1-1-2021

Cross-Layer Optimization on Wireless Visual Sensor Networks for 3D Indoor Monitoring: A Top-Down Approach

Zhonghui Wang
University of Mississippi

CROSS-LAYER OPTIMIZATION ON WIRELESS VISUAL SENSOR NETWORKS FOR 3D INDOOR MONITORING: A TOP-DOWN APPROACH

A Dissertation
presented in partial fulfillment of requirements
for the degree of Doctor of Philosophy
in the Computer and Information Science
The University of Mississippi

by

Zhonghui Wang

December 2021

ABSTRACT

As one of the latest advances in wireless sensor network (WSN), wireless visual sensor network (WVSN) bears a number of distinct features compared to its predecessor, mainly due to its directional visual coverage on the sensing field. This also renders previous solutions proposed for WSN may not work well for WVSN. For example, k-coverage has been introduced to WSNs for fault tolerance and high monitoring quality by omni-directional sensors but becomes incapable to cope with the directional coverage of sensors in WVSNs. Moreover, compared to scalar data in WSNs, visual data are much larger, difficult to aggregate and usually streamed in real-time, which also brings unprecedented challenges to data routing, load balance and traffic scheduling in the network. In this dissertation, we strive to tackle these issues with a top-down across layer approach. At the Application layer, we take an initial step to tackle 2-Angular-Coverage problem for WVSNs. To maximize the information captured by 2 visual sensors, we deploy the visual sensors from different directional angles and further extend the conventional concept of 2-coverage to "2-angular-coverage". With this as building block, we develop a greedy heuristic and an enhanced Depth First Search (DFS) algorithm to address 2-Angular-Coverage problem. At the Network layer, we note that the widely used tree topology in WSNs may yield sub-optimal results, and propose to optimally balance traffic load by using the full network topology, transforming it into a generalized max-flow problem and designing an efficient solution. Finally at the Link layer, we propose a Time Division Multiple Access (TDMA) based scheduling algorithm which is promising to achieve contention-free visual streaming data collection. The optimization information from upper layers will also be used for cross-layer optimization to further improve the quality of visual streaming and shorten the transmission delay.

DEDICATION

To my wife and my parents, Xi Jin, Jun Wang and Xiurong Deng

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Feng Wang, and committee members Dr. Dawn Wilkins, Dr. Yixin Chen and Dr. Shan Jiang, for guidance and patience throughout my graduate study at The University of Mississippi.

I would like to thank my family for their encouragement and supporting me to pursue a degree in Doctorate of Science.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Wireless sensor networks (WSNs) have attracted significant interest from both academia and industry on applications in security, medical, military, environment, industrial and transportation fields as mentioned by Akyildiz et al. (2002); Chen. A typical architecture of a WSN consists of distributed sensors that capture, process, transmit and receive data to collaboratively serve users via the base station, which usually works as a gateway and helps coordinate the network. As one of the most recent advances of WSNs, wireless visual sensor networks (WVSNs) further expand the capacities of such networks with cameras mounted on sensor nodes that allow the recording and processing of images and/or videos. This unique feature makes WVSNs different from traditional WSNs in several ways and renders the solutions proposed for traditional WSNs inapplicable to solve the problems in WVSNs. For example, Figure 1.1 shows the newly emerged shopping applications such as Amazon go and Alibaba Tao Cafe [1], where customers may purchase products without being checked out by a cashier or using a self-checkout station. All the operations such as product counting, user account matching, and payment processing are achieved through visual sensors with deep learning algorithms and data fusion technologies. The functionalities of visual sensors are to monitor which product has been grabbed in/out of the shelf by which customer, examine whether the monitored customer has the corresponding face record in the database, detect if the customer is with higher probability to be a thief due to the weird actions or walking paths, etc. In order to achieve these goals, the whole shopping area should be fully monitored by visual sensors, and the target tracking function should also be implemented. No doubt such complex tasks can hardly be achieved by traditional WSNs. In general, a wireless

[1] https://syncedreview.com/2018/01/22/amazon-go-vs-alibaba-tao-cafe-staffless-shop-showdown/

Figure 1.1. Amazon Go as the newly emerged Grab & Go Store[2]

visual sensor node consists of five hardware components which are image capturing device, processing unit, memory, radio module and power supply. Charge-coupled device webcams (CCD) and complementary metal-oxide semiconductor imaging devices (CMOS) are two common image capturing devices used by visual sensor nodes. CMOS imaging devices are smaller, cheaper and consuming lower power than CCD webcams, but their image resolutions are relatively low compared with CCD webcams. The processing unit includes two types which are non-programmable logic unit and programmable logic unit. Non-programmable logic unit has better performance and lower power consumption. Therefore, it is preferable for high volume sensor node deployment. However, the non-programmable processing unit cannot be altered to meet changing functionality requirements. On the other hand, the design of programmable logic unit allows it to be altered for other functionalities. This

[2]This original picture is from: https://www.cnet.com/news/amazon-might-open-thousands-of-its-grab-and-go-stores-over-the-next-three-years/

flexibility makes programmable logic unit dominate the processing units in WVSNs. Due to the larger size of captured visual data (image/video), the memory and storage requirements of WVSNs are much higher than those for WSNs. The radio modules of WVSNs can be classified into two categories as low rate wireless personal area networks (WPANS) and high rate wireless local area networks (WLANs), respectively. IEEE 802.15.4 and Bluetooth are the typical WPAN radio modules in WVSNs. These kind of radio modules can only provide shorter transmission range (10 to 20 m) and lower transmission rate (250 Kbps and 1 Mbps), but they have lower power consumption. On the other hand, IEEE 802.11b is a popular example of WLAN radio module in WVSNs which can support longer transmission range (around 100 m) and higher transmission rate (11 Mbps), but this longer transmission range and higher data rate comes with higher cost in price and power consumption. Such hardware configurations further lead to several major differences of WVSNs from traditional WSNs as discussed in Yap and Yen (2014).

First, unlike traditional sensors such as temperature or light sensors that are often considered with omni-directional sensing range, the sensing range of a visual sensor is usually deemed as fan-shape in 2D and pyramid-shape in 3D. Moreover, most cameras on visual sensors provide pan, tilt and zoom functionality with more monitoring flexibility. This, however, makes visual sensors relatively more expensive than traditional sensors, and the tasks of deploying them become even more challenging as not only the location but also other factors such as the working direction, field of view and aspect ratio can affect the area of coverage and thus the monitoring quality, not to mention that to improve cost-effectiveness, it is usually preferred to only use a minimal number of visual sensors in the considered monitoring space.

In the literature, enormous efforts have been devoted to address various issues at different layers of WSNs. However, most of them may not well fit in the WVSNs and 3D indoor monitoring scenario. At the application layer, extensive efforts have been devoted to achieve k-coverage for WSNs deployed in the sensing area by considering omni-directional

sensing range as seen in scalar sensors, where the larger k is, the better fault tolerance and monitoring quality will be. However, in WVSNs, due to the limitation of FoV of the visual sensor, the k-coverage solutions proposed for WSNs become incapable of solving the problem in WVSNs. Moreover, the purpose of using k-coverage in WVSNs may be not only for fault tolerance, but also to achieve angular coverage for the object as mentioned by Chow et al. (2007); Yildiz et al. (2014)

From the perspective of network layer, we note that the tree network topology, which is widely used in WSNs, may yield sub-optimal results in term of alleviating traffic routing bottlenecks, since it eliminates the opportunities that some sensors in the network may connect to multiple neighbors closer to the base station. Therefore, causing the existing solutions for WSNs are insufficient to solve the problem in WVSNs.

