Priority Allocation in Distributed scheduling

A Project Report

submitted by

ARCHANA T R

in partial fulfilment of the requirements

for the award of the degree of

MASTER OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

MAY 2013

THESIS CERTIFICATE

This is to certify that the thesis titled **Priority Allocation in Distributed scheduling**, submitted by **Archana T R**, to the Indian Institute of Technology, Madras, for the award of the degree of **Bachelor of Technology** and **Master of Technology**, is a bona fide record of the research work done by her under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. Srikrishna Bhashyam
Project Guide
Associate Professor
Dept. of Electrical Engineering
IIT-Madras, 600 036

Place: Chennai

Date: 6th May 2013

ACKNOWLEDGEMENTS

First and foremost, I am deeply indebted to my advisor Dr. Srikrishna Bhashyam for his guidance and encouragement throughout the course of this project.I would like to express my sincere gratitude for his continuous support,valuable insights and also for allowing me to work at my own pace.

I would like to thank my faculty adviser, Dr Andrew Thangaraj for his help and advice for the past 5 years. Just when I had been pondering if Electrical Engineering was my cup of tea, the courses on Communication systems by Dr. Bhaskar Ramamurthy and other courses in Wireless communications rekindled my interest in the subject. For that, I am very grateful and would like to extend my heart-felt thanks to all the Professors in the Department.

It is a pleasure to thank all my friends and wingmates- Pavitra,Varsha,Kavita,Sadhana, Rini,Swetha,Anindita for making my stay at IITM a lot more interesting.I would also like to thank my batchmates-Vijay,Abishek,Jana,Koganti,Kabra and all my labmates for all their help.

Finally,I thank my sister Ramya,the amazing person she is,for always being there.And as I fall short of words to express my gratitude to my parents,I conclude saying I respect them for being my most gentle of crtics and my best of friends.

ABSTRACT

KEYWORDS: Distributed scheduling, FlashlinQ,QoS,Priority assignment

In the present scenario,scheduling and resource allocation form essential components of wireless data systems.Scheduling refers to the problem of determining which users will be active in a given time-slot whereas resource allocation refers to allocating resources such as bandwidth,Power among the scheduled users.Consequently,there is a growing interest in designing distributed scheduling algorithms for wireless networks which ensure optimal throughput given certain fairness constraints.In this project,we study distributed scheduling as implemented in FlashlinQ ,proposed by Xinzhou *et al.*, which is a synchronous, OFDM-based system incorporating channel-state aware link-scheduling.In FLQ,a random priority allocation method is used to ensure fairness across links.Simulations for different priority-assignment methods were run using an analytical model of FlashlinQ, in MATLAB.A comparison is made between these priority assignment algorithms based on queue length-distribution and average throughput and the best one is suggested.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

FLQ	FlashlinQ
OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency Division Multiple Access
PHY	Physical Layer
MAC	Medium Access Control Layer
P2P	Peer-to-Peer
QoS	Quality of Service
SIR	Signal to Interference Ratio
SNR	Signal to noise ratio
Tx	Transmitter
Rx	Receiver

NOTATION

L_1,L_2,L_3	Links
γ_{Tx}	Transmit Yielding Threshold
γ_{Rx}	Receive Yielding Threshold
$\mathcal{C}$	Instantaneous Capacity
sd^2	Direct-link gain variance
N_0	Noise Variance
P_i	Transmit Power of Transmitter i
H_{ij}	Link gain between Tx of link L_i and Rx of link L_j
P	Priority vector
$QL_{t,i}$	Queue length of link L_i in time-slot t
λ_i	Poisson data arrival rate at link L_i
R	Average rate
QL	Queue length-based priority assignment
QLH	Queue length and channel based priority assignment

CHAPTER 1

INTRODUCTION

In a wireless environment,scheduling is mandatory because

- Communication resources are shared amongst geographically separated users.
- Transmission Interfere with each other.
- Channel conditions vary frequently leading to impairments in transmissions such as fading,attenuation etc.,

Therefore,in a wireless network with interfering links,a scheduling policy is necessary to resolve contention between various links attempting to transmit.In other words, scheduling refers to the problem of identifying the set of users allowed to transmit and their corresponding rates and power-levels.In centralized protocols,a base-station or a designated node co-ordinates access to resources while in a distributed scheduling,a node independently decides whether to transmit or not in the absence of a coordinator. The popular max-weight and back-pressure algorithms are throughput optimal but they are centralised algorithms and have high computational complexity.There has been a growing necessity to design distributed scheduling algorithms with low-complexity and low implementation overheads.

