無線通信系統應用資源控制之設計

研究生：羅文嶽　　　　　　　　　　　　指導教授：黃經堯

國立交通大學

電子工程學系　電子研究所碩士班

摘要

由於在一個無線通信系統內的有限的射頻資源，特別是當那些資源沒有使用中，使資源的佔有減到最小是其關鍵。由於那些封包之間的不可預測到達時間的原因，使用一個固定間歇計時器來釋放所有連結資源是不足以來使連結資源最小化。在這篇論文裡，提出一種基於資料應用特性的分析提出三種資料流量測方式來分類不同資料應用程式，和提出一個基於實際資料流統計的適應性間歇定時器控制演算法，並藉由cdma2000的技術的端對端模擬來量化驗證結果。

Application based Resource-Aware Controls in Wireless Communication Systems

Student: WenYue Lou Advisor: ChingYao Huang
Department of Electronics Engineering & Institute of Electronics
National Chiao-Tung University, HsinChu, 2004

Abstract

With limited RF resources in a wireless communication system, it is critical to minimize the occupancy of resources especially when those resources are not active. Due to the uncertainly caused by unpredictable inter-arrival time between packet calls, a fixed dormancy timer before releasing all connection resources is not sufficient to minimize the use of connecting resource. In this thesis, three traffic measurements are proposed to categorize different data applications based on the characteristics of the data applications, and an Adaptive Dormant Timer (ADT) control algorithm based on actual data traffic statistics will be proposed and quantified by an end-to-end simulation based on cdma2000 technologies.

Acknowledgement

At first, I want to thank ChingYao Huang, my advisor, for sharing the research experience and knowledge. Also thanks to his proposal when I met difficulty. Then I want to thank my group members for offering the help in research to me, offering some new knowledge to me, and discussing in the research. Thank ChenTzu Kao for offering a lot of suggestions to me on network, sharing the knowledge on some TCP and RLP to me, and the part of helping on TCP in the simulation. Thank ChenKun Chuang for helping in the part of error model and MAC re-transmission. Thank YiLin Tsai for the suggestion of random access in cdma2000. Thank HueiYuan Su for helping me to set up the traffic model. Thank all the members in the lab for helping on no matter lessons or at the thesis finally.

CONTENTS

Figures

Tables

Chapter 1 Introduction

With data burst transmission, the setting of the dormancy timer is important to ensure the saving in the required resources and the power budget. After finishing a packet-call transmission, the dormancy timer designed for delaying releasing of the physical connection is critical to prevent excessive call setup process. Typically, an active data session consists of multiple packet calls[1]. To optimize the use of resources, usually, the system can not afford to allocate RF resources for an entire data session. Instead, the resources will be allocated only when the call is active in transmission.

Traditionally, the power saving in a wireless communication system is done mostly by the fast power control and sleep-mode operations by increasing the sleeping time at mobiles [1]. Besides, there are some fine-tune proposals to further reduce the power consumption. In [2-3], the operation of the mobile handset will consume the battery power. The component of the power consumption can be divided into three parts, which are in receive unit, processing unit, and interface unit. The processing unit and the interface units can switch off when no data arrives at the receive unit and switch on when the data is received. The units switch on immediately when receiving the packets will cause too many operations. The queuing buffer can be used to avoid some on-off operations. The system decides to transmit the packet in the buffer when the packets

---

[1] The packet-call is defined as one action to download (or send) data files. Fore example, the packet-call can be triggered by a mouse click to download a file or a web page.

in the buffer are over some threshold.

The fraction buffering threshold can archive the more power saving than the integer buffering threshold. In the paper [4, 5], it analyzes some different power saving schemes. It points out that there are some fundamental problems to be solved for the discontinuous reception. For analyzing the different systems, there are some similar approaches for the power saving schemes. Basically, the system has a buffer to save the packets without requesting for transmission and the mobiles in the power-saving mode only wake up periodically to monitor the channel. Another approach is to synchronize the wake up. The system is usually the central authority. There are some wireless communication systems which transmit specific control packets to inform the packet information in the system buffer. The systems have a wake up windows that the mobiles wake up to receive the specific packet in this period. The mobiles can not wake up every wake up window. The paper provides an asynchronies wake up algorithm (IPS[2]) where the central control periodically transmits inform packets. The mobiles wake up asynchronies and listen to the channel. If the mobile receives any inform packets, the mobiles will increase the wake up time for listening. The increase time is the period of the specific packet. The mobiles can change the duty circle to match the system requirement. At a potential cost of extra power consumption, increasing the duty circle can decrease the packets delay and increase the transmission rate, but require more power. In [6], the control hold mode in the cdma2000 system has a lot of benefits such as power saving,

---

[2] In Band Power-saving Protocol

reducing overload and the channel interference, and etc. However, the original system can not fully obtain the benefits from the control hold mode because the MS has to continuously monitor the F-PDCCH[3] in order to get the MAC_ID. This action causes higher power consumption and gives an overload to the decoder. A new channel F-WUCH[4] is created to reduce the overhead from continuously monitoring the F-PDCCH.

Some modern radio resource management schemes and researches are introduced as below. In the paper [7], due to limited system capacity, limited number of base station, and limited battery power of mobile, dedicated channels for packet service users are allocated on demand and released shortly in the end of the service. However, release the channel and re-establishing increase the delay and additional overhead because the process has to exchange the user information between the base station and the mobile. The cdma2000 MAC utilizes timer entities that provide trigger for MAC state transition. The relationship between the radio resource saving and packet delay in the cdma2000 system is provided. In the paper [8], because the systems need to provide high quality voice services as well as high speed packet data, a mechanism sharing the radio and network resources efficiently by utilizing the characteristics of the heterogeneous traffic. Accurate characterization of packet data traffic is critical to investigate the radio resource management problems. Packet switching introduces smaller average access delay and higher the efficiency of resource utilization, but it also introduces larger average packet delay for all users. By a burst

---

[3] Forward Packet Data Control Channel

[4] Forward Wake Up Channel

switching scheme, the system can smaller the average per packet delay. In the paper [9-10], the resource consumptions of the mobile tracking have two main components, (1) the system uses wireless resources for paging users and (2) the mobiles uses power for listening the information to detect the location change and for the registering the new location. If a user registers the location for a long duration, the consumption of (2) can be reduced and the overload of paging the mobile will increase because of the less close track of the mobile location. The system resources must trade-off between resources of (1) and (2). In the paper [11], the proposed resource management scheme is comprised of a combination of power distribution, rate allocation, service scheduling, and connection admission control. The performance analysis in terms of new connection blocking probability, handoff connection dropping probability, and resource utilization is given.

For heterogeneous services, different data packet-calls have different traffic characteristics in terms of sizes, the inter-arrival time between packet-calls, and etc. With user behavior, it is difficult to set up a proper dormancy timer to terminate the connection. In this thesis, the resource-aware control based on actual data traffic statistics will be proposed to improve the use of the system resources in term of the required hardware resources and the power budget.

The thesis is organized as follows: In Chapter 2, the dormancy timer problem will be addressed. The characteristics of various data applications are investigated in Chapter 3. In Chapter 4, an adaptive Active-to-Dormant timer is proposed. The system simulation model and flow are proposed and the performance of the proposed algorithm is evaluated in Chapter 5. The simulation result and analysis are proposed in Chapter 6. Finally, the conclusions and the future works are included in Chapter 7.

Chapter 2 Mac State and Dormancy Timer Problem

2.1 Mac States

Multiple MAC states can save the channel and power resource and provide flexibility in the system. In IS-95 system, there are two MAC states, active state and idle state. When there are no data or voice transmission, the mobile will transmit form the active state to the idle state and release all of the resource. In cdma2000 system, there are more states than IS-95. The data transmission is burst and not continuous. A new state, control hold state, is created to save the channel resource and power for data transmission. In control hold state, the mobile releases the dedicated traffic channel resources and reduces the Reverse-Pilot Channel rate to one-eighth. The state also can provide a fast channel reassignment and fit the data transmission characteristic. Figure 2-1 is the MAC states transmission diagram. **FCH** is fundamental channel in the cdma2000 system. When there are no data or voice to transmit, the channel rate can down to one-eighth of 9.6Kbps. **SCH** is Supplemental channel in the cdma2000 system which supports the four transmission rate (19.6K, 38.4K, 76.8K, 153.2K bits per second).

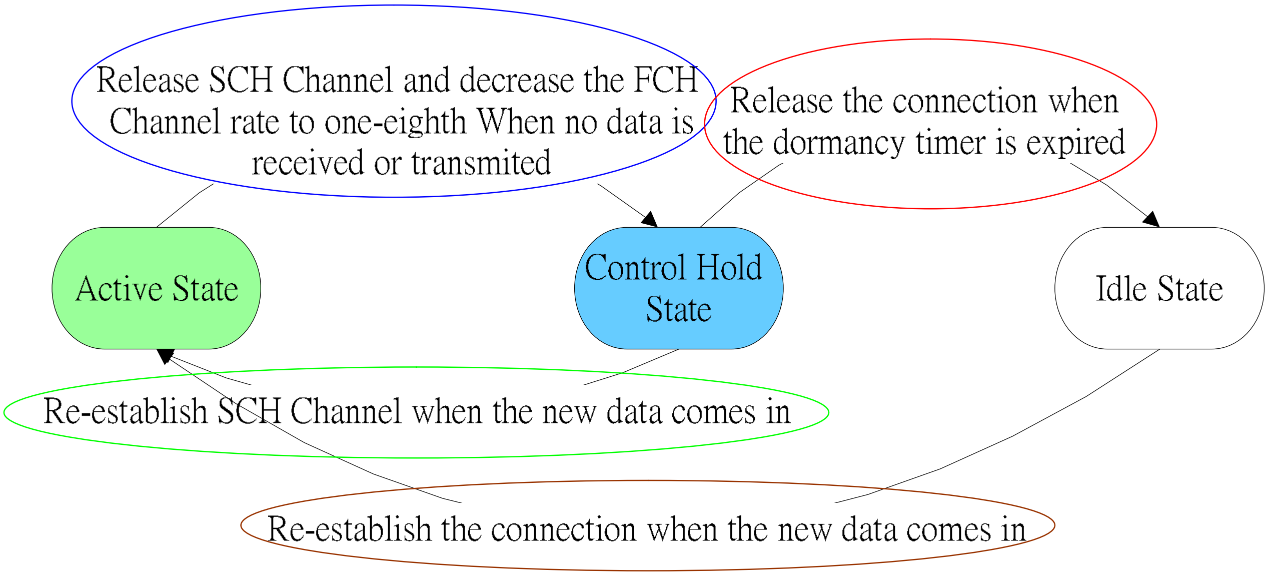

Figure 2-1, MAC States transformation

2.2 Dormancy Timer

The Dormancy timer plays an important role of the algorithm. What is the dormancy timer is introduced first by using the web browsing as an example. Then the relationship between the thinking time and the dormancy timer and the challenge of choosing the dormancy timer are discussed.

With data burst transmission, the setting of the dormancy timer is important to ensure the saving of the required resources and the power budget. After finishing a packet-call transmission, the dormancy timer designed to delay the releasing of the physical connection is critical to prevent extra call setup process if the inter-arrival time of two packet-calls are short.

Typically, an active data session consists of multiple packet calls. To optimize the use of resources, the system can not afford to allocate resources for entire data session. Instead, resources are allocated only when the call is in active (in transmission).

In the Figure 2-2, where **Setup Time** means the delay between the mouse clicking and the traffic channel establishment. **Network Delay** is the delay between first request and the data arriving at the mobile. **Queuing Delay** is the delay waiting for setup up SCH. **Thinking Time** is the duration between the end of the transmission and the next request ion of transmission data. **Dormancy Timer Duration** is the duration for waiting the dormancy timer expired.

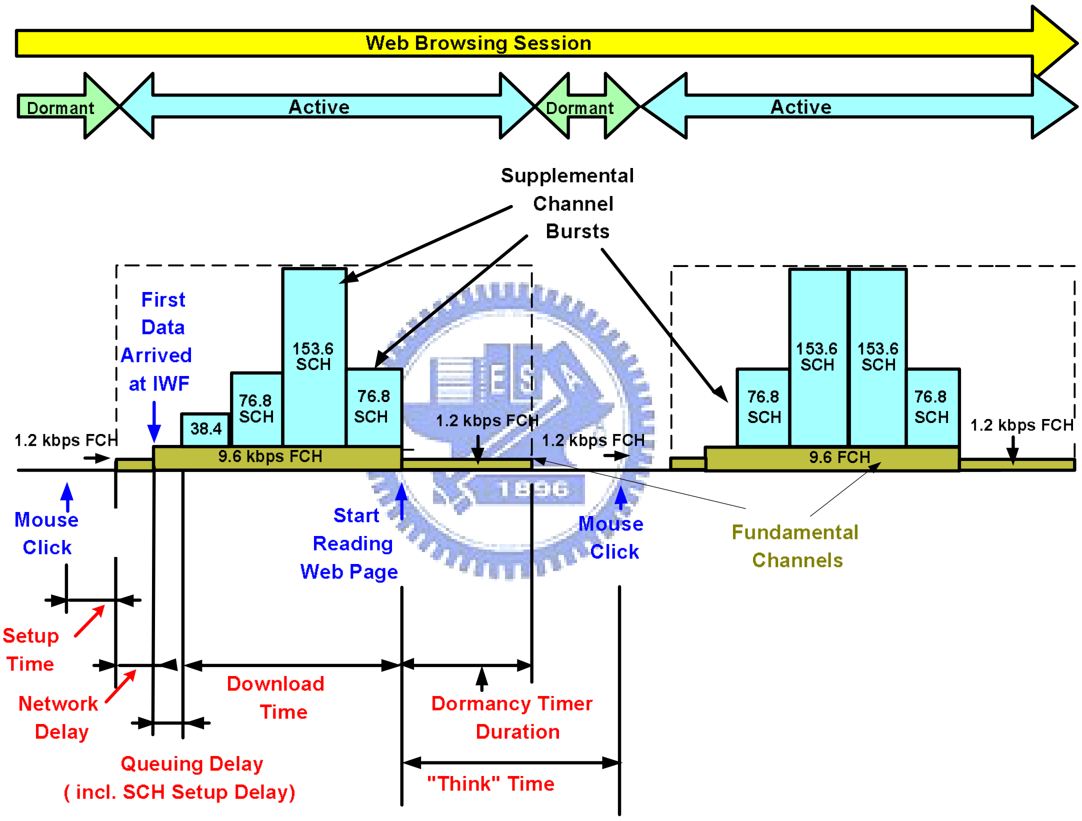

Figure 2-2, Web Browsing Model – cdma2000 Transmission Flow

As shown in the Figure 2-2, during each data-packet call, there exist four states; they are the setup state, transmission state, maintaining state, and dormant state. Except the dormant state, RF and channel elements are allocated to maintain the connection (defined as an active state). The packet-call will be moved from the active state to the dormant state when the dormancy timer expired. Typically, after finishing a packet-call download, a so-called thinking (or reading) time is used for calculating the remaining time before the next packet-call. The challenge here is to dynamic adjust a proper dormancy timer in order to minimize the usage of resources.

