# Application-aware Scheduling in Multichannel Wireless Networks with Power Control

MINH DUC NGUYEN

KTH Electrical Engineering

Master's Degree Project
Stockholm, Sweden March 2012

XR-EE-LCN 2012:003

# Master Thesis Report

# Scheduling in Multichannel Wireless Networks with Power Control

Duc Nguyen

Supervisor: György Dán

Examiner: György Dán

05 March 2012

Stockholm, Sweden

# Acknowledgments

For the completion of this master thesis, the author would like to show gratitude to:

- Mr. György Dán, Assistant Professor at School of Electrical Engineering at KTH, Royal Institute of Technology for his precious guidance and patience.
- Tuan-Anh Tran, PhD Student at Balliol College at the University of Oxford for his helpful advice and proof-reading this report.
- My family for their endless support.

And to all my friends, without your help and encouragement, I couldn't go this far. Thank you.

# *Abstract*

Scheduling algorithm is the algorithm to allocate system resources among processes and data flows. Joint channel-assignment and workload-based (CAWS) is a recently developed algorithm for scheduling in the downlink of multi-channel wireless systems, such as OFDM. Compared to well known algorithms, CAWS algorithm has been proved to throughput optimal with flow-level dynamics.

In this master thesis project, we design a system that accounts for power control and for the characteristics of common radio channels. We evaluate the efficiency of the algorithm under a diverse set of conditions. We also do analysis of CAWS algorithm under different traffic density.

**Keywords**: scheduling algorithm, channel-assignment and workload-based

# Table of Contents

# *List of Figures*

# *List of Abbreviations*

| | |
|---|---|
| CAWS | channel-assignment and workload-based |
| FFQ | fluid fair queuing |
| ISI | inter-symbol interference |
| LOS | light of sight |
| MCM | multi-carrier modulation |
| OFDM | orthogonal frequency-division multiplexing |
| SCFQ | self-clocked fair queuing |
| TDM | time division multiplexing |
| WFQ | weight fair queuing |
| $WF^2Q$ | worst-case fair weighted fair queuing |

# Chapter 1 - Introduction

## 1.1 Overview

Generally, resources, which can be abundant at the moment, are always limited in any system. Whether capital, machinery, or human labor, the system must be executed with stability and fairness. Hence how to distribute those resources, a.k.a. scheduling, is always an important task.

Each technical field has its own set of scheduling and wireless communications is not an exception. In wireless communications, the scarcity of radio resources is known for a long time. The need for efficient use of spectrum has lead to multi-channel techniques, such as OFDM. This creates more challenges for scheduling, especially if the scheduler should be aware of the traffic flows.

And under the consideration of arrival and departure of flows, a recently proposed algorithm has been mathematically proved to be throughput optimal for multi-channel wireless networks. That is the Hybrid joint channel-assignment and workload-based (CAWS) scheduling [8].

## 1.2 Goals

This master thesis aims to examine the Hybrid CAWS algorithm's efficiency under cross-layer conditions, which are power control and channel correlation. Those conditions are carefully studied with various simulation scenarios. This thesis also does comparisons between Hybrid CAWS and a well known algorithm (MaxWeight) to verify the improvement of network performance in term of blocking probability, number of flows, and file-transfer delay.

## 1.3 Method

To verify the efficiency of the algorithm, simulation approach has been chosen for this thesis work. A system based on the Hybrid CAWS algorithm has been built in simulation and tested in various scenarios to evaluate the improvement of network performance due to the superiority of the algorithm.

The system and all scenarios are designed and implemented on MATLAB, a familiar numerical computing environment.

## 1.4 Report Structure

The remainder of this report is organized as follows. Chapter 2 contains necessary background understanding for the subject and the evaluating conditions. Chapter 3 explains the design of scenarios and the corresponding algorithms. Chapter 4 gives detailed configurations and investigated behaviors. Chapter 5 presents simulation results and analysis in depth. Chapter 6 concludes the thesis and discusses some interesting thought for further studies.

# Chapter 2 - Background

This chapter presents essential understanding for scheduling algorithms in wireless networks. Both MaxWeight, the widely-used reference algorithm, and Hybrid CAWS, the examining algorithm, are explained. This chapter also gives proper information for those evaluating scenarios, which are channel model and power control.

## 2.1 Channel Model

To efficiently utilize the resources of any communications system, it is critical for researchers and engineers to fully understand the characteristics of the network links [12]. In the case of wireless networks the links are radio signals; and the transmitted signal will differ at receiving endpoint due to multiple impairments.

The attenuation is the reduction of the signal's strength as a function of the transmission distance. The attenuation is also frequency dependent, which leads to distortion. Apart from attenuation, the signal can be subject to interference from signals emitted in the surrounding of the radio receiver, often referred to as noise. Multipath interference introduces when signal travels different ways due to obstacles. And refraction can happen in propagation through atmosphere.

Among all those technical impairments, the most challenging issue could be multipath interference (a.k.a. multipath fading). Due to environment obstacles, multiple waves of the signal can travel different ways (direct light of sight – LOS, reflection, scattering, or diffraction) and reach the receiver at different strengths and phases. Depending on the differences in phases of received signals, there is reinforcement or cancellation effect; [12] so the compound signal at the receiver can be stronger or weaker than the transmitted signal.

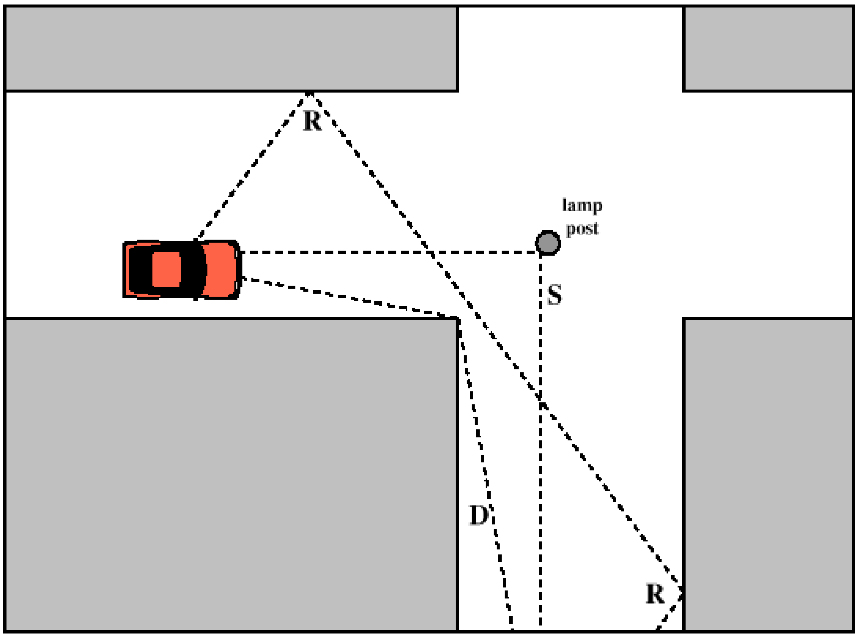


Figure 2-1 Multipath Interference [12]

To design a wireless communication system, the engineer must estimate the effects of all those impairments, especially the effects of multipath interference (fading), hence establish model of multipath fading channel. He also needs to take into account important characteristics of a wireless channel.

In wireless communication, the signal can be propagated in multiple channels, and one main characteristic of multipath propagation is time spread. As there are multiple paths, signals travel through those paths will arrive the receiver at different time and have different strength. Another main characteristic of a wireless channel is time variant. As the environment changes over time, the multiple waves of signals that arrive at the receiver are different (including differences in strength, arrival time, and number of waves). The changes are very difficult to predict, and hence, time-variant channel models are often approached statistically.

### 2.1.1 Multipath Channel Model

There are several probability distributions that can be used to model multipath channel. When there are a lot of paths between the transmitter and receiver and no dominant part (the direct LOS path), the Rayleigh distribution can be used to model the channel. If a dominant path (which can heavily affect the channel characteristics) exists, the Rician distribution can be used.

Also, there are alternatives for Rayleigh distribution, such as Nakagami-m distribution, which is more complex but also more flexible [12].

### 2.1.2 Multi-Carrier Model

Due to the characteristic of multipath propagation, signal waves travel at different lengths, resulting in different arrival times at receiver. This delay causes part of a given signal symbol to spread into the subsequent symbols, hence interfering with those subsequent symbols. Additional distortion of amplitude and phase happened on the travel also contributes to the interference. This form of distortion is called inter-symbol interference (ISI), which is common in multipath propagation [11].

To avoid inter-symbol interference (ISI) and utilize bandwidth resource, a channel is often divided into smaller sub-channels. For the same data, the data rate transmitted in each sub-channel is much lower. Hence a guard interval can be inserted to avoid ISI. That is the principle of Multi-Carrier Modulation (MCM) [21].

