creative commons
COMMONS DEED

저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

공학박사 학위논문

Multi-Attribute Operation Policies for Multi-RAT Smartphone

스마트폰에서의 다속성 기반 다중 네트워크 운용 최적화 기법 연구

2015년 2월

서울대학교 대학원

전기·컴퓨터공학부

이 원 보

공학박사 학위논문

Multi-Attribute Operation Policies for Multi-RAT Smartphone

스마트폰에서의 다속성 기반 다중 네트워크 운용 최적화 기법 연구

2015년 2월

서울대학교 대학원

전기·컴퓨터공학부

이 원 보

Multi-Attribute Operation Policies for Multi-RAT Smartphone

지도교수 최 성 현

이 논문을 공학박사 학위논문으로 제출함

2015년 1월

서울대학교 대학원

전기·컴퓨터공학부

이 원 보

이원보의 공학박사 학위 논문을 인준함

2014년 12월

위 원 장: 이 병 기 (인)

부위원장: 최 성 현 (인)

위 원: 박 세 웅 (인)

위 원: 진 성 근 (인)

위 원: 최 영 준 (인)

Abstract

Multi-Attribute Operation Policies for Multi-RAT Smartphone

Today's smartphones integrate multiple radio access technologies (multi-RAT), e.g., 3G, 4G, WiFi, and Bluetooth, etc. Moreover, state-of-the-art smartphones can activate multiple RAT interfaces simultaneously for the parallel transmission. Therefore, it is becoming more important to select the best RAT set among the available RATs, and determine how much data to transfer via each selected RAT network. We propose *Energy, Service charge, and Performance Aware (ESPA)*, an adaptive multi-RAT operation policies for smartphone with supporting system design and multi-attribute cost function for smartphones' Internet services including multimedia file transfer and video streaming services. ESPA's cost function incorporates battery energy, data usage quota, and service specific performance, simultaneously. These attributes are motivated by the growing sensitivity of today's smartphone users to these attributes.

Each time the individual attributes are calculated and updated, ESPA selects the optimal RAT set that minimizes the overall cost. It can activate only the best one RAT interface or exploit multiple RATs simultaneously. The primary benefit of the ESPA is that it enables the smartphone to always operate in the "best" mode without the need for user's manual control; the energy saving mode if the remaining battery energy is becoming nearly depleted; the cost-saving mode if the remaining data quota is almost running out; or, the performance-oriented mode if remaining data quota and battery

energy are both sufficient.

From Chapter 2 to Chapter 4, we cope with file transfer, video streaming, and standby mode for our proposed algorithms. The proposed algorithms are based on the service specific cost or utility models, which also take into account practical issues related to user satisfaction metrics.

First, for file transfer mode, we apply the transfer completion time as the performance metric, and the energy consumption and service charge for downloading a specific size of file are simultaneously considered. Furthermore, we especially take into account a problem that the computational complexity exponentially increases as the number of available RATs increases. We propose a heuristic linear search algorithm to find the optimal RAT set without significant performance degradation. Secondly, for video streaming mode, we consider the HTTP-based video streaming model exploiting multipath with LTE and WiFi networks. Based on analysis of the energy consumption and data usage for the video streaming services, we propose a multi-RAT based video streaming algorithm that balances between the video quality, i.e., the performance metric, and the total playback time with currently given battery energy and data quota. Finally, we cope with the battery energy leakage issue of the smartphone in the standby mode due to intermittent traffic generated by some applications running on background. We analyze the energy-consuming factors in the standby mode and smartphone usage patterns of multiple users, and then, propose a usage pattern-aware deep sleep operation algorithm to save the battery energy in the standby mode.

Simulation results based on real measurement data of the smartphone show that the ESPA algorithms indeed choose the "best" operational mode by maintaining dynamic balance among the performance, energy consumption, and service charge considering the currently provided services and the remaining resources.

Keywords: LTE/WiFi, multi-RAT, smartphone, file transfer, video streaming, energy

saving, data quota saving, balancing multi-attributes, MADM, multi-attribute decision making

Student Number: 2010-30226

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Energy, Service Charge, and Performance aware Multi-RAT Operation Policies for Smartphone

Today's smartphone users are becoming increasingly more sensitive to not only the perceived performance of the received service quality, but also the service charges that they have to pay. In addition, the growing appetite for large-volume multimedia data applications likely leads to faster drainage of their battery life. Consequently, smartphone users are nowadays much more conscious of their usage patterns, and this often forces users to frequently adapt their usage preferences depending on the status of their smartphones and services. For example, when the battery life becomes more important, a smartphone user might prefer communication methods that maximizes the remaining battery life rather than fast download. Otherwise, when a user has consumed most of his/her monthly data quota, the user is likely to limit its use of LTE service and opt for free WiFi hotspots instead.

Yet in another example, a user might want to download a large file as quickly as possible using multiple network interfaces simultaneously if the user is not concerned

with battery life or cost. That is made possible in the real world because state-of-the-art smartphones are capable of the parallel transmission by concurrently activating LTE and WiFi interface to boost up the download speed [1, 2]. Users would be given more options of the RAT selection for a specific service as the number of available radio access technologies (RATs), e.g., 3G, 4G, WiFi, and Bluetooth, in smartphones increases. Therefore, it is important to develop a strategy to choose the best RAT set to maximize the user satisfaction or to minimize the cost for various services.

In this dissertation, we propose the *Energy, Service charge, and Performance Aware (ESPA)* Multi-RAT operation policies to optimally use multiple radio access technologies (multi-RAT) such as LTE and WiFi in smartphones.

We begin by first formulating a multi-attribute cost function considering n RAT-based smartphone characteristics. In particular, we incorporate three attributes, i.e., Energy consumption (E), Service charege (S), and Performance (P) that are considered most important for typical smartphone users, into the cost function. In the algorithms, we especially cope with file transfer and video streaming services for heavy load cases of smartphone usage. Furthermore, we take into account the battery energy leakage due to background traffic in standby mode of the smartphone and propose an energy saving algorithm in the standby mode properly turning off and on the data network interface.

While the file download speed can be boosted up by exploiting multiple RAT paths, the energy consumption and data quota usage might be increased according to the newly activated RAT interfaces. Moreover, it is noted that a change of the network interface selection caused by switching on an interface yields additional delay and energy overheads for the initial connection setup. Such factors should be considered when selecting the best network(s) especially for small size files because the portion of RAT interface's state transition overheads might become more significant to the overall

cost. Therefore, we take these overheads into account as dependent parameters in the energy and time cost attributes, respectively. We propose a best RAT selection and the corresponding segment allocation algorithm with low computational complexity, which, furthermore, dynamically updates the network selection criteria based on the ongoing throughput performance.

Because mobile video streaming services are the biggest part of mobile data traffic sources [3] in these days, we cope with optimizing the operating parameters for the video streaming services, i.e., the video encoding rate, chunk download cycle, multi-RAT (LTE/WiFi) selection. Those factors simultaneously influence the video quality, energy consumption, and data quota usage, and therefore, need to be jointly optimized. Based on numerical analysis of the expected energy consumption and data usage, we formulate an objective function to maximize the sum utility incorporating the video quality and available playback time. We show that the proposed algorithm effectively balances the video quality and available watching time with given resources through comparison with other static schemes.

Even though the file transfer or video streaming services, i.e., heavy traffic services, intensively consume energy and data, the frequency in use of these services is not relatively high. Statistics in [4, 5] represent that smartphones spend over-80% of a day in the standby mode waiting the next usage. Moreover, some applications running in the background generate light load traffic periodically even in the standby mode. Therefore, the energy consumption in the standby mode is considerable for the long-term scale. We study an anatomy of the smartphone power consumption in the standby mode for the both cases connected to either LTE or WiFi considering the background traffic. We propose an energy saving algorithm in the standby mode exploiting user's smartphone usage patterns composed of timestamp logs of the start and end time of each standby mode gathered from multiple smartphone users. Through the real usage

trace-based evaluation, we show that the proposed algorithm adapts each user's usage pattern and effectively save the battery energy.

1.2 Overview of Existing Approaches

1.2.1 Multi-attribute based network selection

Network selection schemes in heterogeneous networks can be classified according to two different points of view: *network-centric* and *user-centric* approaches.

The network-centric approaches consider the overall system performance in a centralized or distributed manner. Several studies [6–8] assume a centralized controller that controls each user's network selection based on an integrated cost function. Cost function-based network selection strategy in [6] proposes a strategy to achieve a balanced point between the call blocking probability and the average received signal strength. In [8], the authors propose a system capacity maximization strategy based on resource allocation considering parallel transmission by activating multiple RATs. Recently in [7], the authors consider more diverse attributes such as spectral efficiency, energy consumption, and fairness. They propose a central global resource controller that manages the resources of several heterogeneous wireless networks to balance these multiple attributes in a system operator's perspective.

The decentralized network-centric approach is usually based on cooperative [9] or non-cooperative game [10]. This approach focuses on the relationship or interaction among *players* that can be users or networks considering *payoffs* that result from their *actions*. The game theoretic algorithms also consider system-wide performance such as call blocking, sum throughput, fairness, and convergence time.

On the other hand, the user-centric approach is also widely studied based on *Always Best Connected* scheme [11]. In this approach, a cost or utility function is also

formulated in terms of various attributes such as user's signal quality, throughput, delay, energy, service charge, and security. While earlier studies mostly focus on improving user's throughput or delay performance by applying appropriate vertical handoff algorithms [12, 13], the energy consumption and service charge are also recently investigated and considered in [14–16].

In order to consider those multiple attributes for network selection, MADM is widely used [17–19]. Several MADM methods such as SAW, Multiplicative Exponential Weighting (MEW), Technique for Order Preference by Similarity to Ideal Solution (TOPSIS), Analytic Hierarchy Process (AHP), and Grey Relational Analysis (GRA) are popularly used for MADM. These methods are basically used to select the best single network by ranking networks based on weighted multiple attributes. However, our approach takes into account the simultaneous usage of multiple networks as well as a single network selection, and hence, adopt SAW method for easy modeling and analysis of the detailed multi-attribute cost function.

1.2.2 Energy and quota-aware video streaming services

video streaming services have been studied in terms of data usage or energy consumption independently [20–22]. In [20], authors propose a dynamic cache management algorithm that considers network speed and power states of 3G and 4G network interfaces, i.e., idle, active, and tail states. According to achievable network throughput and user's viewing time distribution, the algorithm optimize the video cache size to reduce the average power during the playback. The video encoding rate selection algorithm is proposed in [22] maximizes the sum utility of all the viewed videos with given data budget. However, smartphone users are generally concerned about both the remaining data quota and battery energy. Typical users may stop playing video when either available data quota or energy is about to be depleted.

The authors [21] deal with energy consumption issue for the video streaming in wireless environments. They propose to exploit crowd-sourced viewing time statistics for each video, and then schedule video streaming with video download size and request time in order to reduce unnecessary energy consumption. However, this algorithm focuses on only reducing the energy overhead although data quota is another important factor. Insufficient data quota may discourage users to stop watching video streaming even if the battery energy is sufficient. Besides, the proposed algorithm is only useful when there exist crowd-sourced viewing time statistics.

1.2.3 Multi-path based approaches

Transport layer protocols for supporting multi-path based data communications have been studied, where the typical protocols are Stream Control Transmission Protocol (SCTP) [23] and Multipath TCP (MPTCP) [24]. These protocols aim to enhance the connection reliability or to increase network bandwidth. Recently, energy efficient multi-path TCP (MPTCP) schemes by exploiting multiple network paths, e.g., cellular and WiFi, in mobile devices are proposed [25–27]. They improve the energy efficiency (Mb/Joule) of MPTCP rather than the aggregate throughput. In this paper, we focus furthermore on balancing the performance, energy consumption, and data quota usage by finding the optimal RAT set among available n RATs and file segment allocation accordingly.

On the other hand, multi-path data transfer can be also supported by application layer protocols. The *segmented file transfer* is widely used in peer-to-peer file sharing systems [28], and the *HTTP range request* [29] is exploited to download a single file over multiple links [30]. These protocols are based on the client side request, and hence, the file can be transferred through end-to-end multiple links only with the supporting applications in the client and server side. Furthermore, the application layer

protocols are more flexible to allocate file segments to activated multiple RAT paths, and therefore, we adopt the range request protocol type for our system model.

1.3 Main Contributions

1.3.1 File transfer mode

We generalize the multiple network interface activation problem for n RAT-equipped smartphones. We show that the generalized version is the joint combinatorial and piecewise linear minimization problem that can be still properly optimized. We propose a heuristic algorithm to find the optimal RAT set and corresponding segment allocation with low computational complexity. Even though the number of possible RAT combinations exponentially increases as the number of available RATs increases, the proposed linear search algorithm reduces the size of the search space from exponential to linear. We also propose a dynamic update algorithm that adapts the selected RAT set and segment allocation during the transfer according to time-varying network condition. In the performance evaluation, we verify that the proposed algorithm shows no significant performance difference from the full search algorithm. Furthermore, we show that the parallel activation scheme improves the RAT diversity gain compared with the vertical handoff. With time-varying throughput variation of RATs, the performance enhancement through the dynamic update algorithm is also evaluated.

1.3.2 Video streaming mode

We propose an algorithm balancing the video quality and the available playback time with given battery energy and data quota. we formulate a normalized utility function, incorporating the utility of a chosen video encoding rate and the corresponding data usage and energy consumption. For this purpose, we numerically analyze how HTTP-

based video streaming services affect the data usage and energy consumption simultaneously. In the numericaly analysis, we also cope with the expected energy and data waste caused by user's leaving in midstream during the playback. Base on the analysis, we firstly propose a sub-algorithm that searches the optimal chunk download cycle to maximizes the available playback time considering the data usage and energy consumption simultaneously. Then, we propose a complete algorithm that optimizes joint operating parameters, i.e., video encoding rate, multi-RAT (LTE/WiFi) selection with corresponding optimal chunk download cycle. We show that the proposed algorithm selectivley saves more depelting resources, otherwise it improves the video quality as much as possible.

1.3.3 Standby mode

We present an anatomy of smartphone power consumption especially for the standby mode with the connected network between LTE and WiFi. We discuss major considerations in battery energy drain of smartphones according to connected network, usage pattern, and background traffic generation Based on the power consumption estimation in the standby mode, we propose a deep sleep mode algorithm that periodically turns off and on the data network interface to save the energy consumed by the background applications during the standby mode. Through the real usage trace driven simulation, we show the proposed scheme properly adapts a specific usage pattern and effectively improve the energy saving gain.

1.4 Organization of the Dissertation

The rest of the dissertation is organized as follows. In Chapter 2, we describe the multiple $n/$, RAT-based file transfer model in consideration We present problem for-

mulation and model the multi-attribute (time, energy, and quota) cost function. Based on the cost model, we propose a heuristic algorithm to select the best RAT set and the corresponding optimal file segment allocation to the activated network paths. We especically cope with the computational complexity and network throughput dynamicity issues for the proposed algorithm.

In Chapter 3, we propose an energy, quota and performance-aware video streaming algorithm in smartphones. We addopt HTTP-based video streaming model and exploit multi-RAT, i.e., LTE and WiFi considering simultaneous activation of them. The proposed algorithm chooses the best operating parameters, i.e., video encoding rate, chunk download cycle, and network selection. The algorithm considers user's video watching pattern such as the average watching duration, leaving-in-midstream probability. Based on the user-specific profile, it balances between the video quality (encoding rate) and available playback time according to the remaining battery energy and data quota.

In Chapter 4, we present an anatomy of the power consumption at the network interface of smaprthone in the standby mode. We consider the background traffic generation that causes the major battery drainage in the standby mode. Basically, LTE consumes much more energy per packet than WiFi, however, WiFi can occasionally cause additional energy overheads such as beacon reception, scanning, keep-alive messaging in the link layer level. Based on the standby mode power model, we propose a deep sleep mode (DSM) algorithm that periodically turns off and on the data network interface. We exploit multiple users' smartphone usage traces for the evaluation and show that the proposed algorithm effectively saves the energy consumed in the standby mode.

Finally, Chapter 5 concludes the dissertation with the summary of contributions and discussion on the future work.

Chapter 2

File Transfer Mode

2.1 Introduction

As the number of available radio access technologies (RATs), e.g., 3G, 4G, WiFi, and Bluetooth, in today's smartphone increases, users are given more options of the RAT selection for a specific service. Furthermore, state-of-the-art smartphones are capable of the parallel transmission by concurrently activating multiple RATs such as LTE and WiFi to boost up the download speed [1, 2]. Therefore, it is important to develop a strategy to choose the best RAT set to maximize the user satisfaction or to minimize the cost for data service.

These days, smartphone users are becoming more sensitive not only to the perceived performance of the service quality, but also to their service charges that they have to pay. Also, the growing appetite for large-volume multimedia data applications likely leads to faster battery drainage. Therefore, we propose a multi-attribute cost-based multi-RAT interface activation scheme that enables smartphones to automatically choose the optimal RAT set among the available n RATs.

We begin by first formulating a multi-attribute cost function considering n RAT-

based smartphone characteristics. In particular, we incorporate three attributes, namely, file transfer completion time (T), energy consumption (E), and data usage quota (Q), that are considered most important for typical smartphone users, into the cost function. The three cost terms T, E, and Q are merged into the multi-attribute cost function based on the Simple Additive Weighting (SAW) method, which is widely used for Multi-Attribute Decision Making (MADM) algorithm [17, 19].

With this multi-attribute cost function as the primary decision vehicle, the objective of the proposed scheme is then to determine which RAT interface(s) should be activated considering the energy and delay overhead for turning-on/off these interface(s), and then calculate how much data should be transmitted through each selected RAT path to minimize the overall cost. The proposed algorithm enables n RAT-based smartphone to adaptively activate and select the optimal RAT set to minimize the multi-attribute cost according to the file size and estimated throughput, while satisfying the energy and cost requirements.

In our previous work [31], we studied an adaptive network interface activation scheme considering LTE/WiFi-enabled smartphones. The multi-attribute cost function with regard to the transfer completion time, energy consumption, and service charge are modeled for three modes, namely, LTE-only, WiFi-only, and LTE/WiFi-parallel mode, based on the delay and energy measurement of commercial smartphone. However, the previous study is only limited to the two-RAT case, and the issues resulting from extension to the case of more than two RATs are not considered.

For n RAT-based interface activation scheme, a generalized problem formulation with n RATs, where $n \geq 2$, is needed. Furthermore, a low complexity algorithm is needed for dynamic update of the RAT selection during run-time because the search space to find the optimal RAT set is exponentially proportional to the number of available RATs.

In this work, we formulate an objective function composed of two subproblems: (i) piecewise linear optimization problem for the optimal segment allocation to activated RAT paths, and (ii) combinatorial problem to select the optimal RAT set considering parallel transmission over multiple RATs. Our main contributions are as follows:

- We generalize the multiple network interface activation problem for n RAT-equipped smartphones. We show that the generalized version is the joint combinatorial and piecewise linear minimization problem that can be still properly optimized.
- We propose a heuristic algorithm to find the optimal RAT set and corresponding segment allocation with low computational complexity. Even though the number of possible RAT combinations exponentially increases as the number of available RATs increases, the proposed linear search algorithm reduces the size of the search space from exponential to linear.
- We also propose a dynamic update algorithm that adapts the selected RAT set and segment allocation during the transfer according to time-varying network condition.

In the performance evaluation, we verify that the proposed algorithm shows no significant performance difference from the full search algorithm. Furthermore, we show that the parallel activation scheme improves the RAT diversity gain compared with the vertical handoff. With time-varying throughput variation of RATs, the performance enhancement through the dynamic update algorithm is also evaluated.

The rest of this chapter is organized as follows. In Section 2.2, the n RAT-based system model in consideration is described. We present problem formulation and model the multi-attribute (T-E-Q) cost function in Section 2.3. Then, we analyze the characteristics of the problem in Section 2.4, and propose a heuristic algorithm in Section 2.5.

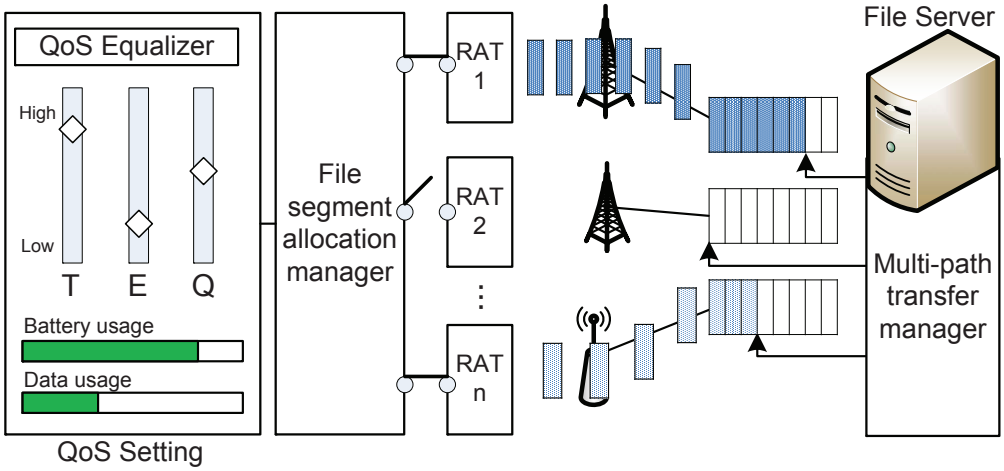


Figure 2.1: Multi-RAT file transfer model.

The performance of the proposed algorithm is evaluated via simulations in Section 2.6. Finally, we summarize and conclude Chapter 2 in Section 2.7.

2.2 System Model

Fig. 2.1 shows the system model that supports parallel or switched data transfer through selected RAT path(s) among n available RATs. In order to fully utilize each RAT bandwidth, a large data file is splitted into a number of small fixed-size segments, e.g., 100 kB, and a set of such segments are allocated to each selected RAT path and transmitted by the server. The receiver combines segments received via multiple RAT paths to reconstruct the original file in the application layer. The process can be supported by the existing multiple flow-based application layer protocol such as the *segmented file transfer* widely used in P2P file sharing systems [28] and the *HTTP range request* to download a single file over multiple links [30]. These protocols are based on the client side request, and hence, the file can be transferred through end-to-end multiple links only with the supporting applications in client and server terminals.

We only focus on the file download case supported by an end-to-end application layer protocol in the system model since the file download is more dominant than the upload counterpart in the real life. To support the upload case, the request-based end-to-end protocol is not enough, and it would need a new protocol, e.g., push-based protocol or a proxy server located between the client and server to aggregate file segments transmitted through multiple RAT paths. Such a protocol issue is out of scope in this chapter.

On the other hand, for the energy, service charge, and performance awareness, the quality of service (QoS) equalizer obtains weighting and normalization factors based on user preference and device status for each QoS term, respectively. The device status includes information on the remaining battery energy and remaining data quota, and the user preference involving weighting factors for the transfer completion time (T), energy consumption (E), and data quota usage (Q) is set through the QoS equalizer as described in Fig. 2.1. Then, the network interface manager calculates the multi-attribute cost function based on the weighting and normalization factors, and determines which RAT set is selected and how many segments are requested for each selected RAT path.

2.3 Problem Formulation

For the optimization problem, we define $\mathcal{N}_A$ as a set of all the available RATs and $\mathcal{S}_A$ as a set of all the possible RAT combinations. For example, a smartphone with $\mathcal{N}_A = \{WiFi, 3G, 4G\}$ will have $\mathcal{S}_A = \{\{WiFi\}, \{3G\}, \{4G\}, \{WiFi, 3G\}, \{WiFi, 4G\}\}$ if 3G and 4G cannot be activated at the same time. Assume that a set including the selected RAT(s) is denoted by set S ($\in \mathcal{S}_A$), and $\mathbf{x}$ is a (nx1) file segment allocation vector for the selected n RATs, where each element x_i denotes the allocated segment

size transferred via RAT i. Then, the multi-attribute cost function $F_S(\mathbf{x})$ is represented by the normalized sum of the transfer completion time, energy consumption, and data usage quota:

$$F_S(\mathbf{x}) = \alpha T_S(\mathbf{x}) + \beta E_S(\mathbf{x}) + \gamma Q_S(\mathbf{x}), \tag{2.1}$$

where $T_S(\mathbf{x})$, $E_S(\mathbf{x})$, and $Q_S(\mathbf{x})$ are transfer completion time, energy consumption, and data usage quota, respectively. α, β, and γ are weighted normalization factors to integrate all these attributes into the cost function, where each one is represented by the multiplication of the user-defined weighting factors $\omega_{\{t,e,q\}}$ and normalization factors as follows: $\alpha = \omega_t/T_{\max}$, $\beta = \omega_e/E_{\max}$, and $\gamma = \omega_q/Q_{\max}$. By these normalization factors, each cost term is represented by the ratio of the cost to the maximum cost, i.e., $T_{\max}$, $E_{\max}$, and $Q_{\max}$. In [31], we apply the maximum tolerable transfer completion time, the remaining battery energy, and the remaining data quota for the maximum cost of each cost term. In addition to the normalization factor, a user can require the limitation of energy and data usage, denoted by E_{req} and Q_{req}, respectively, for the given file transfer service.

Therefore, the optimization problem with the normalized multi-attribute cost function and required energy/quota constraints is formulated as follows:

$$\begin{aligned} &\min_{S \in S_A, \mathbf{x}} \; F_S(\mathbf{x}), \\ &\quad \text{s. t. } \; E_S(\mathbf{x}) \le E_{\text{req}}, \;\; Q_S(\mathbf{x}) \le Q_{\text{req}}, \\ &\qquad \sum_{i \in S} x_i = B_{\text{tr}}\,, \;\; x_i > 0, \; \forall i \in S, \end{aligned} \tag{2.2}$$

where B_{tr} is the size of the file to be transferred. The objectives of the problem are to find out the best RAT set (S^*) with the corresponding segment allocation vector

($\mathbf{x}^*$). In the next section, we briefly explain the T-E-Q cost model for n RAT-based file transfer services and present the proposed optimization problem.

2.3.1 T-E-Q cost modeling

n RAT-based transfer completion time (T) cost

Assuming that a device can access multiple RATs simultaneously, the completion time of the file transfer is determined by the longest time among the selected RAT path(s). Therefore, the file transfer completion time, $T_S(\mathbf{x})$, can be modeled as follows:

$$T_S(\mathbf{x}) = \max_{i \in S} \left(t_i^{\text{tr}} + t_i^{\text{sw}} \right), \qquad \text{where } \; t_i^{\text{tr}} = x_i / r_i, \tag{2.3}$$

where t_i^{tr} is the transfer completion time through RAT i for the segment size x_i, which is a part of the B_{tr}-byte file. r_i denotes the achievable throughput of RAT i, and t_i^{sw} represents the turning-on delay of RAT interface i including the access delay until it can actually transfer data. This is a state-dependent variable, that is equal to zero if the corresponding network is already connected when the file transfer is about to start. From the measurement study [31], the turning-on delay is different for each RAT, and we assume that the value is initially given by manufacturers or obtained by smartphone's self-training.

n RAT-based energy consumption (E) cost

To transmit or receive packets, the battery energy is consumed by CPU and activated RAT interfaces. When generating packets, the energy is consumed by CPU to read/write data buffers and to attach/detach TCP, IP, and MAC headers. Moreover, each RAT interface consumes energy to transmit/receive packets through the wireless channel. Therefore, we take these factors into account for the energy consumption

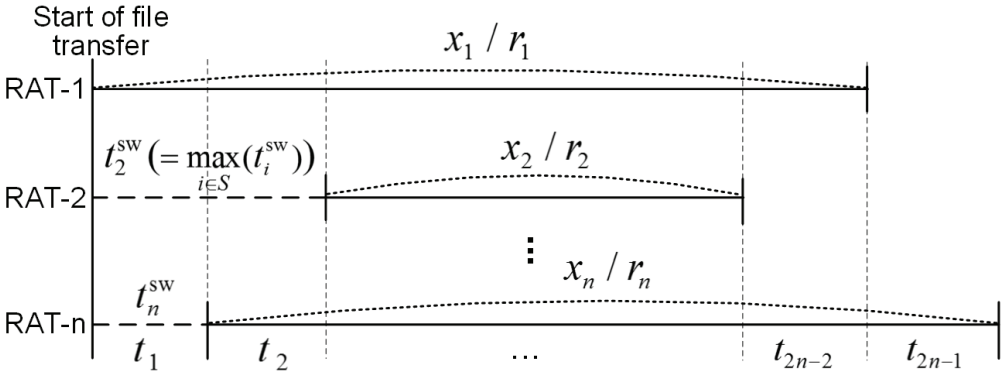


Figure 2.2: Parallel transfer model with n RATs.

model written as $E_S(\mathbf{x}) = E_S^{net}(\mathbf{x}) + E_S^{cpu}(\mathbf{x})$, where $E_S^{net}(\mathbf{x})$ and $E_S^{cpu}(\mathbf{x})$ represent the energy consumption at the activated RAT interfaces and CPU, respectively. The display power is not affected by the change of activated RAT interfaces, and we assume that the display is turned off during the file transfer service. If needed, we can simply add an approximated constant power level according to the current display information such as the brightness or RGB values of pixels.