Chapter 3 Characteristics of Data Traffic

The analysis of the characteristics of data traffic can provide a good prediction to the dormancy timer. First, we will find the good components for the prediction and then the analysis of relationships between each component and thinking time is provided.

Different from the traditional data traffic models [12], to design a proper dormancy timer, we propose to categorized the data traffic characteristics into three categories: (1) uplink and downlink ratio, (2) packet call size, and (3) channel usage.

3.1 Uplink and Downlink Ratio

The ratio of the uplink and downlink traffic is an important indicator of the characteristics. Different traffic types have different uplink-and-downlink ratios. For example, the voice traffic is a real-time interactive communication. The symmetric traffic makes the ratio close to one. For the video and audio streaming applications, the ratio will be small in the download condition and large in the upload condition. The discussion below will be all in the download condition and the value will be inversed in the upload condition. The ratio is also small for FTP protocol, where downlink occupies the most data traffic for the connection. Because the packet-call size is small and the request of packet-calls is larger in the HTTP data session, the ratio is higher as comparing to the FTP protocol. The simulation results are show in Figure 3-1 and Figure 3-2. The uplink-and-downlink ratio in the HTTP data session is higher than that in the FTP session. In short, the uplink and downlink ratio can be used as an indication to predict the data traffic type.

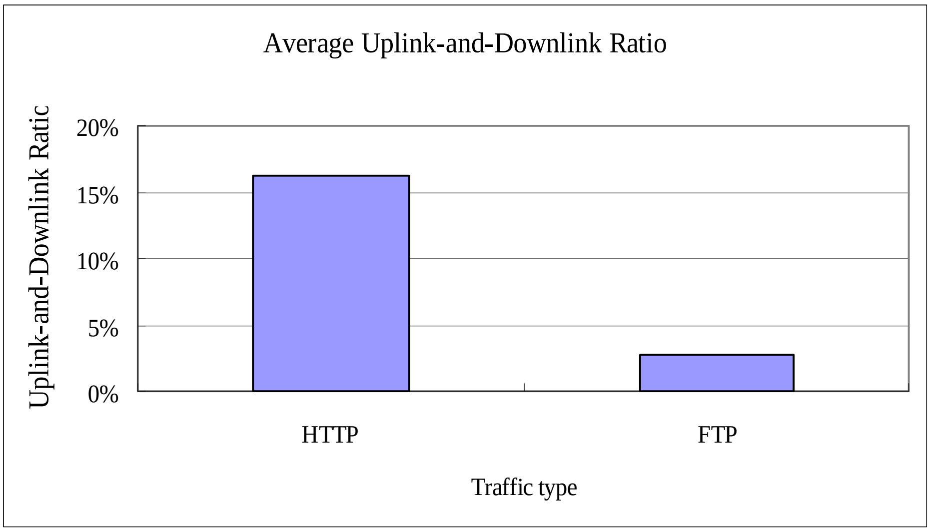

Figure 3-1, average uplink-and-downlink ratio

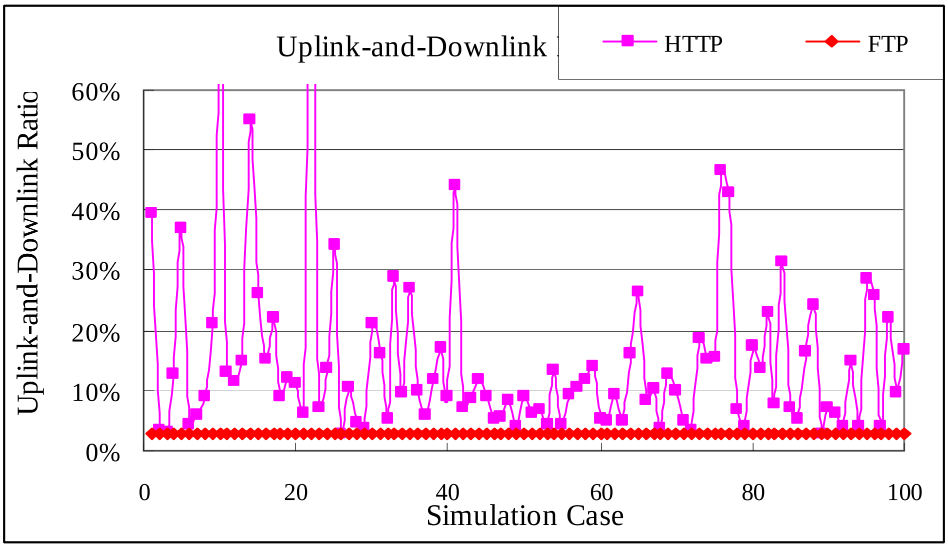

Figure 3-2, uplink-and-downlink ratio (100 cases)

3.2 Packet Call Size

The packet call size of each data transmission is useful to predict the data content. From the 1xEV-DV traffic model [12], the average packet-call size of HTTP is about 60Kbytes and the average packet call size of FTP is about 2Mbytes. The simulation results in the Figure 3-3 and

Figure 3-4 are show that the packet-call size with the FTP Protocol is much larger than that with the HTTP Protocol. When the relationship between the packet call size and the content of the data is discussed in the same application, there is some positive relationship to the dormancy timer; the thinking (or reading) time is usually longer for the same applications with larger packet-call size. So the packet-call size has a positive correlation with the thinking time. Besides, the packet-call size can be used to predict the specific application. For example, the size of SMS is small as compared to other data applications.

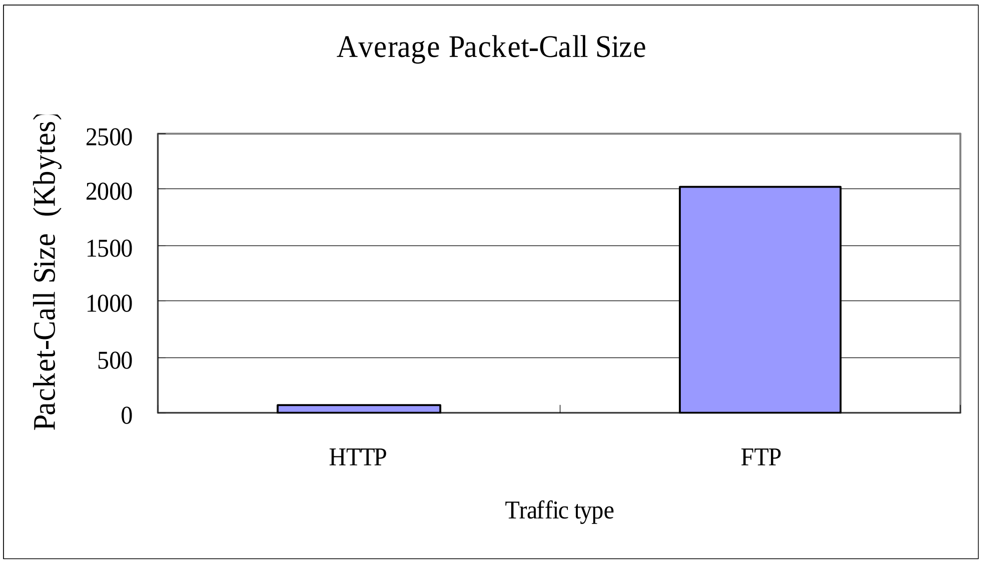

Figure 3-3, average packet-call size

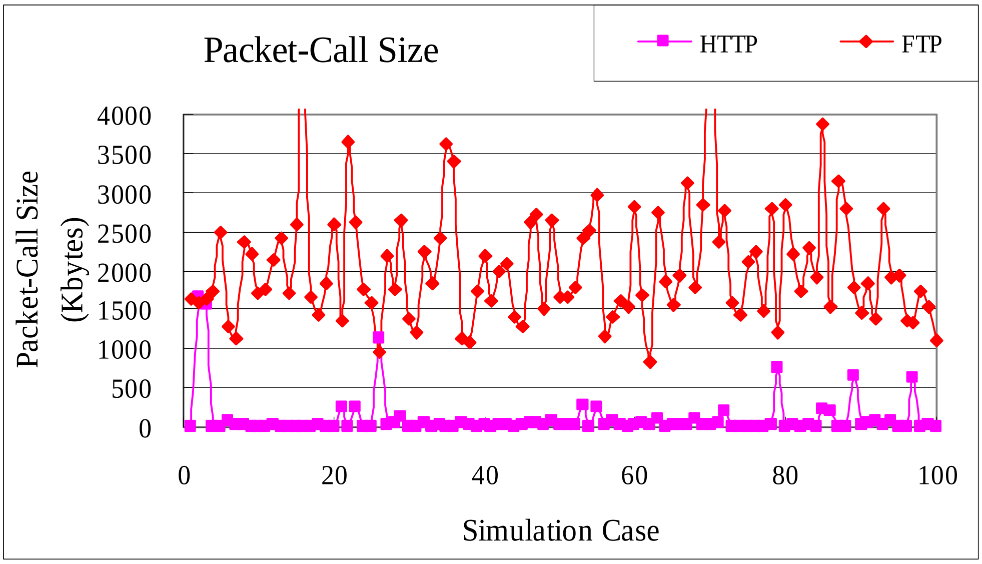

Figure 3-4, packet-call size (100 cases)

3.3 Channel Utilization

The channel utilization is another important indicator to predict the data traffic characteristics. Different protocols and packet-call sizes have different channel usage. For example, the channel utilization of UDP will be higher than TCP. TCP is a connection-oriented protocol and will perform a 3-way handshake and congestion control with slow-start implementation. UDP on the other hand is connectionless and best-effort transmission. UDP doesn't need to transmit the addition handshake packets and UDP tries its best to transmit the packets. So UDP can use the channel than TCP better. When a user transmits a smaller size over a TCP connection, the channel utilization will smaller. The ratio of the TCP overhead, 3-way handshake and slow start, is smaller with larger transmission time. The TCP overhead usually has almost the same impact over a TCP connection. The larger file size is, the smaller the ratio of the TCP overhead is. From the model [12], the average file size of FTP is larger than HTTP. As shown in Figure 3-5 and Figure 3-6, the channel utilization is larger in FTP as compared to HTTP.

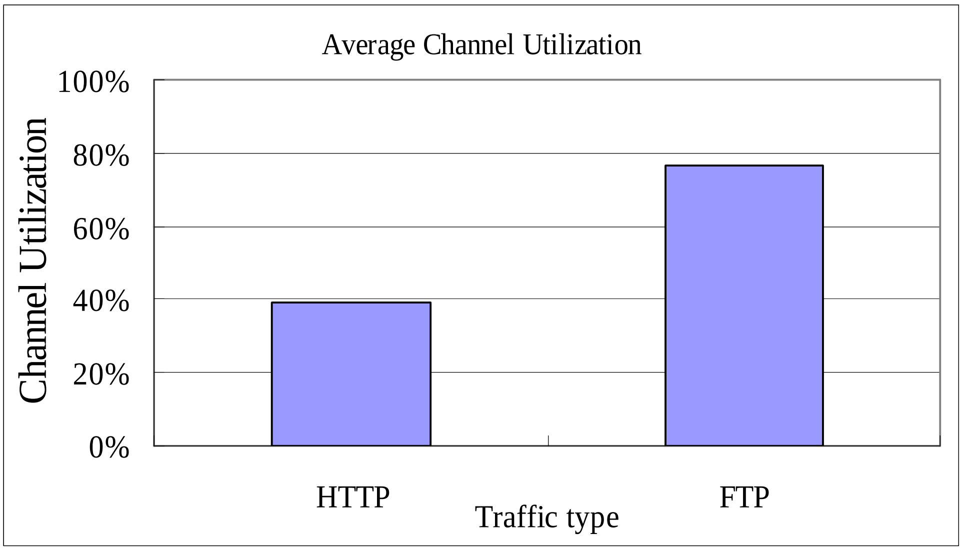

Figure 3-5, average channel utilization

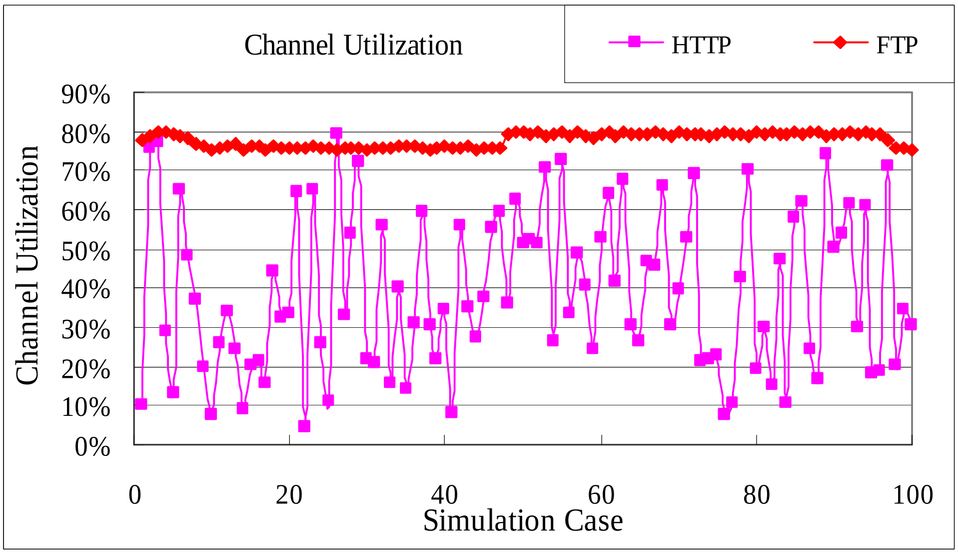

Figure 3-6, channel utilization (100 cases)

As stated, each measurement index has its own value of predict the traffic characteristics, but to identify an application and calculate the required dormancy timers, it is difficult to use only one measurement index. In the following chapter, we will discuss the issues and propose an Adaptive Dormant Timer (ADT) algorithm based on above indications to properly trigger the transition from the active state to the dormant state.