MCM [21], especially orthogonal frequency-division multiplexing (OFDM), a special form with densely spaced sub-carriers with overlapped spectrum, is widely used in wireless communications.

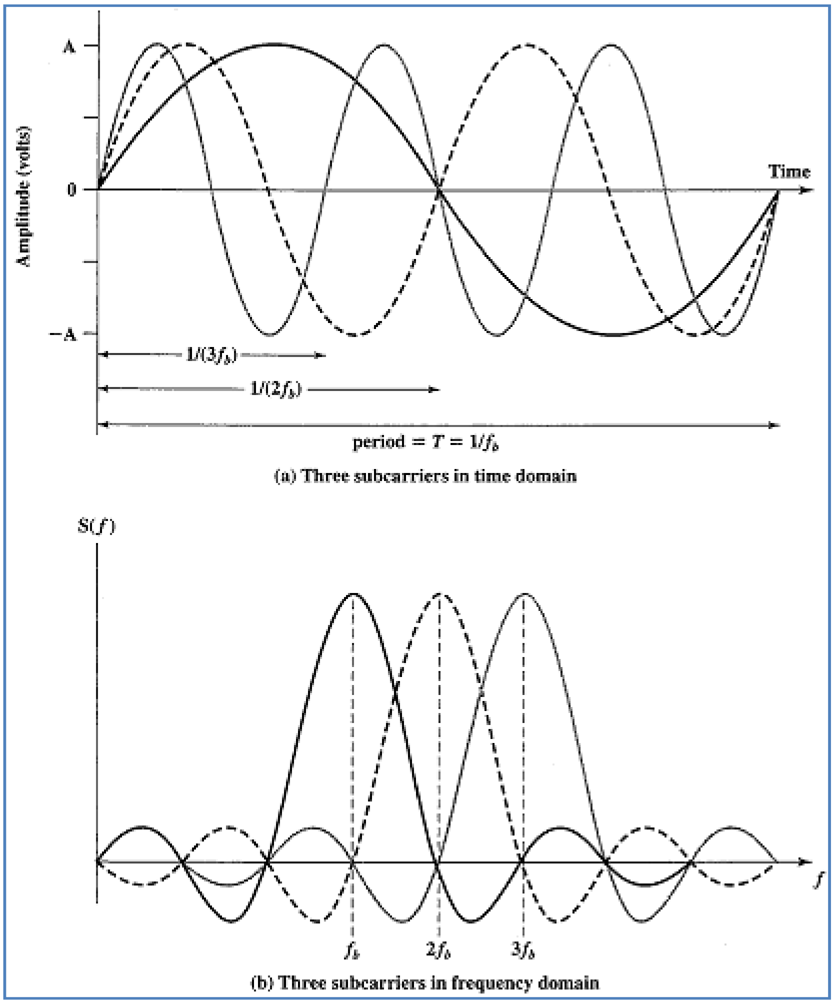


**Figure 2-2 OFDM [11]**

In OFDM, though sub-carriers are overlapped, they are selected to be orthogonal, or to not interfere with each other. As can be seen in figure Figure 2-2 OFDM [11], as subcarriers' period are selected to be inversed to multiple of a base frequency ($f_b$), they don't interfere with each other because when one subcarrier has the highest signal strength, the others signal strength will be zero.

Multi-carrier models can be approached by considering multipath fading channel model for sub-carriers. Each sub-carrier is a Rician or Rayleigh (or Nakagami-m) fading channel, and the series of those sub-carriers can be modeled as mutually dependent fading channels with correlation [21].

Mathematically, a multi-carrier channel H is a size-N complex amplitude matrix of N sub-carriers. Each element of the matrix H is a complex correlated amplitude variable at different sub-carrier ($i = 0,1, \dots, N-1$).

$$H_i = r_i e^{jf_i}$$

Vector H can be generated through a linear operation on an initial vector G:

$$H = A\, G$$

G is an initial vector with N independent and identically distributed complex Gaussian components. A is an appropriate N by N matrix to create a jointly Gaussian with N correlated components.

Though vector H is random, its correlation matrix is not.

$$R_H = \begin{bmatrix} 1 & \frac{1-j\sqrt{C}}{1+C} & \cdots & . & \frac{1-j(N-1)\sqrt{C}}{1+(N-1)^2C} \\ \frac{1+j\sqrt{C}}{1+C} & 1 & & . & . \\ \vdots & & \ddots & & \vdots \\ . & . & & 1 & \frac{1-j\sqrt{C}}{1+C} \\ \frac{1+j(N-1)\sqrt{C}}{1+(N-1)^2C} & . & \cdots & \frac{1+j\sqrt{C}}{1+C} & 1 \end{bmatrix}$$

$$C = 4\Pi^2 \left(\frac{T_{rms}}{T_S}\right)^2$$

With $T_{rms}$ is time spread, and $T_S$ is symbol duration (OFDM frame duration).

Coefficients of the correlation matrix can be written as:

$$[R_H(C)] = a_{mn}(C)$$

$$a_{mn}(C) = \frac{1 + jC(m-n)}{1 + C^2(m-n)^2}, \quad m, n = 1,2, \dots, N$$

From there, matrix A can be calculated through the correlation matrix

$$A = E\sqrt{D}$$

With $E$ is the eigenvectors of $R_H$ and $D$ is the eigenvalues of the correlation matrix.

## 2.2 Power Control

Power control is another essential component of wireless communications. Although the communication's performance depends greatly on channel condition, such as multipath fading and ISI, it can be influenced by the adaptation of signal transmitting, i.e. power control. Therefore researching power allocation plays an important part in channel modeling.

### 2.2.1 Water-filling Algorithm

Consider the communication channel model:

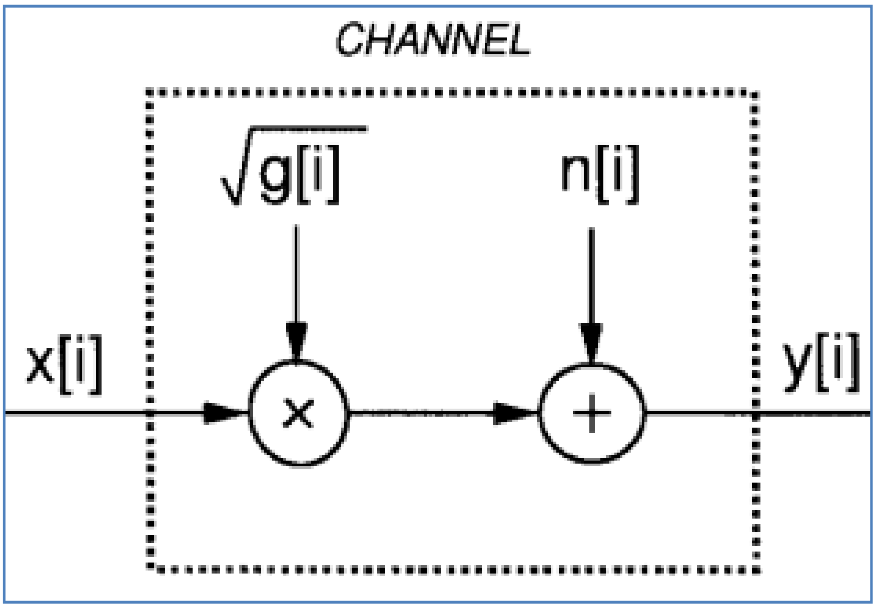


**Figure 2-3 Communication channel model [3]**

The input signal $x[i]$ will be influenced by the channel, resulting as the output signal $y[i]$ at the receiver. The model can be formulated as [14]:

$$y(i) = \sqrt{g(i)} * x(i) + n(i)$$

In the expression above, $x(i)$ and $y(i)$ are input and output signals respectively with the discrete time index $i$. Additive White Gaussian noise (AWGN) $n(i)$ with a constant variance $\sigma^2$ and multiplicative channel fading coefficient $\sqrt{g(i)}$ are environment factors.

If a communication channel with bandwidth B subject to AWGN of power N is given average transmitting power S, the Shannon–Hartley theorem proves the channel capacity C:

$$C = B * \log_2(1 + \frac{S}{N})$$

Notations allocated power on channel fading stage $g_k$ as $S_k$. If channel fading statistics are confined to limited values $g_1, g_2, \dots g_m$, with probabilities $p_1, p_2, \dots p_m$, the data rate maximization problem can be expressed as:

$$max_{S_k} \sum_{k=1}^{m} p_k \log_2\left(1 + \frac{S_k g_k}{\sigma^2}\right)$$

$$subject\ to \sum_{k=1}^{m} p_k S_k \leq S$$

$$S_k \geq 0$$

The optimal solution for the above problem is the well known water-filling algorithm.