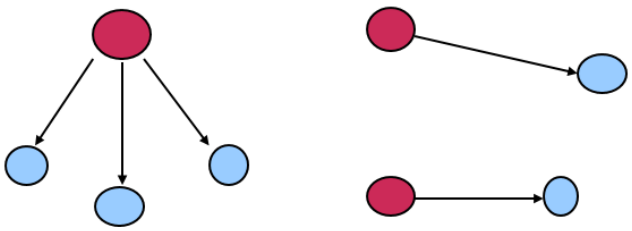


Figure 1.1: Centralized scheduling and Distributed scheduling

1.1 Motivation for FlashlinQ

Traditionally, the scheduling problem has been studied only in the context of Medium Access Control(MAC)protocols,ignoring the PHY layer.Simplistic channel models, such

as, the collision model are considered and also, the transmission ranges are chosen arbitrarily assuming interference does not occur outside this range.On the other hand,there has been extensive research in the field of information theory,formulating extensions of Shannon's work.Here,we aim to determine achievable rates in simultaneous,successful transmissions for given power levels for different channel models.Still,cooperative communication techniques have had limited success-Pantelidou and Ephremides (2009), Gupta *et al.*

In this light,we study the channel-state aware(SIR-based) distributed scheduling implemented in FlashlinQ,a synchronous peer-to-peer wireless PHY/MAC network architecture which utilizes the parallel channel-access offered by OFDM, and a tone-matrix based analog signalling scheme ,proposed by Xinzhou *et al.*.

1.2 Organization of thesis

The organization of this thesis is as follows.

- Chapter 2 discusses the key ideas involved in the scheduling algorithm of FlashlinQ.
- Chapter 3 discusses the system model considered for simulating the scheduling behaviour in MATLAB. Also, proposes the various criteria based on which priority assignment to links can be done.
- Chapter 4 compares the scheduling behaviour when different priority assignment methods are used and presents some plots and results obtained.
- Chapter 5 concludes this thesis discussing possible future works.

CHAPTER 2

Distributed scheduling in FlashlinQ

FlashlinQ has the following attributes

- Synchronous
- P2P,PHY/MAC network architecture
- utilizes the parallel channel access offered by OFDM
- Analog energy-level based signalling scheme
- SIR-based(channel-state aware) distributed scheduling
- Significant gains over CSMA/CA with RTS/CTS
- No hidden terminal/exposed node problem

It is a complete system architecture including

- *Timing and frequency synchronization*: Since the system operates in licensed cellular spectrum,synchronization is achieved using already existing infrastructure.
- *Peer-discovery*: a mechanism to discover the presence of nodes in the neighbourhood.
- *Link-management*: a protocol to assign a unique connection ID(CID) to the links in the system.
- *Channel-state aware* distributed power,data-rate and link-scheduling protocols.

2.1 Crucial Design Ideas

Since we are mainly dealing with the link-scheduling policy,we assume that links are already established between nodes and study an example with 2 links to understand the key ideas used.Consider 2 uni-directional links L_1 and L_2.T_{xi} and R_{xi} are the Transmitter and Receiver nodes of link Li.Here,the two links have direct-link gains $\{|H_{11}|^2, |H_{22}|^2\}$ and cross-link gains $\{|H_{12}|^2, |H_{21}|^2\}$ where $|H_{ij}|^2$ denotes the fraction of the Tranmit Power P_i received at the Receiver of link j when Transmitter of link i

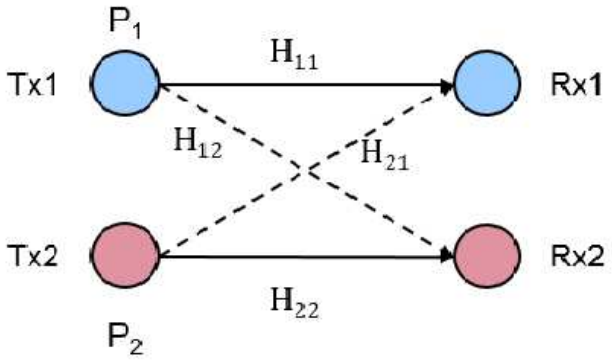


Figure 2.1: Scheduling:two-link scenario

transmits.If the cross-links gains are small compared to direct-link gains and the two links do not interfere with each other,they can be simultaneously scheduled.

In cases where only one link can be scheduled,there arises the problem of priority assignment.In our example,let us consider link L_1 has priority 1 and link L_2 has priority 2.This means that link L_2 yields if it being scheduled causes to severe damage to the SIR at R_{x1}.In short,the protection condition to be satisfied :

$$\frac{P_1|H_{11}|^2}{P_2|H_{21}|^2} \geq \gamma_{Tx} \tag{2.1}$$

Even if the above condition is satisfied,it is not necessary that link L_2 be scheduled.It is important that R_{x2} is able to maintain a sufficient SIR if scheduled,for the transmission to be successful.Simply put,the following condition should be satisfied at R_{x2} else,it yields and is not scheduled in the particular slot.This potentially allows others links with better channel conditions to be scheduled thus allowing close spatial packing and increased system throughput.

$$\frac{P_2|H_{22}|^2}{P_1|H_{12}|^2} \geq \gamma_{Rx} \tag{2.2}$$

Re-randomizing the priority order ensures fairness across links. The main mechanism used to enable distributed determination of the above two criteria by providing the information needed to estimate various SIR s is a two analog-tone-signal exchange

consisting of a inverse power echo and a direct power signal.The direct power signal is sent by a transmitter to allow all receivers to estimate the power received from that transmitter.The inverse power echo is sent by a receiver to allow all other transmitters to estimate 2.1.For a detailed description refer to Connection scheduling in Xinzhou *et al.*.

