-Yu Tsai, Yao-Liang Chung y Zsehong Tsai; "Introduction to Packet Scheduling Algorithms for Communication Networks". Jun Peng (Ed.), ISBN: 978-953-307-114-5, InTech, pp 263-285. Septiembre de 2010.
- [2] Chen Shuang, Xia Xuezhi. "Researches on Optimal Scheduling Model for Aircraft Landing Problem". Information Engineering, 2009. ICIE '09. WASE International Conference on. Vol 1 pp 418-421, 2009.
- [3] Yamada, T, Kimura, S, Iwatsuki, K, Imai, T. "New High Speed Optical Routing Architecture Based on First-Come-First-Serve Principle ". Optical Fiber Communication and the National Fiber Optic Engineers Conference, pp 1-3, 2007.
- [4] Xin Yuan y Zhenhai Duan; "Fair Round-Robin: A Low-Complexity Packet Scheduler with Proportional and Worst-Case Fairness". IEEE Transactions on computers, Vol. 58, No. 3, pp 365-379, Marzo 2009.
- [5] Hao Yu, "A Novel Round-Robin Based Multicast Scheduling Algorithm for 100 Gigabit Ethernet Switches". INFOCOM IEEE Conference on Computer Communications Workshops, pp 1-2, 2010.
- [6] Ihsan Shahwan, Ali Muhammed, Muath Obaidat, Roger Dorsinville. "WiMax: Cross Layer Bandwidth Allocation Strict Priority Based Adaptive modulation and Coding" International Conference on Communications and Information Technology (ICCIT), pp 73-76, Aqaba 2011.
- [7] Liu, D, Lee, Y.-H. "An efficient scheduling discipline for packet switching networks using earliestdeadlinefirst round robin* ". Computer Communications and Networks, pp 5-10, 2003.
- [8] El Korbi I, Saidane, L.A. "Analytical Model of the Earliest Deadline First Policy over 802.11". The Fourth Advanced International Conference on Telecommunications, pp 180-187, 2008.

- [9] Hanada M, Nakazato H. "ORC-GPS: Output Rate Controlled Scheduling Policy for Delay Guarantees". IEEE International Conference on Communications. ICC '07. pp 530-535, Junio 2007.
- [10] Hanada, M, Nakazato, H, Watanabe, H. "Output Rate-Controlled Generalized Processor Sharing: End-to-end delay bounds calculation". TENCON 2010 - 2010 IEEE Region 10 Conference, pp 779-784, 2010.
- [11] Sunggon Kim, Ju Lee, Dan Sung. "Bandwidth allocation for fluid input generalized processor sharing servers". IEEE Transactions on communications. Vol 58, issue 5, pp 1514-1523, 2010.
- [12] Torabzadeh M, Yusheng Ji. "MIMO Packet-by-Packet Generalized Processor Sharing: MPGPS". International Conference on Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. pp 9-12, 2007.
- [13] Fair Scheduling in Wireless Packet Networks Songwu Lu, Student Member, IEEE, Vaduvur Bharghavan, and R. Srikant, Member IEEE.
- [14] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm," in Proc. ACM SIGCOMM'89, pp. 1-12.
- [15] T. S. Eugene Ng, I. Stoica, and H. Zhang, "Packet Fair Queueing Algorithms for Wireless Networks with Location-Dependent Errors," in Proc. INFOCOM'98, Mar. 1998, pp. 1103-1111.
- [16] J. Gomez, A. T. Campbell, and H. Morikawa, "The Havana Framework for Supporting Application and Channel Dependent QOS in Wireless Networks," in Proc. Seventh International Conference on Network Protocols, 1999.
- [17] M. Shreedhar and G. Varghese. Efficient Fair Queueing using Deficit Round Robin. In Proceedings of ACM SIGCOMM, Berkeley, California, 1995.
- [18] S. Borst, "User-level performance of channel-ware scheduling schemes in wireless data networks," IEEE INFOCOM, Mar. 2003, vol. 1, pp. 321-331.