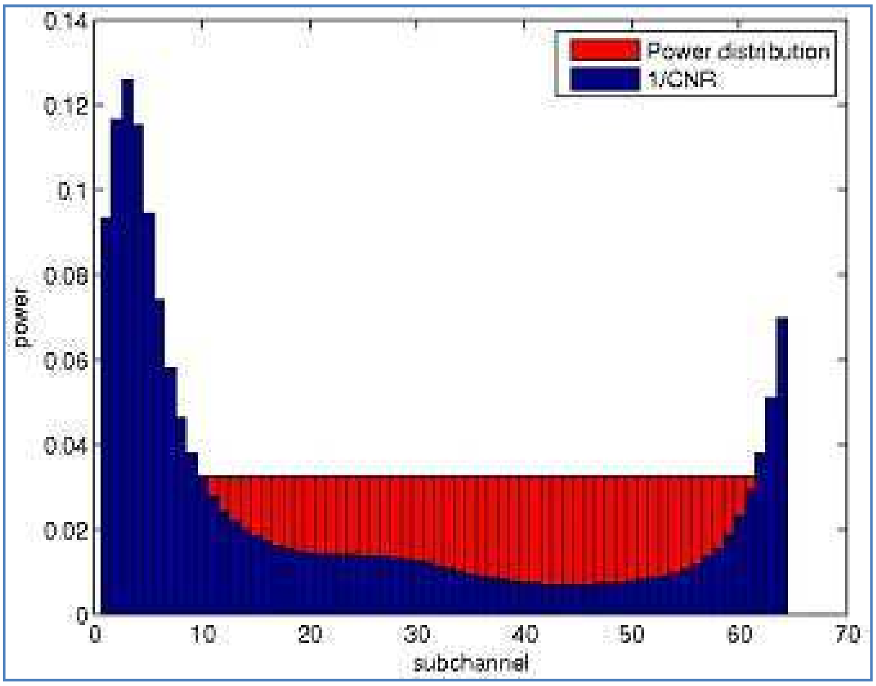


Figure 2-4 Water-filling algorithm [19]

According to the algorithm, the power will be "poured" inversed proportional with the channel strength, similar to the case of water [19]. The numerical algorithm also has an iterative approach [15]. And for the specific OFDM system, there are also various studies of this algorithm [5][7].

### 2.2.2 Constant Power Water-filling Algorithm

Though the water-filling algorithm is optimal, the logarithmic function is complex to execute and insensitive to exact power allocation; hence it is not very effective. An alternative to the exact solution to the water-filling problem is the so called constant power water-filling, which is significantly simpler, but performs close to optimal [8].

Mathematically, constant power water-filling algorithm does not solve the optimal issue directly but approximates it using Lagrangian to calculate lower bound. Since logarithmic function is only sensitive to small value, constant power water-filling tries to find the cut-off point to allocate equal power or not.

$$S_k = \begin{cases} S_0 & if\ p_k \geq p_0 \\ 0 & if\ p_k < p_0 \end{cases}$$

Denotes $m^*$ as number of channel states with positive power and $b_k$ as number of bits allocated in a sub-channel. Solving Lagrangian gives the lower bound in duality gap $\Gamma$, which is the difference between primal value and approximated value:

$$\Gamma \leq \frac{1}{\ln 2}\sum_{k=1}^{m^*} p_k * \left(\frac{\frac{\sigma^2}{p_k}}{S_0 + \frac{\sigma^2}{p_k}}\right) \leq \frac{1}{ln2}\sum_{k=1}^{m^*} p_k 2^{-b_k}$$

To obtain the smallest duality gap, the largest value of $m^*$ is required. A strategy has been deduced:

- Assume channel gains are order: $p_1 \geq p_2 \ldots \geq p_m$ and set $m^* = m - 1$
- Compute $S_0 = S / \sum_{k=1}^{m^*} p_k$
- If $\frac{\sigma^2}{p_{m^*+1}} \geq S_0 + \frac{\sigma^2}{p_1}$, set $m^* = m - 1$ and repeat previous step. If not, set $m^* = m + 1$ and move to next step.
- Compute $b_k = \log\left(1 + \frac{S_0 p_k}{\sigma^2}\right)$ for $k = \overline{1..m^*}$

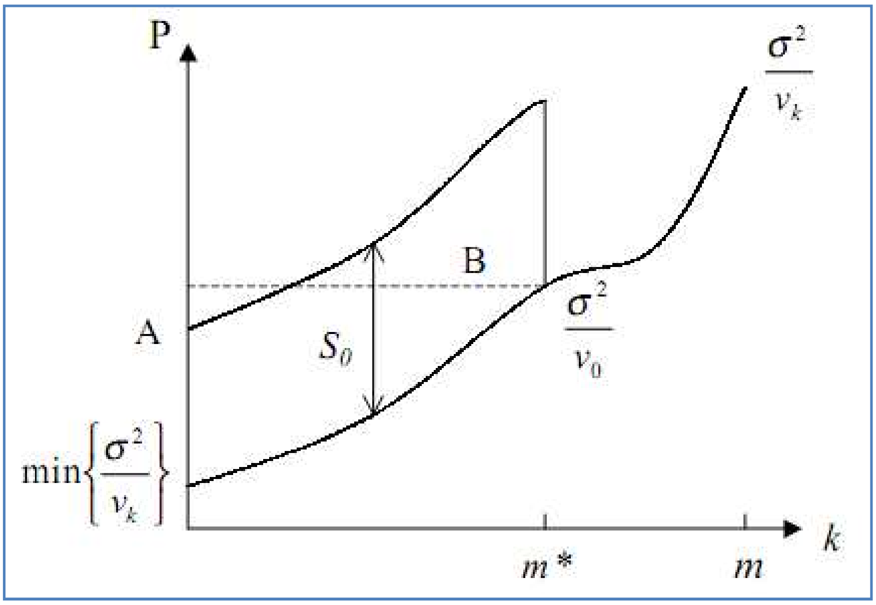


**Figure 2-5 Constant Power Allocation [8]**

As illustrated in Figure 2-5, as long as level A is lower than level B, the attained data rate $R = \sum_{k=1}^{m^*} p_k b_k$ is sub optimal with proved lower bound:

$$\Gamma \leq \frac{1}{ln2} \sum_{k=1}^{m*} p_k 2^{-b_k}$$

Moreover, the procedure is very low complexity as in each step it requires a division instead of a logarithmic operation [14]. Those operations are only carried out when the cut-off point is solved.

## 2.3 Scheduling

Scheduling algorithm is the method to allocate system resources (bandwidth, promptness, and buffer space) among processes or data flows. It is usually executed to maintain system's stability, to achieve fairness between flows and/or to guarantee performance of service. Though there are a lot of types of schedulers, they are always needed to be efficient, protective, flexible and simple [16].

### 2.3.1 Work Conserving Scheduling

Scheduling can be categorized into two types: work conserving scheduling and non work conserving scheduling. In work conserving scheduling, the scheduler is never idle when there is data in queue. There are several remarkable types of work conserving scheduling [16].

Virtual clock scheduling imitates the time division multiplexing (TDM) system [17]. In virtual clock, each packet is assigned a virtual transmission time, based on the arrival traffic pattern and the connection reservation, similar to the time in TDM systems. Packets then are sent according to the order of virtual transmission times.

Weight fair queuing (WFQ) is another scheduling that approximates the fluid fair queuing (FFQ) [10]. In WFQ, if transmission is ready for a time slot, the scheduler will choose the packet (among queued packets) that would complete transmission first in the equivalent FFQ.

Worst-case fair weighted fair queuing ($WF^2Q$) is a similar scheduling [1]. However, it only selects a packet in those that have been serving in the equivalent FFQ system and that would complete transmission first.

Self-clocked fair queuing (SCFQ) tries to estimate the system's virtual time in WFQ with the virtual time of currently serviced packet [4]. Complexity is reduced in this SCFQ but inaccuracy is also introduced.

### 2.3.2 Non-work Conserving Scheduling

In non-work conserving scheduling, eligibility time is assigned to each packet, and the scheduler can be idle (even with waiting packets) if there is no packet with eligible time. There are also several remarkable types of non-work conserving scheduling algorithms, including jitter earliest-due-date, stop-and-go, hierarchical round robin, and rate-controlled static priority [16].

Compare to work conserving scheduling, non-work conserving scheduling has higher delay due to idle time, hence also lower overall throughput. However, it can guarantee end-to-end delay bound, and the bound can be in the whole network environment than in a single node.

### 2.3.3 MaxWeight Scheduling

MaxWeight is a work conserving scheduling that guarantees network stability as well as maximum throughput [13]. In this scheduling, the main policy is to simultaneously choose and serve a subset of queues at maximum 'weight'. The weight of a queue is the product of feasible instantaneous service rate for a queue (if it is used) and the current backlog of that queue. Obviously, there are constraints on those scheduled queues, such as interference conditions.

$$Select\ f^* \in \arg\max F_f(t) R_{if}(t)$$

$$Transmit\ \min\{F_f(t), R_{if}(t)\}$$

Denote $f^*$ as the selecting flow in the set of flows; $F_f(t)$ as the number of packets (backlog) of flow $f$ at time $t$; and $R_{if}(t)$ as the number of packets that can be served (feasible instantaneous service rate) by channel $i$ at time $t$.