In summary,the ideas discussed here are:

- A fair priority assignment mechanism
- A Transmit-yielding criterion to protect the Receiver at higher priority links.
- A Receive-yielding criterion to improve network spatial packing.

2.2 Priority Assignment

Signalling mechanism: FlashlinQ exploits the flexibility of parallel, single-tone channels afforded by OFDM to construct an energy-level based (analog) signalling mechanism that provides a miniaturized template of data transmissions, but without collisions. This mechanism enables all links to observe and infer (both from interference and rate perspectives) what would happen if they were to transmit data, but without actually spending the resources needed to perform the data transmissions and without realizing the resulting contentions. The operation timeline of FlashlinQ is shown below:

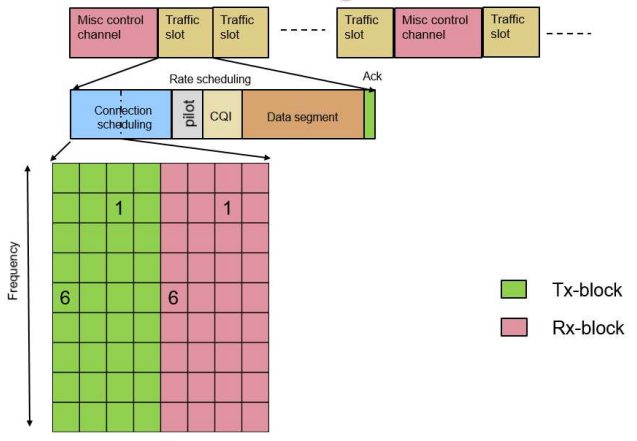


Figure 2.2: FlashlinQ operation timeline

Data transmission occurs in slotted time of around 2ms each. Within each slot,link and rate scheduling is followed by the actual data transmission.Every 1s, resources are allocated for other channels such as peer discovery and link management.

2.3 Connection Scheduling

Each link is assigned a unique connection ID (CID) which is an index between 1 and 112. Each of these CIDs correspond to a pair of single tones (one each for the transmitter and the receiver) within a tone matrix comprising 112 tone pairs (28 parallel tones, over two blocks - the Tx-block and R_x-block - of four OFDM-symbols). The mapping from the CID s to the actual tone pair within the matrix is random. The tone matrix of single-tone pairs satisfies the following requirements: these tones are orthogonal and analog, and a natural priority ordering is assigning the highest priority to the top-left tone-pair and the lowest to the bottom-right, and with lexicographical ordering (the x coordinate has higher priority than the y coordinate). The random remapping of CIDs to tone pairs for each time-slot ensures fairness across links

- The Tx-block is used by potential transmitters to make a request to be scheduled. The transmitter node transmits power on the symbol and tone allocated to it in the Tx-block. This request is a direct power signal, that is it is sent at power that would be used for the traffic channel. All the potential receivers listen to the Tx-blocks and determine if they need to perform Rx-yielding.
- The Rx-block If a receiver chooses to Rx-yield, it does not respond, other-wise it transmits power on its symbol and tone using its inverse echo power level described earlier in this section. All the transmitters listen to all the tones and symbols in the Rx-block to determine whether to Tx-yield.

2.4 Algorithm description

In each time-slot t,

- Links are assigned a random priority order
- For link L to be scheduled, both Tx and Rx of the link should allow data transfer.
- $Tx - yielding$: occurs if link with priority L causes too much interference to an already scheduled link $1, 2, ..L - 1$
- $Rx-yielding$: occurs when Rx-node of L does not see sufficient SIR.(Interference is taken to be the sum of all interference from all higher priority links)
- Cascaded Tx and Rx yielding: Iterations leading to channel state-aware maximal matching
- Re-randomize the priority at each time slot : fairness across links

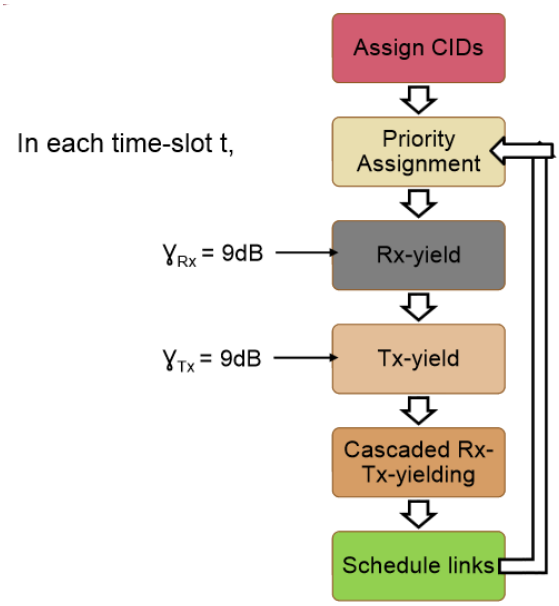


Figure 2.3: Algorithm Description

- Perform Rate Scheduling: For each scheduled link,each interfering Tx guarantees a certain SIR,but together they might cause the actual SIR to fall below yield threshold!