The energy consumption of each RAT interface is generally composed of the base energy, transmission/reception energy, promotion/tail energy,[1] and turning-on/off energy. Once these components are modeled, we can assume that the parameters can be applied to all the same types of RAT interfaces. The energy consumption of selected RAT interface(s) for data transfer can be expressed as follows:

$$E_S^{net}(\mathbf{x}) = \sum_{i \in S} \left\{ p_i^{net}(r_i) \frac{x_i}{r_i} + e_i^{\mathrm{sw}} + e_i^{\mathrm{oh}} \right\}, \tag{2.4}$$

where $p_i^{net}(r_i)$ is the average power consumption for data transfer through RAT i,

[1] For a power save mode operation, wireless interfaces wake up only when data transfer is requested. After the data transfer is completed, it waits for a certain duration before going back to the sleep mode. The energy overhead for the former and latter cases are referred to as the promotion and tail energy, respectively.

and it is modeled by the function of the achievable throughput r_i. Huang *et al.* show in [32] that the power consumption of 3G, LTE, and WiFi for both of the uplink and downlink transmissions, and their throughput-power curves are approximately fitted to a linear function of the throughput variable. Therefore, we apply the linear fit model for the power consumption according to the achievable throughput for RAT i as follows: $p_i^{net}(r_i) = a_i r_i + b_i$, where a_i and b_i are the linear fit coefficients for RAT i. e_i^{oh} is the aggregate energy overhead including promotion and tail energy to transfer file segments through RAT i. e_i^{sw} is the state-dependent energy overhead due to turning on RAT i's interface, which is set to zero if the RAT interface is already turned on.

On the other hand, it is known that the CPU power is linearly proportional to the CPU usage [33], which is in turn linearly proportional to the packet generation rate. The packet generation rate is proportional to the network throughput, and hence, the CPU power can be also approximated by a linear fit model corresponding to the sum data rate. Fig. 2.2 describes an example of the parallel transmission through n RAT paths, where a smartphone initially connected to RAT-1. As shown in the figure, the horizontal (time) line is divided into $2n-1$ sections, where each section is determined by a different set of activated RAT paths. Accordingly, each section has a different sum rate determined by the activated RAT path set. Assuming that Section k of duration t_k achieves the sum rate of R_k, the CPU energy consumption model is formulated as follows:

$$\begin{aligned} E_S^{cpu}(\mathbf{x}) &= \sum_{k=1}^{2n-1} (cR_k + d)t_k = c\sum_{k=1}^{2n-1} R_k t_k + d\sum_{k=1}^{2n-1} t_k \\ &= cB_{\text{tr}} + dT_S(\mathbf{x}) \end{aligned} \tag{2.5}$$

where c and d are the linear fit coefficients — we apply $c = 3$ mW/Mb/s and $d = 465$ mW referring to [31]. As the equation shows, the CPU energy consumption is

neither related to the number of sections nor the corresponding sum rate, but is only related to the total file size and transfer time.

n RAT-based service quota (Q) cost

Some RATs such as 3G and 4G incur the service charges whereas other RATs such as WiFi and Bluetooth are free of charge. The service charge according to the file segment size for each RAT path is modeled as follows:

$$Q_S(\mathbf{x}) = \sum_{i \in S} q_i x_i, \tag{2.6}$$

where q_i is the service charge rate per byte for RAT i, which is zero for the free network. If the service charge is Q_{p} for B_{p} bytes per month, we model $q_i = \frac{Q_{\text{p}}}{B_{\text{p}}}$.

2.3.2 Optimization problem

Integrating all the T, E, and Q cost terms in (2.3), (2.4), (2.5), and (2.6) into (2.2), the normalized cost function can be formulated as a piecewise linear function of the segment allocation vector $\mathbf{x}$, i.e., $F_S(\mathbf{x}) = \max(\mathbf{u}_i^T \mathbf{x} + v_i)$. Then, the objective function with constraints in (2.2) is represented as follows:

$$\begin{aligned} &\min_{S \in \mathcal{S}_A, \mathbf{x}} \left\{ \max_{i \in S} (\mathbf{u}_i^T \mathbf{x} + v_i) \right\}, \\ \text{s. t. } &\max_{i \in S} (\mathbf{g}_i^T \mathbf{x} + h_i) \leq E_{\text{req}}, \;\; \mathbf{q}^T \mathbf{x} \leq Q_{\text{req}}, \\ &\sum_{i \in S} x_i = B_{\text{tr}}, \;\; x_i > 0, \; \forall i \in S, \end{aligned} \tag{2.7}$$

where

$$
\begin{aligned}
\mathbf{u}_i &= \frac{\alpha+\beta d}{r_i}\mathbf{I}_i + \beta\left(\mathbf{a}+\mathbf{b}\right) + \gamma\mathbf{q}, \\
v_i &= (\alpha+\beta d)t_i^{\text{sw}} + \beta\sum_{j\in S}(e_j^{\text{sw}}+e_j^{\text{oh}}) + \beta c B_{\text{tr}}, \\
\mathbf{g}_i &= \frac{d}{r_i}\mathbf{I}_i + \left(\mathbf{a}+\mathbf{b}\right), \;\; h_i = dt_i^{\text{sw}} + \sum_{j\in S}(e_j^{\text{sw}}+e_j^{\text{oh}}) + cB_{\text{tr}}, \\
\mathbf{a} &= (a_1,\cdots,a_n), \mathbf{b} = (b_1/r_1,\cdots,b_n/r_n), \mathbf{q} = (q_1,\cdots,q_n),
\end{aligned}
$$

where n is the number of elements in set S, and $\mathbf{I}_i$ is the indicator vector that the i-th element is 1 and all the other elements are 0. Then, the objective function forms two subproblems: (i) a piecewise linear minimization to find the optimal segment allocation vector x^* with a given RAT set S, and (ii) a combinatorial problem to find the optimal RAT set S^* among all the candidate RAT sets $\mathcal{S}_A$. The piecewise linear function is known to be convex, and the minimization problem can be solved transforming it to an equivalent linear programming (LP) by forming the epigraph problem [34] with variable t_S as follows:

$$
\begin{aligned}
&\min\; f \\
\text{s. t. }\; &\mathbf{u}_i^T\mathbf{x} + v_i \le f,\;\; \mathbf{g}_i^T\mathbf{x} + h_i \le E_{\text{req}}, \\
&\mathbf{q}^T\mathbf{x} \le Q_{\text{req}},\;\; \sum_{i\in S} x_i = B_{\text{tr}}\,, \quad x_i > 0,\;\; \forall i \in S.
\end{aligned}
\tag{2.8}
$$

The constraint $x_i > 0$ can be relaxed as $x_i \ge 0$ to properly solve the optimization problem. If a solution of Subproblem (i) for RAT set S contains $x_i = 0$, where $i \in S$, this solution can be ignored because the optimal solution of the objective function would be surely in the set $S\backslash i$. The optimal RAT set S^* is obtained by comparing the normalized costs provided by solving Subproblem (i) for all the possible RAT set

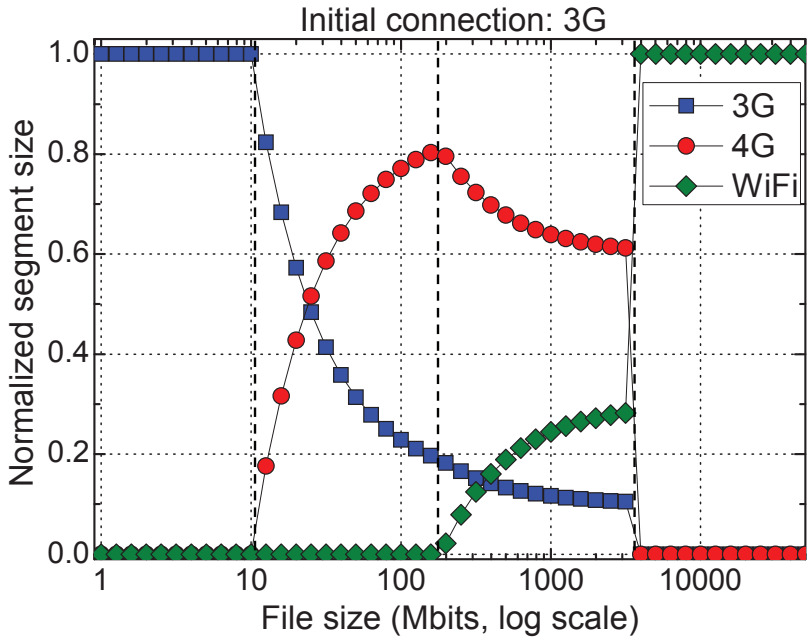


Figure 2.3: Segment allocation with initial connection of 3G.

$S \in \mathcal{S}_A$.

2.4 Numerical Analysis

Taking a look at the objective function in (2.7), let's assume that $\mathbf{u}_i^T\mathbf{x}^* + v_i$ is the maximum for the optimal segment allocation vector $\mathbf{x}^*$, and $\mathbf{u}_j^T\mathbf{x}^* + v_j$ is the second maximum, where $x_i > 0$ and $x_j > 0$. Then, the difference between these two values is represented as: $\Delta f_{ij} = (\alpha + \beta d)\left(\frac{xi}{r_i} - \frac{x_j}{r_j} + t_i^{\text{sw}} - t_j^{\text{sw}}\right)$. If $\Delta f_{ij} > 0$, it can be reduced to zero by decreasing x_i and increasing x_j maintaining $x_i + x_j$ as the same size. Because the cost is linearly proportional to x_i, decreasing x_i means that the optimal cost is also decreased. This is a contradiction with the original value is the optimal. Therefore, we can conclude that $\Delta f_{ij} = 0$, i.e., $\mathbf{u}_i^T\mathbf{x}^* + v_i = \mathbf{u}_j^T\mathbf{x}^* + v_j$, $\forall i, j \in S, i \neq j$ with the optimal segment allocation vector $\mathbf{x}^*$. From this equation and

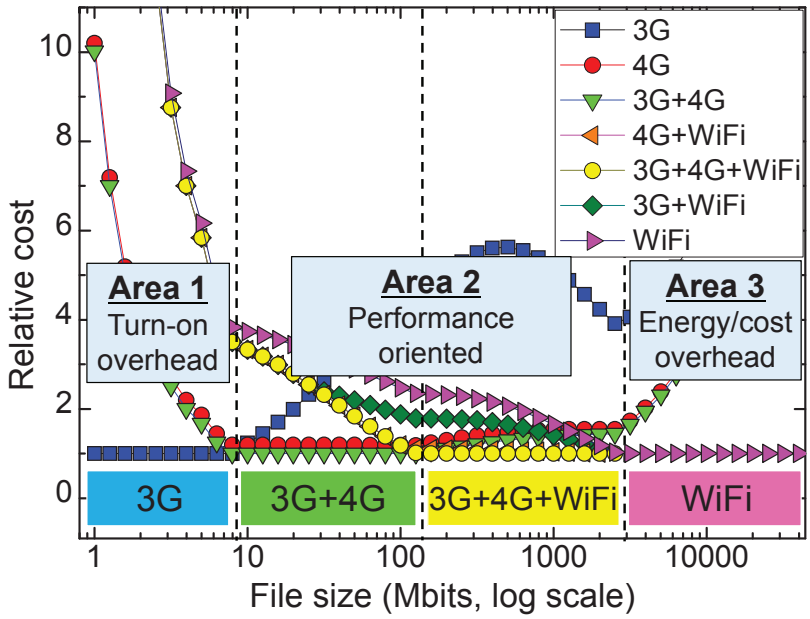


Figure 2.4: File size to relative cost with initial connection of 3G.

$\sum_{i \in S} x_i = B_{\text{tr}}$, the segment size x_i is calculated as follows:

$$x_i = \frac{r_i}{\sum_{k \in S} r_k} \left(B_{\text{tr}} - \sum_{k \in S \setminus i} (t_i^{\text{sw}} - t_k^{\text{sw}}) r_k \right), \forall i \in S. \tag{2.9}$$

This is the optimal segment allocation if it satisfies the other constraints in (2.7) and the cost for RAT set S is the minimum among all the sets in $\mathcal{S}_A$. When (2.9) violates any constraints in (2.7), the algorithm starts to iteratively find the optimal solution of (2.7).

We apply Sedumi [35], a well-known convex optimization tool based on the interior point method, to solve Subproblem (i) for all the possible RAT combinations and find the optimal RAT set that minimizes the cost function. For simulations, we apply coefficients of the throughput-to-RAT interface power curve and throughput-

to-CPU power curve for 4G (LTE) and WiFi, respectively, from the case study presented in [31]. Additionally, the employed coefficients for 3G downlink are $a_{3g} = 35$ mW/Mb/s and $b_{3g} = 810$ mW, which are obtained from Samsung Galaxy S2 HD LTE, the model used for the measurement reported in [31]. We set the average throughput for 3G, 4G, and WiFi networks as 5, 25, and 15 Mb/s, respectively, for the baseline simulation. The battery energy for $E_{\max}$ is set to 2000 mAh, i.e., 7400 mWh at 3.7 V, and the data quota for $Q_{\max}$ is assumed to be 10 GB. In addition, the turning-on delay of each network interface is assumed to be 2 s, 2 s, and 7 s for 3G, 4G, and WiFi, respectively.

First, we consider the case that the smartphone is initially connected to 3G. Fig. 2.3 illustrates a simulation result of the optimal segment allocation of each RAT path for varying file sizes. Fig. 2.4 shows the relative cost of each RAT selection with the corresponding optimal segment allocation while normalizing the minimum cost for each file size to one. Each color box in the bottom of the figure represents the optimal RAT set for the corresponding file size range.

In the figure, the relative cost of the currently connected network, i.e., 3G, for small size files is the least among the relative costs of all the possible RAT sets. That is because the delay and energy overhead to activate the other RAT interface(s) is relatively too high compared with the actual file transferring cost (Area 1). On the other hand, as the file size increases, it benefits from additionally activating the other RAT interface(s) and boosting up the download speed, meaning to decrease T cost (Area 2). However, when either the file size is very large or the transfer energy and/or data quota meets the required limitation, the optimal RAT set converges to RAT(s) that are free of charge and consume lower energy (Area 3).

In this scenario, we assume that 3G and 4G are compatible, i.e., able to be activated simultaneously. However, if 3G and 4G are incompatible, i.e., unable to be simultane-

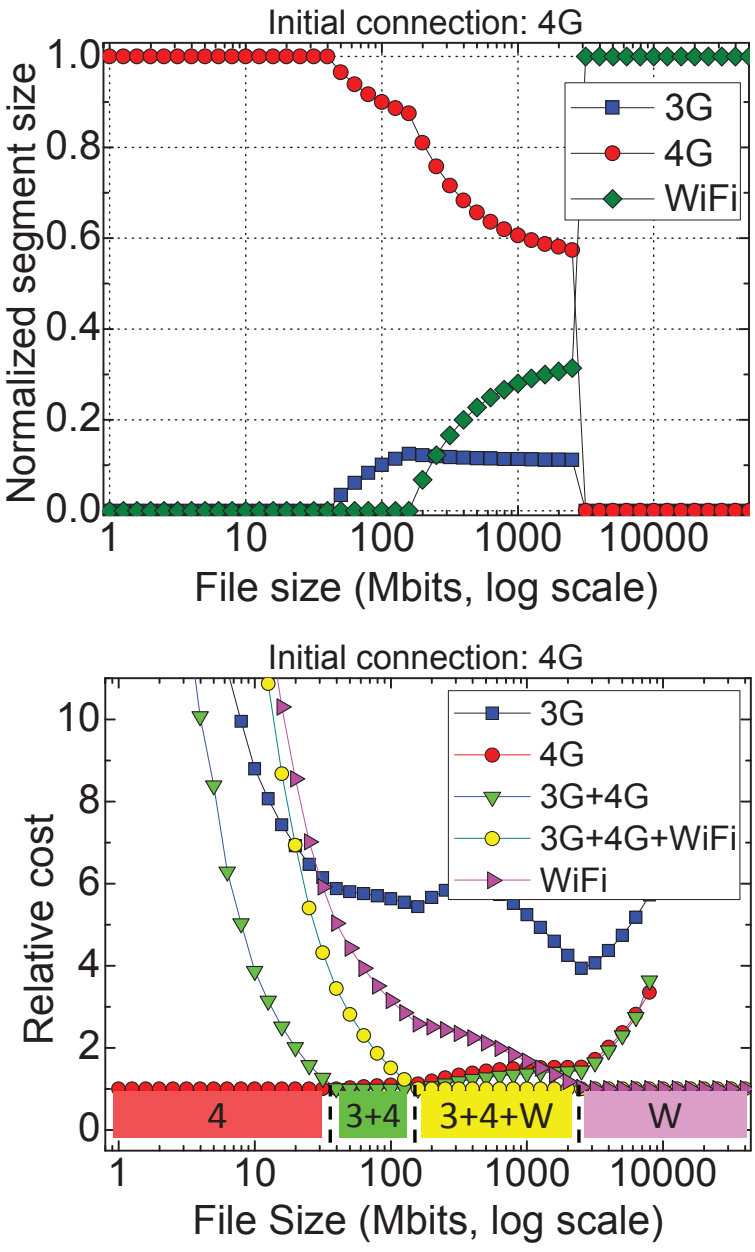


Figure 2.5: Network selection and segment allocation initial 4G connection case.

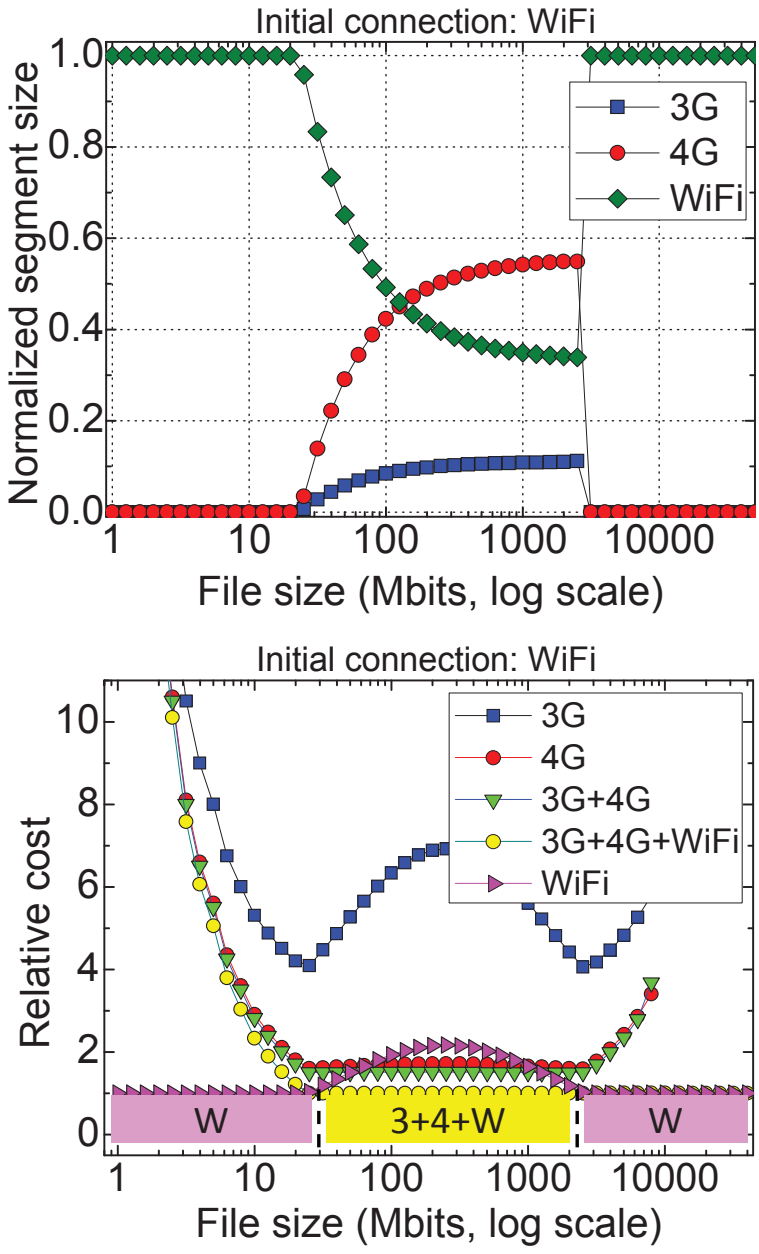


Figure 2.6: Network selection and segment allocation for initial WiFi connection case.

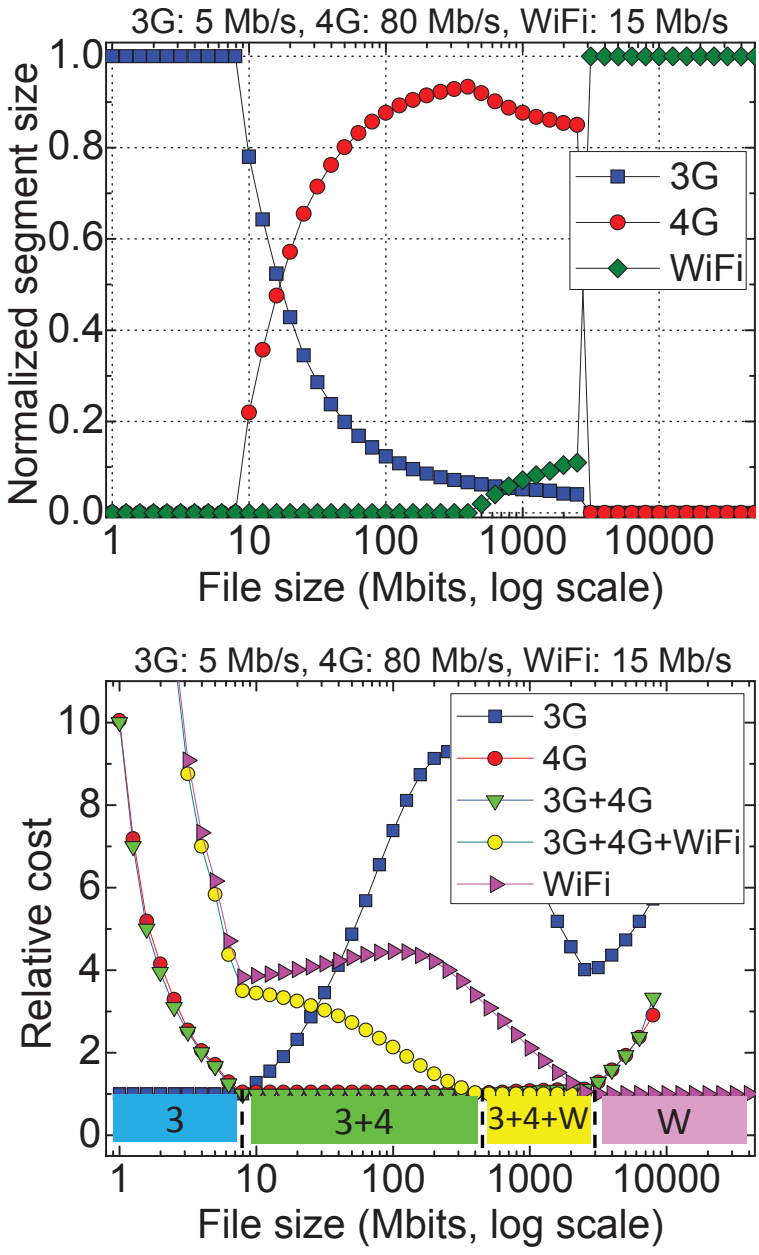


Figure 2.7: Network selection and segment allocation for high 4G throughput case.

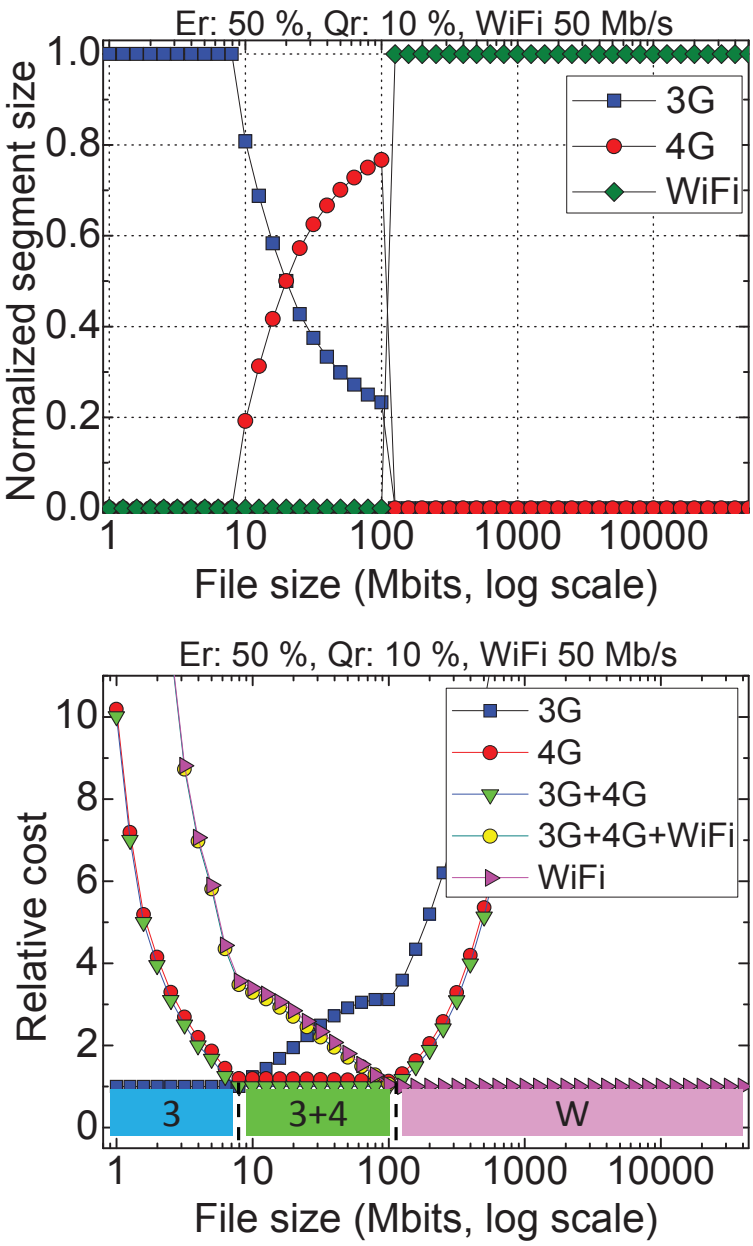


Figure 2.8: Network selection and segment allocation for limited energy/data case.

ously activated, the next best RAT set, i.e., {3G, WiFi} or {4G, WiFi}, will be selected instead of {3G, 4G} or {3G, 4G, WiFi} for that range. Our proposed algorithm, explained in the following section, takes into account the compatibility test, and hence, this issue can be resolved.