Chapter 4 Adaptive Active-to-Dormant Control Algorithm

In this chapter, an Adaptive Active-to-Dormant Control algorithm is proposed based the uplink-and-downlink ratio, packet-call sizes, and channel utilization. We will introduce the meaning for the data components in the prediction and how to prediction the thinking time from the components. Then the design challenges are discussed and the solutions (*dormancy estimation table* and *adaptive dormant timer*) are provided. Finally, we show the detail of creating the dormancy estimation table and how to predict the dormancy timer from the value of the dormancy estimation table.

4.1 Design Challenges

The design challenge are (1) how to predict the dormancy timer from the thinking time, (2) how to predict the data type, (3) the accurate dormancy timer might be different even for the same data type but different data contents, and (4) the human behavior affects the dormancy timers.

In order to resolve those challenges, a mapping scheme from the thinking time to dormancy timer is proposed. First, different data applications can be predict from the uplink-and-downlink ratio and the channel utilization. After knowing (predicting) the transmission protocol, the packet-call size will then be used to finalize the proper thinking time and the dormancy timer. With this approach, a 3-dimensinal metric categorized by uplink-and-downlink ratio, channel utilization, and packet-call size is created. Finally, the human behavior problem can be overcome by training the prediction value (update the value every packet call).

4.2 Thinking Time Prediction

The main idea of the algorithm is to get the accurate dormancy timer which triggers the mobile goes into the dormant state (sleeping mode). From Chapter 3, there are three data type predictions, the uplink-and-downlink ratio, the packet-call size, and the channel utilization, to determine the thinking time. With the accurate thinking time, the system can then decides when to move the data call to the dormant state.

The ratio of uplink and downlink can be though as the interaction level of the transmission. In the 1xEV-DV [12] traffic model, FTP and the video and audio streaming have longer reading time (thinking time). As discussed in Chapter 3, because low uplink-and-downlink ratio applications apply to FTP and the video and audio streaming and higher uplink-and-downlink ratio traffics apply to voice and HTTP, we may think that the high interactive level, like voices and HTTP, has a negative correlation to the delay between the packet calls.

In general, the packet call size can predict the length of the information content for the same application and the specific data applications. In other words, the more information we have, the more reading time we might need to spend. In this case, the thinking time has a positive correlation to the packet call size. For a specific data application, short message service (SMS), the packet-call size is small and can easily be predicted from packet call size measurement.

Furthermore, the channel usage can also be considered as the reference of the average transmission size over a TCP connection and the protocol type. By knowing the protocol type, different the protocol type has different relationship of the thinking time. For example, the video streaming has a long think time. On the other hand, the web browsing which uses HTTP has a smaller think time.

4.3 Dormancy Estimation Table

A 3-dimensinal decision metric based on three measurement indexes is created to calculate the proper dormancy timer. The 3-D metric records the value of the average "thinking" time, T_{mean}(n), and the standard deviation of the thinking time, T_{std}(n), at nth update. The T_{mean}(n) and T_{std}(n) values will be updated on every packet call. The updated element T^{i}_{mean} and T^{i}_{std} is chosen based on the category *i* which is classified by the uplink-and-downlink ratio, channel utilization, and packet call size. The chosen element is updated based on an IIR filter:

$$T_{mean}(n+1) = T_{mean}(n) \times (1-a) + T_{think} \times a \qquad (1)$$

$$\begin{aligned} T_{\mathrm{var}}(n+1) &= T_{\mathrm{var}}(n) \times (1-b) + (T_{think} - T_{mean})^2 \times b \\ T_{std}(n+1) &= \sqrt{T_{\mathrm{var}}} \end{aligned} \qquad (2)$$

where *a and b* are less than one and T_{think} is the current thinking time (inter-arrival time). The updated T_{mean}(n+1) and T_{std}(n+1) will be restored in the decision metric for next data application with the same category.

The system will set the default values for the metric. The values will be updated when the user uses the mobile. After a long time training for the values, the elements in the metric will be closed to the user behavior.

In Table 4-1, the table is 3-dimensinal table (3 x 4 x 2). There are 24 different T_{mean} in the table, where T^{x}_{mean} is the T_{mean} for category x. The role values are packet call size and the column values are channel usage. In the intersection point of each the column value and the row value, there are two values of ratio. For example, T^{3}_{mean} is the T_{mean} category 3 in which the uplink and downlink ratio is smaller than 50%, the packet call size is smaller than 100K bytes, and the channel usage is greater than 60%~80%.

Table 4-1, Example for dormancy estimation table

Packet call	Ratio1	Channel usage			
size (bytes)	Ratio2	<40%	40%~60%	60%~80%	>80%
<100K	<50%	T^{1}_{mean}, T^{1}_{std}	T^{2}_{mean}, T^{2}_{std}	T^{3}_{mean}, T^{3}_{std}	T^{4}_{mean}, T^{4}_{std}
	>=50%	T^{5}_{mean}, T^{5}_{std}	T^{6}_{mean}, T^{6}_{std}	T^{7}_{mean}, T^{7}_{std}	T^{8}_{mean}, T^{8}_{std}
100K~1000K	<50%	T^{9}_{mean}, T^{9}_{std}	T^{10}_{mean}, T^{10}_{std}	T^{11}_{mean}, T^{11}_{std}	T^{12}_{mean}, T^{12}_{std}
	>=50%	T^{13}_{mean}, T^{13}_{std}	T^{14}_{mean}, T^{14}_{std}	T^{15}_{mean}, T^{15}_{std}	T^{16}_{mean}, T^{16}_{std}
>100K	<50%	T^{17}_{mean}, T^{17}_{std}	T^{18}_{mean}, T^{18}_{std}	T^{19}_{mean}, T^{19}_{std}	T^{20}_{mean}, T^{20}_{std}
	>=50%	T^{21}_{mean}, T^{21}_{std}	T^{22}_{mean}, T^{22}_{std}	T^{23}_{mean}, T^{23}_{std}	T^{24}_{mean}, T^{24}_{std}

4.4 Adaptive Dormancy Timer

The resource consumptions are calculated by

$$R_{total} = r_{setup} \times t_{setup} + r_{tran} \times t_{tran} + r_{hold} \times t_{hold} \tag{3}$$

where R_{total} is the total resource consumption for a mobile. r_{setup} is the average required resource when a mobile establishes a new connection. r_{tran} is the average required resource when a mobile transmit packets. r_{hold} is the average required resource in the control hold state. The t_{setup} is the total duration in the setup condition. The t_{tran} is the total duration in the transmission state. The t_{hold} is the total duration in the control hold state.

The t_{tran} depends on the packet call size, the transmission bandwidth, the scheduling delay, and etc. To decide (or estimate) the dormancy timer, following rules are applied. Here, without considering the resources used in the data transmission, the equation (3) can be rewritten as

$$R^{2}{}_{total} = r_{setup} \times t_{setup} + r_{hold} \times t_{hold} \tag{4}$$

With the dormancy trigger, the t_{hold} can be shortened as the $t_{dormant}$ (the dormancy timer). In other words, instead of holding the resources continuously, an early active-to-dormant trigger can prevent the excessive use of the resources. At a cost of potential reactive process, the proposed algorithm should try to minimize the resources used for both the setup process and the dormancy process for each packet call. The required resource when the dormancy timer is $t_{Dormant}$ is shown below

$$R_{Dormant} = \begin{cases} r_{hold} \times t_{Dormant} + r_{setup} \times t_{setup} & if\ t_{think} > t_{Dormant} \\ r_{hold} \times t_{think} & otherwise \end{cases} \tag{5}$$

For N packet calls, the required resources can be calculated as :

$$R_{Dormant} = \frac{M(r_{setup} \times t_{setup} + r_{hold} \times t_{Dormant}) + (N - M)(r_{hold} \times t_{think})}{N} \tag{6}$$

where $R_{Dormant}$ is the total resource consumption with the dormancy timer of $t_{Dormant}$. T_{think} is the thinking time between two packet calls. N is total packet-call numbers. M is the packet-call numbers which need to re-establish the connection.

There are two different conditions. For multiple packet calls, when the actual thinking time between packet calls is larger than the dormancy timer, the required resources are $r_{hold} \times t_{Dormant} + r_{setup} \times t_{setup}$ (including the re-establish of a connection). In the other case, when the user holds the channel for the entire inter-arrival time, the required resource is $r_{hold} \times t_{think}$. An early active-to-dormant should be triggered if $r_{hold} \times t_{Dormant} + r_{setup} \times t_{setup}$ is smaller than $r_{hold} \times t_{think}$. From the above discussion, to decide the dormancy timer, following rules are applied:

$$t_{Dormant} = \begin{cases} T_{Min} & If\ T_{mean} - c \times T_{std} > T_{Min} + e \times t_{SETUP} \\ T_{Min} & If\ T_{mean} < T_{Min} \\ T_{mean} + d \times T_{std} & otherwise \end{cases} \qquad (6)$$

Where T_{Min} is a minimum dormancy timer in the system. The parameter *c* and *d* in the equation (6) will be changed in order to achieve the performance requirement. The parameter *e* is calculated as follows: the resource consumption by releasing the channel at T_{Min} is $r_{hold} \times T_{Min} + r_{setup} \times T_{setup}$ and the resource consumption maintaining the channel is $r_{hold} \times t_{think}$. When $T_{Min} + {}^{r_{setup}}\!/_{r_{hold}} \times t_{setup} \le t_{think}$, the system can get benefits by releasing the channel at T_{Min}. So the parameter *e* is the equal to ${}^{r_{setup}}\!/_{r_{hold}}$.

From equation (6), the final dormancy timer will become the minimum dormancy timer if the estimated interval of two packet calls is either less the T_{Min} or larger than the $T_{Min} + e \times T_{setup} + c \times T_{std}$. Otherwise, the dormancy timer is $T_{mean} + d \times T_{std}$. The design philosophy is the system should terminate the connection and release the channel resource when the thinking time is long enough. On the other case, the dormancy timer should be chosen to prevent the excessive reactivate triggers. From the Figure 4-1, when the thinking time is long enough and the system will gain more recourses from the fast termination of channels. In this condition, the areas in the right side of the threshold ($T_{mean} - c \times T_{std}$) in the Figure 4-1 are probability of the accurate prediction. By choosing c parameter, the system can choose the percentage of the delay in the right side of the $T_{Min} + e \times T_{SETUP}$. In the other condition, the system wants to provide an accurate dormant timer for the connection. By the accurate timer, the system will reduce the consumption either holding the channel too long or entering the dormant state too early and wasting additional resource. We want to make the $T_{Dormant}$ to cover as many areas as possible. By choosing d parameter in the Figure 4-1, the system can decide the percentage of the accurate prediction of the inter-arrival time. The longer $T_{Dormant}$ can make a higher percentage of the accurate prediction, but it has an additional overhead to increase the active time. It's a trade off problem to choose d parameter value.

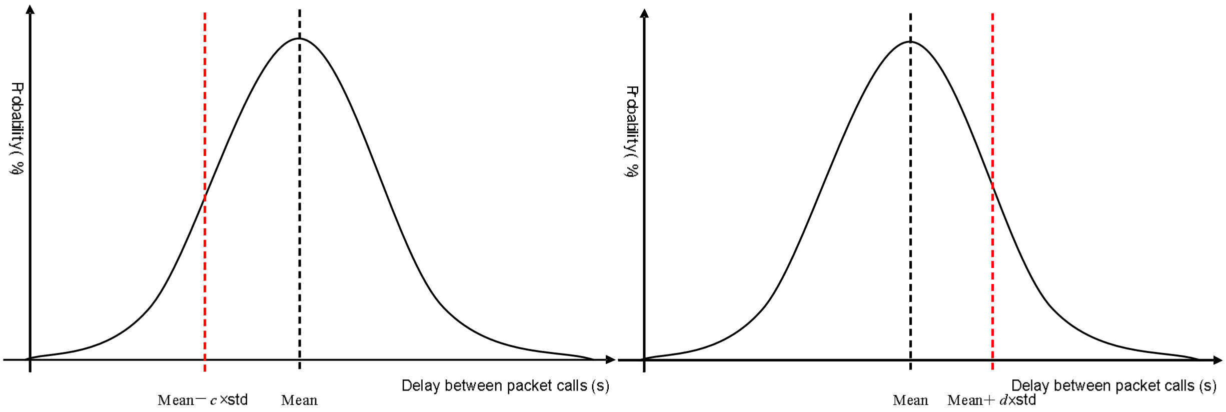

Figure 4-1, Delay Distribution

Chapter 5 Simulation Model and Performance Metric

The simulation procedure is introduced and the parameters values, such as the file size and the assumption timer are defined. Finally, the performance metric is proposed in the chapter.

5.1 Simulation Model and Flow

The simulation flow is show in the Figure 5-1. The simulation result is calculation for the average of one hundred traffics. First, the traffic is generating based on the traffic model provided by 1xEVDV [12]. Based on Table 5-1, Table 5-2, Table 5-3, and parameter assumptions on Table 5-4, in the traffic generator block, the block generates the data traffics (HTTP or FTP) with various MTU sizes, and then it generates the packets and the inter-arrival time between packet calls. Besides the two original traffics, the modify Ftp is created to test the algorithm with the condition when the data applications are the same and packet call sizes are different. The modify ftp is also to identify the performance for the small thinking time.

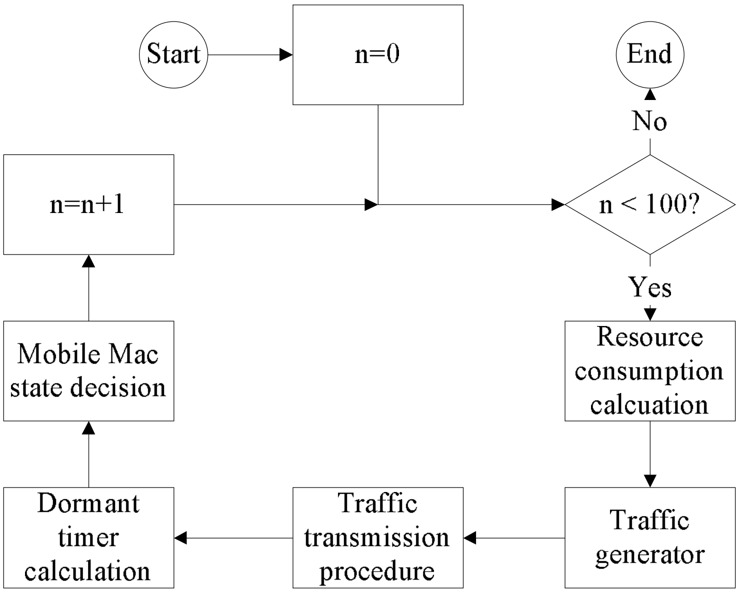

Figure 5-1, simulation flow of the algorithm

After the traffic generator, the simulation goes into the traffic transmission procedure. The block will run TCP protocol to transmit the traffic. A slow-start and three way handshake procedures are considered in the simulation. For HTTP protocol, the download procedure of is based on the HTTP/1.0-Burst mode or HTTP/1.1-Persistent mode [12]. For FTP protocol, the FTP protocol is used for downloading files [12]. After all packets have been transmitted, the uplink and downlink ratio, the packet call size, and the channel usage will be calculated from the history of the traffic and then goes into the next step, the dormant timer calculation. The dormant timer is calculated based on the Figure 5-2. First, use the uplink and downlink ratio, the packet call size, and the channel usage to categorize and then update the values of T_{mean} and T_{std} by equation (1) and (2) in the 3-dimensinal decision metric. After having the proper T_{mean} and T_{std}, the $T_{Dormant}$ is calculated form the equation (6). Finally, based on the assumptions, listed in Table 5-4, the required resources for all packet calls are calculated and will be discussed in the next section.