In each time-slot, we concentrate on assigning priorities to the links based on a certain criteria and then compare the performance between random priority assignment and other mechanisms.

CHAPTER 3

Priority Allocation

3.1 Importance of Priority ordering

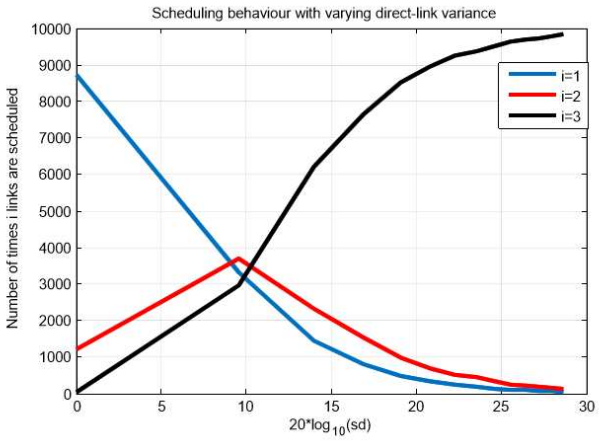


Figure 3.1: Scheduling behaviour with varying sd

From the above plot,we figure that when the channel conditions are good,almost all links in the system are scheduled irrespective of the priority allocation. But in most cases, when all links cannot be scheduled, it becomes necessary to be able to give priority to certain links over others. We see that for lower values of sd, only one link,with highest priority,is scheduled maximum number of times.In such cases,it is optimal to schedule links with higher queue length or high direct-link gains.

3.2 System model

The MATLAB model of FlashlinQ developed can be used to accommodate any number of links.For simplicity in analysing the scheduling behaviour,we consider a 3-link model.The 3 links L_1, L_2, L_3 have direct-link gains $\{|H_{11}|^2, |H_{22}|^2, |H_{33}|^2\}$ and cross link gain at the Rx of L_j because of Transmission by Tx of L_i is $|H_{ij}|^2$ where $i, j \in 1, 2, 3, i \neq j$.Transmit-yielding threshold(γ_{Tx}) and Receive-yielding threshold are both set to be 9dB.

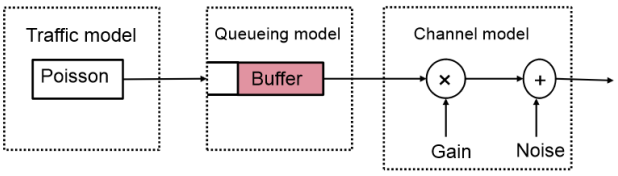


Figure 3.2: System Model

3.2.1 Channel model

In each time-slot,in each link,the channel varies as a rayleigh fading model (refer Tse (2005),Chang *et al.*)

$$H_{ii} = randn(1) + j * randn(1) \tag{3.1}$$

Though channel varies with each time-slot,we assume flat-fading in any given slot. At the Rx,at an instant n:

$$\begin{aligned} y[n] &= ||h||x[n] + w[n] \\ w[n] &\sim iidCN(0, N_0) \\ SNR &= \frac{P_t|h|^2}{N_0} \end{aligned} \tag{3.2}$$

Consider a real-valued scalar Gaussian-noise channel of the canonical form:

$$Y = \sqrt{snr}X + N \tag{3.3}$$

where snr denotes the signal-to-noise ratio of the observed signal,and the noise $N \sim N(0, 1)$ is a standard Gaussian random variable independent of the input, X.

When the distribution $P(x)$ of the input X is standard Gaussian,the input-output mutual information is then the well-known channel capacity under constrained input power [5]

$$I(snr) = C(snr) = log_2(1 + snr)bits/s/Hz \tag{3.4}$$

When a link is scheduled,Rate = C else Rate = 0. In our study,the channel gains are varied so as to satisfy the condition:

$$E[|h_{ij}|^2] = sd^2, if(i = j) \tag{3.5}$$

$$= 1, otherwise$$

3.2.2 Queueing model

We consider a Single-input Single-output(SISO) model. We assume an infinite buffer at the transmitter of each link and also the knowledge of channel gain at each sample interval.Therefore, rate-adaptive transmissions and strong channel coding can be used to achieve error-free transmission. Thus, the service rate of the queue is equal to the instantaneous channel capacity R. For simplicity we adopt a fluid model,where the size of a packet is assumed to be infinitesimal. The arrival process follows a poisson distribution with arrival rate λ .In each time-slot t,the Queue length is calculated as

$$QL_{t,i} = (QL_{t-1,i} + A_t - R_t t_s)^+ \tag{3.6}$$

For $t = 0, QL_0 = 0$.Take $ts = 1$.A_t is the amount of traffic arrival.R_t is the data that can be successfully transmitted in that time slot.$QL_{t,i}$ is the Queue length at time slot t in link L_i.