Medium access control (MAC) is one of the critical issues in the design of WSN link layer. As in most wireless networks, collision, which is caused by two nodes sending data at the same time over the same transmission medium, is a great concern in WSNs. To address this problem, a sensor network has to employ a MAC protocol to arbitrate access to the shared medium in order to avoid data collision from different nodes and meanwhile to fairly and efficiently share the bandwidth resources among multiple sensor nodes. According to different sensor network application, the MAC protocols can be divided into two primary categories which are contention-based protocols and contention-free protocols. Contention-based MAC protocol like S-MAC mentioned by Yadav et al. (2009), which grows from IEEE 802.11, enhances the energy efficiency, while it may result in a non-deterministic per-hop delay. On the other hand, TDMA belongs to contention-free category which does not have a problem of retransmission caused by collision also called collision-free MAC protocol. In TDMA based MAC protocols, time are divided into time-frames and each time-frame is further divided into a fixed number of time-slots. Each node is allocated a time-slot in a time-frame and is allowed to transmit only in the allocated time-slot. However, since the visual data is much larger compared with scalar data in terms of size, and even worse for

the sensor nodes closer to the base station required to relay the accumulated data for other sensor nodes, single time-slot cannot handle this accumulated streaming flow for each heavy duty sensor node.

As the differences discussed above, the existing solutions used in WSNs are generally incapable of solving the problems in WVSNs.

For 2-coverage in WVNSs, to maximize the information that can be captured by 2 visual sensors covering a 3D location, we propose to deploy the visual sensors from different directional angles and further extend 2-coverage to "2-angular-coverage" to denote such unique requirement in WVSNs. We demonstrate our simulated WVSN testing platform which emulates a 3D space and highlights our objective to provide efficient 2-angular-coverage in the indoor environment by minimizing the number of visual sensors required to cover the full regions through precision angle view monitoring. We first formulate the general problem in a continuous space and then discretize our model via a lattice based approach, which can achieve arbitrary approximation precision by adjusting the grid granularity. In addition, we also create a strategy to determine the degree of coverage for the WVSN, where a given location needs to be ensured to be covered by at least 2 visual sensors that fulfill the angular coverage requirement.

In order to increase the reliability of WVSN, the network can be powered by batteries (e.g., during an emergency deployment or a power outage). However, the attribute of visual sensor can quickly drain the battery power and significantly damage the network lifetime. Therefore, the deployment of WVSNs must not only guarantee the 3D indoor area coverage and connectivity between visual sensors and the base station, but also ensure that the traffic loads are well balanced and collision-free multiple access among network in order to prolong the lifetime of the entire network. At the network layer, beyond achieving the 2-angular-coverage goal, we also take consideration of traffic load balance. We propose to optimize the traffic load balance by working on the full network topology, transforming it into a generalized max-flow problem and proposing an efficient solution.

Finally, we take a cross-layer approach, providing a solution across visual sensor coverage, streaming data delivery, network traffic load balance and wireless link scheduling. We first model the wireless visual sensor deployment and scheduling as a cross layer optimization problem. We then use a divide and conquer strategy to propose two sub-algorithms, which, for a given network topology, can achieve optimal traffic load balance and link scheduling, respectively. Moreover, we develop SEED, a SchEduling-aware Enhanced Depth-first search algorithm that can yield optimal solution if given enough time. Evaluation results show that our solution can greatly reduce the overall communication latency and achieve excellent load balance while minimizing the total number of sensors.

CHAPTER 2

2-ANGULAR-COVERAGE DEPLOYMENT AT APPLICATION LAYER

2.1 BACKGROUND

Extensive efforts have been devoted to achieve k-coverage for WSNs deployed in the sensing area by considering omni-directional sensing ranges as seen in scalar temperature sensors, where the larger k is, fault tolerance and monitoring quality is better. However, in WVSNs, the sensing range of a visual sensor is often directional, which renders the k-coverage solutions proposed for WSNs incapable of solving the problem in WVSNs. For instance, when an intruder gets into the monitored area, if only the back of the intruder has been captured by the visual sensor, it may not be useful to recognize the identity of the intruder. However, the face of the intruder can be captured if k-angular-coverage strategy has been used.

2.2 RELATED WORK

The k-coverage problem generally states that any point within the monitored area must be located in the sensing range of at least k-sensors mentioned in Malek et al. (2016). As an example, if the deployment area is defined as 4-coverage, every point within the area is covered by at least four sensors. This means that up to three of the four nodes sensing the same monitored region can fail and the area will still be covered (monitored) by one of the four nodes. Full k-coverage denotes that every point is covered by all k-sensors. However, it is often suggested that guarantee 100% coverage in WVSNs is very difficult in a random deployment scenario. Therefore, Liu et al. (2008) proposed a directional k-coverage metric (DKC). DKC demonstrates the coverage quality measured in terms of a probability guarantee. It utilizes a summation function to calculate the probability of coverage for a target by

a camera considering the visual sensor's field of view, deployed nodes and available sensors covering the same region. We have implemented a contrasting narrative in our work, where we consider the angular direction for deployed visual sensors within the coverage area. Several studies have explored the issue of the coverage problem with some specifically addressing k-coverage. We will discuss this matter further and explore the distinction in our current work. Katsuma et al. (2009) proposed a scheduling method which addressed the k-coverage problem to extend the network lifetime of WSNs. In their approach a simple 2D model with an omni-directional sensing field was employed. In contrast, our model applies a pan-tilt directional D perspective, where the visual sensor has vertical and horizontal freedom. Similarly, Chow et al. (2007) directly studies the angle coverage problem in visual sensor networks. In their work, the authors modeled the Minimum Cover problem and developed a distributed algorithm to determine the minimum set of sensors using an omni-directional sensing field. Hefeeda and Bagheri (2007) address the k-coverage problem in dense sensor networks. The work formulated the k-coverage problem as an optimal hitting set problem. The proposed distributed algorithm in their work ensured k-coverage of the monitored area without requiring the location of the sensor nodes deployment. All papers highlighted above implement simple 2D models with 360°sensing field within a WSN, resulting in impractical real world application settings and thus incurring inconsistencies. Another work implements the k-coverage problem as discussed by Ammari (2011), which addresses the problem in a three dimensional aspect but the sensing field range is modeled as an omni-directional sphere and the authors focus on solutions for wireless sensor networks which is inapplicable to our problem. As highlighted previously, introducing WVSNs into an environment presents additional challenges that are not often attributed to WSNs such as coverage quality that depends on the orientation of the visual sensor. In our work, we use the minimum number of visual sensors to achieve k-coverage from different directional perspectives to monitor 3D areas of interest.

2.3 PROBLEM FORMULATION

We consider a 3D indoor space, where some 3D areas are required to be fully covered by visual sensors. The visual sensors can only be deployed at certain location (e.g., on the ceiling and/or the upper half of the wall), which introduces additional challenges. After a visual sensor is deployed, its facing direction will be adjusted to a certain direction and then stay there during the whole monitoring period.