Table 5-1, HTTP Traffic Model

Component	Distribution	Parameters	PDF
Main object size (S_M)	Truncated Lognormal	Mean=10710 bytes Std=25032 bytes Min=100 bytes Max=2 Mbytes	$f_x = \frac{1}{\sqrt{2p}\, sx} \exp\left[\frac{-(\ln x - m)^2}{2s^2}\right]$ $x \ge 0,\ s = 1.37,\ m = 8.35$
Embedded object size (S_E)	Truncated Lognormal	Mean=7758 bytes Std=126168 bytes Min=50 bytes Max=2 Mbytes	$f_x = \frac{1}{\sqrt{2p}\, sx} \exp\left[\frac{-(\ln x - m)^2}{2s^2}\right]$ $x \ge 0,\ s = 2.36,\ m = 6.17$
Number of embedded objects per page (N_d)	Truncated Pareto	Mean=5.64 Max=53	$f_x = \frac{a k^a}{x^{a+1}},\ k \le x < m$ $f_x = \left(\frac{k}{m}\right)^a,\ x = m$ $a = 1.1,\ k = 2,\ m = 55$
Reading time (D_{pc})	Exponential	Mean=30 sec	$f_x = \lambda e^{-\lambda x},\ x \ge 0$ $\lambda = 0.033$
Parsing time (T_p)	Exponential	Mean=0.13 sec	$f_x = \lambda e^{-\lambda x},\ x \ge 0$ $\lambda = 7.69$

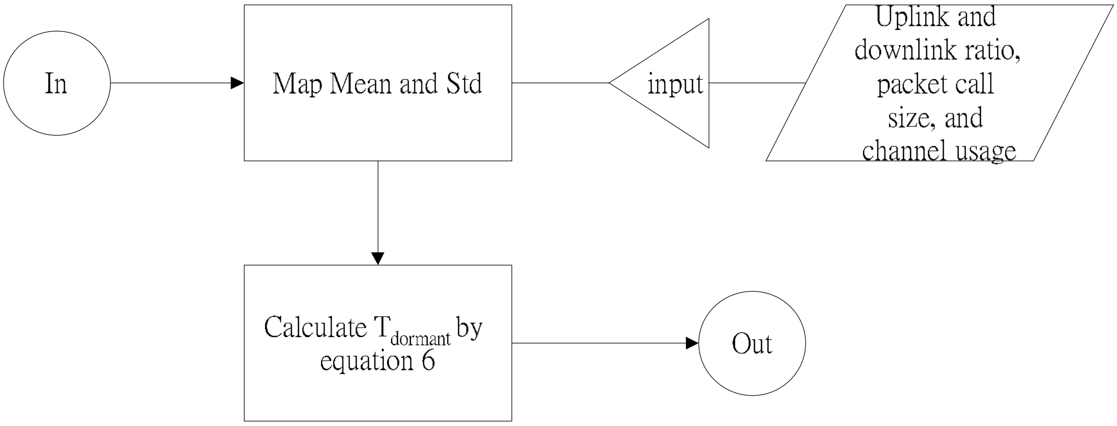

Figure 5-2, Dormancy timer calculation flow

Table 5-2, FTP Traffic Model

Component	Distribution	Parameters	PDF
File size (S)	Truncated Lognormal	Mean=2Mbytes Std=0.722 Mbytes Max=5 Mbytes	$f_x = \frac{1}{\sqrt{2p}\,sx}\exp\left[\frac{-(\ln x - m)^2}{2s^2}\right]$ $x \ge 0,\ s = 0.35,\ m = 14.45$
Reading time (D_{pc})	Exponential	Mean = 180 sec.	$f_x = l e^{-lx}, x \ge 0$ $l = 0.0083$

Table 5-3, Modify Ftp Traffic Model

Component	Distribution	Parameters	PDF
File size (S)	Truncated Lognormal	Mean=8486bytes Std=3210bytes Max=5 Mbytes	$f_x = \frac{1}{\sqrt{2p}\,sx}\exp\left[\frac{-(\ln x - m)^2}{2s^2}\right]$ $x \ge 0,\ s = 0.35,\ m = 9.0$
Reading time (D_{pc})	Exponential	Mean = 10 sec.	$f_x = l e^{-lx}, x \ge 0$ $l = 0.1$

Table 5-4, Assumption simulation values

Component	Value (s)
a and *b* in the equation (2)	0.1, 0.1
c, *d*, and *e* in the equation (3)	1, 0.5, 2
The minimum dormant timer	1.0~5.0
The delay for the channel setup up	1.0
Network Delay and Queuing Delay	2.0
The default dormant timer without algorithm	5.0~15.0

5.2 The Performance Evaluation

The performance is evaluated in terms of the power consumption, the channel resource usage, and total transmission delay between packet calls. The system performance is analyzed from three parts, the active mode part, the control hold mode part, and setup channel part. The performance matrix is defined in Table 5-5 will be used to quantify the proposed ADT scheme.

Table 5-5, Performance Matrix

Component	Value
Power consumption	$\frac{(P_{tran} \times T_{tran} + P_{hold} \times T_{hold} + T_{setup} \times P_{setup})_{with\ algorithm}}{(P_{tran} \times T_{tran} + P_{hold} \times T_{hold} + T_{setup} \times P_{setup})_{without\ algorithm}}$
Channel resource usage	$\frac{(Ch_{tran} \times T_{tran} + Ch_{hold} \times T_{hold} + Ch_{setup} \times T_{setup})_{with\ algorithm}}{(Ch_{tran} \times T_{tran} + Ch_{hold} \times T_{hold} + Ch_{setup} \times T_{setup})_{without\ algorithm}}$
Transmission Delay	$\frac{(\text{Thinking time} + \text{Setup Time} + \text{Network Delay})_{with\ algorithm}}{(\text{Thinking time} + \text{Setup Time} + \text{Network Delay})_{without\ algorithm}}$

From Table 5-5, the P_{tran} and Ch_{tran} are the average power consumption and the average channel resource usage. The P_{hold} and Ch_{hold} are the average power consumption and the average channel resource usage in the control hold state, and the P_{setup} and Ch_{setup} are the average power consumption and the average channel resource usage for a call setup. The T_{tran}, T_{hold}, and T_{setup} are the wasting time for each different process. The assumption set of values is shown in Table 5-6. We take the power consumption and the required channel resource in the transmission condition as one unit. The power consumption and the required channel resource in the other conditions are normalized to the transmission condition.

Table 5-6, parameter value of the simulation

	hold	setup	tran
P	0.125	1	1
Ch	1	1	1

Chapter 6 The Simulation Results

In this chapter, the impacts on the required resources are studied. Figure 6-1, the proposed ADT reduces the resources significantly. As shown, there are 50% saving for the channel resource usage and 25% saving for the power consumption. Also, as expected, for the transmission delay, this is slight more consumption when the algorithm is implemented. The saving can be further increase for larger packet-call size like FTP shown in Figure 6-2. But when the packet-call size is small, Figure 6-3 shows that saving in power consumption and transmission delay are small and the required channel resources are higher than no ADT case.

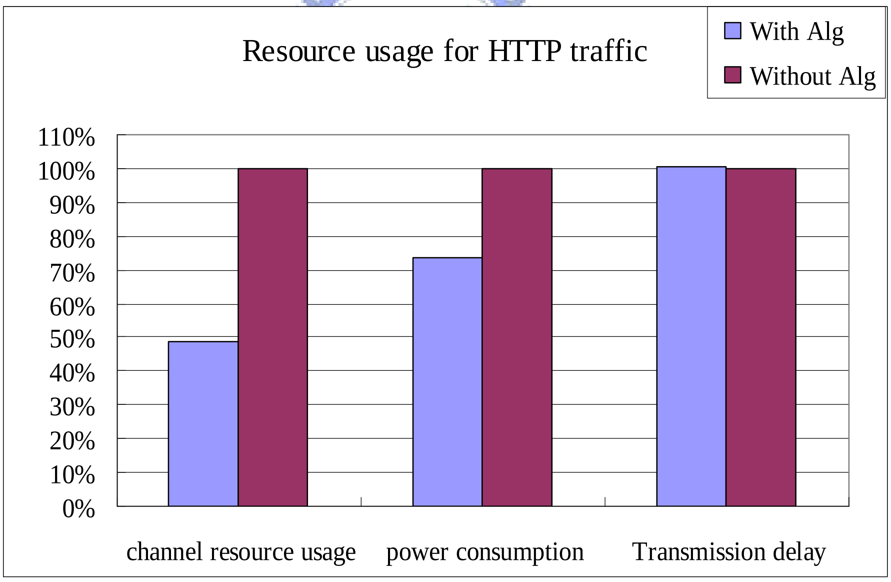

Figure 6-1, Resource usage for HTTP traffic

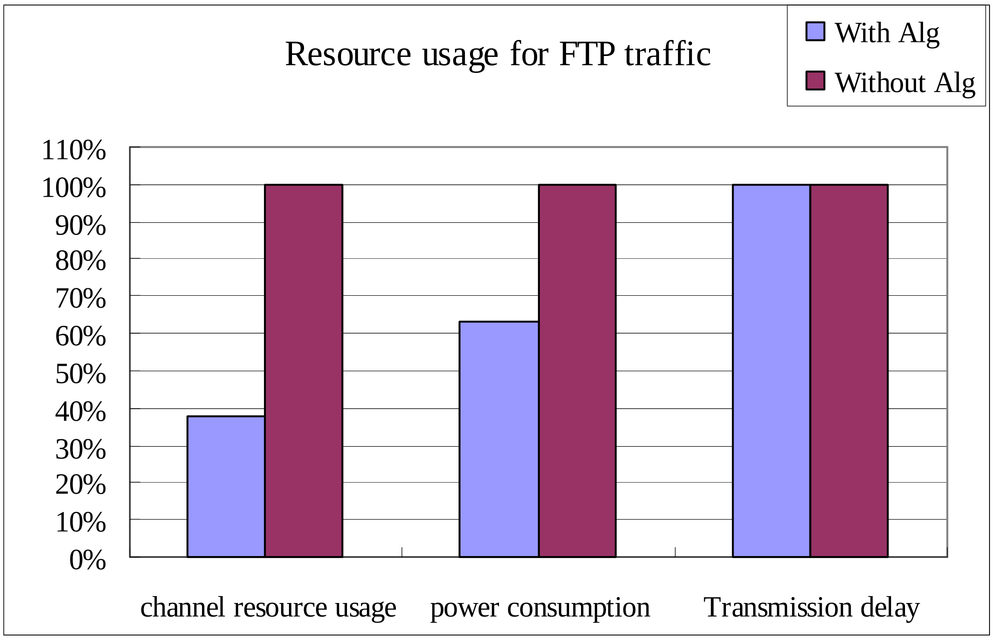

Figure 6-2, Resource usage for FTP traffic

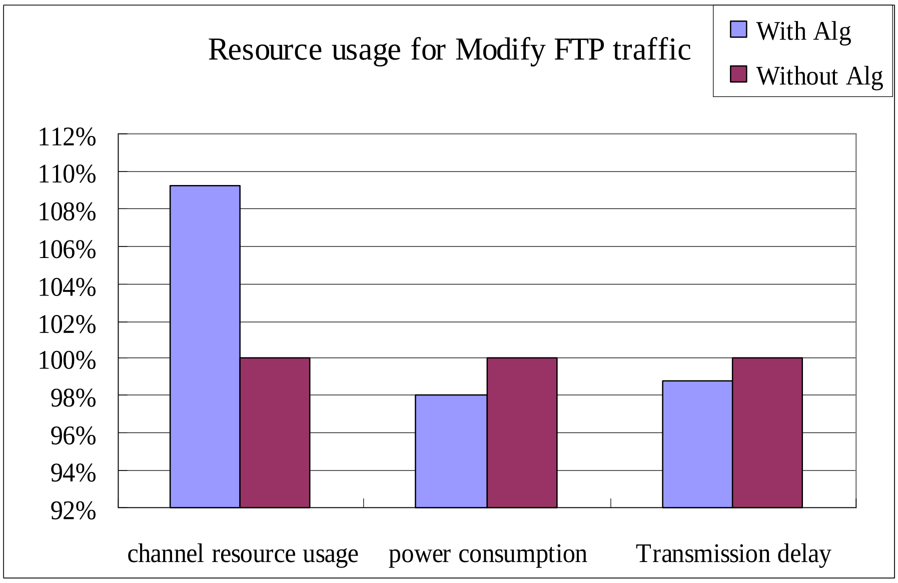

Figure 6-3, Resource usage for modify ftp traffic

The system gains benefits from fast releasing of the channel or has a proper holding time. From Table 5-1, the mean value of the inter-arrival time is 30 seconds. Because it has a long thinking time, the channel can be released earlier. From Figure 6-1, the power consumption decreases about 25% and the required channel resources decrease by 50%. In Figure 6-2, the mean value of the inter-arrival time is 180 seconds. Because it has a very long thinking time, the results with the algorithm will release the channel at T_{Min} and then re-establish for another new packet call. In this case an easily active-to-dormant trigger can benefit the required resources. On the contrary, with small packet-call size (the modified FTP), from Figure 6-3, the mobile can get slight saving on the power resources with less transmission delay, but the channel required resources are increased. The mean value of the inter-arrival time is not long enough to release the channel quickly. The mobile will decide to hold the channel. By the algorithm, the mobile can wait for an accurate dormancy timer and eliminate unnecessary channel setup. The power consumption and transmission delay are decreased because the channel re-establishments are decreased. But the longer channel holding time makes the required channel resources to be increased.