Fig. 2.5–2.8 describe the other cases of the RAT selection and segment allocation according to the various device conditions. The optimal RAT set for each file size is represented in the color box, where 3G, 4G, and WiFi are abbreviated to 3, 4, and W, respectively. Fig. 2.5 presents a case that the initial connection is 4G instead of 3G and the other condition is the same as the baseline simulation. In this case, the portion of the {4G} range increases because of its higher throughput (25 Mb/s) than that of 3G (5 Mb/s). A case that the initial connection is WiFi is also presented in Fig. 2.6. In this case, only two kinds of RAT set, i.e., {WiFi} and {3G, 4G, WiFi}, are chosen.

Fig. 2.7 presents the case that the 4G throughput is very high, i.e., 80 Mb/s, which is 16 times higher than that of 3G when the initial connection is 3G. In this case, the {3G, 4G} range increases and the segment size allocated to 4G is much more increased to make full use of its high throughput performance. However, the {3G} range remains because it still has an advantage for the small size file due to the turning-on delay and energy overhead of the other interfaces.

On the other hand, if the energy and data budgets are exhausted or the WiFi throughput is very high, the optimal RAT set changes as shown in Fig. 2.8. In this case, the remaining energy is 50%, the remaining data quota is 10%, and WiFi throughput is 50 Mb/s. Then, WiFi that is free of charge and consumes less power while performing even better, and hence the {WiFi} range is much more extended. The only shortcoming of WiFi is the longer turning-on delay, i.e. 7 s, and therefore, the {3G} and {3G, 4G} ranges still exist when the file size is relatively small.

2.5 Proposed Algorithm

First, we propose a heuristic algorithm, called Bi-directional Linear Search (BLS), which finds the optimal RAT set with optimal segment allocation vector by solving (2.7) with low computational complexity. It considers the characteristics discussed in the previous section to reduce the search space of the available RAT set. Second, we propose a dynamic update algorithm that adapts the selected RAT set and segment allocation vector over time according to the time-varying network condition.

2.5.1 Bi-directional linear search algorithm

The BLS algorithm basically reduces the search space of candidate RAT sets by selectively visiting possible RAT combinations. It searches candidate RAT sets from the single-RAT set to the set including all the available RATs by adding or subtracting a RAT sequentially. At first, two types of RATs are visited, i.e., the currently connected RAT and another RAT that is free of charge or consumes the lowest power. If the former one is different from the latter one, the search direction is splitted into two directions, which is the reason why the proposed algorithm is called BLS. If the both RATs are the same, the algorithm searches just in one direction. The detailed procedure is explained below.

At first, n available RATs are sorted in two ways — in ascending orders of *the turning-on delay* and *the E-Q cost* , i.e, the normalized sum of the energy and quota cost, respectively, as shown below:

$$\begin{aligned} D_{\pi_1} < D_{\pi_2} < \ldots < D_{\pi_n}, \\ C_{\psi_1} < C_{\psi_2} < \ldots < C_{\psi_n}, \end{aligned} \tag{2.10}$$

where D_{π_i} and C_{ψ_i} are the turning-on delay and E-Q cost for the available RATs

$\pi_i, \psi_i \in \mathcal{N}_A$, $\forall i \in \{1, 2, \ldots, n\}$, respectively. Then, RAT π_1 has the minimum turning-on delay, which is likely to be the currently connected RAT, and RAT ψ_1 has the minimum E-Q cost at the current smartphone state. It starts then to find the minimum cost function with candidate RAT sets that are chosen in the order from $\{\pi_1\}$ to $\{\pi_1, \ldots, \pi_n\}$ by adding a RAT to the previous RAT set like {Area 1 $\rightarrow$ Area 2} in Fig. 2.4, i.e., the search direction is from the short turning-on delay to the performance oriented way. A similar process is executed with candidate RAT sets chosen from $\{\psi_1\}$ to $\{\psi_1, \ldots, \psi_n\}$ like (Area 3 $\rightarrow$ Area 2) in the figure, i.e., the search direction is from the low energy/quota overhead to the performance oriented way. If a newly added RAT is not able to be concurrently activated with RATs in the previous set and the turnning-on delay or E-Q cost of the new one is smaller than that of the previous one, the newly added one replaces the previous one. The details of the proposed algorithm, namely Bi-directional Linear Search (BLS) algorithm, are presented in Algorithm 1.

When a file transfer service is requested, the transferred file size is stored to B_{tr}. S_π and S_ψ represent currently tested sets in each direction, which are initially set to RAT π_1 and ψ_1 (Line 2). The normalized cost of a single-RAT connection is directly calculated by allocating the whole file size to that RAT connection, which is denoted by $F_{\pi_i}(B_{tr})$. Then, the optimal (minimum) normalized cost F^* and the optimal RAT set S^* are initialized by the minimum value among the normalized costs with single-RAT connections and the corresponding RAT (Line 3). $\mathbb{S}_c$ represents a set of already tested RAT sets and is initialized by $\{\{\pi_1\}, \{\pi_2\}, \ldots, \{\pi_n\}\}$ (Line 4) because all the single-RATs have been tested in Line 3. This set is used to check out previously tested RAT sets and to prevent redundantly testing the same RAT set.

Each loop in the algorithm searches the optimal RAT set with the optimal segment allocation in the different direction as ordered in (2.10). In *Search direction I*, the available RAT sets are tested from $\{\pi_1\}$ to $\{\pi_1, \ldots \pi_n\}$, and the available RAT sets

Algorithm 1 Bi-directional Linear Search (BLS) Algorithm

Initialize:

$B_{\text{tr}} \leftarrow \text{getFileSize}()$

$S_\pi \leftarrow \pi_1,\ S_\psi \leftarrow \psi_1,\ Othervariables \leftarrow \emptyset$

$(F^*, S^*) \leftarrow \Big(\min_{\pi_i \in \mathcal{N}_\mathcal{A}} F_{\pi_i}(B_{\text{tr}}),\ \arg\min_{\pi_i \in \mathcal{N}_\mathcal{A}} F_{\pi_i}(B_{\text{tr}}) \Big)$

$\mathbb{S}_c \leftarrow \{\{\pi_1\}, \{\pi_2\}, \ldots, \{\pi_n\}\}$

Search direction I: ▷ Area 1 → Area 2 (in Fig. 2.4)

for $i = 2$ to n **do**
  $S'_\pi \leftarrow S_\pi \cup \pi_i$
  if π_i is incompatible with $\exists \pi_k \in S_\pi$ **then**
    $S'_\pi \leftarrow S'_\pi \setminus \pi_k$
  end if
  if $S'_\pi \notin \mathbb{S}_c$ **then**
    $S_\pi \leftarrow S'_\pi,\ \mathbb{S}_c \leftarrow \mathbb{S}_c \cup \{S_\pi\}$
    $(\mathbf{x}_\pi, F_\pi) \leftarrow OptimalAlloc(B_{\text{tr}},\ S_{\text{cur}},\ S_\pi,\ \hat{\mathbf{r}}_{S_\pi})$
    if $F_\pi < F^*$ **then**
      $(\mathbf{x}^*, F^*, S^*) \leftarrow (\mathbf{x}_\pi, F_\pi, S_\pi)$
    end if
  end if
end for

Search direction II: ▷ Area 3 → Area 2 (in Fig. 2.4)

for $j = 2$ to n **do**
  $S'_\psi \leftarrow S_\psi \cup \psi_j$
  if ψ_j is incompatible with $\exists \psi_k \in S_\psi$ **then**
    $S'_\psi \leftarrow S'_\psi \setminus \psi_k$
  end if
  if $S'_\psi \notin \mathbb{S}_c$ **then**
    $S_\psi \leftarrow S'_\psi,\ \mathbb{S}_c \leftarrow \mathbb{S}_c \cup \{S_\psi\}$
    $(\mathbf{x}_\psi, F_\psi) \leftarrow OptimalAlloc(B_{\text{tr}},\ S_{\text{cur}},\ S_\psi,\ \hat{\mathbf{r}}_{S_\psi})$
    if $F_\psi < F^*$ **then**
      $(\mathbf{x}^*, F^*, S^*) \leftarrow (\mathbf{x}_\psi, F_\psi, S_\psi)$
    end if
  end if
end for

Finalize:

$(\mathbf{x}_{\text{cur}},\ F_{\text{cur}},\ S_{\text{cur}}) \leftarrow (\mathbf{x}^*, F^*, S^*)$

Build the alternative RAT set $\mathbb{S}_{\text{alt}}$ for S_{cur}

are tested from $\{\psi_1\}$ to $\{\psi_1, \ldots \psi_n\}$ in *Search direction II*. For each search direction, the union of the previous RAT set S_π/S_ψ and RAT π_i/ψ_j is saved to S'_π/S'_ψ (Lines 6 and 19). If the newly added RAT cannot be concurrently activated with any RAT in the previously tested set, the incompatible RAT in the previously tested set is removed from the newly tested RAT set (Lines 7–8 and 20–21). If the RAT set S'_π/S'_ψ is not in the set of the already tested RAT sets $\mathbb{S}_c$, S_π/S_ψ and $\mathbb{S}_c$ are updated by the new RAT set (Lines 10–11 and 23–24). With the updated RAT set S_π/S_ψ, the optimal segment allocation for each selected RAT is obtained by solving Subproblem (i) in the function *OptimalAlloc()* (Lines 12 and 25). S_{cur} is the activated RAT set when the algorithm starts, and the RAT activation overhead would be added when $S_{\text{cur}} \neq S_\pi/S_\psi$. $\hat{\mathbf{r}}_{[S_\pi/S_\psi]}$ is the estimated throughput vector of the RAT set S_π/S_ψ.

The function *OptimalAlloc()* with these arguments solves (2.7) and returns the optimal segment allocation vector and its cost ($\mathbf{x}_{[\pi/\psi]}$, $F_{[\pi/\psi]}$). If the returned cost is less than the minimum cost F^*, the optimal segment allocation vector, the optimal cost, and the optimal RAT set are updated by the returned values and the new RAT set (Lines 14 and 27). After both the loops finish, the RAT set with the corresponding segment allocation vector is finally updated with the optimal results. (Line 31). In addition, the set of the alternative RAT sets $\mathbb{S}_{\text{alt}}$ is built, where the alternative set replaces any RAT in S_{cur} with the incompatible RAT(s) (Line 32). This set is used for the proposed dynamic update algorithm, and the details are discussed in the next section.

By the BLS algorithm, the number of visited RAT combinations at most is reduced from 2^n to $2n - 1$, where n is the number of available RATs. For performance evaluation, *OptimalAlloc()* has two versions: *1)* to solve Subproblem (i) by the existing convex tool, and *2)* to solve it by examining the segment allocation vector in (2.9) first, and then using the convex tool only when (2.9) is infeasible. We denote the first method by "BLS w/ cvx_tool" and the second method by "fBLS w/ cvx_tool", where

Algorithm 2 Dynamic Update Algorithm

while file transfer **do**
  if update timer expires **then**
    $B_r \leftarrow B_{\text{tr}} - B_{\text{done}}$
    $(\mathbf{x}_{\text{done}}, F_{\text{done}}) \leftarrow CalCost(B_{\text{done}}, S_{\text{old}}, S_{\text{cur}}, \bar{\mathbf{r}}_{S_{\text{cur}}})$
    $(\mathbf{x}_{\text{todo}}, F_{\text{todo}}) \leftarrow OptimalAlloc(B_r, S_{\text{cur}}, S_{\text{cur}}, \hat{\mathbf{r}}_{S_{\text{cur}}})$
    $(\mathbf{x}_{\text{upd}},\ F_{\text{upd}}) \leftarrow (F_{\text{done}} + F_{\text{todo}},\ \mathbf{x}_{\text{done}} + \mathbf{x}_{\text{todo}})$
    if $F_{\text{upd}} > F_{\text{cur}} + m_{\text{cur}}$ **then**
      $B_{\text{tr}} \leftarrow B_r$
      go to BLS algorithm
    else
      while $\forall i,\ S_i^{\text{alt}} \in \mathbb{S}_{\text{alt}}$ **do**
        $(\mathbf{x}_i^{\text{alt}}, F_i^{\text{alt}}) \leftarrow OptimalAlloc(B_r, S_{\text{cur}}, S_i^{\text{alt}}, \hat{\mathbf{r}}_{S_i^{\text{alt}}})$
      end while
      $o \leftarrow \arg\min_i F_i^{\text{alt}}$
      if $F_o^{\text{alt}} < F_{\text{todo}} - m_{\text{alt}}$ **then**
        $S_{\text{old}} \leftarrow S_{\text{cur}}, B_{\text{tr}} \leftarrow B_r, B_{\text{done}} \leftarrow 0$
        $(\mathbf{x}_{\text{cur}},\ F_{\text{cur}},\ S_{\text{cur}}) \leftarrow (\mathbf{x}_o^{\text{alt}}, F_o^{\text{alt}}, S_o^{\text{alt}})$
      else
        $(\mathbf{x}_{\text{cur}},\ F_{\text{cur}},\ S_{\text{cur}}) \leftarrow (\mathbf{x}_{\text{upd}}, F_{\text{upd}}, S_{\text{cur}})$
      end if
    end if
  end if
end while

"f" means "fast."

2.5.2 Dynamic update algorithm

During the file transfer, the network conditions of the activated RATs, e.g., channel and load, may change dynamically. Accordingly, the throughput of each RAT fluctuates, and the initially estimated throughput is likely to be different from the actual average throughput. The increased throughput is out of our consideration because it reduces the original cost. However, throughput degradation of the selected RAT causes the

increase of the original cost, and the optimal solution might be changed. Therefore, we propose a dynamic update algorithm in Algorithm 2 to adapt the selected RAT set to the dynamic network condition.

For the dynamic update algorithm, the optimality of the network selection and segment allocation is periodically checked by monitoring the achieved throughput during the transfer [31]. B_r is the remaining file segment size, i.e., B_{tr} subtracted by the already transferred file segment size B_{done} . The cost F_{done} for the segment allocation vector of the already transferred data $\mathbf{x}_{\text{done}}$ is calculated with the cumulative average throughput vector $\bar{\mathbf{r}}_{S_{\text{cur}}}$ of the activated RATs (Line 4). The optimal segment allocation vector $\mathbf{x}_{\text{todo}}$ and the cost F_{todo} for the remaining file segment size are re-calculated by the *OptimalAlloc()* with the current RAT set and its estimated throughput vector $\hat{\mathbf{r}}$ (Line 5). We adopt the autoregressive and moving average (ARMA) estimator [36] for throughput estimation as follows:

$$\hat{r}_i = \sigma \hat{r}_{i-1} + \frac{1-\sigma}{K} \sum_{j=0}^{K-1} r_{i-j}, \tag{2.11}$$

where we set $K = 10$ and $\sigma = 0.95$ in the simulation.

If the updated cost F_{upd} with the segment allocation vector $\mathbf{x}_{\text{upd}}$, which is the sum of the results in Lines 4 and 5, exceeds the sum of the current cost and specific margin m_{cur}, "BLS algorithm" in Algorithm 1 can be re-visited to search the better RAT set (Lines 7–9).

On the other hand, if some RATs cannot be activated simultaneously with each other, the optimal solution might be the case that these RATs are sequentially used. For example, assume that S_A = {{3G}, {4G}, {WiFi}, {3G, WiFi,}, {4G, WiFi}}, but {3G, 4G} and {3G, 4G, WiFi} are not in S_A. The optimal activation strategy could be activating 4G and WiFi for some time, and then turn off 4G and activating 3G

with WiFi. Even though, this case is not directly included in our problem formulation, the proposed dynamic update algorithm in Algorithm 2 deals with this issue. At first, RATs that cannot be activated simultaneously are defined as the alternative set and stored in a database. When the best RAT set is obtained, the alternative RAT set $\mathbb{S}_{\text{alt}}$ is built accordingly (Line 32 in Algorithm 1). For example, if S_A is given as mentioned above and the optimal RAT set is $\{$3G, WiFi$\}$, the alternative RAT set is $\{$4G, WiFi$\}$ when 3G and 4G are incompatible. Then, the optimal segment allocation vector and the corresponding cost for each alternative RAT set is calculated for the remaining file size (Line 12), and the RAT set of the minimum cost among the alternative RAT sets is found out (Line 14). If the minimum alternative cost is m_{alt} less than the cost F_{todo}, where m_{alt} is a predefined margin, the current RAT set is updated to the alternative RAT set and corresponding segment allocation (Lines 15–17). Otherwise, the current cost F_{cur} and segment allocation vector $\mathbf{x}_{\text{cur}}$ are substituted by F_{upd} and $\mathbf{x}_{\text{upd}}$, respectively (Line 19).

2.6 Performance Evaluation

In the future smartphones, much more diversified types of RATs and channels will be available, and hence, it will achieve larger RAT diversity gain as the number of RATs increases at the cost of complexity increase to find the best RAT set. In order to see the performance of the proposed algorithm, we assume that a smartphone is equipped with various sets of RATs — from 3 to 6 RATs, where 3G, 4G, and WiFi interfaces are common and the other interfaces have arbitrary parameters for throughput, energy consumption, and data usage quota. We also assume that any combination among the available RATs can be activated concurrently. However, for the case of 3 RATs with 3G, 4G, and WiFi, we especially deal with one more case where it is unable to si-

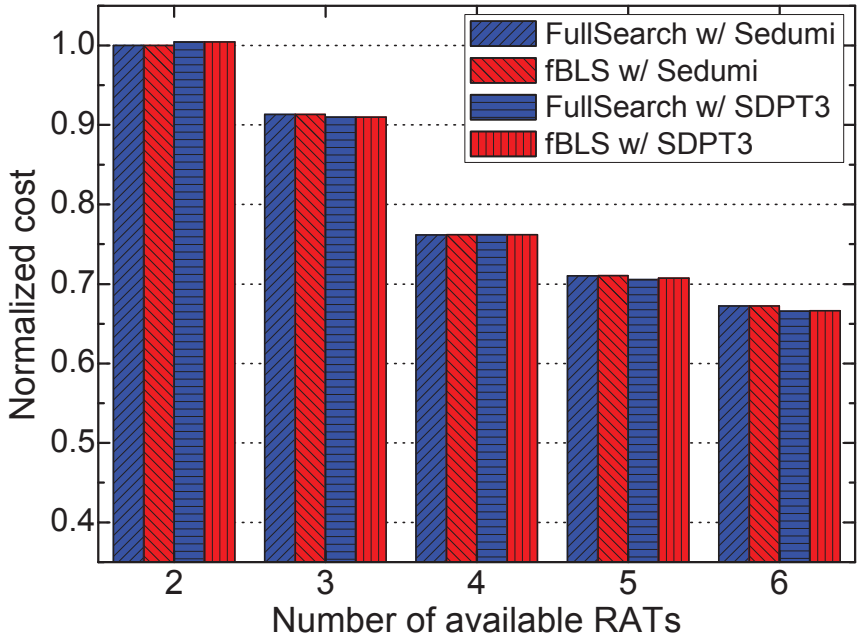


Figure 2.9: Comparison of RAT diversity gain.

multaneously activate {3G, 4G} and {3G, 4G, WiFi}, i.e., 3G and 4G can be only exclusively used. This case is denoted by "2 RATs" instead of "3 RATs" in the figure of simulation result. For simulation environments, we consider four different conditions, where the average achievable throughput of 4G and WiFi respectively varies as the following: {25, 15}, {5, 15}, {25, 5}, {5, 5} Mb/s. We assume that the probability density function of the file size follows a lognormal distribution where the mean and standard deviation are 7.17 and 2.41, respectively, referring to [37]. The expected optimal costs for the four different conditions are averaged. With a given RAT set, we use the convex optimization tools, Sedumi [35] and SDPT3 [38], to solve Subproblem (i) in the objective function.[2] We consider the full-search method, which searches all the

[2]LP is typically known to have polynomial time complexity in the worst case. However, Subproblem (i) is always solved in the constant iteration level by these tools for the scope in consideration (2–6 RATs). Therefore, we assume that the computational complexity is dominated by Subproblem (ii).

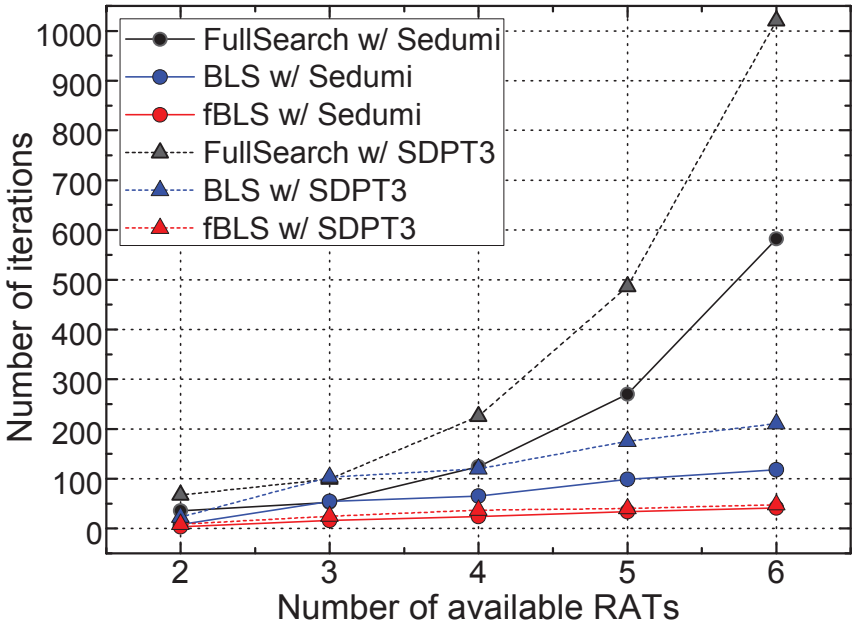


Figure 2.10: Comparison of complexity.

possible RAT combinations among the available RAT set, to compare with the proposed algorithms. The full-search method achieves the optimal result while the search space exponentially increases as the number of available RATs increases.

Fig. 2.9 shows the simulation result that represents normalized costs, which are normalized by the cost of the full-search method with Sedumi. The normalized cost of the optimal RAT set with the optimal segment allocation vector decreases as the number of available RATs increases, which shows the RAT diversity gain. Furthermore, we can see no significant performance gap between the full-search algorithm and fBLS in our simulation environment where the maximum number of RATs is 6.

On the other hand, the performance enhancement in the complexity is presented in Fig. 2.10. Because the computational complexity of the full-search algorithm is $O(2^n)$, the number of iterations to search the optimal RAT set increases exponentially as shown in the figure, while that of BLS just linearly increases as the number of

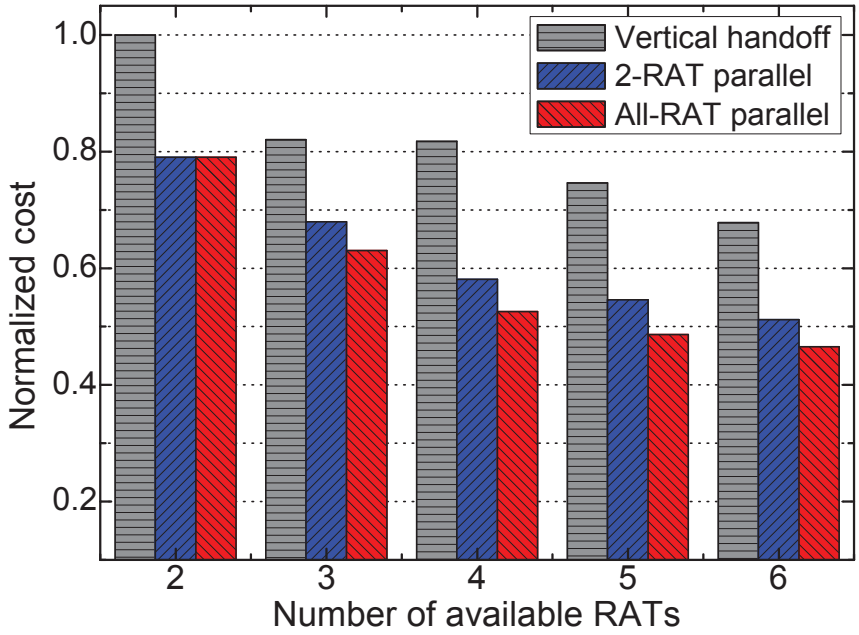
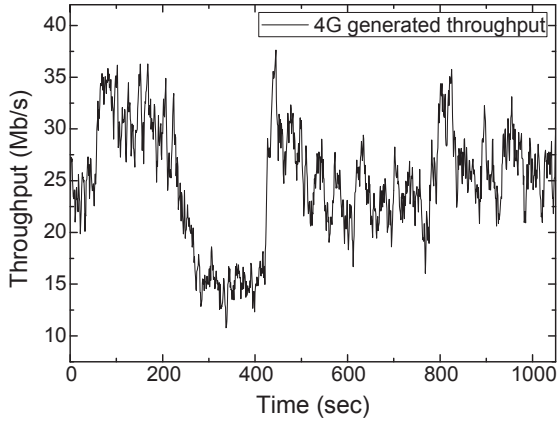


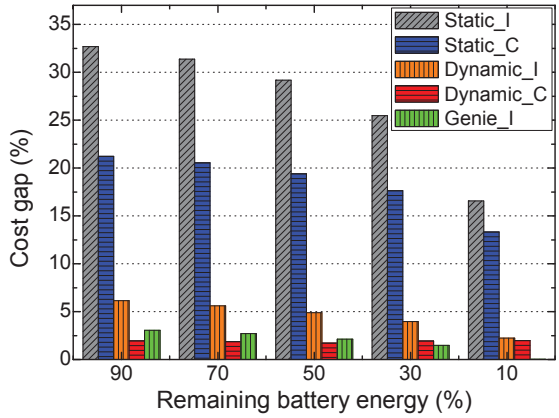
Figure 2.11: Comparison of network selection strategies.

available RATs increases. Moreover, fBLS even further reduces the iterations thanks to the initial examination with (2.9). The elapsed time of one iteration with 2.4 GHz CPU was about 0.01 s, and hence, the total elapsed time to find the optimal network operation mode with fBLS is under 0.5 s even for 6 RATs, where it is 6 to 10 s for the full-search algorithm.

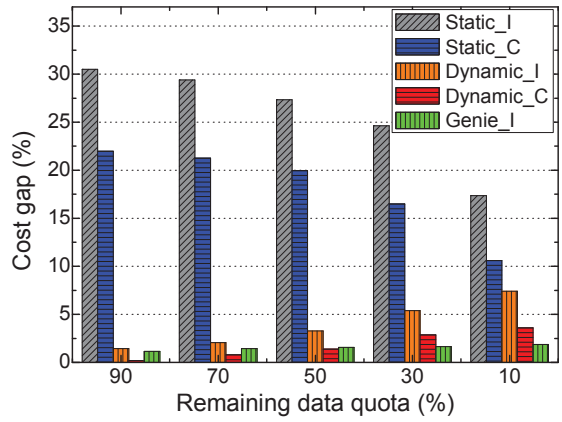
Fig. 2.11 presents a performance comparison between the vertical handoff and proposed scheme. The vertical handoff implies that only a single best RAT is selected for a file transfer, which has the minimum normalized cost among the available RATs. In addition, we compare two conditions for the proposed scheme. One is the *2-RAT parallel* scheme that can simultaneoulsy activate at most two RATs among the available RATs, and the other is the *All-RAT parallel* scheme that can exploit all the possible RAT combinations including activating all the available RATs. The simulation environment is



(a) Snapshot of 4G generated throughput.



