MASTER

User conflict prevention in a connected lighting system

Xu, Y.

Award date:
2015

Department of Mathematics and Computer Science

System Architecture and Networking Group

User Conflict Prevention in a Connected Lighting System

Master Thesis

by

Yiting Xu

<y.xu@student.tue.nl>

University Supervisor: dr. Tanir Ozcelebi <t.ozcelebi@tue.nl>

Company Supervisor: Henk Stevens <henk.stevens.2@philips.com>

Tutor: Salih Serdar Guclu <s.s.guclu@tue.nl>

Eindhoven, the Netherlands

28th September 2015

Abstract

In office environment, it is important to achieve a satisfying lighting condition so that office workers are able to work productively. Recently, intelligent connected lighting systems are designed to provide better and personalized lighting for office workers. This master thesis proposes a system architecture for a connected lighting system for an open plan office area, in which office workers can not only set their personal lighting settings in their smartphones, but also ask for specific settings on ambient parameters. A user's requirements are grouped as a profile. However, in most cases, people are not easy to get desks where exactly meet their profiles by themselves, so profile conflicts may arise.

In this master thesis, the problem is formulated as a constrained optimization problem, and we propose an algorithm to prevent the potential conflicts. The system assigns an optimal desk for the office worker according to his/her profile and the current system settings, and tries to prevent the potential conflicts by maximizing user satisfaction. Then three system architectures are proposed and compared with regard to response time, scalability, availability, etc. We select a suitable architecture and simulate it in Cooja, a network simulator of Contiki OS. Network performance and quality attributes of the system are tested in the simulation. Users are able to get an optimal desk with a maximal user satisfaction so that potential conflicts are prevented. The response time for an office worker's check-in is 61 ms in average, if a powerful server is provided. Scalability and fault tolerance are properly fulfilled. Other results and discussions about the system are presented.

Acknowledgement

In the first place, I would like to thank Philips Lighting B.V. for giving me the chance to put my knowledge into practice and offering me a wonderful opportunity to work within the area of Embedded Systems. In particular, I want to thank Henk Stevens, who gave me a lot of insights, as well as the knowledge on system architecture. I learned a lot from Henk during my internship and when I needed support, he was always ready for me despite his busy schedule. In addition, I would like to thank dr. Tanir Ozcelebi of the Eindhoven University of Technology for his scientific support and control of the master project process, especially for checking the theoretical parts. Next, I would like to thank Salih Serdar Guclu for the support he gave, especially in the simulation part. Last but not least, I would like to thank my parents for their moral support. Thanks to my parents, I am able to study, to develop myself and to prepare myself for the future. I am eternally grateful for that.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

6LoWPAN	IPv6 over Low power Wireless Personal Area Networks
AP	Access Point
CoAP	Constrained Application Protocol
CPU	Central Processing Unit
CSMA	Carrier Sense Multiple Access
CSV	Comma Separated Values
DMIPS	Dhrystone MIPS
DODAG	Destination Oriented Directed Acyclic Graph
HVAC	Heating, Ventilating, and Air Conditioning
IoT	Internet of Things
IP	Internet Protocol
LLN	Low-power and Lossy Networks
MAC	Medium Access Control
MCU	Microcontroller Unit
MIPS	Million Instructions per Second
MTTF	Mean Time To Failures
MTTR	Mean Time To Repair
OS	Operating System
RAM	Read-Access Memory
RDC	Radio Duty Cycling
RISC	Reduced Instruction Set Computing
ROM	Read-Only Memory
RPL	Routing Protocol for Low-power and Lossy Networks
UDP	User Datagram Protocol
VLC	Visual Light Communications
WSN	Wireless Sensor Network

1. Introduction

Connected lighting systems are prosperously developing recently, and can be applied in many scenarios. This chapter firstly describes the background of lighting industry, and then explains what connected lighting systems are, as well as their application scenarios. Since the connected lighting system in this master thesis is used in an open plan office area, a definition of the open plan office is followed. Finally, the motivation of doing this master project is stated, followed by the thesis layout.

1.1. Background

Nowadays, the lighting industry is undertaking a revolution towards intelligence and energy efficiency. Philips Lighting B.V., a global market leader in lighting industry, is driving innovation in professional lighting systems to allow lights to achieve a degree of intelligence, when combined with controls and software, and linked into a network. Such connected lighting systems [1] can be applied in many scenarios, like road lighting, arena lighting, office lighting, etc.

This master thesis focuses on connected lighting systems for open plan office areas, where office workers may have different lighting and ambient requirements for their personal preferences or specific tasks. Especially in an open plan office area, in which office workers have flexible workspace, workers need to adapt the light settings to personal preferences more often than when they have a dedicated working place. Good lighting tailored to individual tasks help staff work more efficiently during the day.

1.2. Connected Lighting Systems

Connected lighting systems not only provide illuminated indoor and outdoor spaces, but also deliver value beyond illumination to the users and managers of spaces. In a connected lighting system, every lighting point serves additional functionalities beyond high-quality, reliable illumination: they are connected to an intelligent system that provides information and services required by stakeholders. Furthermore, connected lighting systems pursue energy-efficiency, personalized spaces, performance tracking, optimized management, transformable environments, integrations with other systems, etc. [1] By using a connected lighting system, customers can monitor and manage the lighting system remotely. Energy consumption and occupancy related data, collected by the connected lighting system, would be easily shared with other parties to improve the building performance or to perform data analytics. In different scenarios, people, spaces and luminaires are connected within a connected lighting system, so it is able to generate extra value for customers, beyond simple illumination.

For public spaces, connected lighting systems are one of the important components of intelligent cities, and they exist in every public space in a city: roads, parks, tunnels, airports, arenas, and bridges. With connected lighting systems, those facilities can be illuminated remotely, flexibly and efficiently. By using energy-efficient luminaires, the system could also achieve low energy consumption, reduced costs, and optimized maintenance processes. From the perspective of residences, they can enjoy a better experience. For example, by dynamic color-changing LED lighting, brilliant features in the municipality are created that can make a tremendous impact on the local community and attract more visitors. Another example is sports arenas. The right lighting is vital for sports venues, from smaller professional sports stadiums to major multi-functional event arenas, a connected lighting system makes the difference between an average game and an exciting event, clear visibility on the field helps players give a peak performance, and architectural lighting enhances the atmosphere, attracts visitors, but reduce operational costs.

Besides the outdoor applications, connected lighting systems also work well in indoor environments: food retailers, large retailers, restaurants, gas stations, etc. Take retailers as an example, dynamic lighting and sophisticated controls create a unique in-store ambience, giving customers a reason to stay and keep coming back. Connecting with people's smartphones, the system could provide online guidance, more detailed information about products, etc. A case study [2] is that Philips recently introduced a system that connects in-store LED lights with consumers' smartphones. Using downloadable apps, people will be able to locate items on their shopping lists or get coupons as they pass the products on the aisles. Retailers can send targeted information such as recipes and coupons to consumers based on their precise location within stores, while gaining benefits of energy-efficient LED lighting. This system uses VLC (Visual Light Communications) to talk with consumers' smartphones.

1.3. Open Plan Office Areas

In the indoor scenarios, office environment is not negligible, because people usually spend forty hours per week in their offices, and office lights energize people to work. Connected lighting systems can optimize people's office into a personalized workspace, as well as controllable and energy-efficient lighting spaces, to reach an ideal state for productive workforces. In addition, intelligent IP luminaires can merge with a facility's IT and power infrastructures so that the lighting system can serve as a pervasive data communication platform. What is more, lighting management becomes one aspect of a system that includes other important services, including HVAC (Heating, Ventilating, and Air Conditioning), maintenance, day lighting, environmental and chemical monitoring and compliance.

Typically, an office building is composed of various areas: reception areas, corridors, meeting rooms, open areas, private offices, and even restaurants and parking areas, etc. Among them, open plan office area is the most important area for ordinary office workers, where most of people spend most of their working time. In open plan office area, there are various activities and many individuals stay in the same place, the lighting system should adapt to such an adaptable working environment. It should create a pleasant, motivating workspace, and can be personalized to help team members to work more productively. This kind of area is usually a flexible workspace, where people are able to select their desks when entering the office and people sit differently everyday.

Therefore, this thesis focuses on an open plan office area and the proposed system tries to meet lighting and ambient requirements for office workers who work in this area. However, in open plan office areas, different people may have different tasks, which require different lighting settings. Besides, personal preferences may also affect the needs of lighting settings, which are related to their ages, handicaps, moods, etc. For example, a study [3] shows that elderly people have different preferences on LED lightings than young people. Handicaps related to lights might be color-blindness. Therefore, the connected lighting system in this area should provide people with personalized lightings. It is obvious that satisfying lightings that are tailored to individual tasks help staff work more efficiently. Compared with a dedicated working place, workers in open plan office area need to adapt the light settings to personal preferences more often.

Besides personalization, the second requirement that a connected lighting system for open plan office areas should meet is related to the requirements of system performance. To be more specific, shorter response time, higher throughputs, lower utilization of resources, higher bandwidth, and shorter data transmission time could be metrics of performance for a system. These requirements mainly depend on the system architecture, and greatly differ among different architectures. Performance also affects user satisfaction, for instance, a shorter response time of the system brings better satisfaction for users.

Overall, a connected lighting system usually meets the requirements of personalization and user-interactive performance. This thesis formulates the problem in terms of personalization, analyzes and compares three proposed network architectures to pursue better system performance, and finally provides a simulation for the selected system architecture in order to show those features.

1.4. Motivation of Thesis

A connected lighting system for open plan office area is designed in this thesis. In such a connected lighting system, office workers can define their preferences of the luminaires via app on their smartphones. These preferences turn to be his/her personal lighting profile and will be shared with the connected lighting system. However, in most cases, people are not easy to get desks where exactly meet their profiles, and conflicts may arise. This thesis provides an algorithm based on constrained optimization problem, such that conflicting settings will be prevented within the capabilities of the lighting system. The conflict prevention algorithm tries to avoid conflicts at the moment when users check in with their profile, by assigning an optimal desk to maximize their user satisfaction. This thesis also proposes three different network architectures to fit in such an adaptable working environment and personal lighting profiles, and one of them is selected to be simulated and tested, in terms of performance, user satisfaction, functional and non-functional requirements.

1.5. Thesis Layout

This thesis is organized as follows. Chapter 2 describes the problem, defines the personal lighting profile and explains profile conflicts. Chapter 3 formulates the problem as a constrained optimization. Three different network architectures of a connected lighting system are proposed in Chapter 4. Chapter 5 measures several criteria based on assumptions to find the best architecture used for this system. Chapter 6 describes the simulation tools and processes used for this system. Chapter 7 conducts nine experiments, analyzes the results, and discusses the performance of the conflict prevention algorithm and the selected system architecture. Chapter 8 concludes the thesis and lays out the future work.

2. Problem Description

As mentioned in Chapter 1, connected lighting system serves for different areas in an office building, and here the problem is limited to the application scenario of an open plan office area. Besides, in order to realize personalization, a connected lighting system is proposed, which allows office workers to define their preferences of the light settings as well as other ambient parameters, via an app on their smartphones. These preferences turn to be his/her "personal lighting profile" and will be shared with the connected lighting system. However, in most cases, people are not easy to get desks where exactly meet their profiles, and profile conflicts may arise. A solution needs to be found, to prevent the conflict settings, with the capabilities of the connected lighting system.

This chapter introduces the application scenario that the connected lighting system works for, defines the attributes used in the personal lighting profile, and explains the profile conflicts with some use cases. Finally, some related works are found and introduced in this chapter.

2.1. Application Scenario

In a flexible open plan office workspace, people are free to select whichever desk he/she prefers if it is not occupied yet. When a new-coming office worker enters the office, a connected lighting system firstly fetches his/her personal lighting profile via an interface, for instance via an app on his/her smartphone. Then the system arranges a desk according to his/her personal lighting profile as well as the current desk occupancy in this open plan office area. The system finds an optimal desk and tries to prevent conflicts if there happens conflicts on his/her personal light settings with surroundings. The system sends back the optimal desk to his/her smartphones, and he/she can either accept or decline this result.

At the very beginning, an open plan office area layout needs to be defined, so that the system architecture could be proposed to suit for the office layout. A reference guide of Philips Lighting B.V. [4] gives three typical designs for open plan office areas: standard, advanced and premium. They use the same office plan view, which is shown in the Figure 2.1 (unit: mm), but use different luminaires for the three level designs. Luminaires are distributed by 3x8, and the tables and seats are located by 2x8, with every two of them grouped as a set. The luminaires are allocated evenly on the plane of the ceiling, 1800mm gap between two neighboring luminaires in row, and 2400 mm gap between two neighboring luminaires in column. The first and third rows of the luminaires are aligned to the table and seat beneath. All luminaires are in the same type.

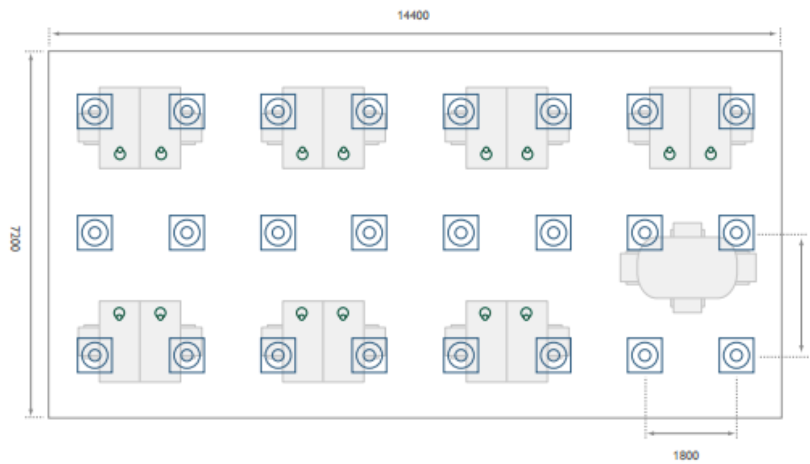


Figure 2.1 Bird's eye view of a connected lighting system for open plan office area

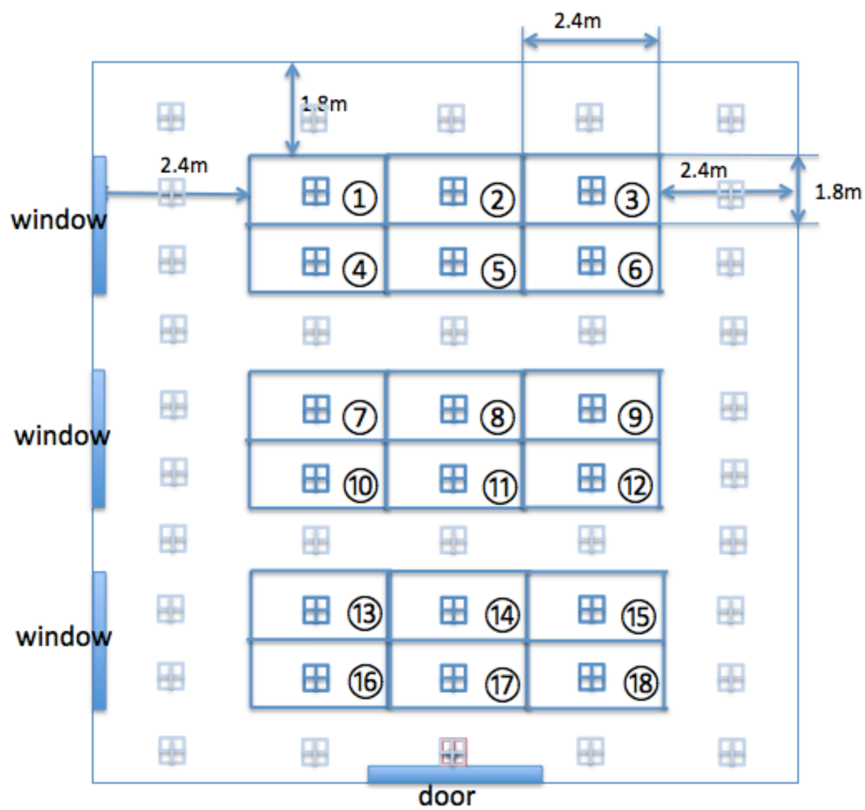


Figure 2.2 Bird's eye view of the office layout and luminaire layout

Inspired by Figure 2.1, the layout will be used in the proposed connected lighting system is shown in Figure 2.2. This office can accommodate 18 office workers. The office layout and luminaire layout are shown in Figure 2.2. The open plan office area is composed of 18 desks and seats. Each desk is 1.8m x 2.4m, with one luminaire aligned exactly above the center of the desk (labeled as dark luminaires in Figure 2.2). Desks are allocated into three groups, each with six desks close to each other. The number in circles in the figure represents the number of desks. Three windows are included in this office, each of them are 3.6m in width, aligned to one group of the desks. Luminaires are allocated evenly by 5x10, and the gap distance is 1.8m and 2.4m in column and row respectively. Luminaires use the same type of LEDs. The distance from the ceiling plane to the floor plane is 2.5 m, for a typical office.

2.2. Personal Lighting Profile

Personal lighting profile is a dataset, which contains office worker's preferences on the lighting settings, as well as some other ambient parameters. A personal lighting profile must contain a profile ID to identify a user. The system forwards a profile ID number to every user when he/she creates an account. Lighting preferences could contain illuminance, color temperature, etc. Office workers may have different preferences on illuminance and color temperature, depends on their specific tasks and ages. For example, a study [5] shows that the elderly prefer bright and light colors and dislike pale and dark colors, so luminaires in a warm color temperature may better suit for the elderly. Noteworthy, the values of illuminance and color temperature are hard to tell directly by human eyes, so the app should show some visualized figures to help people select their preferred illuminance and color temperature, rather than just giving a blank space to input the values.

Other parameters involve personal feelings like window side or aisle side, as some people would prefer to sit beside window because a study of Jaffe, E [6] shows that workers in windowless offices lose more sleep at night. Temperature can also be contained into the profile, because some people would like to work in a colder place, while others prefer warmer places.

Some other parameters were considered when designing the contents of a profile. Social parameters could also be mentioned, like sitting beside a familiar person. However, this requires privacy related information from users, and the friendship among people is not easy to define. Therefore, the proposed system would not contain social parameters into profile. A personal lighting profile could also contain calendars, like meeting time or holiday time. If the person is in a meeting, the desk could be released again. During the holiday time, half the office maybe closed. This could be a functionality offered by the system, instead of offered by office workers. We also do not consider this attribute into a profile.

Although many attributes could be included in the personal lighting profile, a set of personal lighting profile is defined as follows for the proposed connected lighting system. Other attributes that people may concern about could be found in the result of a survey in Appendix A.

(1) Illuminance: Illuminance is a measure of how much luminous flux is spread over a given area, in a unit of lux. A typical illumiance for office lighting is in the range of 320~500 lux [7]. We do not consider sun light as a light source.

(2) Color Temperature: The color temperature of a light source is the temperature of an ideal black-body radiator that radiates light of comparable hue to that of the light source, in a unit of Kelvin (K). Color temperatures over 5,000K are called cool colors (bluish white), while lower color temperatures (2,700–3,000 K) are called warm colors (yellowish white through red) [8]. There is no existing standard to regulate the color temperature in office, but a study [9] shows that a cooler (higher color temperature) light is used to enhance concentration in offices. Therefore, for this connected lighting system, a range of color temperature is set as 3000K~6500K.

(3) Window Side Preference: Window side means that the desk is just beside a window or not, and it is easily defined as preferred or not preferred.

(4) Temperature: Temperature in an office is not easy to define because many factors affect it. Window side may have higher temperature because of sunshine, and the places nearer to air-conditioner may have a peak value in the office. Summertime and wintertime also differs. A study on "thermal comfort" [10] recommends that the temperature in an office should be held constant in the range of 21-23°C. While a standard [11] shows that an acceptable range of 23-28°C and 20-25°C for summertime and wintertime respectively, depending on different humidity. In general, to give a bigger choice range, the range of temperature in this connected lighting system is set as 19-27°C.

2.3. Configuration of Open Plan Office Area

Figure 2.3 shows the configuration of the open plan office area, which will be used in the following chapters to formulate problem and propose system architectures based on it. We assume that the luminous intensity of each luminarie is given by an initial setting, and users are not allowed to control the luminaires directly on the desk. If the system allows users to adjust the luminaires, a new conflict with neighboring office workers nearby is likely to rise. The conflict that the connected lighting system tries to prevent is introduced in Section 2.4.

The other parameters: color temperature, window side and temperature are already fixed in this open plan office area. Color temperature provided by the luminaires from the top row to the bottom row in the figure is set as 6000K, 5500K, 5000K, 4500K, 4000K and 3500K respectively. It also assumed that the temperature in this area is allocated in column: the left column enjoys 25°C; the middle one has 23°C while the right column is in 21°C zone. This layout of temperature is based on the distances to the windows, and temperature can be controlled and maintained by the HVAC system in this building. The windows is not considered as light sources for simplicity, though sunlight could be a light source, but it is time varying and hard to calculate and it also affects temperature and illuminance in this office room. The desk No. 1, 4, 7, 10, 13, and 16 are recognized as a desk with window side.

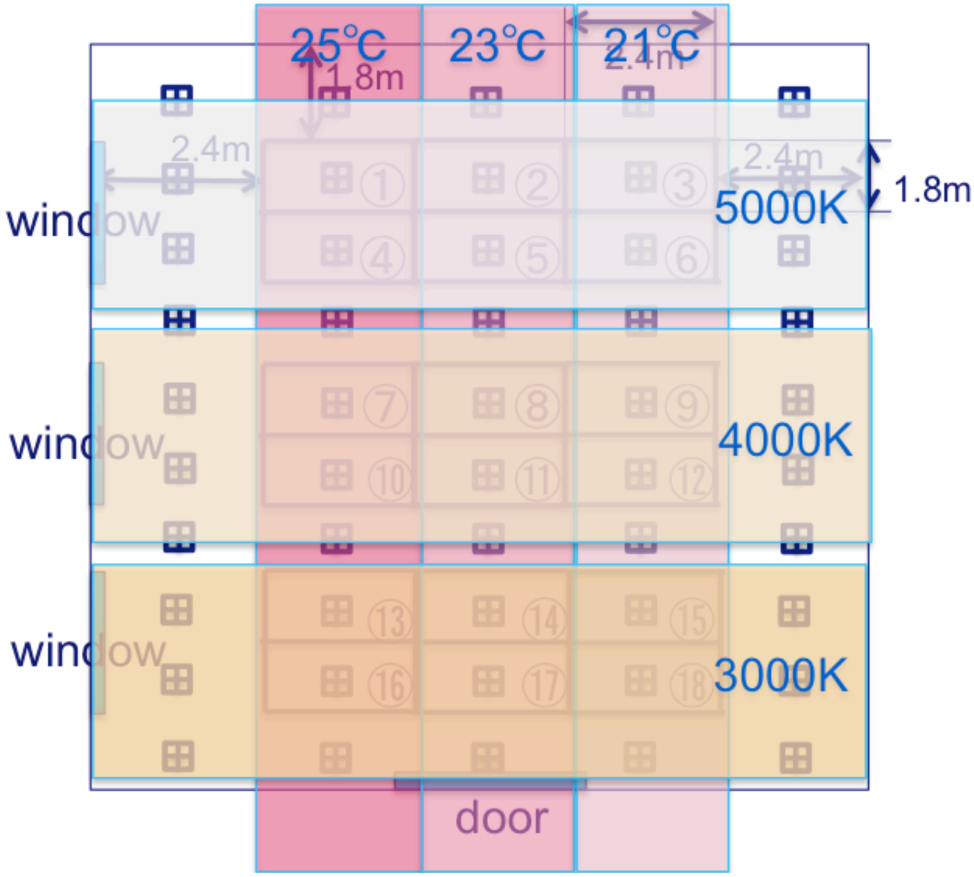


Figure 2.3 Configuration of the open plan office area

2.4. Profile Conflicts

Profile conflicts may come when the system finds a desk for an office worker in the abovementioned office, because the office may not completely satisfy with all the parameter of people's personal lighting profile. Some of the parameters have conflicts with the current distributions of lighting, ambient settings and desk occupancy. In order to understand the conflict problems well, two use cases are listed here:

(1) Use case 1: All desks are available, and at one moment, a person comes in with a personal lighting profile of 400 lux, 5200K, window side preferred and 21°C. Actually in this open plan office area, there is no desk exactly meets all the parameters of his/her preferences. Thus, the connected lighting system should provide a solution to best meet his/her requirements or in other words, to maximize his/her user satisfaction.

(2) Use case 2: Only desk No. 1 is already occupied at one moment; when a person with a preference of 400 lux, 6000K, window side preferred and 25°C comes into the office, the system should have allocated the desk No.1 to this person because the settings of desk No.1 matches this profile. However, this desk is already occupied, so an optimal solution among other 17 desks

should be found by the connected lighting system. The system should get a solution according to an algorithm, which will be proposed in the next chapter.

Since users are not able to adjust luminaires, we assume that there is no illuminance conflict with neighboring users. Thus, we need to prevent conflicts at the moment that an office worker enters this office. Figure 2.4 shows how the conflict prevention mechanism works inside the connected light system.

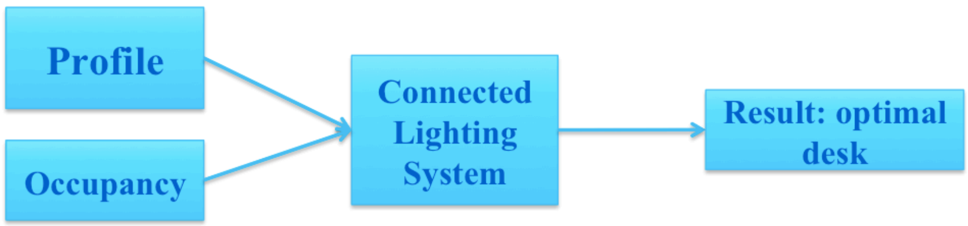


Figure 2.4 Inputs and outputs of the connected lighting system

The system fetches the personal lighting profile from an office worker's smartphone, collects current occupancy information from all the LED controllers above the desks, and gives the user a result of optimal desk based on a certain algorithm. The decision-making mechanism will be given in the next chapter.

2.5. Related Work

Khaled M. Khalil, et al. [12] gives a general idea on how to select conflict-resolving strategy, and lists a comparison of intelligent techniques for knowledge conflict resolutions. From the comparisons of different strategies, we select to use 'Searching based Technique' as our strategy, which uses some kind of searching algorithms to find solutions compatible across the agents' community. In our case, the goal is to try to prevent conflicts and finding an optimal desk is a solution for an office worker who requests a desk to suit for his/her profile. Such a strategy is a kind of mediation strategy, which can be used as centralized or distributed topology. So we can at least propose two architectures for this system: one is centralized style, the other one is distributed style. However, this strategy only serves for single conflict, and has no learning mechanism.

G.R. Newsham, et al. [13] gives a general lighting design for open plan offices. It gives some suggestions for illuminance selection and indiviadual lighting controls. Especially, it gives some recommendations for illumination and luminance selection for open plan office: illumiance on desk working surfaces: 400~500 lux and luminance on major surfaces > 30 cd/m^2. That is a good reference for us to decide the range of illuminance value for the desks in the open place office, so that we can set a range for illuminance value for the profiles accordingly. However, this paper also analyzes the desktop illuminance levels and illuminance from daylight. Since they did not give any formula or recommendation values for desktop illuminance and daylight, we do not consider these two parameters into our formulization.

X. Wang, et al. [14] gives a model for comfort and energy efficiency in light distributions for an indoor environment. They proposed a concept of satisfaction function, which describes relationship between illuminance and the user' s satisfaction with the lighting condition. About 20% of the energy consumption can be saved by adopting their proposed illuminance strategy. What we can learn from this paper is that the way they formulate and calculate user satisfaction, where he models user satisfaction by Gaussian distribution function. In addition, they use logarithm to compare the actual brightness and expected brightness value because human eye senses brightness approximately logarithmically over a moderate range. That's why we also use logarithm in our formulization for the illuminance attribute.

Xin Wang, et al. [15] models a smart LED lighting control system in terms of energy consumption and user comfort, and uses adaptive simulated annealing to generate the trade-off curve between increasing user comfort and reducing energy consumption in different office layouts. Their method of formulating the problem into an optimization problem is a good reference for this project. However, they think that energy consumption and user comfort is a pair of trade-off. In our case, energy consumption is not a big issue. What we only care about is to improve the user comfort, or we say 'user satisfaction'. Because potential conflicts are related to user satisfaction, instead of energy consumption, we only consider user satisfaction as a single objective. We would not formulate into a multiple objective optimization problem.

H. Yang [16] researches on signal processing for LED lighting systems, in particular, the primary role of such systems: illumination rendering and sensing. In this dissertation, the analytical model of illumination rendering by a single LED is our interest, and from this part of this work, we find a simple model to describe the spatial illuminance distribution and a complex model: Generalized Lambertian Model. In our case, we need to calculate the illuminance of desk by the sum of effects from all the luminous intensity values of the LEDs in the office, so the simple model given in this dissertation would help us.

P. Thirumal, et al. [17] uses Multi-objective genetic algorithm to optimize the indoor air quality characteristics. In this paper, three objectives are mathematically modeled, and then use genetic algorithm to optimize the problem. The paper shows a good example of how to model a parameter like temperature and humidity, and how to find a relation among these objectives. However, in our problem, no objectives are found to be interrelated, or trade-off, that is why we gave up using Multi-objective optimization or Pareto analysis.

F. Petrushevski, et al [18] describes how lighting control agents manage conflicts when multiple users share luminaires for their lighting control system. In their proposed conflict management algorithm, they classify the luminaires into non-shared and shared luminaires when a user checks into the office. If there are no shared luminaires, then the system adjusts outputs of these luminaires by single-user control. They think effects of these adjustments on lighting conditions for existing users are considered as negligible. If there are shared luminaires, the algorithm tries to satisfy lighting preferences of the new user with non-shared luminaires only. However, in our assumptions, the system does not allow users to control the luminaires, and we do not need to classify the luminaries into shard and non-shared. In their formulization, we can find that they use Lagrange optimization methods, because they have one single objective, and allocate weights for different luminaires. We can also use the similar way to formulate our problem, since a profile has several attributes, and we could find a single objective for the optimization problem to combine those attributes, and allocate weights for different related attributes.

3. Problem Formulization

Previous chapters describe the problem. In order to prevent the conflicts in the office, the problem will be formulated in this chapter. An algorithm is proposed for the conflict prevention, whose aim is to maximize user's satisfaction. The problem is formulated as a constrained optimization problem, so constraints are listed and the values of some parameters in the algorithm are discussed and set. Finally, we propose a method to get the illuminance of desks by calculation instead of using sensors.

3.1. Profiles, Desks and Luminaries

Profiles, desks and luminaries are the three main objects in this system, so at first they should be defined. Here we assume m profiles totally in the system, also means that m people have the authority to enter this open plan office area, each profile is represented as:

$$p_i, i \in [1,2,3,\ldots,m] \tag{3.1}$$

In each profile, four attributes are contained: T_{pi}, E_{pi}, C_{pi}, L_{pi}. They are expected temperature, illuminance, color temperature and location of the profile p_i respectively. Here, location means window side or not. What is more, people can also define the weights for each attribute in their smartphones, because some people probably care about temperature more, while others may pay attention to illuminance more. Therefore, α_{Tpi}, α_{Epi}, α_{Cpi}, α_{Lpi} are defined for each corresponding attributes of the profile p_i.

Similarly, n desks are located in the room, and each desk is represented as:

$$d_j, j \in [1,2,3,\ldots,n] \tag{3.2}$$

For the desk d_j, four attributes are fixed: T_{dj}, E_{dj}, C_{dj}, L_{dj}. They are the temperature, illuminance, color temperature and location value of the desk d_j. These values are already defined according to the configuration of open plan office area, described in Section 2.3 (see Figure 2.3), and here these values are listed in Table 3.1.

Table 3.1 The values of T_{dj}, C_{dj}, L_{dj}

j	T_{dj} (℃)	C_{dj} (K)	L_{dj}	***j***	T_{dj} (℃)	C_{dj} (K)	L_{dj}
1	25	3000	1	10	25	4000	1
2	23	3000	0	11	23	4000	0
3	21	3000	0	12	21	4000	0
4	25	3000	1	13	25	5000	1
5	23	3000	0	14	23	5000	0
6	21	3000	0	15	21	5000	0
7	25	4000	1	16	25	5000	1
8	23	4000	0	17	23	5000	0
9	21	4000	0	18	21	5000	0

Since E_{dj} is initialized differently by system administrators, we will give a setting for E_{dj} when simulating the system.

Similarly, s luminaires are used for the system, and each luminaire is represented as:

$$l_k, k \in [1,2,3,\ldots,s] \tag{3.3}$$

For the luminaire l_k, I_{lk} is its luminous intensity under the source.

Since it is considered that the number of profiles is greater than the number of desks in the room, and also the number of luminaires is greater than the number of desks (see Figure 2.3), the relationship among the three objects is defined as:

$$m > n,\ s > n \tag{3.4}$$

3.2. User Satisfaction

User satisfaction is a measure of how products or services meet or surpass customer expectation. As described previously, user satisfaction is what we measure for the conflict prevention. User satisfaction can be quantified by his/her satisfaction function, which ranges from 0 to 1. A satisfaction function is given in X. Wang, at el. [14], which indicates that user satisfaction is distributed as a Guassian function, related to the difference between an expected and actual attribute, shown in (3.5). An example satisfaction function for illuminance is used in X. Wang, at el. [15].

$$S(x) = \alpha e^{-\frac{(e-r)^2}{2\sigma^2}} \tag{3.5}$$

, where *S(x)* is the satisfaction function for the expected value *e* and the real value *r*, shown in Figure 3.1. The weight α and the standard deviation σ need to be found later as α is the maximum satisfaction of the user and σ is thought as tolerance of the user. α is the maximum value when *e*=*v*, and it is the maximum value that *S(x)* can reach. In Figure 3.1, the shape of Gaussian distribution function depends on the value of σ.