The evaluation of the relationship between the inter-arrival time and the resource is studied. From Figure 6-4, the transmission delay is the smallest when the inter-arrival time is about 10 seconds. This is because the system decides to hold the resource and the timer is more accurate compared to the default timer of 10 seconds. When the inter-arrival time is smaller than the default timer and if the ADT is also smaller than the default timer, the adapt dormancy timer scheme might increase the numbers of the setup process, that will increase the total transmission time. When the inter-arrival time is larger than the default value, the channel will be released quickly. The reactivate process will also increase the transmission delay. Form the Figure 6-5, and Figure 6-6, the system can save more power and resources for larger the average inter-arrival time. That is due to an early active-to-dormant trigger is activated. As shown, the saving increases when the inter-arrival time is equal to about 15 seconds and larger.

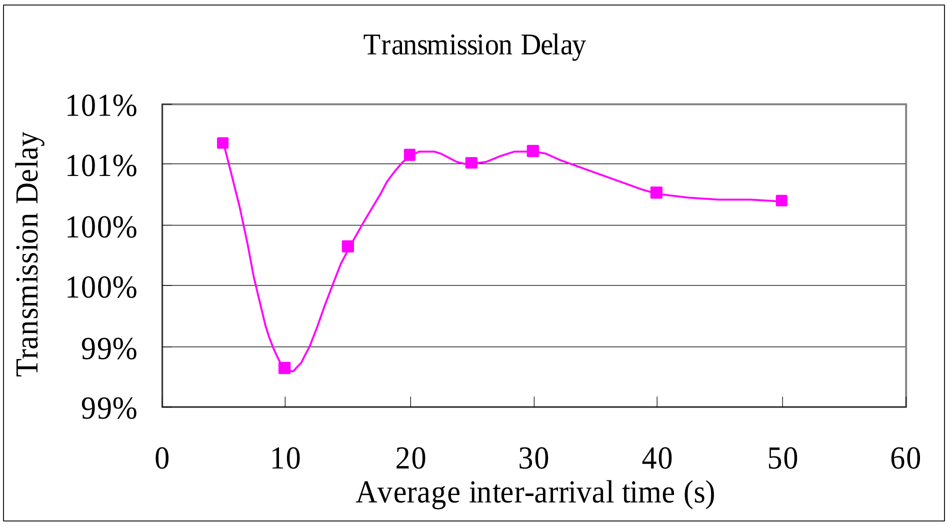

Figure 6-4, Reverse access channel usage for various inter-arrival time

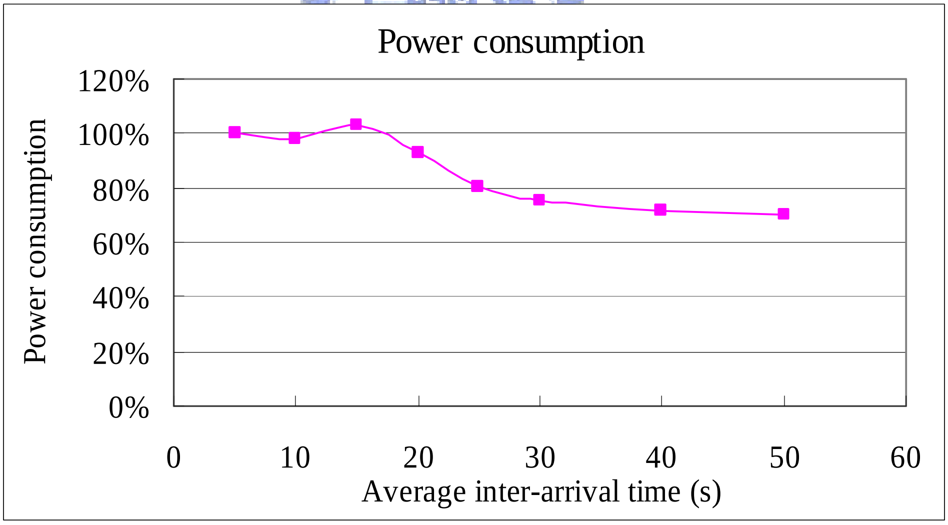

Figure 6-5, power consumption for various inter-arrival time

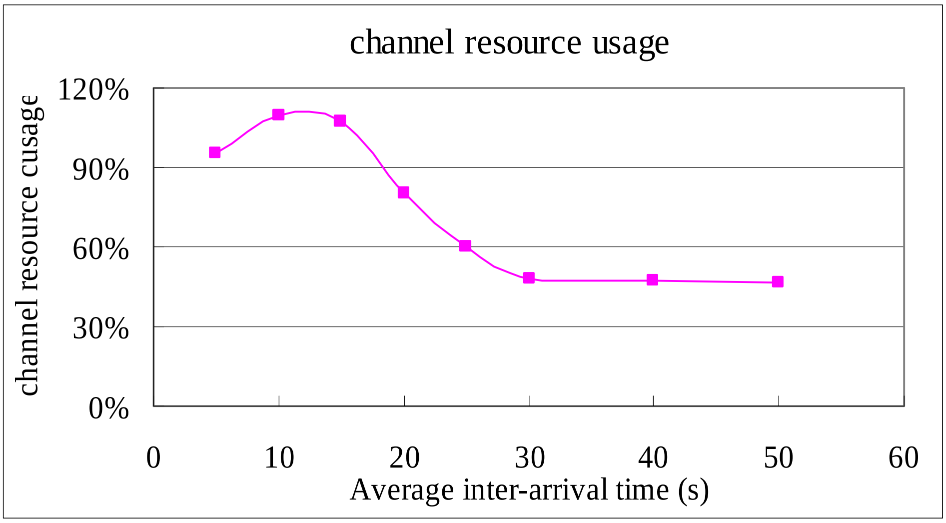

Figure 6-6, channel resource consumption for various inter-arrival time

With various default dormancy timer and different inter-arrival time, all performances are evaluated in Figure 6-7 to Figure 6-9. For the transmission delay, then the inter-arrival time increases, the delays with different dormancy timers converge to the delay with implementing the ADT. For the power consumptions and the required channel resources, they have the similar distances for different default dormancy timer when the inter-arrival is large. A relative big change happens when the inter-arrival in between 5 and 15 seconds for all of them.

The delays converge to the delay with implementing the ADT because the connections are almost released and re-established in the next packet call when the inter-arrival time is larger. The transmission delay with different dormancy timer will be no different. The power consumptions and the required channel resources are similar (relative to the dormancy timer) whether the default dormancy timer is larger or not. When the inter-arrival time is smaller than 15 seconds, the system will usually decide to hold the resource. With the inter-arrival time is 5 seconds, the dormancy timer with ADT will slight larger than 5 seconds. The transmission delay with the default dormancy timer of 5 seconds is larger than with algorithm. This is the same as the inter-arrival time of 10 seconds. When the inter-arrival time is about 15 seconds, the system will decides to release the connection quickly. But there are some cases to hold the connection. So the performance will between holding the connection and releasing it quickly.

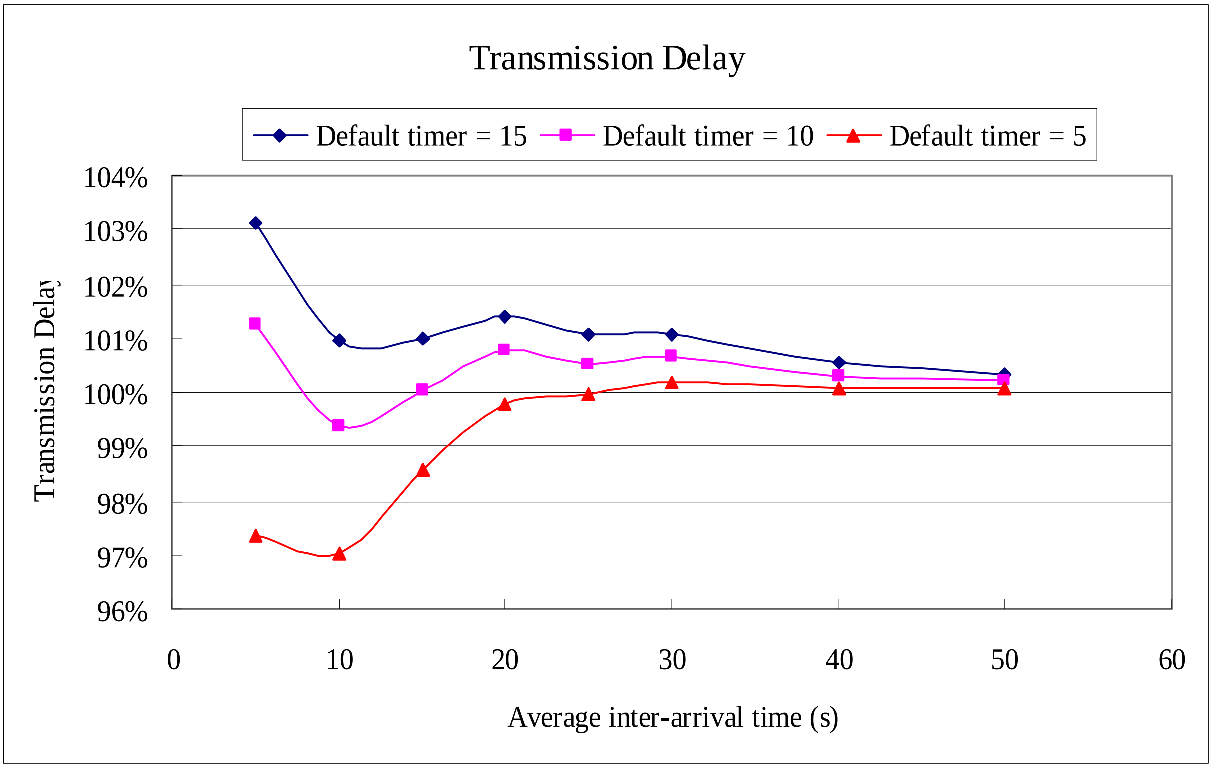

Figure 6-7, RACH occupancy for different default dormancy timer

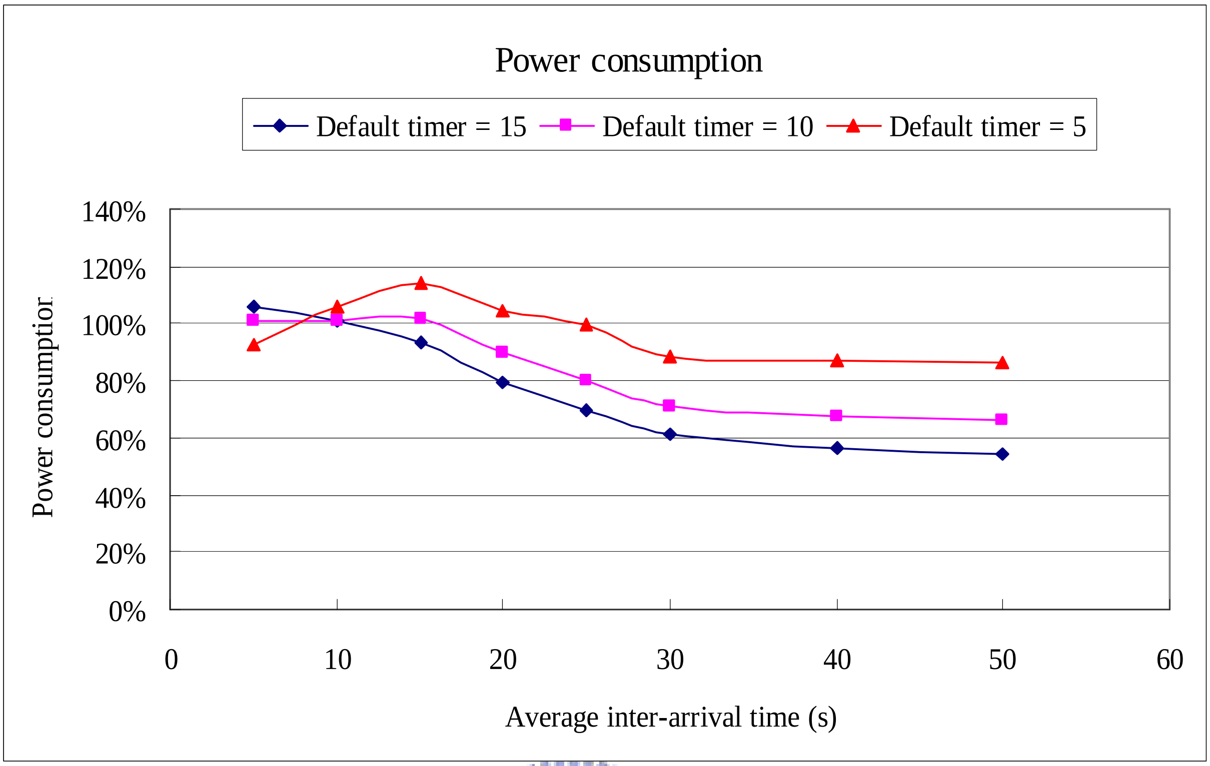

Figure 6-8, Power resource for different default dormancy timer

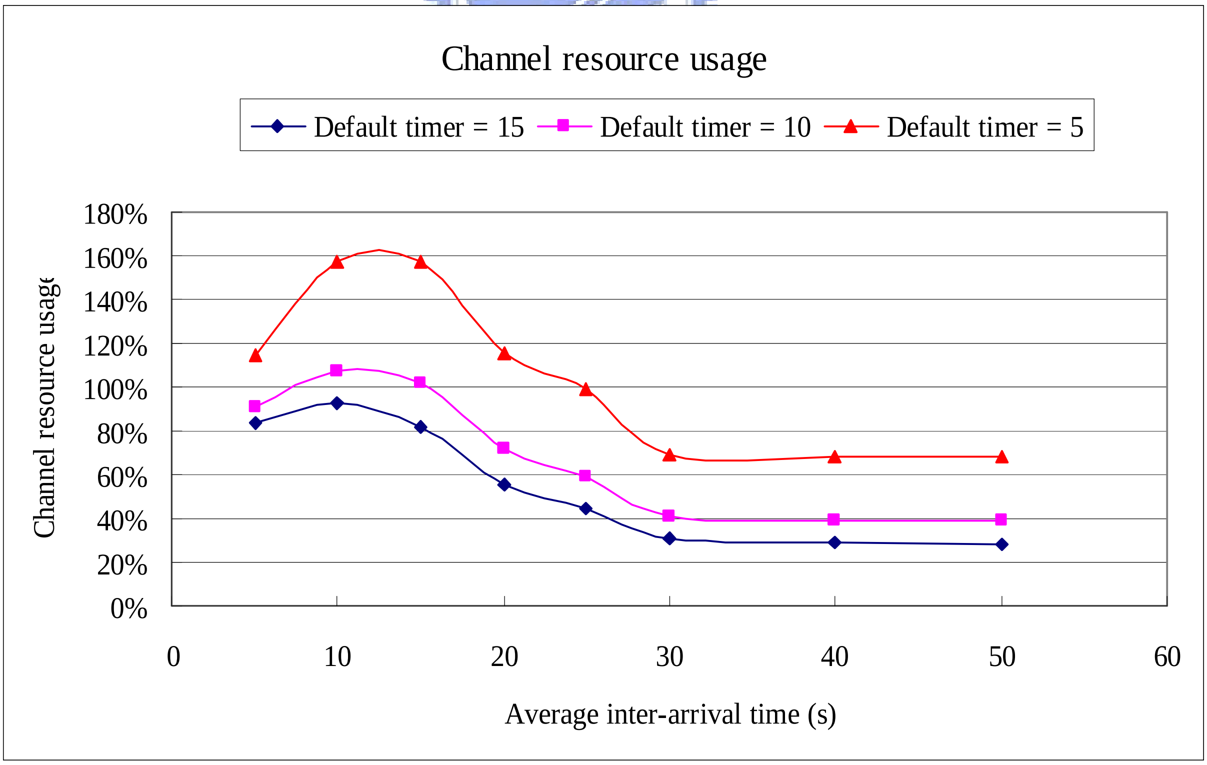

Figure 6-9, Channel resource usage for different default dormancy timer

The analysis of the parameter c in equation (6) is show below. From equation (6), the larger parameter c lets the system to decide to release the channel harder. Because the mobiles hold the channel for longer time, the power consumption and the required channel resource has less improvement. But the transmission delay has some improvements. Besides, the parameter only affects the system when the inter-arrival time is small and the system decides to release the channel.