(b) Cost gap comparison in various battery energy status.



(c) Cost gap comparison in various data quota status.

Figure 2.12: Performance comparison of the static algorithm and dynamic update algorithm.

the same as the previous evaluation, and the average cost is normalized by the cost of the vertical handoff scheme with two available RATs, which is the maximum among all the results and is set to one. As the number of available RATs increases, the vertical handoff scheme also benefits from the RAT diversity. For example, with six RATs, the normalized cost is 32% less than that with two RATs. Utilizing parallel connections enhances the performance even further. For example, the normalized costs of the 2-RAT parallel and All-RAT parallel schemes with six RATs are decreased by 26% and 36% compared with that of the vertical handoff, respectively.

To evaluate the performance of the dynamic update algorithm, we artificially generate time-varying throughput as shown in Fig. 2.12(a) for each available RAT. Three RATs, namely, 3G, 4G, and WiFi, with average throughput of 5, 25, and 15 Mb/s, respectively, are considered. The update period of the dynamic update algorithm is set to 10 s. In the simulation, a smartphone downloads several files with the size of $\{500, 1000, 2000, 4000, 8000, 16000\}$ Mbits, and measures the normalized cost. The simulation is iterated 100 times for each scheme and the results are averaged.

The comparison of the proposed algorithms is presented in Figs. 2.12(b) and 2.12(c) in terms of the remaining battery energy and data quota. "*Static*" represents the static algorithm, which searches the best RAT set and optimal segment allocation vector at the start of the file transfer and maintains the strategy until the download is completed. "*Dynamic*" represents the dynamic update algorithm, which monitors the achieved throughput and updates the optimal RAT set and segment allocation vector according to Algorithm 2. We set both the update margins m_{cur} and m_{alt} to 0.1 times the calculated cost. "*Genie*" searches the best RAT set and segment allocation vector with the exact average throughput for the entire transfer time assuming it knows future throughput. The suffix "*_I*" represents that 3G and 4G are incompatible, and hence S_A = {{3G}, {4G}, {WiFi}, {3G, WiFi}, {4G, WiFi}} while the suffix "*_C*" repre-

sents that all the RATs are compatible with one another and all the RAT combinations are available. The cost of *Genie_C*, which performs the best among all the compared schemes, is set to the reference value. Then, the cost gaps, i.e., the ratios of the average increased cost of the proposed schemes to the cost of *Genie_C*, are presented in Figs. 2.12(b) and 2.12(c).

From the simulation results, we confirm that the performances of the compatible cases are generally better than those of the incompatible cases. In some cases, *Dynamic_C* performs even better than *Genie_I* because of more options of the RAT selection including $\{$3G, 4G$\}$ and $\{$3G, 4G, WiFi$\}$. Second, the cost gap of the static algorithm can be significantly increased due to the dynamic network condition. However, the dynamic update algorithm can effectively reduce the cost gap. In particular, we observe that the cost gap can be suppressed under 5% by *Dynamic_C* for all the status of remaining energy and data.

2.7 Summary

In this chapter, we have proposed and analyzed the multi-RAT selection and segment allocation scheme for smartphones based on a multi-attribute cost function. The cost function properly normalizes and integrates the transfer completion time, energy consumption, and service charge in n RAT-based smartphones for file transfer services. Based on the device's current status-dependent cost model, the proposed scheme adaptively activates n RAT interfaces to minimize the cost function. Considering the increasing number of RATs integrated in the future smartphones, the proposed algorithm, namely fBLS (fast bidirectional linear search), achieves a nearly optimal RAT selection and segment allocation with reduced computational complexity. It shows no significant performance gap from the full-search algorithm, while the exponential complexity is

reduced to linear. In addition, we have shown that the performance gain increases as the number of available RATs increases due to the RAT diversity, and it can be more improved by exploiting the parallel activation of multiple RATs.

Chapter 3

Video Streaming Mode

3.1 Introduction

Today's mobile networks such as Long Term Evolution (LTE) and Wi-Fi are fast enough to support high quality video services including High Definition (HD) video. However, such high quality video services drive smartphones and their users to fast consume battery energy and available data quota reserved for users' data plan. Therefore, data-hungry users experience unsatisfactory playback time of multimedia services or energy-hungry smartphones suffer from short lifetime for multimedia services. To address this problem, video streaming services have been studied in consideration of data usage or energy consumption [20–22]. However, smartphone users are generally concerned about both remaining data quota and battery energy. Typical users may stop playing video when either available data quota or energy is about to deplete. Otherwise, they would enjoy the best quality of the video as long as the remaining data quota and battery energy are allowed.

Meanwhile, a recent study reveals that smartphone users tend to watch only about 60% of online video clips to the end [39]. Such tendency results in undesirable waste

of data and energy for the following reason. Typical HTTP-based video streaming services adopt video data prefetching schemes in order to combat various unfavorable obstacles against satisfactory services such as unstable network bandwidth and wireless channel quality fluctuation. In case when a user quits an ongoing video streaming, the prefetched video data shall be dropped. It implies that the user ends up wasting the energy and data consumed for the dropped video data. The expected amount of the waste would naturally increase as the viewed video quality is enhanced.

In this work, we propose an operation policy employing both LTE and WiFi interfaces. Simply, we adopt the term of *multi-RAT* standing for both LTE and WiFi interfaces. The proposed policy balances between the video quality, i.e., video encoding rate, and the video playback time allowed with remaining energy in a battery. For proper operation of the policy, we formulate an objective function maximizing a normalized utility. It is obtained by subtracting the normalized cost of the playback from the utility of video quality. The normalized cost of the playback is determined by the data quota usage and the energy consumption for ongoing playback. The objective function is tuned with the network mode selection, video encoding rate selection, and chunk download cycle for its maximization.

The problem formulation steps are summarized as follows. Firstly, we solve a subproblem to find the optimal chunk download cycle that maximizes overall video playback time of smartphone with given remaining battery energy and data quota, furthermore considering the unnecessary energy and data waste. In the proposed policy, we adopt the statistics from [39] that all video clips are not fully played to the end. Then, we design a normalized cost function in terms of both data usage and energy consumption while incorporating expected data and energy waste. By minimizing the cost, we can selectively save more depleting resource as much as possible by monitoring their current status. After that, saved data quota and/or battery energy are used for other

video streaming services later. Consequently, we can achieve the goal maximizing the overall playback time. Secondly, we incorporate the solved sub-problem into the design of the proposed operation policy. From the fact that LTE and WiFi interfaces can be simultaneously serviced for recent smartphones, we define three network operation modes, i.e., *LTE-only*, *WiFi-only*, and *LTE+WiFi*. To our best knowledge, we first propose an operation policy considering the *LTE+WiFi* mode. In the proposed operation policy, the most appropriate network operation mode is selected to maximize the utility function. For the performance evaluation close to the real world, we apply a practical user's daily pattern regarding video watching time. Then, we show that the proposed algorithm effectively balances the average video encoding rate and total playback time with given various data quota and battery energy level. The algorithm maximizes the video quality, i.e., video encoding rate, as much as possible so far as the remaining resources are enough to play the number of the expected video arrivals of the day with the selected video quality.

The remainder of this chapter is organized as follows. In Section 3.2, the HTTP and multi-RAT based video streaming models are described. We analyze the data and energy usage for the video playback in Section 3.3, and formulate the objective function to balance the video quality and total video playback time in Section 3.4. The performance evaluation is presented in Section 3.5, and finally, we summarize this chapter in Section 3.6.

3.2 System Model

3.2.1 HTTP-based playback model

We consider HTTP-based video streaming for our playback model. It is the most popular method for today's video streaming services. As illustrated in Fig. 3.1, a video

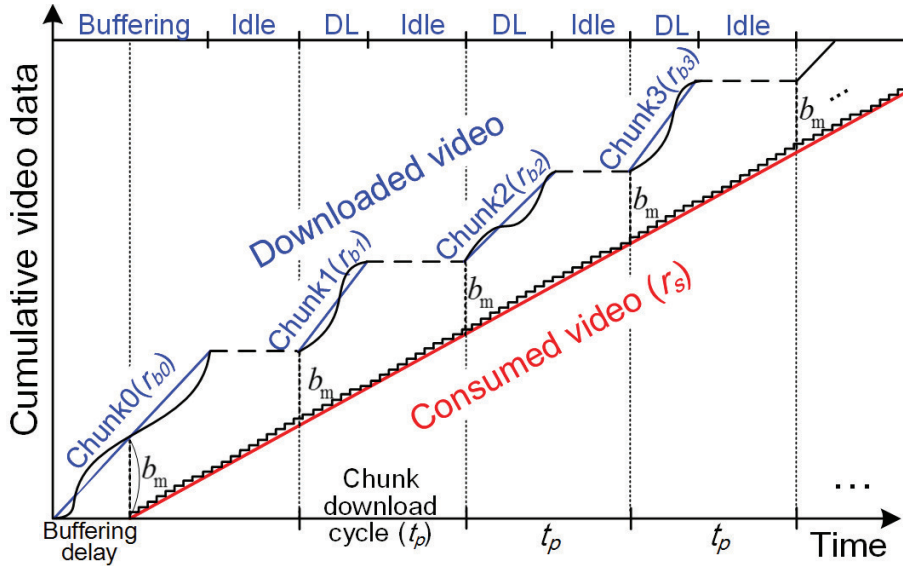


Figure 3.1: Typical online video playback model.

service begins by downloading a small piece of the entire video data, called a *chunk*. Only if the player's buffer is filled with video data of a predefined size b_m, the video playback starts. This initial buffering is typically used to prevent buffer underflow due to time-varying network bandwidth and wireless channel quality fluctuation.

Once the amount of unplayed video data in the buffer reaches the threshold b_m, the video player requests another chunk, whose size is b_p bytes, for the next cycle. This cyclic procedure, referred to as the *Chunk Download Cycle (CDC)*, repeats until the user quits the service or all the chunks are downloaded. Note that $b_m = r_s \cdot T_m$, where T_m and r_s are possible playback time with the minimum buffered data and video encoding rate, respectively. In this work, weassume that a constant value of T_m is given and r_s is chosen among the available encoding rate set for each video.

The slopes of the lower red and upper blue lines represent the average video encoding and chunk download rates, respectively. A new TCP session is established at the *DL (ON) phase*, and a chunk of a specific size is requested for download while

the video data in the buffer are being played simultaneously. When downloading the chunk is completed in the cycle, the TCP session is closed. Then, only the video data remaining in the buffer are played for the *Idle (OFF) phase*. The long-term average download rate is converged to the video encoding rate by using the ON-OFF cycle even though the network bandwidth should be larger than the encoding rate.

To maximize the playback time, the period of CDC, denoted by t_p, needs to be adjusted according to the current data quota, battery energy, and user's leaving-in-midstream probability. For example, if the current battery energy is running out, the smartphone better increases t_p to save the energy because it reduces the occurrence of the energy overhead for the tail time[1] at the end of every *DL* phase. However, the increment may cause more data and energy waste if the user stops playing the video prior to the completion of the entire playback. Consequently, we can recognize that there exists a trade-off relationship in the period of the CDC. In this work, we assume the network bandwidth is fast enough to smoothly play considered video clips.

3.2.2 LTE/WiFi-based multipath video streaming model

Multipath-based communications mainly aim at enhancing the connection robustness and boosting the data transfer speed. The multipath-based approaches exploiting heterogeneous networks such as 3G/4G cellular and WiFi are discussed and studied especially for video streaming services typically demanding the high throughput and reliability. Traditionally, previous efforts focus on quality-adaptive bandwidth aggregation algorithms that determine how much and when to schedule video segments to the activated multiple networks considering their dynamics to properly support a specific video quality. However, recently another attribute besides the video quality, i.e.,

[1] whenever a data transfer is completed, the network interface waits for a while, called *tail time*, prior to its switching to the sleep mode.

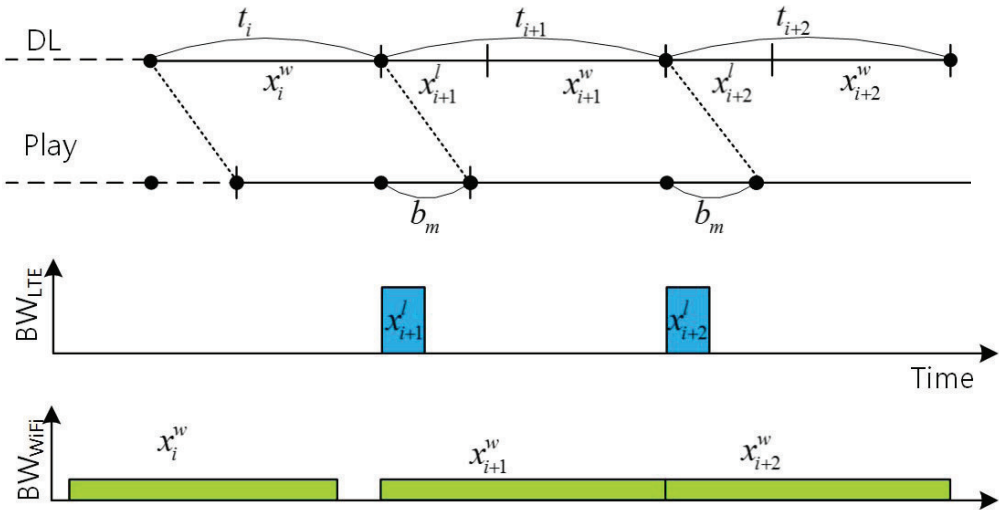
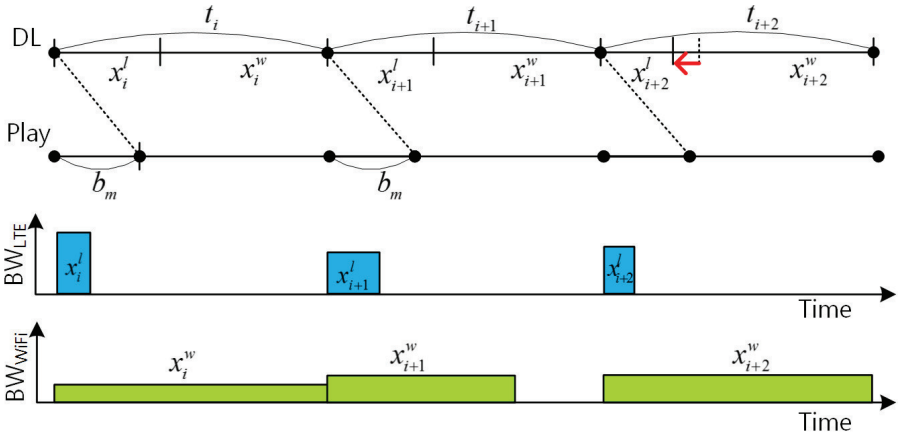


Figure 3.2: LTE/WiFi multi-path video streaming algorithm.

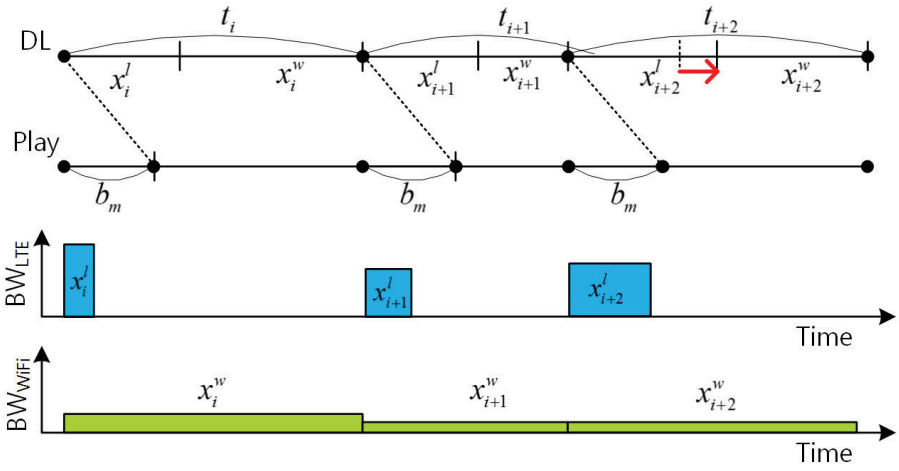
the energy efficiency, is also studied for the multipath-based video streaming protocols [20].

To support the energy, service charge, and performance-aware video streaming algorithm with multiple network paths, i.e, LTE and WiFi, we propose the multipath video streaming model based on HTTP as described in Fig. 3.2. The upper two lines represent video playback time positions, where the first and second lines mean the corresponding downloaded video and playbacked video, respectively. The lower two graphs represent LTE and WiFi bandwidths (y-axis) with the corresponding downloaded timing (x-axis).

When WiFi is available, and a smartphone supports the *LTE+WiFi* multipath data transfer, it activates WiFi and starts downloading the next chunk for a while to estimate the achievable throughput of the WiFi network. During that time, a multipath-based streaming algorithm computes the video utility proposed in this work for all the possible video parameter settings. If the optimal network selection with the corresponding encoding rate is *LTE+WiFi*, the next chunk is divided into two parts, i.e, the



(a) Increasing WiFi throughput



(b) Decreasing WiFi throughput

Figure 3.3: Effect of time-varying WiFi throughput.

LTE and WiFi parts, and requested in the next chunk download cycle. We allocate the front of the chunk to the LTE part because the LTE link is generally more reliable and has plenty of bandwidth than the WiFi link. That also gives more time to recover the rearward chunk when the WiFi network is unexpectedly unavailable.

Basically, the WiFi is free of charge and consumes much less power than the LTE, and therefore, the proposed scheme fully utilizes the WiFi bandwidth at first. Then the remaining amount of the video chunk is downloaded via the LTE network as described in Fig. 3.2. In this streaming policy, two cases may occur due to the network bandwidth dynamics, which are represented in Fig. 3.3(a) and 3.3(b). If the average WiFi throughput in the CDC (t_{i+1}) increases than the previously estimated throughput as shown in Fig. 3.3(a), downloading the WiFi part in the chunk is completed before the next CDC starts. Then, the size of the WiFi part can be increased while reducing the LTE part to save more energy and LTE data consumption by recalculating the optimal CDC by updating throughput estimation reflecting the increased level for the next CDC (t_{i+2}). Furthermore, if the WiFi throughput becomes high enough to cover the current video encoding rate, all the chunk is allocated to the WiFi part removing the LTE part in the next CDC.

On the contrary, if the average WiFi thorughput in the CDC (t_{i+1}) is lower than the estimated throughput as shown in Fig. 3.3(b), the remaining video size meets b_m before the current CDC ends. Then, the next CDC starts earlier requesting the more LTE part in the next chunk to compensate the lack of WiFi throughput. In this case the optimal CDC also can be recalculated by updating the throughput estimation reflecting the decreased level for the next CDC (t_{i+2}). Moreover, if the WiFi throughput is too low, the video utility with the lower encoding rate would be better than that with the original encoding rate. Therefore, the next video chunk can be requested with the lower encoding rate.

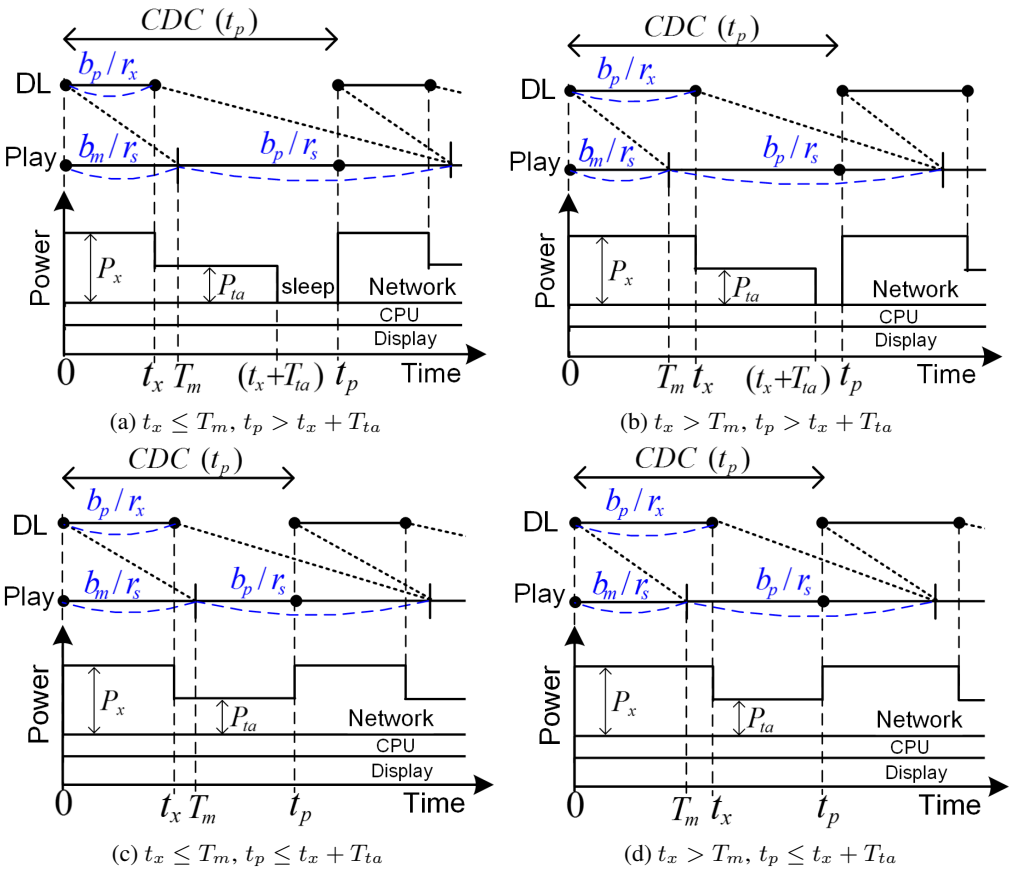


Figure 3.4: Four types of power consumption.

3.3 Chunk Download Cycle based Analysis

For the cost function formulation, we analyze the data and energy consumption based on the CDC. As shown in Fig. 3.4, chunks are individually downloaded in each cycle while the player continues to consume video data for its playback. Accordingly, the network interface consumes the power for data receptions during the download time t_x. After that, the power consumption level falls to the base level P_{ta} for the tail time T_{ta}. If $t_x \leq T_m$ as described in Figs. 3.4(a) and 3.4(c), the chunk download bis completed before the remaining data in the buffer reaches b_m. Otherwise, the chunk downloaded in the previous CDC is consumed before the current chunk download is completed as represented in Figs. 3.4(b) and 3.4(d). On the other hand, the network interface might not be able to sleep if t_p is less than $\{t_x + T_{ta}\}$ as presented in Figs. 3.4(c) and 3.4(d). All these four cases according to the inequality conditions have to be considered to determine the average data and energy consumption including the expected waste. The parameters used in the analysis are summarized in Table 3.1.

In this work, we handle the expected data usage and power consumption metric at the network interface, which is tunable with t_p. Meanwhile, the power consumption of display and CPU has nothing to do with t_p, and hence, we simply assume that they are constant.

3.3.1 Data and energy consumption rate

The average data consumption rate during the video streaming is identical to the video encoding rate r_s. However, the average network interface power consumption, $\overline{P}$, is involved with t_p as described in Fig. 3.4. Then, $\overline{P}$ is represented as follows:

Table 3.1: Description of video parameters

Parameter	Description
r_s	Video source encoding rate
t_p	The period of Chunk Download Cycle (CDC)
b_p	Chunk size per download cycle ($= r_s t_p$)
r_x	Average chunk transfer rate
t_x	Chunk transfer time ($= b_p/r_x$)
T_m	The minimum buffer-time
b_m	The minimum buffer threshold ($= r_s T_m$)
γ	The ratio of r_s to r_x ($= r_s/r_x$)
P_x	Network interface receive power ($= ar_x + b$)
P_{ta}, T_{ta}	Network interface tail power and tail time

$$\overline{P}=\begin{cases}\frac{1}{t_p}\{P_x t_x+ P_{ta}T_{ta}\} = \gamma P_x + \frac{E_{ta}}{t_p}, & \text{if } t_p > t_x + T_{ta},\\ \frac{1}{t_p}\{P_x t_x+ P_{ta}(t_p - t_x)\}= \gamma(P_x - P_{ta}) + P_{ta}, & \text{otherwise,}\end{cases} \tag{3.1}$$

where $t_x = \gamma t_p$ from Table 3.1, and the energy for the tail time $E_{ta} = P_{ta}T_{ta}$. From this equation, we observe that $\overline{P}$ can be controlled by t_p under the condition that $t_p > t_x + T_{ta}$. Otherwise, t_p is not related to $\overline{P}$ since the network interface cannot switch to the sleep mode.

3.3.2 Expected waste of data and energy

The amount of data and energy waste is determined by the time duration between the beginning instant and the stopping instant of the playback prior to the end of the video clip. If the user leaves outside the initial buffering phase, the user's leaving instant t_l, i.e., watching time, must be always located within a CDC range because the same CDC is repeated. Then, we can define a relative leaving instant t, representing the leaving instant within the last CDC, which is expressed as $t = t_l \bmod t_p$. The authors of [40]

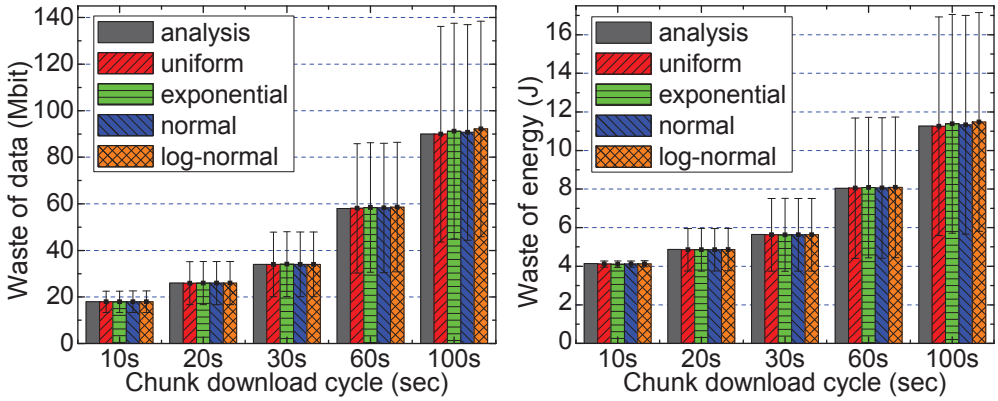


Figure 3.5: Comparison of data and energy waste according to different distribution of user's leaving instant.

empirically reveal that the remainder of modulo operation over arbitrary random variables tends to be uniformly distributed in the denominator range. It implies that the relative leaving instant t approximately follows a uniform distribution regardless of the watching time distribution. This approximation is more accurate when $t_l \gg t_p$, and generally it is true that *watching time* is much longer than t_p.