As can be seen, one of advantageous aspect of MaxWeight is its simple information input. Only current backlog and feasible instantaneous service are required; traffic parameters or rate distributions are not.

In various wireless communications scenarios, MaxWeight has shown to be an effective scheduling. Even channel-level or packet-level dynamics, even multi-hop wireless networks, joint congestion control or cross-layer control [2], MaxWeight is always powerful and optimal. However, since this scheduling tends to favor flows with larger backlogs despite their non-optimal service rate, it fails to achieve network stability and maximum throughput under the presence of flow-level dynamics.

### 2.3.4 Channel-assignment and Workload-based Scheduling

This failure motivates the developing new scheduling algorithms that are throughput-optimal for networks with flow-level dynamics. A recently CAWS algorithm has been proposed for multi-channel wireless network [8]. It has been mathematically proved to be optimal in the presence of flow arrivals and departures.

Channel-assignment and workload-based is a scheduling in multi-channel wireless networks concerning dynamic flow arrivals and departures [8]. It can be divided into two distinctive parts:

- Channel-assignment: to the calculate channel assignment vector $h_f(t)$. Each component $h_{if}(t)$ is the number of remaining time slots that flow $f$ will be served on channel $i$ at time $t$ (non-negative integers). This vector can be solved:

  $$OPT_f = min \sum_{i \in M} Q_i(b_f) h_{if}$$

  $$F_f \leq \sum_{i \in M} h_{if} R_{if}^{\max}$$

  At small scale, exhaustive search is feasible with a few for loops. However, when the number of channel increases, another approach must be considered, such as a mixed integer linear program.

- Workload-based scheduling: to actual select flow and transmit.

$$Select\ f \in \arg R_{if}(t) = R_{if}^{\max}\ and\ h_{if}(t) > 0$$

$$Transmit\ \min\{F_f(t), R_{if}(t)\}: \begin{cases} F_f(t): h_{if}(t) = 0 \\ R_{if}(t): h_{if}(t) = h_{if}(t) - 1 \end{cases}$$

Through simulation, CAWS has been proved to have better performance than MaxWeight algorithms and to retain stability in cases that MaxWeight cannot.

Though being better than MaxWeight, CAWS is not free from weakness. One of the weaknesses is that the algorithm has to wait for a complete file before processing. This introduces both large waiting time and transfer delay for large files. Another weakness of CAWS is the impractical assumption of knowing maximum service rate in advance.

To overcome those two weaknesses, the hybrid CAWS algorithm is also presented to combine the advantages of both MaxWeight and CAWS algorithms for a better performance. Compared to pure CAWS, the hybrid algorithm has two more steps:

- Channel learning: to resolve maximum service rate issue. In learning period $D,$ a pre-defined time slots from flow arrival, the base station keeps measuring channel and define the maximum service rate as the maximum service rate of that time.

$$\tilde{R}_{if}^{\max}(t) = \max\{\tilde{R}_{if}^{\max}(t-1), R_{if}(t)\}$$

- Scheduling choosing: to resolve poor performance at low traffic issue. At each time slot, scheduling (MaxWeight or CAWS) will be selected after backlog comparison

$$\sum_{f \in L(t)} F_f(t) \leq \sum_{f \in S(t)} F_f(t)$$

  Denote $L(t)$ as the set of transient flows, which haven't fully arrived, and $S(t)$ as the set of resident flows, which arrivals are completed. If the inequality is true, CAWS will be used for resident flows. Otherwise, MaxWeight algorithm will be used for resident flows.

# Chapter 3 - Design

This chapter explains the design of scenarios and the corresponding algorithms that are used in MATLAB simulations. Only concepts have been presented. Detail parameters are presented in the next chapter.

## 3.1 System Model

For comparison with previous work, a similar system model to [8] has been used.

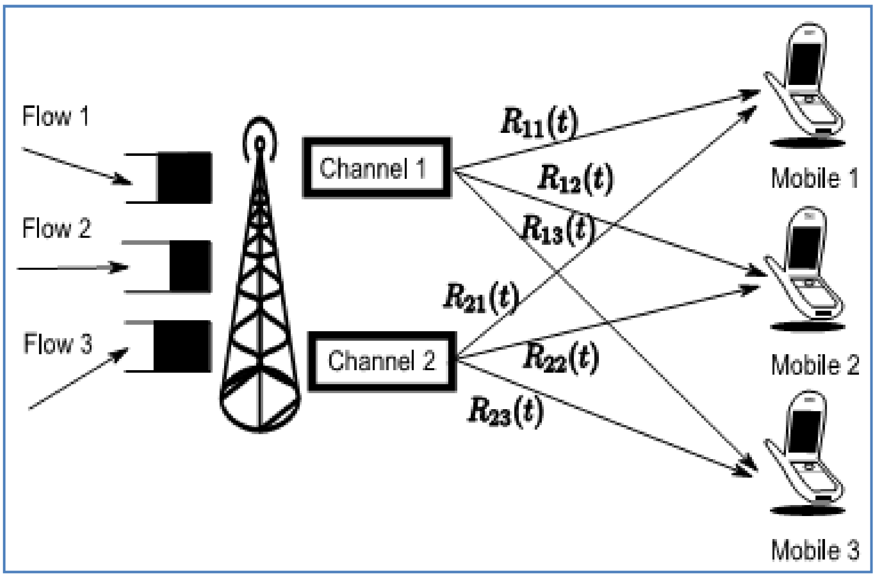


**Figure 3-1 A general system model [8]**

The central network model is a wireless downlink network with a single base station. There are multiple users (mobiles) which join the network to receive a file (flow) from a remote source and leave the network immediately after download completion. Time is divided into slots and frequency bandwidth is divided into sub-channels. At each time slot, a channel can serve only one flow, but that flow can be served by multiple channels simultaneously.

The channel model is described in the next section. Its main item is the instantaneous service rate $R_{if}(t)$, the rate (number of packets) that flow $i$ can be served by channel $f$ at timeslot $t$. This rate has a lower bound of a positive probability to be maximum rate.

$$\Pr\left(R_{if}(t) = \max\left(R_{if}\right)\right) > 0 \;\forall(i, f, t)$$

The traffic model is Poisson distribution for flow arrival rate and Pareto distribution for file size. It is should be mentioned that a flow is also divided into packets, and $F_f$ denotes the number of packets in the file for flow $f$. Based on the last packet arrival, flows also categorized into two types: transient (the last packet has not arrived) and resident (the last packet has arrived). In correspondent, $L(t)$ denotes the set of transient flows at time slot t and $S(t)$ denotes the set of resident flows.

## 3.2 Channel Model

In this thesis, multi-carrier fading channel model in [21] is used.

First, we need to create the channel correlation matrix. This matrix depends on the number of sub-channels and the ratio (delay spread over OFDM frame duration).

Second, we need to create the complex channel autocorrelation matrix.

Last, we need to create correlated jointly Gaussian subcarriers. It can be done by independent random complex amplitude and the complex channel autocorrelation matrix.

## 3.3 Power Control Algorithm

In this thesis, Constant Power Water-filling is the algorithm that is chosen. Basically, the algorithm is to find the largest number of sub-channels with the smallest duality gap [14].

To do that sub-channels must be arranged in the order of channel gain's decrement.

$$\nu_1 \geq \nu_2 \ldots \geq \nu_m$$

Denote $\nu_0$ the cut-off point to distribute constant power $S_0$ to any channel that has higher channel gain $\nu_k \geq \nu_0$. Denote $m^*$ the largest $k$ that satisfies above condition. The algorithm is to find the value of $m^*$.

- Set $m^* = m - 1$
- Compute $S_0 = S / \sum_{k=1}^{m^*} p_k$
- If $\frac{\sigma^2}{v_{m^*+1}} \geq S_0 + \frac{\sigma^2}{v_1}$, decrease $m^*$ one unit and repeat the previous step. Otherwise, increase $m^*$ one unit and go to next step.
- Compute $b_k = \log\left(1 + \frac{S_0 v_k}{\sigma^2}\right)$ for $k = 1, \dots, m^*$.
- Compute $R_{if}(t) = b_i * BW$.

## 3.4 Scheduling Algorithm

In this thesis, there are two scheduling algorithms to be examined: MaxWeight and Hybrid CAWS.