2.3.1 Network Continuous Model

Consider a 3D indoor space, let $\boldsymbol{A}$ denote the 3D areas that must be fully covered (i.e., continuously monitored) by the visual sensors denoted as $\boldsymbol{S}$ and $\boldsymbol{L}$ denotes the mountable positions within the monitored area that can be used to deploy the visual sensors. The location and direction of the visual sensor is defined as a tuple (L, D). We assume for a 3D indoor space, the location L is represented as a 3D coordinate for the visual sensor placement (x, y, z) and the direction D=(x', y', z') is a point on the surface of a unit sphere (which we call a facing direction sphere) with its radius equal to 1 and centered at $(0, 0, 0)$. We denote $face$ to represent the facing direction vector from $(0, 0, 0)$ to (x', y', z'). As mentioned earlier, the interest point specifically within the 3D area in $\boldsymbol{A}$ that we want to monitor is denoted as a. Additionally, we assume R_S is the maximum sensing range of the visual sensor. $\boldsymbol{cover}(L, D, R_S)$ denotes the area that a visual sensor can cover, a function of the location, direction and maximum sensing range of the visual sensor. For this work, we model the visual sensors as a frustum with parameters shown in Figure 2.1: $far\ field, near\ field, field\ of\ view$ (FOV) and *aspect ratio*. We formulate our problem as to find a set of locations and directions of visual sensor nodes where, S=$(L_1, D_1), (L_2, D_2), ..., (L_n, D_n)$, subject to the following constraints:

(1) Sensor Location Constraint:

$$\forall (L, D) \in \boldsymbol{S}, L \in \boldsymbol{L};$$

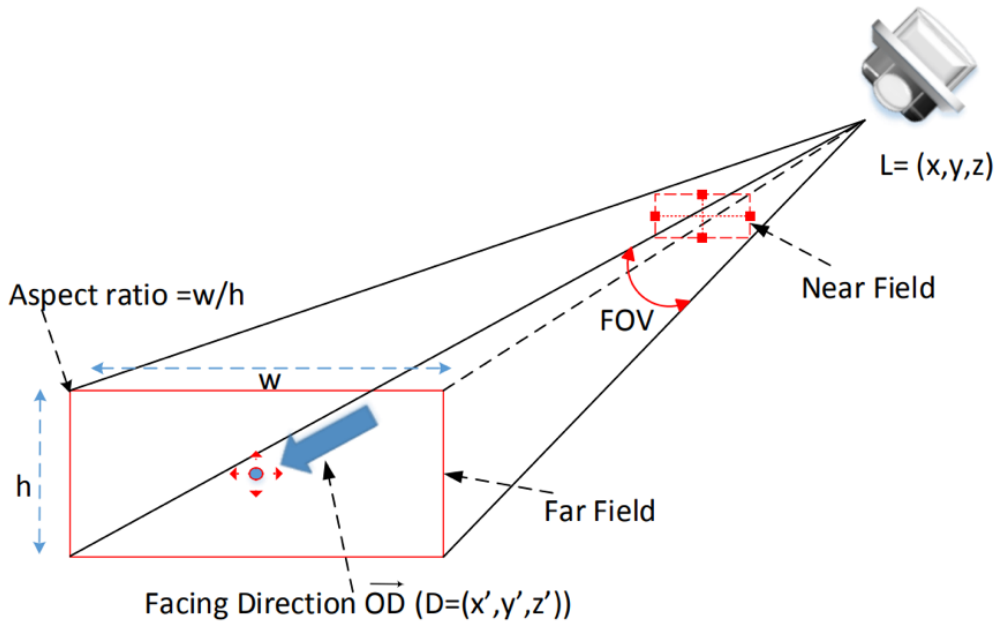

Figure 2.1. Sensor Parameters

(2) Region Angular 2-Coverage Constraint:

$$\forall a \in \boldsymbol{A};$$

$$\exists (L_i, D_i), (L_j, D_j) \in \boldsymbol{S}(i \neq j); \text{ such that,}$$

$$a \in \boldsymbol{cover}(L_i, D_i, R_S) \cap \boldsymbol{cover}(L_j, D_j, R_S) \text{ and}$$

$$120^\circ \leq \measuredangle(\overrightarrow{\mathrm{aL_i}}, \overrightarrow{\mathrm{aL_j}}) \leq 180^\circ$$

The meaning of the second constraint is that the monitoring point a is in the covered frustum of visual sensors deployed at both L_i and L_j to allow 2-angular coverage. This constraint will allow for fault tolerance within the network, where if one sensor fails another sensor will be available to continue to monitor the space. Moreover, when two visual sensors work simultaneously, the 2-angular-coverage can greatly enrich the captured visual information if an object presents at a. Our objective is thus to minimize $|S| = n$, where n is the number of visual sensors.

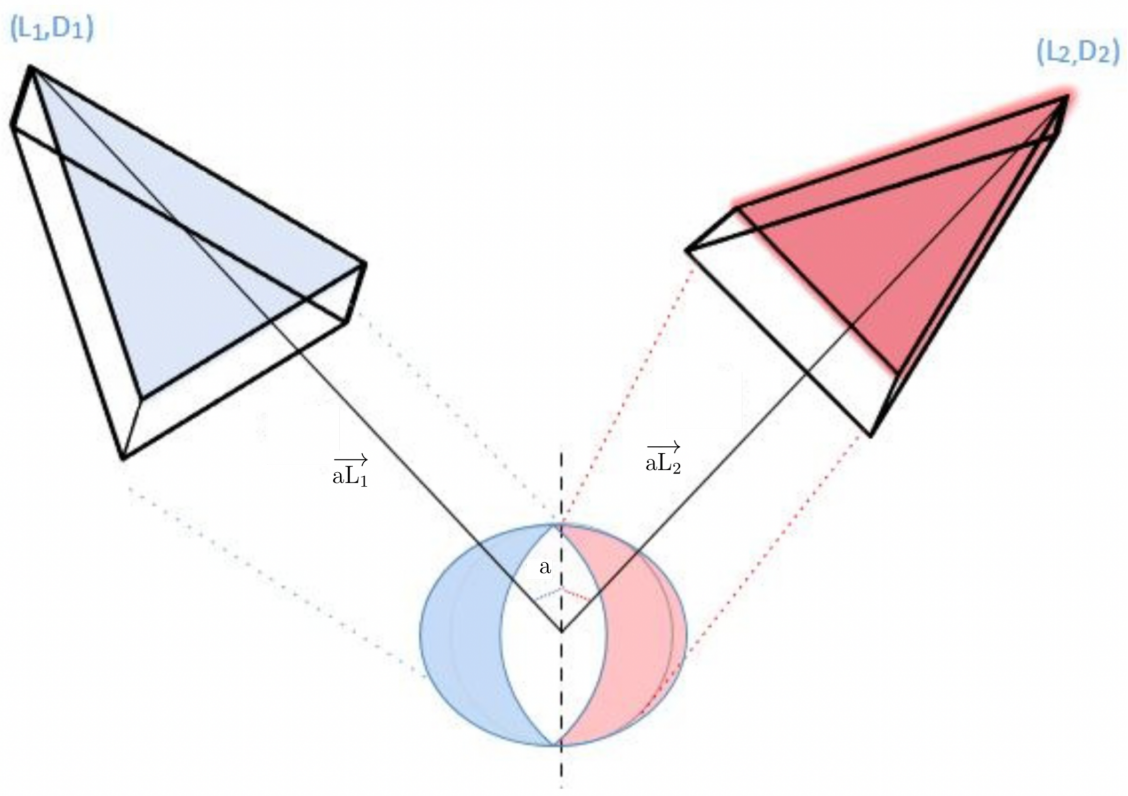

Figure 2.2. 2-Angular-Coverage

As formulated in the angular 2-coverage constraint, there must exist at least 2 visual sensors that can monitor a given point to provide varied angular coverage. This feature will allow for potential applications in tracking of a target (i.e., facial tracking). Consider this scenario where the requirement is to provide some type of facial tracking in a given monitored space. The monitored space can be defined into regions within a visual sensor network, where the camera placement should provide different angles of the tracked target. Though overlapped visible objects may occur with the WVSN target tracking, the image can provide completely different scenes considering the angle. Based on this assumption we propose to consider if the visual sensor captures the point from different viewing directions for which the angle between their direction is at least greater than or equal to 120° difference in the coverage angle. A representation of 2-angular-coverage with the visual sensor direction constraint for a point in the monitored area is shown in Figure 2.2: vectors $\overrightarrow{aL_1}$ and $\overrightarrow{aL_2}$.