We first show a comparison between the numerical analysis and simulation result of the expected data and energy waste in Fig. 3.5 according to t_p. The *analysis* presented later is conducted based on the assumption that the leaving instants in a single CDC are uniformly distributed. The simulation result is from different watching time distributions before the user's leaving, i.e, *uniform, exponential, normal, and lognormal distribution*, where the average watching time is 300 s. Each bar and vertical line represents the mean and standard deviation of the data and energy waste, respectively. The result shows no significant difference from one another, and therefore we claim that the expected data and energy waste can be estimated regardless of the watching time distribution.

We present a detailed analysis of the data and energy waste for a single CDC as fol-

lows. The expected data waste can be obtained from the expectation of the remaining buffered data assuming user's leaving occurs between 0 and t_p. First, when $t_x \leq T_m$ as presented in Fig. 3.4(a) or 3.4(c), the expected data waste is derived by

$$\begin{aligned}\overline{Q}_w = \frac{1}{t_p}\Bigg[\int_0^{t_x}\big\{(r_s T_m - r_s t) + r_x t\big\}dt + \int_{t_x}^{T_m}\big\{(r_s T_m - r_s t) + b_p\big\}dt \\ + \int_{T_m}^{t_p}\big\{b_p - (r_s t - r_s T_m)\big\}dt\Bigg] = \frac{1}{2} r_s\Big\{(1-\gamma)t_p + 2T_m\Big\},\end{aligned} \tag{3.2}$$

where t is the relative leaving instant, which is assumed to be uniformly distributed in $[0, t_p]$, and hence, its probability density function is obtained by $1/t_p$. In the same way, the expected value for $t_x > T_m$ can be easily obtained, of which the result is exactly the same as (3.2).

The energy waste is represented as the consumed energy for chunk downloading (E_{dl}) subtracted by the energy portion for the watched video (E_{used}) as follows:

$$\begin{aligned}E_w &= E_{dl} - E_{used} \\ &= \begin{cases} E_{k-1}(t_p) + E_k(t) - \overline{P}\cdot(t_p - T_m + t), & \text{if } t < T_m, \\ E_k(t) - \overline{P}\cdot(t - T_m), & \text{otherwise,} \end{cases}\end{aligned} \tag{3.3}$$

where $E_k(t)$ denotes the consumed energy to download the k-th chunk for t from the start of the CDC. Both equations are then represented as $E_k(t) + \overline{P}\cdot(T_m - t)$ because $E_{k-1}(t_p) = \overline{P}\cdot t_p$. When $t < T_m$, the video data downloaded in the previous cycle is being consumed, and the consumed energy for the current cycle ($E_k(t)$) is therefore totally wasted. On the other hand, when $t \geq T_m$, the network interface wastes the energy needed to receive the unplayed data in the current cycle.

Then, the expected energy waste $\overline{E}_w$ for $t_p > t_x + T_{ta}$ is obtained by

$$
\begin{aligned}
\overline{E}_w = \frac{1}{t_p}\Bigg[& \int_0^{t_x} P_x t dt + \int_{t_x}^{t_x+T_{ta}} (P_x t_x + P_{ta} t) dt \\
& + \int_{t_x+T_{ta}}^{t_p} (P_x t_x + P_{ta} T_{ta}) dt + \int_0^{t_p} \overline{P}(T_m - t) dt \Bigg] \\
= & \frac{P_x \gamma (1-\gamma)}{2} t_p + E_{ta}\Big(T_m - \frac{T_{ta}}{2}\Big)\frac{1}{t_p} + P_x T_m \gamma + \frac{E_{ta}}{2}.
\end{aligned}
\tag{3.4}
$$

In the same manner, we obtain $\overline{E}_w$ for $t_p \leq t_x + T_{ta}$ by

$$
\overline{E}_w = \frac{(P_x + P_{ta})\gamma(1-\gamma)}{2} t_p + P_x T_m \gamma + P_{ta} T_m (1-\gamma). \tag{3.5}
$$

As a result, both data and energy consumption models including their waste are represented as a function of t_p.

3.4 Proposed Scheme

3.4.1 Problem formulation

Our objective is balancing the video quality, represented as a function of the video encoding rate, and total playback time. We formulate the objective function of this work as follows:

$$
\begin{aligned}
U(S^*, r_s^*, t_p^*) = & \max_{S, r_s, t_p} \ \alpha F_u(r_s) - (1-\alpha) F_c(S, r_s, t_p), \\
& \text{s.t. } S \in S_A, \ \ r_s \in R_s, \\
& t_{\min} \leq t_p \leq t_{\max},
\end{aligned}
\tag{3.6}
$$

where S, r_s, and t_p are the network selection, video encoding rate, and the period of

CDC, respectively, and S^*, r_s^*, and t_p^* represent their optimal values. $U(S^*, r_s^*, t_p^*)$ is the corresponding optimal utility function that we try to find out. The utility function is a weighted sum of the normalized video utility and negative cost, denoted by $F_u(r_s)$ and $F_c(S, r_s, t_p)$, representing the video quality and total playback time, respectively. α is a tuning parameter that reflects the user preference between the video quality and total playback time, where the default value is 0.5. As α becomes greater, the algorithm operates enhancing the video quality more while the playback time reduces, and vice versa. S_A and R_s are available sets of the network selection and encoding rate, respectively, e.g., S_A={ {LTE}, {WiFi}, {LTE, WiFi} }, $R_s = \{r_1, r_2, \cdots, r_m\}$, where r_1 is the lowest video encoding rate and r_m is the highest video encoding rate. $t_{\min}$ and $t_{\max}$ are the minimum and maximum values that t_p can be set. These values are discussed in Section 3.4.2.

3.4.2 Subproblem I: Playback time maximization

Based on the analysis, we propose a policy for the video playback time maximization, called EQ-video. First, we have the average data usage from (3.2) by $\overline{Q} = r_s \cdot E\{t_v\} + p_{\text{cut}} \cdot \overline{Q}_w$, where $E\{t_v\}$ and p_{cut} are the expected video watching time and the probability of user's leaving-in-midstream, respectively.[2] Next, from (3.4) and (3.5), the average energy consumption is derived by

$$\overline{E} = \overline{P} \cdot E\{t_v\} + p_{\text{cut}} \cdot \overline{E}_w. \tag{3.7}$$

From the equations for $\overline{Q}$ and $\overline{E}$, we define normalized data cost $C_q = \overline{Q}/Q_r$ and energy cost $C_e = \overline{E}/E_r$, where Q_r and E_r are the remaining data quota and battery energy, respectively. In these equations, the costs indicate the comparative degrees

[2]These values can be predicted by processing the user's history data. We apply the Auto-Regressive-Moving-Average (ARMA) tool for the prediction.

required for a video playback with given remaining resources including data quota and battery energy at a playback beginning time.

C_q, $\overline{Q}$, $\overline{Q}_w$ and t_p are linked in a chain with linear equations, thus leading to the fact that C_q is a linear increasing function of t_p. Therefore, C_q is convex in terms of t_p. Similarly, we can reason that C_e is also a linear increasing function of t_p when $t_p \leq (t_x + T_{ta})$ from (3.1) and (3.5), and hence C_e is convex in this case. When $t_p > (t_x + T_{ta})$, C_e is convex if

$$\frac{\partial^2 C_e}{\partial {t_p}^2} = \frac{2E_{ta}}{{t_p}^3 E_r}\Big(E\{t_v\} + p_{\text{cut}} T_m - \frac{T_m}{2}\Big) \geq 0. \tag{3.8}$$

Otherwise, C_e is not a convex function but an increasing function because the following is always true:

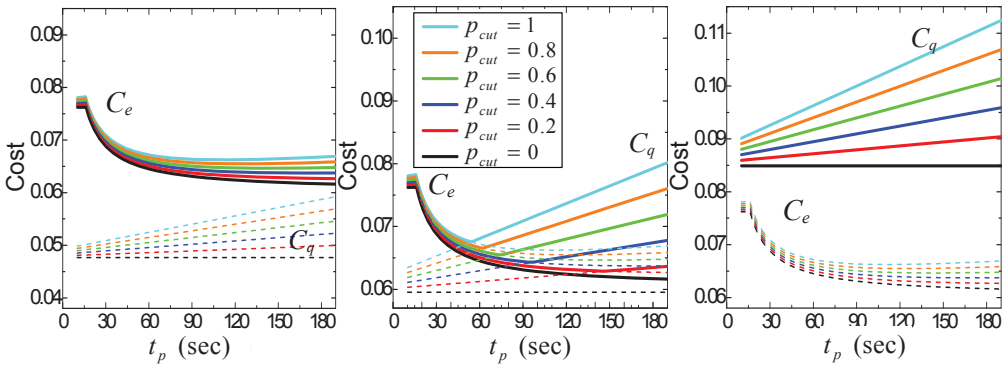
$$\frac{\partial C_e}{\partial t_p} = \frac{P_x \gamma(1-\gamma)}{2} - \frac{E_{ta}}{t_p E_r}\Big(E\{t_v\} + p_{\text{cut}} T_m - \frac{T_m}{2}\Big) > 0, \tag{3.9}$$

where $0 < \gamma (= \frac{r_s}{r_x}) < 1$ from our assumption. Therefore, C_e is at least quasiconvex in this region because the increasing function is quasiconvex.

From these cost functions, we derive the objective function for *EQ-video* by

$$\begin{aligned} C(t_p^*) = \min_{t_p} \{\max(C_q, C_e)\}, \\ s.t.\ T_m \leq t_p \leq T_{\text{dur}}, \end{aligned} \tag{3.10}$$

where T_{dur} is the video playback duration, and $t_p \geq T_m$ from $b_p \geq b_m$. In fact, it is known that the maximum of quasiconvex functions preserves quasiconvexity, and hence, the objective function has the optimal solutions in two regions, i.e., $T_m \leq t_p <$



(a) Energy-limited (b) Intermediate (c) Data-limited

Figure 3.6: Three types of normalized cost function.

$(t_x + T_{ta})$ and $(t_x + T_{ta}) \leq t_p \leq T_{\text{dur}}$, respectively. Therefore, the optimal t_p can be obtained by choosing the smaller one of the two solutions in the two regions.

Fig. 3.6 shows examples for three types of the cost function intersections according to different remaining data quota and battery energy status for the case when (3.8) is satisfied. Each solid line represents the normalized cost for the different p_{cut}. First, Fig. 3.6(a) represents the case when the remaining battery energy is relatively deficient, and hence, the normalized energy cost C_e is higher than the normalized data cost C_q for all t_p. Second, the two cost functions intersect each other as shown in Fig. 3.6(b) in the intermediate condition of the remaining resource status. Finally, Fig. 3.6(c) presents the case when the remaining data quota is relatively deficient, and hence, C_q is higher than C_e for all t_p.

3.4.3 Subproblem II: Balancing between encoding rate and total playback time

Based on the cost model discussed in Section 3.4.2, the complete objective function to balance the video quality and total playback time, which is conceptually presented

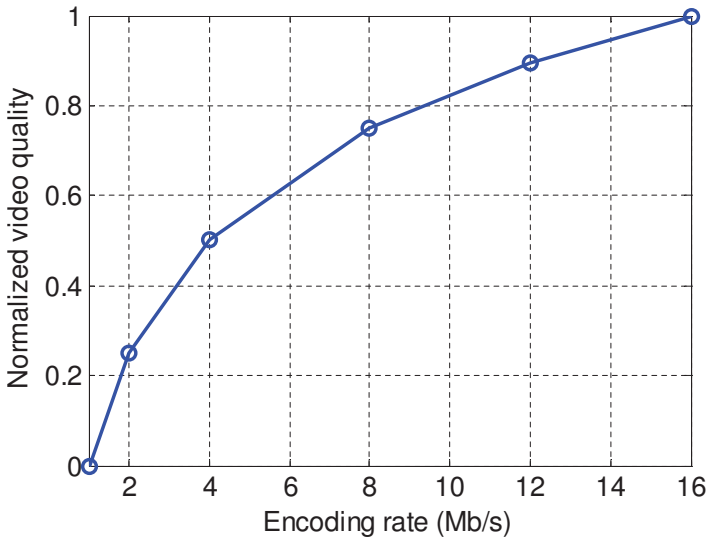


Figure 3.7: Normalized video quality function.

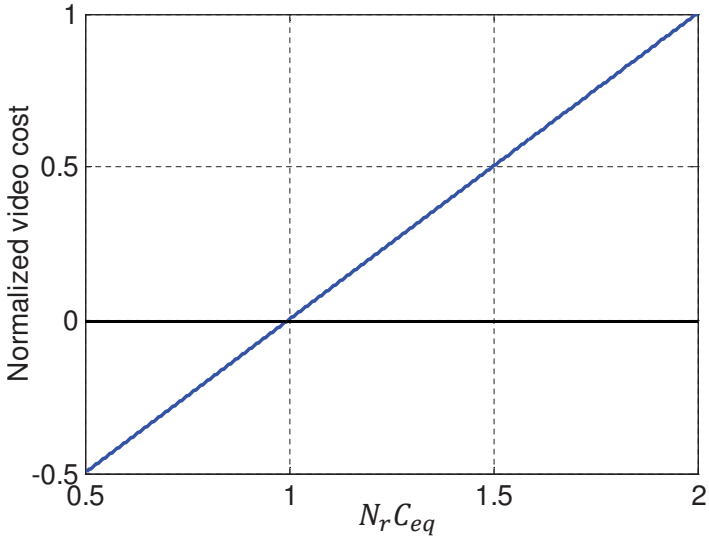


Figure 3.8: Normalized video cost function.

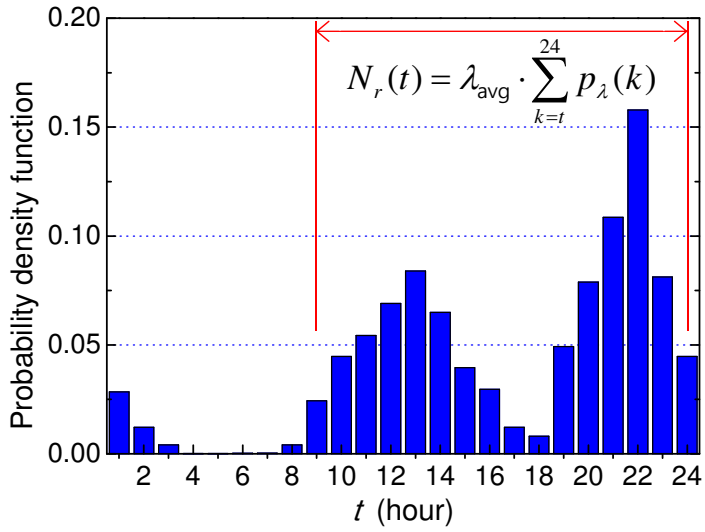


Figure 3.9: Example of daily video arrival distribution.

in (3.6), can be represented as follows:

$$\begin{aligned} U(S^*, r_s^*, t_p^*) = \max_{S, r_s} & \alpha \frac{f_u(r_s) - f_u(r_{\min})}{f_u(r_{\max}) - f_u(r_{\min})} \\ & - (1-\alpha) \left\{ f_c\big(N_r(t) C_{eq}(S, r_s, t_p^*)\big) \right\}, \\ \text{s.t. } & S \in S_A, \;\; r_s \in R_s, \\ & T_m \le t_p \le T_{\text{dur}}, \end{aligned} \tag{3.11}$$

where $f_u(r_s)$ is a logarithmic function for the video utility according to the encoding rate r_s, and it is normalized by the $\max$-$\min$ method that is commonly used in MADM. Then, the normalized video utility has a value between 0 and 1 as described in Fig. 3.7. $f_c(x)$ is the normalized cost function proportional to x, where x is $N_r(t)C_{eq}(S, r_s, t_p^*)$ in (3.11), where $N_r(t)$ is the expected number of video arrivals from the current time t to the end time of the day, i.e., 12 a.m. N_r is based on user-specific video view-

ing statistics composed of the number and time of daily video watching requests. $C_{eq}(S, r_s, t_p^*)$ is the LTE/WiFi-based energy and data cost function of the network selection S, selected encoding rate r_s, and the corresponding optimal CDC t_p^*. In this section, the cost function C_{eq} becomes the function of S, r_s, t_p with multiple network interfaces while C_{eq} is only the function of t_p with given r_s and single network in the previous section. This cost function implies that its reciprocal can be regarded as the available number of videos that can be watched with the current data quota and battery energy resources, denoted by N_{eq}.

On the other hand, based on the user-specific video viewing statistics, the smartphone builds an hourly video arrival distribution by averaging the number of video arrivals for the same hour of several days. Fig. 3.9 shows an example of the daily video arrival distribution with hourly time bins. Then, $N_r(t)$ is calculated by expectation of the video arrivals from t to the end time of the day as represented in the equation on the top of the figure, where λ_{avg} is the daily video arrival rate. We omit arguments of N_r and C_{eq} for convenience' sake. Then, we design the normalized cost function $f_c(N_r C_{eq})$ as described in Fig. 3.8, which implies the following characteristics:

1) If the multiplication of N_r and C_{eq} is one, which means that N_{eq} is exactly the same as N_r, the normalized cost function is zero.

2) The case when $N_r C_{eq}$ is greater than one, equivalently $N_r > N_{eq}$, means that the cost with the current setting is too high to cover the whole number of the remaining video arrivals, and hence, the normalized cost is positively increased.

3) Otherwise, when $N_r < N_{eq}$, the cost with the current setting is so low, and the resources are likely to remain at the end time of the day, and hence, the normalized cost is negatively decreased. The negative sign at the normalized cost function in (3.11) changes the negative cost to the positive value and increases the overall utility function.

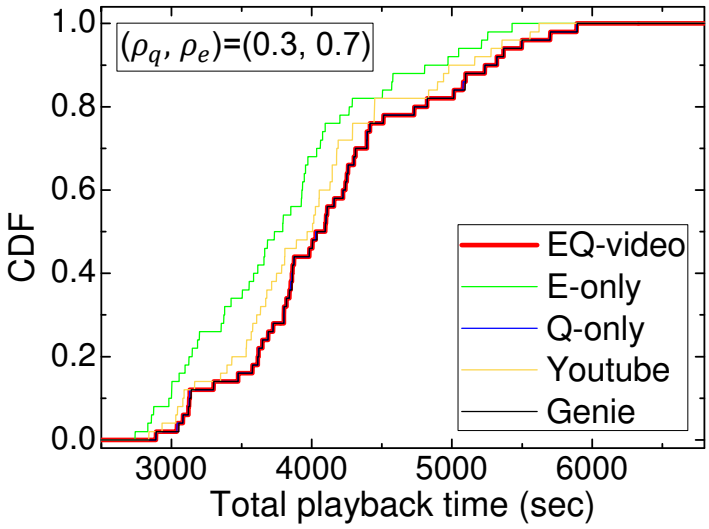
3.5 Performance Evaluation

3.5.1 Maximization of playback time with a single path

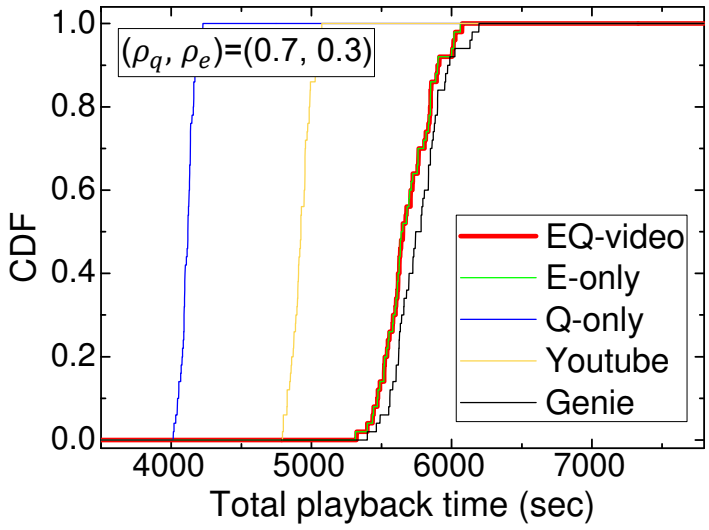
We use a Matlab-based simulator to evaluate the performance of *EQ-video*. First, a number of videos are generated with various attributes composed of the watching time, leaving-in-midstream indicator, and video encoding rate. The video watching time follows the normal distribution with the average duration of 300 s, and user's leaving-in-midstream probability is assumed to be 0.4. We assume that the video encoding rate is randomly chosen by the user among $\{0.5, 1, 2\}$ Mb/s for each video and used as an input to find out the optimal t_p. The average network throughput is set to 8 Mb/s. In the simulation, videos are played until either the initially given data quota or the battery energy depletes. The total data quota and battery energy are set to 2 GB and 2000 mAh, respectively. Then, the normalized remaining resources ρ_q and ρ_e, which are the ratios of Q_r and E_r to the total data quota and battery energy, respectively, are set in the range from 0 to 1. We apply the linear-fit model for the power-throughput relationship of the network interface power consumption, where the coefficients are obtained from [31] assuming the LTE environment. T_m and T_{ta} are set to 5 s and 10 s, respectively.

We compare *EQ-video* with the following schemes: *Genie* exhaustively searches the optimal t_p for each video assuming that it is omniscient to know the watching time of every played video, including whether the playback is stopped in midstream or not. For *E-only*/*Q-only*, t_p is adaptively set by only considering either the remaining data quota or energy. *YouTube* follows the chunk download pattern of YouTube's native application, which has about 40 s download time and 60 s idle time, making the cycle equal to 100 s.

In Fig. 3.10, the cumulative distribution functions (CDFs) of the total playback



(a) Data limit condition



(b) Energy limit condition

Figure 3.10: Comparison of video streaming schemes.

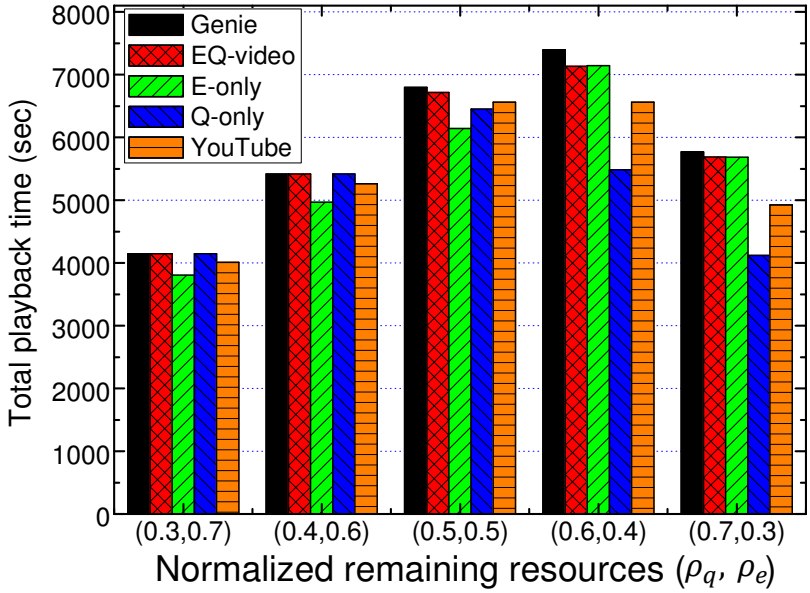


Figure 3.11: Average playback time of different schemes (sec).

time for 50 experiments with different schemes are presented. Figs. 3.10(a) and 3.10(b) show the cases when (ρ_q, ρ_e) are set to $(0.3, 0.7)$ and $(0.7, 0.3)$, respectively. The average performance for each scheme with various (ρ_q, ρ_e) is presented in Fig. 3.11, where the x-axis and y-axis represent different (ρ_q, ρ_e) cases and the average playback time, respectively. The performance gap in percentage between *Genie* and the other scheme is summarized in Table 3.2, where the first row represents the different (Q_r, E_r) cases. In the result, the maximum ratios of the reduced playback time for *Q-only*, *E-only*, and *YouTube* compared to *Genie* are 28.6%, 9.6%, and 14.6%, respectively among the five cases of (ρ_q, ρ_e).

On the other hand, *EQ-video* performs like the *Q-only* in a data-limited condition (Fig. 3.10(a)) while it performs like the *E-only* in an energy-limited condition (Fig. 3.10(b)). Furthermore, it considers both of the remaining data quota and battery energy in the intermediate condition, e.g., (0.5, 0.5). Therefore, we confirm that

Table 3.2: Comparison of available watching time (sec)

(ρ_q, ρ_e)	(0.3, 0.7)	(0.4, 0.6)	(0.5, 0.5)	(0.6, 0.4)	(0.7, 0.3)
Genie	4148	5417	6797	7393	5767
Proposed	0	0	-1.2%	-3.5%	-1.4%
E-only	-8.3%	-8.3%	-9.6%	-3.4%	-1.5%
Q-only	0	0	-5.1%	-25.8%	-28.6%
YouTube	-3.3%	-2.9%	-3.5%	-11.3%	-14.6%

EQ-video selectively performs well for all the cases of the remaining resource status, and the maximum ratio of the reduced playback time compared to *Genie* is only 3.5% when (ρ_q, ρ_e) is (0.6, 0.4).

3.5.2 Balancing between video quality and playback time with LTE/WiFi multiple networks

First of all, we generate an example distribution of daily video watching arrivals to evaluate the proposed algorithm that balances between video quality and playback time. 3.9 describes the video arrival distribution, where one bar is the probability function of video arrival in the corresponding time-bin that represents one hour. In this example case, the daily video arrival rate λ_{avg} is assumed to be 10, and the arrivals follow Poisson distribution with λ_{avg}. In the simulation, the video arrival distribution is empirically constructed by measuring video arrivals for a specific time, e.g., for 10 days. Based on the distribution of empirical data, the remaining video arrivals in a specific time t for the day in consideration is calculated by the average rate multiplied by the sum of the probability mass function for $t < k < 24$ as represented in Fig. 3.9.

The video arrival distribution based simulation is conducted with the simulation setup explained as follows: As explained above, the video requests arrive in a day according to the Poisson process with the average number of the arrivals is 10. Time-varying throughput for LTE and WiFi is generated that follows the normal distribution

with average 12 Mbit/s and 8 Mbit/s, respectively. The available video encoding rate set is in $\{1,2,4,8,12,16\}$ Mbit/s. The average video watching time is assumed to be 5 minutes and to follow the lognormal distribution. The average leaving-in-midstream probability is set to 0.4 for this simulation.