### 3.4.1 MaxWeight scheduling

MaxWeight scheduling is quite simple. In each time slot and at each channel, the base station chooses the flow that has the highest product of its current backlog and its feasible instantaneous service rate. That flow will be served to the correspondent mobile user at the minimum of its backlog and service rate.

$$Select: f^* \in \arg max_{f \in L(t)} F_f(t) * R_{if}(t)$$

$$Transmit: \min \{F_{f^*}(t), R_{if^*}(t)\}$$

MaxWeight scheduling is used to provide comparisons for Hybrid CAWS scheduling.

### 3.4.2 Hybrid CAWS scheduling

Hybrid CAWS scheduling can be divided into 4 steps.

First step is flow learning. The base station records new arrival flows (including the instantaneous service rate of each channel for those flows) and receives arrival packets for existing flows.

Second step is channel learning. The base station updates new instantaneous service rates for all channels and flows. It also checks to update maximum service rates if the flow is still in learning period.

$$If\ t \leq b_f + D: R_{if}^{max}(t) = \max \{R_{if}^{max}(t-1), R_{if}(t)\}$$

Third step is channel assignment. Whenever there is an update in maximum service rate, the base station recalculates the assignment vector for the related flow by solving optimization problem:

$$OPT_f(t) = min\Sigma_{i\in M} Q_i\big(b_f\big)h_{if}(t)$$

$$subject\ to: F_f(t) \leq \Sigma_{i\in M} h_{if}(t)R_{if}^{max}(t)$$

If the number of channels is few, exhaustive search can be used by a stack of loops. However, recursive function is a much scalable resolve.

Final step is scheduling choice. Scheduling is selected after the comparison of the total backlog of transient flows and resident flows at current time slot.

$$Compare: \Sigma_{f\in L(t)} F_f(t)\ vs.\ \Sigma_{f\in S(t)} F_f(t)$$

- Workload-based scheduling is chosen if the total backlog of resident flows is not smaller than that of transient flows. For each channel, a flow is selected when its maximum service rate at that channel is not higher than the current service rate and the assignment vector of the flow at that channel is not empty.

  $$Select: f \mid R_{if}^{max}(t) \leq R_{if}(t)\ \&\ h_{if}(t) \neq 0$$

  Similar to the case in MaxWeight algorithm, the flow will be served to the correspondent mobile user at the minimum of its backlog and service rate. After serving, the vector assignment of that flow is deducted one.

  Several flows can satisfy selection conditions. In that case, the base station could select the flow based on some parameters, such as arrival time. However, in this project design, ties are broken uniformly.

- MaxWeight scheduling is chosen if the total backlog of resident flows is smaller than that of transient flows. Later procedure is mentioned in previous section.

In short, hybrid CAWS is the focus of the project. It will be examined carefully under the effect of power control algorithm and channel correlation.

# Chapter 4 - Simulation

This chapter gives the detail configurations that are used in simulations and the information for actual scenarios.

## 4.1 Parameters

There are two types of system model: ideal base station and practical base station. In the practical model, the base station can handle maximum of 40 flows simultaneously while there is no restriction in the ideal model.

The arrival rate varies from $\lambda = 0.05$ to $\lambda = 0.5$ with the step of 0.05 to show variation results. File size (number of packets of a flow) follows Pareto distribution with minimum possible value $x_m = 50$ and decay factor $\alpha = 2$.

In the channel model, there are 5 sub-channels that have been used. They are either independent or correlated with the ratio (delay spread relative to frame duration) of 0.01, depending on the scenarios.

In power control, channel gains are divided into 10 steps relatively and equally. Noise power is standardized to 1, and signal power is 0.1.

In scheduling algorithms, there is only one noticeable parameter: learning period has been set to 20 time slots.

The arrival rate and file size are chosen similar to ref. [8] for comparisons. High correlated ratio is chosen to spot differences easier. And the signal power is chosen for the power control algorithm to work (if signal power is higher, most of sub-channels will satisfy cut-off point, resulting in almost equal power distribution).

## 4.2 Scenarios

There are two main sets of scenarios to examine the effect of power control and channel correlation to scheduling algorithms.

### 4.2.1 Effect of power control

The first simulation scenario is to examine the effect of power control on both MaxWeight algorithm and Hybrid CAWS algorithm. There are four sets of simulation for these two algorithms, both with ideal base station and practical base station. The practical simulation sets show the blocking probability of algorithms with different arrival rates. The ideal sets can show two other properties: the average number of flows in the system and the average delay of served flows.

### 4.2.2 Effect of channel correlation

The second simulation scenario is to examine the effect of correlation to only Hybrid CAWS algorithm. The identical parameters have been examined under the comparison of three sets: scheduling under ideal environment, scheduling under power control, and scheduling under both power control and correlation.

# Chapter 5 - Results

This chapter presents simulation results and intuitive analysis. In both scenarios, simulation period of 3000 time slots has given concrete results.

## 5.1 Scenario 1

The first scenario shows the weight of constant power water-filling algorithm to the two scheduling algorithms. Practical base station has been simulated to give the average blocking probability.

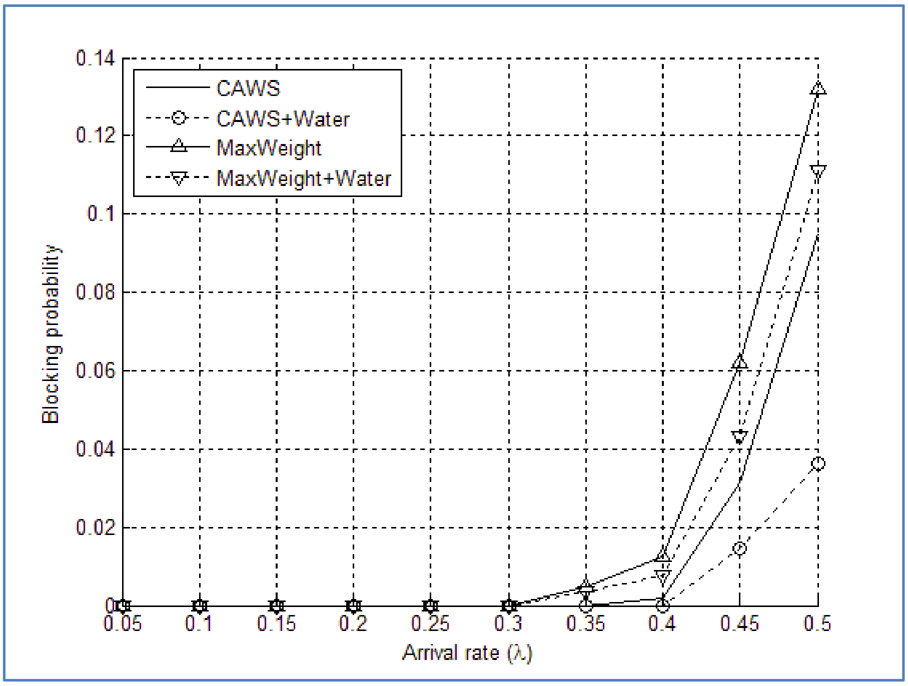


**Figure 5-1 Scenario 1: blocking probability**

Figure 5-1 confirms the stability of Hybrid CAWS algorithm over MaxWeight algorithm. It also shows that the power control algorithm has positive effect on Hybrid CAWS scheduling. As can be seen from the right part of the figure, the Hybrid CAWS scheduling under power control has less than 50% of flow blocking probability compared to the scheduling with power distributing equally. The blocking probability has been reduced to less than 4% in the highest arrival rate scenario.

This can be explained by the strength of the power control algorithm itself. With constant power water-filling algorithm, more power is distributed to better sub-channels (which has higher channel gain). As data rate is related to the power, the sum of data rates in all sub-channels increases. As a result, more flows can be served and less will be denied.

Other simulations with limitless base station also strengthen the above explanation.

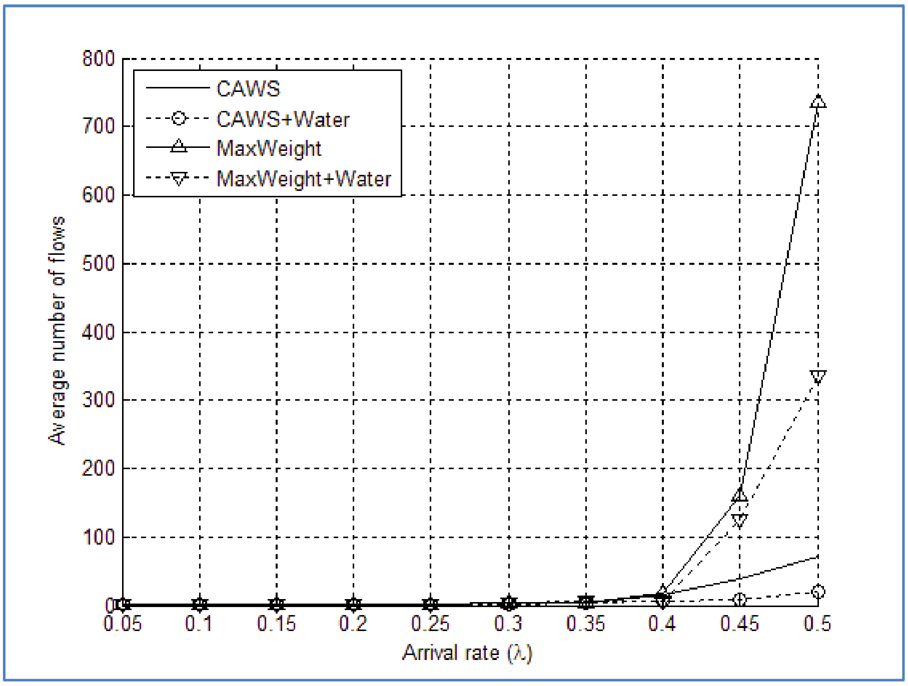


**Figure 5-2 Scenario 1: average number of flows**

Figure 5-2 shows that hybrid CAWS algorithm works very well with power control algorithm. The average number of flows reduces significantly with that better power distribution. In the highest load case, the average number of flows only reaches 19.