- [19] A. Jalali, R. Padovani, and R. Pankaj, "Data throughput of CDMA-HDR a high efficiency-high data rate personal communication wireless system," in Proc. IEEE VTC, May 2000, pp. 1854-1858.
- [20] Melander, B, Bjorkman, M, Gunningberg, P, "First-come-first-served packet dispersion and implications for TCP". Global Telecommunications Conference. GLOBECOM 2002. IEEE, vol 3, pp 2170-2174.
- [21] Loher, U, "Efficiency of first-come-first-served algorithms". Information Theory, 1998. Proceedings. 1998 IEEE International Symposium.
- [22] Chaskar, H.M, Madhow, U, "Fair scheduling with tunable latency: a round robin approach". Global Telecommunications Conference, 1999. GLOBECOM '99. Vol 2, pp 1328 - 1333.
- [23] Do V.L, Yun K.Y, "An efficient frame-based scheduling algorithm: credit round robin". High Performance Switching and Routing, 2003, HPSR. Workshop, pp 103-110.
- [24] Oki E, Zhigang Jing, Rojas-Cessa R, Chao H.J, "Concurrent round-robin-based dispatching schemes for Clos-network switches". Networking, IEEE/ACM Transactions. Vol 12, Issue 16, pp 830-844, 2002.
- [25] Schmitt J, Zdarsky F, "A case for simplicity in providing network quality of service: class-based strict priority queueing". Networks, 2004. (ICON 2004). Proceedings. 12th IEEE International Conference. Vol 2, pp 809-813, 2004.
- [26] Yong Liu, Weibo Gong, "On fluid queueing system with strict priority". Decision and Control, 2001. Proceedings of the 40th IEEE Conference. Vol 2, pp 1923-1928, 2001.

- [27] Georges J.-P, Divoux, T, Rondeau, E, "StrictPriority versus Weighted Fair Queueing in Switched Ethernet Networks for Time Critical Applications". Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International . Pp 141, 2005.
- [28] Duong, T.Q, Zepernick H.-J, Fiedler M, "Cross-Layer Design for Integrated Mobile Multimedia Networks with StrictPriority Traffic". Wireless Communications and Networking Conference (WCNC), 2010 IEEE. PP 1-6, 2010.
- [29] Lim T. M, Lee B. S, Yeo C. K, "Quantum-Based EarliestDeadlineFirst Scheduling for Multiservices". Multimedia, IEEE Transactions. Vol 9, Issue 1, pp 157-168, 2007.
- [30] Sushu Zhang, Chatha K.S, Konjevod G, "Approximation algorithms for power minimization of earliestdeadlinefirst and rate monotonic schedules". Low Power Electronics and Design (ISLPED), 2007 ACM/IEEE International Symposium. PP 225-230, 2007.
- [31] Byung Kook Kim, Shin K.G, "Scalable hardware earliest-deadline-first scheduler for ATM switching networks". Real-Time Systems Symposium, 1997. Proceedings., The 18th IEEE. PP 210-218, 1997.
- [32] Veciana G, Kesidis G, "Bandwidth allocation for multiple qualities of service using generalizedprocessorssharing". Global Telecommunications Conference, 1994. GLOBECOM '94. Communications: The Global Bridge., IEEE. Vol 3, pp 1550-1554, 1994.
- [33] Fonseca N.L.S, de M Pereira F, Arantes, D.S, "On the performance of generalizedprocessorssharing under long-range dependent traffic". Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE. Vol 3, pp 2508-2512, 2002.
- [34] Oottamakorn C, Mao S, Panwar S.S, "On GeneralizedProcessorSharing With Regulated Multimedia Traffic Flows". Vol 8, Issue 6, pp 1209-1218, 2006.
- [35] Nandita D, Kuri J, Jamadagni H.S, "Optimal call admission control in generalizedprocessorssharing (GPS) schedulers". INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE. Vol 1, pp 468-477, 2001.