3.2.3 Traffic model

For simplicity,we consider the data arrival to be poisson with rate λ. A Poisson process models random events,in this case data arrivals,as emanating from a memoryless process.

3.3 Different Priority assignment mechanisms

3.3.1 Random Priority Allocation

The basic FlashlinQ model supports random priority allocation,using a tone-matrix based mechanism.This ensures that over a period of time, all links receive equal resource-allocation. For our simulations we generate the vector P ,in each time-slot as:

$$P = randperm(n) \tag{3.7}$$

In each time-slot,the priority vector is a random permutation of n links,where n is the number of links in the system.Link P(i) has priority i.

3.3.2 Channel based Priority assignment

When the direct-link gains of all the links have similar variance,this priority assignment gives results which are not very different from random priority allocation.This is because FlashlinQ scheduling is inherently channel-state aware.

$$arg\max_i h_1, h_2, ...h_i \tag{3.8}$$

In MATLAB,

$$[a, P] = sort(X,' descend'); \tag{3.9}$$

where $X(i) = |H_{ii}|^2$

In each time-slot,vector P returns the priority vector based on channel-state.In cases where the direct-link gains of links vary significantly,the links with better channel conditions are favoured for channel-use.

3.3.3 Queue length based Priority assignment

In this case,priority is assigned based on the queue length at the Tx of each link.In case of 3.3.1,all links are scheduled almost equal number of times,irrespective of queue length and average rate acheived by each link.

$$arg\max_i QL_1, QL_2, ..QL_i \tag{3.10}$$

At the end of each slot,the queue length is calculated as : for t>1 and each i :1 to n

$$QL_{t,i} = (QL_{t-1,i} + arrrate(i) - C(i))^+$$

$QL(t;i)$is the Queue length at time slot t in link Li.The priority vector for the next slot is calculated as

$$[vec, P] = sort(QL(t,:),' descend') \tag{3.11}$$

3.3.4 Queue length and channel based Priority assignment

This priority assignment method takes into account both Queue length and Channel conditions.For each link,we calculate the product of QL of previous slot and H(direct link gain) in the current slot and find priority vector P.

$$arg\max_i QL * H_1, QL * H_2, ...QL * H_i \tag{3.12}$$

$$PL(i) = QL(t-1,i)X(i)$$

$$\text{where } X(i) = |H_{ii}|^2$$

$$[vec, P] = sort(PL,' descend') \tag{3.13}$$

This priority assignment mechanism gives precedence to those links which have a higher queue length and are also able to meet certain rate requirements due to favourable channel gains.

3.4 Distributed Scheduling

The baseline FlashLinQ connection scheduling protocol enforces a random priority allocation among links. Over time, this mechanism makes sure all links obtain a similar share of the channel use. However, it is desirable for the system to be able to give higher priority to certain links over others.The tone-matrix can be divided into multiple sub-blocks representing different priority levels (priority ordering across blocks of tones). A link is assigned multiple tone-pairs at any timeslot, the choice of which tone-pair to use dynamically depends on the queue-length/backlog or packet delay.

For example,consider the link with CID 6,which is assigned 3 tone-pairs.Depending

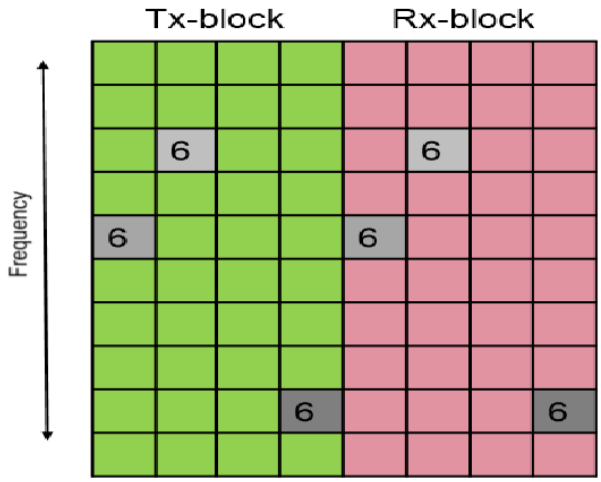


Figure 3.3: Distributed Scheduling assigning multiple tone pairs

on the queue length/backlog,the link selects the tone-pair which consequently assigns a higher priority or a lower priority.This way,each node is assigned a suitable priority with no queue length information from the other nodes i.e., in a distributed way.This mechanism of assigning priorities is suitable for scenarios with lesser users.