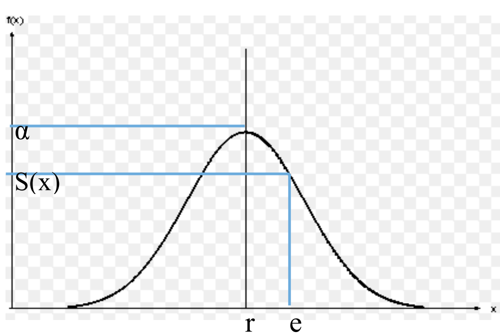


Figure 3.1 Satisfaction function

3.3. Constrained Optimization

In mathematical optimization, the method of constrained optimization is a strategy for finding the local maxima and minima of a function subject to some constraints [19]. A typical optimization problem is defined like:

$$\textit{maximize} \quad f(x, y) \tag{3.6}$$

$$\textit{subject to} \quad c_1 < g(x, y) < c_2 \tag{3.7}$$

3.3.1. Objective Function

In order to find an optimal desk, user satisfaction should be maximized, and user satisfaction is defined as $f(p_i, d_j)$, shown in (3.8):

$$\forall i, j: maximize\ f(p_i, d_j) \tag{3.8}$$

Since four attributes are included in one personal lighting profile, four factors need to be maximized together: $\Delta T(p_i, d_j)$, $\Delta E(p_i, d_j)$, $\Delta C(p_i, d_j)$, and $\Delta L(p_i, d_j)$, shown in (3.9). According to the constrained optimization, the four factors are combined together by different corresponding weights given by the profile.

$$f(p_i, d_j) = \alpha_{Tpi} * \Delta T(p_i, d_j) + \alpha_{Epi} * \Delta E(p_i, d_j) + \alpha_{Cpi} * \Delta C(p_i, d_j) + \alpha_{Lpi} * \Delta L(p_i, d_j) \tag{3.9}$$

, where $\Delta T(p_i, d_j)$, $\Delta E(p_i, d_j)$, $\Delta C(p_i, d_j)$, and $\Delta L(p_i, d_j)$ represent the user satisfaction of each attribute for a pair of profile p_i and desk d_j, meaning that user satisfaction of people's expectation on temperature, illuminance, color temperature and location and the desk's actual corresponding attributes, shown in (3.10), (3.11), (3.12) and (3.13):

$$\Delta T(p_i, d_j) = e^{-\frac{(T_{pi} - T_{dj})^2}{2\sigma_T^2}} \tag{3.10}$$

$$\Delta E(p_i, d_j) = e^{-\frac{(\ln E_{pi} - \ln E_{dj})^2}{2\sigma_E^2}} \tag{3.11}$$

$$\Delta C(p_i, d_j) = e^{-\frac{(C_{pi} - C_{dj})^2}{2\sigma_C^2}} \tag{3.12}$$

$$\Delta L(p_i, d_j) = 1 - |L_{pi} - L_{dj}| \tag{3.13}$$

Noteworthy, according to Fechner's law [20], human eye senses brightness approximately logarithmically over a moderate range, so *ln* is used for E_{pi} and E_{dj} before getting their difference.

Besides, instead of using Gaussian distribution, $\Delta L(p_i, d_j)$ is represented as an absolute value of the difference between L_{pi} and L_{dj}, then minus by 1. Because L_{pi} can be either 1 or 0, and L_{dj} also can either be 1 or 0, we do not model it into Gaussian distribution. $\Delta L(p_i, d_j)$ is used to explain that if L_{pi} and L_{dj} have the same value, which means the actual desk meets profile's location preference, so that $\Delta L(p_i, d_j)$ becomes 1; otherwise, $\Delta L(p_i, d_j)$ is 0.

3.3.2. Constraints

Previously, the objective function of user satisfaction is defined, but for a constrained optimization problem, constraints should also be defined.

First, the four attributes from a personal lighting profile have their certain range, as discussed in Section 2.2:

$$19^oC \leq T_{pi} \leq 27^oC \tag{3.14}$$

$$320lux \leq E_{pi} \leq 500lux \tag{3.15}$$

$$3000K \leq C_{pi} \leq 6500K \tag{3.16}$$

$$L_{pi} = \begin{cases} 0, \text{ not preferred} \\ 1, \text{ preferred} \end{cases} \tag{3.17}$$

Second, the four corresponding weights should subject to that they are summed up to 1, but each of them is within the range of 0 and 1:

$$\alpha_{Tpi} + \alpha_{Epi} + \alpha_{Cpi} + \alpha_{Lpi} = 1 \tag{3.18}$$

$$\alpha_{Tpi}, \alpha_{Epi}, \alpha_{Cpi}, \alpha_{Lpi} \in [0,1] \tag{3.19}$$

Third, the number of profiles and desks, as well as their relationship, are already defined in Section 3.1:

$$p_i, i \in [1,2,3, \dots, m]$$

$$d_j, j \in [1,2,3, \dots, n]$$

$$m > n$$

3.3.3. Weights

As mentioned previously, the weights of four attributes are also fetched from personal lighting profiles, and the algorithm relies on four weights in (3.9). However, humans are not easily able to measure the weights by themselves. Therefore, a smart user interface in the app can be applied to get the four attributes' weights. Inspired by radar chart, users can select four dots along the two diagonals of a square panel, see Figure 3.2. The distance from the selected dot (the green dot in Figure 3.2) to the central point (the red dot in Figure 3.2) is thought as d_T, d_E, d_C, d_L respectively. The app is able to calculate the values of four weights based on these distances, by the formula (3.20), (3.21), (3.22) and (3.23):

$$\alpha_{Tpi} = \frac{d_T}{d_T + d_E + d_C + d_L} \tag{3.20}$$

$$\alpha_{Epi} = \frac{d_E}{d_T + d_E + d_C + d_L} \tag{3.21}$$

$$\alpha_{Cpi} = \frac{d_C}{d_T + d_E + d_C + d_L} \tag{3.22}$$

$$\alpha_{Lpi} = \frac{d_L}{d_T + d_E + d_C + d_L} \tag{3.23}$$

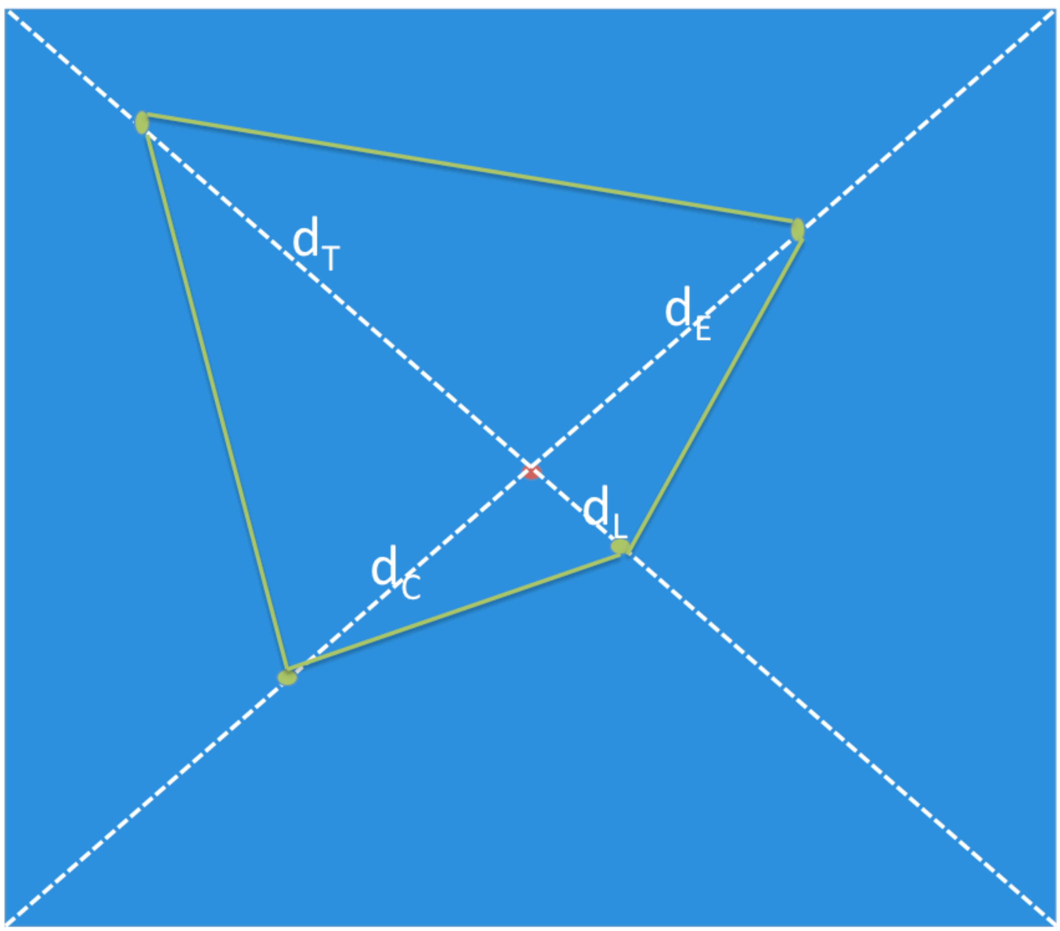


Figure 3.2 The choices of weights

For example, Figure 3.3 illustrates how to set one of the weights as 1, while the rest three weights as 0. Another example in Figure 3.4 shows that a user selects four dots, just to make the four distances to the central point equal, and then the profile generates the four weights as 0.25, 0.25, 0.25 and 0.25 respectively.

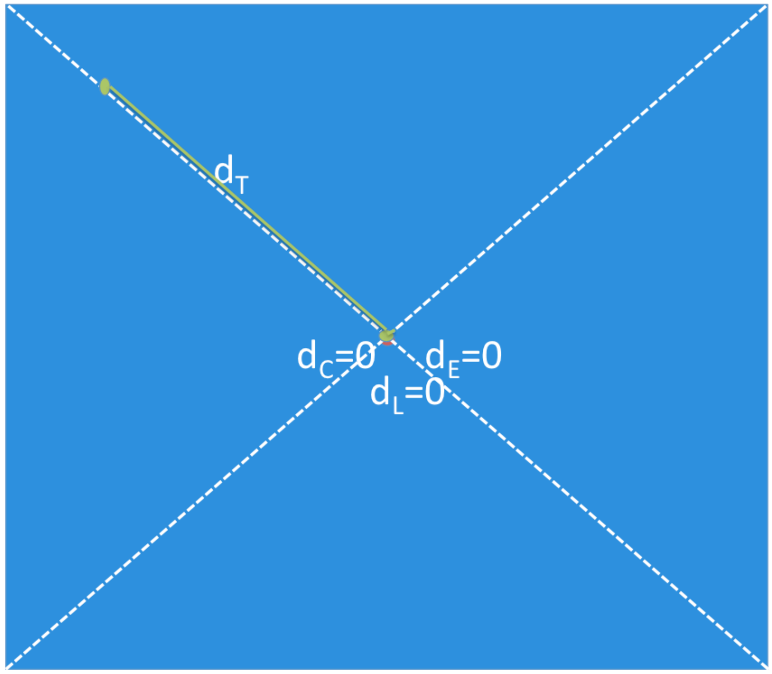


Figure 3.3 An example to set weight of temperature as 1, others as 0

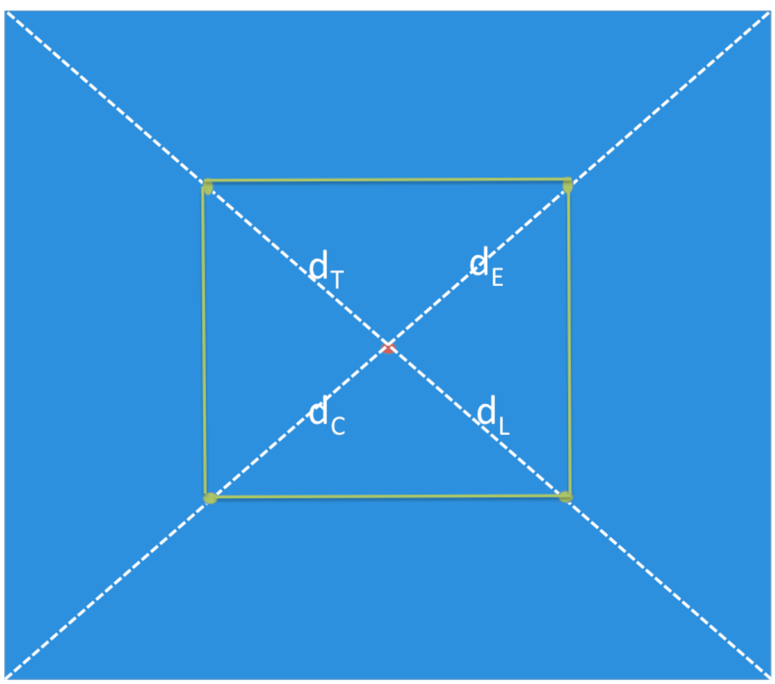


Figure 3.4 An example to set the four weights equally as 0.25

3.3.4. Normalization

As mentioned in Section 3.1, the shape of Gaussian distribution function depends on the value of σ. In order to avoid an extremely small user satisfaction, it is necessary to set a minimum user satisfaction that one attribute could reach. Also, in order to keep the shape of the three Gaussian distribution functions consistent, $\Delta T(p_i, d_j)$, $\Delta E(p_i, d_j)$ and $\Delta C(p_i, d_j)$ should be normalized.

Firstly, according to constraints listed in Section 3.3.2, the maximum difference between expected value of profile and actual value of desk is as follows.

$$T_{pi} - T_{dj} = 27 - 19 = 8 \tag{3.24}$$

$$ln\, E_{pi} - ln\, E_{dj} = ln\, 500 - ln\, 320 = 0.4463 \tag{3.25}$$

$$C_{pi} - C_{dj} = 6500 - 3000 = 3500 \tag{3.26}$$

Secondly, to keep the shape of three Gaussian distribution functions consistent, the minimum user satisfaction of one attribute is set as a constant value, and the three attributes' minimum user satisfaction should be consistent, shown in Figure 3.5.

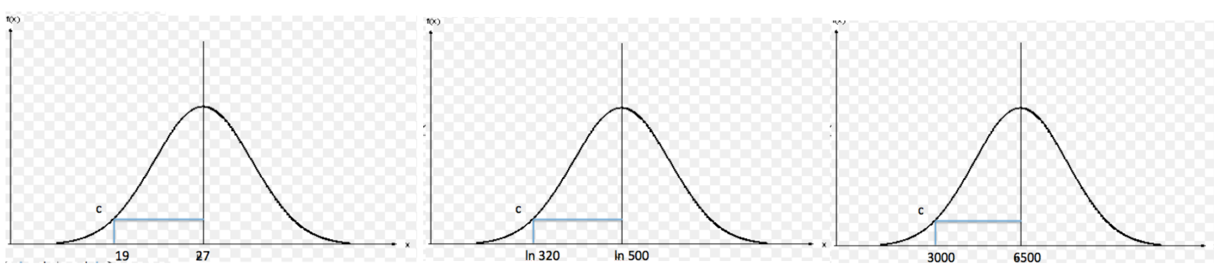


Figure 3.5 Minimum user satisfaction of one attribute is set as a constant value

Thus, the relationship among the three values of σ is as follows.

$$e^{-\frac{8^2}{2\sigma_T{}^2}} = e^{-\frac{0.4463^2}{2\sigma_E{}^2}} = e^{-\frac{3500^2}{2\sigma_C{}^2}} \tag{3.27}$$

$$\frac{17.92}{\sigma_T} = \frac{1}{\sigma_E} = \frac{7842.26}{\sigma_C} \tag{3.28}$$

3.4. Illuminance of Desks

Typically, the illuminance value on desks E_{dj} are got by illuminance sensors (or called brightness sensors). However, to reduce the cost of equipment and to save energy, a method to get the desk's illuminance value E_{dj} based on calculation is used in this formulization. In Section 3.1, there are s luminaires are used for the system, and for each luminaire l_k, I_{lk} is its luminous intensity. The desk's illuminance E_{dj} can be got from all luminaires' effects on this desk, which means to accumulate from all I_{lk}.

A simple model [21] is illustrated in Figure 3.6. Illuminance is calculated by using the basic equation:

$$E_o = \frac{I}{d^2} \tag{3.29}$$

, where E_o is illuminance, I is the intensity directly under the source, and d is the distance from source. This relationship can only be used when surface is directly under the source and normal (perpendicular) to the light ray. However, for all other positions a more generalized formula is:

$$E_i = \frac{I * cos\theta}{d^2} \tag{3.30}$$

, where E_o is illuminance, I is the intensity of the source in the direction toward the point on the illuminated surface, θ is the angle between the line joining the source to the point on the illuminated surface and a line normal (perpendicular) to the illuminated surface. This can also be expressed as the angle between the light ray and a vertical through the center of the source known as the nadir.

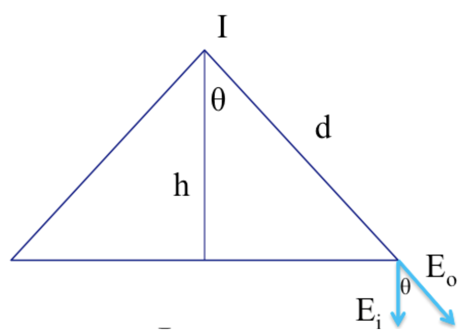


Figure 3.6 A simple model of $E_i = \frac{I * cos\theta}{d^2}$

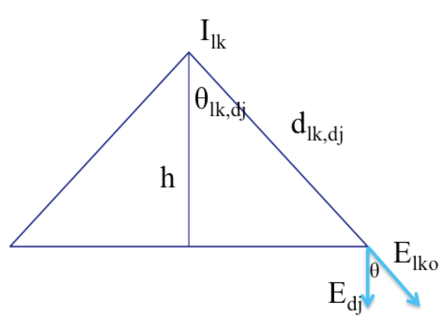


Figure 3.7 The relationship between E_{dj} and I_{lk}

Inspired by the abovementioned simple model, to calculate the desk illuminance E_{dj} from the intensity of all luminaires, (3.31) is used:

$$E_{dj} = \sum_{k=1}^{s} \frac{I_{lk} * cos\theta_{lk,dj}}{{d_{lk,dj}}^2} \tag{3.31}$$

, where E_{dj} is the illuminance of desk d_j, I_{lk} is the luminous intensity of luminaire l_k, $\theta_{lk,dj}$ is the angle between the line joining the source to the point on the illuminated surface and a line normal (perpendicular) to the illuminated surface, and $d_{lk,dj}$ is the distance from source l_k to the desk d_j, see Figure 3.7.

4. Architecture Designs

In this chapter, at first the process of desk assignment and conflict prevention algorithm is explained, and functional and non-functional requirements are proposed. Then, we propose three different system architectures for this connected lighting system. Each architecture is described based on the "4+1 architectural view model" [22], which the views are used to describe the system from the viewpoint of different stakeholders, such as end-users, developers and project managers. Shown in figure 4.1, the four views of the model are logical, development, process and physical view. In addition, selected use cases or scenarios are used to illustrate the architecture serving as the "plus one" view. Finally, the three proposed architectures are compared.

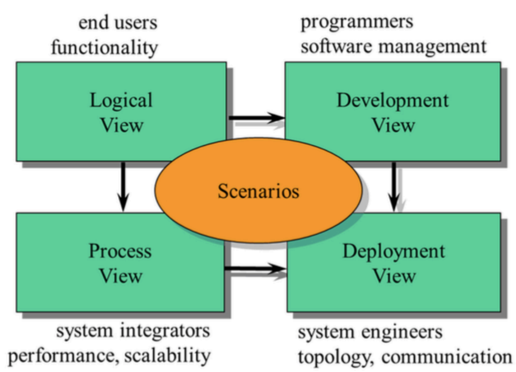


Figure 4.1 Kruchten's "4+1 architectural view model"

4.1. The Process of Conflict Prevention Algorithm

According to Figure 2.4 and the formulization in Chapter 3, the process of conflict prevention algorithm for this connected lighting system is shown in Figure 4.2:

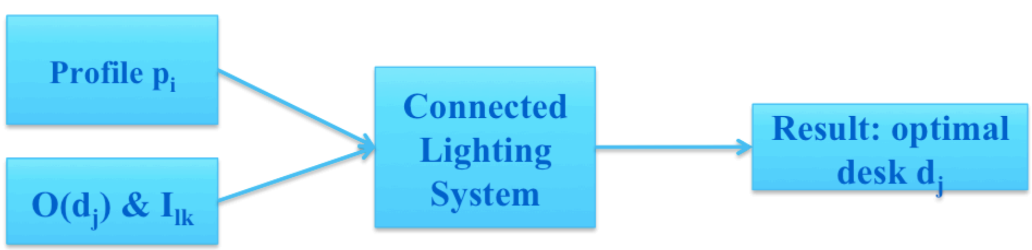


Figure 4.2 The process of conflict prevention

While an office worker with profile p_i comes into this open plan office area, the system extracts T_{pi}, E_{pi}, C_{pi}, L_{pi}, α_{Tpi}, α_{Epi}, α_{Cpi}, α_{Lpi} from the app on the smartphone via a certain user interface. Besides, the system should also get $O(d_j)$ and I_{lk}. $O(d_j)$ is the occupancy information, got from an occupancy detector, which is just above a desk, so it knows whether the desk under is occupied or not. When $O(d_j)$ = 0, the desk is not occupied, but if $O(d_j)$ = 1,it means the desk is already taken. I_{lk} is the luminous intensity of the luminaire, whose value can also be received from luminaire controllers. The system needs luminous intensity value of all luminaire controllers to calculate the illuminance value on the desks, by (3.31). The system then stores and updates the occupancy information and luminous intensity information locally. For each available desk, the system will calculate the user satisfaction for the pair of p_i and d_j, based on formula (3.9). After that, a maximum user satisfaction (*max_user_satisfaction*) is found and the optimal desk number *optimal_j* is known, so the system returns *optimal_j* and *max_user_satisfaction* to the users' smartphones via a certain user interface. The user interface between the system and users' smartphones could be different: via Wi-fi, NFC, Bluetooth, etc., but the choice of user interface is not within the scope of this master project. We assume to apply NFC for user to access the system.

The goal of this algorithm is to prevent conflict for users, and the way is to assign an optimal desk for a single user by maximizing his/her user satisfaction. The conflict prevention algorithm is described by pseudo-code in Algorithm 1:

Algorithm 1. Conflict prevention algorithm

```
Input: p_i
Output: optimal_j & max_user_satisfaction
while input p_i :
    max_user_satisfaction ← 0
    extract ID, T_pi, E_pi, C_pi, L_pi, α_Tpi, α_Epi, α_Cpi, α_Lpi from p_i
    update O(d_j), I_lk from luminaire controllers
    for each j in O(d_j) = 0 do
        f(p_i,d_j)    (see (3.9))
        if f(p_i,d_j) > max_user_satisfaction then
            max_user_satisfaction ← f(p_i,d_j)
    return ID, optimal_j & max_user_satisfaction
```

4.2. Functional Requirements

Functional requirements are usually considered as what the system shall do, which describe the system's behaviors. The proposed connected lighting system shall prevent the potential conflicts for various profiles, as well as perform a satisfying response time:

(1) Conflict prevention: The proposed algorithm prevents the profile conflicts, which are mentioned in Section 2.4. This is the service that the system mainly offers to users, which means it can provide optimal desks for office workers in the open plan office area. Its process can be found in Section 4.1.

(2) Response time: Response time is the period of time that from one person checks into the system with his/her profile, to he/she receives the assigned optimal desk on the smartphone. It contains not only the time that the profile and messages transmit inside the system, but also the time that the system spent on the calculation for the proposed algorithm. Response time is very important for this system because generally people do not want to waste a long time waiting for a response outside the entrance. Maximum endurable response time can be found in a survey (see Appendix A).

4.3. Non-functional Requirements

Non-functional requirements are usually considered as what the system shall be. They describe non-behavioral requirements and indicate how well the system delivers its functionality. So non-functional requirements are often the qualities of a system. For the proposed connected lighting system, four non-functional requirements are given here:

(1) Availability: Availability is the degree to which a system is in a specified operable and committable state at the start of a mission [23]. Simply, availability is the proportion of time that a system is in a functioning condition. Here, for this connected lighting system, the functioning condition should include the basic correct full services like fetching personal light profiles from smartphones, calculating conflict prevention algorithm, and sending message back to the users. Avaiability is often described as a mission capable rate. Mathematically, this is expressed as 100% minus unavailability, or the ratio of the total time a functional unit is capable of being used during a given interval to the length of the interval. More specifically, availability is defined by *MTTF / (MTTF + MTTR)*, where *MTTF* is the mean time to failures and *MTTR* is the mean time to repair. It is usually expected to be a high value, like above 99% within a year.

(2) Scalability: Scalability is the ability of a system to handle a growing amount of work in a capable manner or its ability to be enlarged to accommodate that growth [24]. For the connected lighting system, the scalability has two aspects:

a) Horizontal scaling: It is the ability to add more nodes to a system, usually when nodes are scaled up, functional requirements could not be fully satisfied, so that the bottleneck of the nodes' scaling up should be found. This aspect of scalability mainly depends on the system architecture.
b) Vertical scaling: It is the ability to add resources to a single resource in the system. For example, the maximum number of profiles that the system can operate. It also depends on the chosen system architecture and hardware.

(3) Security: Security is a big issue in computer networks, since many machines are attacked everyday, and many viruses are floating in the Internet. For this connected lighting system, security is in two aspects:

a) System security: From the perspective of the system, all equipment should be protected to avoid invaders, by using some mechanisms like firewalls, etc. Data should also be encrypted during transmission.
b) Profile privacy: Privacy is usually related to security, but from the perspective of human beings. For this system, it should not locally store personal lighting profiles, and the system should not send one's profile to other users. Unauthorized users should not have access to see the internal components of the system.

(4) Fault Tolerance: Fault tolerance is the property that enables a system to continue operating properly in the event of the failure of some of its components [25]. For this connected lighting system, the following aspects can achieve fault tolerance:

a) Anticipating exceptional conditions: All exceptions should be found during architecture design, aiming for self-stabilization so that the system converges towards an error-free state.
b) Duplicating the job in other components (redundancy): If one of the components in the system is down, another component should replace for its job. Therefore, they should duplicate themselves frequently with others.
c) Network is automatically recovered: If interferences happen within the network, the system should have the capability to create a new route for each node so that they are still able to send messages to their destinations.

4.4. Architecture 1: Central Server Style

The first system architecture design uses central server style, it consists of three parts: one server, one room controller, and eighteen desk controllers.

(1) The server is responsible for doing conflict prevention algorithm, collecting and storing occupancy information, updating and storing luminous intensity values, etc. Such a server should be powerful enough, and is considered as already existed, using for identification and verification of office workers.

(2) The room controller, which is the one nearest to the door, is able to read data from the phone reader outside the door, so that it can get profiles from smartphones, and forward them to the server. It gets the result of optimal desk number and the corresponding user satisfaction value, and returns to the smartphone. After that, people can select "Accept" or "Decline" on their smartphones.

(3) Desk controllers are dedicated to desks, so on top of each desk, there is one desk controller hanging on the ceiling, which involves an occupancy detector and LED luminaire. It handles the communication with the server. Besides, it can tell whether there is a human being sitting beneath it or not, by the occupancy detector. All LEDs are configured in a default lighting setting before anybody visits this office.

When people want to enter the open plan office, a smartphone must be validated on a phone reader, attached on the room controller, via a certain user interface, e.g. NFC, to check in, and meanwhile, the phone reader reads the personal lighting profile. After the calculation of conflict prevention algorithm, the result is returned to the user's smartphone, which could even provide a user interface including a map to point out where the desk is located. What is more, the room controller works as a redundancy for the server, which means server backs up its data in the room controller in every certain minute, and if the server is down, the room controller will take over the server role immediately.

This architecture implements a mesh topology, so the server directly communicates with each desk controller as well as the room controller. After the room check-in outside the door, the system will calculate based on the conflict prevention algorithm mentioned in the previous chapter and decide a specific desk for the office worker. He/she receives a message on the smartphone, indicating the desk number, and then he/she can either accept or decline this decision.

If accepted, he/she has to sit in the correct table, and check in again just by putting phone on the phone dock on the desk, then the system will update desk' occupancy information and update luminaire's intensity value. But if he/she does not sit on the correct desk that the system recommends, an error message will be triggered. If declined, the system gives the users their own choice, which means users can manually select a desk: after he/she checks in on a desk, the occupancy information will be updated.

When a person wants to leave permanently, he/she has to check out on the phone dock on the desk. When a person wants to leave temporarily, the desk controller will set a timer for him/her, if the user comes back within 2 hours, then the desk still belongs to him/her; but if the user comes back later than 2 hours, the system considers that the person has left permanently, and this desk will be released again in the occupancy information array stored in the server. For people who want to leave permanently, e.g., get out of office after 6 pm, they have to check out the desk, also via the phone dock on the desk.

The reason why the luminaire's intensity values are updated when a person checks in on a desk is that we use them to calculate the illuminance value on each desk. In case of the system administrators maintain the conflict prevention algorithm, or the luminaires' intensity is re-initialized by system administrators, the algorithm should re-calculate the illuminance value for each desk, based on the luminairs' intensity it gets from all desk controllers. All detailed "4+1" views for this architecture can be found in Figure 4.3 ~ 4.7.

4.4.1. Process View

The process view is using activity diagram (Figure 4.3) to explain the system processes, how they communicate, and the runtime behavior of the system.

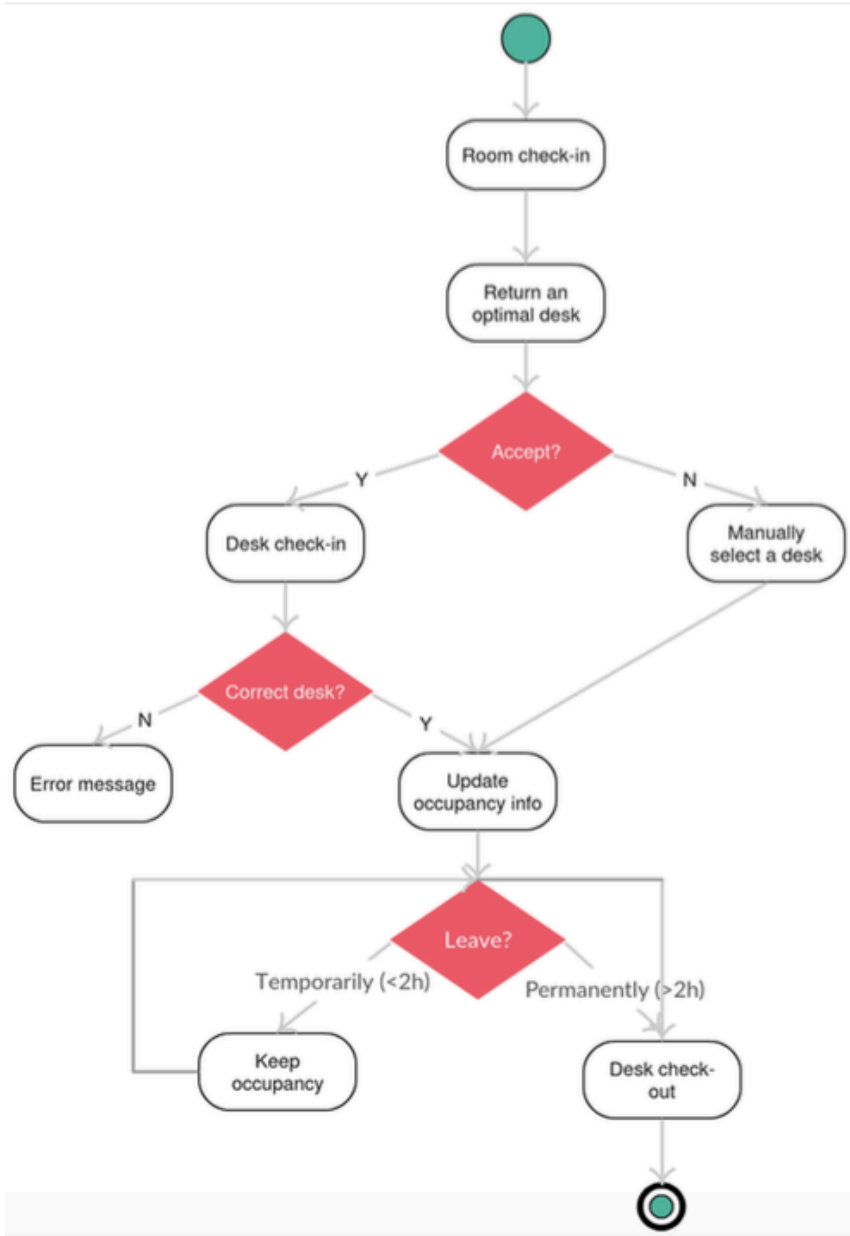


Figure 4.3 Activity diagram of central server style architecture

4.4.2. Logical View

The logical view is concerned with the functionality that the system provides to end-users, using sequence view, shown in Figure 4.4.

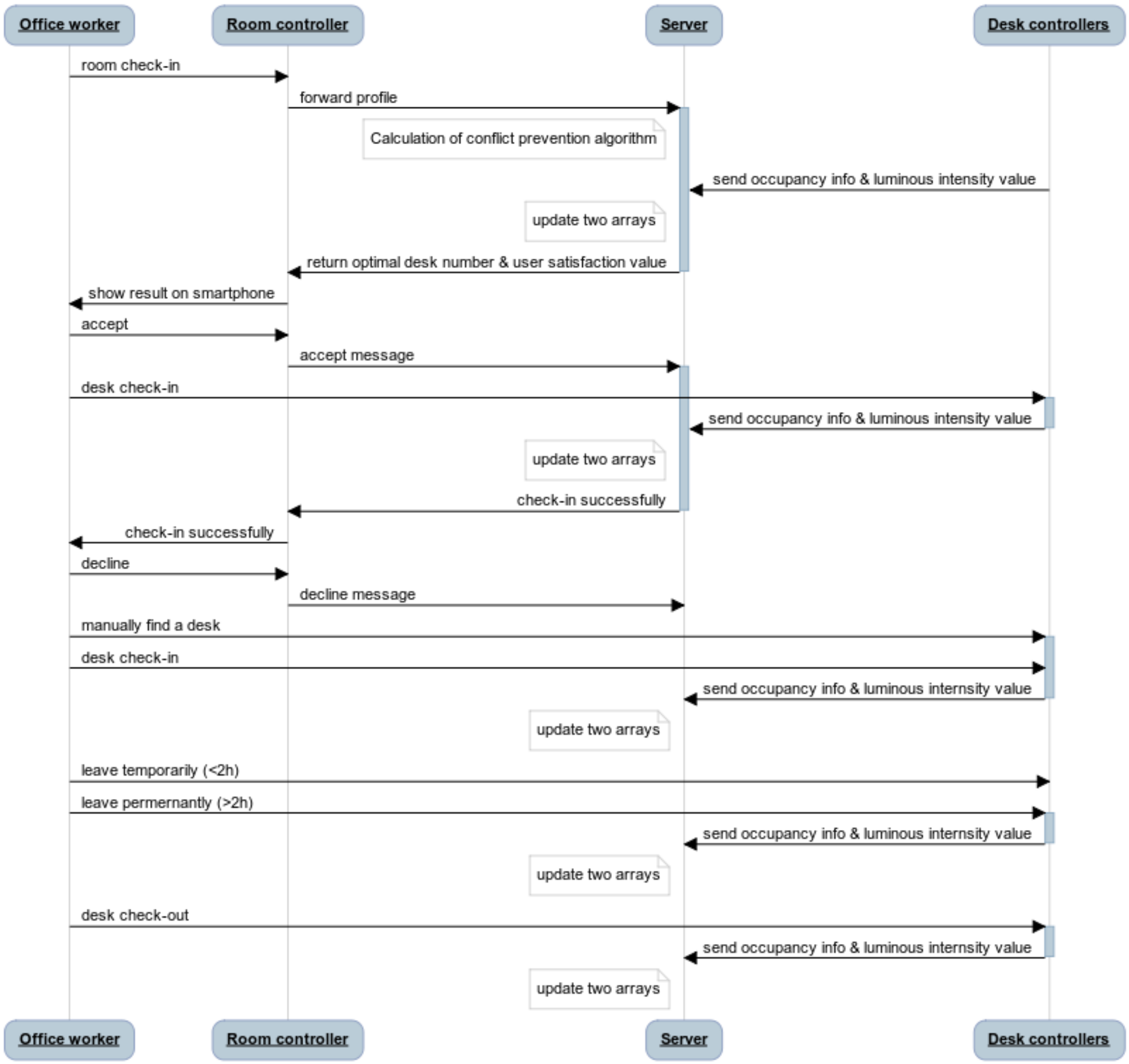


Figure 4.4 Sequence diagram of central server style architecture

4.4.3. Development View

The development view, or implementation view, illustrates a system from a programmer's perspective and is concerned with software management, using component diagram, shown in Figure 4.5.