2.3.2 Network lattice Model

A network lattice model is used to approximate the continuous space model by dividing the monitored space $\boldsymbol{A}$ that must be fully covered into discrete 3D grids. We required that all grid points in the region are at least 2-angular covered. This technique is also used for the mountable positions in $\boldsymbol{L}$ to divide the candidate locations that a visual sensor can be deployed in the 3D monitored space. The tuple (L_C, D) is used to denote the candidate location and direction of a visual sensor. The distance between two neighboring grid points g_A is used to denote the granularity of the grids used in $\boldsymbol{A}$ and g_L denotes the granularity of the grids used in L. In addition, we also divide the surface of the facing direction sphere into grids (like the longitudes and latitudes divide the surface of the earth) and use g_D to denote the granularity. We assume that a wireless visual sensor can only face to a direction where its D falls on a grid point. Figure 2.3: illustrates the facing direction circle. It is easy to see that by adjusting the three granularity parameters, we can easily achieve the required accuracy for approximating the continuous space model.

2.4 SOLUTION

In this section, we tackle the 2-angular-coverage problem for 3D indoor monitoring using a greedy heuristic algorithm. We then propose an enhanced-DFS algorithm with pruning strategies, which if given enough time, can return an optimal solution.

2.4.1 Greedy Heuristic Algorithm

The greedy algorithm works in an iterative manner to deploy visual sensors. In each iteration, the algorithm strives to choose among the available locations in L and find the location where the deployed visual sensor can cover a number of fresh points as seen in Figure 2.4, we update whether the points have 1-coverage, 2-angular-coverage or no coverage. This is implemented by recalculating how many points can be covered by each remaining location in L. Additionally, we then consider those points that are only covered by one visual sensor

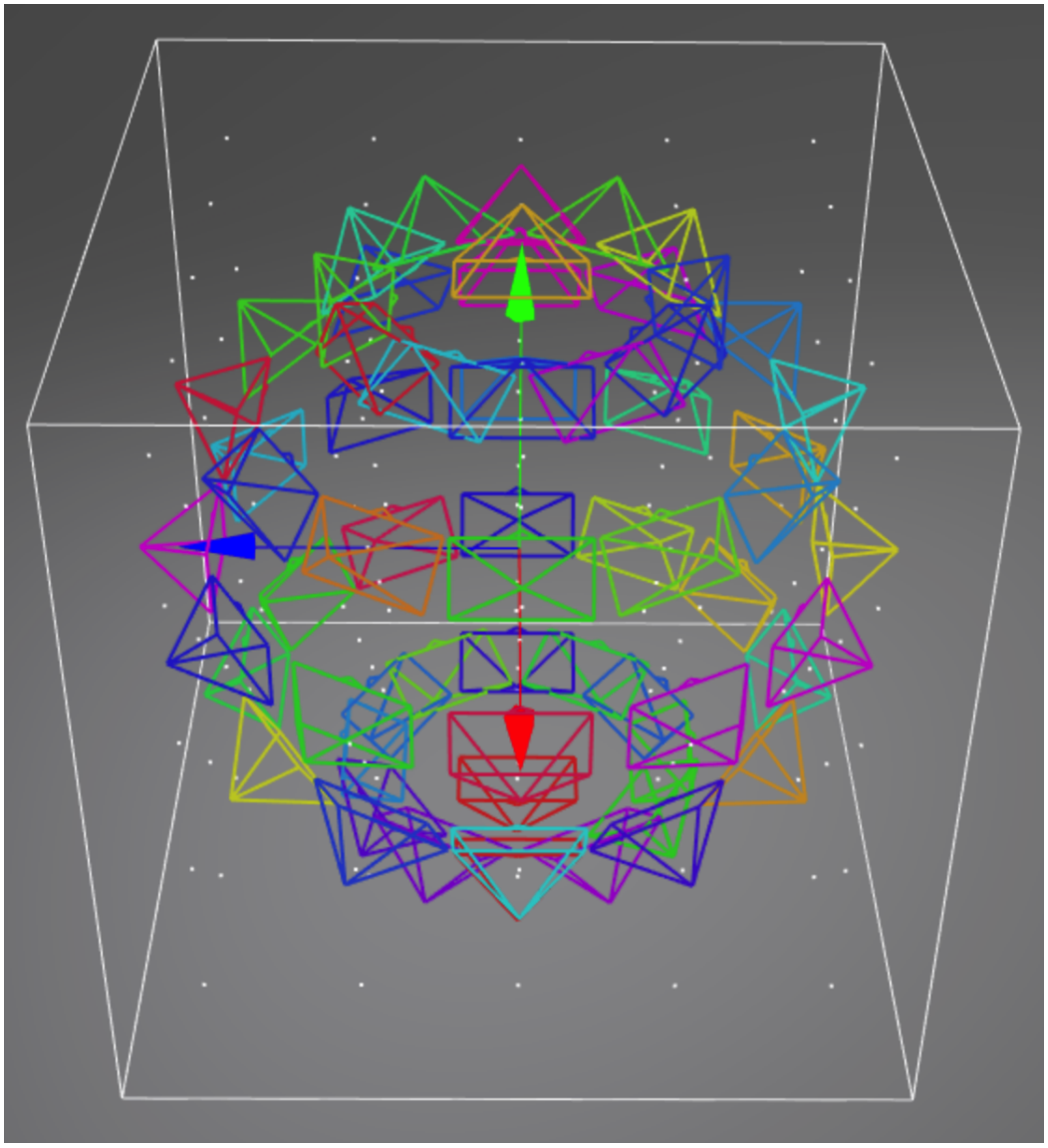

Figure 2.3. Facing Direction Sphere

and check if by placing a visual sensor at a location, which deployment achieves the maximum conversion of points from 1-coverage to 2-coverage by an opposing visual sensor and via an angle different from the previous coverage (as required by the 2-angular-coverage). The one that can maximally cover the most points at 2-angular-coverage after recalculation will then be chosen as the next location to deploy a visual sensor.

Algorithm 1 Greedy Heuristic Algorithm

Input: **A**, D, **L**
Output: Set of S
Initialize: List of **A**, D, **L** and $final$.
while **L** $\neq \emptyset$ and **A** $\neq \emptyset$ **do**
  for all C_L in **L** **do**
    if C_L within R_{max} of any deployed visual sensor **then**
      for all $face \in D$ for L **do**
        select the $face$ with maximum 2-coverage particles
      end for
      record $C_{L\text{-}max}$ into *templist*
    end if
  end for
  choose top one from *templist*
  remove $C_{L\text{-}max}$ from **L**
  update states of particles covered by $C_{L\text{-}max}$
end while
return $final$

2.4.2 Enhanced-DFS Algorithm with Pruning

In Algorithm 1, we provide out enhanced-DFS algorithm to further improve the quality of our solution. In a traditional depth first search approach, the exploration of each branch within our solution set will need to be searched to pick a location in **L** and a facing direction D. However, this is not ideal since the solution space can quickly expand with the size of **L**. In this solution, a pruning technique is implemented which can exploit the search by cutting off the removing most of the solution space that does not include high-quality solutions. We are able to efficiently reduce the size of the candidate location L_C options by starting the search from the result of our greedy heuristic algorithm as the currently found