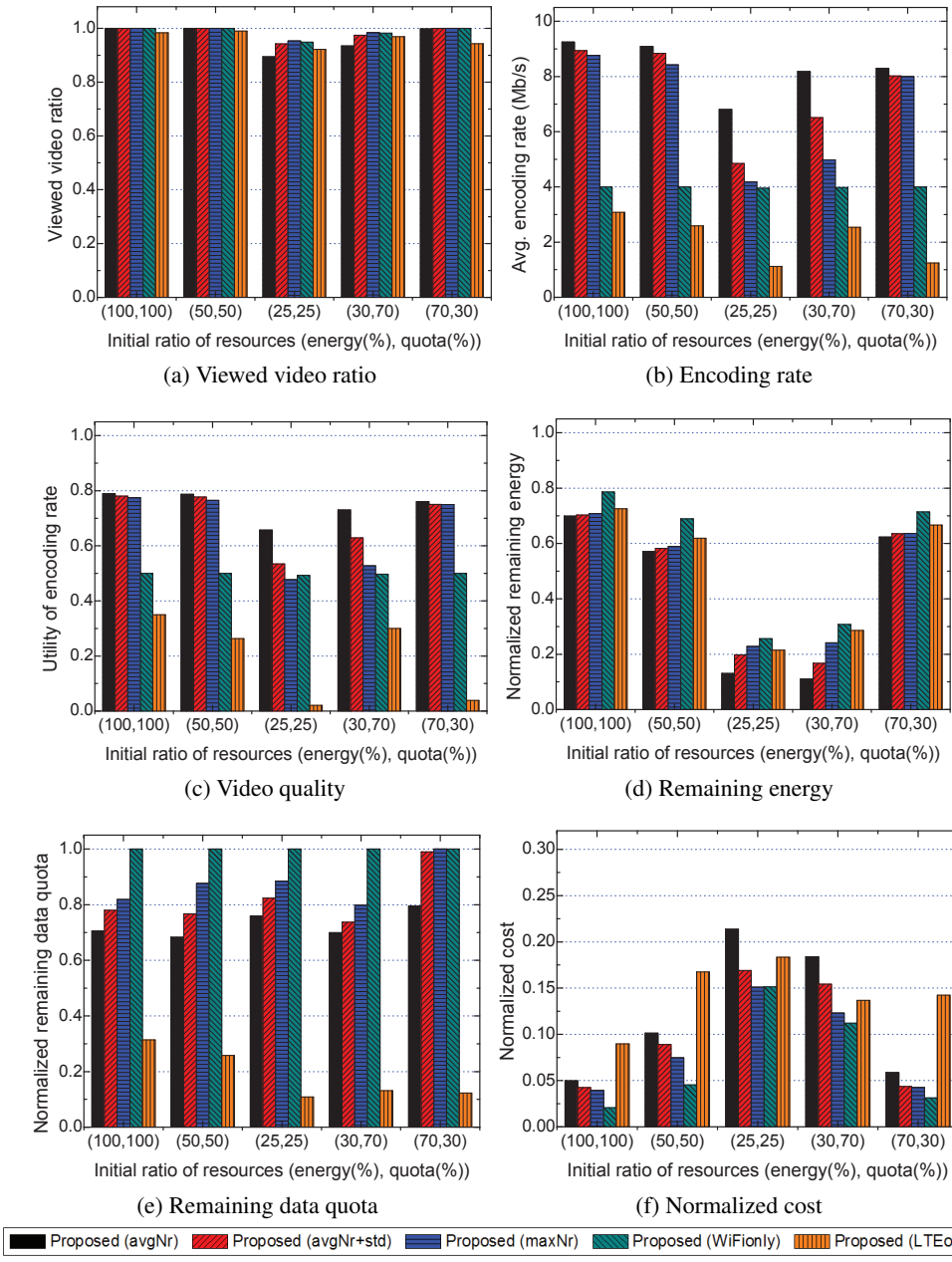
First of all, we compare several variations of the proposed scheme with various parameter setting. Notations represented in Fig. 3.12 are explained as follows:

- $avgN_r$: N_r is just substituted by the average number of video arrivals from the empirical data set.
- $avgN_r$+*std*: N_r is substituted by the sum of the average number of video arrivals and its standard deviation from the empirical data set. It would be slightly larger than the average, and hence the resources are used in a little more conservative way.
- $maxN_r$: N_r is substituted by the maximum number of video arrivals a day in the empirical log data. It make the algorithm operate in the most conservative way for the resources.
- *WiFi-only*: This scheme always use only the WiFi link when it is available. It consumes the least energy and data quota, however, the average video quality would be degraded due to bandwidth limitation of WiFi link.
- *LTE-only*: This scheme always use only the LTE link. It consumes too much energy and data quota, and hence the scheme would mainly focus on maximizing the total playback time at the cost of the video quality.

The simulation results are presented in Fig. 3.12. Each index in x-axis of the figure represents percentages of the initially given data quota and battery energy to the total data quota and battery energy. For example, (30, 70) means that the initially given

data quota and battery energy are 30% of the total data quota and 70% of the total battery energy, respectively. 3.12(a) represents a ratio of the watched video with given resources to the video request arrivals. One means that all the video requests are serviced. The $avgNr$ increases the average video quality as illustrated in Fig. 3.12(b) and (c) even though the watched video ratio is slightly reduced, which is kept still over 90% for all the cases. If a user wants more conservative operation, $avgN_r$+std or $maxN_r$ schemes can be adopted. They increase the watched video ratio at the cost of slight degradation of the video quality. in Fig. 3.12(d) and (e) represent normalized remaining resources after all the video are watched. one means that no resource is used at all, and zero means all the resources are depleted. When a measurement duration (a day) is finished, it is undesirable to remain too much resources, and therefore, the proposed algorithm utilize the resources as much as possible to enhance the video quality so far as the expected watched video ratio is not too low.

Furthermore, we compare the proposed scheme of $avgN_r$+*std* with the other static schemes with either LTE or WiFi path. The static scheme only chooses either the maximum encoding rate $R_s\{\max\}$ or the minimum encoding rate $R_s\{\min\}$. The simulation results are represented in Fig. 3.13. The WiFi-only and/or $R_s\{\min\}$ achieve the better viewed video ratio, covering over-90% of video arrivals, due to the low power consumption and less usage of paid data. However, the video quality is limited because they utilize the data and energy resoureces too conservatively. On the other hand, even though, $R_s\{\max\}$ with the LTE-only shows the best video quality among the static schemes, the viewed video ratio is too low especially for the case for lack of resources.



(a) Viewed video ratio

(b) Encoding rate

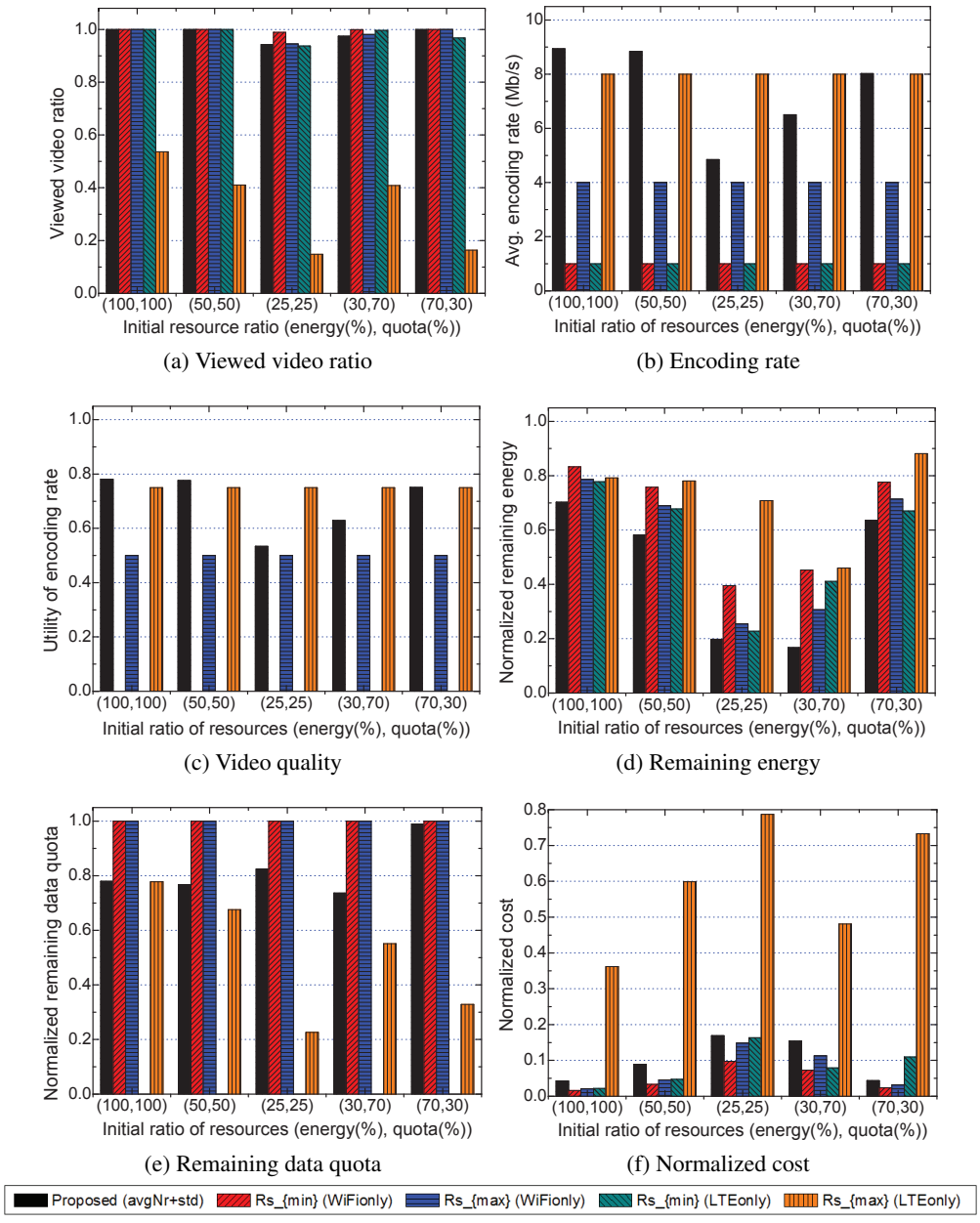
(c) Video quality

(d) Remaining energy

(e) Remaining data quota

(f) Normalized cost

Figure 3.12: Comparison between the proposed schemes.



(a) Viewed video ratio

(b) Encoding rate

(c) Video quality

(d) Remaining energy

(e) Remaining data quota

(f) Normalized cost

Figure 3.13: Comparison between the proposed scheme and static schemes.

3.6 Summary

In this chapter, we consider both data quota and battery energy issues as well as the video quality for HTTP-based video streaming service in a smartphone. We analyze the data usage and energy consumption with a given video encoding rate , especially including the expected data and energy waste when a user stops watching video in the middle of the playback. At first, we propose the chunk download cycle (CDC) adaptation scheme by formulating a normalized cost as a function of the remaining data quota and battery energy in order to maximize the total playback time with the remaining resources. Then, we propose a complete algorithm to find out the optimal operation parameter set of network selection, video encoding rate, and CDC that balances the video quality and the playback time. The simulation results show that the proposed scheme effectively balances the video quality and total playback time according to the remaining data quota and battery energy status.

Chapter 4

Standby Mode

4.1 Introduction

Over 50% of total energy is consumed while a smartphone is not in use with screen off and waits for user's interaction, namely standby mode, since the smartphone spends most of the time in the standby mode, i.e., 85% of a day on average [4]. A large portion of battery drain in the standby mode occurs due mainly to the background traffic generated by applications running on the smartphone [41]. The applications that especially need interactions with the servers through the Internet, periodically generate light load traffic via a data network interface of the smartphone for several purposes: 1) checking new messages or updates, e.g., e-mail, text messages of chatting applications, and weather information, and 2) maintaining TCP connections in the network address translation (NAT) network environment to receive the push data from the servers without connection overhead in the near future. The battery drain caused by background traffic is intensified as the number of applications using the Internet increases.

To alleviate the battery dissipation in the standby mode, previous efforts have tried

to reduce the energy consumption caused by background traffic by delaying packet transmission and traffic shaping techniques [42–44], or reducing 3G tail time [45]. In addition, a lot of applications, which prevent the background traffic by turning off the data network interface, are provided in application stores [46, 47]. Turning off the data network interface during the standby mode and reconnecting to the Internet only when the next usage starts is the simplest way to save the energy without any kernel level modification. However, it may rather consume more energy whenever the standby mode starts in the case that the smartphone usage too frequently arrives because turning off and on the data network interface cause additional energy overhead for the initial connection setup [31]. On the other hand, if the smartphone enters DSM too late, it would lose most of the time to save energy. Therefore, it is critical to determine when to turn off the data network interface to maximize the energy saving gain, which mainly depends on a smartphone usage pattern of each user.

To understand the user-specific smartphone usage pattern [48,49], we exploit timestamp logs of the start and end time of each standby mode gathered from multiple smartphone users. We define the idle duration as the time between the start and the end of a standby mode, and show that the distribution of the idle duration follows one of the heavy tailed distributions, i.e., a lognormal distribution [50], which is commonly adapted to the Internet traffic and communications such as TCP session duration and downloaded file size distribution [51, 52]. We propose a usage pattern-aware energy saving algorithm for smartphones by properly blocking the data network in the standby mode. The proposed algorithm consists of two phases: 1) *learning phase*, where a smartphone measures long-term average power consumption for specific operations such as non-DSM, DSM, and turnning ON/OFF the network interface and builds a usage pattern profile based on the timestamp logs, and 2) *deep sleep mode (DSM) operation phase*, where the smartphone waits for a specific threshold time at

the start of the standby mode and entering DSM turning off the data network interface. When the user starts to use the smartphone, the data network interface immediately turns on reconnecting to the data network [1].

The proposed algorithm adaptively operates according to the user-specific usage pattern profile to lengthen the battery lifetime in real-life environments. Furthermore, we enhance the algorithm to separately operate in the activity and inactivity time, i.e., typically separated by user's sleep time. Because the DSM necessarily causes additional delay of the Internet messages incoming during the standby mode, the enhanced algorithm makes the smartphone enters DSM conservatively in the activity time and aggressively in the inactivity time by applying the different DSM threshold time values. Through the smartphone usage trace-driven simulations, we validate that our algorithm successfully reduces the battery leakage in the standby mode and prevents entering DSM too frequently in the activity time.

The rest of this chapter is organized as follows. In Section 4.2, we present an power anatomy of a smartphone in the standby mode. and analyze several users' usage patterns in Section 4.3. Then, we proposed the DSM algorithm in Section 4.4. The performance evaluation of the proposed algorithms are presented with the simulation results in Section 4.5. Finally, we summarize and conclude this chapter in Section 4.6.

4.2 Standby Mode Power Anatomy of Smartphones

4.2.1 Low power mode operation

WiFi defines two power management modes, namely, *active mode (AM)* and *power save mode (PSM)*. The WiFi interface in *AM* always runs in the *awake state* in which it can transmit and receives the packets, while the WiFi interface in *PSM* toggles be-

[1]

tween the *awake state* and *doze state*, in which the interface is basically turned off. The AP informs the presence of the buffered packets via the traffic indication map (TIM) element in the beacons. The WiFi interface in *PSM* periodically wakes up, i.e., switching from *doze state* to *awake state*, for every delivery TIM (DTIM) period to receive beacons, and checks the TIM element in the beacon. If there exist buffered packets destined to the device, the WiFi interface sends a null data packet (NDP) in order to switch from *PSM* to *AM*, and then receives the buffered packets. The WiFi interface sets the inactivity timer which has a specific timeout value, called *PSM timeout*, and it resets the timer for every packet reception. When the inactivity timer expires, the WiFi interface switches back to *PSM* to save energy by sending an NDP to the AP.

Likewise, LTE has two radio resource control (RRC) states, namely, *RRC_connected* and *RRC_Idle*. Once an eNodeB (i.e., LTE base station) allocates resources to a user equipment (UE), the UE operates at *RRC_connected* state consuming high power. At *RRC_connected* state, the UE first stays in *continuous reception* mode and keeps monitoring physical downlink control channel (PDCCH). The UE starts *discontinuous reception (DRX) operation* and enters *short DRX* mode when *DRX inactivity timer* expires. The UE remains in *short DRX* until *short DRX cycle timer* expires and goes to *long DRX* when it expires. If the UE, which operates in *long DRX*, does not receive any packet until *RRC inactivity timer* expires, it enters *RRC_idle* state by releasing the allocated resources, and saves the energy consumption. When the UE operates in either *short DRX* or *long DRX*, it goes back to *continuous reception* if it receives/transmits a packet. During *RRC_idle* state, the UE sleeps for most of time and periodically wakes up to receive paging messages from eNodeB. Accordingly, it is awake for a few milliseconds every *RRC_idle DRX cycle* period.

We refer to the WiFi *PSM* and LTE *RRC_idle* as the the low power modes of WiFi and LTE, respectively. The energy consumption of the network interfaces in low

power mode is determined by the DTIM period for WiFi and *RRC_idle DRX cycle* period for LTE. The DTIM period is the beacon period multiplied by a DTIM interval and *RRC_idle DRX cycle* is one of 320, 640, 1280, and 2560 ms.

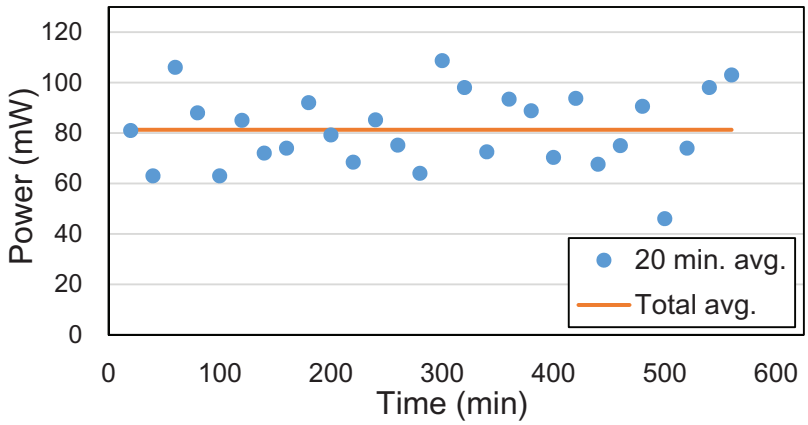
Whenever the last packet reception is completed, the network interface waits for *PSM timeout* in case of WiFi or *RRC inactivity timer timeout* in case of LTE, and then enters the low power mode. The tail time of WiFi and that of LTE are defined as the time duration of *PSM timeout* and that of *RRC inactivity timer timeout*, respectively. The tail power is the power consumption for the tail time, i.e., channel sensing power in *awake state* in case of WiFi, or *long DRX* state power in case of LTE. The average power consumption for WiFi and LTE in the low power mode is about 2.5 mW and 5.6 mW, respectively.

4.2.2 Power consumption for background traffic

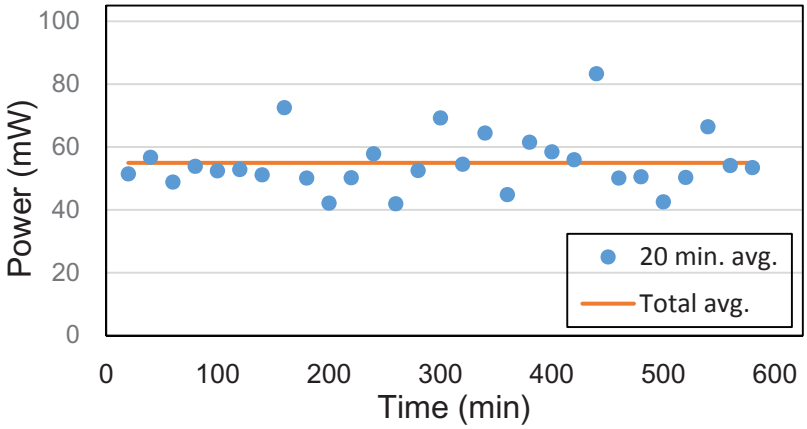
Even though a user turns off the screen and stop using a smartphone, i.e., the smartphone is inactive, the network interface(s) are intermittently activated to serve background traffic generated by network applications during this inactive period. Network applications generate background traffic for 1) *keep-alive messaging*, i.e., the applications periodically send keep-alive messages to their servers so that the servers can deliver new data to the smartphone in a NAT environment. and 2) *application data update*, i.e., such network applications as e-mail, weather forecast, and news applications periodically check and receive updated new data from the servers. The amount of background traffic may increase as a user installs more applications on her/his smartphone, and background traffic frequently incurs the tail power of the network interfaces and severely degrades the battery lifetime.

To empirically investigate how much energy is consumed to serve background traffic, we measure the power consumption of the smartphone in the standby mode

using Monsoon power monitor [53] while network applications are running in the background.[2]



(a) LTE standby mode power consumption



(b) WiFi standby mode power consumption

Figure 4.1: Average power consumption of smartphone on the standby mode.

[2]All the applications except for those running as background services [54] are paused while the display turns off and no *partial_wakelock* [55] is hold. The paused applications are different from terminated applications and they are still alive in Android kernel level and can be resumed by triggers, e.g., users' interactions and Android *alarm* services [56].

Fig. 4.1 shows the measured standby mode power consumption for the case that the LTE and WiFi interfaces are used for 10 hours. We average the power consumption, which is measured every 20 minutes (represented as each "dots" in Fig. 4.1), and average power for 20 minutes range from 60 mW to 120 mW for using LTE interface and from 40 mW to 80 mW for using WiFi interface. Since the longer tail time and higher power consumption of the LTE interface compared to the WiFi interface, the standby mode power consumption of using the LTE interface is obviously higher than that of using the WiFi interface.

The background power consumption differs according to the smartphones status, i.e., the number of network applications installed on the smartphone and the amount of background traffic generated by the network applications, and the channel status of the network with which the smartphone currently associated. The impact of the smartphone status to the background power consumption is inferred by analyzing the background traffic captured with *tcpdump* during the standby mode. One of the possible training methods to analyze the background power consumption and the amount of traffic is measuring the remaining battery level[3] and capturing the traffic during the users do not interact with their smartphone for a long time, e.g., inactivity time. The impact of the network status is experimentally investigated and we discuss it in the following section.

4.2.3 WiFi MAC overhead issue

The smartphone is connected to the Internet via either WiFi and LTE, and none of them is activated in a special case such as *Airplane mode*. LTE has wide coverage and supports a sophisticated handover procedure so that a user connects to the LTE

[3]The battery interface provides the remaining battery level which has the integer values from 0 to 100, and it can be displayed on the notification bar of the screen.

network almost everywhere while WiFi has short coverage, e.g., less than 10 meters for good signal strength. In addition, the smartphone may not always connect to WiFi in practice, and good signal strength is not guaranteed even though WiFi connection is available. Each network has its own merit i.e., LTE has the advantage in terms of the availability but WiFi is more energy-efficient than LTE. Therefore, if WiFi is always available with good signal strength, the smartphones would better connect to the WiFi to save the energy consumption, but practically it is not valid in real environments.

Due to the fluctuation of WiFi signal and user's mobility, the WiFi signal strength varies and affects the WiFi interface power consumption. With low RSS, packets are transmitted with low-order MCS due to the rate adaptation, thus resulting in increased tx/rx airtime ratios, and retransmitted packets incur additional energy consumption between consecutive successfully received data packets. Furthermore, much more MAC frame overheads arise when the RSS is below -80 dBm, e.g., active scanning for searching better access points (APs) and heavy retransmissions of NDPs due to the receive sensitivity difference between the antennas of APs and those of the smartphones.

We let the smartphone connect to WiFi and measure the power consumption of the standby mode by turning off all the background applications to measure only the WiFi power consumption according to WiFi signal strength. We place the smartphone in various locations to obtain the diverse received signal strength of WiFi from -42 dBm to -80 dBm. We capture the packets and classify them into 5 categories, i.e., probe request and response for active scanning ("Probe req/resp"), NDP transmissions ("NDP init"), NDP retransmissions ("NDP retry"), other data packets ("Non-NDP init"), and retransmissions of "Non-NDP init" ("Non-NDP retry"). The measured power consumption and classification of WiFi traffic are presented in Fig. 4.2 according to RSSs. The average standby mode power consumption abruptly increases from -74 dBm RSS with higher power variations while the average power smoothly increases from

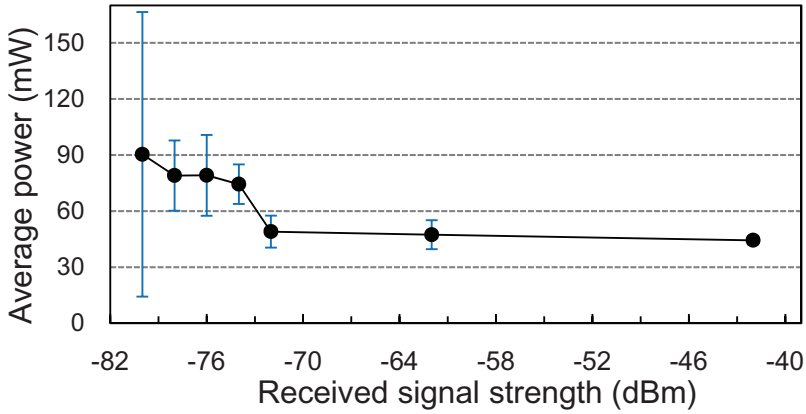
-42 dBm to -72 dBm with fewer power variations as presented in Fig. 4.2(a). The increased power from -74 dBm RSS correspond to the increased packet rates of classified traffic in Fig .4.2 (b). The RSS lower than -74 dBm causes more NDP transmissions/retransmissions and the incidences of active scanning grow. When the RSS exceeds -80 dBm, NDP retransmissions are exacerbated. In this RSS region, the WiFi connections are occasionally lost and the smartphones associated with the APs having better signal strength by active scanning.

From these experiments, we quantitatively check that the amount of WiFi MAC traffic is substantial at bad received signal strength of WiFi, and hence, the increased energy consumption of WiFi interface reduces the battery lifetime of the smartphone. Therefore, connecting to WiFi is not always energy-efficient so that we should select a data network for energy-efficiency with consideration of the network status such as RSS.

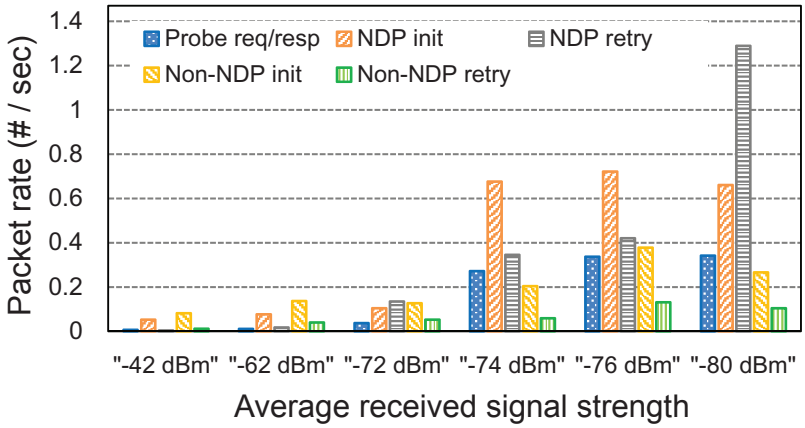
4.3 Usage Log-based Idle Duration Analysis

4.3.1 User-specific daily distribution of idle duration

We implement an Android application to collect the user's usage information, i.e., screen on/off, the information of an associated network (LTE or WiFi), battery information, and running applications with a timestamp for each event. As previously mentioned, the idle duration is defined as the time between the start and the end of a standby mode. The start of a standby mode is defined as the time when the screen is turned off and all the background services [54], e.g., a music playing application, are stopped while the end of the standby mode is defined as the time when the screen is turned on and touched by the user. We installed the logging application on several user's smartphone and gathered the log data for a month.



(a) Average power consumption in various RSS



(b) WiFi MAC overhead frame rates in various RSS

Figure 4.2: WiFi MAC overhead in various received signal strength (RSS) levels.

The Fig. 4.3 shows a cumulative distribution function (CDF) of the idle duration of one user's log data for 10 days. The x-axis represents the idle duration (second) in log-scale, and the blue line describes the empirical data of a real user. The red line represents the lognormal function fitting to the empirical data, of which the probability density function is expressed by $f(x) = \frac{1}{x\sqrt{2\pi}\sigma} e^{-\frac{(\ln x - \mu)^2}{2\sigma^2}}$. The mean μ and variance σ for the lognormal fitting curve are presented in the figure. In summary, the distribution of a user's idle duration is well fitted to the lognormal distribution which is one of the heavy-tailed distributions.

4.3.2 All-day distribution

We gather four users' smartphone usage log data for a month and present the idle duration distributions with the corresponding lognormal fitting curves in Fig. 4.4(a). From the figure, we find that the log-normal fitting can still be applied to the other users idle duration distributions. Furthermore, each user has his/her own eigen-distribution of the idle duration that is generally caused by the different usage pattern from one another.