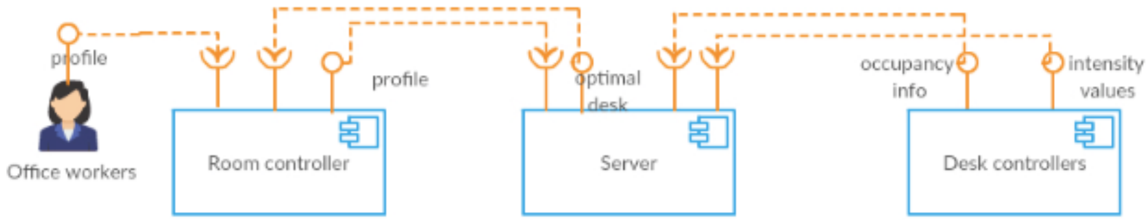


Figure 4.5 Component diagram of central server style architecture

4.4.4. Deployment View

The deployment view, or physical view, depicts the system from a system engineer's point of view. It is concerned with the topology of software components on the physical layer, as well as the physical connections between these components, using a deployment view, shown in Figure 4.6.

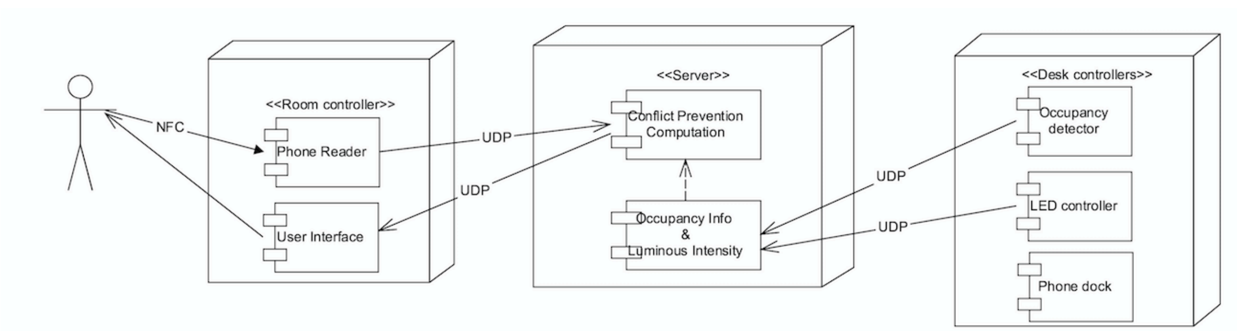


Figure 4.6 Deployment diagram of central server style architecture

4.4.5. Scenarios/Use Cases

The description of an architecture is illustrated using a small set of scenarios, in use cases view, shown in Figure 4.6. The scenarios describe sequences of interactions between objects, and between processes. They are used to identify architectural elements and to illustrate and validate the architecture design.

Except users, system administrators are the other stakeholders, since they are able to manage and maintain the connected lighting system. For example, they are able to initialize the luminous intensity values of each luminaires, update the conflict prevention algorithm, or manage the desk and luminaire layout.

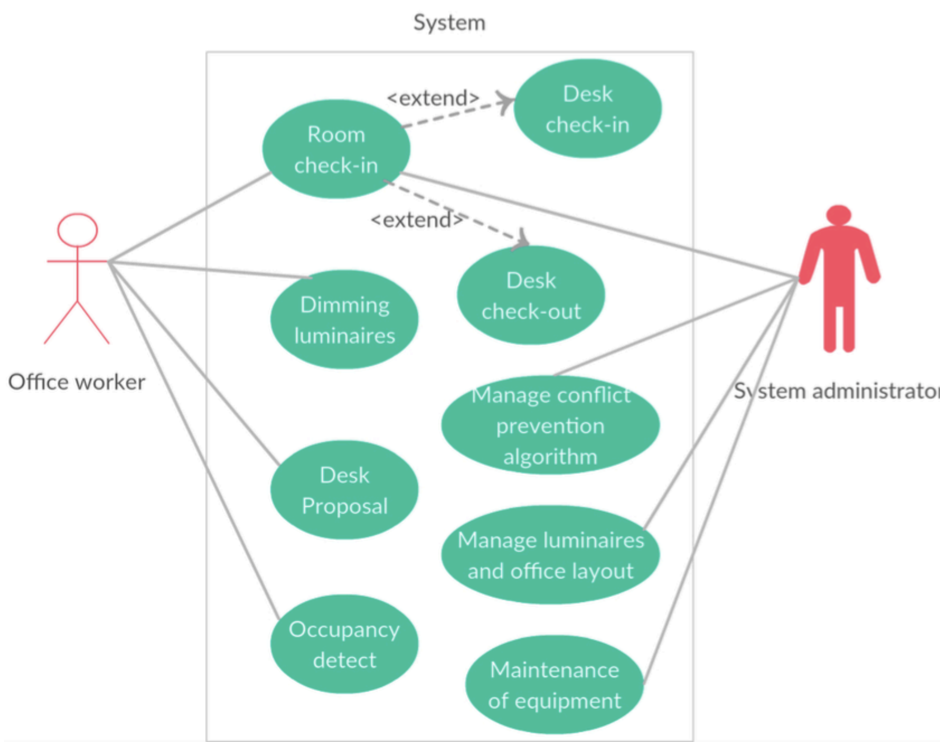


Figure 4.7 Use cases of central server style architecture

4.5. Architecture 2: Distributed Style

The second system architecture design uses distributed style, in which the calculation capability for the conflict prevention algorithm is distributed to each node. This architecture consists of two parts: one room controller, eighteen desk controllers.

(1) The room controller, which is the one nearest to the door, is able to read data from the phone reader outside the door, so that it can get profiles from smartphones, and broadcast them to the eighteen desk controllers. It receives eighteen results of the user satisfaction of the set of a specific desk and the profile, so finally it selects a maximum user satisfaction value among them and sends the result of optimal desk number with its corresponding user satisfaction value to the user, and after this user can select “Accept” or “Decline” on the smartphone.

(2) Desk controllers are dedicated to desks, so on top of each desk, there is one desk controller hanging on the ceiling, which involves an occupancy detector and LED luminaire. It can tell whether there is a human being sitting beneath it or not, by the occupancy detector. It stores locally the value from occupancy detector as a variable and the luminous intensity value of all desk controllers as an array. The luminous intensity value is broadcast to all other desk controllers when user checks in a desk, so that each desk controller knows luminous intensity values of every desk controller. So each desk controller has the capacity to calculate the illuminance on its desk based on the luminous intensity array. All LEDs are configured in a

default lighting setting before anybody visits this office. Furthermore, it gets a profile from the room controller, and then calculates the user satisfaction for that profile with the configuration of its desk. It then sends back the result with user satisfaction value to the room controller. Regarding redundancy, desk controllers can replace room controller. However, as a desk controller includes the LED controller, it cannot replace the desk controller's LED control function. So no full redundancy is provided for the desk controllers.

In terms of the "4+1" views, the process view and use cases are as same as the first system architecture (central server style), so process view of this architecture is shown in Figure 4.3, and Use cases of this architecture are shown in Figure 4.7. Only logical view, development view and deployment view are re-drawn for this architecture, shown in Figure 4.8, Figure 4.9 and Figure 4.10 respectively.

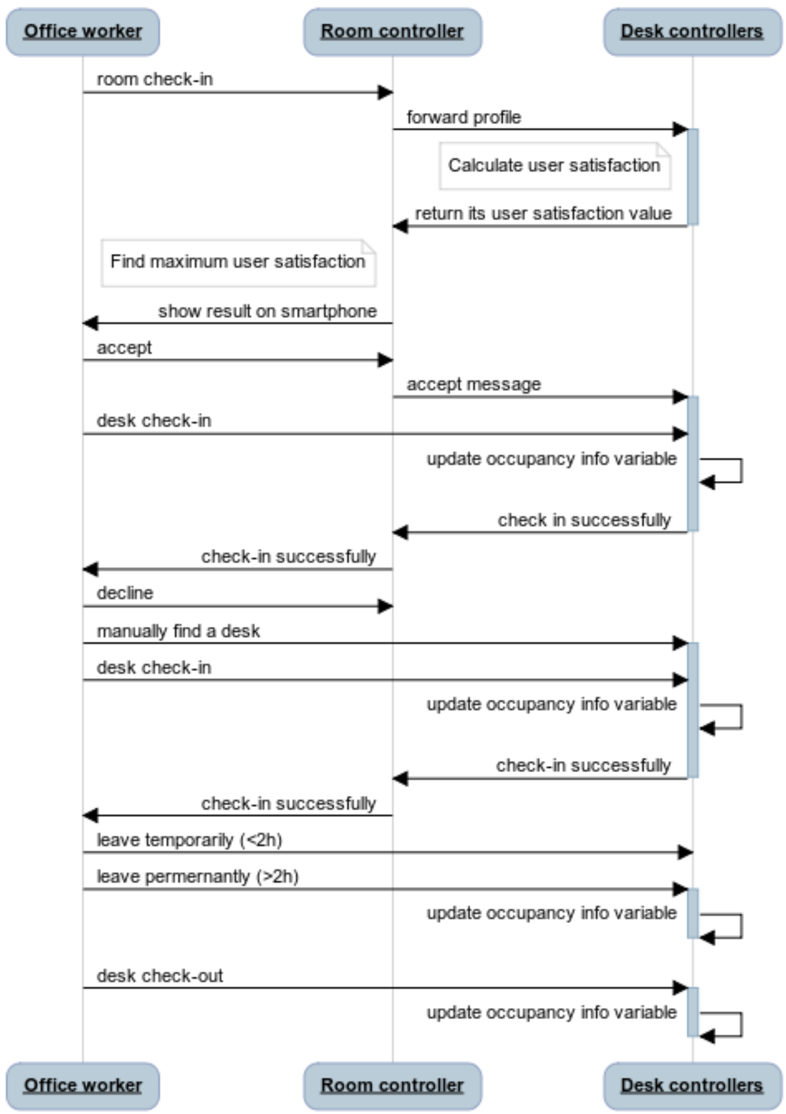


Figure 4.8 Sequence diagram of distributed style architecture

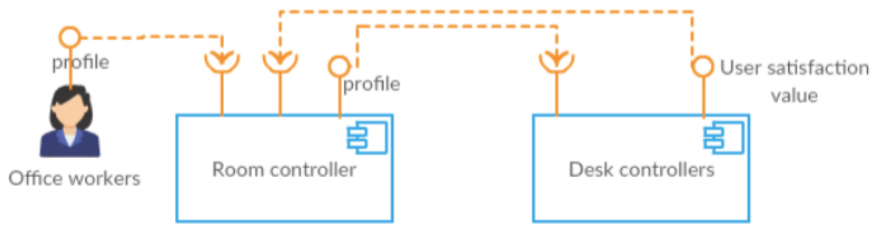


Figure 4.9 Component diagram of distributed style architecture

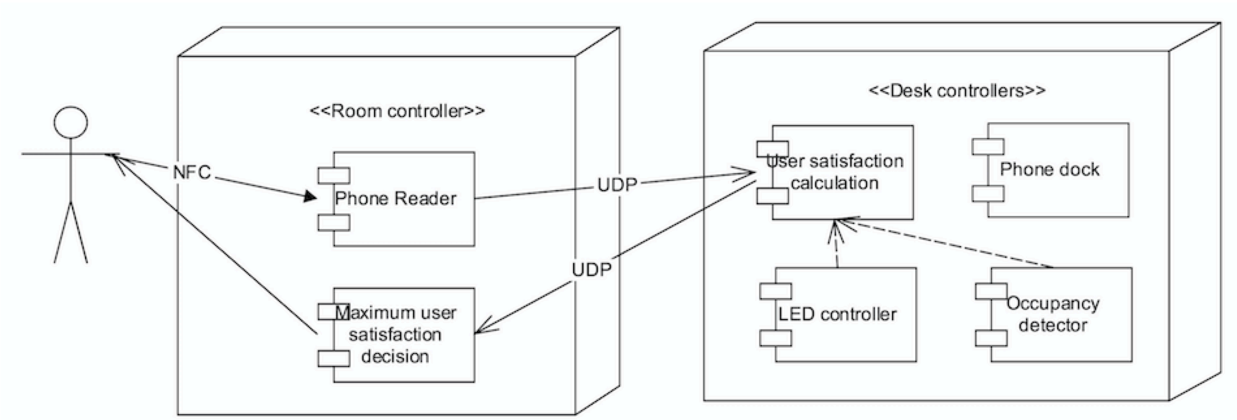


Figure 4.10 Deployment diagram of distributed style architecture

4.6. Architecture 3: Ring Style

The third system architecture design uses ring style, and it also consists of two parts: one room controller and eighteen desk controllers.

(1) The room controller, which is the one nearest to the door, is able to read data from the phone reader outside the door, so that it can get profiles from smartphones, and forward them to the first desk controller in the ring network. It will receive a final result of the optimal desk number and the corresponding value of biggest user satisfaction, so it sends the results back to the smarphone, and after that this user can select “Accept” or “Decline”.

(2) Desk controllers are dedicated to desks, so on top of each desk, there is one desk controller hanging on the ceiling, which involves an occupancy detector and LED luminaire. It is able to tell whether there is a human being sitting beneath it or not by the occupancy detector. All LEDs are configured in a default lighting setting before anybody visited this office. What is more, all desk controllers are linked with each other in a logical ring network. The ring network is created according to their IP addresses. Each desk controller stores locally the occupancy info retrieved from the occupancy detector, and stores the luminous intensity values of all desk controllers as an array. Each desk controller has the capacity to calculate the illuminance on its desk based on the luminous intensity array. The first desk controller gets a profile from the room controller, and it will calculate the user satisfaction for that profile and the configuration of its desk. It will send

its desk number, the result value of user satisfaction, the profile and the array of luminous intensity values to the subsequent controller. Then the subsequent one firstly updates its luminous intensity array, and then calculates its own user satisfaction based on the received profile, and the result is compared with the received result of the preceding controller. It keeps the bigger value of user satisfaction, and sends it with its corresponding desk number and the profile to the next controller, also the array of luminous intensity values is sent. Therefore, the profile forwards along the ring, in the mean time, the calculation is done in each desk controller, and the decision is made by the comparison of neighboring nodes. The last desk controller returns the biggest value of user satisfaction and its corresponding desk number to the room controller. Regarding redundancy, desk controllers can replace room controller. However, as a desk controller includes the LED controller, it cannot replace the desk controller's LED control function. So no full redundancy is provided for the desk controllers.

In terms of the “4+1” views, the process view and use cases are the same as the first system architecture (central server style), so process view of this architecture is shown in Figure 4.3, and Use cases of this architecture is shown in Figure 4.7. Only logical view, development view and deployment view are redrawn for this architecture, shown in Figure 4.11, Figure 4.12 and Figure 4.13 respectively. The following figures differ desk controllers: first desk controller, middle desk controllers, and last desk controller. For the middle desk controllers, we use one of them to represent all of them since they have the same functionality. Actually the middle desk controllers should be linked with each other. Figure 4.11 also assumes that the user’s actions take place in one of the middle desk controllers, where actions could include desk check-in/check-out, manually select a desk, etc.

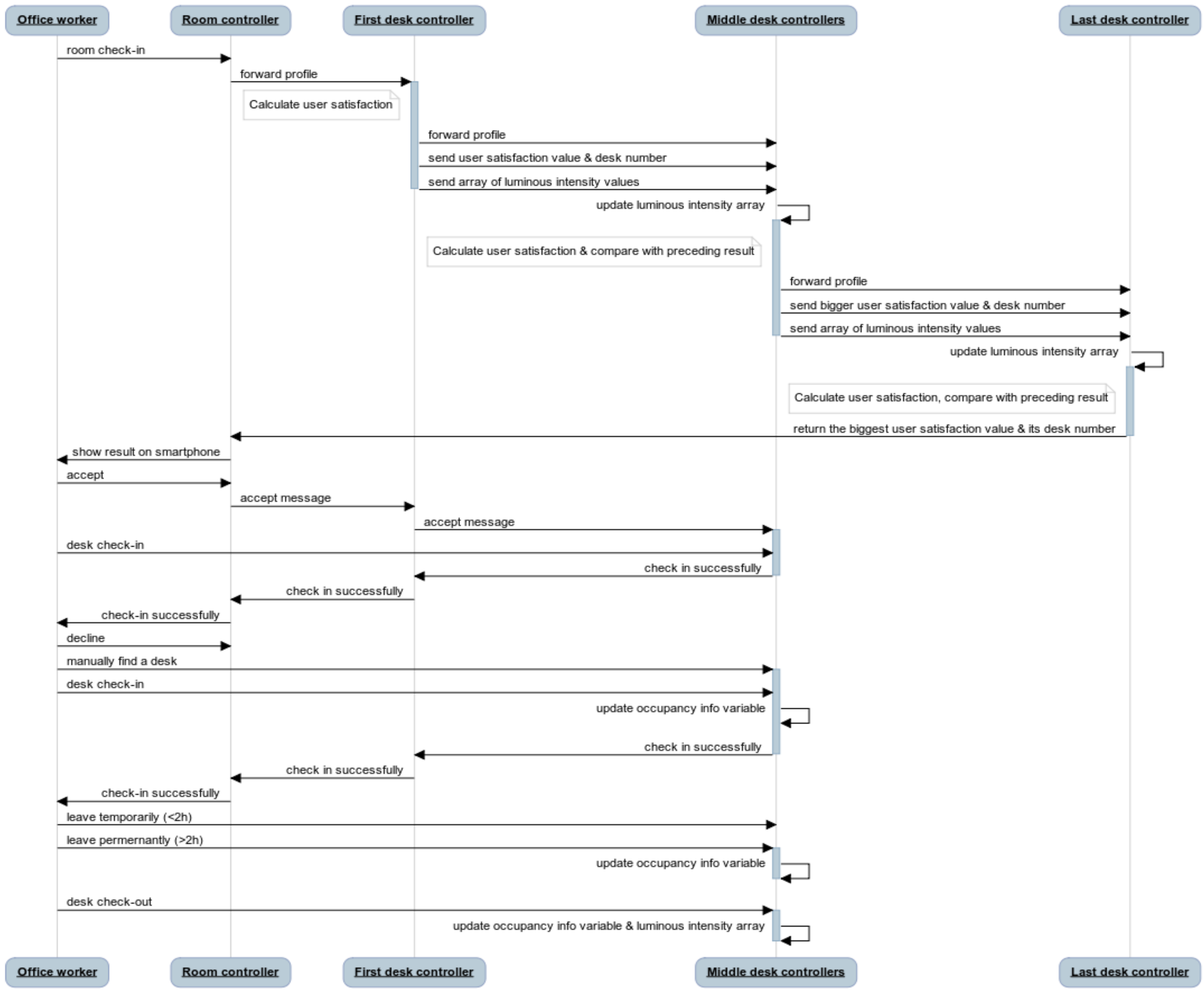


Figure 4.11 Sequence diagram of ring style architecture

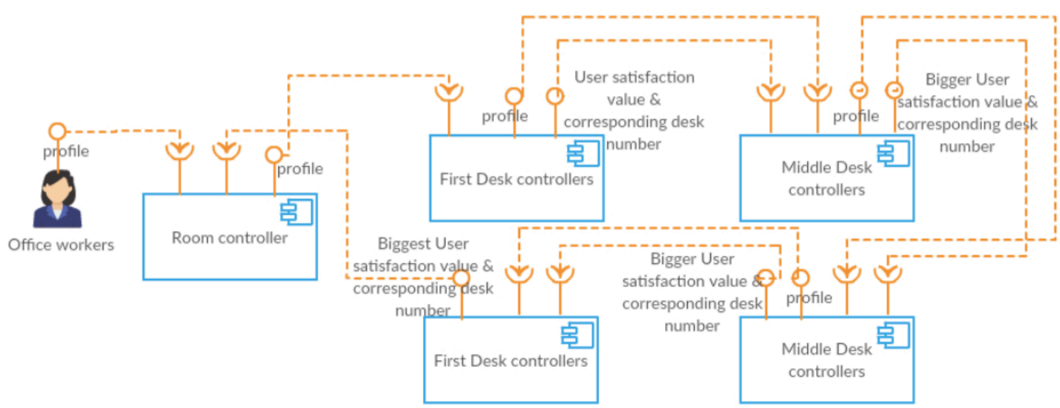


Figure 4.12 Component diagram of ring style architecture

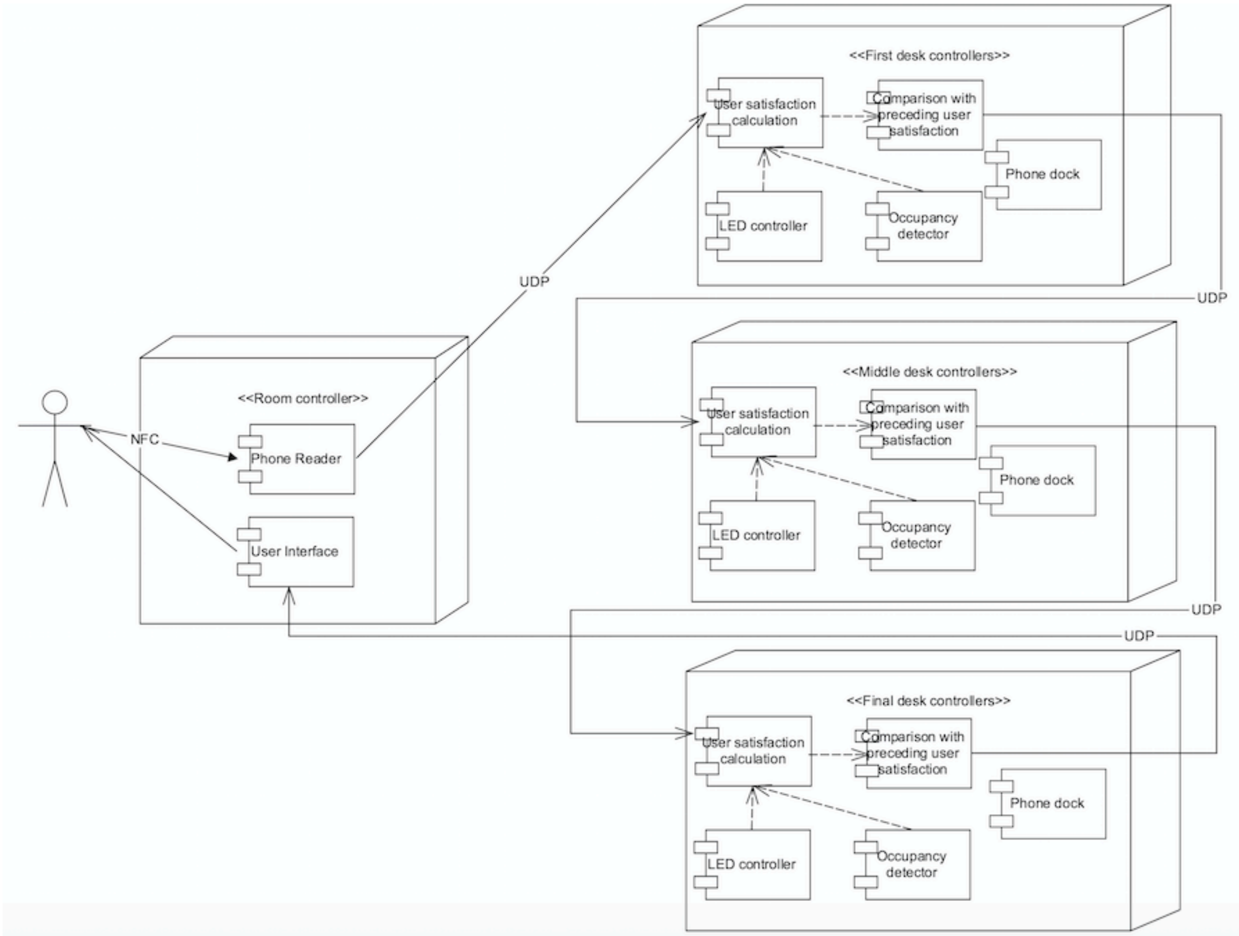


Figure 4.13 Deployment diagram of ring style architecture

4.7. Comparison and Conclusions

In conclusion, the three proposed architectures use different mechanisms to do the conflict prevention algorithm in the network. Central server style is the only one who has a server, where it updates occupancy information and intensity values locally, and calculates the conflict prevention algorithm for the coming profile, so that the profile will not be distributed in the network. After its calculation, the decision of an optimal desk is given to the user. For the distributed style, occupancy information and intensity values are stored locally in the desk controllers, because of no server. Profiles are broadcast to each desk controller, where user satisfaction will be calculated, but the room controller makes the final decision of the optimal desk since it can collect all the user satisfaction results from all the desk controllers. For the ring style architecture, a logical ring is initially created for all the desk controllers, but there is no server. A profile is distributed along the ring network, each node calculates the user satisfaction based on this profile and compare the results with the preceding result, after that, it keeps the bigger user satisfaction, and forwards it to the next node. The calculation is done node by node, and the comparison with neighboring node makes the decision of an optimal desk. A comparison of the three above-mentioned system architecture is made in Table 4.1. The pros and cons of them will be discussed in Chapter 5, in terms of some criteria, e.g. response time, scalability, etc.

Table 4.1 Comparison of the thee proposed architectures

	Central server style	**Distributed style**	**Ring style**
Server	Yes	No	No
Topology	Mesh Topology	Distributed Topology	Ring Topology
Update Occupancy Info & Intensity Info	Store occupancy info array and intensity array in the server	Occupancy variable and intensity variable in each desk controller	Occupancy variable and intensity variable in each desk controller
Profile Distribution	Stay in the server	Distributed to each desk controller	Distributed along the ring network
Redundancy	Room controller works as a redundancy of server, desk controllers are redundancy for each other	Desk controllers can replace room controller, but no fully redundancy for themselves	Desk controllers can replace room controller, but no fully redundancy for themselves
Calculation of Conflict Prevention Algorithm	Calculated in the server	Calculated in a distributed fashion, the results of each user satisfaction is collected by room controller	Calculation is done along the ring: compare its user satisfaction with the previous one and keep the bigger one
Decision Making	Decision made in the server: the desk with the maximum user satisfaction	Room controller decides the final desk number according to the results of user satisfaction collected from all desk controllers	Decision made along the ring: comparison of the user satisfaction between the neighboring two nodes

5. Architecture Decision

In Chapter 4, three different kinds of architectures are proposed for the connected lighting system, then a question comes out: "which of them is the best architecture for this system?" In order to select one suitable architecture from the three, several criteria are raised: equipment cost, response time, user capacity, scalability, and availability. Some of the criteria can be known from the architectural style, but some rely on what facilities and what network performance the architecture uses. Before measuring the criteria, we make some assumptions about the facilities and network performance. Finally, a comparison about those criteria among the three architectures lead to a final decision on which architecture to be used for this system, and the selected architecture will be simulated in Chapter 6.

5.1. Assumptions for Architecture Decision

Some of the criteria among the three proposed architectures, e.g., response time and user capacity, are depended on the specific facilities and the network performance. In order to make a comparison for those criteria, we have to give some assumptions for the facilities and network performance, and keep the three proposed architecture use the same configurations, then some of the criteria based on those assumptions can easily be measured in Section 5.2.

5.1.1. Facilities

Each of the three proposed architectures contains one room controller, and eighteen desk controllers. To make them consistent, we assume that all the nineteen controllers are using the same MCU (Microcontroller Unit) and the same configurations. ARM Cortex-M3 processor [26] is a typical and industry-leading 32-bit processor used for real-time applications, and high-performance low-cost platforms for a broad range of devices including MCU. We assume using it for all the controllers in the three architectures, and we assume that the operational frequency of them is 100 MHz. The parameter that we care about for the criteria are listed in Table 5.1. Because the parameter given is in a range, we fix a specific value for the parameter in the last column.

Table 5.1 ARM Cortex-M3 specifications

Parameter	Value in range	Fixed value
Dhrystone performance	1.25 / 1.50 / 1.89 DMIPS/MHz	1.50 DMIPS/MHz

We should also assume how many instructions the processor will run for computation of the conflict prevention algorithm. Therefore we assume that 1 million instructions after compilation for one user satisfaction computation of one “profile-desk” set, using the formula in (3.9), and 0.1 million instructions for the comparison computation to get the maximum user satisfaction.

In addition, the first proposed architecture contains one server, as described in Section 4.4, we assume that a powerful server is used. We also assume the server is in a “five-nines” standard [27], which means it is 99.999% availability, or 5.26 minutes downtime in a whole year. In terms of availability of microcontrollers, we assume that the fault comes once in one single day, and the recovery time for the above-mentioned MCU is 5 minutes.

5.1.2. Network Performance

The MCUs construct a wireless sensor network, and we assume that it uses Zigbee protocol, which is based on IEEE 802.15.4 standard, and usually used for small, low-power digital radios. E. D. Pinedo-Frausto, et al [28] mentions that “As message sizes can vary from 25 to 128 bytes, …, Our tests show that minimum-sized messages can be safely sent at 40 ms rates, but for maximum-sized messages the minimum sent rate is 50 ms. ” Therefore, we assume that the size of a personal lighting profile is 128 bytes, and can be sent at 50 ms per hop; the size of an occupancy information and luminous information message is assumed as 32 bytes, so it can be sent at 40 ms per hop. Since there are eighteen nodes totally in the network of the open plan office (see Figure 2.2), and the distance between two neighboring nodes are about 1.8 – 2.4 m. While E. D. Pinedo-Frausto, et al [28] also mentions that “Panasonic’s board we observed losses of up to 10% at 15 meters but of 80% at 20 meters. For Freescale’s 13192-EVB we had better results without any losses at 20 meters”, we assume that for the used MCU boards, they are at 10 meters without any losses. Therefore, we consider that transmission in the wireless sensor network has 1.5 hops in average. Thus, the hop-to-hop transmission time of one profile is 50 ms, and the average desk-controller-to-server transmission time of one profile is 80 ms. The hop-to-hop transmission time of one occupancy and intensity info message is 40 ms, and the average desk-controller-to-server transmission time for that message is 60 ms. What is more, we assume that the length of the computation result of user satisfaction of one “profile-desk” set is the same as the length of one occupancy and intensity info message. The assumed transmission time is concluded in Table 5.2.

Table 5.2 Assumed transmission time

Message	**Hop-to-hop transmission time**	**Desk-controller-to-server transmission time**
Personal lighting profile	50 ms	75 ms
Occupancy & luminous intensity info	40 ms	60 ms
Result of user satisfaction of one “profile-desk” set	40 ms	60 ms

Other components, like occupancy detectors, luminaires, are not considered into the measurement of the criteria, because they are not related to the transmission and computation during room check-in.

5.2. Criteria

As described in Chapter 4, the functional requirements of this system are response time and conflict prevention. The response time depends on what kind of architecture is used. User capacity is related to response time and the architectural style. Therefore, response time and user capacity are two key criteria for selecting architecture. What is more, the non-functional requirements of this system are availability, scalability, security and fault tolerance. Availability and scalability can be known by the assumptions of the system architecture, but security and fault tolerance are not easily measured, so that we put availability and scalability into the criteria of architecture decision. Besides, various architectures use different equipment so they cost differently. The costs of equipment are easy to compare, so we also have this criterion. All in all, five criteria will be measued and compared for the three proposed architectures, they are: equipment cost, response time, user capacity, scalability and availability.

5.2.1. Equipment Cost

Equipment cost is the cost that spent for purchasing the equipment using for one system. As described in Chapter 4, the first architecture contains one server, one room controller and 18 desk controllers for desks; both the second and third architectures involve one room controller and 18 desk controllers, without a server. As we assume, those room controller and desk controllers will use the same hardware and same configurations, so the only difference for the three architectures is whether they have a server or not. Obviously, the first architecture has a server, so that its equipment cost is relatively higher than the other two architectures. However, the server used in this architecture can be considered as already existed in the infrastructure of the office building, which is already used for identification and verification of office workers. So we can take a part of the server on lease. In this case, the cost of a server is considered as a little higher since we just need to pay for the lease. Therefore, the equipment cost of the three proposed system architectures are listed in Table 5.3.

Table 5.3 Equipment cost of the three proposed architectures

	Central server style	Distributed style	Ring style
Equipment cost	Higher	Average	Average

5.2.2. Response Time

Response time is the time from the moment that the office worker checks in at the room controller to the moment that he/she gets an optimal desk result on the smartphone. For the first architecture (central server style), the total response time includes:

(1) The transmission time of one profile from the room controller to the server: 50 ms

(2) The computation time of the algorithm on the server can be neglected because of the powerful server, but here we assume it as 1 ms.

(3) The transmission time of the result from the server to the room controller: 40 ms.

Here, the transmission between server and room controller is considered as one-hop because we put them close to each other. Therefore, the total response time of the first architecture is 91ms.

However, the time of updating occupancy information and luminous intensity values is not included in the response time, which is considered as a desk-controller-to-server delay between the desk controllers and server. The desk-controller-to-server delay should be tested in simulation, because it is related to scalability.

For the second architecture (distributed style), the total response time involves:

(1) The total transmission time of one profile from the room controller to one desk controller is $T_1 = 75\ ms$, because this transmission is thought as 1.5 hops in average.

(2) The computation time of one user satisfaction of one "profile-desk" set in the desk controller:

$$T_2 = 1\ M\ Instructions / \left(1.5 \frac{DMIPS}{MHz} * 100\ MHz\right) = 6.67\ ms \quad (5.1)$$

(3) The total transmission time of one result user satisfaction from one desk controller to the room controller is $T_3 = 60\ ms$, which is also 1.5 hops in average.

(4) The computation time of maximum comparison in the room controller:

$$T_4 = 0.1\ M\ Instructions / \left(1.5 \frac{DMIPS}{MHz} * 100\ MHz\right) = 0.667\ ms \quad (5.2)$$

However, after the first profile is transmitted from server to the first desk controller, the profile's transmission from server to the second desk controller happens simultaneously as the computation in the first desk controller. Similarly, the transmission of the first user satisfaction result happens simultaneously as the profile transmission from the room controller to the second desk controller. The workflow of this architecture is shown in Figure 5.1, in terms of time.