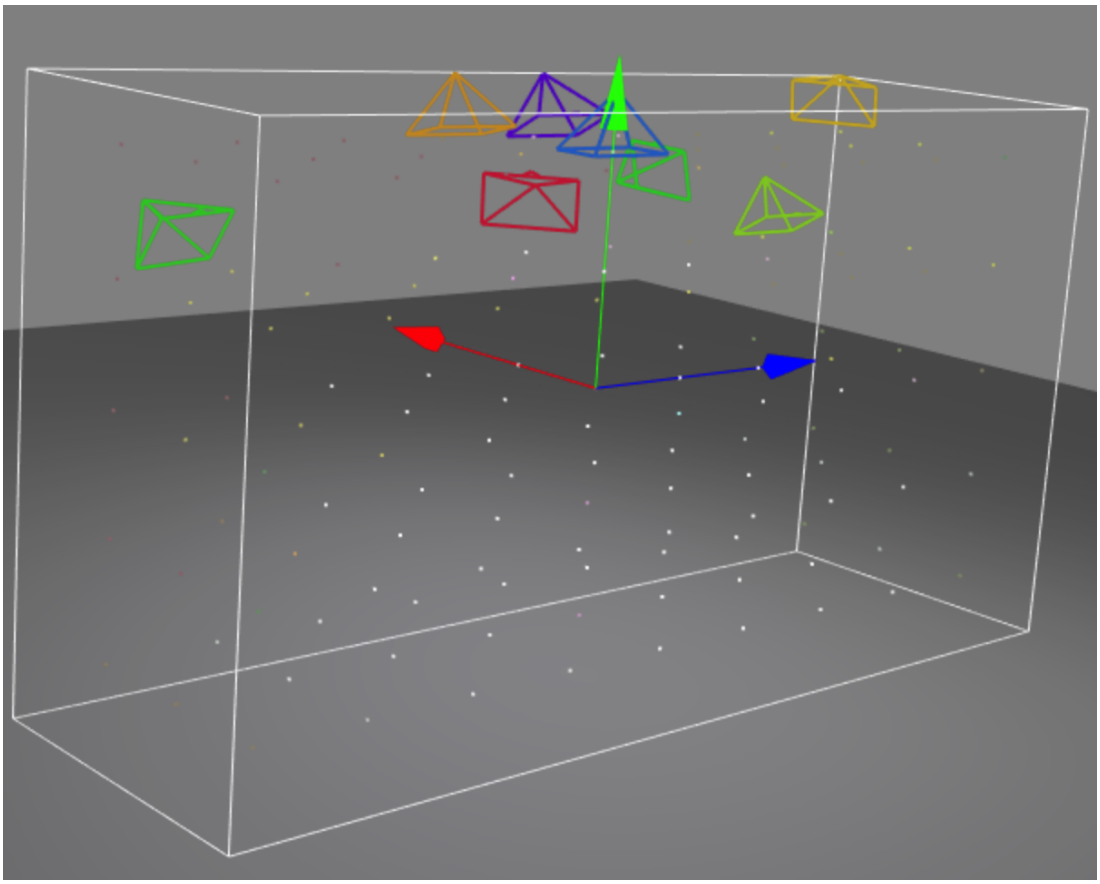

Figure 2.4. Deployment Perspective

best solution. The search branches (*depth*) that already have used equal or more number of visual sensors compared to the currently found best solution can be pruned (lines 1-4). Additionally, the selection of the location in each iteration is chosen by using the default order (e.g., sort all L_C by decreasing order based on the maximum number of points that a visual sensor at a deployable location can convert a point from 1-coverage to 2-angular-coverage or from no-coverage to 1-coverage (lines 8-14)). This approach allows our algorithm to quickly find high quality solutions and skip as many low quality solutions as possible. When a search branch is pruned or explored fully, the search will update all the parameters and will continue the search using its previous status (lines 18-22). The search will halt after all points in the 3D indoor space is 2-angular-covered. An optimal set of visual sensor deployment location is then returned (line 26).

Theorem 1. *The enhanced DFS algorithm with pruning can return the optimal solution to solve the discrete version problem given enough time.*

Proof. The enhanced DFS algorithm incorporates three enhancements which improve the performance of the algorithm. The first enhanced approach that we apply is to prune some of the branches as the graph is traversed based on a greedy heuristic. Traditionally, a DFS algorithm searches a structure by selecting a root node and explores each branch then backtracks. A generalization of the traditional DFS algorithm is considered to be a brute force approach where you search each branch until an optimal solution is returned. We can deem our solution to be a brute force approach with some of the infeasible solution space reduced. Thus, we will prove that our pruning feature in the enhanced DFS algorithm will not eliminate optimal solutions. We select and expand on search branches by determining the maximum number of fresh points covered by the chosen location and direction. Given that F is non-negative where $F \geq 0$ and if current $depth \geq best$, we do not need to check it further. As a result, we can cut the branch. Specifically, since the number of F_R is recalculated after each iteration of the search, the cost function is non-decreasing as the search step traverses the graph for a feasible solution.

□

2.5 PERFORMANCE EVALUATION

The efficiency and validity of our algorithms are evaluated by extensive simulations. We implement a simulation environment built using both Java and JavaScript, where the core deployment algorithms are implemented by Java to conceptualize and emulate the actual deployment and 2-coverage scenarios. In our evaluation, we implement both the greedy heuristic and enhanced-DFS algorithms. Figure 2.5 shows how the number of visual sensors change as the length of the 3D indoor space increase (i.e., default setting for indoor space equals length X 60 X 100), which highlights the flexibility of the system. The enhanced-DFS algorithm outperforms the greedy algorithm as seen in the figure, especially when the length of the indoor space is less than 200. As the monitoring space increases the number of visual sensors are similar. Additional testing was conducted to evaluate both algorithms. The greedy algorithm results were analyzed and the number of required visual sensors was reduced by 37% over the baseline solution and our enhanced-DFS algorithm can further reduce the number of visual sensors by up to 22% over our greedy heuristic algorithm.

Algorithm 2 Enhanced-DFS Algorithm

Input: **L**=∅, **A**=∅, D=∅
Output: Minimized set of $|S| = n$, ensuring 2-angular-coverage
Initialize: List of **A**, D, **L** and *list*
DFS(*depth*)
if *depth* $\geq$ *best* **then**
  return
end if
if **A** == 0 **then**
  update *best*
else
  for all l_C in **L** **do**
    compute all $face \in D$ for L_C;
    select $face$ with 2-coverage;
    store L_C selected;
  end for
  sort **L** by descending order;
  store $L_C \rightarrow Queue$
  while $Queue \neq \emptyset$ **do**
    $L_C \leftarrow Dequeue$
    DFS(*depth*+1)
    add removed undeployed L_C back to **L**
  end while
end if
return *list*