From Figure 6-10 to Figure 6-12, the line with the value of the parameter c equaling to 0.75 and the line with the value of the parameter c equaling to 1.0 overlap when the inter-arrival time is larger than 30 seconds. he line with the value of the parameter c equaling to 1.0 and the line with the value of the parameter c equaling to 1.25 overlap when the inter-arrival time is larger than 40 seconds. The larger value of the parameter c lets the point of releasing the channel larger. From Figure 6-10, the savings of the transmission delay are more when the value of the parameter c is larger because the system has higher probability to hold the channel. Figure 6-11 and Figure 6-12, the savings of the power consumption and the required channel resource are smaller when the value of the parameter c is larger because of longer average channel holding time.

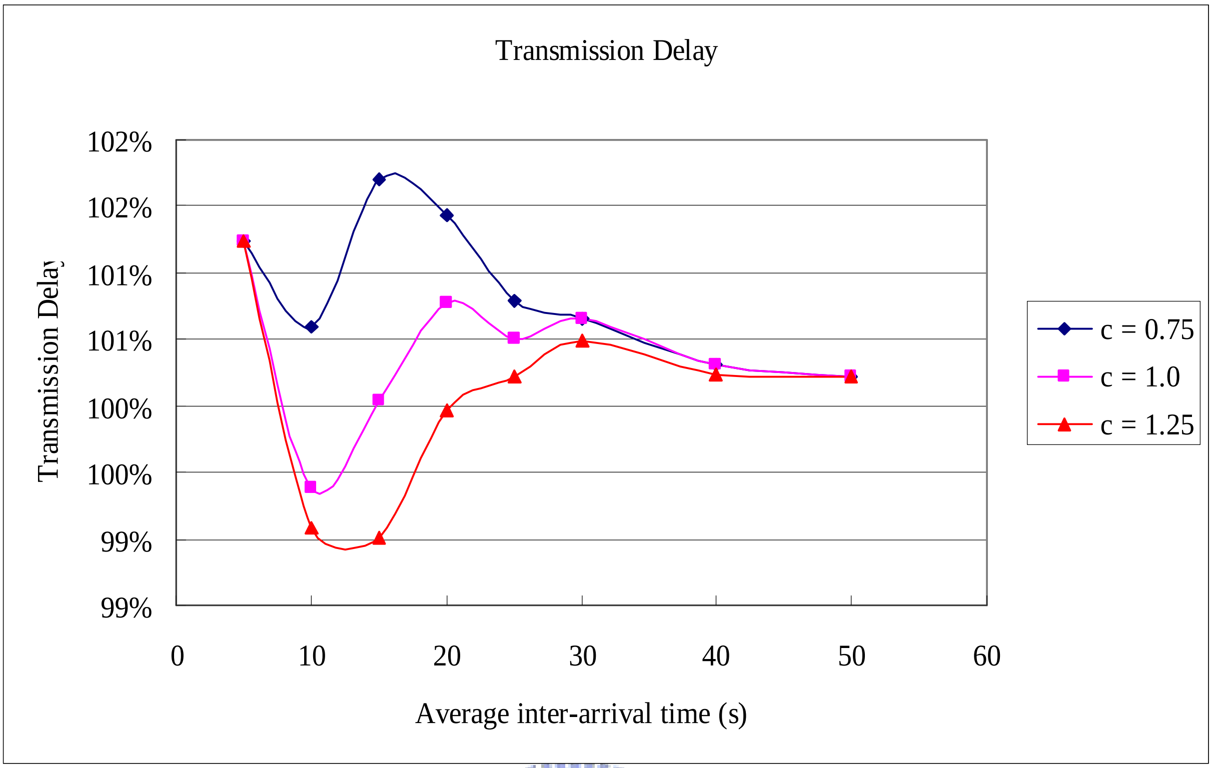

Figure 6-10, Transmission delay for various parameter *c*

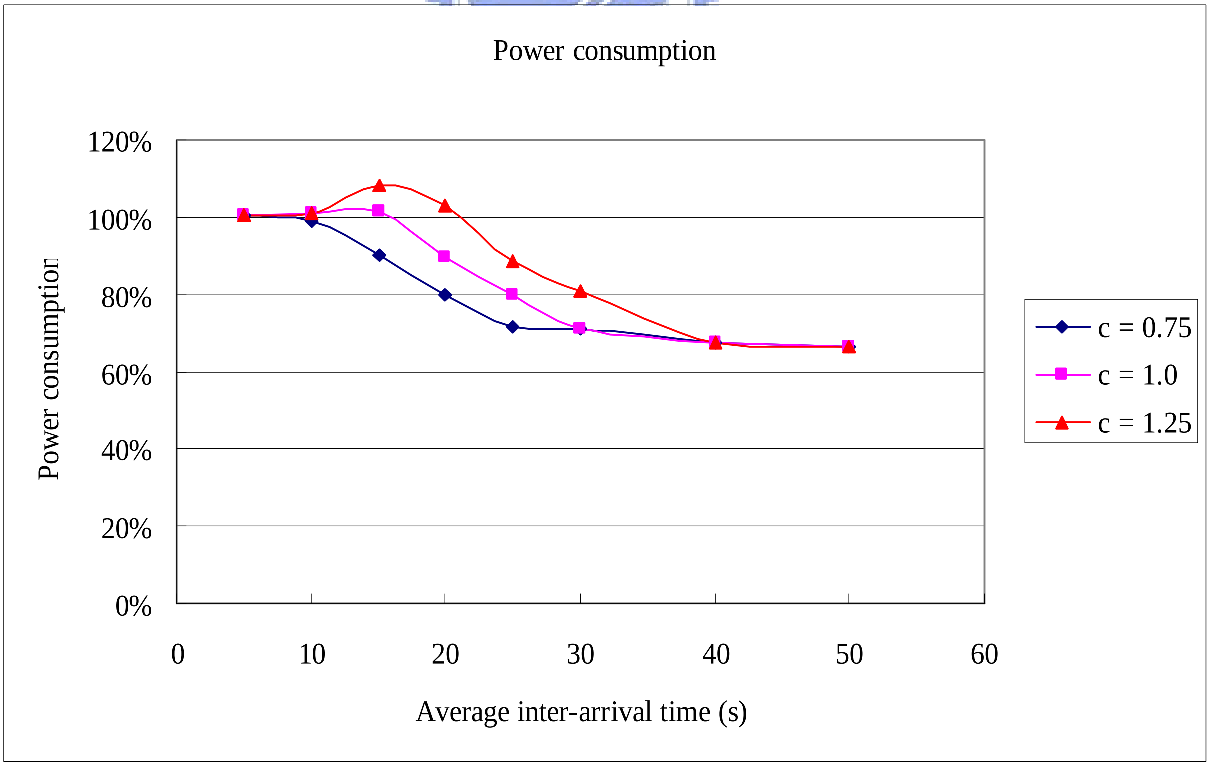

Figure 6-11, Power consumption for various parameter *c*

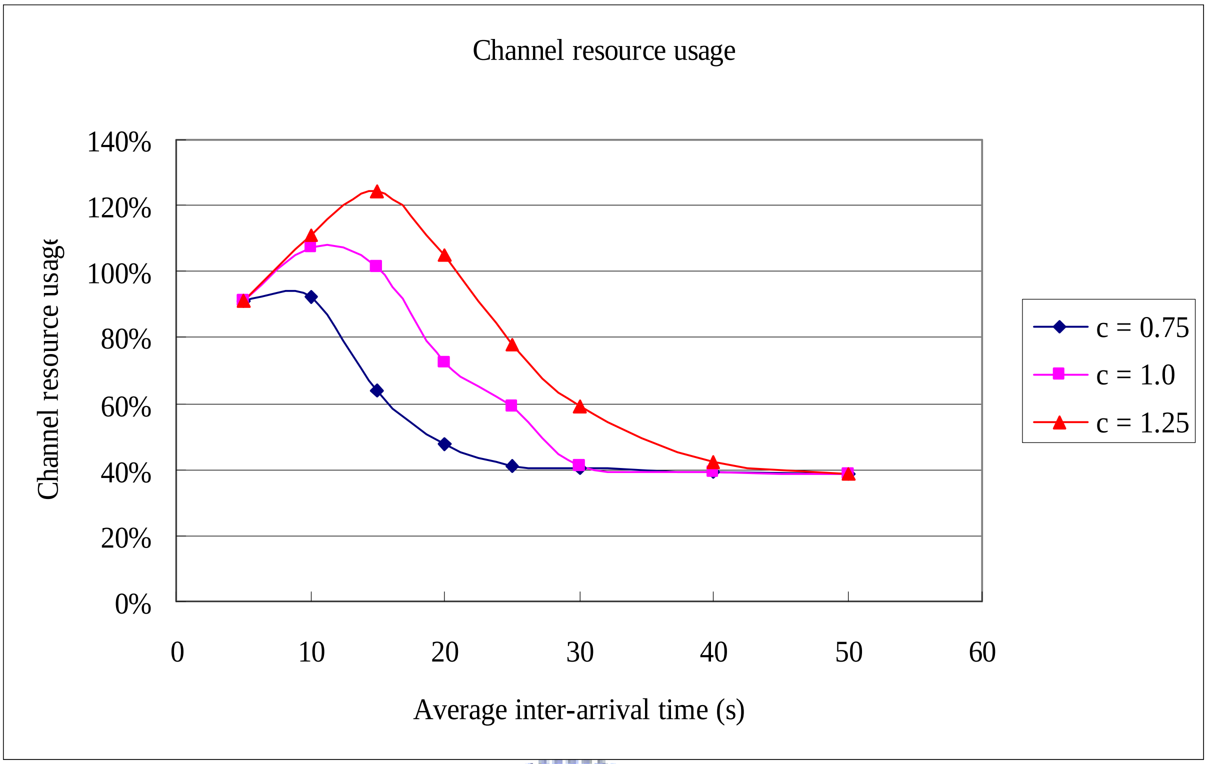

Figure 6-12, Channel resource usage for various parameter *c*

The analysis of the parameter ***d*** in equation (6) is show below. Larger ***d*** decreases the setup times because the system holds the resources for longer time. The parameter affects the performance only when the inter-arrival time is not larger enough.

From Figure 6-13 and Figure 6-14, smaller ***d*** will save slight the channel and power resource by holding the resource for shorter period when the inter-arrival time is smaller than 25 seconds. From Figure 6-15, because the larger value of the parameter ***d*** decreases the probability for re-establishing the channel and decreases the transmission delay, the transmission delay has improvement for larger the value of the parameter ***d***.