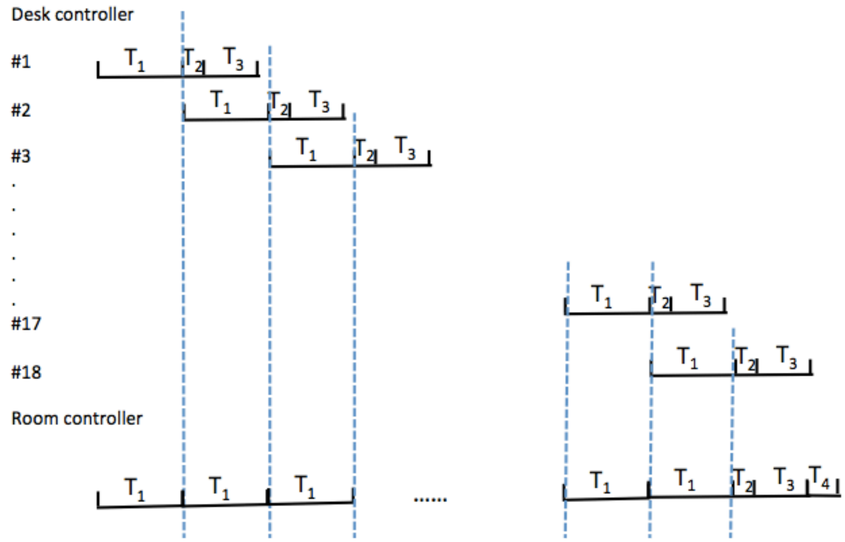


Figure 5.1 The workflow of distributed style architecture in terms of time

Therefore, from Figure 5.1, we can calculate the total response time of the second architecture based on eighteen profile transmission time plus the last-time computation time in one desk controller plus the last-time result transmission time plus the comparison's computation time in the room controller:

$$18 * T_1 + T_2 + T_3 + T_4 = 1417.33\ ms \tag{5.3}$$

For the third architecture (ring style), the total response time includes:

(1) The transmission time of one profile from the room controller to the first desk controller is $T_5 = 50\ ms$, which is thought as one-hop transmission.

(2) The computation time of one "profile-desk" set from one middle desk controller is:

$$T_6 = 1\ M\ Instructions / \left(1.5 \frac{DMIPS}{MHz} * 100\ MHz\right) = 6.67\ ms \tag{5.4}$$

(3) The computation time of one user satisfaction comparison with the previous result in one middle desk controller is:

$$T_7 = 0.1\ M\ Instructions / \left(1.5 \frac{DMIPS}{MHz} * 100\ MHz\right) = 0.667\ ms \tag{5.5}$$

(4) Because of the logical ring network, the transmission along the ring is assumed to use desk-controller-to-server transmission time in Table 5.3. The transmission time of one profile and one user satisfaction result from one middle desk controller to the next neighboring one:

$$T_8 = 75\ ms + 60\ ms = 135\ ms \tag{5.6}$$

(5) The transmission time of the result from the last desk controller to the room controller is $T_9 = 40\ ms$.

Therefore, the total response time of the third architecture can be calculated based on $T_5 \sim T_9$. There are 16 middle desk controllers totally, so T_5 is calculated once only for the first desk controller, T_6 should be calculated 18 times for all the desk controllers, T_7 should be calculated 17 times, except the first desk controller, T_8 should be calculated 17 times for the first desk controller and the16 middle desk controllers, and T_9 should be calculated once for the last desk controller. Thus, in total, the response time of the third architecture is:

$$T_5 + 18 * T_6 + 17 * T_7 + 17 * T_8 + T_9 = 2516.4\ ms \tag{5.7}$$

The response time of the three proposed architectures is listed and compared in Table 5.4.

Table 5.4 Response time of the three proposed architectures

	Central server style	Distributed style	Ring style
Response time	91 ms	1417.33 ms	2516.4 ms

5.2.3. User Capacity

User capacity is the attribute to describe how many office workers can enter the room every second. If there are many people waiting outsides to enter the office, people have to wait until the previous person gets a desk from the system, which means the users are coming one by one, not simultaneously. Therefore, the user capacity is the reciprocal of the response time, and they are listed in Table 5.5.

Table 5.5 User capacity of the three proposed architectures

	Central server style	Distributed style	Ring style
User capacity	11 people	0.7 people	0.4 people

5.2.4. Scalability

As explained in Section 4.3, scalability is the ability of a system to handle a growing amount of work in a capable manner or its ability to be enlarged to accommodate that growth. It includes horizontal and vertical scalability. Horizontal scalability is related to the nodes, which is the ability of this system to scale up or down the nodes. Vertical scalability is related to profiles, but the number of profiles in this system is limited because the office room can only contain 18 people. If more people want to enter this room, they should wait until at least one desk is available. Here, vertical scalability also depends on horizontal scalability, which means if more desks are added into the office room, more people (profiles) can be assigned to a desk.

Therefore, we only consider horizontal scalability. For the first (central Server style) and the second architecture (distributed style), if more nodes are added into this system, the system should create new routes for each node, and it costs much effort. However, scaling up is easier for the third architecture (ring style) as the node just needs to release the connection of its predecessor and successor, and then create a new route with its predecessor and successor respectively. Compared with the first and second architecture, it does not create new routes for all nodes, but only affect two neighboring nodes, so the third architecture is considered as easier for scalability. Scalability is concluded in Table 5.6.

Table 5.6 Scalability of the three proposed architectures

	Central server style	Distributed style	Ring style
Scalability	Average	Average	Better

5.2.5. Availability

In Section 5.1.1, we assume that the server is considered as 99.999% availability, or 5.26 minutes downtime in a whole year, while the fault comes once in one single day for the microcontrollers, and the recovery time for the controllers is 5 minutes. According to those assumptions, we calculate the availability of the system based on fully availability, which means if any component of the system is down, the time is counted as unavailability. Mathematically, the availability is expressed as 100% minus unavailability, for the first architecture (central server style), the unavailability is:

$$(5.26\, min\ + 5\, min * 365\, times)/365\, days = 0.4473\% \tag{5.8}$$

Thus, the availability for the first architecture is 1-0.4473% = 99.5527%

Since the second and third architectures do not have a server, so the unavailability for them is:

$$5\, min * 365\, times/365\, days = 0.3472\% \tag{5.9}$$

Thus, the availability for the second and third architecture is 1-0.3472%=99.6528%

In conclusion, the availability of the three proposed architectures are listed in Table 5.7.

Table 5.7 Availability of the three proposed architectures

	Central server style	Distributed style	Ring style
Availability	99.5527%	99.6528%	99.6528%

5.3. Comparison and Conclusions

Section 5.2 measures five criteria based on some assumptions made in Section 5.1. In this section, we decide which architecture is the best one for this connected lighting system. We combine all the criteria into Table 5.8 so that we can see the comparisons obviously.

As the results, the equipment cost of central server style architecture is a little higher than the rest two, but the central server style has a better performance in response time and user capacity. Although ring style architecture has the best scalability, it is terrible in response time and user capacity. For the availability, central server style architecture is just a bit lower than the rest two. All in all, according to the comparison of five criteria, central server style architecture is considered as the best architecture from the three, so it is selected and simulated in the next chapter.

Table 5.8 The comparisons of six criteria of the three proposed architectures

	Central server style	Distributed style	Ring style
Equipment cost	Higher	Average	Average
Response time	91 ms	1417.33 ms	2516.4 ms
User capacity	11 people	0.7 people	0.4 people
Scalability	Average	Average	Better
Availability	99.5527%	99.6528%	99.6528%

6. Simulation

Chapter 5 decides to use central server style architecture for this connected lighting system. This chapter introduces the simulation, mainly including which tools are used for the system simulation, how to setup the simulated system, and how the system works.

6.1. Simulation Tool

In order to simulate a wireless sensor network, many simulators are available online, for instance, NS2, NS3, OPNET, Cooja, etc. We choose to use Cooja, provided by Contiki OS [29]. Contiki OS is an open source operating system for IoT (Internet of Things), and it connects tiny low-cost, low-power microcontrollers to the Internet. It has many good features:

(1) Contiki provides powerful low-power Internet communication standards, supports fully standard IPv6 and IPv4, along with the recent low-power wireless standards: 6LoWPAN, RPL, CoAP.

(2) With Contiki, development is easy and fast. Contiki applications are written in standard C, with the Cooja simulator, Contiki networks can be emulated before burned into hardware.

(3) Contiki runs on a range of low-power wireless devices, many of which can be easily purchased online.

(4) Contiki is developed by a worldwide team of developers, and it has an active community, which means developers can get help easily from the other developers online.

(5) Contiki is open source software, so it is freely used both in commercial and non-commercial systems and the full source code is available.

(6) It provides many useful examples as well as tools, which can save developers' time.

The Cooja network emulator is the simulation environment provided by Contiki OS. It is an extensible Java-based simulator capable of emulating Tmote Sky and other nodes. Devices often make up large wireless networks, but developing and debugging for such networks are hard. However, Cooja makes this tremendously easier that it allows developers to see their applications run in large-scale networks and in extreme details on fully emulated hardware devices.

6.2. Simulation Setup

6.2.1. Network

The simulated system runs on Contiki OS and uses the emulator, Cooja, to make up a wireless sensor network. For our simulated system, Contiki OS is in version 2.7.

As designed in Section 4.4, the whole system includes three parts: one server node, one room controller node, and eighteen desk controller nodes. In Figure 6.1, Node 1 is the server node, Node 2 is the room controller node, and Node 3~20 are the desk controller nodes. The layout of these nodes is applied according to the plan view of the office, which is already defined in Figure 2.2. The distances between the nodes are exactly the same as the configuration in Figure 2.2, and a 10m-background grid is shown in this figure.

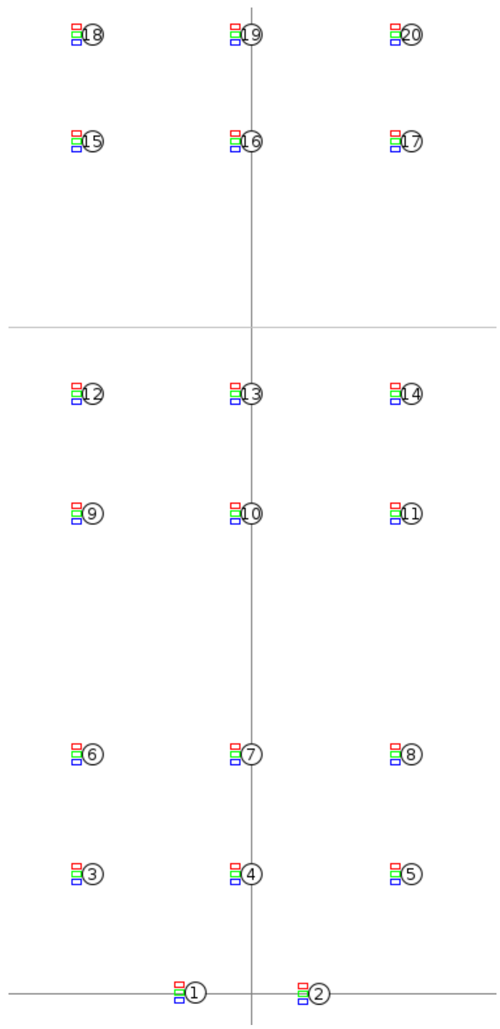


Figure 6.1 Layout of nodes in the simulation

Figure 6.2 [30] shows a Contiki Network Stack. For this simulated system, the physical layer uses CC2420 transceiver, which is mounted with Z1 mote (see Section 6.2.2).

For the RDC (Radio Duty Cycling) layer, 'nullRDC' is used for this simulated system, instead of the default 'ContikiMAC' mechanism. In a low-power network, the radio transceiver is usually switched off as much as possible to save energy [31], while RDC layer is handling the switch of radio transceiver. Making it ‘null’ never turns off the radio, and keeps radio awake all the time, so it will improve the response time of the system, compared with using the default RDC layer. To change RDC layer in Cooja, a new ‘project-conf.h’ is introduced, and the macro is defined in it: #define NETSTACK_CONF_RDC nullrdc_driver

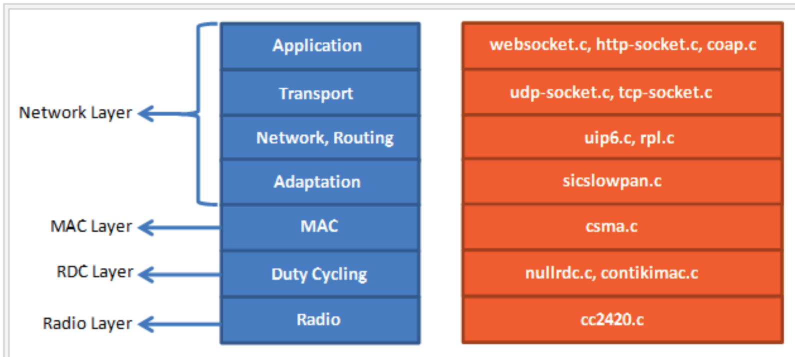


Figure 6.2 Contiki network stack

The MAC (Medium Access Control) layer sits on top of the RDC layer. The MAC layer is responsible for avoiding collisions at the radio medium and retransmitting packets if there were a collision. Contiki provides two mechanisms for MAC layer: a CSMA (Carrier Sense Multiple Access) mechanism and a ‘nullMAC’ mechanism that does not do any MAC-level processing. Here, we use the default CSMA mechanism for the simulated system.

For the network layer of the simulated system, we use UIP6 and use RPL as routing protocol: UIP6 represents that IPv6 addresses are used for nodes, where IPv6 addresses are distributed from ‘aaaa::c30c:0:0:1’ to ‘aaaa::c30c:0:0:14’ respectively for Node 1 to Node 20. RPL [32] is the Routing Protocol for Low-power and Lossy Networks (LLNs). RPL provides a mechanism where multipoint-to-point traffic from devices inside the LLN towards a central control point and point-to-multipoint traffic from the central control point to the devices inside the LLN are supported. Thus, RPL forms routing graph from a root node or AP (Access Point), and it builds acyclic graph from the root called DODAG (Destination Oriented Directed Acyclic Graph) [32].

For the transport layer, we use UDP (User Datagram Protocol) because it is connectionless, we do not need to spare much resource on handshaking dialogues, rather than TCP. We set the port as 1234.

Figure 6.3, 6.4, 6.5 show the initialization of the three types of nodes respectively, and in these figures, the abovementioned network configurations are printed out.

```
00:01.197  ID:1  Rime started with address 193.12.0.0.0.0.0.1
00:01.208  ID:1  MAC c1:0c:00:00:00:00:00:01 Contiki 2.7 started. Node id is set to 1.
00:01.216  ID:1  CSMA nullrdc, channel check rate 128 Hz, radio channel 26
00:01.229  ID:1  Tentative link-local IPv6 address fe80:0000:0000:0000:c30c:0000:0000:0001
00:01.235  ID:1  Starting 'Server process' 'Server process: broadcast'
00:01.241  ID:1  IPv6 addresses: aaaa::c30c:0:0:1
00:01.244  ID:1  fe80::c30c:0:0:1
```

Figure 6.3 Initialization of server node (Node 1)

```
00:01.048  ID:2  Rime started with address 193.12.0.0.0.0.0.2
00:01.058  ID:2  MAC c1:0c:00:00:00:00:00:02 Contiki 2.7 started. Node id is set to 2.
00:01.067  ID:2  CSMA nullrdc, channel check rate 128 Hz, radio channel 26
00:01.080  ID:2  Tentative link-local IPv6 address fe80:0000:0000:0000:c30c:0000:0000:0002
```

Figure 6.4 Initialization of room controller node (Node 2)

```
00:01.713  ID:3  Rime started with address 193.12.0.0.0.0.0.3
00:01.724  ID:3  MAC c1:0c:00:00:00:00:00:03 Contiki 2.7 started. Node id is set to 3.
00:01.733  ID:3  CSMA nullrdc, channel check rate 128 Hz, radio channel 26
00:01.745  ID:3  Tentative link-local IPv6 address fe80:0000:0000:0000:c30c:0000:0000:0003
00:01.749  ID:3  Starting 'desk controller process'
00:01.755  ID:3  IPv6 addresses: aaaa::c30c:0:0:3
00:01.758  ID:3  fe80::c30c:0:0:3
```

Figure 6.5 Initialization of one of the desk controller nodes (Node 3)

According to Section 5.1.1, the transmission range of each node is set as 10 meters, and interference range is set the same as transmission range. Receive and transmit success ratio is set as 100% for all nodes, see Figure 6.6. This setting guarantees that each sent message from a sender will be successfully received by a radio, however, collisions still exist, and that is why CSMA is used for MAC layer.

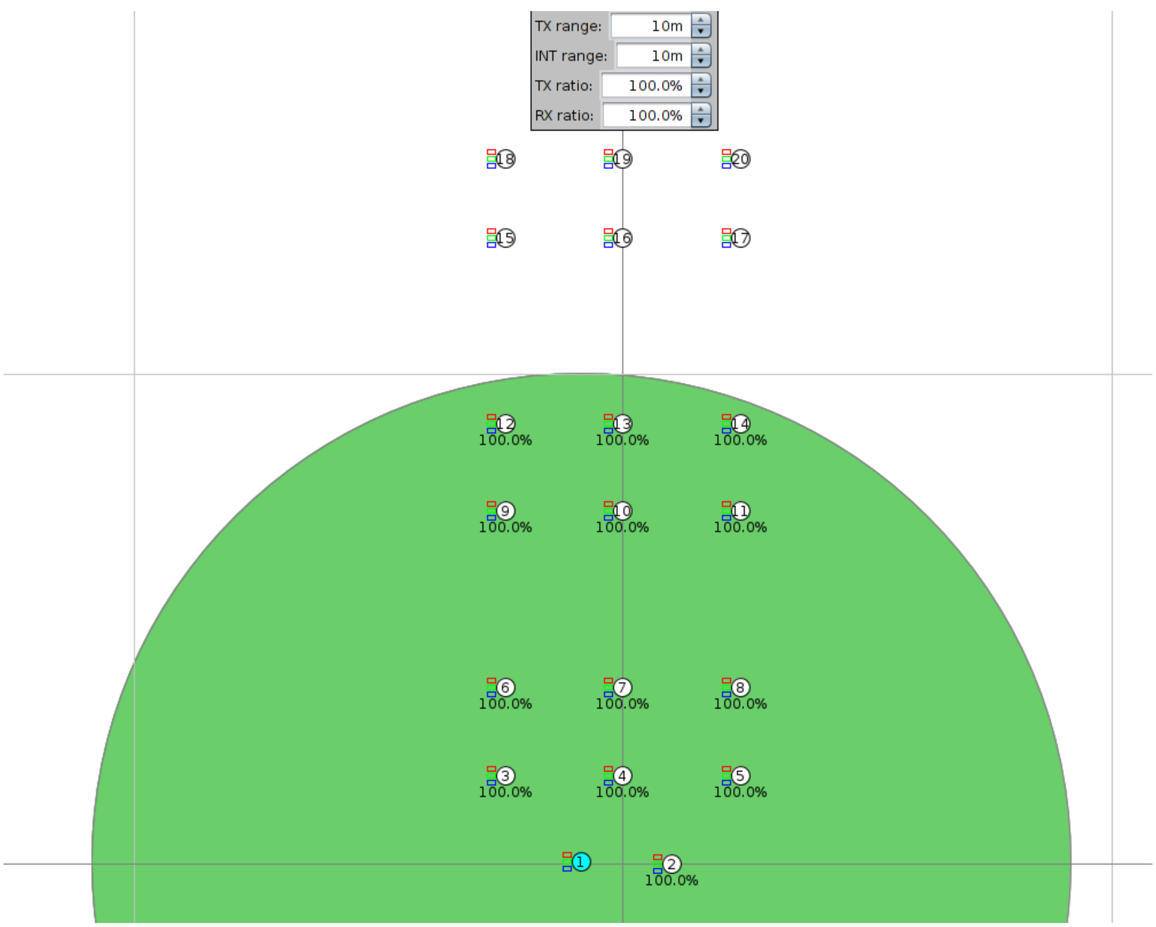


Figure 6.6 Transmission range and interference range of nodes

6.2.2. Hardware

We choose to use Zolertia Z1 mote for every node in this simulated system, and its configurations are listed in Table 6.1 [33]. Z1 motes own various inputs and outputs to allow users to interact with the network, including button, temperature sensor, light sensor and LED.

Table 6.1 Configurations of Zolertia Z1 motes

Parameter	Value
CPU	MSP430F2617
Instruction Set	16-bit RISC
Clock Speed	16MHz
RAM	8KB
Flash memory	92KB
Transceiver	CC2420
Radio	IEEE 802.15.4 compliant
	2.4 GHz & 250Kbps

6.3. Simulation Workflow

The simulation workflow is designed according to Figure 4.2, and shown as a flow diagram in Figure 6.7.

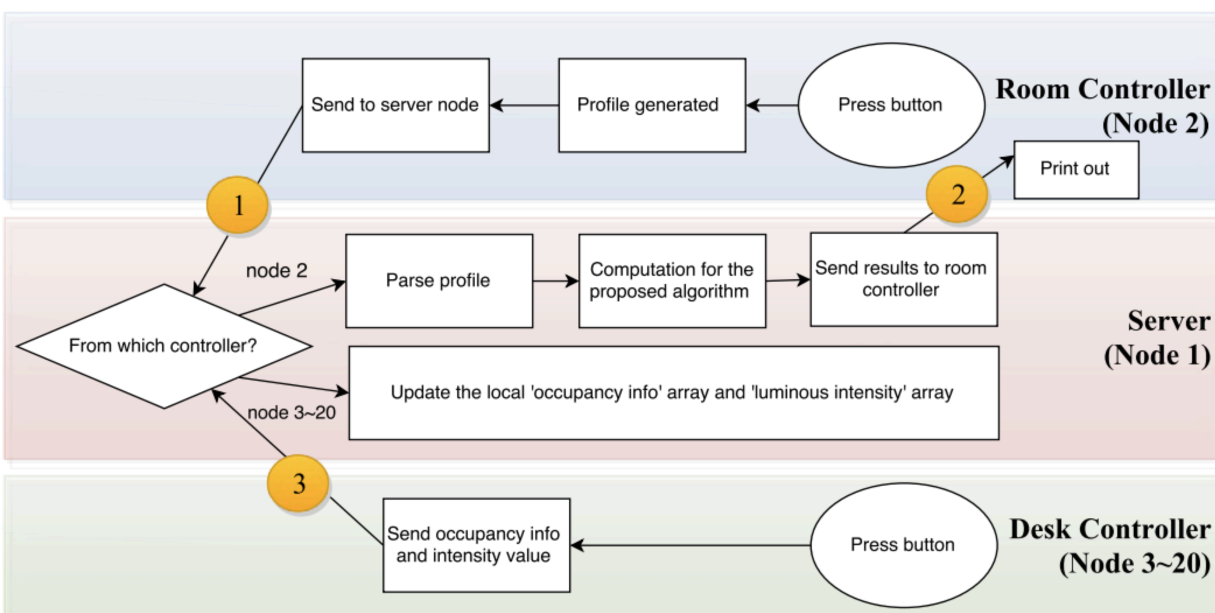


Figure 6.7 Simulation workflow

We use Zolertia Z1 mote for all the nodes. While the button of the room controller (Node 2) is pressed, which means that an office worker enters the office. A profile in CSV format comes in the room controller node. An example of this profile is “ID: 487, T: 25, C: 4500, E: 400, L: 0, AT: 0.30, AC: 0.20, AE: 0.10, AL: 0.40”. Then it sends the profile directly to the server node (Node 1) by unicast communication, and this is a one-hop communication because server node is in the transmission range of the room controller node. After the server node receives it, the profile firstly is parsed, and the attributes are extracted. Then it uses these attributes to calculate the conflict prevention algorithm, gets user satisfaction value for each available desk, and compares these user satisfaction values to find the maximum user satisfaction and it corresponding desk. In the conflict prevention algorithm, (3.9) is used, where $\sigma_T, \sigma_E, \sigma_C$ is set as 3.2683, 0.1823, 1429.88 respectively according to (3.28), to make the minimum value of each attribute’s user satisfaction be 0.05 (except location attribute), see Section 3.3.4. Finally, it returns the optimal desk number as well as its corresponding user satisfaction value to the room controller. After the room controller node receives the result, it prints it out on the screen of the simulation software.

Figure 6.8 illustrates an example of printed message on the simulation screen, where Node 2 (room-controller) sends a CSV-formatted profile to Node 1 (server) at the time 46.454s, and Node 1 receives it at the time 46.480s with length of 61 bytes, then it calculates according to the conflict prevention algorithm, and finally sends the result message including person’s ID, optimal desk result and user satisfaction value back to Node 2. Node 2 receives it at the time 47.170s. Overall, the response time for this office worker’s room check-in is considered as 47.170s - 46.454s = 716ms.

```
00:46.454   ID:2    id:487,T:25,C:4500,E:400,L:1,AT:0.30,AC:0.20,AE:0.10,AL:0.40
00:46.480   ID:1    Data received from fe80::c30c:0:0:2 on port 1234 from port 1234 with length 61
00:47.155   ID:1    Calculating conflict resolution algorithm......
00:47.170   ID:2    ID:487,Result:12,User Satisfaction:0.9985
```

Figure 6.8 An example of printed messages in Cooja

While the button of one of eighteen desk controller nodes (Node 3~20) is pressed, which means an office worker checks in on the desk. So the occupancy detector detects a change. The desk controller sends the occupancy information and the luminous intensity value to the server node by unicast communication. The server node knows which node it sends from, and can modify its local ‘occupancy information’ array as well as ‘luminous intensity’ array accordingly.

In Figure 6.7, three types of message communication are shown in circles, and their features are listed in Table 6.2, involving message content, packet size, starting point, and destination.

The first type of message is transmitted from the room controller node to the server node by unicast communication when the button of Node 2 is triggered. It contains the profile, which has a length of 61 bytes totally. After the computation for conflict prevention algorithm, the server node transmits the second type of message back to the room controller, which contains the results of computation: optimal desk number and user satisfaction. This message contains 41 bytes, and an example of this type of message can be found in Figure 6.8. The third type of message is a unicast communication, which contains a 39-byte message like "Occupancy: 1, Luminaire Intensity: 1500", and it is transmitted from one of the desk controllers to the server node by button trigger.

Table 6.2 Three types of messages

Message Type	**Message Content**	**Packet Size (Payload)**	**Starting Point**	**Destination**
1	Profile	61 bytes	Room controller (Node 2)	Server (Node 1)
2	Result of conflict prevention algorithm	41 bytes	Server (Node 1)	Room controller (Noder 2)
3	Occupancy information and luminous intensity value	39 bytes	Desk controller (Node 3~20)	Server (Node 1)

7. Experiments and Results

Based on the simulated system, nine experiments are conducted to see the features of the proposed conflict prevention algorithm, and to examine the performance and quality attributes of the system architecture as well as the performance of network. This chapter mainly introduces the experiments' setup, process, and the results are given and discussed.

7.1. Experiments

Nine experiments are conducted in the simulated system. This section introduces how these experiments set up. Results are shown and discussed in the next section.

7.1.1. Outline of Experiments

Before going to the details of the experiments, we should first explain the terminology that we are using in these experiments.

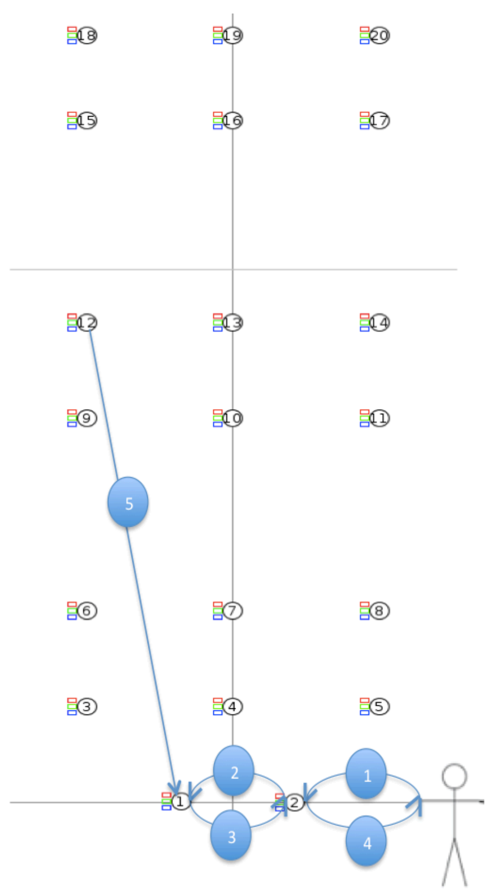


Figure 7.1 Explanation of response time and desk-controller-to-server delay

(1) Response time: Response time is the response time of room check-in, which is the duration from the time that a person checks in at the entrance of the office to the time that he/she receives an allocated desk on the smartphone. The response time includes two parts: communication time and computation time. Communication time includes four phases, shown as circles in Figure 7.1:

a) Phase 1 shows the time that a profile transmits from smartphone to room controller, while in the simulated system, this is triggered by a pressed button;

b) Phase 2 is the time that this profile is forwarded to the server node from room controller;

c) Phase 3 is the time that a server returns the result of desk number and user satisfaction after its computation;

d) Phase 4 represents the time that this result transmits to the smartphone, while this is simulated as printing out on the output screen.

Computation time is the time that the server node spends on calculating the algorithm according to formulization in Chapter 3.

(2) Desk-controller-to-sever delay: Desk-controller-to-server delay is the time that a desk controller sends successfully its occupancy information and light intensity values to the server node. For example, in Figure 7.1, circle No.5 represents the desk-controller-to-server delay of the desk controller node No. 12.

(3) User satisfaction: User satisfaction shows how satisfied the person will be if system assigns a desk according to his/her profile. The definition can be found in Section 3.2, and the formula used for calculation is (3.9).

(4) Entering order: Entering order means the order of a set of 18 different profiles inputs into the system. The proposed algorithm is related to the entering order, meaning that different entering order may cause different desk assignment results and individual's user satisfaction.

We have conducted nine experiments, and Table 7.1 lists all the experiments with brief introductions and what we have measured for each experiment. Noteworthy, only Experiment 3 is run on Cooja motes, which have no limit on hardware resources, but all the other eight experiments are run on Z1 motes, as introduced in Section 6.2. The reason why using Cooja motes for Experiment 3 is to simulate a server with powerful performance, so only communication time is counted for the response time, but no computation time. That is easier to compare with the response time in Experiment 2.

7.1.2. Experiment Setup

This section introduces the common experiment setups for all experiments, so the following experiments are conducted based on the following settings:

Table 7.1 Outline of all the experiments

Experiment No.	Brief Description	Measurement
Experiment 1	Desk assignment: 18 profiles entering the office in a certain order, possible conflicts happen; the proposed algorithm is compared with three other desk assignment methods.	User satisfaction
Experiment 2	Baseline experiment: The same 18 profiles in Exp. 1, but shuffled 100 times to enter the office with 100 different orders; this experiment runs on Z1 motes.	1. User satisfaction 2. Response time 3. Desk-Controller-to-Server Delay
Experiment 3	Same as Experiment 2, but runs on Cooja motes.	Response time
Experiment 4	Extreme Cases 1: Ideal setting, where profiles exactly match the desk setting (no conflicts).	User satisfaction
Experiment 5	Extreme Cases 2: 18 same profiles entering the office.	User satisfaction
Experiment 6	Extreme Cases 3: When location is the only attribute matters.	User satisfaction
Experiment 7	Scalability: Scale up the number of desk controller nodes.	1. Response time 2. Desk-Controller-to-Server Delay
Experiment 8	Fault Tolerance: Check whether network automatically creates route when interferences happen.	Desk-Controller-to-Server Communication
Experiment 9	Energy Consumption: Calculate the energy consumption of the three types of boards.	Power

Table 7.2 Initialized configurations of desk controller nodes

Node id	T_{dj} (℃)	C_{dj} (K)	E_{dj} (Lux)	L_{dj}	*Node id*	T_{dj} (℃)	C_{dj} (K)	E_{dj} (Lux)	L_{dj}
3	25	3000	325	1	12	25	4000	415	1
4	23	3000	335	0	13	23	4000	425	0
5	21	3000	345	0	14	21	4000	435	0
6	25	3000	355	1	15	25	5000	445	1
7	23	3000	365	0	16	23	5000	455	0
8	21	3000	375	0	17	21	5000	465	0
9	25	4000	385	1	18	25	5000	475	1
10	23	4000	395	0	19	23	5000	485	0
11	21	4000	405	0	20	21	5000	495	0

(1) The initialized status of the office is that all desks are not occupied, which is an entirely empty office. So initially we have 18 available desks.

(2) Table 7.2 is the initialized office setting, which means when every experiment starts, the desks in this office are configured according to Table 7.2. The desk numbers are from 3 to 20, the same as their node ID. For the eighteen desk controller nodes, the value of temperature, color temperature, and location in Tabl 7.2 is set according to Table 3.1. Initially, we uniformly distribute the illuminance values of desks (from 325 lux to 495 lux, in a 10 lux interval), see Table 7.2.

(3) We assume that every office worker accepts the optimal desk provided by the system, so after he/she comes in, that allocated desk becomes occupied.

(4) Table 7.3 lists the settings of the 18 profiles used in the experiments. This set of 18 profiles will be used for all experiments except Experiment 4, 5, 6 because the three experiments simulate extreme cases. Especially for Experiment 2 and 3, this set of 18 profiles will be shuffled by 100 times.

Table 7.3 The settings of the 18 profiles used in the experiments

Profile	ID	T_{pi} (℃)	C_{pi} (K)	E_{pi}	L_{pi}	AT_{pi}	AC_{pi}	AE_{pi}	AL_{pi}
1	432	19	5000	465	1	0.30	0.25	0.10	0.35
2	235	20	4500	335	0	0.10	0.15	0.15	0.60
3	679	21	4000	355	0	0.30	0.30	0.20	0.20
4	614	22	3500	375	0	0.25	0.25	0.25	0.25
5	918	23	3000	475	0	0.20	0.30	0.15	0.35
6	119	24	3500	405	1	0.30	0.20	0.40	0.10
7	203	25	4000	425	0	0.60	0.20	0.15	0.05
8	485	26	4500	345	1	0.30	0.25	0.15	0.30
9	459	27	5000	365	0	0.30	0.30	0.20	0.20
10	570	27	4500	385	0	0.45	0.25	0.15	0.15
11	487	26	4000	485	1	0.30	0.25	0.10	0.35
12	236	25	3500	495	0	0.10	0.15	0.15	0.60
13	677	24	3000	445	1	0.30	0.30	0.20	0.20
14	414	23	3500	395	0	0.25	0.25	0.25	0.25
15	581	22	4000	415	0	0.20	0.30	0.15	0.35
16	910	21	4500	325	0	0.30	0.20	0.40	0.10
17	333	20	5000	435	0	0.60	0.20	0.15	0.05
18	105	19	4000	455	1	0.30	0.25	0.15	0.30

7.1.3. Experiment Process

After introducing the common setup of the experiments, we are going to describe the processes of each experiment in detail.

(1) Experiment 1:

Experiment 1 is conducted in order to check how smart the proposed algorithm is, compared with other desk assignment methods. In this experiment, we input a group of 18 profiles, shown in Table 7.3. The value of each attribute is uniformly distributed among the 18 profiles, while none of the profiles can match one specific desk setting perfectly. The weights of attributes are randomly given.

Besides the proposed conflict prevention algorithm, three methods of desk assignment for this group of profiles are proposed and tested:

(a) Method 1: Each time a profile comes in, the system gives him/her the minimum desk number ID among available desks, which means it assigns desks by node ID.

(b) Method 2: Each time a profile comes in, the system randomly finds a desk for him/her among all available desks.

(c) Method 3: In this method, we assume that there exists someone who knows everybody's profile as well as is able to accurately predict the entering order of this group of profiles. Such a human can smartly analyze from the perspective of the entire office, so that he provides the best desk assignments for every profile.