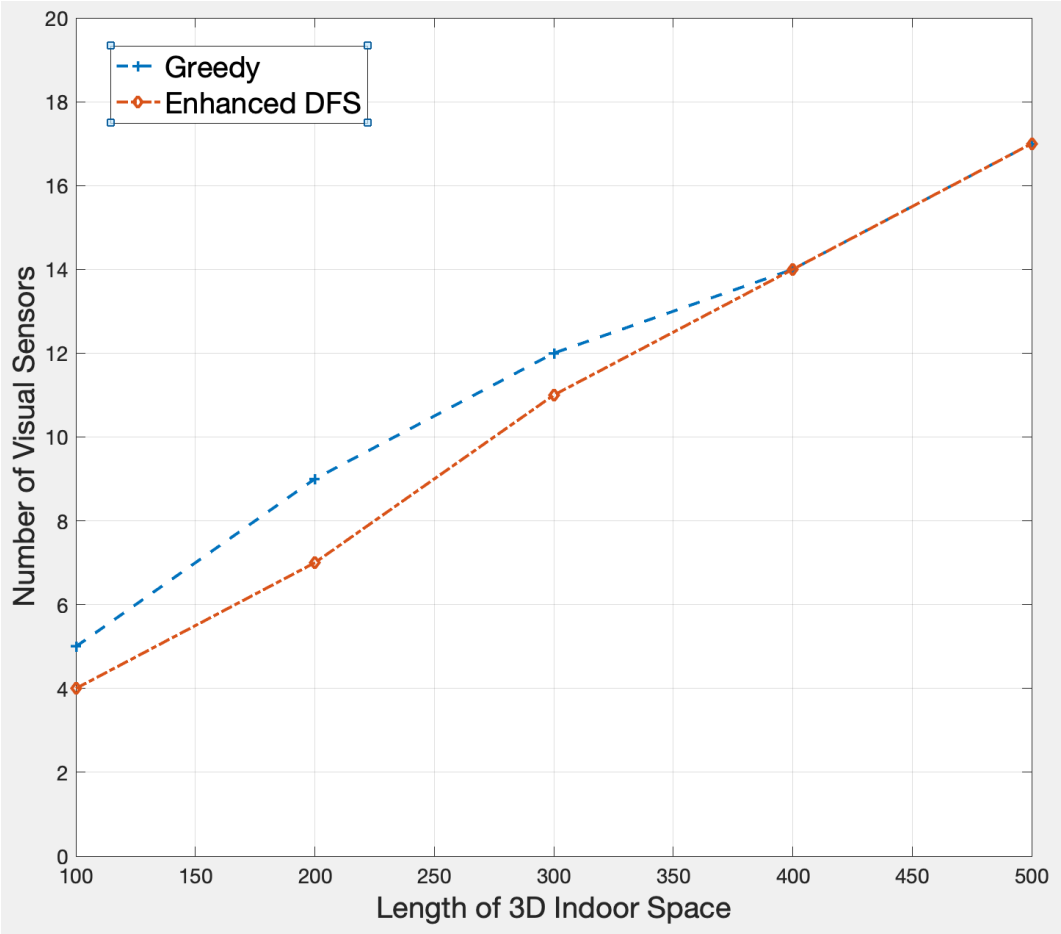

Figure 2.5. Varying Length of 3D Indoor Space

CHAPTER 3

TRAFFIC LOAD BALANCE AT NETWORK LAYER

3.1 BACKGROUND

In order to prolong the lifetime of entire wireless visual sensor network, an efficient network layer routing protocol has to be carefully designed. We note that although most of existing works on traffic load balance in WSNs are modeled by tree topology rooted at the base station such as works done by (Soro and Heinzelman, 2005, 2009), using the topology may yield sub-optimal results since it eliminates the opportunities that some sensor nodes in the network may connect to multiple neighbors closer to the base station to further balance the traffic loads. One example is illustrated in Figure 3.1, where each sensor nodes B to F need to report 1 unit of visual data traffic flow to the base station A. If the tree topology is used, the bottleneck will be 4 units of traffic flows on B or C. If we carefully consider over the entire network topology, a better solution can be obtained to further reduce the bottleneck on node B or C to 3 units of traffic flows, which represents a 25% of reduction from using the tree topology. Although it might be argued that multiple tree topologies can be used alternatively so as to still have a better traffic load balance, yet the involved multiple topology maintenance and the switch overhead can be non-trivial and easily offset the possible load balance benefit therein. Therefore, instead of tree topology we propose to optimize the traffic load balance by directly working on the full network topology, transforming the problem to find the most load balanced topology into a generalized max-flow problem and proposing an efficient algorithm to solve it.

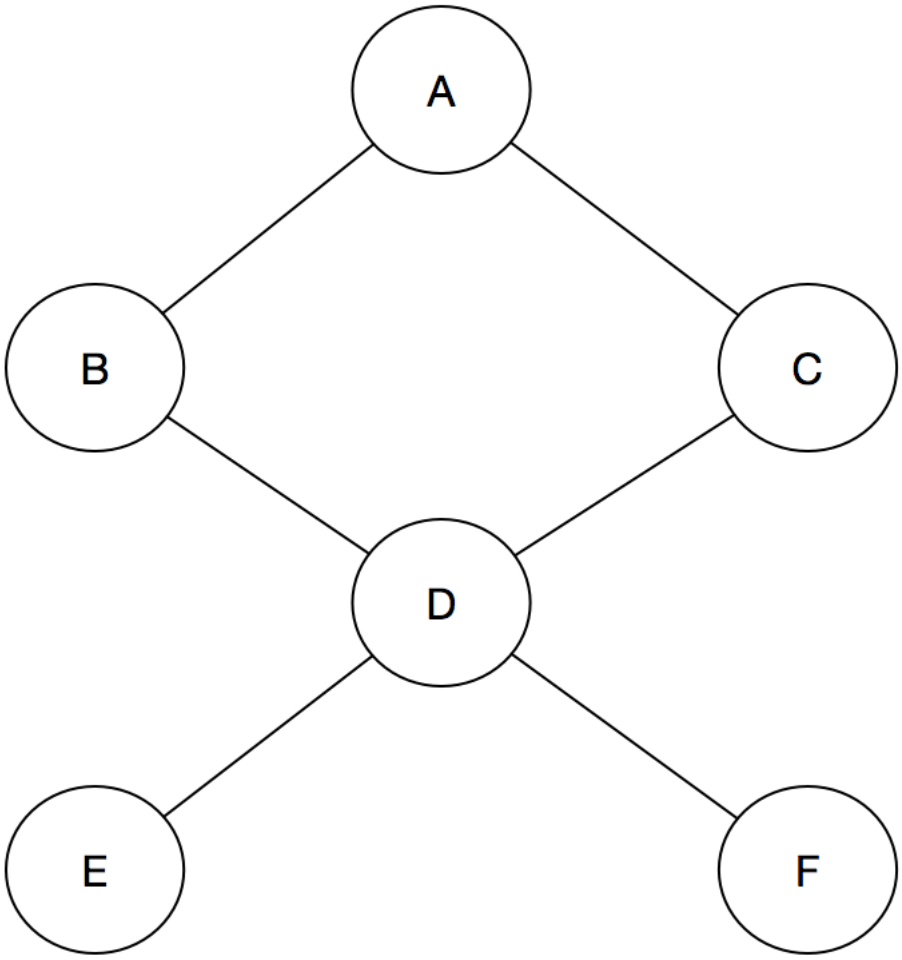

Figure 3.1. Example on limitation of using tree topology for traffic load balance.

3.2 RELATED WORK

WSNs have renovated many research areas by providing reliable and scalable technologies over the past decade. As one of the most recent advances of WSNs, a fast-growing interest is also seen in WVSNs Tavli et al. (2012). Although there have been numerous studies to explore and address the deployment problem in WSNs Raha et al. (2012); Akshay et al. (2010); Li et al. (2003), the unique feature on wireless visual sensor and the resulting differences on sensing range, device cost, data collection pattern, etc. Render these previously proposed solutions cannot be directly applied to solve the problem in WVSNs Megerian et al. (2005).