It is noticeable that the average number of flows in MaxWeight scenario is much higher than the parameter of practical simulation sets (maximum of 40 flows simultaneously). That explains the drastic increase of blocking probability in the blocking probability figure.

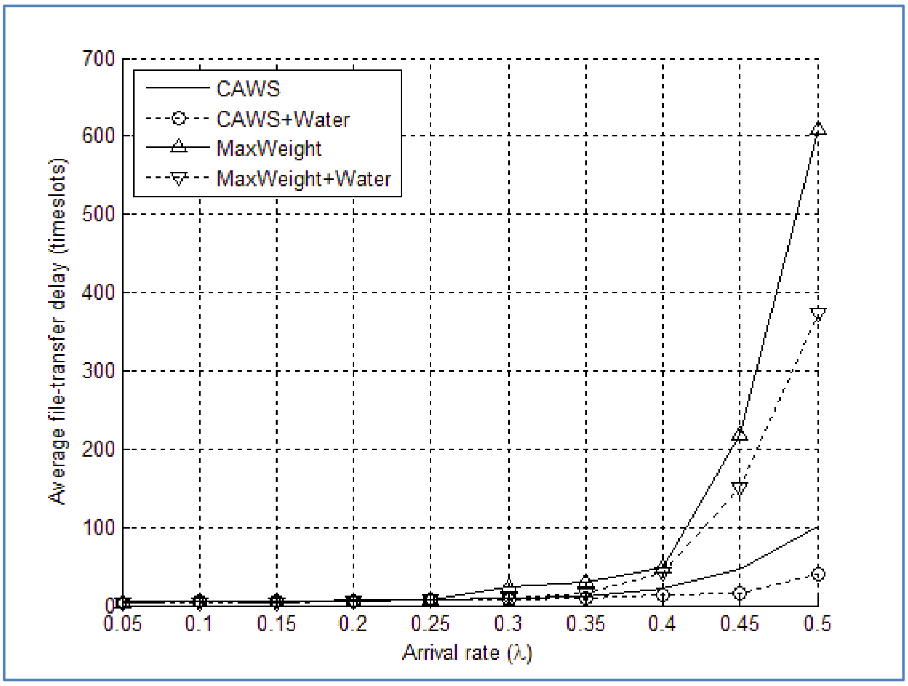


**Figure 5-3 Scenario 1: average delay**

Figure 5-3 also strengthens the hypothesis. The examining algorithm cooperates with power control algorithm to lessen the average delay.

## 5.2 Scenario 2

The second scenario shows the weight of correlation to Hybrid CAWS scheduling.

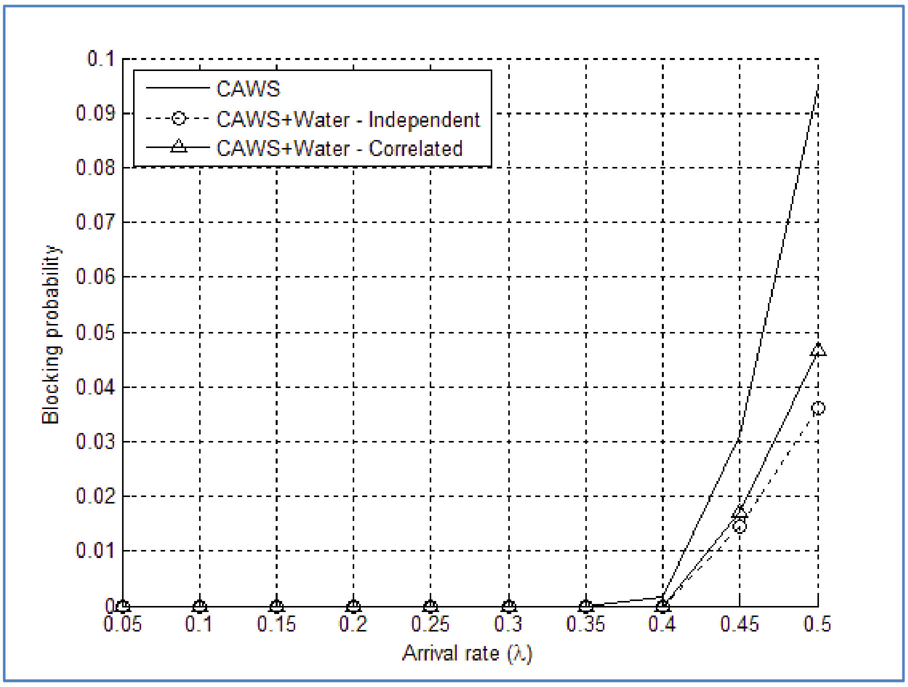


**Figure 5-4 Scenario 2: blocking probability**

Figure 5-4 shows the negative effect of channel correlation. Although the blocking probability of the scenario with correlation is still better than that of the scenario without power control or channel correlation, the result has been worse than the scenario with only power control algorithm. The correlation has increased the blocking probability of approximately 10%.

This undesired consequence can be explained as follow. In correlation channel model, all sub-channels are highly correlated. The neighboring sub-channels only have small differences in channel gain. Hence all sub-channels gain will have quite similar values and power control algorithm will be less effective. Power will be distributed equally to most sub-channels unless that channel has sudden drop value. This will negate part of the use of power control.

With ideal base station, simulation also gives similar results.

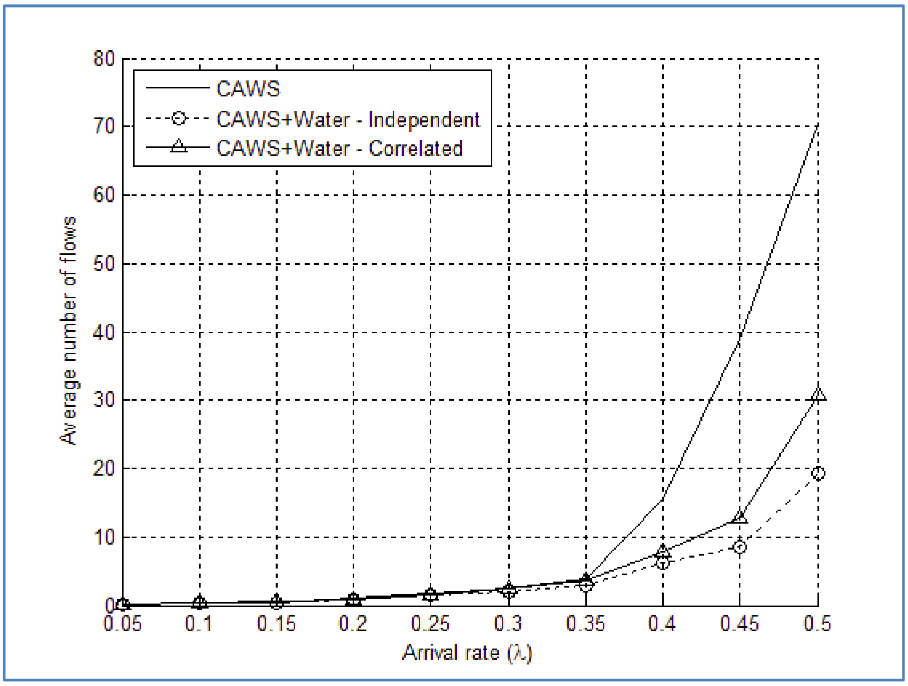


**Figure 5-5 Scenario 2: average number of flows**

Figure 5-5 has shown the same effect of correlation. Correlation negates part of the effect of constant power water-filling algorithm, though the result is still better than without those models. There is only a small note that with the arrival rate of 0.5, the second and third simulations have the average number of flows less than 40, which can explain the stability of those models in Figure 5-4.