(2) Experiment 2:

Experiment 2 uses the 18 profiles listed in Table 7.3, and shuffles by 100 times to check each profiles' user satisfaction and response time. We use the 'random.shuffle(array)' function in Python to shuffle the 18 profiles by 100 times. This function follows Fisher–Yates shuffle algorithm [34], which is able to generate an unbiased random permutation of the source data [35]. The generated 100 shuffled sets of profiles can be found in Table B.1 in Appendix B.

We also assume that the office worker goes to the allocated desk after getting the results, and the occupancy detector in the corresponding desk controller node detects him/her, so that the desk controller node sends a message back to the server node, which includes the updated occupancy information. Therefore, the desk-controller-to-server delay can also be got by each profile's check in. This experiment is run on Z1 motes, which means that the motes run on the hardware configurations in Table 6.1, where CPU frequency is only 16MHz. The results can be found in Table B.1 in Appendix B.

(3) Experiment 3:

Experiment 3 is conducted under the same settings of Experiment 2, except using different hardware. In this experiment, we use Cooja mote, which is a virtual mote, so it has no limit on hardware resources. We conduct this experiment to compare with the results from Experiment 2, especially the response time. We analyze the difference of response time in Experiment 2 and Experiment 3 in the next section. The results can be found in Table B.1 in Appendix B.

(4) Experiment 4:

Experiment 4 is an extreme case, where we input 18 profiles that exactly match the desk settings, so that we can check whether the proposed algorithm works well for non-conflict situations. Those profiles' settings can be found in Table 7.4.

Table 7.4 The 18 profiles that exactly match the desk settings

Sequence	ID	T_{pi} (℃)	C_{pi} (K)	E_{pi}	L_{pi}	AT_{pi}	AC_{pi}	AE_{pi}	AL_{pi}
1	432	21	5000	465	0	0.30	0.25	0.10	0.35
2	235	23	3000	335	0	0.10	0.15	0.15	0.60
3	679	25	3000	355	1	0.30	0.30	0.20	0.20
4	614	21	3000	375	0	0.25	0.25	0.25	0.25
5	918	25	5000	475	1	0.20	0.30	0.15	0.35
6	119	21	4000	405	0	0.30	0.20	0.40	0.10
7	203	23	4000	425	0	0.60	0.20	0.15	0.05
8	485	21	3000	345	0	0.30	0.25	0.15	0.30
9	459	23	3000	365	0	0.30	0.30	0.20	0.20
10	570	25	4000	385	1	0.45	0.25	0.15	0.15
11	487	23	5000	485	0	0.30	0.25	0.10	0.35
12	236	21	5000	495	0	0.10	0.15	0.15	0.60
13	677	25	5000	445	1	0.30	0.30	0.20	0.20
14	414	23	4000	395	0	0.25	0.25	0.25	0.25
15	581	25	4000	415	1	0.20	0.30	0.15	0.35
16	910	25	3000	325	1	0.30	0.20	0.40	0.10
17	333	21	4000	435	0	0.60	0.20	0.15	0.05
18	105	23	5000	455	0	0.30	0.25	0.15	0.30

(5) Experiment 5:

Experiment 5 is also an extreme case. We want to know what happens if a group of exactly same profiles entering together into the office? Thus, we use one profile to enter the office continuously by 18 times. The profile used in this experiment is: "ID: 432, T: 25, C: 4500, E: 400, L: 0, AT: 0.30, AC: 0.20, AE: 0.10, AL: 0.40".

(6) Experiment 6:

Experiment 6 is another extreme case with a group of 18 profiles, but we fix all of their weights to AT = 0, AC = 0, AE = 0, AL = 1, and then fix everybody's location value to 1. Therefore, everybody asks for a desk with window side, and they only care about location. The results of the three experiments of extreme cases are shown in Table 7.6.

(7) Experiment 7:

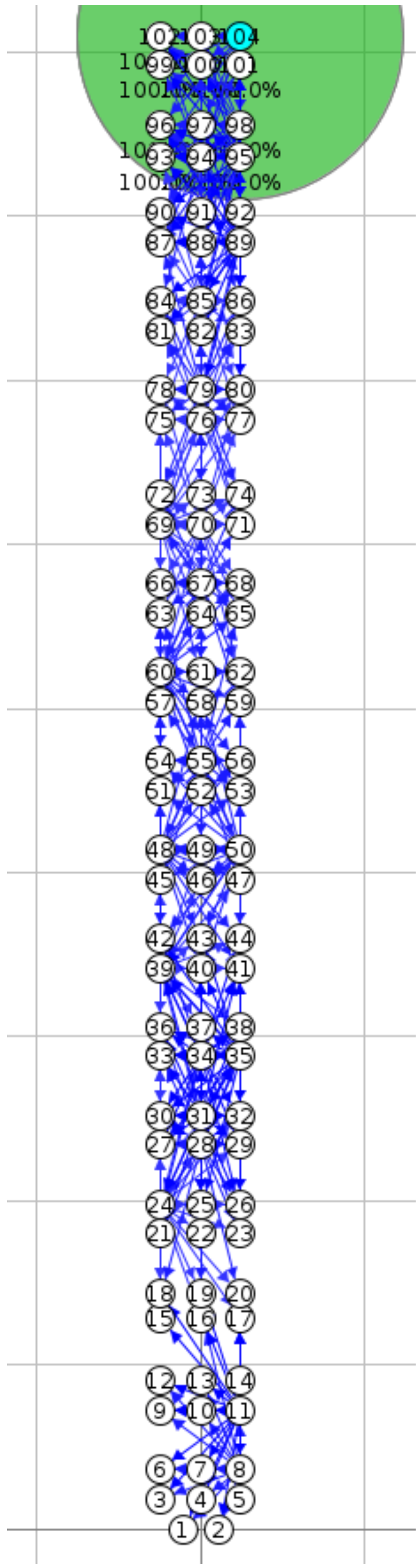


Figure 7.2 An example of the extended office layout with 102 desk controller nodes

Experiment 7 is designed for scalability. We extend the office according to the rule of the office layout, so we add six desk controller nodes as a set each time, and examine its response time and desk-controller-to-server delay. We only look at the maximum desk-controller-to-server delay, which is the farthest node from the server node. Figure 7.2 is an example of extended office with 102 desk controller nodes.

(8) Experiment 8:

From the non-functional requirements in Section 4.3, we know that there are three aspects when handling fault tolerance. However, due to the limited resources of hardware, the one related to duplicated work was not simulated. Fault tolerance can be checked from the other two aspects:

a) When exceptions happen, the system knows how to fix the errors. For example, if one of the attributes of profile is not in the correct range (see Section 3.3.2), the system will automatically round it to the near border of the range. Also, the weights of different attributes are checked within the calculation process. However, if all desks are occupied, the system cannot provide any desk for the office worker, so the system returns a message “All desks are occupied!”, shown in Figure 7.3.

```
02:40.783   ID:2    id:105,T:19,C:4000,E:455,L:1,AT:0.30,AC:0.25,AE:0.15,AL:0.30
02:40.844   ID:1    Data received from Room controller...Calculating conflict resolution algorithm....
02:40.860   ID:2    ID:105,Result:20,User Satisfaction:0.6325
02:41.682   ID:2    id:449,T:25,C:4500,E:400,L:1,AT:0.30,AC:0.20,AE:0.10,AL:0.40
02:41.701   ID:1    Data received from Room controller...All desks are occupied!
```

Figure 7.3 Fault tolerance example: "All desks are occupied!"

b) Experiment 8 is conducted to see how the system reacts when there are interferences in the network. To simulate interference, we add a jamming node (Node 21), which always broadcasts a jamming message. The message is in the size of 7 bytes: “jamming”, and its transmission range is set as 10 meters, its location can be seen in Figure 7.4. It broadcasts this message every 10 ms. The network in this experiment is configured without CSMA for MAC layer, so we introduce ‘nullMAC’ for MAC layer in order to allow interferences.

(9) Experiment 9:

Different components on a board can get its energy consumption by using ‘energest.h’, and we are able to get the radio receiving and transmitting time by measuring the times that it is working on. This time can be multiplied with the voltage and a pre-measured current, which approximates power consumption of a transceiver [36]:

$$Power(mW) = \frac{rx+tx}{cpu+lpm} * 10mA * 3V \tag{7.1}$$

, where 10mA and 3V are the operational current and voltage, found in Z1 mote's datasheet [2]. *rx* is the time that the radio was in receiving mode, and similarly, *tx* is the time that the radio in sending mode. We also need the total time that has passed, which is computed by summing time spent by the cpu in active mode (cpu) and low-power mode (lpm). These times are printed out, and they are the time that the transceiver receives and transmits data for sending one message. We run the system for a set of 18 profiles, after allocating all 18 desks, the energy message is printed out for server, room controller and desk controllers respectively.

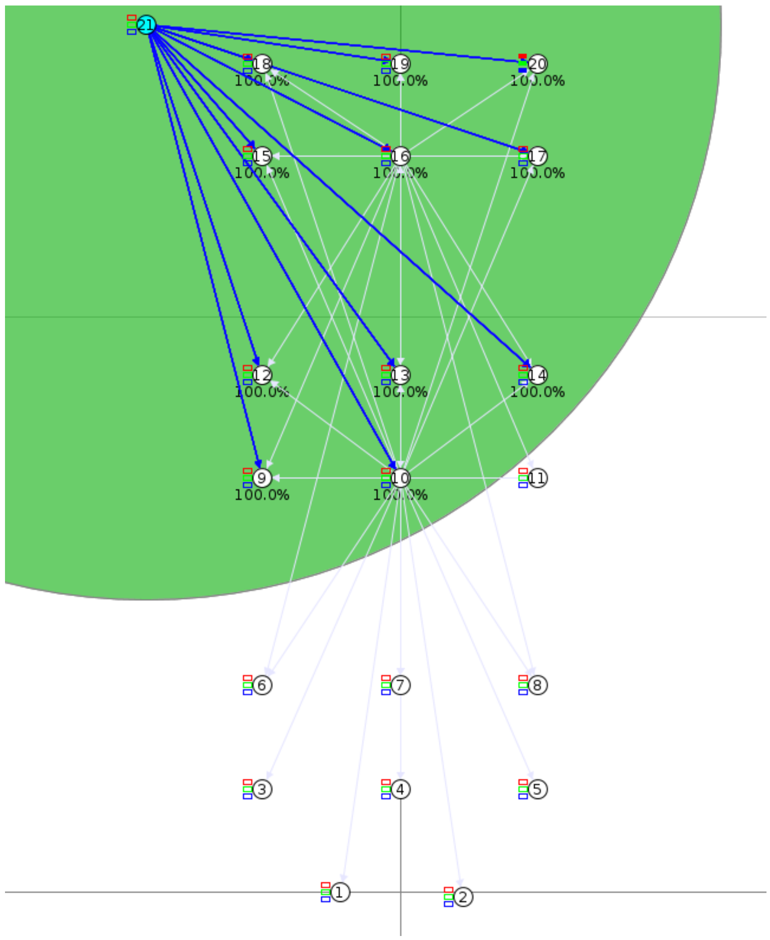


Figure 7.4 Jamming node (Node 21) is added

7.2. Results and Discussions

After experimental setup, we conducted nine experiments, which were described in the previous section. Here, in this section, we can see the results from the experiments, and the results are discussed in terms of the features of the proposed algorithm, the network performance and quality attributes of the system architecture.

7.2.1. Desk Assignment

The results of the proposed algorithm are printed out as shown in Figure 7.5. We have verified these values of user satisfaction by manually calculating formula (3.9) for the 18 profiles, and the results given by the system are correct.

```
ID:2    id:432,T:19,C:5000,E:465,L:1,AT:0.30,AC:0.25,AE:0.10,AL:0.35
ID:1    Data received from Room controller...Calculating conflict resolution algorithm......
ID:2    ID:432,Result:18,User Satisfaction:0.7549
ID:2    id:235,T:20,C:4500,E:335,L:0,AT:0.10,AC:0.15,AE:0.15,AL:0.60
ID:1    Data received from Room controller...Calculating conflict resolution algorithm......
ID:2    ID:235,Result:5,User Satisfaction:0.9300
ID:2    id:679,T:21,C:4000,E:355,L:0,AT:0.30,AC:0.30,AE:0.20,AL:0.20
ID:1    Data received from Room controller...Calculating conflict resolution algorithm......
ID:2    ID:679,Result:11,User Satisfaction:0.9540
ID:2    id:614,T:22,C:3500,E:375,L:0,AT:0.25,AC:0.25,AE:0.25,AL:0.25
ID:1    Data received from Room controller...Calculating conflict resolution algorithm......
ID:2    ID:614,Result:8,User Satisfaction:0.9737
ID:2    id:918,T:23,C:3000,E:475,L:0,AT:0.20,AC:0.30,AE:0.15,AL:0.35
ID:1    Data received from Room controller...Calculating conflict resolution algorithm......
ID:2    ID:918,Result:13,User Satisfaction:0.9094
ID:2    id:119,T:24,C:3500,E:405,L:1,AT:0.30,AC:0.20,AE:0.40,AL:0.10
ID:1    Data received from Room controller...Calculating conflict resolution algorithm......
ID:2    ID:119,Result:12,User Satisfaction:0.9708
ID:2    id:203,T:25,C:4000,E:425,L:0,AT:0.60,AC:0.20,AE:0.15,AL:0.05
ID:1    Data received from Room controller...Calculating conflict resolution algorithm......
ID:2    ID:203,Result:9,User Satisfaction:0.9294
ID:2    id:485,T:26,C:4500,E:345,L:1,AT:0.30,AC:0.25,AE:0.15,AL:0.30
ID:1    Data received from Room controller...Calculating conflict resolution algorithm......
ID:2    ID:485,Result:6,User Satisfaction:0.8786
ID:2    id:459,T:27,C:5000,E:365,L:0,AT:0.30,AC:0.30,AE:0.20,AL:0.20
ID:1    Data received from Room controller...Calculating conflict resolution algorithm......
ID:2    ID:459,Result:10,User Satisfaction:0.7588
ID:2    id:570,T:27,C:4500,E:385,L:0,AT:0.45,AC:0.25,AE:0.15,AL:0.15
ID:1    Data received from Room controller...Calculating conflict resolution algorithm......
ID:2    ID:570,Result:15,User Satisfaction:0.7177
ID:2    id:487,T:26,C:4000,E:485,L:1,AT:0.30,AC:0.25,AE:0.10,AL:0.35
ID:1    Data received from Room controller...Calculating conflict resolution algorithm......
ID:2    ID:487,Result:3,User Satisfaction:0.8410
ID:2    id:236,T:25,C:3500,E:495,L:0,AT:0.10,AC:0.15,AE:0.15,AL:0.60
ID:1    Data received from Room controller...Calculating conflict resolution algorithm......
ID:2    ID:236,Result:19,User Satisfaction:0.9185
ID:2    id:677,T:24,C:3000,E:445,L:1,AT:0.30,AC:0.30,AE:0.20,AL:0.20
ID:1    Data received from Room controller...Calculating conflict resolution algorithm......
ID:2    ID:677,Result:7,User Satisfaction:0.6970
ID:2    id:414,T:23,C:3500,E:395,L:0,AT:0.25,AC:0.25,AE:0.25,AL:0.25
ID:1    Data received from Room controller...Calculating conflict resolution algorithm......
ID:2    ID:414,Result:14,User Satisfaction:0.9098
ID:2    id:581,T:22,C:4000,E:415,L:0,AT:0.20,AC:0.30,AE:0.15,AL:0.35
ID:1    Data received from Room controller...Calculating conflict resolution algorithm......
ID:2    ID:581,Result:16,User Satisfaction:0.9078
ID:2    id:910,T:21,C:4500,E:325,L:0,AT:0.30,AC:0.20,AE:0.40,AL:0.10
ID:1    Data received from Room controller...Calculating conflict resolution algorithm......
ID:2    ID:910,Result:4,User Satisfaction:0.8586
ID:2    id:333,T:20,C:5000,E:435,L:0,AT:0.60,AC:0.20,AE:0.15,AL:0.05
ID:1    Data received from Room controller...Calculating conflict resolution algorithm......
ID:2    ID:333,Result:17,User Satisfaction:0.9628
ID:2    id:105,T:19,C:4000,E:455,L:1,AT:0.30,AC:0.25,AE:0.15,AL:0.30
ID:1    Data received from Room controller...Calculating conflict resolution algorithm......
ID:2    ID:105,Result:20,User Satisfaction:0.5793
```

Figure 7.5 The printed results of proposed algorithm method in Experiment 1

The results of the other three desk assignment methods are listed in Table 7.5. According to Figure 7.5 and Table 7.5, Figure 7.6 is generated, showing that the average and minimum user satisfaction of method 1,2,3 and the proposed algorithm. We can tell from Figure 7.6 that method 1, 2, 3 has average user satisfaction of 0.7335, 0.7362 and 0.8709 respectively. The average user satisfaction of the proposed algorithm is 0.8584, which is better than the first and second desk assignment methods. However, it is a little worse than the third method of desk assignment. The third method is considered as the most optimized solution, because it knows all profiles' information and their entering order, so that it has the ability to reserve a desk for somebody, and it has the ability to look to the future. Minimum user satisfaction of the proposed algorithm reaches the same value as the smart method of desk assignment. Overall, the proposed algorithm can provide a quite accurate desk assignment to satisfy people's requirements, but still it is not the smartest algorithm. To have a better accuracy on desk assignment, the algorithm should have the ability to predict who will come in the future and what profile he/she will have.

Table 7.5 The results of Experiments 1

Profile	Assigned Desk (Method 1)	User Satisfaction (Method 1)	Assigned Desk (Method 2)	User Satisfaction (Method 2)	Assigned Desk (Method 3)	User Satisfaction (Method 3)	Assigned Desk (Proposed Algorithm)	User Satisfaction (Proposed Algorithm)
1	1	0.5141	12	0.6837	12	0.6837	18	0.7549
2	2	0.9021	15	0.2167	5	0.9300	5	0.9300
3	3	0.9324	11	0.9540	11	0.9540	8	0.9540
4	4	0.6381	10	0.9637	8	0.9737	7	0.9737
5	5	0.9028	5	0.8480	13	0.9094	19	0.9094
6	6	0.7509	9	0.9592	9	0.9592	12	0.9708
7	7	0.9294	4	0.7681	15	0.9019	15	0.9294
8	8	0.5459	3	0.8726	6	0.8786	6	0.8786
9	9	0.6605	14	0.6164	10	0.7588	10	0.7588
10	10	0.7461	19	0.6652	16	0.6965	9	0.7177
11	11	0.5237	6	0.8551	18	0.9313	3	0.8410
12	12	0.9050	16	0.9042	19	0.9185	16	0.9185
13	13	0.7990	17	0.5039	3	0.8315	13	0.6970
14	14	0.8292	18	0.5013	7	0.9627	11	0.9098
15	15	0.8992	20	0.8697	14	0.9859	14	0.9078
16	16	0.3758	7	0.7907	4	0.8586	4	0.8586
17	17	0.7692	13	0.7491	17	0.9628	17	0.9628
18	18	0.5793	8	0.5300	20	0.5793	20	0.5793
Average		0.7335		0.7362		0.8709		0.8584

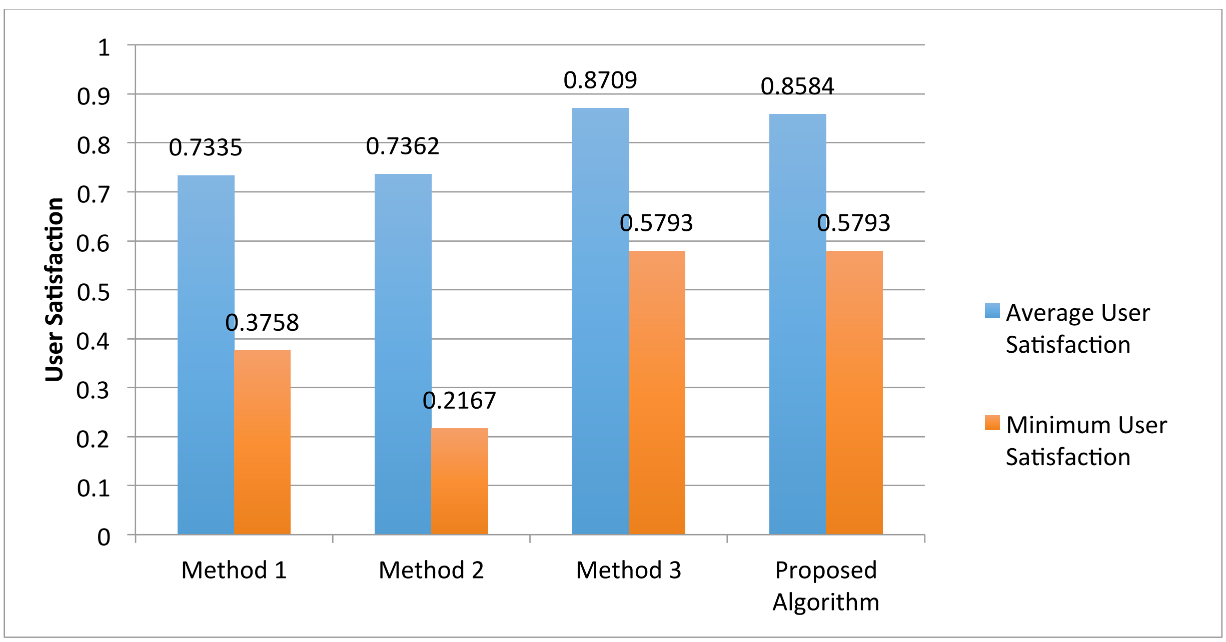


Figure 7.6 The average and minimum user satisfaction of three desk allocation methods compared with the proposed algorithm method

7.2.2. User Satisfaction

In Experiment 2, we shuffle the 18 profiles for 100 times, and the results of each profile's user satisfaction are listed in Table B.1 in Appendix B. We add up the 100 values of user satisfaction on each of the entering orders, and get the average user satisfaction for each of the entering orders, generated in Figure 7.7.

From the results in Table B.1, we can see that different entering orders of the same 18 profiles lead to different desk assignment results, as well as different user satisfaction for each profile. Figure 7.7 tells us that the average user satisfaction decreases when profiles enter, which means the earlier a profile enters, the higher user satisfaction it is likely to get; otherwise, the later he/she enters, a desk with less user satisfaction he/she may get. All in all, the user satisfaction of the proposed algorithm depends on the entering order.

The average user satisfaction for all the profiles in Table B.1 is 0.8594. From the statistics in Table B.1, we can also tell that the minimum user satisfaction in one shuffled set of 18 profiles is in the range of 0.35 ~ 0.70.

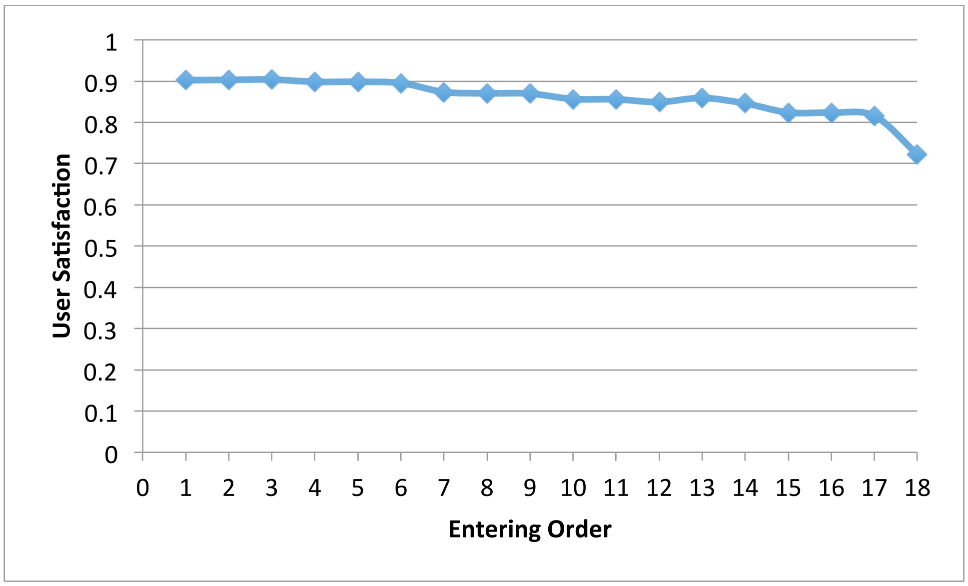


Figure 7.7 User satisfaction vs. entering order

7.2.3. Extreme Cases

Experiment 4, 5, 6 are three different extreme cases. We conclude the results in Table 7.6, and we analyze the results respectively.

Table 7.6 The results of Experiment 4, 5, 6

Profile	Desk Number (Exp.4)	User satisfaction (Exp.4)	Desk Number (Exp.5)	User satisfaction (Exp.5)	Desk Number (Exp.6)	User satisfaction (Exp.6)
1	12	1	12	0.9861	3	1
2	9	1	9	0.9859	6	1
3	15	1	15	0.9724	9	1
4	18	1	18	0.9522	12	1
5	6	1	6	0.8960	15	1
6	3	1	3	0.8676	18	1
7	10	1	10	0.5366	4	0
8	13	1	13	0.5315	5	0
9	16	1	16	0.5148	7	0
10	19	1	19	0.4941	8	0
11	1	1	7	0.4522	10	0
12	11	1	11	0.4297	11	0
13	4	1	4	0.4264	13	0
14	14	1	14	0.4199	14	0
15	17	1	17	0.4010	16	0
16	20	1	20	0.3805	17	0
17	8	1	8	0.3511	19	0
18	5	1	5	0.3291	20	0

(1) No conflict situation:

The first two columns in Table 7.6 are the results of Experiment 4, in which everybody can get a desk with user satisfaction of 1. It shows that the proposed algorithm not only works for conflicts, but also provides accurate desks for non-conflict profiles.

(2) Same profiles:

As we can see from the middle two columns in Table 7.6, the user satisfaction decreases with a single profile comes continuously for 18 times, and the least user satisfaction is 0.3291, which is below the range of minimum user satisfaction in Section 7.2.2: 0.35~0.7.

(3) Location only matters:

In the last two columns in Table 7.6, the only six window-site desks are assigned to the first six people, in the manner of "first-come-first-serve". However, the rest 12 people have user satisfaction of 0, so the system would randomly give them one desk from the available desks. Only this case leads to a user satisfaction of 0, and random desk assignment is only introduced in this extreme use.

In normal cases, because of the fact that the minimum user satisfaction values of the other three attributes are set as 0.05 (see Section 6.3), if one of the weights of these attributes is not 0, the final user satisfaction would not be 0. Therefore, the system would not randomly assign any available desk for the user.

7.2.4. Response Time of Room Check-in

Experiment 2 and Experiment 3 are conducted to investigate the response time of individual's check-in, and the results can be found in Table B.1 in Appendix B. We get the average response time of 18 profiles in 100 entering orders for Experiment 2 and 3, and generate Figure 7.8 and Figure 7.9 respectively.

Figure 7.8 is the result of Experiment 2, which runs on Z1 motes. The x-axis is the ID of entering order, where we calculate the average response time of 100 entering order for each of the entering orders. It is easy to tell from the figure that the later a person enters the office, the less his/her response time is. It is because of the fact that the later a person enters, the less desks are available, so the proposed algorithm would calculate less, and cause the response time less. The average response time for the entire 18 entering orders is 396.8 ms.

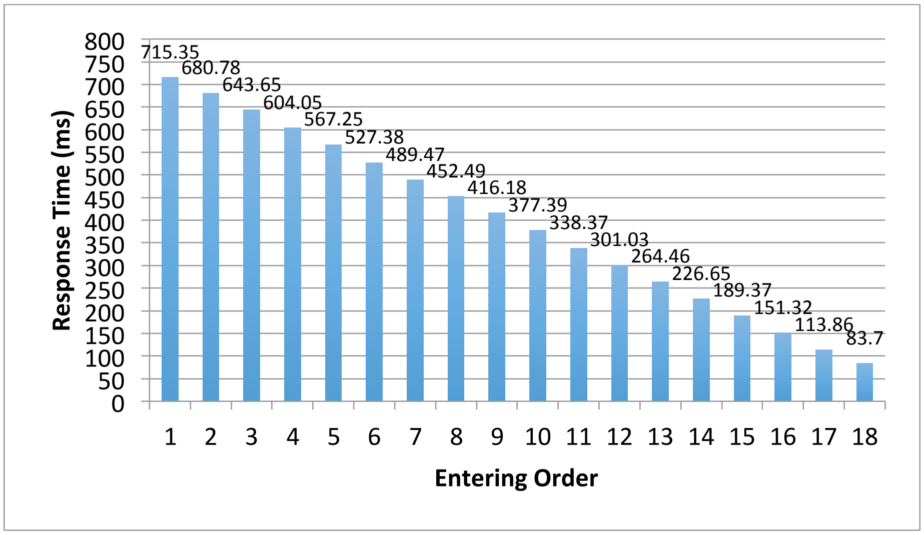


Figure 7.8 Response time vs. entering order (Z1 motes)

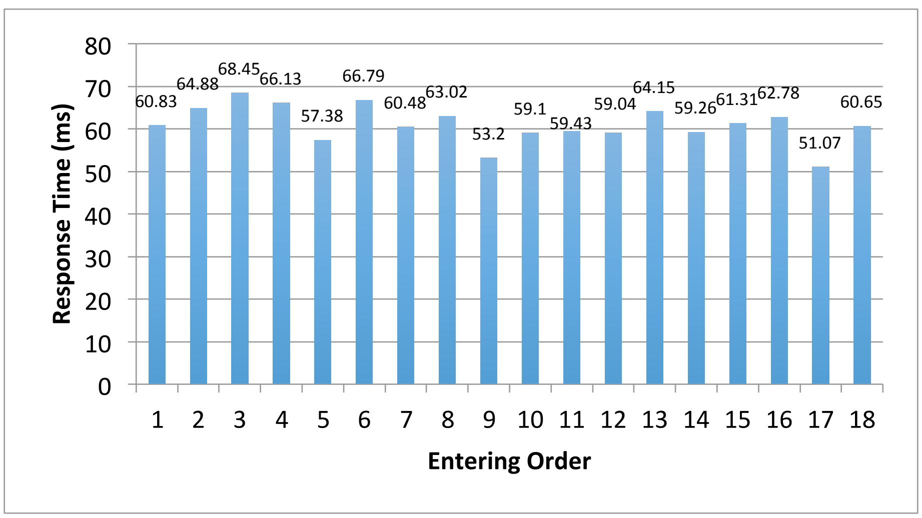


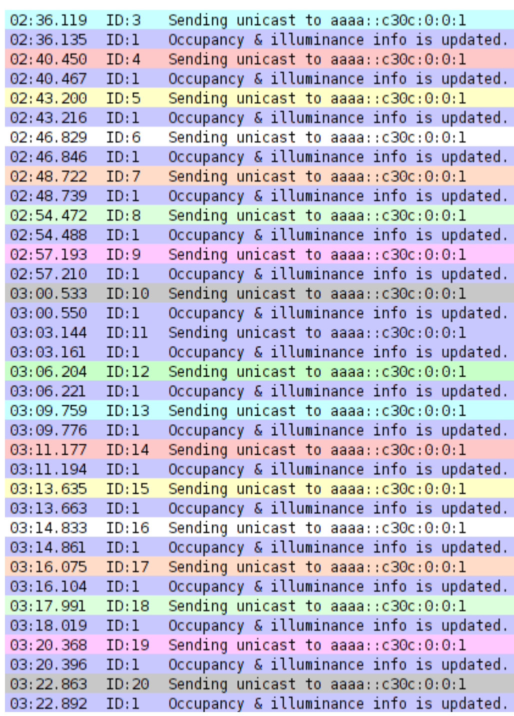
Figure 7.9 Response time vs. entering order (Cooja motes)

However, in Figure 7.9, which shows that response time does not depend on the ID of entering order. That is because Experiment 3 is run on Cooja motes, which actually are virtual motes, so the calculation time is not counted into the response time. Therefore, Figure 7.9 tells us that the average communication time for a profile check-in is 61ms, and we can consider that the user capacity of this system is 16 persons in one second.

Compared Figure 7.8 with Figure 7.9, we can find that the calculation time dominates in the response time of a profile's check-in. Thus, if we have a more powerful CPU on the server node, the system performs a better response time. The communication time for the profile's check-in is only 61 ms. If we assume that a powerful enough server is used, the response time can minimize to 61 ms, which is lower than our expectation: 91 ms (see Table 5.8).

7.2.5. Desk-Controller-to-Server Delay

Desk-controller-to-server delay is the duration from the time that one of the desk controller nodes sends its message to the time that the server receives it. In Experiment 2, we not only counted the response time, but also recorded the desk-controller-to-server delay after each time an office worker gets a result. We assumed that he/she accepts the optimal desk that the system assigns, and goes to the assigned desk directly, so that we simulate that the corresponding desk controller node transmits a message to the server node and the server node updates the occupancy information array. Each desk-controller-to-server delay is recorded in Table B.1 in Appendix B.



```
02:36.119  ID:3   Sending unicast to aaaa::c30c:0:0:1
02:36.135  ID:1   Occupancy & illuminance info is updated.
02:40.450  ID:4   Sending unicast to aaaa::c30c:0:0:1
02:40.467  ID:1   Occupancy & illuminance info is updated.
02:43.200  ID:5   Sending unicast to aaaa::c30c:0:0:1
02:43.216  ID:1   Occupancy & illuminance info is updated.
02:46.829  ID:6   Sending unicast to aaaa::c30c:0:0:1
02:46.846  ID:1   Occupancy & illuminance info is updated.
02:48.722  ID:7   Sending unicast to aaaa::c30c:0:0:1
02:48.739  ID:1   Occupancy & illuminance info is updated.
02:54.472  ID:8   Sending unicast to aaaa::c30c:0:0:1
02:54.488  ID:1   Occupancy & illuminance info is updated.
02:57.193  ID:9   Sending unicast to aaaa::c30c:0:0:1
02:57.210  ID:1   Occupancy & illuminance info is updated.
03:00.533  ID:10  Sending unicast to aaaa::c30c:0:0:1
03:00.550  ID:1   Occupancy & illuminance info is updated.
03:03.144  ID:11  Sending unicast to aaaa::c30c:0:0:1
03:03.161  ID:1   Occupancy & illuminance info is updated.
03:06.204  ID:12  Sending unicast to aaaa::c30c:0:0:1
03:06.221  ID:1   Occupancy & illuminance info is updated.
03:09.759  ID:13  Sending unicast to aaaa::c30c:0:0:1
03:09.776  ID:1   Occupancy & illuminance info is updated.
03:11.177  ID:14  Sending unicast to aaaa::c30c:0:0:1
03:11.194  ID:1   Occupancy & illuminance info is updated.
03:13.635  ID:15  Sending unicast to aaaa::c30c:0:0:1
03:13.663  ID:1   Occupancy & illuminance info is updated.
03:14.833  ID:16  Sending unicast to aaaa::c30c:0:0:1
03:14.861  ID:1   Occupancy & illuminance info is updated.
03:16.075  ID:17  Sending unicast to aaaa::c30c:0:0:1
03:16.104  ID:1   Occupancy & illuminance info is updated.
03:17.991  ID:18  Sending unicast to aaaa::c30c:0:0:1
03:18.019  ID:1   Occupancy & illuminance info is updated.
03:20.368  ID:19  Sending unicast to aaaa::c30c:0:0:1
03:20.396  ID:1   Occupancy & illuminance info is updated.
03:22.863  ID:20  Sending unicast to aaaa::c30c:0:0:1
03:22.892  ID:1   Occupancy & illuminance info is updated.
```

Figure 7.10 An example of the printed results of desk-controller-to-server delay

One example of the printed results of desk-controller-to-server delay is given in Figure 7.10. Figure 7.11 is generated for each entering order, according to Table B.1. We can find that Node 3~14 have 16~17 ms desk-controller-to-server delay while Node 15~20 have 27~28 ms for desk-controller-to-server delay. That is because of the transmission range of each node is set as 10 meters in this office (see Figure 6.6), the server node is in the transmission range of Node 3~14, so they transmit messages by one hop to the server node, and its desk-controller-to-server delay (16~17 ms) is the one-hop transmission time. However, the server node is not in the transmission range of Node 15~20, and they have to take 2 hops to reach the server node, which results in 27~28 ms for their desk-controller-to-server delay.