To this end, Chow et al. Chow et al. (2007) proposed a simple model to provide maximum angular coverage in WVSNs to generate a minimum set of sensors to cover all objects of interests. In Sheramin et al. (2010), a Depth First Search algorithm was developed to address both sensing coverage and network connectivity issues. These pioneering works mainly considered 2D deployment scenarios, which, however, may be less effective and inaccurate when being applied to 3D indoor monitoring scenarios. In Liang et al. (2011), the objective of the paper is to maximize the area coverage of a randomly deployed directional sensor using a greedy algorithm. A directional sensor network is implemented in Wu and Lu (2013) for coverage problem with consideration to rotatable angle. However, both of them are formulated in a simple 2D model and similar greedy algorithms to solve the problems. A novel approach to sensor deployment is provided in paper TYen (2014) where a PTZ camera is used in a WVSN. The difference between a PTZ WVSN and a traditional WSN has been highlighted in that there are extended FoV coverage and semi data source nodes. The work in paper Yen (2013) is very similar to ours, where a WVSN is used for the sensor coverage problem. Using a greedy heuristic approach FoVIC where the target is to cover the maximum number of uncovered nodes within the area via a mathematical model. Again, both papers TYen (2014) and Yen (2013) are modeled using a 2D approach (i.e. the latter emphasizing a mathematical model) and use numerical analysis to evaluate their solution.

On the other hand, Kouakou et al. Kouakou et al. (2012) tackled the problem of indoor space target coverage by formulating a k-coverage problem in 3D space, where the proposed heuristic algorithm can ensure k-coverage of the monitored areas with the consideration of deployment costs and obstacles. To address target coverage with PTZ cameras, Munishwar et al. Munishwar and Abu-Ghazaleh (2013) formulated the maximal clique generation problem and transformed it into the MaxFoV problem, where two polynomial time algorithms were proposed to reduce the number of candidate FoVs (B-EFA) and find an optimal set of FoVs for PTZ cameras (G-EFA). The mainly limitation of the paper mentioned above in Kouakou et al. (2012) is that they only provided a greedy heuristic algorithm to tackle the problem which cannot solve the problem optimally, where our solutions, not only provide the fast solution through greedy heuristic algorithm, but also supply an optimal solution named enhanced-DFS which can optimally solve the problem. The authors in Peng et al. (2013) proposed a greedy iteration scheduling based algorithm for coverage rate optimization with a 3D model. However, one assumption of their work is that the monitored scene of the 3D directional sensor network is a 2D plane, and the location of each sensor is on the same horizontal plane which highly limited the ability of using 3D model, where our solution is monitoring the entire 3D space area and the location of sensor is not limited on the same horizontal plane as they did.

3.3 PROBLEM FORMULATION

In this chapter,the problem formulation is based on previous working step. Our aim is to use minimum number of visual sensors to fully cover all the required 3D indoor areas. Moreover, the visual traffic loads collected at visual sensors must also be well balanced among the network in order to alleviate the potential bottleneck and prolong the over all lifetime.

We use R_{max} to denote the communication range achievable for each wireless visual sensor. Our objective is to find the deployment plan $S = \{(L_1, D_1), (L_2, D_2), ...\}$ and the data path $P_{(L,D)}$ for each sensor node $(L, D) \in S$, so as to (a) minimize $|S|$, and (b) minimize

the traffic load of the wireless visual sensor with maximum traffic load among all the sensor nodes $\min[\max_{1\leq i\leq N} TL_i]$ subject to the following constrains:

(1) Area Coverage Constraint:

$$\forall a \in \mathbf{A}, \exists (L, D) \in S, \text{ such that } a \in cover(L, D, R_S) \text{ and } \mathbf{A} \subseteq \sum_{\bigcup_{(L,D)\in S}} cover(L, D, R_S) \ ;$$

where $cover(L, D, R_S)$ represents the subset of particles covered by visual sensor (L, D, R_S).

(2) Network Connectivity Constraint:

$$\begin{aligned} &\forall (L, D) \in S, \exists P_{(L,D)} = \{(L_{p1}, D_{p1}), (L_{p2}, D_{p2}), ..., (L_{pk}, D_{pk})\} \\ &\text{such that } |LL_{p1}| \leq R_{max}, |L_{pk}BS| \leq R_{max} \\ &\text{and } |L_{pj}L_{pj+1}| \leq R_{max} \text{ for } j = 1, 2, ..., k-1. \ ; \end{aligned}$$

where $P_{(L,D)} = \{(L_{p1},D_{p1}),(L_{p2},D_{p2}),...,(L_{pk},D_{pk})\}$ denotes the communication path from a visual sensor node (L, D) to the base station BS. For a sensor (L', D'), you can just use whether $(L', D') \in P_{(L,D)}$ to test if this sensor is on the path from sensor (L, D) to the base station. $TL_{(L,D)}$ used to denote the traffic load on wireless visual sensor (L, D). Without loss of generality, we assume the visual data traffic rate collected from each visual sensor is 1 unit. We then have

$$TL_{(L,D)} = \sum_{1 \leq m \leq N} (L'_m, D'_m) \times 1(unit)$$

3.4 Solutions

In this section, we first present our load balance algorithm that can achieve optimal solution for a given network topology. We then use it as a building block to further solve our formulated WVSN deployment problem for 3D indoor monitoring.

3.4.1 Optimal Load Balance on Given Network Topology

Common strategies to balance the traffic loads flowing in WSNs are mainly based on tree topology, with the base station as the root. Yet, as aforementioned in Figure 1, using tree topology may yield sub-optimal results since it eliminates the opportunities that some sensor nodes in the network may connect to multiple neighbors closer to the base station to further balance the traffic loads. To this end, we transform the load balance problem into a generalized max-flow problem to optimally balance the traffic loads. In particular, as illustrated in Figure 3.2, we add a virtual source node which directly connects with each sensor node in the given network topology except for the base station. We call each edge added to connect the source node and one non-base-station node a virtual edge and assign 1 unit of traffic flow as the link capacity on each virtual edge. The maximum number of traffic flows that can be delivered from the source to the base station is thus equal to the number of non-base-station nodes in the topology. In order to count the traffic flows through each non-base-station node rather than counting flows on edge, each non-base-station node is divided into two sub-nodes as *entrance-node* and *exit-node*, and then an edge is added to connect them. Every traffic flow originally going through the non-base-station node now must go into its *entrance-node* and out from its *exit-node*. Initially, we assign 1 unit of traffic flow as the link capacity to each non-virtual edge. The allowed max-flow to the base station

is then calculated based on the condition given above. For each following iteration, the link capacity of each non-virtual edge will be increased by 1 unit until the calculated max-flow is equivalent to the number of traffic flows needed to be delivered from the source node (i.e., the number of non-base-station nodes). Then the path for each non-base-station node can be obtained from the corresponding max-flow result.