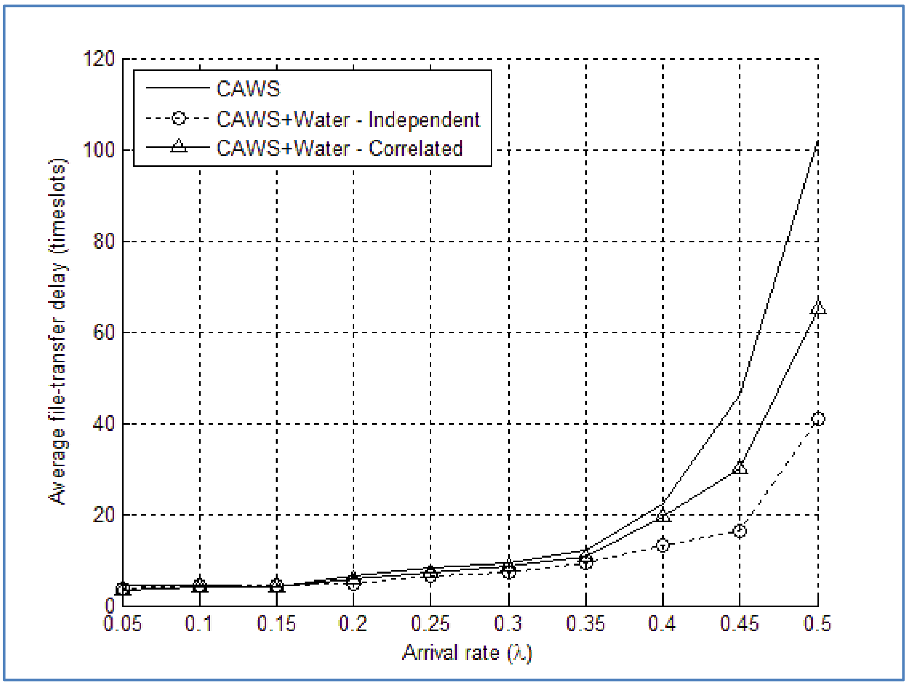


**Figure 5-6 Scenario 2: average delay**

Figure 5-6 also reinforces the assumption Correlation has lessened the impact of the constant power water-filling algorithm, hence slightly increase the average delay.

# Chapter 6 - Conclusions

## 6.1 Achieved Goals

This master thesis has documented an approach for a newly proposed scheduling algorithm. Related work in several fields, including scheduling algorithm, power control and channel model has been studied and evaluated.

MATLAB simulations have been carried in several scenarios to examine the effect of different constraint conditions to scheduling. Final results show the effectiveness of hybrid CAWS scheduling algorithm under practical conditions. Though correlation has undesired affection, good power control algorithm has given CAWS scheduling significant improvement. The simulation also validates the superior of CAWS in achieving stability compared to MaxWeight algorithm.

## 6.2 Limitation

Even though the work has been carried at highest effort, it still has limitations due to both time restriction and understanding limitation.

Due to objective constraint, several parameters are rather small, which can have an impact on the believability of the results. The not very high number of sub-channels causes difficulties in realizing the important of scheduling algorithm. The limited steps of channel gains also make the effect of power control algorithm less clearly.

Due to subjective constraint, only one power control algorithm is considered. Even the channel model is still quite simple, which can be enhanced more.

## 6.3 Future work

One of the improvements is to increase key parameters, including the number of sub-channels and the number of steps in power control. Higher number of sub-channels gives clearer image of scheduling algorithm and higher number of steps strengthens power control algorithm.

Another improvement is to choose a more practical channel model. Since the channel for high data rate transmissions is normally wideband and frequency-selective, an OFDM systems over Nakagami fading channel in [18] can be better suited.

# References


[1]. J. Bennett and H. Zhang, *WF$^2$Q: Worst-case fair weighted fair queuing, July 95*, INFOCOM ’96.

[2]. L. Georgiadis, M.J. Neely, L. Tassiulas, *Resource allocation and cross-layer control in wireless networks (Foundations and Trends in Networking)*, Now Publishers Inc, Apr. 2006.

[3]. A. Goldsmith and P. Varayia, *Capacity of fading channel with channel side information*, IEEE Transactions on Information Theory, vol. 43, no. 6, pp. 1986 – 1992, Nov. 1997.

[4]. S. Golestani, *A self-clocked fair queuing scheme for broadband applications*, Proc IEEE INFOCOM ’94, pp. 636 – 646, Jun. 1994.

[5]. J. Jang and K. Lee, *Transmit Power Adaptation for Multiuser OFDM Systems*, IEEE Journal on Selected Areas in Communications, vol. 21, no. 2, pp. 171 – 178, Feb. 2003.

[6]. Z. Kang, K. Yao, and F. Lorenzelli, *Nakagami-m fading modeling in the frequency domain for OFDM system analysis*, IEEE Communications Letters, vol. 7, no. 10, pp. 484 – 486, Oct. 2003.

[7]. S. Katsturia, J.T. Aslanis, and J.M. Cioffi, *Vector coding for partial response channels*, IEEE Transactions on Information Theory, vol. 36, no. 4, pp. 741 – 762, Jul 1990.

[8]. S. Liu, L. Ying, R. Srikant, *Scheduling in Multichannel Wireless Networks with Flow-Level Dynamics*, Proc. SIGMETRICS 2010, New York City, Jun 2010.

[9]. N. McKeown et al., *Achieving 100% throughput in an input-queued switch*, IEEE Transactions on Communications, vol. 47, no. 8, pp. 1260 – 1267, Aug 1999.

[10]. A. Parekh and R. Gallager, *A generalized processor sharing approach to flow control – The single node case*, Proc. INFOCOM '92, 1992.

[11]. J. Proakis, *Digital Communications*, 4th ed., McGraw-Hill, 2000.

[12]. W. Stallings, *Wireless Communications and Networks*, 2nd ed., NJ Prentice Hall, 2005.

[13]. P. van de Ven, S. Borst, and S. Shneer, *Instability of MaxWeight Scheduling Algorithm*, IEEE INFOCOM 2009, pp. 1701-1709, Apr 2009.

[14]. W. Yu and J.M. Cioffi, *On Constant Power Water-filling*, IEEE International Conference on Communications 2001, vol. 6, pp. 1665 - 1669, Jun. 2001.

[15]. W. Yu et al., *Iterative Water-filling for Gaussian Vector Multiple Access Channels*, IEEE Transactions on Information Theory, vol. 50, no. 1, pp. 145 – 152, Jan 2004.

[16]. H. Zhang, *Service disciplines for guaranteed performance service in packet switching networks,* Proc. IEEE, vol. 83, no. 10, pp. 1374 – 1396, Oct. 1995.

[17]. L. Zhang, *Virtual clock: A new traffic control algorithm for packet switching networks*, Proc. ACM SIGCOMM '90, pp. 19 – 29, Sep. 1990.

[18]. R. Zhang and L. Cai, *Packet-level channel model for wireless OFDM systems*, IEEE Global Telecommunications Conference, 2007, pp. 5215 – 5219, Nov. /Dec. 2007.