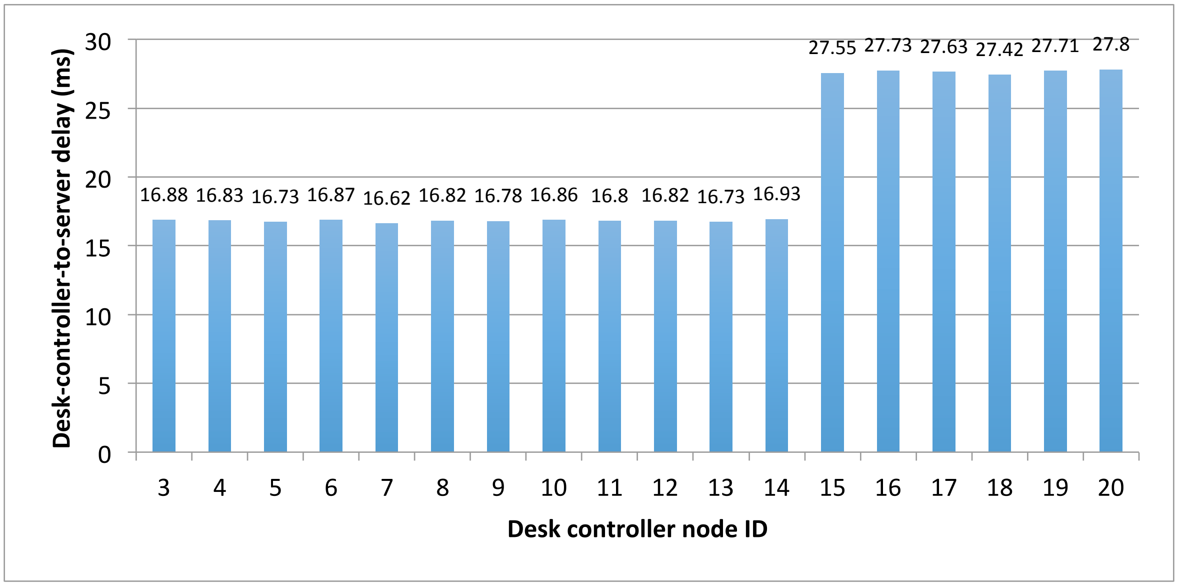


Figure 7.11 Desk-controller-to-server delay

7.2.6. Scalability

Experiment 7 is conducted for scalability, which runs on Z1 motes, so the response time includes the computation time by a 16MHz CPU. Results about the average response time can be seen in Figure 7.12, while Figure 7.13 is for the desk-controller-to-server delay. When scaling up the number of the desk controller nodes, the average response time keeps at 396~398 ms, that is because it is one hop from the room controller to the server node, so its transmission time is still fixed when scaling up the number of desk controllers from 18 to 156. Thus, scaling does not affect the response time for users.

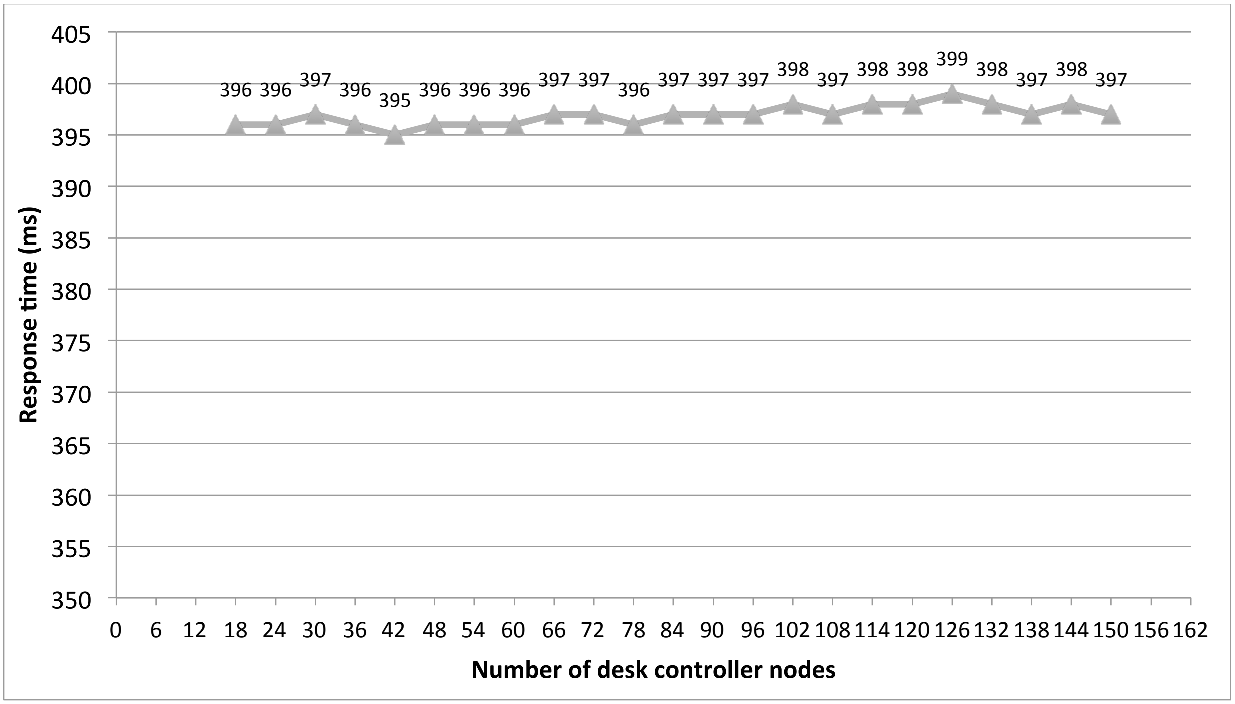


Figure 7.12 Average response time when scaling the number of desk controller nodes (with Z1 motes)

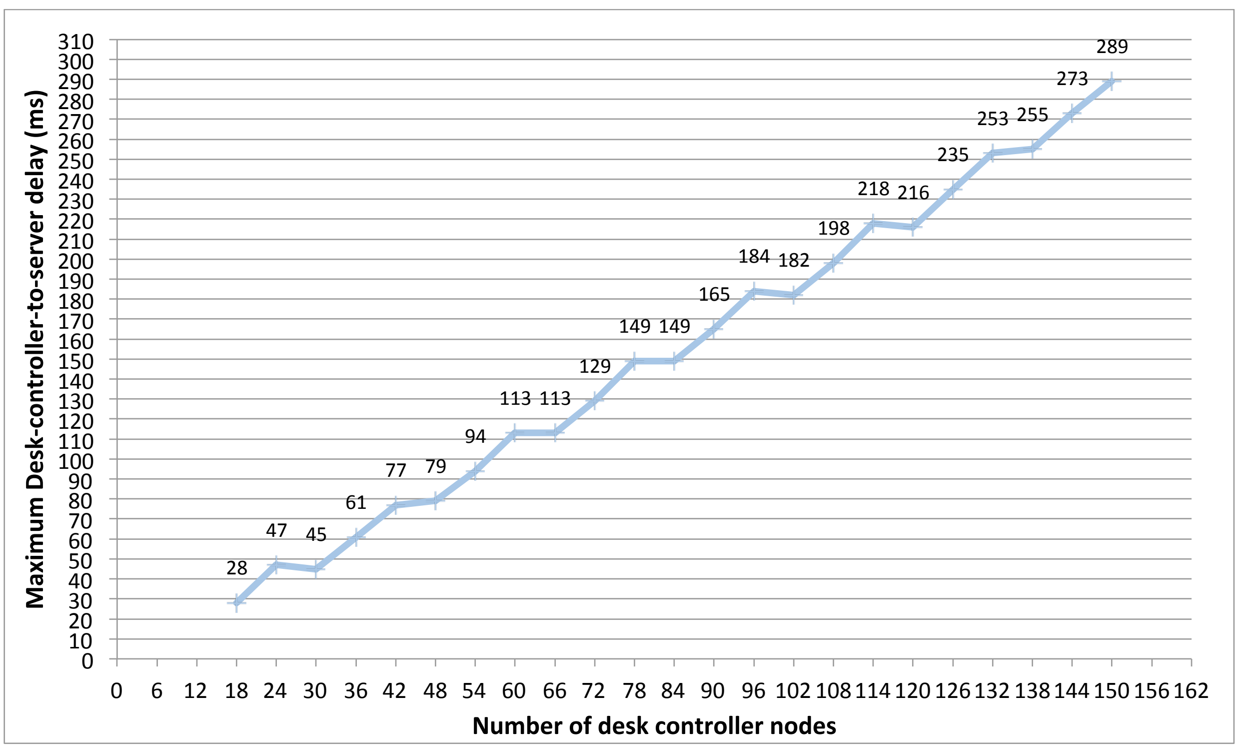


Figure 7.13 Maximum desk-controller-to-server when scaling the number of desk controller nodes (with Z1 motes)

However, we can see from Figure 7.13 that the desk-controller-to-server delay increases when scaling up the number of desk controller nodes because when we add nodes, the farther node takes more hops than the nearer nodes.

There is a human benchmark online [37], and it tests human's reaction time. Its result shows that the average (median) human reaction time is 215 ms, which means humans are not likely to react to an action within 215 ms. According to human's reaction time, we can set the threshold of the desk-controller-to-server delay as 215 ms. If the message from the desk controller nodes can reach the server node within 215 ms, humans would not tell the internal components and communications within this system. According to this threshold, we find that 114 is the maximum number of desk controllers, because the node with maximum hops among the 114 desk controller nodes takes 218 ms to reach the server node.

What is more, at the moment that a person checks in, the system should wait for at least 215 ms and then start to do the calculation for the proposed algorithm, because the system should wait and see if there are some updated messages coming from the farthest desk controller nodes to make sure that all available desks are considered into its calculation.

7.2.7. Fault Tolerance

To see whether the network can find a route when interference comes, we conduct Experiment 8. In order to introduce interferences, we use 'nullMAC' as MAC layer. The printed message on the screen of Cooja (see Figure 7.14) shows that Node 20 can still find its route to the server node after the jamming node (Node 21) initialized and broadcast jamming messages every 10 ms. Furthermore, its desk-controller-to-server delay is 30 ms, which is a bit more than the average desk-controller-to-server delay of Node 20 (27.8 ms in Figure 7.11). That is because we use RPL as routing layer, so the desk controller node is able to find a route when there are interferences in its transmission range. Thus, we think that the network has the ability to recover from 10ms-interval interference automatically.

```
01:54.301   ID:21   Rime started with address 193.12.0.0.0.0.0.21
01:54.312   ID:21   MAC c1:0c:00:00:00:00:00:15 Contiki 2.7 started. Node id is set to 21.
01:54.320   ID:21   CSMA nullrdc, channel check rate 128 Hz, radio channel 26
01:54.333   ID:21   Tentative link-local IPv6 address fe80:0000:0000:0000:c30c:0000:0000:0015
01:54.338   ID:21   Starting 'UDP broadcast example process'
01:56.098   ID:20   Sending unicast to aaaa::c30c:0:0:1
01:56.128   ID:1    Occupancy & illuminance info is updated.
```

Figure 7.14 A printed message when a jamming node is added

7.2.8. Energy Consumption

Experiment 9 is conducted to get the energy consumption. From the printed messages in Figure 7.15, 7.16 and 7.17, we can calculate the power consumption of radio transceivers on the three types of nodes by (7.1), and the results are summarized in Table 7.7, where we can find that the power consumption of radio on the three types of nodes are approximately 29.7 mW, which can be considered as an energy efficient radio.

```
04:15.572  ID:2   id:581,T:22,C:4000,E:415,L:0,AT:0.20,AC:0.30,AE:0.15,AL:0.35
04:15.633  ID:1   Data received from Room controller...Calculating conflict resolution algorithm......
04:15.642  ID:1   energy rx: 8234942
04:15.645  ID:1   energy tx: 2051
04:15.647  ID:1   cpu: 486561
04:15.650  ID:1   lpm: 7848365
04:15.662  ID:2   ID:581,Result:20,User Satisfaction:0.9335
```

Figure 7.15 Energy consumption message on server node

```
02:22.736  ID:2   id:581,T:22,C:4000,E:415,L:0,AT:0.20,AC:0.30,AE:0.15,AL:0.35
02:22.742  ID:2   energy rx: 4606948
02:22.745  ID:2   energy tx: 2502
02:22.748  ID:2   cpu: 121522
02:22.751  ID:2   lpm: 4519556
02:22.848  ID:1   Data received from Room controller...Calculating conflict resolution algorithm......
02:22.866  ID:2   ID:581,Result:20,User Satisfaction:0.9335
```

Figure 7.16 Energy consumption message on room controller node

```
03:35.761  ID:2   ID:1,Result:3,User Satisfaction:0.0000
03:38.332  ID:20  Sending unicast to aaaa::c30c:0:0:1
03:38.341  ID:20  energy rx: 7077777
03:38.344  ID:20  energy tx: 1077
03:38.346  ID:20  cpu: 193890
03:38.349  ID:20  lpm: 6915747
03:38.373  ID:1   Occupancy & illuminance info is updated.
```

Figure 7.17 Energy consumption message on one of the desk controller nodes

Table 7.7 Power consumption of radio on three types of nodes when transmitting messages

	Server	Room Controller	Desk Controller
Radio Power (mW)	29.6475	29.7955	29.8701

8. Conclusions

After experiments are conducted in Chapter 7, the results are given and discussed. This chapter concludes the whole project: at first, some conclusions on the results will be made. After that, we will discuss what have been learned and observed during the project, as well as what can be useful for the lighting industry. Finally, we lay out some future works.

8.1. Conclusions

According to the results in Chapter 7, we can conclude that there are some features of the proposed conflict prevention algorithm:

(1) The user satisfaction of the system's desk allocation method is better than ID desk allocation and random desk allocation, but it still can be optimized since there exists one method to get better user satisfaction. To approach the best user satisfaction, the system either remembers a history of user's profiles, or includes user's usual entering time into the profile, so that the system has a certain capability to predict and look to the future.
(2) The average user satisfaction for individuals is 0.8594, but the minimum user satisfaction in a set of 18 profiles varies for difference cases, in the range of 0.35~0.7.
(3) The algorithm performance depends on the entering order of users, so different entering orders result in different desk assignment results, and different user satisfaction for individuals. Usually, the later a profile comes in, the less the user satisfaction it is likely to get.

Chapter 7 also reflected some quality attributes and performance of the proposed system architecture and network. Here we conclude some of its features:

(1) The response time differs for different entering orders, the later an office worker comes in, the quicker he/she can receive an optimal desk. The response time in Z1 motes is bigger than Cooja motes because CPU frequency of Z1 motes is only 16 MHz, in which the complex computation takes more time. However, the response time on Cooja motes reflects the network communication time, which is 61 ms in average. This is an acceptable response time, and it is below than our expectation (91 ms, in Table 5.8).
(2) Desk-controller-to-server delay replies on how many hops when the desk controller node sends a message to the server. Increasing the number of desks and desk controllers does not affect response time of user's room check-in. However, it affects the maximum desk-controller-to-server delay. Since human's average reaction time is 215 ms, the maximum number of desk controller nodes should be 114 in order to make desk-controller-to-server delay below human's average reaction time.

(3) The results in fault tolerance experiment shows that the simulated system fulfills two cases of the non-functional requirement of fault tolerance, especially it can automatically recover from 10ms-interval interferences.
(4) The radio power consumption of Z1 motes are similar among the three types of components, which is approximately 29~30 mW, so the network is energy-efficient.

Overall, the results show that the simulated system fulfills properly the functional and non-functional requirements of the system.

8.2. Discussions

From the comparison with a smart desk assignment method, we can see that the proposed algorithm can be improved in some aspects. For example, if the system remembers the history of a user's usual entrance time, the conflict prevention algorithm could be smarter. It could reserve a desk for a specific person, and wait for him/her, if his/her user satisfaction is greatly better than other people's profiles on the same desk's configuration. The other way is to include the information of individual's usual entrance time into profile. So every office worker inputs his/her usual entrance time into the profile, and then the system remembers every office workers' usual entrance time. The system would assign desk by maximizing the total user satisfaction of the entire office, instead of maximizing an individual's user satisfaction. Since it has the knowledge of future profiles, such a mechanism can improve the total/average user satisfaction of the whole office, and it might make the algorithm independent to the entering order.

For the simulation, we simulated another version for the same system architecture, in which there is a border router connected to the Internet, so that the proposed algorithm is computed on Linux by Python file, instead of running on boards. Besides, the wireless sensor network is able to visit the external world in that case. In such a system, the server node is able to connect to the building infrastructure, so other components of the building infrastructure are able to externally and remotely fetch some data about the luminaires. For example, energy consumption of luminaires can be collected in order to get the total energy consumption of the building, and the occupancy information can be a data source to get the number of attended people in the building. System administrators are able to easily control and maintain this system remotely.

The simulation verifies for the proposed connected lighting system that the conflict prevention algorithm has the potential of being used. The system architecture is also feasible with some good performance and reasonable quality attributes. Appendix C provides three potential business models for such a connected lighting system, which could be a reference for lighting companies to enter the market.

8.3. Future Works

For the personal lighting profile, we only included four attributes. However, actually more attributes can be included into the profile, like friendship, entrance time, teamwork, meeting time, etc. If we involve friendship, we should think about privacy issues. A survey in Appendix A shows some ideas from potential users about the personal lighting profile.

The desk assignment in the proposed system is based on maximizing individual's user satisfaction. The desks may be assigned by other methods, for example, setting the objective as making the minimum user satisfaction in the office not below than a threshold; or assigning desks based on fairness, a simple fairness could be minimizing the gap between maximum and minimum user satisfaction in the office. For different desk assignment algorithm, different system architecture could be proposed and tested.

For this proposed system, the illuminance on a desk is not sensed, instead, we proposed a model to calculate and get the illuminance on desks. This method does not rely on the use of brightness sensors, so it can save the cost of the system. However, this model is not verified in real world, so it would be nice to verify this model in the future, to see the gap between the realistic and the theoretical illuminance value on a desk. If applicable, an optimal algorithm could be proposed to amend this gap.

The proposed system architecture actually has a problem. For example, at a given time instance, a person checks in at the room entrance, and at the same time, one person checks out at the desk, so this desk is still labeled as occupied because it takes about 218 ms to update the occupancy information from a farthest desk controller node in an office with 114 desk controller nodes. However, actually this desk is a candidate for that profile, and it might be an optimal desk. Thus, this person might miss a better result. The problem comes from mutual exclusion for the occupancy data. We need to solve this problem by introducing some typical mutual exclusion devices like semaphore, monitors, etc. Future work can work on finding a good solution for that, or finding a better architecture without this problem.

Different network configurations can be tested for the proposed connected lighting system so that we can compare and decide to use a best mechanism for each layer in the network. Three architectures were proposed for such a system, but due to time limitation, we only simulated the central server style architecture. However, it will be interesting if we can see the rest two architectures be simulated, and see the comparisons among the three architectures in details. Or some other new architectures can be proposed, and compared with the central server style.

Bibliography

[1] Philips Lighting. Connected lighting systems. Available from http://www.lighting.philips.com/main/systems/connected-lighting.html

[2] LaMonica, M. (2014, February 18). Philips Creates Shopping Assistant with LEDs and Smart Phone. *IEEE Spectrum*.

[3] Yamagishi, M., Yamaba, K., Kubo, C., Nokura, K., & Nagata, M. (2008). Effects of LED lighting characteristics on visual performance of elderly people. *Gerontechnology 2008*, 7(2): 243.

[4] Philips Lighting. Application Inspiration - Office LED Lighting. Available from http://www.lighting.philips.com/b-dam/b2b-li/en_AA/Applications/masthead-pdfs/Application-Inspiration-Office-LED-Lighting-Interactive-Guide-INT.pdf

[5] Cheng, H., Lee, K., & Lee, H. (2007, February 19). Color Preference of the Korean Elderly. *IASDR07*, Vol. 33, No. 5, October 2008.

[6] Nelson, R. J., Fonken, L., & Bedrosian, T. (2015). Endocrine Effects of Circadian Disruption. *Annual review of physiology*, 78(1).

[7] EN12464-1 - European standard: Light and lighting - Lighting at work places - Part 1: Indoor Work Places, 2002.

[8] Wyszecki, G., & Stiles, W. S. (1982). *Color science* (Vol. 8). New York: Wiley.

[9] Paschotta, R. (2008). *Encyclopedia of laser physics and technology* (p. 219). Weinheim: Wiley-VCH.

[10] Huizenga, C., Abbaszadeh, S., Zagreus, L., & Arens, E. A. (2006). Air quality and thermal comfort in office buildings: results of a large indoor environmental quality survey. *Center for the Built Environment*.

[11] CSA Standard CAN/CSA Z412-00 (R2011) - Office Ergonomics.

[12] Khalil, K. M., Abdel-Aziz, M., Nazmy, T. T., & Salem, A. B. M. (2014). Intelligent Techniques for Resolving Conflicts of Knowledge in Multi-Agent Decision Support Systems. *International Conference on Intelligent Computing & Information,* 2013, p235.

[13] Newsham, G. R., Veitch, J. A., Reinhart, C. F., & Sander, D. M. (2004). Lighting design for open-plan offices. *Construction Technology Update*, p62.

[14] Wang, X., Jalalirad, A., Tjalkens, T., & Linnartz, J. (2012). Modeling of comfortable and energy-efficient light distributions for indoor environment. *International Conference on the Effects of Light on Wellbeing*. 2012.

[15] Wang, X., Linnartz, J., & Tjalkens, T. (2013). Optimization by adaptive simulated annealing: Intelligent light control using an array of dimmable LEDs. *Proceedings of 34th Symposium on Information Theory in the Benelux*.

[16] Yang, H. (2010). *Signal Processing for LED Lighting Systems: Illumination Rendering and Sensing.* Technische Universiteit Eindhoven.

[17] Thirumal, P., Amirthagadeswaran, K., & Jayabal, S. (2014). Optimization of Indoor Air Quality Characteristics in an Air-Conditioned Car Using Multi-objective Genetic Algorithm. *Arab J Sci Eng Arabian Journal for Science and Engineering,* 8307-8317.

[18] Petrushevski, F., Sipetic, M., & Suter, G. Conflict management in a personalized, space model based lighting control system. *UbiComp '12 Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pp. 568-571.

[19] Bertsekas, D. P. (2014). *Constrained optimization and Lagrange multiplier methods*. Academic press.

[20] Sobkowicz, P., Thelwall, M., Buckley, K., Paltoglou, G., & Sobkowicz, A. (2013). Lognormal distributions of user post lengths in Internet discussions-a consequence of the Weber-Fechner law?. *EPJ Data Science*, 2(1), 1-20.

[21] Agrawal, G. P. (2012). *Fiber-optic communication systems* (Vol. 222). John Wiley & Sons.

[22] Kruchten, P. (1995). Architecture blueprints---the "4 1" view model of software architecture. *Tutorial Proceedings on TRI-Ada '91 Ada's Role in Global Markets: Solutions for a Changing Complex World - TRI-Ada '95*.

[23] Närman, P., Franke, U., König, J., Buschle, M., & Ekstedt, M. (2014). Enterprise architecture availability analysis using fault trees and stakeholder interviews. *Enterprise Information Systems*, 8(1), 1-25.

[24] Chang, B. R., Tsai, H. F., Cheng, J. C., & Tsai, Y. C. (2014). High Availability and High Scalability to in-Cloud Enterprise Resource Planning System. *Intelligent Data analysis and its Applications,* Volume II, pp. 3-13.

[25] Chiu, P. C., Hsu, Y. S., & Chiu, C. T. (2014). Analysis of Cluster Ring Controller/Area Networks for Enhanced Transmission and Fault-Tolerance in Vehicle Networks. *Advanced Microsystems for Automotive Applications 2014*, pp. 101-110.

[26] Tiwari, R. K., & Agrahari, S. K. (2014). ARM Processor Based Embedded System for Remote Data Acquisition. *International Journal of Electronics Communication and Computer Engineering*, 5(1), 8.

[27] Sun, J., Gong, W., Dong, X., Zhang, X., & Wang, Y. (2014, June). High availability analysis and evaluation of heterogeneous dual computer fault-tolerant system. *Software Engineering and Service Science (ICSESS), 2014 5th IEEE International Conference*, pp. 460-464.

[28] Pinedo-Frausto, E., & Garcia-Macias, J. An experimental analysis of Zigbee networks. *2008 33rd IEEE Conference on Local Computer Networks (LCN)*.

[29] Dunkels, A., Schmidt, O., Finne, N., Eriksson, J., Österlind, F., Tsiftes, N., & Durvy, M. (2011). The Contiki OS: The Operating System for the Internet of Things.

[30] Network Stack. (2014, October 27). Available from http://anrg.usc.edu/contiki/index.php/Network_Stack

[31] Change MAC or Radio Duty Cycling Protocols. Available from https://github.com/contiki-os/contiki/wiki/Change-mac-or-radio-duty-cycling-protocols

[32] Sehgal, A. (2013). Using the Contiki Cooja Simulator.

[33] Zolertia. Z1 Datasheet. (2010, March 11). Available from http://zolertia.com/sites/default/files/Zolertia-Z1-Datasheet.pdf

[34] Fisher, Ronald A.; Yates, Frank (1948) [1938]. Statistical tables for biological, agricultural and medical research (3rd ed.). *London: Oliver & Boyd*. pp. 26–27.

[35] Black, P. E. (2005). Fisher-Yates shuffle. *Dictionary of Algorithms and Data Structures*, p.19.

[36] Gonizzi, P., & Duquennoy, S. (2013, September 17). Hands on Contiki OS and Cooja Simulator: Exercises (Part II). Available from https://team.inria.fr/fun/files/2014/04/exercise_partII.pdf

[37] Reaction Time Test. Available from http://www.humanbenchmark.com/tests/reactiontime

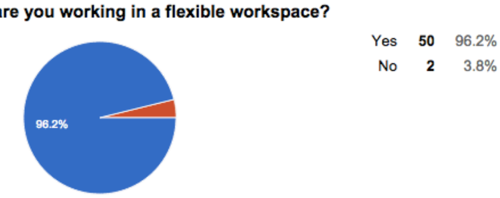
[38] Colina, A., Vives, A., & Bagula, A. (2015, March 1). IoT in 5 days V1.0. Available from http://wireless.ictp.it/school_2015/book/book.pdf

[39] CC2420 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver. Available from http://www.ti.com/lit/ds/symlink/cc2420.pdf

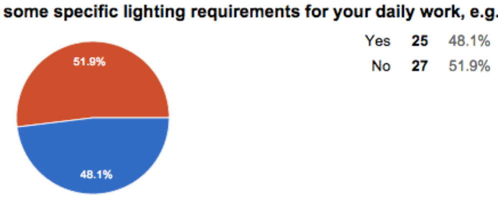
Appendix A Results of a Survey from I&E Thesis

I&E thesis is written in the course of "1ZS30 - Innovation and Entrepreneurship Thesis" (6 ECTS), which is a mandatory course for EIT Digital students to apply Innovation and Entrepreneurship knowledge into practical. In this course, students have to propose three potential business models for the projects they are working in master project, as well as do a dominant step in the Innovative Development Framework. For my I&E thesis, a survey has been conducted online, related to this master project. Following questions are asked to the participants, and they are all within the scope of this master project and they are questions that were risen up during doing the master thesis. Hopefully this survey will also be useful for future work, or the next step of this project. The survey was distributed via Facebook, so the participants are mostly young people (20-30 years old). Finally received 52 useful responses, the summary of the survey is shown below:

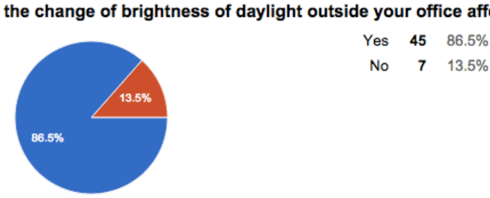
Did you or are you working in a flexible workspace?



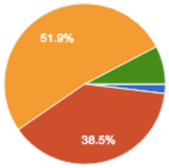
Do you have some specific lighting requirements for your daily work, e.g., illuminance (lux), or color temperature (K) ?



Do you think the change of brightness of daylight outside your office affect your work efficiency?



Do you feel uncomfortable in your office because the luminaries above your head give you unsuitable lights?

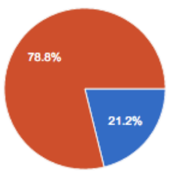


Always	**1**	1.9%
Often	**20**	38.5%
Sometimes	**27**	51.9%
Seldom	**4**	7.7%
Never	**0**	0%

Would you like to adjust the brightness of the luminaire above your head if the lighting system provides such a mechanism?

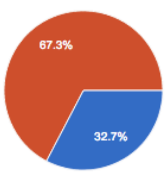
Yes	**52**	100%
No	**0**	0%

Would you like to change a seat during your work?



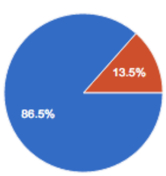
Yes	**11**	21.2%
No	**41**	78.8%

Would you like to consider energy consumption when choosing a seat in a flexible workspace?



Yes	**17**	32.7%
No	**35**	67.3%

Do you think sitting in a specific location matters(e.g., a desk next to a window or next to aisle), in a flexible workspace?

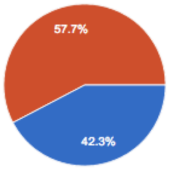


Yes	**45**	86.5%
No	**7**	13.5%

Do you think sitting near familiar colleagues or friends matters, in a flexible workspace?

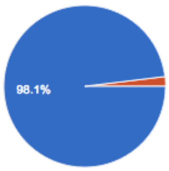
Yes	**52**	100%
No	**0**	0%

Do you think sitting the distance to the entrance of the office or a restroom matters, in a flexible workspace?



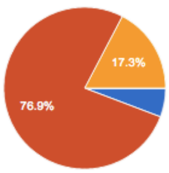
Yes	22	42.3%
No	30	57.7%

If there is a connected lighting system for your office to allocate an optimal seat based on your lighting preferences, would you like to give a try?



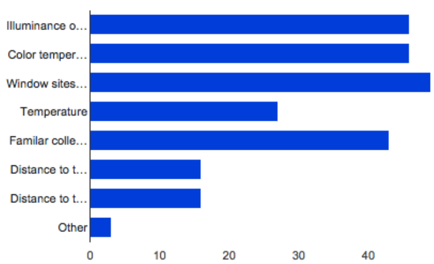
Yes	51	98.1%
No	1	1.9%

How many seconds is endurable for your waiting, after check in the abovementioned system with your smartphone?



<1s	3	5.8%
1-5s	40	76.9%
5-10s	9	17.3%
>10s	0	0%

The following parameters are the potential factors to affect user satisfaction in the abovementioned system. Please tick all the parameters that you think affect your seat decision.



Illuminance of luminaire	46	88.5%
Color temperature of luminaire	46	88.5%
Window sites or aisle sites	49	94.2%
Temperature	27	51.9%
Familar colleagues / friends nearby	43	82.7%
Distance to the entrance	16	30.8%
Distance to the restroom	16	30.8%
Other	3	5.8%

Others: Seat type (soft or hard); Desk type (high or low or flexible); Fengshui, which is a Chinese philosophical system of harmonizing everyone with the surrounding environment so that people feel lucky when sitting in a specific location.