CHAPTER 4

Plots and Observations

In the simulation results shown below,we consider 3 links and the number of iterations(time-slots) $N = 10000$.

4.1 Varying Yielding thresholds

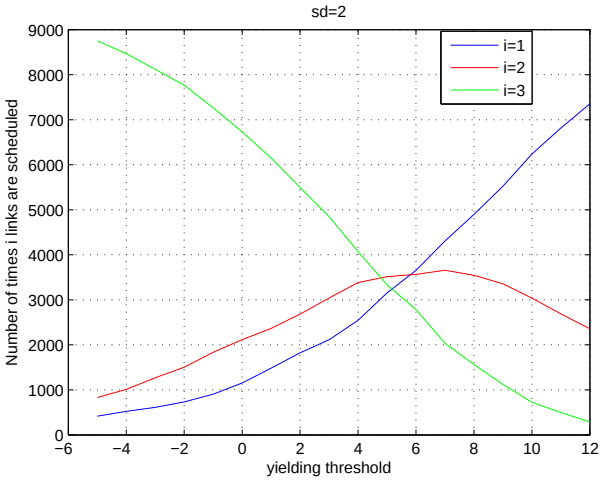
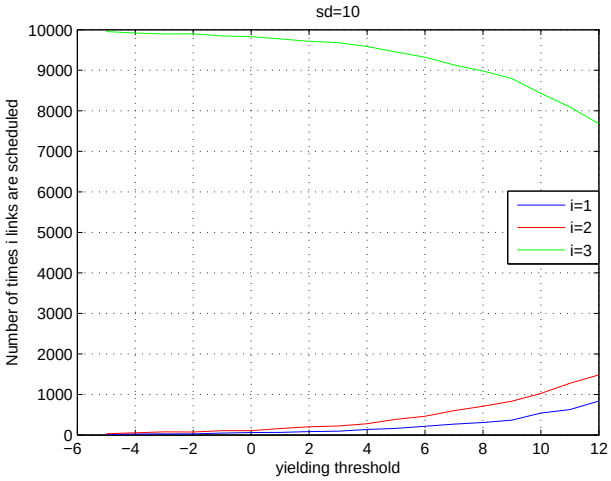


Figure 4.1: Number of times i links scheduled vs Yielding threshold



Parameters considered for the above simulations:

$$H_{ii} \sim N(0, sd^2)$$
$$H_{ij} \sim N(0, 1), i \neq j$$

We note that if the channel conditions are good,even for higher yielding thresholds,all 3 links have good channel access.However,with lower yielding thresholds, performance degrades since all links transmit simultaneously but with bad SIRs.An optimized choice of the yielding threshold can achieve a good throughput in both strong and weak interference scenarios(refer Yoon *et al.*) as mentioned in Xinzhou *et al.*,we choose $\gamma_{Tx} = \gamma_{Rx} = 9dB$ which was determined as optimal based on both simulation and implementation results.

4.2 Comparison

Here we analyse the performance of the scheduling algorithm when different priority assignment methods are used for different cases.

4.2.1 Queue length vs load

Case 1

Similar direct-link gains,different traffic arrival rate

Parameters:$h_{ii} \sim N(0, sd^2) i \in 1, 2, 3, sd = 5. \lambda_1 : \lambda_2 : \lambda_3 = 1 : 2 : 3$

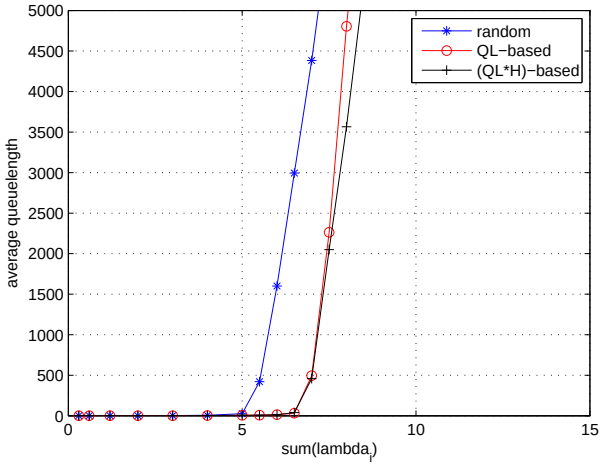


Figure 4.2: Queue length vs load for Case 1

We note that the cut-off for random-priority allocation occurs at $\Sigma\lambda_i$ = 5 as opposed to $\Sigma\lambda_i$ = 7 and $\Sigma\lambda_i$ = 6.5 in case of QL and QLH based priority assignment.This shows the system following the QL and QLH-based priority assignment mechanism can supports a 30 percent extra load.There is not much different between QL and QLH based curves;this implies that the channel information is redundant.This might be because FlashlinQ scheduling is inherently channel-state aware.