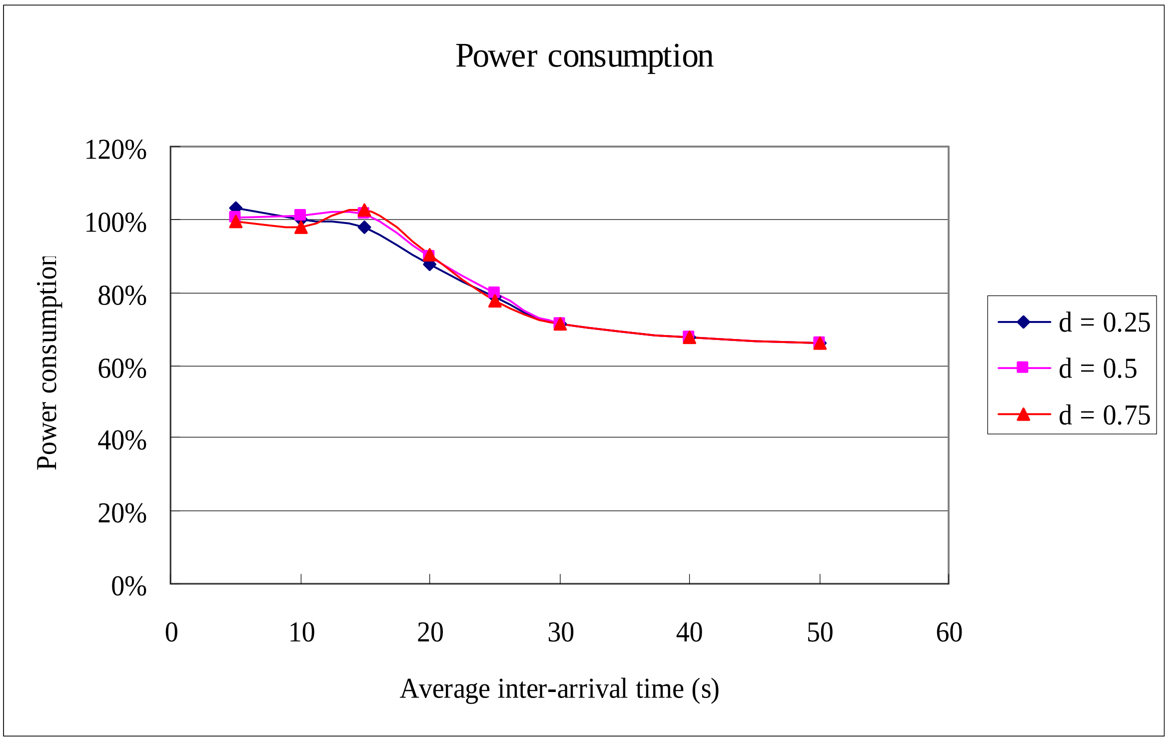

Figure 6-13, Power consumption for various parameter ***d***

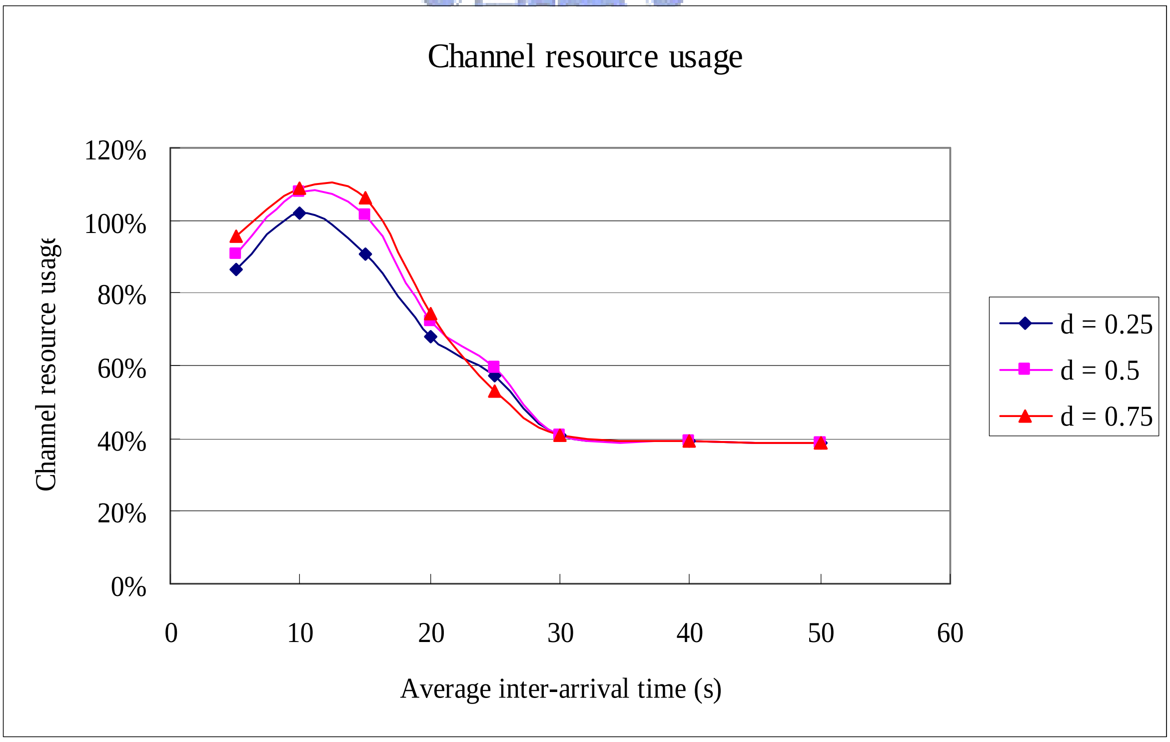

Figure 6-14, Channel resource usage for various parameter ***d***

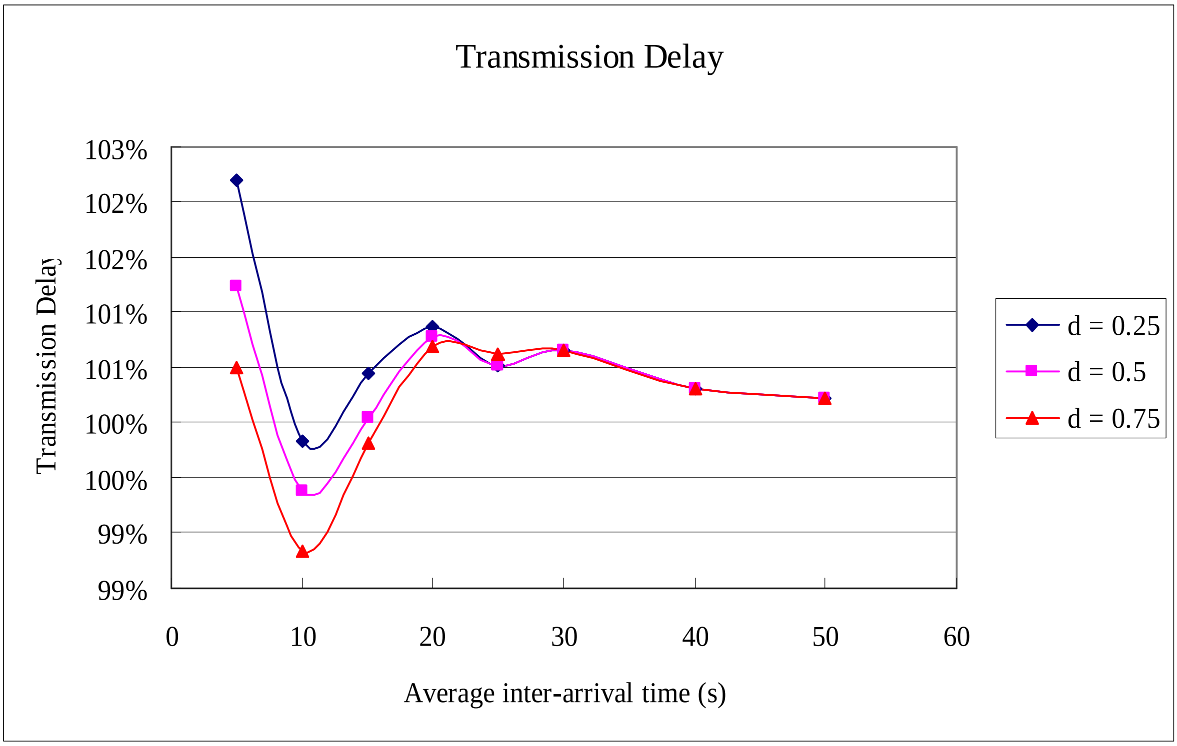

Figure 6-15, Transmission delay for various parameter ***d***

The analysis of the parameter ***e*** in equation (3) is show below. Larger value of the parameter ***e*** makes the system to release the resource more hardly. So the results are similar as the results for the lager value of the parameter ***c***.

From Figure 6-16 and Figure 6-17, the power consumption and the channel usage will be lower when ***e*** is smaller because of releasing the channel more easily. From Figure 6-18, smaller ***e*** has larger transmission delay.

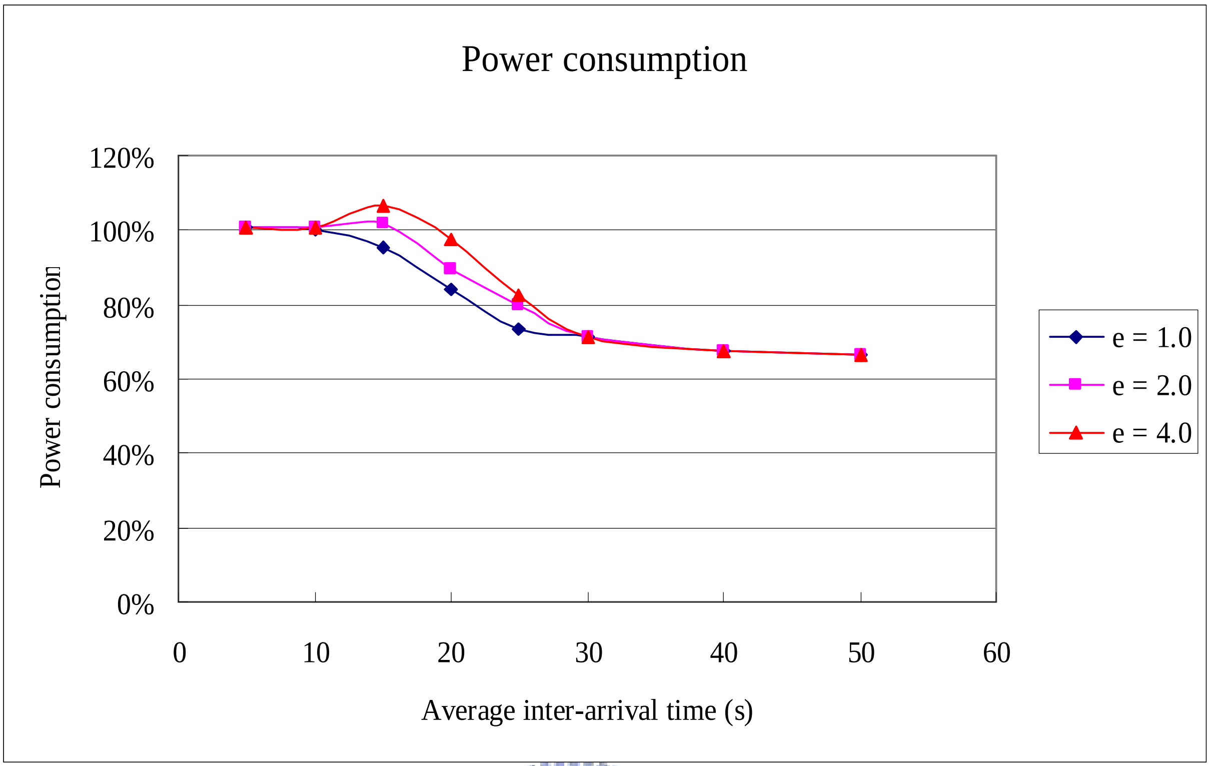

Figure 6-16, Power consumption for various parameter *e*

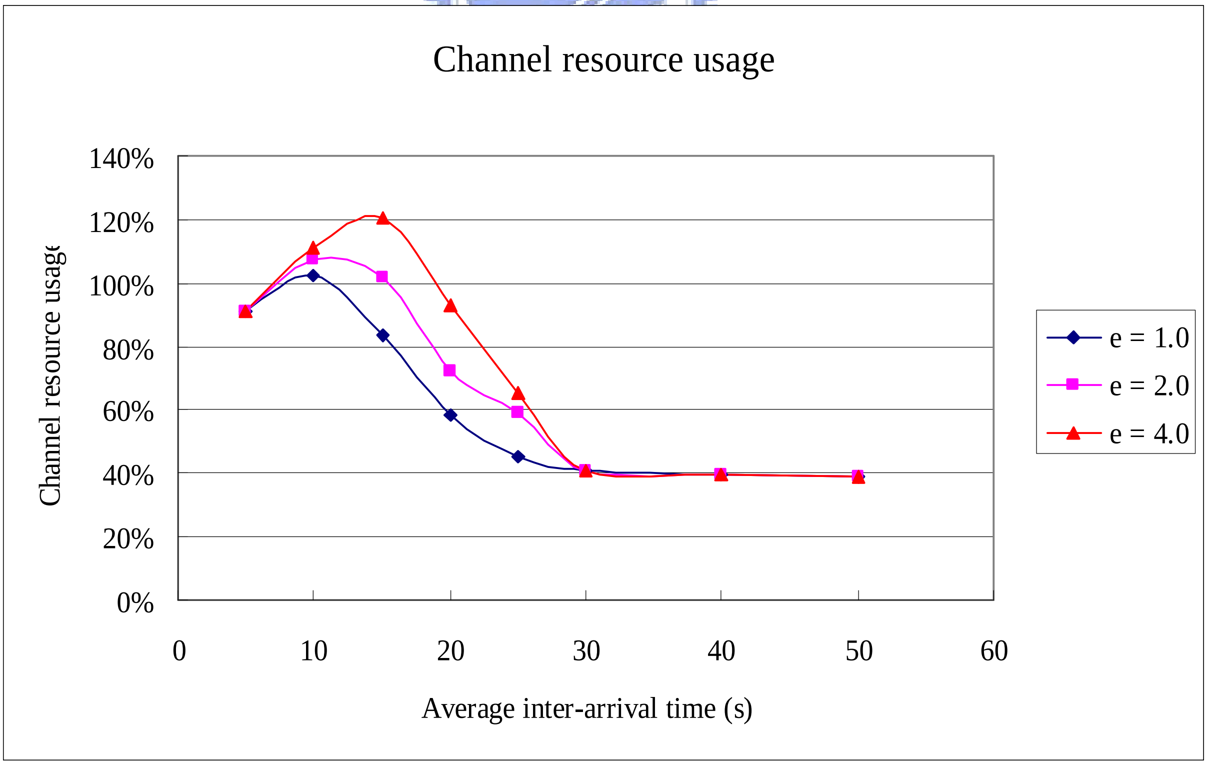

Figure 6-17, Channel resource usage for various parameter *e*

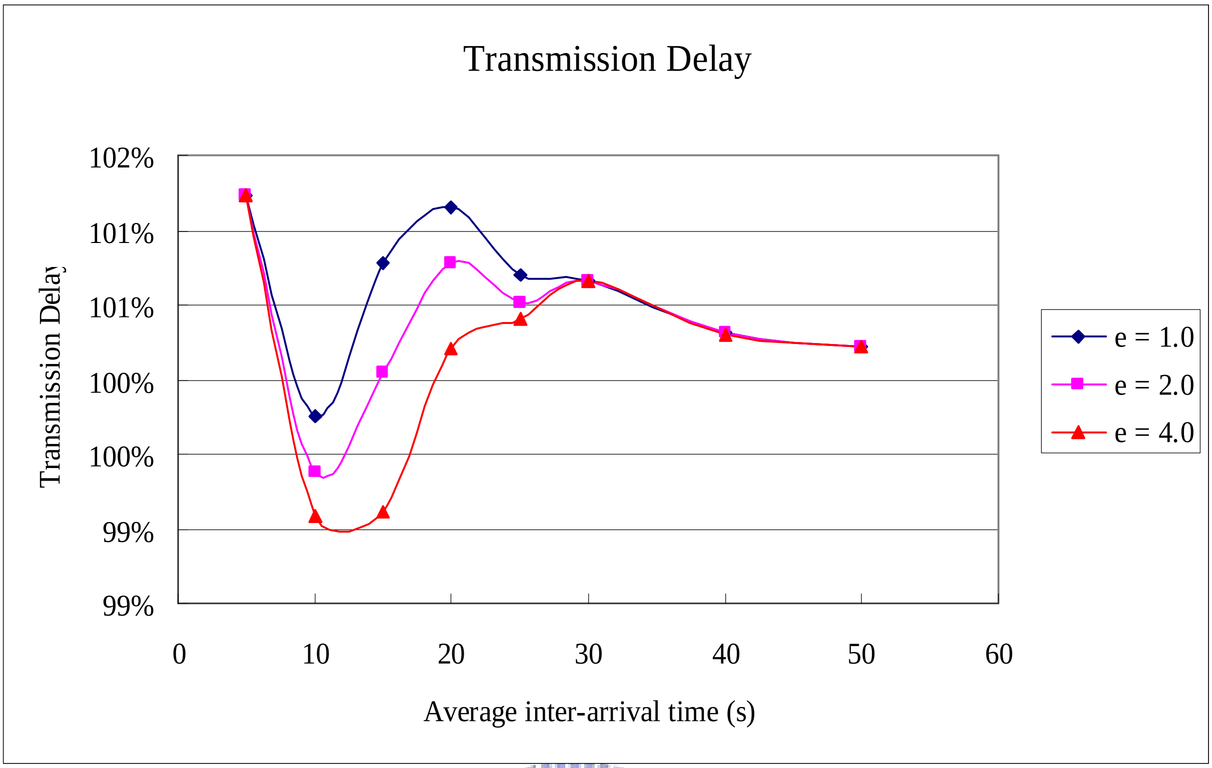

Figure 6-18, Transmission delay for various parameter *e*

Finally, the different minimum dormancy timer is compared. Smaller minimum dormancy timer will gain more benefits because the more quickly release the channel. When the inter-arrival time is very large, the improvement will be as the value of $T_{Min} / (Default\ T_{dormant})$. Form Figure 6-19 and Figure 6-20, the smaller minimum dormancy timer has better performance when inter-arrival time is larger and almost similar performance when inter-arrival time is small. When the inter-arrival time is small, the system holds the resource and the dormancy timer has less relationship with the minimum dormancy timer.