Appendix B Statistics of Experiment Results

In Table B.1, the column names are abbreviated due to the limit of page width, they should be:

ID – Profile ID

US – User Satisfaction

RT-C – Response Time (Cooja motes) from Experiment 3 (ms)

RT-Z – Response Time (Z1 motes) from Experiment 2 (ms)

DD – Desk-controller-to-server Delay (ms)

Table B.1 The statistics of results in Experiment 2 and Experiment 3

ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD
485	0.9466	86	741	17	910	0.8944	22	716	17	105	0.7376	38	716	17	203	0.9487	103	687	17	677	0.907	44	707	16
119	0.9708	88	685	17	459	0.7588	86	674	16	236	0.9297	92	685	17	105	0.7041	62	683	17	432	0.7549	61	688	17
432	0.7549	104	638	16	679	0.954	70	634	17	235	0.93	71	662	17	235	0.93	86	661	17	236	0.9297	57	651	17
236	0.9297	93	613	17	105	0.7376	44	599	17	203	0.9294	50	572	17	119	0.8824	96	611	17	235	0.93	86	620	17
910	0.8944	44	556	17	432	0.7549	42	566	16	485	0.8786	20	589	17	677	0.8315	90	561	16	487	0.9311	28	570	17
459	0.7588	16	527	17	333	0.9628	98	526	17	918	0.9028	90	509	16	570	0.7464	23	547	17	119	0.8824	60	531	17
414	0.9627	10	475	17	614	0.9737	40	500	17	459	0.7588	33	492	17	581	0.9895	18	489	17	459	0.7588	24	492	17
570	0.7177	36	463	17	570	0.7583	23	462	17	570	0.7177	96	463	16	432	0.7549	84	451	16	581	0.9895	62	448	17
679	0.954	74	407	17	235	0.9021	15	430	16	487	0.9313	19	419	17	414	0.9627	22	405	17	679	0.926	41	402	17
333	0.9628	10	369	16	236	0.9297	24	377	17	677	0.8315	72	377	17	485	0.878	38	397	17	333	0.9628	73	374	17
677	0.879	21	337	17	677	0.879	38	342	17	119	0.785	42	341	16	236	0.9185	54	348	17	203	0.9019	28	335	16
203	0.844	107	294	17	581	0.9859	102	303	17	414	0.9325	37	298	17	679	0.954	20	292	17	485	0.8726	93	315	17
918	0.8842	98	257	28	119	0.8516	95	266	27	614	0.9301	20	266	28	918	0.8842	68	259	28	570	0.6965	21	277	27
487	0.841	66	236	28	414	0.9627	30	226	28	910	0.6995	22	218	27	614	0.9737	98	232	28	918	0.9028	67	222	28
105	0.5793	33	189	28	918	0.8118	44	187	28	432	0.5987	26	180	28	333	0.9628	56	180	28	414	0.9098	12	189	28
614	0.9737	26	155	27	485	0.8726	39	160	27	333	0.9392	26	145	26	910	0.8586	32	148	28	910	0.8586	17	150	27
581	0.8798	13	116	28	203	0.844	70	117	27	679	0.7628	82	107	28	459	0.7381	13	113	28	105	0.5793	54	115	28
235	0.9021	41	79	28	487	0.3882	43	80	28	581	0.8798	60	103	28	487	0.3882	42	78	27	614	0.7251	81	111	28
ID	**US**	**RT-C**	**RT-Z**	**DD**	**ID**	**US**	**RT-C**	**RT-Z**	**DD**	**ID**	**US**	**RT-C**	**RT-Z**	**DD**	**ID**	**US**	**RT-C**	**RT-Z**	**DD**	**ID**	**US**	**RT-C**	**RT-Z**	**DD**
105	0.7376	76	715	17	119	0.9708	81	725	16	918	0.9094	39	688	17	414	0.9851	79	696	17	459	0.7588	71	714	17
119	0.9592	10	687	17	235	0.93	86	701	17	333	0.9628	13	680	17	581	0.9895	61	675	17	485	0.9466	49	702	16
432	0.7549	54	644	16	614	0.9737	52	649	17	459	0.7588	65	638	17	679	0.954	97	638	17	570	0.7461	59	657	17
203	0.9019	58	583	17	203	0.9294	46	576	17	485	0.9466	61	621	16	918	0.9028	91	584	17	677	0.879	84	599	16
235	0.93	28	580	17	487	0.9313	108	569	17	235	0.93	82	584	17	235	0.93	53	587	17	105	0.7002	11	560	17
677	0.879	67	523	16	485	0.8786	63	550	17	236	0.9185	38	525	17	614	0.9737	25	540	16	487	0.9313	93	531	17
918	0.9094	15	467	17	677	0.8315	101	489	16	203	0.9487	107	459	16	236	0.9185	106	483	17	432	0.5987	68	486	17
614	0.9737	59	456	17	459	0.7588	55	449	17	614	0.9737	92	458	17	570	0.7583	39	462	17	236	0.9297	83	466	17
679	0.954	64	406	17	236	0.9297	108	426	17	910	0.8586	15	418	17	487	0.9556	77	424	18	119	0.8143	102	417	17
487	0.841	28	386	17	414	0.9627	60	362	16	581	0.9895	10	369	17	677	0.879	29	374	17	235	0.93	46	383	17
570	0.7464	11	350	17	910	0.8586	45	333	17	105	0.7002	41	341	17	432	0.7549	84	345	17	910	0.8586	44	332	16
910	0.8586	80	301	16	679	0.954	54	292	17	570	0.6965	48	308	17	910	0.8586	57	293	17	203	0.844	72	294	17
485	0.484	9	278	27	432	0.7527	107	256	27	487	0.9313	101	269	26	485	0.878	16	279	26	918	0.8842	31	262	28
581	0.9859	38	223	28	581	0.9859	63	223	28	119	0.8824	83	226	28	333	0.9628	81	222	28	581	0.9895	56	223	28
236	0.9185	66	190	28	105	0.5934	59	184	28	414	0.9627	25	193	28	459	0.7381	71	190	27	333	0.9392	59	181	27
333	0.9628	99	142	26	918	0.8118	96	151	28	432	0.593	101	145	28	119	0.7674	94	154	28	614	0.9737	91	152	28
459	0.7381	29	112	28	333	0.9392	105	108	28	677	0.8315	32	116	27	203	0.6825	66	116	28	414	0.7768	66	116	27
414	0.7177	99	78	28	570	0.6965	40	79	27	679	0.9074	52	75	28	105	0.5793	97	79	28	679	0.5546	65	75	28

Table B.1. The statistics of results of Experiment 2 and Experiment 3 (Cont. I)

ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD
581	0.9895	88	705	17	105	0.7376	64	716	17	679	0.954	14	709	17	570	0.7583	21	733	17	432	0.7549	104	720	17
432	0.7549	62	682	17	333	0.9628	45	677	16	918	0.9094	51	653	17	679	0.954	102	674	17	414	0.9851	37	662	17
236	0.9185	106	646	16	614	0.9737	51	647	17	333	0.9628	25	641	17	485	0.9112	54	666	17	485	0.9466	104	667	17
614	0.9737	43	609	17	918	0.9094	16	578	17	105	0.7376	41	604	17	235	0.93	54	625	17	333	0.9628	86	599	17
105	0.7376	44	564	17	236	0.9185	106	567	16	203	0.9294	106	537	17	203	0.9019	30	540	16	614	0.9737	24	572	16
203	0.9294	39	508	16	235	0.93	38	546	17	236	0.9185	100	533	16	581	0.9895	29	523	17	910	0.8944	88	525	16
485	0.8786	49	506	17	119	0.9592	66	495	17	414	0.9851	90	477	17	459	0.7588	40	482	17	459	0.7381	69	483	17
119	0.8706	40	455	17	485	0.8786	79	470	17	432	0.7549	106	458	17	910	0.8586	48	451	17	236	0.9297	33	449	17
459	0.7381	100	415	17	679	0.954	14	411	17	459	0.7381	58	418	17	432	0.7549	107	401	16	203	0.9487	12	390	17
235	0.93	101	386	17	414	0.9851	65	363	16	677	0.879	385	373	16	105	0.6442	10	372	17	105	0.7002	28	375	16
677	0.8315	70	341	17	203	0.9019	17	326	17	485	0.878	25	357	17	918	0.9028	40	335	17	235	0.9237	76	349	17
679	0.954	90	292	16	459	0.7381	75	298	17	235	0.93	104	307	17	487	0.8551	82	305	17	570	0.6652	43	308	17
910	0.8586	70	263	27	487	0.9313	62	273	28	614	0.9737	95	270	28	677	0.8315	107	269	28	119	0.8824	58	265	28
487	0.9214	36	228	28	581	0.9859	31	227	27	581	0.9859	31	228	27	333	0.9628	34	218	28	679	0.9074	77	224	27
570	0.6507	58	197	28	570	0.6507	97	196	28	910	0.8586	23	182	28	236	0.9185	92	191	27	918	0.9028	91	187	28
918	0.8842	100	155	26	432	0.593	18	146	27	487	0.841	58	159	27	119	0.7508	42	149	28	487	0.841	103	159	27
414	0.769	25	116	28	910	0.8586	11	111	28	570	0.6507	87	119	28	414	0.8292	24	115	28	581	0.8697	34	116	28
333	0.9392	55	73	28	677	0.8315	20	88	28	119	0.5304	45	76	28	614	0.7111	81	95	27	677	0.6457	19	77	27
ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD
485	0.9466	66	744	17	119	0.9708	40	729	17	910	0.8944	87	714	17	918	0.9094	69	691	17	918	0.9094	17	691	17
918	0.9094	47	652	16	570	0.7583	86	696	17	581	0.9895	65	668	17	459	0.7588	39	678	16	581	0.9895	97	669	17
203	0.9487	55	825	17	581	0.9895	73	639	16	414	0.9851	101	626	17	487	0.9556	23	648	17	570	0.7583	82	659	17
414	0.9851	84	584	17	333	0.9628	97	599	17	203	0.9487	42	583	16	570	0.7583	73	622	16	333	0.9628	10	601	16
677	0.879	28	556	17	679	0.954	79	569	17	614	0.9737	15	574	17	679	0.954	81	550	17	105	0.7376	31	571	17
679	0.954	67	519	16	235	0.93	102	548	17	485	0.9466	17	550	17	485	0.8786	45	550	17	677	0.879	103	520	17
333	0.9628	77	486	17	432	0.7549	20	486	17	105	0.7002	49	492	17	236	0.9185	17	492	17	487	0.9313	84	506	17
105	0.7002	43	454	17	910	0.8586	70	452	17	677	0.879	91	455	17	432	0.7549	84	452	17	614	0.9737	96	457	17
119	0.8143	57	418	18	614	0.9737	69	425	16	487	0.9313	28	418	16	333	0.9628	66	414	17	432	0.7527	48	418	17
487	0.9313	97	389	17	487	0.9214	44	382	17	679	0.954	38	376	17	910	0.8944	210	368	16	119	0.8706	103	380	16
459	0.7381	69	338	17	459	0.7588	16	338	17	459	0.7381	309	334	17	105	0.7002	17	333	17	414	0.9627	67	333	17
235	0.93	29	308	17	414	0.9627	56	299	17	918	0.9028	57	294	17	581	0.9859	79	295	17	485	0.8726	57	313	17
570	0.6652	56	274	28	236	0.9185	12	264	28	236	0.9185	92	266	28	614	0.9737	45	266	27	679	0.9324	89	260	27
581	0.9859	91	227	28	485	0.8786	34	238	28	119	0.7674	98	230	28	203	0.844	87	222	28	910	0.8586	32	224	28
614	0.9737	68	195	28	918	0.8842	44	186	27	432	0.5987	92	183	28	677	0.8315	84	190	27	235	0.8902	52	197	28
432	0.593	12	139	27	105	0.5793	78	157	27	235	0.9021	36	151	27	414	0.9627	26	154	27	236	0.9185	44	150	28
236	0.8391	23	114	26	677	0.8315	35	113	28	333	0.9392	58	108	28	119	0.7071	49	116	28	203	0.844	64	117	28
910	0.6572	100	75	28	203	0.844	82	78	28	570	0.5884	56	79	27	235	0.8516	36	95	28	459	0.6051	53	76	27
ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD
432	0.7549	11	721	17	485	0.9466	27	742	17	203	0.9487	38	685	17	414	0.9851	75	698	17	203	0.9487	52	686	16
105	0.7376	27	679	17	414	0.9851	101	659	17	119	0.9592	18	690	17	485	0.9466	27	704	16	485	0.9466	50	704	17
414	0.9851	23	621	16	487	0.9556	97	646	17	432	0.7549	86	643	17	570	0.7461	184	656	16	235	0.93	39	663	17
679	0.954	9	596	17	236	0.9297	105	609	16	485	0.8786	60	632	17	487	0.9313	39	615	17	614	0.9737	95	609	17
614	0.9737	95	573	17	105	0.7002	9	562	17	581	0.9895	54	561	17	910	0.8944	29	556	17	119	0.8824	104	571	16
235	0.93	36	546	17	614	0.9737	307	531	17	679	0.954	81	523	17	203	0.9019	30	510	17	581	0.9895	36	524	17
570	0.7583	88	505	17	570	0.6965	98	502	16	333	0.9628	245	480	16	679	0.954	70	487	16	677	0.8315	31	486	17
918	0.9094	16	440	17	679	0.954	9	443	17	570	0.7464	97	468	17	459	0.7381	73	446	17	432	0.7549	58	444	16
119	0.8824	57	419	16	910	0.8944	40	409	17	910	0.8944	11	408	17	235	0.9058	80	431	17	459	0.7588	50	413	16
459	0.7381	51	377	17	677	0.879	57	378	17	487	0.9214	53	386	17	236	0.9297	62	377	17	105	0.7002	71	373	17
485	0.878	74	356	17	203	0.8811	12	329	16	459	0.7381	92	336	17	918	0.9028	18	336	17	910	0.8586	26	335	17
910	0.8586	96	296	17	432	0.5987	107	294	17	235	0.9058	39	314	17	105	0.6442	66	299	18	487	0.4926	63	300	17
581	0.9859	67	261	28	459	0.7011	40	261	27	918	0.9028	93	260	28	333	0.9628	91	260	28	236	0.905	51	268	27
333	0.9628	83	218	28	235	0.9021	50	235	28	105	0.6442	58	225	27	581	0.8798	101	223	27	918	0.9028	56	225	28
677	0.8315	72	189	28	581	0.9859	18	189	28	236	0.9185	103	189	27	119	0.8824	59	189	28	333	0.9628	87	184	28
487	0.4926	67	158	27	918	0.9028	105	150	28	677	0.8315	108	153	26	432	0.593	104	145	28	570	0.6965	35	159	27
203	0.81	43	116	27	333	0.9392	18	107	27	414	0.9013	36	116	28	677	0.8315	32	116	28	679	0.954	15	108	28
236	0.8838	86	74	28	119	0.7674	12	77	28	614	0.7111	34	95	28	614	0.9301	40	112	27	414	0.7177	13	78	28

Table B.1. The statistics of results of Experiment 2 and Experiment 3 (Cont. II)

ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD
581	0.9895	45	705	17	910	0.8944	77	712	17	333	0.9628	53	716	16	236	0.9297	105	719	17	236	0.9297	35	719	17
570	0.7583	77	697	16	614	0.9737	29	686	17	459	0.7588	93	674	17	614	0.9737	71	685	17	235	0.93	57	704	16
414	0.9851	12	624	17	235	0.9237	32	667	16	203	0.9487	31	611	17	487	0.9556	33	648	16	614	0.9737	83	648	17
679	0.954	53	602	17	918	0.9094	99	589	17	570	0.7583	88	618	17	679	0.954	106	591	17	581	0.9895	93	590	17
105	0.7376	35	571	17	459	0.7588	23	561	17	236	0.9297	106	576	17	105	0.7041	64	559	17	485	0.9466	26	586	17
487	0.9313	83	539	16	432	0.7549	101	528	16	235	0.93	16	546	16	485	0.8786	96	548	16	677	0.907	60	524	16
918	0.9028	10	473	17	485	0.9466	45	511	17	105	0.7002	50	492	17	918	0.9028	108	473	17	918	0.9028	85	473	17
910	0.8944	86	442	16	203	0.9487	40	421	17	918	0.9028	66	433	17	333	0.9628	102	456	17	203	0.9019	66	432	16
235	0.9058	49	432	17	679	0.9074	94	407	17	432	0.7549	38	418	17	203	0.9019	48	391	17	105	0.6972	22	413	17
485	0.8786	85	399	17	105	0.7002	21	370	17	677	0.879	16	376	17	235	0.93	33	388	17	679	0.9172	31	370	17
203	0.9019	61	329	16	570	0.6965	30	347	17	679	0.954	105	330	16	432	0.7549	87	339	16	414	0.9098	73	328	16
333	0.9628	35	296	17	487	0.8551	45	302	17	119	0.7674	100	303	17	414	0.9851	38	290	17	487	0.8551	109	310	17
677	0.8315	76	263	27	119	0.8143	77	269	28	414	0.9325	37	262	28	570	0.6965	25	275	28	333	0.9628	89	261	28
459	0.7381	42	221	26	333	0.9628	48	215	28	485	0.4891	73	238	28	581	0.9859	11	223	28	459	0.7381	48	224	28
614	0.9301	68	195	27	581	0.8798	58	189	28	487	0.4926	41	195	27	119	0.7674	103	190	26	432	0.593	55	186	26
119	0.7554	44	153	27	414	0.9013	34	151	27	614	0.9032	45	156	28	677	0.6457	69	151	27	910	0.8586	17	148	27
432	0.593	66	105	28	236	0.8838	18	113	28	910	0.616	20	111	28	459	0.7011	76	112	28	570	0.6652	40	120	28
236	0.9185	101	73	28	677	0.8315	88	80	28	581	0.9078	75	77	28	910	0.616	101	103	28	119	0.7674	17	77	28
ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD
485	0.9466	11	742	16	459	0.7588	39	715	17	235	0.93	43	743	17	487	0.9556	9	725	17	910	0.8944	89	713	17
105	0.7376	85	677	17	485	0.9466	65	702	17	333	0.9628	59	675	17	679	0.954	45	669	16	235	0.9237	75	706	17
679	0.954	56	632	17	414	0.9658	74	624	17	485	0.9466	30	665	16	235	0.93	58	663	17	677	0.907	65	638	17
235	0.93	80	625	17	119	0.9708	30	609	17	614	0.9737	55	613	17	119	0.9592	98	609	17	581	0.9895	9	589	16
432	0.7549	37	564	17	235	0.93	26	590	17	119	0.9708	76	574	17	677	0.879	83	559	17	333	0.9628	89	567	17
414	0.9851	29	507	17	570	0.7177	54	541	17	581	0.9895	54	523	18	105	0.7002	223	523	17	414	0.9851	24	511	17
677	0.879	22	490	16	679	0.954	71	490	17	910	0.8586	93	492	16	570	0.7464	55	507	17	485	0.9466	12	510	17
918	0.9094	77	436	17	614	0.9737	51	463	17	570	0.7464	61	462	17	333	0.9628	85	450	17	918	0.9028	72	436	17
910	0.8586	87	409	17	677	0.879	32	414	17	679	0.954	50	421	17	614	0.9737	37	424	17	487	0.9313	18	422	16
581	0.9859	98	374	16	910	0.8586	98	382	17	459	0.7381	94	368	17	918	0.9094	21	367	17	570	0.7177	48	388	17
459	0.7381	10	336	17	105	0.6972	45	339	17	677	0.879	81	340	17	236	0.9185	14	344	16	432	0.593	90	338	17
333	0.9628	86	296	17	918	0.9028	69	294	17	487	0.9313	23	305	17	459	0.7381	28	307	17	105	0.6442	85	304	16
570	0.7177	36	271	27	581	0.9859	107	261	28	918	0.9028	78	256	28	485	0.8726	168	278	28	614	0.9737	37	268	28
236	0.9185	74	228	28	487	0.841	93	234	28	414	0.9098	48	219	28	432	0.7549	93	223	28	203	0.8487	38	225	28
203	0.81	87	187	28	203	0.844	60	189	28	236	0.9185	51	189	28	203	0.81	96	182	28	236	0.9185	10	189	27
614	0.9737	67	156	28	432	0.5987	13	142	28	432	0.7527	25	146	28	581	0.9859	90	151	27	459	0.7381	42	154	28
119	0.7674	58	115	28	236	0.9185	73	113	28	203	0.8074	91	117	27	414	0.9013	60	117	28	679	0.8738	95	110	28
487	0.3882	49	79	27	333	0.9392	39	95	28	105	0.5793	68	111	28	910	0.616	64	74	28	119	0.7674	14	77	28
ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD
333	0.9628	105	716	17	459	0.7588	17	716	17	918	0.9094	57	690	16	677	0.907	45	708	17	459	0.7588	107	716	17
119	0.9708	49	688	17	570	0.7583	269	697	17	119	0.9708	85	688	17	918	0.9094	82	650	17	236	0.9297	42	681	17
910	0.8944	77	639	17	203	0.9487	40	616	16	333	0.9628	85	640	17	487	0.9313	9	651	16	679	0.954	62	629	16
432	0.7549	53	602	17	679	0.954	10	597	17	432	0.7549	93	610	17	679	0.954	59	590	17	333	0.9628	51	605	17
203	0.9294	97	548	17	119	0.8824	94	571	17	570	0.7583	17	579	16	581	0.9859	38	550	17	614	0.9737	265	573	17
614	0.9737	75	534	17	910	0.8944	63	525	17	910	0.8944	275	521	17	459	0.7588	48	529	17	918	0.9028	34	515	17
487	0.9214	13	494	16	677	0.8315	69	491	16	203	0.9019	36	476	17	614	0.9737	38	496	16	105	0.7376	108	495	16
679	0.954	100	452	17	485	0.878	96	478	17	614	0.9737	9	456	17	432	0.7527	64	457	17	677	0.879	20	453	17
105	0.6442	85	419	17	414	0.9658	54	401	17	235	0.9237	56	428	17	570	0.7583	49	424	17	487	0.9313	12	416	18
485	0.8786	27	393	17	432	0.7549	62	368	17	459	0.7588	14	374	17	119	0.8824	68	382	16	119	0.9592	21	381	17
677	0.8315	70	339	17	581	0.9859	73	335	17	485	0.8786	18	355	16	333	0.9628	91	334	17	203	0.9019	88	327	17
570	0.7464	98	312	17	918	0.9028	102	297	17	679	0.9074	89	295	17	203	0.844	27	298	18	235	0.93	77	311	17
581	0.9895	10	266	28	236	0.9185	64	270	28	487	0.841	73	267	28	485	0.8726	43	275	28	485	0.8726	75	276	28
235	0.9021	101	231	27	487	0.4866	83	233	28	105	0.5793	100	220	27	910	0.8944	78	222	28	432	0.593	94	222	28
459	0.7381	31	184	27	614	0.9737	59	188	28	236	0.9185	92	190	28	235	0.9021	34	195	28	570	0.6965	74	197	28
236	0.9185	83	152	26	235	0.9021	58	155	27	581	0.9078	25	147	26	236	0.9185	95	151	27	581	0.9859	80	151	28
414	0.9627	51	117	28	105	0.5934	93	111	28	414	0.9627	95	108	28	414	0.9627	58	108	28	414	0.9013	80	116	28
918	0.7748	33	78	28	333	0.9392	105	73	28	677	0.6457	98	95	27	105	0.5793	85	87	28	910	0.5727	51	74	27

Table B.1. The statistics of results of Experiment 2 and Experiment 3 (Cont. III)

ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD
614	0.9737	71	726	17	203	0.9487	11	686	17	432	0.7549	93	722	16	414	0.9851	21	697	17	105	0.7376	107	717	17
203	0.9487	23	648	17	910	0.8944	25	676	17	414	0.9851	41	660	17	236	0.9297	89	684	16	614	0.9737	53	688	17
333	0.9628	89	637	16	333	0.9628	27	637	17	236	0.9297	25	647	17	105	0.7376	72	644	17	485	0.9466	30	664	17
432	0.7549	50	608	17	614	0.9737	72	611	17	677	0.907	66	596	17	679	0.954	12	595	17	414	0.9851	25	581	17
910	0.8944	61	564	17	679	0.954	41	569	16	203	0.9294	24	543	17	119	0.9592	50	572	16	119	0.8824	58	570	16
677	0.879	73	522	17	105	0.7041	29	535	17	918	0.9028	60	509	16	432	0.7549	9	531	17	677	0.8315	93	526	17
679	0.954	65	489	17	487	0.9313	26	496	17	581	0.9895	17	487	17	459	0.7381	17	491	17	203	0.9019	19	475	16
459	0.7588	55	447	17	459	0.7588	88	448	18	235	0.93	72	463	17	487	0.9214	97	460	17	235	0.93	98	464	17
414	0.9658	84	397	16	414	0.9658	98	398	17	485	0.8786	83	431	17	677	0.879	65	413	17	918	0.9094	42	397	16
235	0.9021	66	391	17	432	0.7527	64	376	17	459	0.7381	39	379	17	203	0.8195	79	368	16	910	0.8586	23	370	17
119	0.9592	84	345	17	236	0.9185	62	343	16	105	0.7002	51	339	16	581	0.9859	93	338	17	432	0.7549	52	334	16
487	0.9214	83	308	17	235	0.9021	81	313	17	487	0.841	94	311	17	333	0.9628	42	295	17	333	0.9628	78	293	17
918	0.9028	104	256	28	485	0.8786	26	277	27	910	0.8586	71	253	28	614	0.9737	81	266	28	679	0.954	76	260	28
236	0.9185	49	227	28	918	0.9028	44	222	28	570	0.6652	13	239	28	570	0.6507	31	235	27	581	0.9859	47	227	28
485	0.8726	26	199	28	677	0.8315	63	190	28	614	0.9737	33	194	27	918	0.8739	96	187	28	459	0.7381	91	183	28
105	0.6442	41	150	27	581	0.9859	11	152	28	679	0.9074	53	145	28	910	0.8944	56	145	27	236	0.9185	95	152	28
570	0.6965	27	121	27	119	0.7278	55	115	27	333	0.9628	42	108	28	485	0.8726	23	122	28	570	0.6507	73	120	28
581	0.8697	105	80	28	570	0.5266	11	80	28	119	0.5304	12	111	28	235	0.8516	12	95	28	487	0.3882	101	79	27
ID	**US**	**RT-C**	**RT-Z**	**DD**	**ID**	**US**	**RT-C**	**RT-Z**	**DD**	**ID**	**US**	**RT-C**	**RT-Z**	**DD**	**ID**	**US**	**RT-C**	**RT-Z**	**DD**	**ID**	**US**	**RT-C**	**RT-Z**	**DD**
614	0.9737	67	726	17	581	0.9895	33	706	17	414	0.9851	103	698	16	432	0.7549	45	719	17	581	0.9895	43	706	17
918	0.9094	100	656	17	432	0.7549	61	683	17	235	0.93	96	706	17	677	0.907	65	669	17	235	0.93	41	706	16
570	0.7583	30	654	16	105	0.7376	108	642	16	459	0.7381	94	637	17	119	0.9592	19	649	17	910	0.8586	98	640	18
119	0.9708	74	614	17	677	0.879	97	595	17	910	0.8586	85	603	17	333	0.9628	93	602	17	432	0.7549	73	603	17
581	0.9895	20	562	16	119	0.9592	93	572	17	570	0.7583	51	580	16	679	0.954	13	558	16	487	0.9556	83	569	16
677	0.879	78	520	16	487	0.9214	50	542	16	918	0.9094	49	516	17	235	0.93	94	545	17	333	0.9628	54	526	17
235	0.93	45	506	17	910	0.8944	99	479	17	203	0.9487	64	466	17	910	0.8586	106	490	17	414	0.9851	44	476	17
487	0.9313	18	465	17	679	0.954	92	447	17	581	0.9895	69	451	17	918	0.9094	80	437	16	119	0.9592	88	457	16
679	0.9172	85	407	17	235	0.9064	48	425	17	614	0.9737	47	421	16	487	0.9214	96	420	17	105	0.7002	46	417	17
485	0.878	101	393	16	570	0.6965	96	391	17	487	0.9313	94	387	17	570	0.7464	38	390	17	918	0.9028	24	364	17
203	0.844	12	337	17	459	0.7011	100	335	17	105	0.7002	105	340	17	614	0.9737	18	347	17	677	0.879	10	333	17
910	0.8586	32	294	17	333	0.9628	74	291	17	236	0.9185	17	304	17	236	0.9185	23	303	17	459	0.7381	35	299	17
459	0.7011	82	264	28	236	0.905	84	266	28	677	0.879	90	264	27	105	0.6442	44	267	26	679	0.954	100	259	27
236	0.905	45	227	28	485	0.8726	18	239	28	432	0.5987	29	223	28	581	0.9078	20	224	27	203	0.8195	89	227	28
105	0.5934	9	187	27	918	0.9028	72	189	28	333	0.9392	65	180	28	485	0.8786	100	199	28	485	0.8726	62	200	28
414	0.9627	14	146	28	614	0.9737	45	156	27	119	0.8143	89	150	28	459	0.6546	69	151	28	570	0.5884	60	160	28
333	0.9392	43	114	28	203	0.7681	64	117	28	679	0.9074	50	113	27	414	0.7177	76	111	27	614	0.9737	76	119	28
432	0.5141	89	103	28	414	0.7177	28	78	28	485	0.8726	88	82	28	203	0.8074	50	78	28	236	0.8838	81	95	27
ID	**US**	**RT-C**	**RT-Z**	**DD**	**ID**	**US**	**RT-C**	**RT-Z**	**DD**	**ID**	**US**	**RT-C**	**RT-Z**	**DD**	**ID**	**US**	**RT-C**	**RT-Z**	**DD**	**ID**	**US**	**RT-C**	**RT-Z**	**DD**
918	0.9094	11	693	17	570	0.7583	62	734	17	203	0.9487	97	685	17	485	0.9466	30	740	17	487	0.9556	81	724	17
614	0.9737	19	687	17	614	0.9737	53	690	17	570	0.7583	37	696	16	333	0.9628	44	674	17	679	0.954	39	668	17
487	0.9556	70	649	17	236	0.9297	103	644	17	677	0.879	99	633	17	119	0.9708	221	648	17	119	0.9592	22	647	17
910	0.8944	47	599	17	235	0.93	39	626	17	432	0.7549	30	604	17	235	0.93	66	623	17	235	0.93	22	623	17
581	0.9895	39	551	16	487	0.9556	69	573	17	235	0.93	96	582	17	432	0.7549	15	559	16	614	0.9737	61	575	17
119	0.9592	54	533	17	910	0.8586	39	527	17	485	0.878	92	550	17	910	0.8586	66	525	17	432	0.7549	99	526	16
235	0.9064	97	507	17	414	0.9851	9	477	16	459	0.7588	73	491	17	459	0.7588	37	485	17	581	0.9895	65	482	17
236	0.9185	90	452	17	203	0.9019	18	431	17	119	0.8606	12	456	17	105	0.7002	64	446	16	910	0.8586	31	449	17
677	0.879	64	412	16	485	0.8786	24	433	17	236	0.9185	11	423	16	570	0.7274	20	427	17	459	0.7588	107	415	17
679	0.9074	9	364	17	918	0.9028	96	369	17	918	0.9028	95	365	16	236	0.9185	103	383	16	105	0.7002	98	374	17
203	0.9019	17	326	17	679	0.954	44	333	16	414	0.9401	77	336	17	581	0.9895	101	334	17	485	0.8786	22	359	17
432	0.7549	67	303	16	459	0.7381	16	297	17	910	0.8586	33	289	17	203	0.8487	109	298	17	570	0.6965	44	314	17
570	0.6965	97	275	28	119	0.7745	38	270	26	679	0.926	31	251	28	487	0.841	26	270	27	677	0.8315	77	270	28
333	0.9628	63	224	27	432	0.5987	49	222	28	105	0.6442	93	224	27	414	0.9627	74	222	28	414	0.9627	14	226	28
459	0.6546	79	184	28	581	0.9859	93	186	28	581	0.9078	22	186	28	918	0.8842	108	189	28	918	0.8842	38	189	27
485	0.8726	45	159	27	677	0.8315	50	154	28	333	0.9628	68	149	27	614	0.9737	96	151	28	203	0.8195	10	151	28
105	0.5793	25	115	28	333	0.9392	92	108	27	614	0.7111	24	118	28	679	0.7728	30	108	27	333	0.9628	20	108	28
414	0.9013	42	78	28	105	0.4786	94	111	28	487	0.841	71	78	28	677	0.5975	78	79	28	236	0.8838	61	88	27

Table B.1. The statistics of results of Experiment 2 and Experiment 3 (Cont. IV)

ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD
203	0.9487	50	683	17	414	0.9851	40	696	17	119	0.9708	104	726	17	918	0.9094	21	693	16	679	0.954	87	709	17
487	0.9313	59	688	17	918	0.9094	94	658	17	236	0.9297	73	685	17	570	0.7583	11	694	17	485	0.9466	101	708	17
570	0.7583	34	657	17	677	0.907	102	630	16	485	0.9466	89	662	17	679	0.954	105	632	17	677	0.907	42	637	17
236	0.9297	85	614	16	485	0.9466	95	624	17	333	0.9628	48	598	17	459	0.7588	98	602	17	432	0.7549	91	602	17
614	0.9737	70	572	17	333	0.9628	66	568	17	487	0.9313	36	577	16	487	0.9556	86	573	16	459	0.7588	93	565	16
485	0.8786	89	546	17	236	0.9185	96	531	17	679	0.954	58	521	17	203	0.9019	66	503	17	236	0.9297	90	532	17
910	0.8944	72	484	17	235	0.93	85	503	17	235	0.93	64	509	17	119	0.8824	22	495	17	235	0.93	44	504	17
235	0.9237	62	464	17	614	0.9737	19	457	17	918	0.9028	70	437	16	614	0.9737	92	458	16	105	0.7002	80	441	17
918	0.9028	109	403	17	119	0.8824	24	417	17	614	0.9737	40	422	17	581	0.9859	56	408	17	614	0.9737	14	421	17
459	0.7588	38	375	18	459	0.7381	102	372	16	581	0.9859	39	369	17	236	0.9185	24	382	16	570	0.6965	99	387	17
119	0.8516	95	341	17	432	0.7549	79	338	17	203	0.9019	34	326	16	414	0.9627	89	333	17	487	0.8551	87	344	17
333	0.9628	48	300	17	105	0.7002	83	300	17	459	0.7588	15	303	17	677	0.8315	78	301	17	333	0.9628	12	292	16
414	0.9098	63	257	28	203	0.81	76	257	28	432	0.593	31	263	28	432	0.7549	88	261	28	203	0.8195	41	260	28
677	0.8315	88	229	28	570	0.6191	24	237	28	910	0.8586	85	217	27	485	0.4891	61	240	28	414	0.9627	22	223	27
581	0.9078	46	186	26	910	0.7811	99	180	28	414	0.8292	13	193	28	910	0.8944	92	181	28	119	0.7674	76	193	28
105	0.5793	96	150	28	679	0.9074	47	153	27	105	0.6107	50	156	28	105	0.5934	29	150	28	918	0.8842	48	149	28
679	0.8738	15	107	27	581	0.8697	37	116	28	570	0.6652	79	118	27	235	0.8516	38	117	28	581	0.8697	63	117	28
432	0.4892	37	80	28	487	0.841	55	78	28	677	0.8315	78	77	28	333	0.7892	87	80	27	910	0.8586	78	75	28

ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD
235	0.93	45	742	17	581	0.9895	104	705	17	203	0.9487	93	685	17	485	0.9466	40	740	17	459	0.7588	69	717	17
910	0.8586	47	679	17	333	0.9628	59	676	17	414	0.9851	53	658	17	677	0.907	47	671	17	677	0.907	68	673	17
487	0.9556	32	642	17	203	0.9487	9	623	17	485	0.9466	25	665	17	414	0.9851	76	621	17	414	0.9658	20	625	17
119	0.9592	104	609	17	487	0.9313	93	618	17	119	0.8824	22	608	17	918	0.9094	37	579	17	236	0.9185	36	607	17
432	0.7549	77	565	17	236	0.9185	76	574	16	432	0.7549	31	558	16	119	0.8824	14	571	17	235	0.93	34	585	16
414	0.9851	54	512	17	414	0.9851	52	513	18	679	0.954	59	524	17	235	0.93	37	542	17	432	0.7549	46	532	17
485	0.8786	95	485	17	910	0.8944	22	481	17	614	0.9737	64	494	17	432	0.7549	107	487	16	105	0.7041	76	485	17
459	0.7381	37	451	16	119	0.9592	19	456	17	487	0.9214	86	458	17	105	0.7002	38	446	17	333	0.9628	9	453	16
677	0.8315	12	415	17	432	0.7527	102	415	17	918	0.9094	13	397	16	570	0.6965	76	430	17	679	0.954	95	408	17
333	0.9628	32	367	17	677	0.879	48	371	17	581	0.9859	51	373	17	236	0.9185	24	385	17	570	0.7177	53	386	17
679	0.954	60	337	17	105	0.6442	56	342	17	459	0.7381	92	337	17	203	0.81	92	330	17	614	0.9737	95	347	16
570	0.7274	23	314	17	614	0.9737	69	304	17	677	0.8315	15	303	17	581	0.9895	104	301	17	119	0.8824	40	307	17
105	0.7002	76	263	28	485	0.8726	41	274	28	235	0.93	26	269	28	910	0.8586	52	253	28	918	0.9028	78	255	27
236	0.9185	83	227	27	918	0.9028	27	215	28	570	0.6652	30	237	27	614	0.9737	46	229	27	487	0.841	79	234	28
581	0.9859	59	190	28	570	0.6965	29	196	27	910	0.8586	76	181	28	487	0.841	36	195	27	910	0.8586	73	188	27
203	0.81	81	152	27	679	0.954	81	147	28	333	0.9628	79	142	28	459	0.6384	102	152	26	581	0.9859	73	151	28
614	0.9737	69	119	27	235	0.9021	102	113	28	236	0.8838	59	113	28	679	0.9074	39	113	28	485	0.4794	49	123	28
918	0.7748	109	87	28	459	0.6051	86	76	28	105	0.4098	10	76	28	333	0.9392	106	73	28	203	0.596	24	78	28

ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD
918	0.9094	27	690	16	105	0.7376	55	715	17	614	0.9737	11	727	17	119	0.9708	39	726	17	432	0.7549	35	722	17
236	0.9185	13	683	17	487	0.9313	63	686	17	459	0.7588	48	677	17	581	0.9895	23	677	17	485	0.9466	99	703	17
105	0.7376	18	643	17	581	0.9895	81	630	17	570	0.7583	34	657	17	333	0.9628	50	638	16	614	0.9737	42	646	16
235	0.93	10	621	17	614	0.9737	104	613	17	910	0.8944	100	605	16	487	0.9313	66	621	17	203	0.9487	82	579	17
614	0.9737	9	572	18	432	0.7527	47	571	16	679	0.954	102	562	17	918	0.9028	56	544	16	581	0.9895	52	558	17
677	0.879	99	525	17	910	0.8944	62	521	17	414	0.9658	70	512	17	105	0.7041	12	532	17	333	0.9628	14	521	17
459	0.7588	72	494	17	459	0.7588	72	492	17	485	0.9112	57	515	17	679	0.954	248	484	17	414	0.9851	9	473	17
119	0.9592	56	457	17	414	0.9627	38	440	17	203	0.9019	65	436	17	614	0.9737	93	457	17	677	0.879	94	444	17
485	0.878	50	432	17	918	0.8842	25	408	16	487	0.9313	87	419	16	459	0.7588	20	411	17	910	0.8944	46	407	17
487	0.9313	45	376	16	485	0.9466	70	395	17	432	0.5987	30	367	17	910	0.8944	10	373	17	105	0.7002	49	377	16
910	0.8586	40	331	17	677	0.879	95	342	17	333	0.9392	12	332	17	235	0.9021	93	355	16	679	0.954	93	343	17
432	0.5987	28	299	17	235	0.9237	89	310	16	119	0.8824	78	308	18	203	0.9019	32	292	17	570	0.6965	57	312	17
679	0.954	26	250	28	679	0.8738	42	257	28	581	0.9859	28	272	27	414	0.9098	77	257	28	459	0.7011	92	260	28
414	0.9627	93	219	28	203	0.844	103	227	28	677	0.8315	107	226	28	677	0.879	49	226	27	236	0.905	20	227	28
581	0.9859	27	185	27	119	0.7674	45	194	27	105	0.4876	95	194	27	485	0.8726	73	199	28	487	0.841	92	198	28
333	0.9392	68	142	28	236	0.9185	99	152	27	235	0.9021	97	155	28	570	0.6965	56	159	27	918	0.9028	91	151	27
203	0.844	40	119	27	570	0.5563	54	119	28	236	0.9185	100	112	28	432	0.593	15	111	27	235	0.9021	40	118	26
570	0.6147	86	103	28	333	0.9392	86	103	28	918	0.9028	38	95	28	236	0.9185	84	88	28	119	0.5304	80	77	28

Table B.1. The statistics of results of Experiment 2 and Experiment 3 (Cont. V)

ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD
485	0.9466	9	742	16	677	0.907	97	705	17	570	0.7583	227	732	17	614	0.9737	53	726	17	333	0.9628	97	713	17
677	0.907	84	674	17	485	0.9466	23	702	17	581	0.9895	79	671	17	105	0.7376	49	681	17	236	0.9297	43	683	17
487	0.9313	81	643	17	581	0.9895	82	627	17	235	0.93	250	665	17	203	0.9294	86	611	17	119	0.9708	45	648	17
570	0.7464	78	617	17	236	0.9185	31	608	17	105	0.7376	44	609	17	677	0.879	77	599	17	614	0.9737	77	615	17
333	0.9628	48	562	16	570	0.7464	27	578	16	333	0.9628	78	562	16	333	0.9628	45	565	16	581	0.9895	97	563	16
581	0.9895	30	518	17	918	0.9028	55	505	17	459	0.7588	46	524	17	570	0.7464	11	541	17	910	0.8944	105	526	17
105	0.7002	36	490	17	333	0.9628	95	490	16	119	0.8824	19	495	16	487	0.9313	42	502	16	203	0.9294	78	474	17
432	0.593	59	449	17	203	0.9019	47	440	17	203	0.9019	65	444	17	679	0.954	80	451	17	432	0.7549	44	455	17
910	0.8944	26	407	17	119	0.8824	93	419	17	614	0.9737	16	419	17	414	0.9658	98	393	17	487	0.9214	32	424	17
203	0.8487	101	378	17	235	0.93	13	386	17	485	0.8726	44	394	17	459	0.7381	100	372	17	570	0.7464	20	391	17
414	0.9627	77	325	16	614	0.9737	13	343	16	414	0.9627	82	325	16	236	0.9185	53	344	16	235	0.9021	20	350	17
459	0.7381	47	301	17	105	0.6972	45	308	17	910	0.8586	75	296	17	235	0.93	34	309	17	918	0.9028	21	295	17
918	0.8842	53	257	28	414	0.9401	38	260	28	432	0.7549	45	252	28	485	0.878	63	278	27	414	0.9098	58	265	27
236	0.9185	64	234	27	679	0.9074	54	224	27	677	0.6302	42	231	27	918	0.9028	75	224	28	105	0.6107	82	231	28
679	0.954	22	181	28	910	0.8586	49	182	28	487	0.4926	87	190	28	910	0.8586	97	186	28	677	0.8315	12	190	28
614	0.9737	41	154	26	432	0.593	82	145	28	918	0.853	22	155	28	119	0.7674	53	154	28	485	0.4794	92	162	28
119	0.7674	93	115	28	459	0.7381	73	113	28	236	0.9042	10	113	27	581	0.9859	14	111	28	679	0.7728	59	113	27
235	0.9021	84	79	27	487	0.841	49	82	27	679	0.7728	58	87	28	432	0.593	10	79	28	459	0.7011	90	76	28
ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD
432	0.7549	49	721	17	119	0.9708	226	726	17	485	0.9466	73	739	17	459	0.7588	37	714	17	679	0.954	27	708	17
235	0.93	107	705	16	105	0.7041	267	682	17	333	0.9628	57	673	17	487	0.9556	67	682	17	414	0.9851	68	660	17
333	0.9628	27	639	17	487	0.9313	28	649	16	105	0.7376	102	639	16	485	0.9466	102	663	17	581	0.9895	16	632	17
459	0.7588	12	599	17	203	0.9019	31	583	16	487	0.9313	63	610	17	432	0.7549	103	601	16	910	0.8944	98	600	17
485	0.9466	72	587	16	333	0.9628	93	563	17	614	0.9737	60	570	17	333	0.9628	28	562	17	235	0.9058	83	592	16
236	0.9297	82	528	17	677	0.879	65	519	17	414	0.9851	18	506	16	614	0.9737	79	532	17	614	0.971	87	541	17
910	0.8586	59	486	17	432	0.593	39	488	17	570	0.7274	87	501	17	119	0.8824	38	494	17	333	0.9628	23	484	17
570	0.7461	49	464	17	679	0.954	50	446	17	235	0.93	69	466	17	414	0.9658	74	434	17	485	0.9466	92	473	17
105	0.7002	43	409	17	570	0.7464	64	430	17	119	0.8824	33	418	17	677	0.8315	82	414	16	459	0.7381	104	407	17
203	0.8487	17	359	17	414	0.9658	102	362	16	432	0.7527	13	375	17	581	0.9895	71	384	17	918	0.8842	104	371	17
414	0.9627	71	328	17	485	0.8726	91	357	17	918	0.9028	16	324	17	105	0.7002	57	342	17	105	0.7376	58	342	17
918	0.8842	35	294	17	235	0.93	84	313	17	236	0.9185	93	310	17	918	0.9028	89	290	17	677	0.879	38	305	17
614	0.9737	37	267	28	910	0.8586	68	263	27	581	0.9895	107	261	27	235	0.93	55	268	26	236	0.9185	29	262	27
679	0.954	106	220	28	918	0.9028	35	224	28	459	0.7381	106	224	28	679	0.9074	67	227	28	203	0.9019	17	218	28
119	0.7674	10	194	26	459	0.7381	81	187	28	910	0.8586	104	184	28	570	0.6965	56	195	27	487	0.9313	57	195	28
487	0.4926	39	158	27	236	0.9185	45	153	28	679	0.9074	82	145	28	236	0.9185	68	149	28	570	0.6191	71	159	28
677	0.5975	108	116	28	581	0.9859	38	112	28	677	0.8315	42	115	28	910	0.8586	71	108	27	432	0.593	12	104	27
581	0.8697	24	78	28	614	0.9737	93	80	28	203	0.596	87	78	28	203	0.596	24	87	28	119	0.7674	71	77	28
ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD
679	0.954	264	710	17	581	0.9895	17	706	17	487	0.9556	107	725	17	414	0.9851	39	698	17	677	0.907	37	709	17
459	0.7588	18	677	17	910	0.8944	103	673	17	679	0.954	93	670	16	119	0.9708	99	686	17	581	0.9895	53	667	17
414	0.9658	80	625	16	487	0.9556	65	646	17	570	0.7583	78	655	17	918	0.9094	72	617	16	459	0.7588	106	642	16
235	0.93	84	631	17	570	0.7583	106	616	16	485	0.8786	66	630	17	235	0.93	33	626	17	119	0.9592	67	613	17
203	0.9487	84	535	17	105	0.7002	15	570	17	235	0.93	11	583	17	236	0.9185	17	568	16	679	0.954	81	558	16
581	0.9859	50	526	17	203	0.8859	76	516	17	105	0.7002	27	521	16	614	0.9737	39	535	17	235	0.93	21	547	17
485	0.9466	42	512	17	414	0.9627	64	474	17	910	0.8586	42	487	17	570	0.7583	76	500	17	414	0.9627	19	478	17
105	0.7002	78	453	17	459	0.7381	77	453	17	918	0.9094	105	438	16	333	0.9628	18	451	17	432	0.7549	25	454	17
236	0.9185	98	415	17	236	0.9185	24	422	16	581	0.9859	97	400	17	105	0.7002	34	426	16	333	0.9628	102	411	17
487	0.9313	14	388	17	432	0.7549	92	381	17	677	0.8315	19	378	17	487	0.9313	70	389	17	570	0.7177	50	388	17
333	0.9628	32	335	17	235	0.9237	42	347	17	236	0.9185	15	345	16	203	0.8487	105	330	18	485	0.8786	80	356	16
614	0.9737	42	309	16	677	0.879	61	304	16	119	0.8706	38	303	17	677	0.8315	102	297	17	910	0.8586	32	296	17
570	0.6965	82	269	28	918	0.8842	101	260	28	459	0.7381	77	264	27	581	0.9895	66	266	28	487	0.841	89	268	28
677	0.879	103	226	28	333	0.9628	48	219	28	333	0.9628	53	221	26	485	0.4891	65	236	27	918	0.8842	10	224	28
918	0.9028	36	184	27	679	0.926	96	184	27	414	0.9627	57	193	27	432	0.593	55	183	28	614	0.9737	92	197	27
432	0.593	78	144	28	614	0.9301	86	155	27	614	0.9737	77	156	27	459	0.7381	25	149	27	105	0.5793	70	150	28
910	0.8586	46	112	28	119	0.7674	75	116	28	432	0.7549	54	115	28	679	0.9074	59	115	28	236	0.9185	72	114	27
119	0.7674	37	87	27	485	0.3493	16	80	28	203	0.596	45	87	28	910	0.7907	66	74	28	203	0.844	87	77	28

Table B.1. The statistics of results of Experiment 2 and Experiment 3 (Cont. VI)

ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD
105	0.7376	47	717	17	432	0.7549	23	722	17	235	0.93	41	741	17	236	0.9297	23	722	17	614	0.9737	101	727	17
570	0.7583	75	697	17	677	0.907	98	672	17	679	0.954	58	674	17	119	0.9708	73	688	17	333	0.9628	63	680	17
459	0.7588	16	642	16	119	0.9592	102	648	16	414	0.9851	58	622	16	459	0.7588	31	641	16	918	0.9094	62	616	17
918	0.9094	52	583	17	910	0.8944	101	599	17	203	0.9487	74	572	17	918	0.9028	52	585	17	485	0.9466	108	625	16
614	0.9737	29	571	17	203	0.9019	107	544	17	485	0.9466	40	590	17	487	0.9313	44	575	17	679	0.954	97	563	17
235	0.93	18	547	17	614	0.9737	103	532	17	432	0.7549	102	520	17	203	0.9294	92	504	17	119	0.9708	17	532	17
236	0.9185	68	491	17	487	0.8551	45	494	16	236	0.9297	76	494	17	485	0.8786	87	506	16	459	0.7588	22	485	16
487	0.9313	80	455	16	459	0.7588	98	451	17	614	0.9737	26	461	16	910	0.8944	96	439	17	236	0.9185	82	450	17
485	0.8786	46	432	17	570	0.7274	47	429	17	570	0.7177	26	424	17	235	0.9237	9	422	17	235	0.93	34	429	17
414	0.9627	90	367	17	333	0.9628	33	369	17	119	0.8824	44	379	17	679	0.926	77	363	17	105	0.7002	99	374	16
432	0.7527	95	340	17	485	0.8726	107	355	17	333	0.9628	10	333	17	105	0.7002	57	337	17	487	0.9313	83	339	17
119	0.785	22	303	17	105	0.6442	12	297	16	918	0.9028	99	292	16	614	0.9301	96	309	17	432	0.593	33	301	17
203	0.844	77	261	28	581	0.9895	82	262	28	581	0.9859	50	263	28	333	0.9628	69	260	28	570	0.6965	43	275	28
581	0.9859	79	224	28	235	0.9021	80	232	28	459	0.7381	50	228	28	677	0.8315	37	225	28	910	0.8586	16	219	28
679	0.8738	78	184	28	414	0.9627	56	185	28	105	0.5793	108	190	27	581	0.9859	86	189	28	414	0.9627	86	189	27
333	0.9628	102	142	27	679	0.7728	94	156	27	487	0.841	79	155	27	570	0.6965	68	158	27	677	0.879	94	152	28
910	0.6572	65	110	28	236	0.9185	50	116	28	910	0.8586	54	110	28	414	0.7768	24	117	28	203	0.8074	58	117	28
677	0.4782	12	77	28	918	0.8086	35	74	28	677	0.5779	59	77	28	432	0.593	73	95	28	581	0.9859	41	90	28
ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD
581	0.9895	28	706	17	570	0.7583	103	734	17	485	0.9466	47	741	17	910	0.8944	101	712	17	487	0.9556	35	724	16
333	0.9628	94	678	17	203	0.9487	103	654	17	570	0.7464	37	696	16	614	0.9737	67	690	16	105	0.7041	73	678	17
570	0.7583	103	654	17	918	0.9094	89	612	17	459	0.7381	91	637	17	677	0.907	97	636	16	119	0.8824	35	648	17
203	0.9487	13	587	17	459	0.7588	102	605	17	677	0.907	101	597	17	459	0.7588	97	604	17	614	0.9737	89	609	17
910	0.8944	86	560	16	485	0.8786	24	587	17	910	0.8944	68	558	16	487	0.9313	41	564	17	581	0.9895	13	562	17
459	0.7588	12	520	17	581	0.9895	96	522	17	105	0.7002	293	520	17	333	0.9628	38	529	16	910	0.8944	43	520	16
105	0.7002	51	498	16	677	0.8315	83	483	16	487	0.9313	10	492	17	119	0.9592	78	495	17	333	0.9628	81	488	17
677	0.879	56	452	17	119	0.8516	96	457	17	235	0.9237	27	463	17	918	0.9094	96	441	17	432	0.7549	37	453	17
485	0.8726	9	439	17	235	0.93	63	426	17	614	0.9737	63	421	17	570	0.7177	44	424	17	570	0.7464	35	428	17
236	0.9185	45	384	17	236	0.9185	31	382	17	333	0.9628	16	368	17	581	0.9895	26	374	17	235	0.9237	23	390	17
235	0.9237	51	346	16	910	0.8586	13	328	17	918	0.9094	69	327	16	203	0.8487	88	334	16	918	0.9028	104	329	16
414	0.9627	24	291	17	414	0.9627	103	306	17	432	0.593	31	297	17	236	0.9185	99	305	17	677	0.8315	17	300	17
432	0.7549	68	259	27	679	0.926	38	262	28	203	0.8487	96	258	27	235	0.9021	30	269	27	679	0.9074	99	263	27
487	0.4866	104	234	28	487	0.9313	38	234	28	679	0.9074	33	224	28	485	0.8726	90	239	28	236	0.9185	94	228	28
679	0.926	18	184	28	105	0.6442	28	192	27	119	0.8143	13	193	27	105	0.6442	85	185	27	485	0.878	34	198	28
918	0.8842	73	148	27	333	0.9628	104	149	28	236	0.9185	73	153	28	414	0.9627	45	146	28	414	0.9013	34	150	27
119	0.7071	72	115	28	614	0.7752	67	118	28	581	0.851	41	116	28	679	0.7728	32	108	28	203	0.844	12	117	28
614	0.7111	55	95	26	432	0.593	66	80	28	414	0.5835	92	73	28	432	0.4911	95	73	28	459	0.6051	46	103	28
ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD	ID	US	RT-C	RT-Z	DD
119	0.9708	93	726	17	614	0.9737	13	726	17	614	0.9737	25	725	17	105	0.7376	27	715	17	119	0.9708	73	724	17
570	0.7583	93	695	17	918	0.9094	27	656	17	910	0.8944	27	681	16	235	0.93	21	700	17	414	0.9851	80	660	17
414	0.9851	256	621	17	414	0.9851	96	625	17	487	0.9556	18	644	17	679	0.954	66	633	17	105	0.7041	74	645	17
432	0.7549	45	609	17	679	0.954	95	600	17	570	0.7583	82	615	17	119	0.9592	81	609	17	203	0.9019	89	584	17
677	0.879	73	554	17	432	0.7549	25	568	16	119	0.8824	85	572	16	677	0.879	19	564	16	485	0.8786	108	588	16
236	0.9297	40	542	17	581	0.9859	76	517	17	432	0.7549	77	527	17	918	0.9094	75	509	17	614	0.9737	37	533	17
679	0.954	69	478	17	677	0.907	77	484	17	679	0.954	70	484	17	910	0.8586	83	487	17	677	0.8315	67	491	17
485	0.878	97	474	17	119	0.9592	26	456	17	485	0.878	46	472	17	570	0.7464	53	464	17	235	0.93	57	463	17
105	0.6442	25	410	16	235	0.93	78	427	16	677	0.8315	78	414	16	236	0.9185	18	421	16	432	0.7549	45	408	16
910	0.8944	33	367	17	459	0.7381	80	376	17	203	0.8975	69	369	17	459	0.7381	88	379	17	487	0.5237	85	380	17
203	0.844	10	329	17	105	0.7002	99	337	16	459	0.7588	66	334	17	487	0.9313	37	341	17	459	0.7381	10	338	17
918	0.9028	31	292	17	203	0.8487	24	296	17	105	0.6442	97	297	17	203	0.9019	29	288	17	236	0.9185	28	309	17
459	0.7011	87	267	27	570	0.6652	97	276	26	414	0.9627	54	250	26	432	0.5987	20	260	26	581	0.9895	84	258	27
581	0.9042	66	223	27	485	0.8726	94	241	28	236	0.9185	54	229	28	614	0.9737	24	232	28	918	0.9028	18	228	28
487	0.841	45	195	28	487	0.4222	100	192	28	581	0.9078	71	189	28	485	0.8726	42	202	28	570	0.6191	72	198	28
333	0.9628	97	145	28	236	0.8838	14	156	27	235	0.9021	104	153	27	414	0.9627	79	154	27	910	0.6995	44	145	28
614	0.9301	51	113	28	333	0.9628	50	111	28	333	0.9628	67	111	28	581	0.9859	18	117	26	679	0.8017	43	109	27
235	0.8516	88	95	28	910	0.8586	68	88	28	918	0.7748	86	74	27	333	0.9392	38	95	28	333	0.9392	63	74	28

Appendix C Potential Business Models

I&E thesis is written in the course of "1ZS30 - Innovation and Entrepreneurship Thesis" (6 ECTS), which is a mandatory course for EIT Digital students to apply Innovation and Entrepreneurship knowledge into practical. In this course, students have to propose three potential business models for the projects they are working in master project, as well as do a dominant step in the Innovative Development Framework. For my I&E thesis, except conducting a survey, three potential business models are proposed, and they are shown in Figure B.1 – Figure B.6, using Osterwalder's Business Model Canvas as well as Board of Innovation Framework.

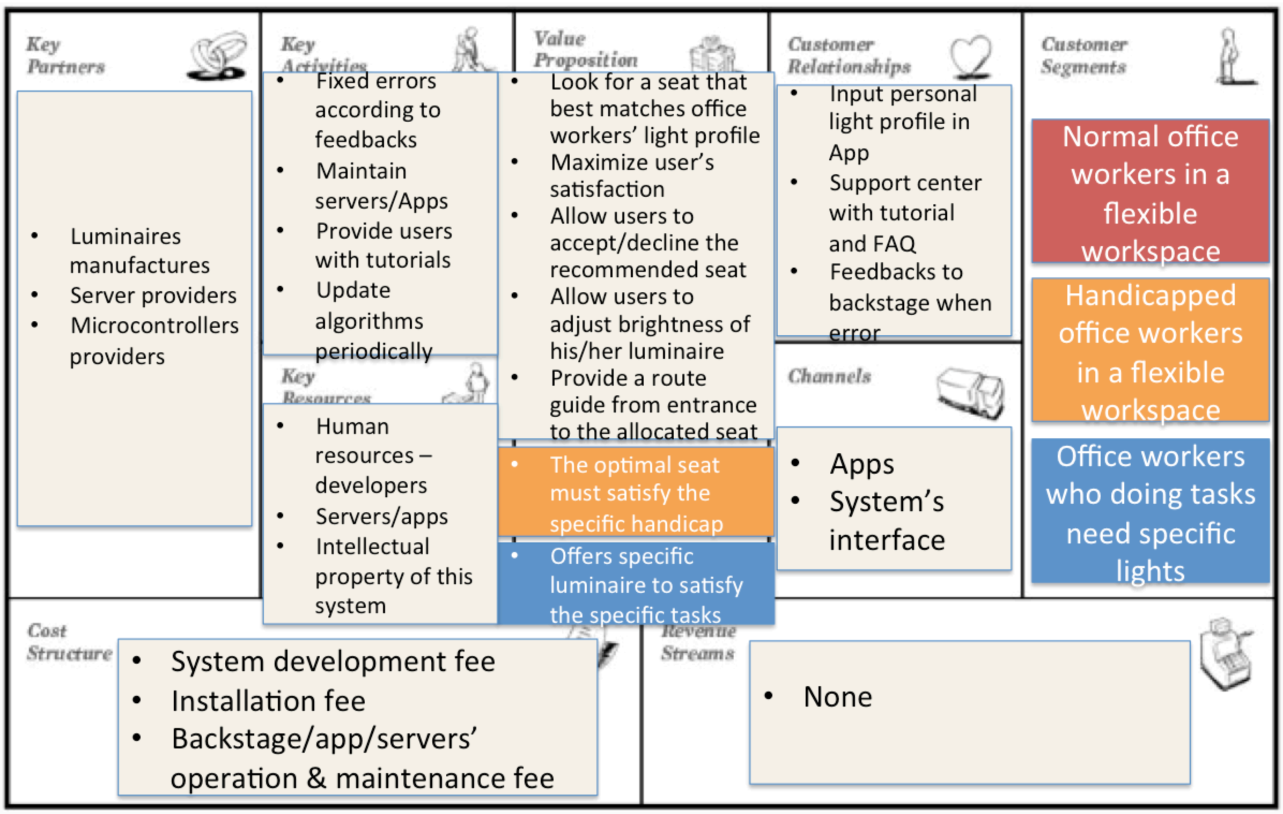


Figure C.1 Business model I (Osterwalder's canvas)

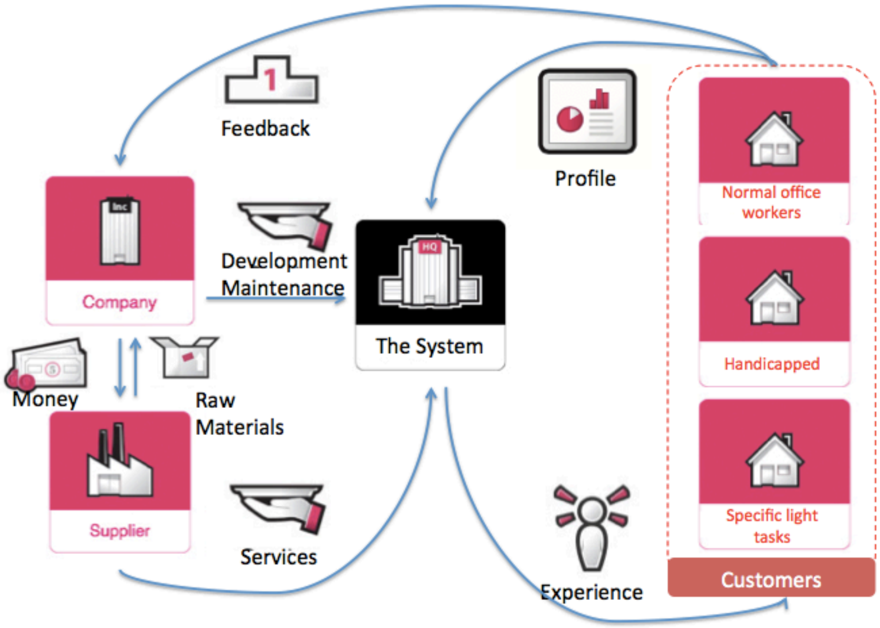


Figure C.2 Business model I (Board of Innovation Framework)

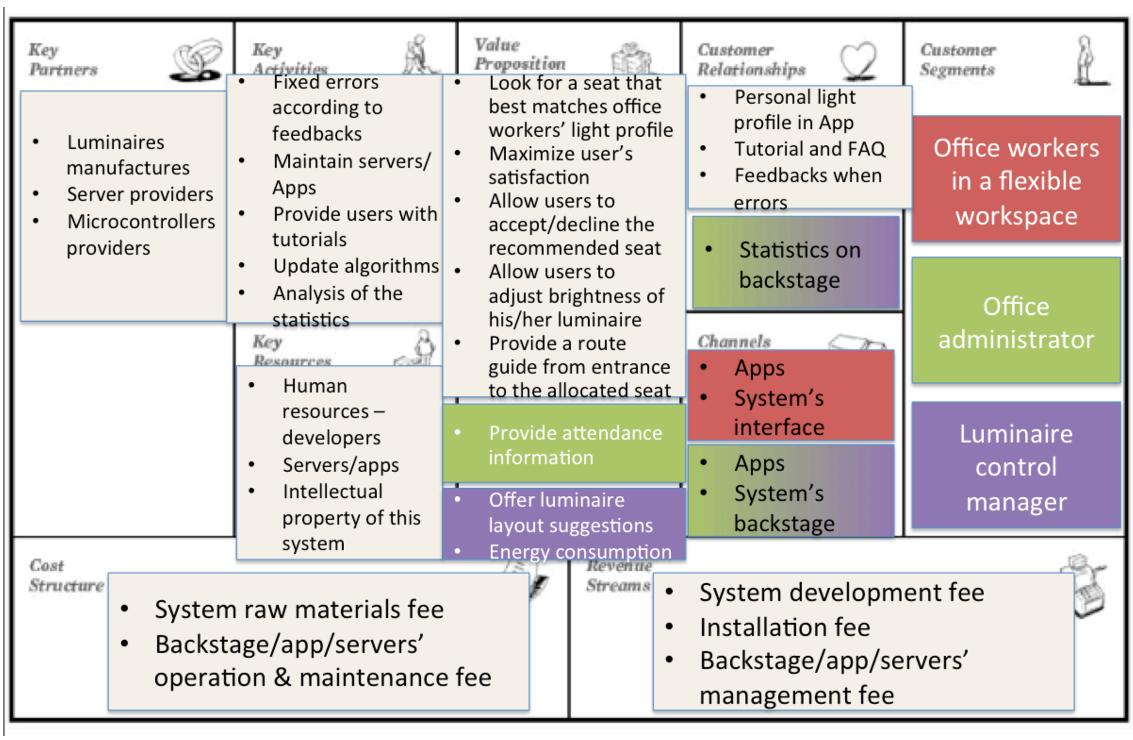


Figure C.3 Business model II (Osterwalder's canvas)

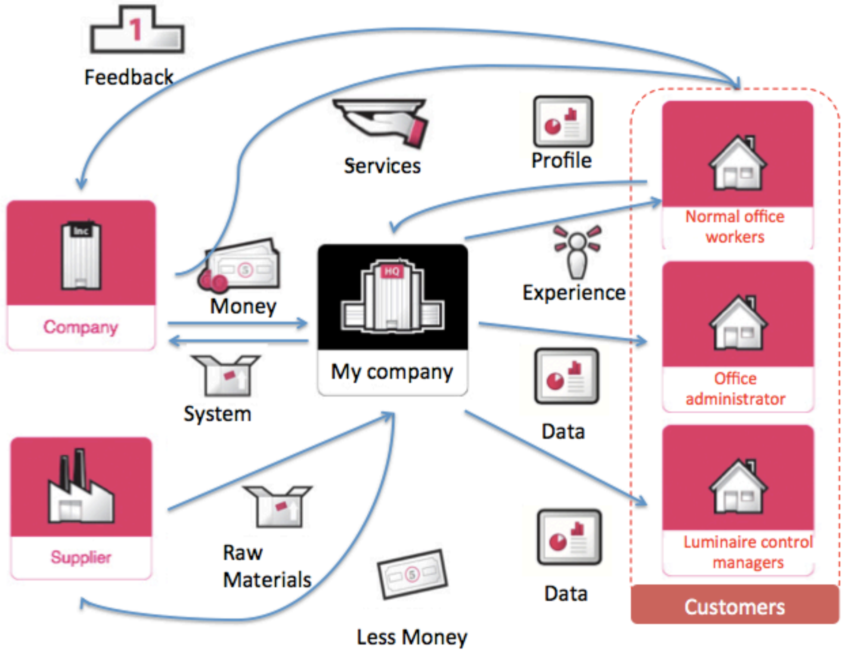


Figure C.4 Business model II (Board of Innovation Framework)

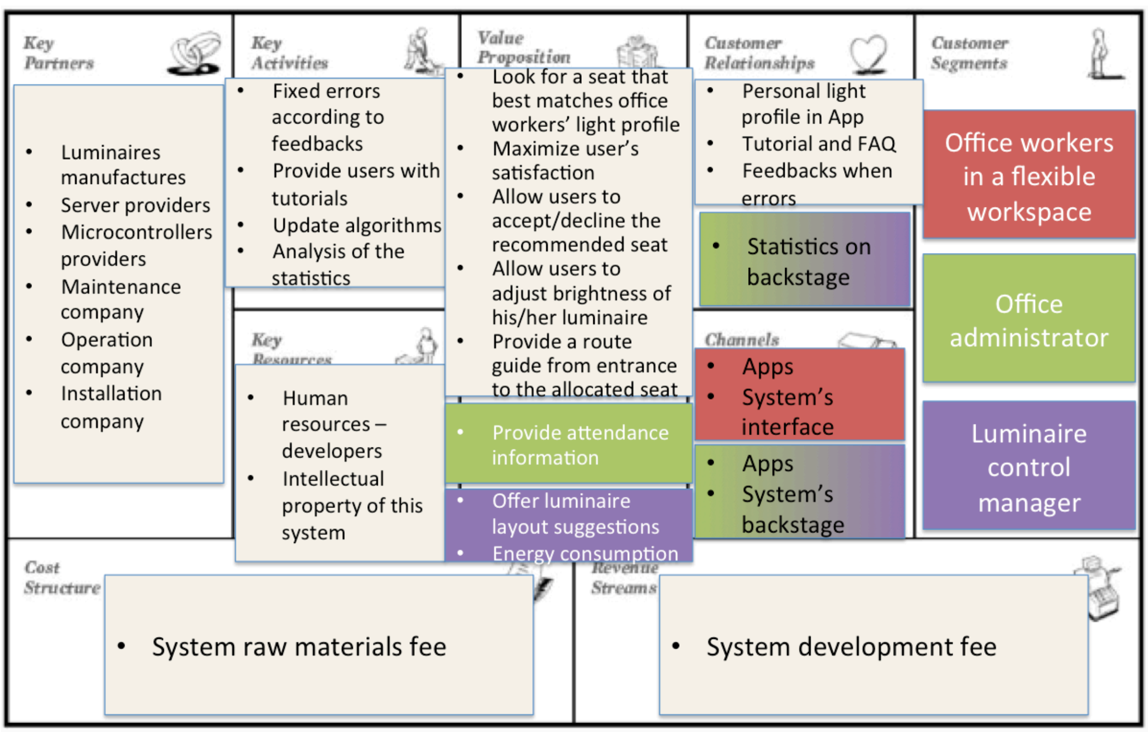


Figure C.5 Business model III (Osterwalder's canvas)

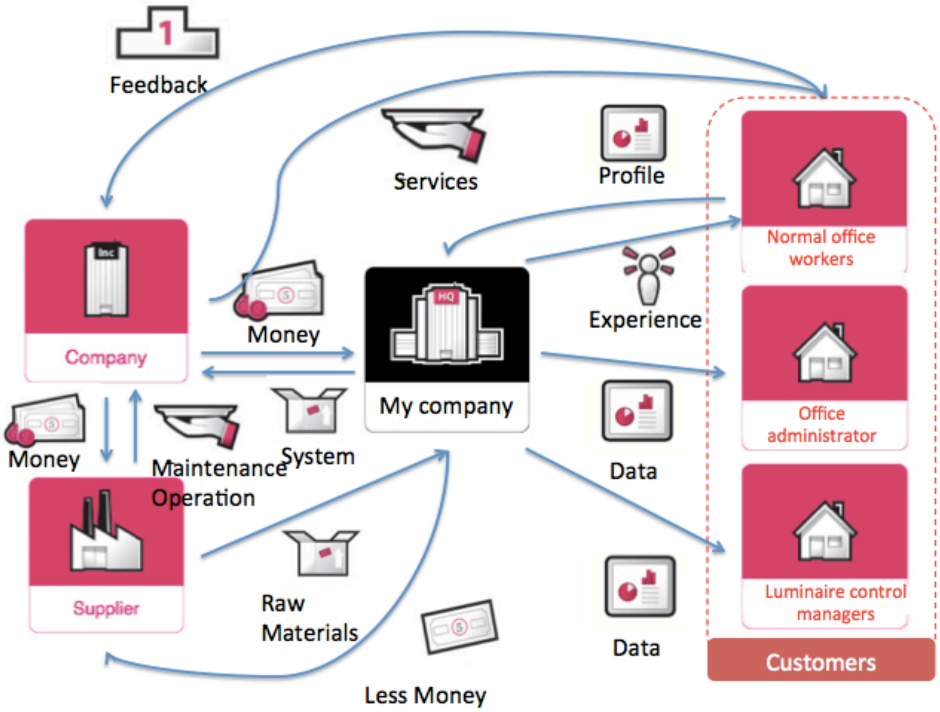


Figure C.6 Business model III (Board of Innovation Framework)