Case 2

Different direct-link gains,same traffic arrival rate

Parameters:$\lambda_1 = \lambda_2 = \lambda_3, |h_{11}|^2 = 2|h_{22}|^2 = 4|h_{33}|^2 h_{33} \sim N(0, sd^2), sd = 5$

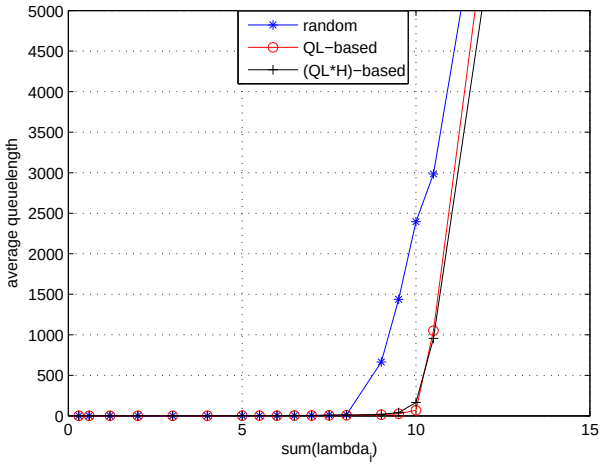


Figure 4.3: Queue length vs load for Case 2

In this case,cut-off occurs at $\Sigma\lambda_i$ = 8 for random priority allocation and at $\Sigma\lambda_i$ = 10 for both QL and QLH based priority assignment.The QL and QLH priority mechanisms are seen to support a 20 percent extra load compared to random priority allocation.

4.2.2 Queue length Distribution

In this part,we consider the queue length distribution in each link individually as well as for the Queue length averaged over all links i.e., the whole system as a single queue(refer Marbach).Here,for each Queue length B (bits) we compare the probability that the queue length exceeds B, for different priority allocation mechanisms.

Case 1

Different direct-link gains,same traffic arrival rate λ

Parameters: $\lambda = 2, |h_{11}|^2 = 2|h_{22}|^2 = 4|h_{33}|^2 h_{33} \sim N(0, sd^2), sd = 5$

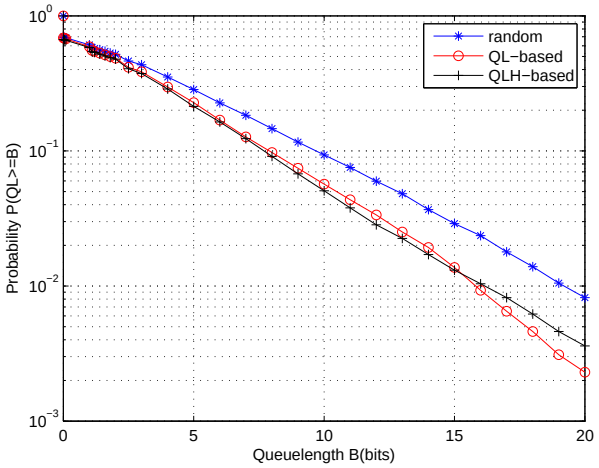


Figure 4.4: Queue length distribution Case 1

The figure 4.4 shows the queue length distribution for the system of 3 links.We see that QL and QLH-based priority assignment is better than random priority ordering.

Case 2

Similar direct-link gains,different traffic arrival rates

Parameters:$\lambda_1 = 1, \lambda_2 = 1.5, \lambda_3 = 2, h_{ii} \sim N(0, sd^2), sd = 5i \in 1, 2, 3$

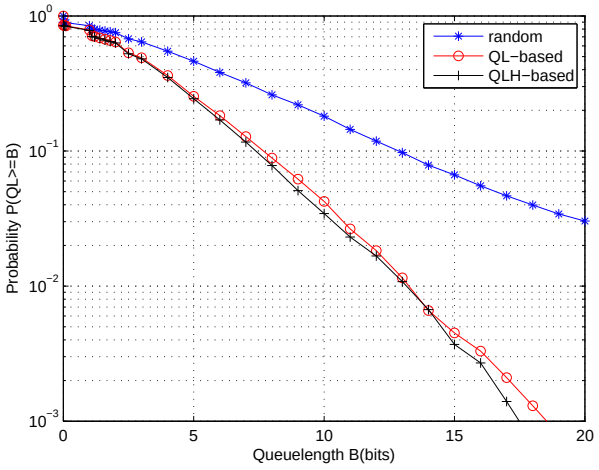


Figure 4.5: Queue length distribution Case 2

From both figures 4.4 and 4.5,we observe that QL and QLH based priority assign-

ment is more favourable as compared to random priority ordering.

4.2.3 Fixed queue length

Let us consider a case where each link has a fixed queue length of b bits at time $t = 0$.We now compare the number of time-slots required,in each of the priority allocation mechanisms, to deplete these b bits.