On the other hand, one user's daily distribution of the idle duration has the self-similarity due to the user's habitual usage pattern. Fig. 4.4(b) represents one user's daily distribution of the idle duration for 5 days, and we can see that each distribution does not get much beyond the total average distribution.

4.3.3 Activity/inactivity time separation

Four users' daily idle duration patterns are presented in Fig. 4.5. Each dot represents an average of the idle durations observed in the 30-minute time bin of the corresponding time of day. If a long idle duration lasts for several time bins, each time bin encountered with the idle duration individually counts the whole duration for its average. We can

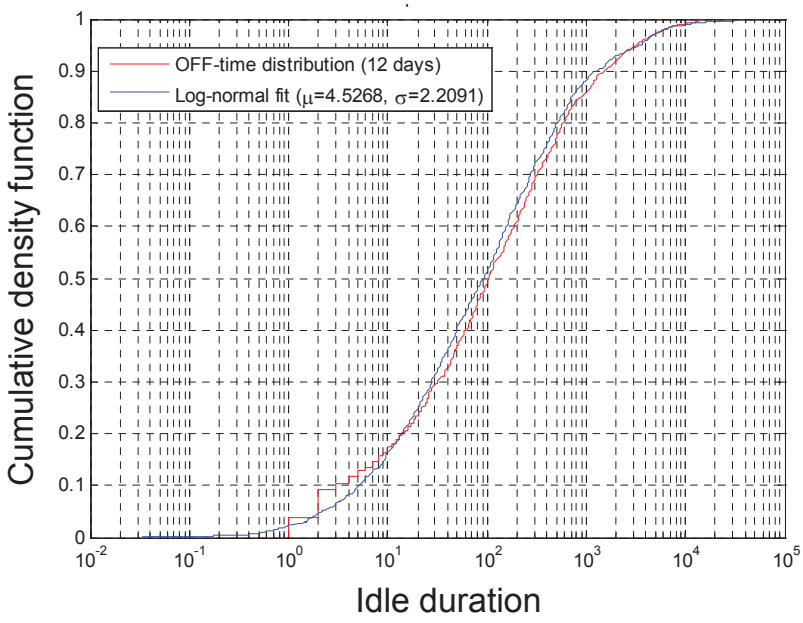
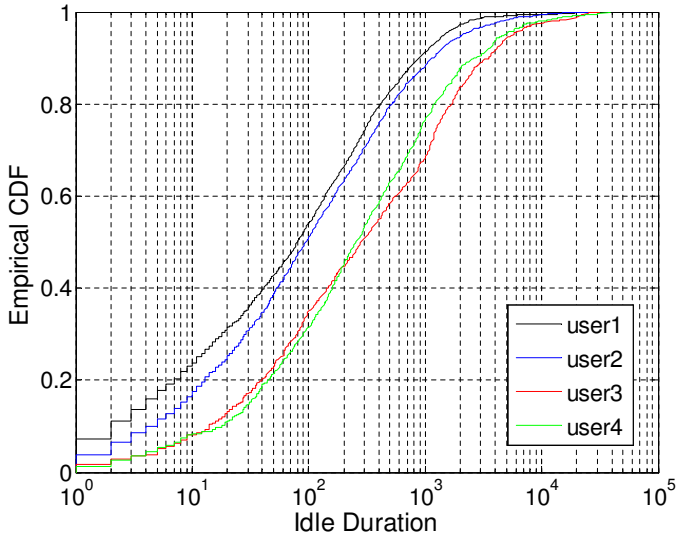
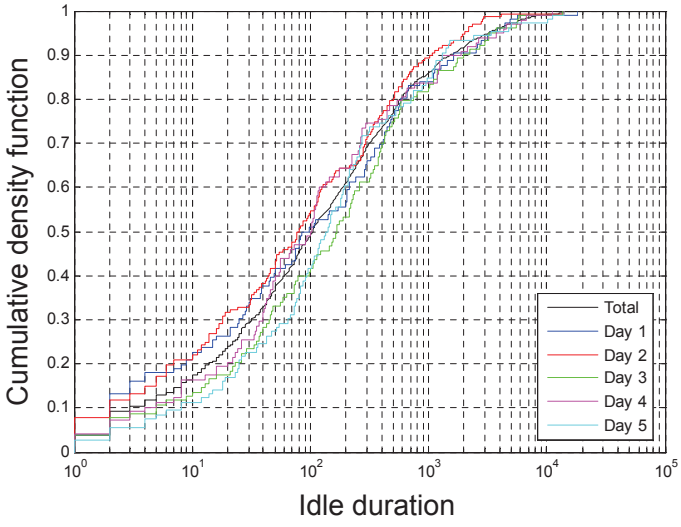


Figure 4.3: CDF of idle duration and lognormal-fit curve.

see that the daily patterns tend to divided into the two parts, i.e., long and short idle durations, which is due mainly to each user's regular sleeping time. We separate the two parts by a specific threshold time, e.g., the minimum idle duration in Fig. 4.5 added by 1 hour, and define the long and short idle duration regions as the *activity time* and *inactivity time*, respectively. With these two regions based on the user specific usage pattern logs, we can apply the energy saving algorithm separately to get preferable effects. For example, the energy saving algorithm may only be applied during the inactivity time while the smartphone is used in the performance oriented way during the activity time.



(a) Distribution of idle duration for different days



(b) Distribution of idle duration for different 4 users

Figure 4.4: Comparison of idle duration distributions.

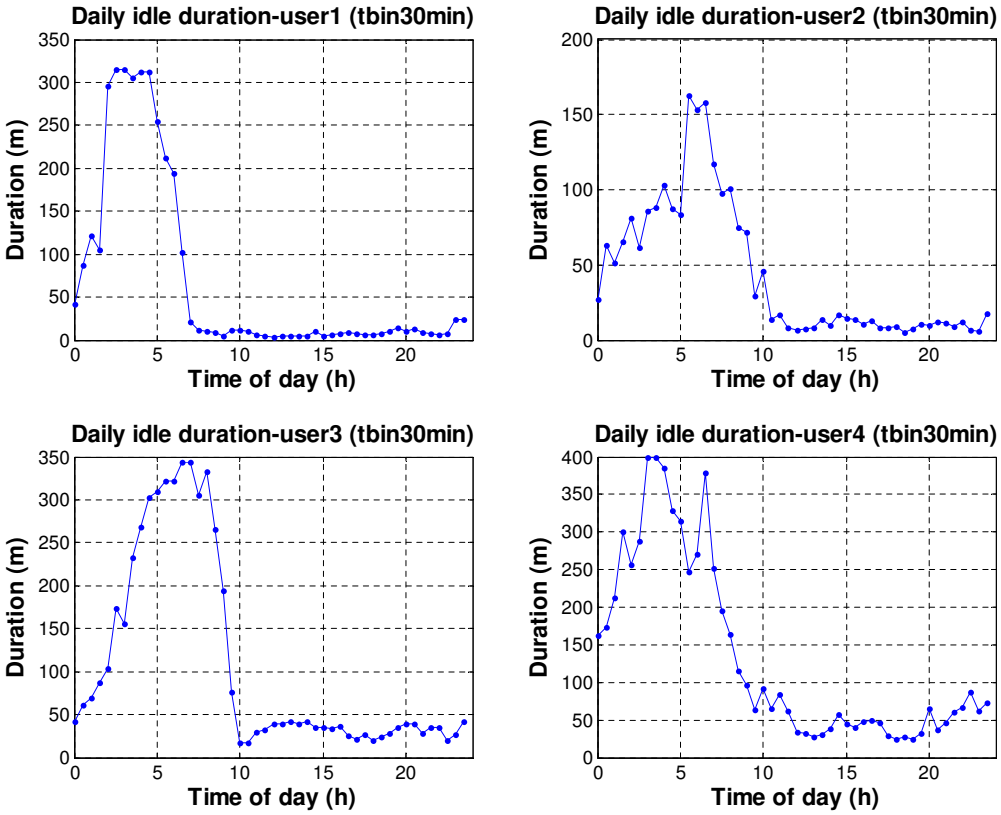


Figure 4.5: Daily idle time distributions of four users.

4.4 Proposed Algorithm

We propose a user behavior-based energy saving algorithm for LTE/WiFi-equipped smartphone in this section. The proposed algorithm has a learning phase and DSM operation.

4.4.1 Learning phase

In the learning phase, the usage analyzer gathers all the time instances of the start and end of the standby mode and formulates the empirical CDF of the idle duration as shown in Fig. 4.3. The energy estimator estimates the energy consumption during the standby mode in the learning phase for the long-term average power consumption of the standby mode as shown in Fig. 4.1. The standby mode power consumption when connected to either WiFi or LTE is individually estimated.

All the previous standby mode time can be included in the learning phase, and the idle duration distribution and long-term average power consumption are updated based on the cumulative log data in the standby mode. Using the usage analysis data, the proposed DSM algorithm is operated as follows.

4.4.2 Deep Sleep Mode (DSM) operation

Energy efficient network connection in standby mode

When a standby mode starts and both of the LTE and WiFi are available, the smartphone chooses connects to the network interface that consumes less energy in the standby mode. Generally, the average power consumption of LTE in the standby mode is higher than that of WiFi due to much higher energy overhead of the background traffic with LTE as shown in Fig. 4.1. However, if the RSS of WiFi is lower than -75 dBm, the extra WiFi power overhead occurs, and thus the standby mode power consumption

of WiFi may exceed that of LTE. Therefore, the smartphone estimates the standby mode power consumption both of LTE and WiFi in the learning phase monitoring the WiFi RSS, and then chooses one of them consuming less average power in the stanby mode.

Data network ON-OFF

In the LTE system, the IP Multimedia Subsystem (IMS) for phone services, e.g., voice over LTE (VoLTE) and short message services (SMS), and the Internet for data services are provided by the different Packet Data Networks (PDNs). A smartphone connects to each PDN via the individual bearer with the corresponding Access Point Name (APN) exploiting a virtual network interface. Fig. 4.6 shows the system structure for the IMS and the Internet services, where the corresponding virtual LTE interfaces are "LTE IP1" and "LTE IP2". If a user connects to the Internet via WiFi, "WiFi IP1" is activated while "LTE IP2" is deactivated.

In this system, to save more energy especially caused by background traffic in the standby mode, we consider turning off the data network interface for the Internet, i.e., either "LTE IP2" or "WiFi IP1" to prevent the background traffic generation. Therefore, the phone services such as VoLTE and SMS are still available through "LTE IP1" even if the Internet data network is disconnected.

DSM algorithm process

Heavy tailed distribution of the idle duration implies that the probability that standby mode time duration tends to be further longer as the time lasts longer. Based on this characteristic, we propose a DSM algorithm that turns off data network interface with the corresponding background applications when a long idle duration is detected.

A process of the DSM algorithm is described in Fig. 4.7 with two options. For both

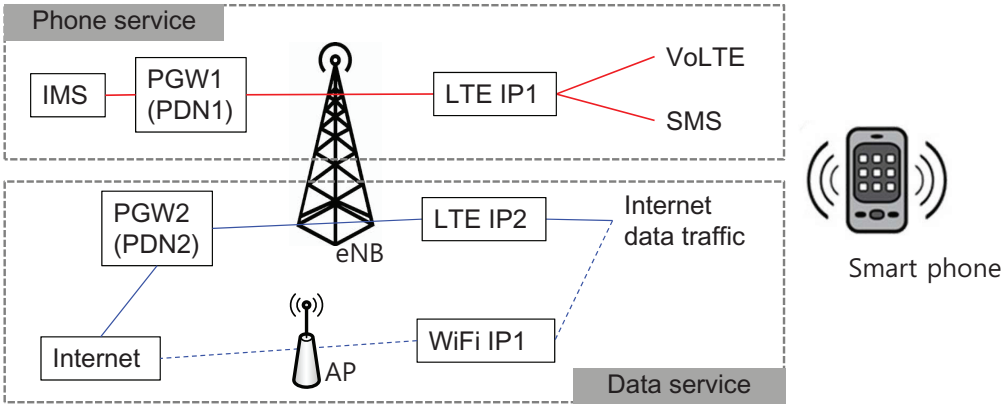


Figure 4.6: Smartphone service network.

of the options, the data network interface is turned off to save energy if an idle duration continues above a threshold, denoted by T_{dsm}^{th}. For Option 1 as shown in Fig. 4.7(a), the data network is reconnected only when the smartphone usage is restarted. In the case of Option 2 as described in Fig. 4.7(b), the data network interface periodically turns on for a while, e.g., 10 s, to check update messages of background applications, where the period of turning on the data network interface T_{onoff} can be set by a user. After that, the data network is disconnected again turning off the network interface to save the energy. When a user turns on and touches the screen, the data network connection is immediately set up and the DSM algorithm process is terminated.

For efficient DSM operation, it is critical to set a proper threshold time T_{dsm}^{th} to enter the DSM. If T_{dsm}^{th} is set too short, the smartphone unnecessarily enters DSM too many times causing the energy overhead for turning on and off the data network interface. On the contrary, if T_{dsm}^{th} is set too long, the smartphone seldom enters DSM and loses most opportunities to save energy. We exploit user-specific distribution of the idle duration, fitted to the lognormal distribution to find the optimal T_{dsm}^{th}. For this purpose, we derive the expected power consumption for the DSM process as a function

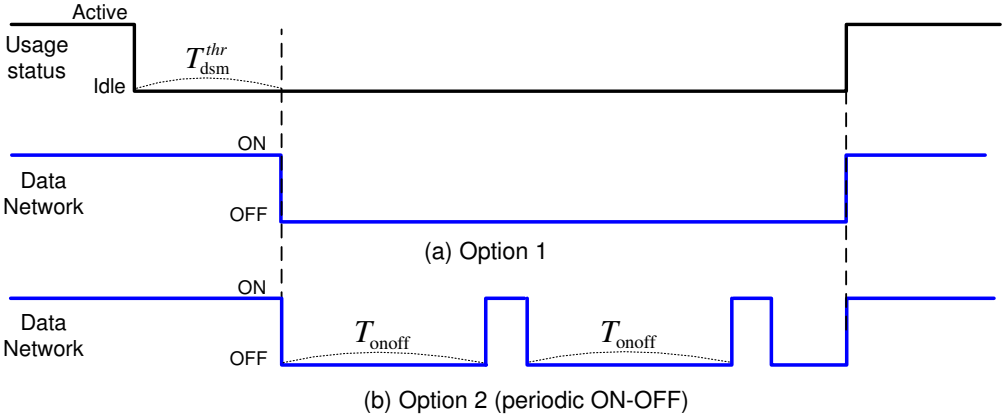


Figure 4.7: Deep sleep mode process.

of T_{dsm}^{th} and T_{onoff} as follows:

$$\bar{P}(T_{\text{dsm}}^{th}, T_{\text{onoff}}) = \Bigg[\int_0^{T_{\text{dsm}}^{th}} P_{\text{static}} \cdot t f_{\text{idle}}(t) dt + \int_{T_{\text{dsm}}^{th}}^{T_{\text{tot}}} f_{\text{idle}}(t) \Big\{ P_{\text{static}} T_{\text{dsm}}^{th} + N_{\text{onoff}}(t) E_{\text{onoff}} + P_{\text{lpm}}(t - T_{\text{dsm}}^{th} - D_{\text{on}}) \Big\} dt \Bigg] \Big/ \int_0^{T_{\text{tot}}} t f_{\text{idle}}(t) dt, \tag{4.1}$$

where t is the elapsed time since the standby mode starts, and $f_{\text{idle}}(t)$ is the lognormal probability density function of the idle duration. P_{static} is the power consumption when the data network is statically connected, N_{onoff} is the total number of the data network ON-OFF with given T_{onoff} for t, E_{onoff} is the energy overhead to turn on and off the data network interface. D_{on} is the ON-duration to check the update message, P_{lpm} is the average power consumption for the low power operation as presented in Section 4.2.1. If we apply the probability mass function of the idle duration from the empirical data, (4.1) is re-written as follows:

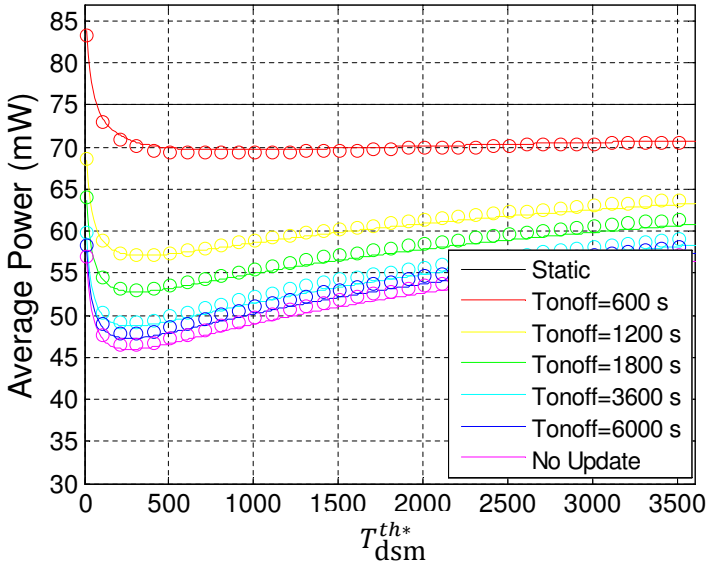
$$\bar{P}(T_{\text{dsm}}^{th}, T_{\text{onoff}}) = \Bigg[\sum_{\tau=0}^{T_{\text{dsm}}^{th}} P_{\text{static}} \cdot \tau p_{\text{idle}}(\tau) + \sum_{\tau=T_{\text{dsm}}^{th}}^{T_{\text{tot}}} p_{\text{idle}}(\tau) \Big\{ P_{\text{static}} T_{\text{dsm}}^{th} + N_{\text{onoff}}(\tau) E_{\text{onoff}} + P_{\text{dsm}}^{\text{idle}}(\tau - T_{\text{dsm}}^{th} - D_{\text{on}}) \Big\} \Bigg] \Big/ \sum_{\tau=0}^{T_{\text{tot}}} \tau p_{\text{idle}}(\tau). \tag{4.2}$$

Fig. 4.8(a) shows the numerical result of the average power consumption of the DSM process according to T_{dsm}^{th} and T_{onoff} with User 1's trace. The solid lines is from the numerical simulation with the lognormal fitting parameters of the idle distribution while the circles represent the real trace-driven simulation. As previously discussed, we can see the quasi-convexity of the average power consumption according to T_{dsm}^{th}, and consequently it has the optimal value T_{dsm}^{th*} for each T_{onoff} as represented in Fig. 4.8(b).

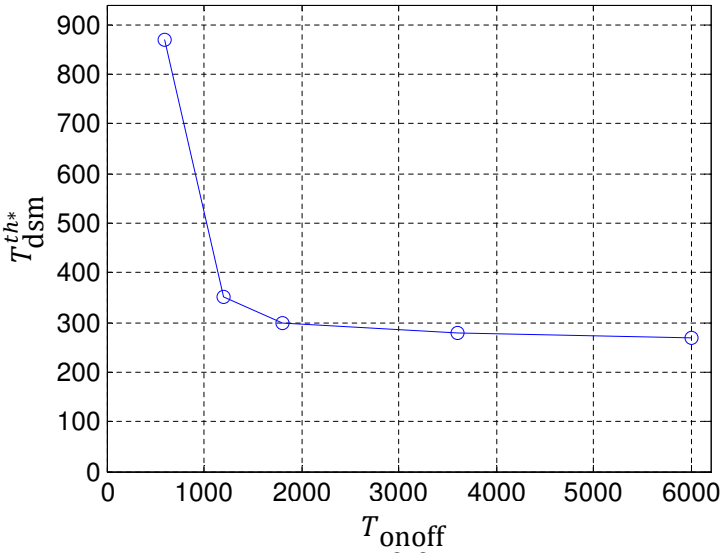
We also confirm that the maximum energy saving gain is reduced as T_{onoff} decreases. Therefore, the DSM algorithm should avoid entering DSM in the case when the estimated power consumption with T_{onoff} set by the user is greater than the power consumption without DSM..

DSM algorithm extension

Even though DSM achieves the energy saving gain, entering DSM during activity time increases the standby mode power consumption due to frequent occurrences of the energy overhead for turning on and off the data network interface. To prevent the smartphones from entering DSM during activity time, we classify a day, i.e., 24 hours, into the activity time and inactivity time according to user's daily idle duration pattern as described in Section 4.3.3. For the time classified as the inactivity time, a user may sleep, enjoy a sport or studies leaving her/his smartphone for a long time, and hence,



(a) T_{dsm} to average power



(b) T_{onoff} to the optimal T_{dsm}^{th*}

Figure 4.8: Numerical result of DSM process.

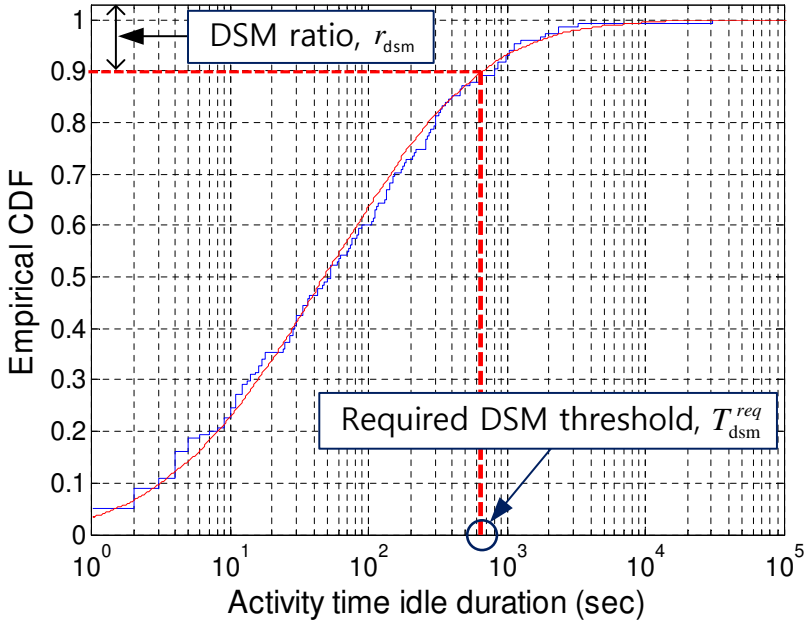


Figure 4.9: Required DSM threshold corresponding to the DSM ratio from user's activity time idle duration CDF.

the smartphone saves the energy consumption by entering DSM after the optimal T_{dsm}^{th*} obtained by the DSM algorithm. On the other hand, during the activity time, the user can set a limit for the smartphone to enter DSM with DSM ratio r_{dsm}, defined as the ratio of the number of DSM entrances to the number of all the standby mode occurrences. Then, we need another DSM threshold, denoted by T_{dsm}^{req}, required for the activity time to restrict the DSM ratio. If the CDF of the activity time idle duration x as shown in Fig. 4.9 is denoted by $F(x)$, we set T_{dsm}^{req} with the probabilistic approach, i.e., $T_{\text{dsm}}^{req} = F^{-1}(1 - r_{\textbf{dsm}})$.

4.5 Performance Evaluation

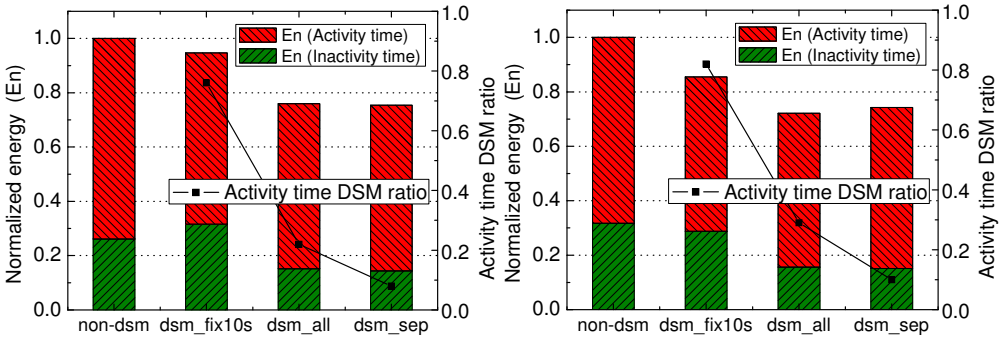
4.5.1 Performance comparison

To evaluate the performance of the DSM algorithm, we consider two kinds of the performance metric. The first is how much the energy consumption during the standby mode is saved, and the second is how well the proposed scheme prevents the smartphone from entering the DSM state during the activity time. The DSM algorithm needs to necessarily consider undesirably disconnecting data network connection when a user frequently use the smartphone during the activity time. Therefore, the proposed scheme adaptively set the T_{dsm}^{th} to enter DSM aggressively in the inactivity time and conservatively in the activity time based on the usage pattern analysis. We simulate the energy saving ratio during the standby mode for several days, and furthermore, we evaluate how well the adaptive setting of T_{dsm}^{th} adapts to user's daily usage pattern.

Fig. 4.10 represents performance comparisons among four different schemes: 1) *non-dsm* without DSM, 2) *dsm_fix10s* that enters DSM after 10 s from the start of the stanby mode, 3) *dsm_all* that always uses the only T_{dsm}^{th}, and 4) *dsm_sep* that exploits separate DSM threshold time for the activity time and inactivity time, where we set the DSM ratio in the activity time to 0.1. Bars represent the energy consumption normalized by the energy consumption of *non-dsm* in the standby mode for 20 days, and each black dot represents the ratio of the number of the DSM occurrences to the total number of standby mode arrivals during the activity time. Fig. 4.11 shows the empirical

Table 4.1: Simulation setup

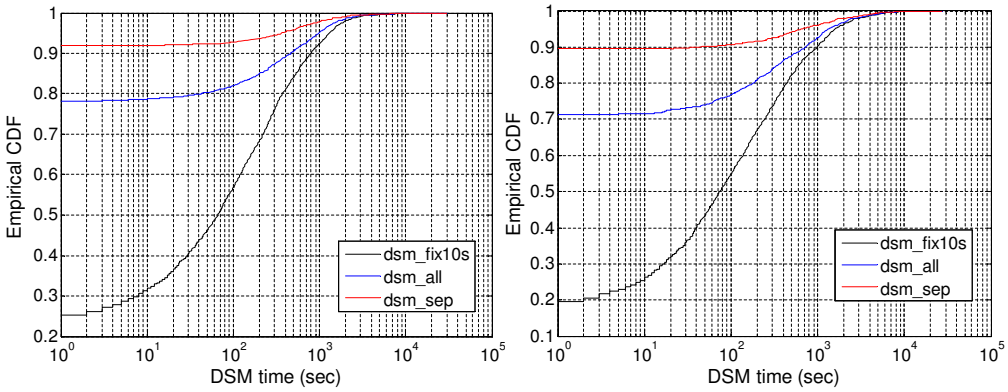
P_{static}	75 mW
P_{idle}	30 mW
E_{onoff}	20 Joule
T_{onoff}	{10, 20, 30, 60, 100, None } min.
D_{on}	10 sec.