[19]. http://www.dsprelated.com/showarticle/119.php

[20]. http://www.mathworks.com/matlabcentral/fileexchange/6990-mixed-integer-lp

[21]. http://www.wirelesscommunication.nl/reference/contents.htm

# Appendix – MATLAB Implementation

## Hybrid CAWS function

```
function y=CAWS(SimulationPeriod,Model,Mode)
% Hybrid joint channel-assignment and workload-based algorithm
% Model: 1-CAWS, 2-CAWS+Water-Independent, 3-CAWS+Water-Correlated
% Mode:  1-Finite, 2-Infinite

% Initialization
NoExperiment    =10;
LearningPeriod  =20;
NoChannel       =5;

% There are three matrixes that store information
% Transient & Resident: size, queue, transient time, resident time,
%                      : compute indicator, first time indicator,
%                      : channel rates, max rates, assignment vector
Transient=[];
Resident =[];
% Record: total {accepted flows, denied flows}
%       : sum of {flows in all timeslots, processed flows, delay}
%       : average {block probability, number of flows, delay}
Record   =zeros(NoExperiment,8);

% Main program
for expt=1:NoExperiment
   for time=1:SimulationPeriod
      disp([expt time]); % For observations
      % Channel state
      RateMatrix=ChannelRate(NoChannel,Model);

      % Flow creation
      [Transient r1 r2]=CreateFlow(Transient,size(Resident,1),...
         NoChannel,0.05*expt,Mode);
      Record(expt,1)=Record(expt,1)+r1;
      Record(expt,2)=Record(expt,2)+r2;

      % Data injection
      [Transient Resident]=InjectData(Transient,Resident);

      % Time increment
```

```
if ~isempty(Transient)
   Transient(:,3)=Transient(:,3)+1;
end
if ~isempty(Resident)
   Resident(:,4)=Resident(:,4)+1;
end

% Scheduling choosing (by comparing total queues)
sumTransient=0;
if ~isempty(Transient)
   sumTransient=sum(Transient(1:end,2));
end
sumResident=0;
if ~isempty(Resident)
   sumResident=sum(Resident(1:end,2));
end

% MaxWeight scheduling
if sumTransient>sumResident
   % Assign channel state
   for channel=1:NoChannel
      Transient=AssignRate(Transient,RateMatrix,channel,...
         NoChannel,LearningPeriod);
   end
   % MaxWeight scheduling for each channel
   for channel=1:NoChannel
      if ~isempty(Transient)
         MW=0;
         flow=0;
         for i=1:size(Transient,1)
            if (Transient(i,2)*Transient(i,6+channel)>MW)
               MW=Transient(i,2)*Transient(i,6+channel);
               flow=i;
            end
         end
         if flow>0
            [Transient r4 r5]=Process(Transient,channel,flow);
            Record(expt,4)=Record(expt,4)+r4;
            Record(expt,5)=Record(expt,5)+r5;
         end
      end
   end
end

% Workload-based scheduling
if ((sumResident>=sumTransient)&&(sumResident>0))
   % Assign channel state
   for channel=1:NoChannel
   Resident=AssignRate(Resident,RateMatrix,channel,...
      NoChannel,LearningPeriod);
   end

   % Compute assignment vector
   Resident=AssignVector(Resident,NoChannel,LearningPeriod);

   % Scheduling for each channel
   for channel=1:NoChannel
      if ~isempty(Resident)
      Temp=[];
         for i=1:size(Resident,1)
```

```
                  if
Resident(i,6+channel)>=…Resident(i,6+NoChannel+channel)
                     if Resident(i,6+2*NoChannel+channel)>0
                        Temp=[Temp i];
                     end
                  end
               end
               if isempty(Temp)
                  flow=randi(size(Resident,1));
                  [Resident r4 r5]=Process(Resident,channel,flow);
                  Record(expt,4)=Record(expt,4)+r4;
                  Record(expt,5)=Record(expt,5)+r5;
               else
                  flow=randi(size(Temp));
                  [Resident r4
r5]=ProcRes(Resident,channel,Temp(flow));
                  Record(expt,4)=Record(expt,4)+r4;
                  Record(expt,5)=Record(expt,5)+r5;
               end
            end
         end
      end

Record(expt,3)=Record(expt,3)+size(Transient,1)+size(Resident,1);
   end
   Record(expt,6)=Record(expt,2)/(Record(expt,1)+Record(expt,2));
   Record(expt,7)=Record(expt,3)/SimulationPeriod;
   Record(expt,8)=Record(expt,5)/Record(expt,4);
   name=strcat('CAWS',int2str(Model+(Mode-1)*3),'.txt');
   save(name,'Record','-ASCII');
end
end
```

## MaxWeight function

```
function y=MW(SimulationPeriod,Model,Mode)
% MaxWeight algorithm
% Model: 1-MW, 2-MW+Water-Independent, 3-MW+Water-Correlated
% Mode:  1-Finite, 2-Infinite

% Initialization
NoExperiment    =10;
LearningPeriod  =20;
NoChannel       =5;

% There are three matrixes that store information
% Transient & Resident: size, queue, transient time, resident time,
%                     : compute indicator, first time indicator,
%                     : channel rates, max rates, assignment vector
Transient=[];
Resident =[];
% Record: total {accepted flows, denied flows}
%       : sum of {flows in all timeslots, processed flows, delay}
%       : average {block probability, number of flows, delay}
Record   =zeros(NoExperiment,8);

% Main program
for expt=1:NoExperiment
   for time=1:SimulationPeriod
      disp([expt time]);
      % Channel state
      RateMatrix=ChannelRate(NoChannel,Model);

      % Flow creation
      [Transient r1 r2]=CreateFlow(Transient,size(Resident,1),...
         NoChannel,0.05*expt,Mode);
      Record(expt,1)=Record(expt,1)+r1;
      Record(expt,2)=Record(expt,2)+r2;

      % Data injection
      [Transient]=InjectDataMW(Transient);

      % Time increment
      if ~isempty(Transient)
         Transient(:,3)=Transient(:,3)+1;
      end

      % Assign channel state
      for channel=1:NoChannel
         Transient=AssignRate(Transient,RateMatrix,channel,...
            NoChannel,LearningPeriod);
      end

      % MaxWeight scheduling for each channel
      for channel=1:NoChannel
         if ~isempty(Transient)
            MW=0;
            flow=0;
            for i=1:size(Transient,1)
               if (Transient(i,2)*Transient(i,6+channel)>MW)
                  MW=Transient(i,2)*Transient(i,6+channel);
                  flow=i;
```

```
                 end
              end
              if flow>0
                 [Transient r4 r5]=Process(Transient,channel,flow);
                 Record(expt,4)=Record(expt,4)+r4;
                 Record(expt,5)=Record(expt,5)+r5;
              end
            end
        end

Record(expt,3)=Record(expt,3)+size(Transient,1)+size(Resident,1);
   end
   Record(expt,6)=Record(expt,2)/(Record(expt,1)+Record(expt,2));
   Record(expt,7)=Record(expt,3)/SimulationPeriod;
   Record(expt,8)=Record(expt,5)/Record(expt,4);
   name=strcat('MW',int2str(Model+(Mode-1)*3),'.txt');
   save(name,'Record','-ASCII');
end
end
```

## Channel Rate function

```
function y = ChannelRate(channel,model)
% To crate channel rates
% Model: 1-Random, 2-Water-Independent, 3-Water+Correlated
% Input : number of channels
% Output: rate for each channel

%{
Water-filling algorithm
Create channel gain vi (uniform distribution is simplest)
Order v1 >= v2 ... >= vm
Step 1: Set m*=m-1
Step 2: Compute S0=S/(p1+p2..+pm*)
Step 3: If sigma^2/v(m*+1) >= S0+sigma^2/v1, set m*=m*-1, repeat step
2
        Otherwise, set m*=m+1, go to the next step
Step 4: Compute bk=log(1+S0*vk/sigma^2) for k=1,...,m*
Then R_if(t)=pk*bk
%}

S      =0.1; % Singal power
sigma2=1;    % Noise powwer
step   =10;

if model==1
   y =randi([1,step],1,channel);
   m =channel;
   S0=S*step/m;
   b =log2(1+(S0*y/sigma2));
   y =floor((b/step)*100);
else
   if model==2
      y   =randi([1,step],1,channel);
   elseif model==3
      ratio=0.01; % Td / Tb, delay spread over OFDM frame duration
      % Create channel correlation matrix
      c   =2*pi*ratio;
      temp=[];
      for n=1:channel
         nc   =(n-1)*c;
         entry=(1+1i*nc)/(1+nc^2);
         temp =[temp entry];
      end
      % Crate complex channel autocorrelation matrix
      Rchan=toeplitz(temp);
      % Create complex sample channel H
      [E,L]=eig(Rchan);
      SQRTL=sqrt(L);
      A    =E*SQRTL/sqrt(2);
      G    =randi([1,step],channel,1)+ 1i*randi([1,step],channel,1);
      H    =A*G;
      y    =abs(H)';
   end
   temp=sort(y,'descend');
   m   =channel-1;
   S0  =S*step/m;
   while (sigma2/temp(m+1))>=(S0+(sigma2/temp(1)))
      m =m-1;
      S0=S*step/m;
```

```
    end
    m =m+1;
    S0=S*step/m;
    for i=1:channel
       if y(i)>=temp(m)
          b   =log2(1+(S0*y(i)/sigma2));
          y(i)=floor((b/step)*100);
       else
          y(i)=0;
       end
    end
end
end
```

## Assign Vector function

```
function y=AssignVector(Resident,NC,LP)
% NC: NoChannel
% LP: LearningPeriod
% Compute assignment vector for each resident flow.
% Initial checking: learning period and first computation

for i=1:size(Resident,1)
   if Resident(i,6)==1 || (((Resident(i,3)+Resident(i,4))<LP) &&...
         Resident(i,5)==1)
      Resident(i,5)=0;
      Resident(i,6)=0;
      f=Resident(i,6+NC+1:6+2*NC); % Max rate matrix
      A=-f;
      B=-Resident(i,2);
      lb=zeros(1,NC);
      ub=Inf(1,NC);
      for j=1:NC
         if f(j)==0
            ub(j)=0;
         else
            ub(j)=ceil(Resident(i,2)/Resident(i,6+NC+j));
         end
      end
      M=1:NC;
      e=2^-24;
      [x v s]= IP(f,A,B,[],[],lb,ub,M,e); % From 'IP.m'
      Resident(i,6+2*NC+1:6+3*NC)=x;
   end
end
y=Resident;
end
```