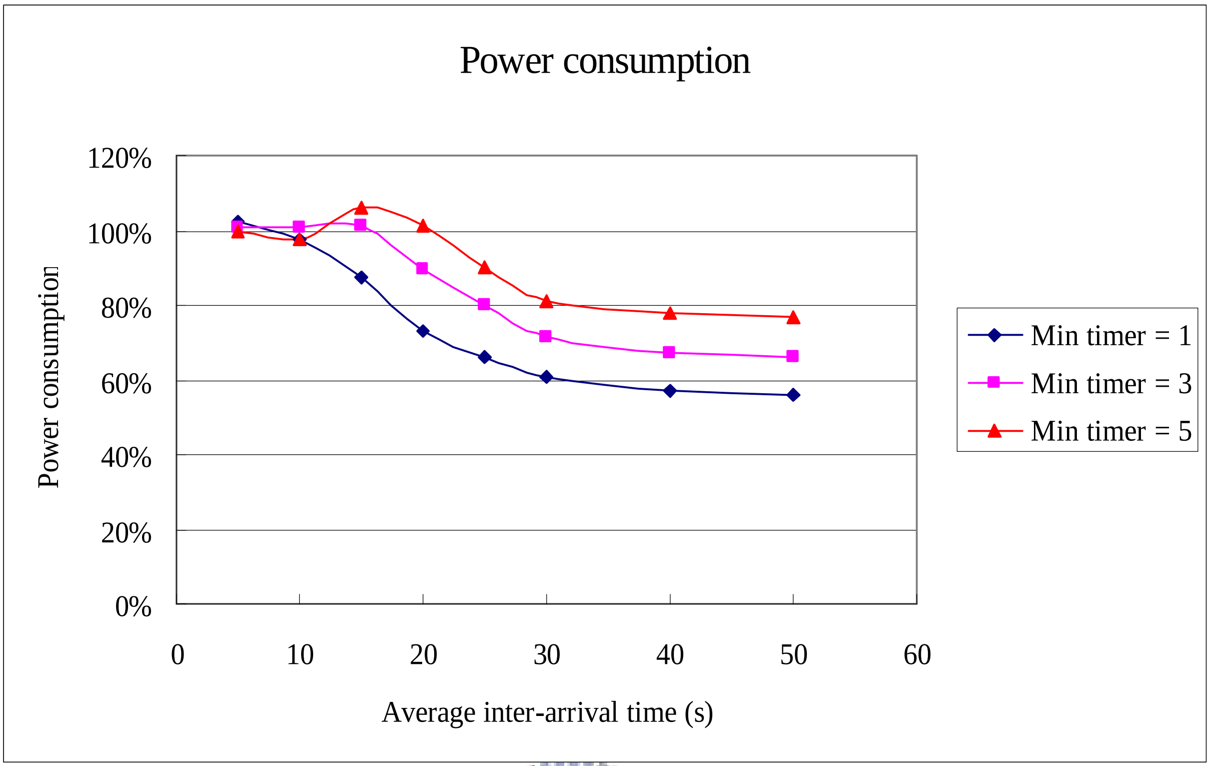

Figure 6-19, Power consumption for various minimum dormancy timers

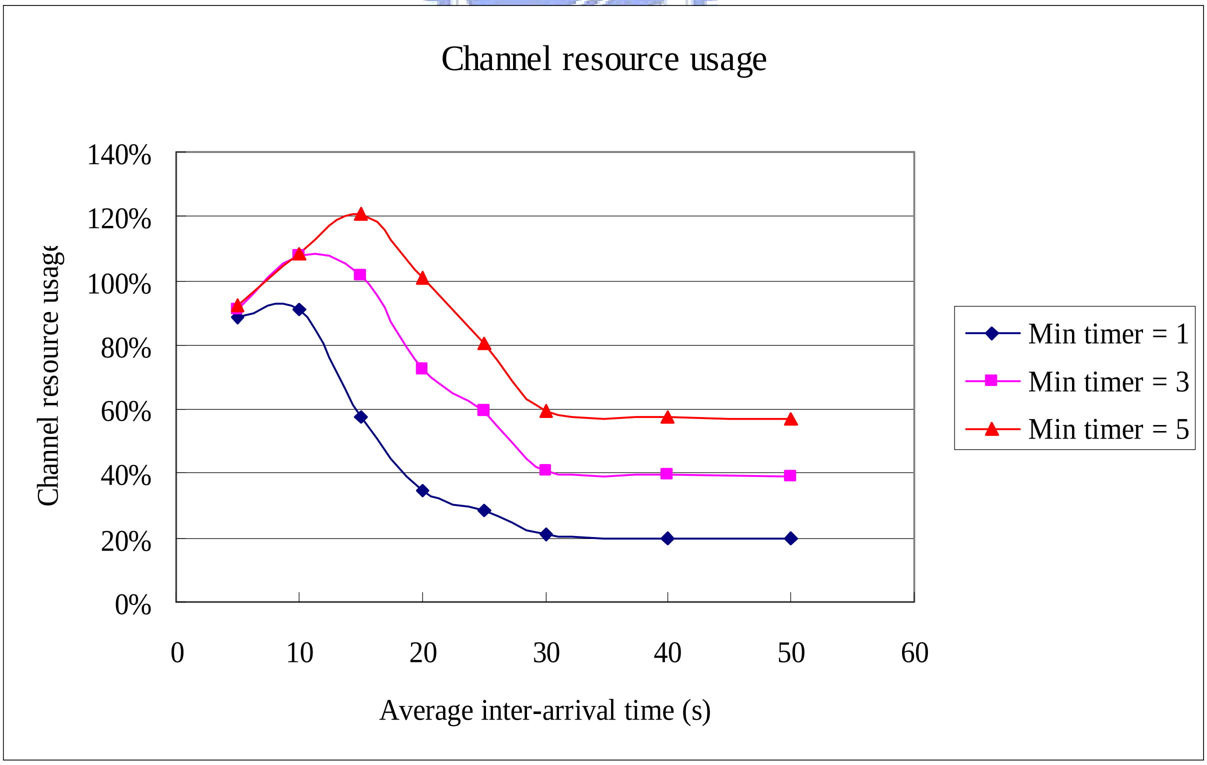

Figure 6-20, Channel resource usage for various minimum dormancy timers

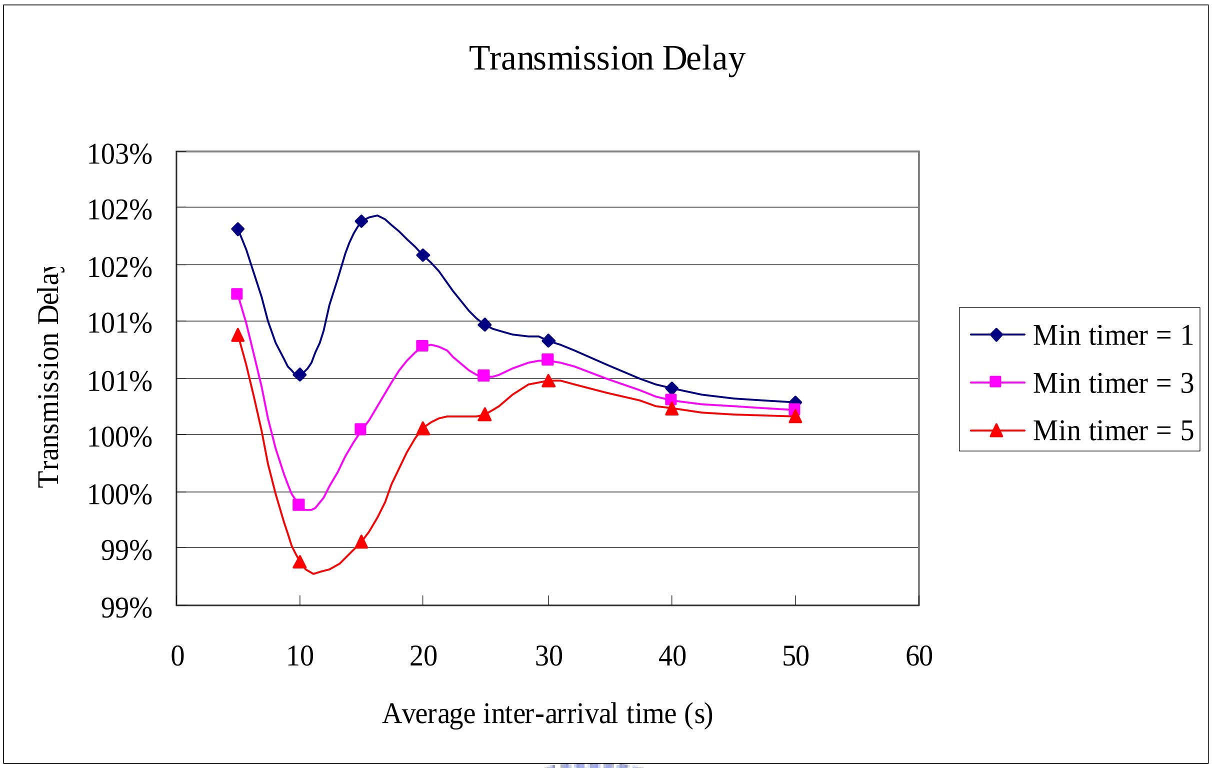

Figure 6-21, Transmission delay for various minimum dormancy timers

Chapter 7 Conclusion and Future Works

7.1 Conclusion

The algorithm has low channel resource usage and power consumption by the accurate perdition of the dormancy timer by either releasing the channel quickly or holding the channel for an accurate duration. It will have a good performance by releasing the resource quickly when the inter-arrival time (delay between packet calls) is long enough. By choosing Smaller T_{Min}, it will save more channel and power resource but increases transmission delay. When the system releases the channel more easily, it will save more channel and power resource but also increases transmission delay. By increasing value of the parameter c and e, the margin between the T_{mean} and the connection releasing threshold will increase. It lets the system release the channel more difficultly. The transmission delay will decrease but the required power and channel resources increase. By increasing value of the parameter d, the dormancy timer will larger. The transmission delay when the system decides to hold the connection will decrease but the power and channel resources increase. By choosing the value of the parameter c, d, and e, the system can trade off between the required power resource, the required channel resource, and the transmission delay.

7.2 Future Works

The math model for the adaptive active-to-dormant control algorithm is proposed with the traffic models. Then the math analysis of the algorithm will be done. Finally, the algorithm will be quantified by the math.

References

[1] Upper Layer (Layer 3) Signaling Standard for cdma2000 Spread Spectrum Systems, Release C, *3GPP2 C.S0005-C Version 1.0, May 28, 2002*

[2] Yang Xiao, C. L. Philip Chen, and Kimberly K. J. Kinateder, “An Optimal Power Saving Scheme for Mobile Handsets”, Computers and Communications, 2001, Sixth IEEE Symposium on, 3-5 July 2001

[3] R. J. Chen, T. Y. Lin, and Y. B. Lin, “Reducing Power Consumption for Mobile Multimedia Handsets”, Tankang Journal of Science and Engineering special issue on Multimedia Computing and Networks, 1999, in press

[4] A. K. Salkintzis and C. Chamzas, “Performance analysis of a downlink MAC protocol with power-saving support”, Vehicular Technology, IEEE Transactions on, Volume: 49 Issue: 3, May 2000

[5] A. K. Salkintzis and C. Chamzas, “An in-band power-saving protocol for mobile data networks”, Communications, IEEE Transactions on, Volume: 46 Issue: 9, Sept. 1998

[6] Changhoi Koo, Hokyu Choi, and Beomsik Bae, “Effective MAC state transition for the mobile station power saving in 3G mobile communication systems”, Wireless Personal Multimedia Communications, 2002. The 5th International Symposium on, Volume: 3, 27-30 Oct. 2002

[7] Jae-Woo So and Dong-Ho Cho, “On effect of timer object for sleep mode operation in cdma2000 system”, Communications, 2000. ICC 2000. 2000 IEEE International Conference on, Volume: 1, 18-22 June 2000

[8] Sunay M.O., Tekinay S., and Ozer S.Z., “Efficient allocation of radio resources for CDMA based wireless packet data systems”, Global Telecommunications Conference, 1999. GLOBECOM '99 , Volume: 1B , 1999

[9] Madhow U., Honig M.L., Steiglitz K., “Optimization of wireless resources for personal communications mobility tracking”, Networking, IEEE/ACM Transactions on , Volume: 3 , Issue: 6 , Dec. 1995

[10] Ho Young Hwang, Sun Jong Kwon, Yun Won Chung, Dan Keun Sung, “A mobility management scheme for reducing power consumption in IP-based wireless networks” Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE , Volume: 3 , 17-21 Nov. 2002

[11] Dongmei Zhao, Xuemin Shen, Mark J.W., “Radio resource management for cellular CDMA systems supporting heterogeneous services”, Mobile Computing, IEEE Transactions on , Volume: 2 , Issue: 2 , April-June 2003

[12] 1xEV-DV Evaluation Methodology – Addendum (V6). 3GPP2, July 25, 2001