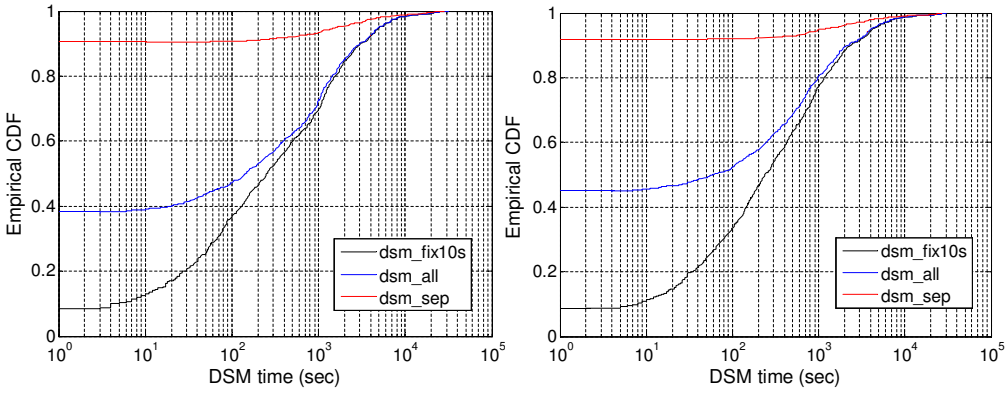
(a) User 1 (b) User 2

(c) User 3 (d) User 4

Figure 4.10: DSM performance comparison.



(a) User 1 (b) User 2



(c) User 3 (d) User 4

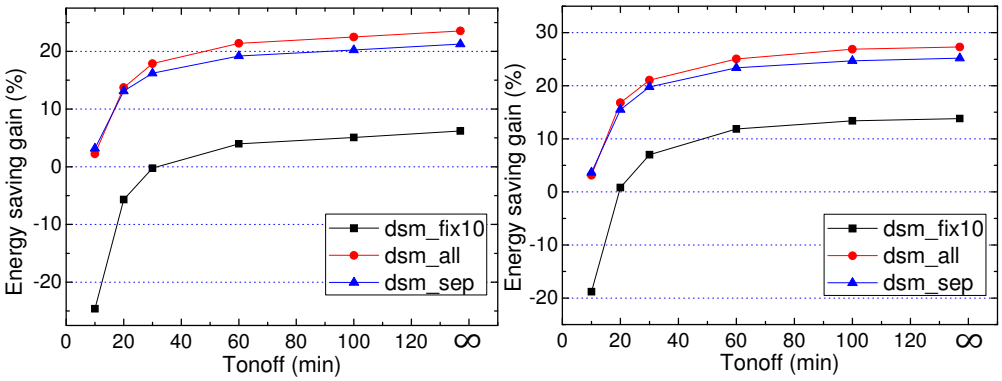
Figure 4.11: DSM time distribution.

CDF of the DSM time distributions for *dsm_fix10s*, *dsm_all*, and *dsm_sep*, where the DSM time is defined as the time that the smartphone stays in DSM for a stnadby mode time, e.g., if the smartphone never enters DSM for a standby mode time the DSM time is zero for the standby mode time.

From the result, *dsm_all* always outperforms the other schemes, and *dsm_sep* operates as intended reducing the activity time DSM ratio to around 0.1 even though the normalized energy consumption slightly increases than *dsm_all*. On the other hand, *dsm_fix10s* only performs good for specific users, and therefore cannot be commonly used for all cases. Furthermore, it enters DSM for over 80% of the standby mode arrivals, and the user would experience unavailability of the data network during the standby mode too much.

4.5.2 Effect of T_{onoff}

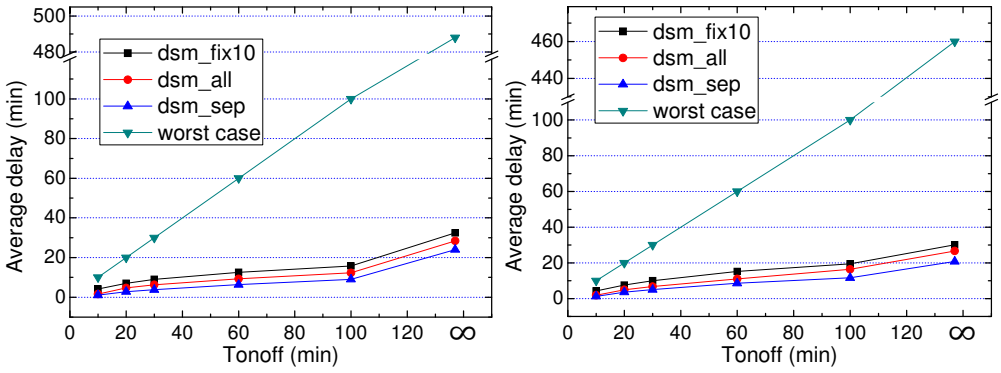
To compensate the incoming data delay due to DSM, the periodic ON-OFF of the data network according to user-set value T_{onoff} can be conducted during the DSM. Then, the maximum delay can be limited by T_{onoff} at the cost of loss of the energy saving gain, which is defined as the ratio of the saved energy to the enrgy consumption with *non-dsm*. In Fig. 4.12, user-specific energy saving gains according to various $T_{\text{onoff}} \in \{10, 20, 30, 60, 100, \infty\}$ are presented for the four users. ∞ means non-periodic ON-OFF. In addition, the corresponding delay for the worst and average cases is represented in Fig. 4.13. From the result we can observe that too short ON-OFF period sharply decreases the energy saving gain and the gain can be negative in some cases such as *dsm_fix10s* with $T_{\text{onoff}} = 10min.$ for User 1 and 2 even if that reduces the delay due to DSM. On the other hand, some setting, e.g., $T_{\text{onoff}} = 30min.$ can achieves much less delay without much degradation of the energy saving gain compared to the non-periodic ON-OFF. Therefore, we can conclude that the proposed



(a) User 1 (b) User 2

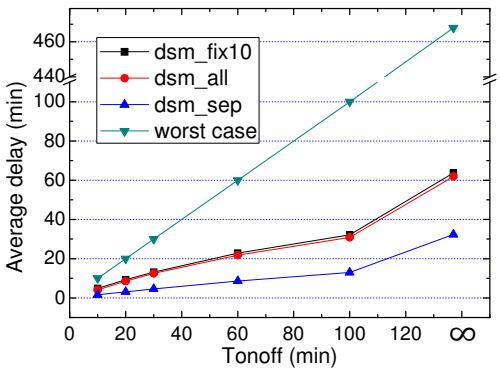
(c) User 3 (d) User 4

Figure 4.12: Standby mode energy saving gain vs. T_{onoff}.

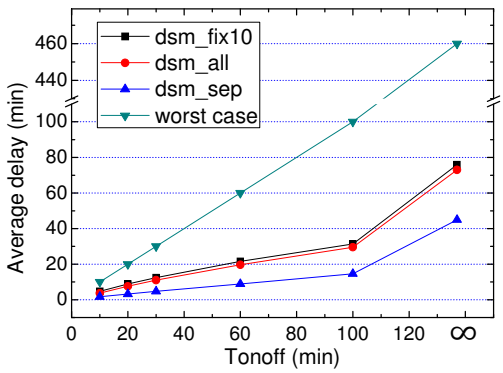


(a) User 1

(b) User 2



(c) User 3



(d) User 4

Figure 4.13: Standby mode additional delay vs. T_{onoff}.

DSM algorithm can properly operates based on the user preference setting reflecting the trade-off relationship between the energy saving gain and incoming data delay by usage pattern-aware estimation of the standby mode power consumption.

4.6 Summary

In this chapter, we present an anatomy of smartphone power consumption especially in the standby mode or light traffic mode. Based on the power analysis we propose algorithms for power saving operation in the smartphone. First, we propose the LTM algorithm to select the best network connection in the light traffic load state between LTE and WiFi according to packet arrival rate and WiFi channel condtion. Second, we propose the DSM algorithm that periodically turns on and off the data network interface to save more energy during the standby mode. It monitors the standby mode duration and triggers the DSM process when the duration exceeds a threshold time. The threshold time is obtained based on analysis of each user's smartphone usage profile and adaptively set to maximize the expected energy saving. We exploit real user's smartphone usage log data to practically evaluate the performance of the proposed scheme, and the results show that our proposed scheme effectively adapts to various environments and saves energy on average 20%.

Chapter 5

Conclusion

5.1 Concluding Remarks

In this dissertation, we propose energy, service charge, and performance aware, namely *ESPA*, network usage optimization algorithms for multi-RAT smartphones. User's satisfaction with a service is related not only to the performance, i.e., the service quality, but also to the battery energy consumption and data quota usage because they have generally limitation on use due to their limited capacity such as the battery capacity and data plan.Therefore, we model multi-attribute based utility/cost functions with regard to energy and data quota as well as the performance for specific applications, i.e., file transfer services and video streaming. Each utility/cost function is represented by a function of the current status of the remaining battery energy and data quota, and therefore the proposed algorithm can selectively save either the energy or data quota according to the remaining resources. When both of the battery energy and data quota are currently sufficient, the algorithm focuses more on maximizing the performance. However, as any of the available resources is depleting, the weight of that term in the utility/cost function increases, and the focus of the algorithm moves to save the

depleting resource.

On the other hand, we also propose the deep sleep mode algorithm. Even though the light usage applications such as web browsing and text messaging consume less energy and data compared to the large file transfer or video streaming, we show that the effect of the proposed algorithm can be considerable because the proportion of the standby mode time is generally dominant among user's daily usage time. We also present that the deep sleep mode algorithm saves the energy consumed by the background traffic during the standby mode.

More into detail, the research contributions of each chapter in the dissertation are summarized as follows.

In Chapter 2, we propose a multiple network interfaces activation scheme especially for file transfer services in smartphones. We generalize the multiple network interface activation problem for n RAT-equipped smartphones. We show that the generalized version is the joint combinatorial and piecewise linear minimization problem that can be still properly optimized. We propose a heuristic algorithm to find the optimal RAT set and corresponding segment allocation with low computational complexity. Even though the number of possible RAT combinations exponentially increases as the number of available RATs increases, the proposed linear search algorithm reduces the size of the search space from exponential to linear. We also propose a dynamic update algorithm that adapts the selected RAT set and segment allocation during the transfer according to time-varying network condition. It adapts to the dynamic network condition, and furthermore, enhances the performance of the RAT condition where some RATs are unable to be activated simultaneously. In the performance evaluation, we verify that the proposed algorithm shows no significant performance difference from the full search algorithm. Furthermore, we show that the parallel activation scheme improves the RAT diversity gain compared with the vertical handoff. With time-varying

throughput variation of RATs, the performance enhancement through the dynamic update algorithm is also evaluated.

In Chapter 3, we propose a HTTP-based multipath video streaming algorithm for LTE/WiFi-equipped smartphones. The proposed algorithm effectively balances the video quality, i.e., video encoding rate and available playback time considering the remaining battery energy and data quota. We numerically analyze how HTTP-based video services affect the data usage and energy consumption simultaneously. We especially cope with the expected data and energy waste when a user leaves watching a video in midstream. Based on the analysis, we formulate a multi-attribute utility function represented by a weighted sum of the normalized video quality and playback time. We first solve a subproblem that maximizes the playback time by finding the optimal chunk download cycle with given battery energy and data quota, assuming the video streaming service is only available until either the battery energy or data quota is exhausted. Then, we find the optimal operating parameter set, i.e., network selection, video encoding rate, and chunk download cycle, that maximizes the multi-attribute utility function. The proposed algorithm fundamentally uses the available WiFi network and exploits the LTE network to support the bandwidth shortage when the estimated WiFi throughput is less than the selected encoding rate. We show that the proposed algorithm improves the multi-attribute utility much more than the case that only uses either WiFi or LTE.

In Chapter 4, we propose an energy saving algorithm in the standby mode of smartphone. We present an anatomy of power consumption of the LTE and WiFi interfaces in the smartphone. Based on the power consumption anatomy, we discuss the battery energy drain due to the light load traffic generated by applications running in the background. Then, we propose the deep sleep mode (DSM) algorithm to save the energy in the standby mode, which properly turns off the data network interface when a long idle

duration is expected. It monitors the idle duration and triggers the DSM process when the duration exceeds a specific threshold time. The threshold time is obtained based on analysis of each user's smartphone usage profile representing a distribution the idle duration, which commonly follows the heavy-tailed distribution, and is adaptively set to maximize the expected energy saving gain. We exploit real user's smartphone usage log data to practically evaluate the performance of the proposed scheme, and the results show that our proposed algorithm adapts to various user cases and effectively saves the energy in the standby mode.

Bibliography

[1] A. Ford, C. Raiciu, M. Handley, S. Barre, and J. Iyengar. (2011, Mar.) Architectural guidelines for multipath tcp development (RFC 6182).

[2] Samsung Galaxy S5 Guide. (2014) Galaxy s5 download booster. [Online]. Available: http://galaxys5guide.com/samsung-galaxy-s5-features-explained/galaxy-s5-download-booster

[3] Cisco. (2014, Feb.) Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013–2018. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html

[4] J.-M. Kang, S. seok Seo, and J.-K. Hong, "Usage pattern analysis of smartphones," in *Proc. IEEE APNOMS '11*, Sep. 2011.

[5] Salesforce. (2014, Feb.) 2014 Mobile Behavior Report. [Online]. Available: http://www.exacttarget.com/sites/exacttarget/files/deliverables/etmc-2014mobilebehaviorreport.pdf

[6] W. Shen and Q.-A. Zeng, "Cost-Function-Based Network Selection Strategy in IntegratedWireless and Mobile Networks," *IEEE Trans. Veh. Technol.*, vol. 57, no. 6, pp. 3778–3788, Nov. 2008.

[7] R. Amin, J. Martin, J. Deaton, L. A. DaSilva, A. Hussien, and A. Eltawil, "Balancing Spectral Efficiency, Energy Consumption, and Fairness in Future Heterogeneous Wireless Systems with Reconfigurable Devices," *IEEE J. Sel. Area. Comm.*, vol. 31, pp. 969–980, May 2013.

[8] Y. Choi, H. Kim, S. wook Han, and Y. Han, "Joint Resource Allocation for Parallel Multi-Radio Access in Heterogeneous Wireless Networks," *IEEE Trans. Wireless Comm.*, vol. 9, no. 11, pp. 3324–3329, Nov. 2010.

[9] C.-J. Chang, T.-L. Tsai, and Y.-H. Chen, "Utility and Game Theory Based Network Selection Scheme in Heterogeneous Wireless Networks," in *Proc. IEEE WCNC '09*, Apr. 2009.

[10] E. Aryafar, A. Keshavarz-Haddad, M. Wang, and M. Chiang, "RAT Selection Games in HetNets," in *Proc. IEEE INFOCOM '13*, Apr. 2013.

[11] E. Gustafsson and A. Jonsson, "Always Best Connected," *IEEE Wirel. Commun.*, vol. 10, no. 1, pp. 49–55, Feb. 2003.

[12] M. Kassar, B. Kervella, and G. P. University, "An Overview of Vertical Handover Decision Strategies in Heterogeneous Wireless Networks," *Comput. Commun.*, vol. 31, no. 10, pp. 2607–2620, Jan. 2008.

[13] A. Roy, J. Shin, and N. Saxena, "Multi-Objective Handover in LTE Macro/Femto-Cell Networks," *J. Commun. Netw.*, vol. 14, no. 5, pp. 578–587, Oct. 2012.

[14] H. Petander, "Energy-aware Network Selection Using Traffic Estimation," in *Proc. ACM MICNET '09*, Sep. 2009.

[15] Y. Choi and S. Choi, "Service Charge and Energy-Aware Vertical Handoff in Integrated IEEE 802.16e/802.11 Networks," in *Proc. IEEE INFOCOM '07*, May 2007.

[16] Y.-L. Chung, Z. Tsai, and C.-H. Yang, "A Study of Quota-Based Dynamic Network Selection for Multimode Terminal Users," *IEEE Syst. J.*, vol. pp, no. 99, p. 1, Jan. 2013.

[17] Q. Song and A. Jamalipour, "A Network Selection Mechanism for Next Generation Networks," in *Proc. IEEE ICC '05*, May 2005.

[18] Y. Yu, B. Yong, and C. Lan, "Utility-dependent Network Selection Using MADM in Heterogeneous Wireless Networks," in *Proc. IEEE PIMRC '07*, Sep. 2007.

[19] Q.-T. Nguyen-Vuong, N. Agoulmine, and Y. Ghamri-Doudane, "A User-centric and Context-aware Solution to Interface Management and Access Network Selection in Heterogeneous Wireless Environments," *Comput. Netw.*, vol. 52, no. 18, pp. 3358–3372, Dec. 2008.

[20] X. Li, M. Dong, Z. Ma, and F. Fernandes, "GreenTube: Power Optimization for Mobile Video Streaming via Dynamic Cache Management," in *Proc. ACM MM'12*, Oct. 2012.

[21] M. A. Hoque, M. Siekkinen, and J. K. Nurminen, "Using Crowd-Sourced Viewing Statistics to Save Energy in Wireless Video Streaming," in *Proc. ACM Mobicom'13*, Oct. 2012.

[22] J. Chen, A. Ghosh, J. Magutt, and M. Chiang, "QAVA: Quota Aware Video Adaptation," in *Proc. ACM CoNEXT'12*, Dec. 2012.

[23] R. Stewart. Stream control transmission protocol (RFC 4960).

[24] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure. Tcp extensions for multipath operation with multiple addresses (RFC 6824).

[25] Q. Peng, M. Chen, A. Walid, and S. H. Low, "Energy Efficient Multipath TCP for Mobile Devices," in *Proc. ACM MobiHoc'14*, Aug. 2014.

[26] Y. sup Lim, Y.-C. Chen, E. M. Nahum, D. Towsley, and R. J. G. and, "How Green is Multipath TCP for Mobile Devices?" in *Proc. ACM AllThingsCellular'14*, Aug. 2014.

[27] S. Chen, Z. Yuan, and G.-M. Muntean, "An Energy-aware Multipath-TCP-based Content Delivery Scheme in Heterogeneous Wireless Networks," in *Proc. IEEE WCNC'13*, Apr. 2013.

[28] J. Pouwelse, P. Garbacki, D. Epema, and H. Sips, "The BitTorrent P2P File-sharing System: Measurements and Analysis," in *Proc. International Workshop on Peer-to-Peer Systems*, Feb. 2005.

[29] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. (1999, Jun.) Hypertext transfer protocol – http/1.1 (RFC 2616).

[30] D. Kaspar, K. Evensen, P. Engelstad, A. F. Hansen, P. Halvorsen, and C. Griwodz, "Enhancing Video-on-Demand Playout over Multiple Heterogeneous Access Networks," in *Proc. IEEE CCNC '10*, Jan. 2010.

[31] W. Lee, J. Koo, S. Choi, and Y. Park, "ESPA: Energy, Usage ($), and Performance-Aware LTE-WiFi Adaptive Activation Scheme for Smartphones," in *Proc. IEEE WoWMoM '14*, Jun. 2014.

[32] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, "A Close Examination of Performance and Power Characteristics of 4G LTE Networks," in *Proc. ACM MobiSys '12*, Jun. 2012.

[33] C. Yoon, D. Kim, W. Jung, C. Kang, and H. Cha, "AppScope: Application Energy Metering Framework for Android Smartphones using Kernel Activity Monitoring," in *Proc. USENIX ATC '12*, Jun. 2012.

[34] S. Boyd and L. Vandenberghe, *Convex Optimization*. , Cambridge, UK: Cambridge Univ. Press, 2003.

[35] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB Toolbox for Optimization over Symmetric Cones," *Optimization Methods and Software*, vol. 11, pp. 625–633, 1999.

[36] W. A. Gardner, *Introduction to Random Processes: with Applications to Signals and Systems, 2nd ed.*, US: McGraw-Hill, Inc., 1990.

[37] R. Pries, F. Wamser, D. Staehle, K. Heck, and P. Tran-Gia, "On Traffic Characteristics of a Broadband Wireless Internet Access," in *Proc. IEEE NGI '09*, Jul. 2009.

[38] R. H. Tütüncü, K. Toh, and M. Todd, "Solving Semidefinite-Quadratic-Linear Programs using SDPT3," *Mathematical Programming*, vol. 95, pp. 189–217, 2003.

[39] Y. Chen, B. Zhang, , Y. Liu, and W. Zhu, "Measurement and modeling of video watching time in a large-scale internet video-on-demand system," *IEEE Trans. on Multimedia*, vol. 15, pp. 2087–2098, 2013.

[40] X. Xing, J. Dang, S. Mishra, and X. Liu, "A Highly Scalable Bandwidth Estimation of Commercial Hotspot Access Points," in *Proc. IEEE INFOCOM '11*, Apr. 2011.

[41] J. Huang, F. Qian, Z. M. Mao, S. Sen, and O. Spatscheck, "Screen-off traffic characterization and optimization in 3G/4G networks," in *Proc. ACM IMC '12*, Nov. 2012.

[42] S. Deng and H. Balakrishnan, "Traffic-aware techniques to reduce 3G/LTE wireless energy consumption," in *Proc. ACM CoNEXT '12*, Dec. 2012.

[43] R. Wang, J. Tsai, C. Maciocco, T.-Y. Tai, and J. Wu, "Reducing power consumption for mobile platforms via adaptive traffic coalescing," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 8, pp. 1618–1629, Sep. 2011.

[44] E. J. Vergara, J. Sanjuan, and S. Nadjm-Tehrani, "Kernel level energy-efficient 3G background traffic shaper for android smartphones," in *Proc. IEEE IWCMC '13*, Jul. 2013.

[45] F. Xu, Y. Liu, T. Moscibroda, R. Chandra, L. Jin, Y. Zhang, and Q. Li, "Optimizing background email sync on smartphones," in *Proc. ACM MobiSys '13*, Jun. 2013.

[46] "Android google play." [Online]. Available: https://play.google.com/store/apps

[47] "Apple itunes." [Online]. Available: http://www.apple.com/kr/itunes/

[48] H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, and D. Estrin, "Diversity in smartphone usage," in *Proc. ACM MobiSys '10*, Jun. 2010.

[49] A. Shye, B. Scholbrock, G. Memik, and P. A. Dinda, "Characterizing and modeling user activity on smartphones: summary," in *Proc. ACM SIGMETRICS Performance Evaluation Review '10*, Jun. 2010.

[50] R. J. Adler, R. E. Feldman, and M. S. Taqqu, *A Practical Guide to Heavy Tails: Statistical Techniques and Applications*. Birkhauser Boston Inc., 1998.

[51] S. Gebert, R. Pries, D. Schlosser, and K. Heck, "Internet access traffic measurement and analysis," in *Proc. TMA Workshop '12*, Mar. 2012.

[52] M. Molina, P. Castelli, and G. Foddis, "Web traffic modeling exploiting TCP connections' temporal clustering through HTML-REDUCE," *IEEE Network*, vol. 14, no. 3, pp. 46–55, May 2000.

[53] "Monsoon power monitor." [Online]. Available: http://www.msoon.com/LabEquipment/PowerMonitor

[54] "Android background service." [Online]. Available: http://developer.android.com/training/run-background-service/index.html

[55] "Android powermanager." [Online]. Available: http://developer.android.com/reference/android/os/PowerManager.html

[56] "Android alarmmanager." [Online]. Available: http://developer.android.com/reference/android/app/AlarmManager.html

초 록

오늘날 스마트폰은 3G, 4G, WiFi 및 블루투스와 같이 다양한 이종 네트워크 접속 기술이 함께 탑재되어 있다. 또한, 최신 스마트폰의 경우에는 이러한 이종 네트워크를 동시에 접속할 수 있는 기술이 탑재되고 있는 추세이다. 따라서, 이종 네트워크 동시 접속을 포함하여 최적의 네트워크 조합을 선택하고, 선택된 네트워크 별로 최적의 전송 데이터 할당량을 결정하는 방법이 필요하다. 본 학위 논문에서는 파일 전송, 비디오 스트리밍 등의 인터넷 서비스를 지원 하기 위한 다중 네트워크 연결 시스템을 디자인하고, 다 속성 기반 다중 네트워크 운용 기법을 제안한다. 이를 위해 서비스 별 특성을 고려한 성능, 에너지 및 서비스 요금에 대한 다 속성 비용 함수를 모델링 하고 분석을 통한 최적화를 진행한다. 또한, 성능 검증을 위하여 실제 스마트폰 측정으로부터 얻은 파라미터에 기반한 시뮬레이션을 수행하고, 제안 알고리즘이 각 서비스 특성과 스마트폰의 현재 에너지 및 데이터 요금 잔량 상태에 따라 성능, 에너지, 요금 각각의 상대적 비용에 균형을 맞추면서 적응적으로 최적 동작 모드를 결정하는 것을 보여준다.

먼저, 제 2장에서는 2개 이상 n개의 라디오 접속 기술이 탑재된 스마트폰에서 대용량 파일 전송 시의 전송 완료 시간, 에너지 소모, 데이터 요금 각각에 대한 비용 함수를 모델링하고, 이를 정규화하여 통합 비용함수를 도출한다. 그리고 통합 비용 함수값을 최소화 하기 위한 네트워크 조합 및 선택 네트워크 별 최적 파일 세그먼트 할당량을 찾는 알고리즘을 제안한다. 이 때, 가용 네트워크 조합은 $2^n - 1$개 되므로,

가용 네트워크 수가 증가함에 따라 최적 네트워크 조합을 찾기 위한 계산량은 지수적으로 증가하게 된다. 따라서 이러한 복잡도를 선형적으로 줄이는 알고리즘을 제안한다. 또한, 네트워크 전송률은 동적으로 변화하기 때문에 초기 설정값의 최적성이 깨어질 수 있는데 이를 보완하기 위해 주기적으로 비용함수를 체크하고 최적 네트워크 선택을 업데이트 하는 알고리즘을 제안한다. 제안 알고리즘의 계산량 및 성능은 실측 기반 시뮬레이션을 통하여 검증한다.

제 3장에서는 LTE/WiFi 네트워크 인터페이스가 탑재된 스마트폰에서 비디오 스트리밍 시의 트래픽 패턴에 따른 에너지 및 데이터 소모량을 분석한다. 이 때, 사용자가 비디오 재생 중간에 시청을 종료할 경우 낭비되는 에너지와 데이터의 예상값을 함께 분석한다. 이를 기반으로, LTE 네트워크 사용 시의 배터리 잔량 및 데이터 요금 잔량에 따른 비디오 스트리밍 서비스 가능 시간을 최대화하는 알고리즘을 제안한다. 뿐만아니라, WiFi사용 가능 시에 LTE와 WiFi를 동시에 사용하여 비디오 데이터를 전송받는 기법을 제안하고, 비디오 품질 및 비디오 시청 가능 시간에 대한 통합 유틸리티 함수값을 최대화 하도록 최적 네트워크 조합 및 비디오 인코딩 레이트를 선택하는 알고리즘을 제안한다. 사용자의 비디오 시청 로그에 기반한 시뮬레이션을 통해 제안 알고리즘이 비디오 품질 및 비디오 시청 가능 시간 사이에 적절한 균형을 맞추며 동작함을 보인다.

한편, 스마트폰은 일반적으로 하루 중 대기모드 상태로 있는 시간이 가장 길다는 특성을 고려하여, 제 4장에서는 LTE 또는 WiFi 접속 상태인 스마트폰이 대기모드일 때, 백그라운드 트래픽에 의한 에너지 소모 특성을 분석한다. 또한, 사용자별 하루 동안의 스마트폰 미사용 시간 분포 특성을 장시간 스마트폰 로그 측정을 기반으로 하여 분석하고 이를 기반으로 대기모드에서의 에너지 절약 기법으로서, 네트워크 연결을 ON/OFF하는 Deep Sleep Mode 알고리즘을 제안한다. 여러 사용자의 단말 사용 시간 로그에 기반한 시뮬레이션을 통해 제안 알고리즘의 대기시간 에너지 절약 성능을 검증한다.

Keywords: 3G, LTE, WiFi, 스마트폰, 다중라디오 접속기술, 파일전송, 비디오 스트

리밍, 에너지 절약, 요금 절약, 다속성 기반 결정 기법