- [36] Chia-Sheng Chang, Kwang-Cheng Chen, "Guaranteed quality-of-service wireless medium access bypacket-by-packetgeneralizedprocessorssharing algorithm". Communications, 1998. ICC 98. Conference Record.1998 IEEE International Conference. Vol 1, pp 493-497, 1998.
- [37] Chia-Sheng Chang, Kwang-Cheng Chen, "Guaranteed QoS access bypacket-by-packetgeneralizedprocessorssharing with channel errors and long propagation". Personal, Indoor and Mobile Radio Communications, 1998. The Ninth IEEE International Symposium. Vol 1, pp 248-252, 1998.
- [38] Tartarelli S, Pagano M, Devetsikiotis M, "Efficient estimation of the cell loss probability in a two-buffer PGPS scheduler". Communications, 2000. ICC 2000. 2000 IEEE International Conference. Vol 3, pp 1320-1324, 2000.
- [39] Tae Joon Kim, "NXG01-2: A WeightedFairQueueing with Optimal Rate and Delay Allocation". Global Telecommunications Conference, 2006. GLOBECOM '06. IEEE. Pp 1-4, 2006.
- [40] Ashour M, Tho Le-Ngoc, "Performance of weightedfairqueueing systems with long range dependent traffic inputs". Electrical and Computer Engineering, 2005. Canadian Conference. Pp 1002-1005, 2005.
- [41] McKillen C, Sezer S, "A weightedfairqueueing finishing tag computation architecture and implementation". SOC Conference, 2004. Proceedings. IEEE International. Pp 270-273, 2004.
- [42] Lu J, Robotham R, "On the implementation of weightedfairqueueing in high

- speed networks”. *Electrical and Computer Engineering*, 2004. Canadian Conference. Vol 2, pp 809-813, 2004.
- [43] Al-Sawaai A, Awan I., Fretwell R, “Analysis of the WeightedFair-Queuing System with Two Classes of Customers with Finite Buffer”. *Advanced Information Networking and Applications Workshops*, 2009. WAINA '09. International Conference. Pp 218-223, 2009.
- [44] Ng T.S.E, Stoica I, Zhang H, “Packet fair queueing algorithms for wireless networks with location-dependent errors”. *INFOCOM '98. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE. Vol 3*, pp 1103-1111, 1998.
- [45] Yuan-Cheng Lai, Chang A, Yu-Chin Szu, Shu-Chung Zheng, “A high-performance and fair scheduler for the wireless network with a multi-state channel”. *Emerging Technologies: Frontiers of Mobile and Wireless Communication*, 2004. Proceedings of the IEEE 6th Circuits and Systems Symposium. Vol 1, pp 149-152, 2004.
- [46] Desilva S, Das S.M, “Experimental evaluation of channelstatedependentscheduling in an in-building wireless LAN”. *Computer Communications and Networks*, 1998. Proceedings. 7th International Conference. Pp 414-421, 1998.
- [47] Inoue M, Wu G, Hase Y, “Channels-tatedependent resource scheduling for wireless message transport”. *Vehicular Technology Conference*, 1998. VTC 98. 48th IEEE. Vol 2, pp 1264-1268, 1998
- [48] Gang Cheng, Kai Xu, Ye Tian, Ansari N, “Core-stateless proportionalfair queueing for AF traffic”. *Global Telecommunications Conference*, 2004. GLOBECOM '04. IEEE. Vol 2, pp 732-736, 2004.
- [49] Vukadinovic V, Karlsson G, “Video streaming performance under proportionalfair scheduling”. *Selected Areas in Communications, IEEE Journal*. Vol 28, Issue 3, pp 399-408, 2010.
- [50] Berger L.T, Kolding T.E, Ramiro-Moreno J, Ameigeiras P, Schumacher L, Mogensen P.E, “Interaction of transmit diversity and proportionalfair scheduling”. *Vehicular Technology Conference*, 2003. VTC 2003-Spring. The 57th IEEE Semiannual. Vol 4, pp 2423-2427, 2003.