Objective:Minimize the number of time-slots required to deliver all data to the intended destination.

Minimize : $E(N)$

subject to :$QL(0) = b, QL(N) = 0$

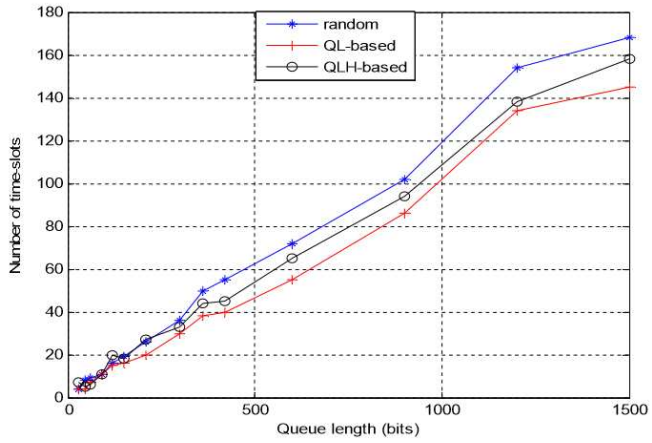


Figure 4.6: Time to deplete the Queue:Fixed initial Queue length

From the figure,we observe that for any given queue length b,the number of time-slots t required is more for random priority assignment than QL and QLH based.

CHAPTER 5

Conclusion

5.1 Contribution of the thesis

From the simulation results,we clearly see that

- QL(queue length) based and QLH(queue length and channel) based priority assignment have significant improvement in the context of Queue length distribution in the system.
- From the plots,it can be inferred that the channel information in QLH based mechanism is redundant.This might be because the FlashlinQ scheduling is inherently channel-state aware.

We also track each link individually and observe the following:

- For links with higher channel gains,H-based priority allocation ensures slightly better performance.
- For links with lower channel-gains,QL-based priority allocation gives a relatively better throughput.But this happens at the cost of allocating lower priority to links with better channel-gains.
- Considering the above points,it is necessary to propose a priority allocation algorithm that takes into account both channel conditions and queue length at each of the links.
- In Queue length based scheduling,when links have similar channel conditions,links with higher traffic get higher priority and thus attains a higher average data rate,as desired.

It was seen that QL-based and QLH-based priority assignment are better when compared to random priority allocation.QLH- based strikes a balance between keeping the queue length in check and prioritising the link which can attain the highest rate.But scheduling behaviour significantly depends on channel-conditions.QL-based scheduling works best for links with poor channel gains but at the cost of links with higher direct-link gains $Tx - yielding$.

5.2 Future work

Models of the traffic offered to the network or a component of the network will be critical to providing high quality of service(QoS). Traffic models are used as the input to analytical or simulation studies of resource allocation strategies.As part of the project,an analytical model of FlashlinQ has been simulated in MATLAB to accommodate any number of links.The various aspects of link-scheduling can be observed for self-similar traffic,which represents real-time traffic-arrival behaviour better.

In our model,we assume the choice of tone-pairs,and consequently the priority,occurs with no error. For example,in QL-based allocation, a link with higher queue length is always assigned a higher priority than other links with lower queue length.It may happen that the lowest of the priorities assigned to a link with lower QL can still be higher than the highest priority assigned to link with higher QL.This error,though minimal can be taken into account.

It can also be modelled to support an adaptive threshold which changes over a slower time-scale, which can be further used for providing QoS or fairness for links..With suitable Power-allocation methods,it can be seen that QLH-based priority assignment ensures that all scheduled links achieve minimum rate requirements.

REFERENCES

1. *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

2. **Chang, L.**, **Q. Xiaowei**, **Z. Sihai**, and **Z. Wuyang**, Queuing analysis of self-similar traffic in rayleigh fading channel. *In Networks Security, Wireless Communications and Trusted Computing, 2009. NSWCTC '09. International Conference on*.

3. **Gupta, A.**, **X. Lin**, and **R. Srikant**, Low-complexity distributed scheduling algorithms for wireless networks. *In INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*.

4. **Marbach, P.**, Distributed scheduling and active queue management in wireless networks. *In INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*.

5. **Pantelidou, A.** and **A. Ephremides** (2009). The scheduling problem in wireless networks. *Communications and Networks, Journal of*.

6. **Xinzhou, W.**, **S. Tavildar**, **S. Shakkottai**, **T. Richardson**, **J. Li**, **R. Laroia**, and **A. Jovicic**, Flashlinq: A synchronous distributed scheduler for peer-to-peer ad hoc networks. *In Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*.

7. **Yoon, H.-W.**, **J. S. Kim**, **S. J. Bae**, **B.-G. Choi**, and **M. Y. Chung**, Performance analysis of flashlinq with various yielding threshold values. *In ICT Convergence (ICTC), 2012 International Conference on*.