Theorem

Theorem 2. *On any given network topology, Optimal Load Balance algorithm can return the optimal solution for minimizing the bottleneck of traffic load flows.*

Proof. The Optimal Load Balance algorithm is transformed from generalized Max-flow algorithm showing above. The correctness of Optimal Load Balance algorithm can be proofed by contradiction. Assume there is another optimal solution that can achieve N with $k' < k$ while our solution can achieve N with k. Then it means that on the topology, with each non-virtual edge of capacity k', the max-flow algorithm can already return N. However, for our algorithm to return k, it means that with k', the max-flow algorithm cannot return N. So that we can proof that the Optimal Load Balance algorithm can produce optimal solution. □

3.4.2 Greedy Heuristic Algorithm

With the optimal load balance algorithm as a building block, we next present the design of our Greedy Heuristic algorithm based on our previous work in Brown et al. (2017), where we can garner locally optimal candidate locations and use them as the start point for our enhanced depth first search algorithm to be proposed in the next subsection. Algorithm 2 shows our Greedy Heuristic algorithm, which, in each iteration, deploys an sensor node the candidate location that can cover the maximum number of fresh grid vertices within the monitored space **A** (i.e., the grid vertices that have not been covered by any previously deployed sensor node) while considering the traffic load bottleneck. Specifically, we use a while loop to check if any coverable 3D regions and candidate placement locations are

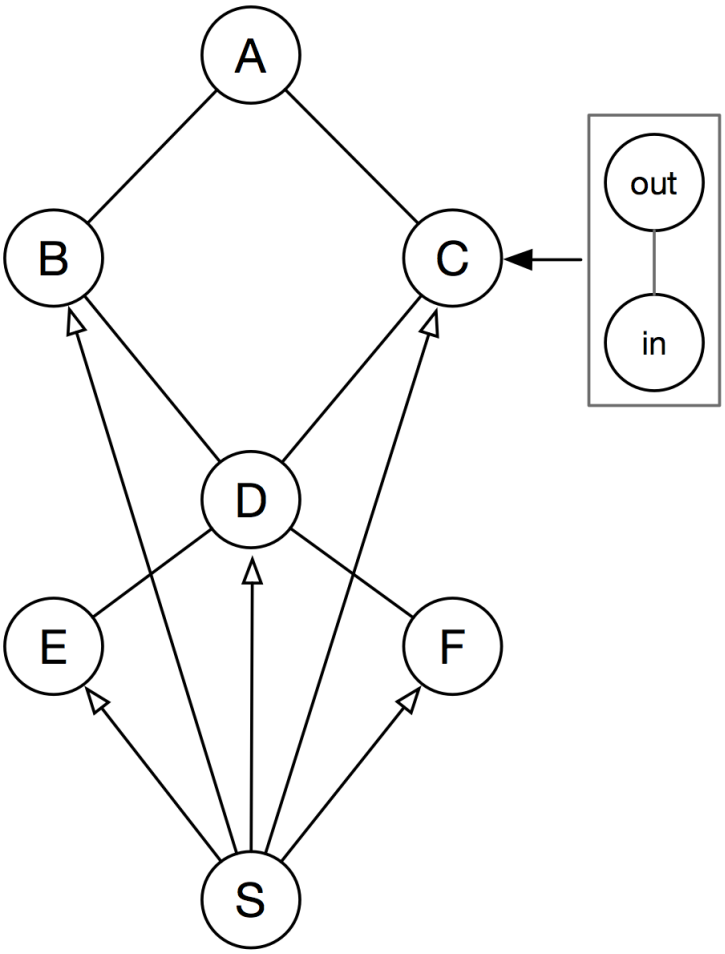

Figure 3.2. Transformed max-flow graph based on Figure 3.1. (A: *base station*, S: *virtual-source*, arrow-edge: *virtual-edge*, in: *entrance-node* of C, and out: *exit-node* of node C)

Algorithm 3 Greedy Heuristic Algorithm

Input: **A**, D, **L**
Output: Set of S

Initialize: List of **A**, D, **L** and $final$.
while **L** $\neq \emptyset$ and **A** $\neq \emptyset$ **do**
  for all C_L in **L** **do**
    if C_L within R_{max} of any deployed visual sensor **then**
      for all $face \in D$ for L **do**
        if $C_{L\text{-}F_c*} > C_{L\text{-}F_c}$ **then**
          update $C_{L\text{-}max}$
        end if
      end for
      record $C_{L\text{-}max}$ into *templist*
    end if
  end for
  sort *templist* based on the number of F_c
  record $C_{L\text{-}max}$ in *templist* when more than one $C_{L\text{-}max}$ exist
  for all $C_{L\text{-}max}$ **do**
    pick up the one with minimum traffic flow bottleneck
  end for
  remove $C_{L\text{-}max}$ from **L** and update **A**
end while
return $final$

available (line 1). In the for loop we evaluate each candidate location C_L in $\mathbf{L}$, checking to see if it is within the communication range, R_{max} (line 3). For each facing direction $face \in D$ (where D denotes the set of facing direction), we check each monitoring point which is within the covered field by current candidate location C_L. We compute all the $face$ for the candidate locations C_L and pick the $face$ covering the maximum number of monitoring points F_c (line 4-8), where F_c denote the points set that are already covered. After the for loop, we record all the candidate locations into a *tempList* for later using and store the max F_c number with each candidate location. Then, the *tempList* is sorted by descending order of the max F_c number. After sorting the *tempList*, if more than one C_L with the same F_c number on the top rank, these candidate locations will be recorded and then evaluated based on our Optimal Load Balance algorithm to calculate the traffic load bottleneck. They are then resorted increasingly according to the traffic load bottleneck. These candidate locations will be recorded and each one of them will be evaluated based on the Optimal Load Balance strategy to calculate the traffic load bottleneck which is a number. Resort the top rank candidate locations rely on the number of traffic load bottleneck by ascending order. At the end of each iteration of the *while* loop, the Greedy Heuristic Algorithm then choose from current $\mathbf{D}$ the candidate location where the deployed visual sensor can cover the maximum number of vertices F_c as well as cause the minimum traffic load bottleneck (line 12-16). After the candidate location is chosen by the Greedy Heuristic Algorithm and a visual sensor is deployed at that location to cover maximum fresh vertices, then the candidate location C_L is recorded into final list, and these vertices are then removed from $\mathbf{A}$ and the chosen candidate location is removed from $\mathbf{L}$. We then start a new iteration to find next candidate location C_L that will cover the maximum number of fresh vertices in $\mathbf{A}$ and cause minimum traffic load bottleneck. The final list of $S = \{(L_1, D_1), (L_2, D_2), ..., (L_n, D_n)\}$ that will fully cover the 3D monitored space is then returned (line 19).

Algorithm 4 Enhanced-DFS Algorithm

Input: **L**, **A**, D
Output: Minimized set of $|S| = n$,
Initialize: List of **A**, D, **L** and *best*

```
DFS(depth)
if depth ≥ best then
  return
end if
if A == 0 then
  update best
else
  for all C_L in L do
    compute all face ∈ D for C_L
    select face with max F_c
  end for
  sort tempList by descending order according F_c
  for all rank (i.e., all C_L with the same max F_c) in tempList do
    resort each rank by ascending order
  end for
  store tempList → Queue
  while Queue ≠ ∅ do
    C_L ← Dequeue
    if C_L within R_max of any deployed visual sensor then
      if lowerBound(F) + depth ≥ best then
        break
      end if
      record C_L* and F_c* then remove them
      DFS(depth+1)
      add F_c* back to A
    end if
  end while
  add all removed C_L* back
end if
return best
```

3.4.3 Enhanced Depth First Search Algorithm

The Greedy Heuristic algorithm deploys visual sensors one by one with local optimal decisions, which may still be sub-optimal from a global point of view. To this end, we further propose an Enhanced-DFS algorithm, which can not only further improve the quality of the solutions found by the Greedy Heuristic algorithm, but also obtain the optimal solutions from the discrete problem if given enough time. Different from Traditional DFS algorithms that exhaustively evaluate each searching branch, out Enhanced-DFS algorithm use branch pruning to cut the candidate location searching branches that are guaranteed not able to improve the currently found best solution, so that the explored solution space can be significantly reduced. Algorithm 3 summarizes our Enhanced-DFS algorithm, where *depth* denotes the number of used visual sensors to cover the space in the current search branch, and *best* is initially the $final$ list obtained from our Greedy Heuristic algorithm. Our first enhancement is to determine its feasibility by comparing the current search branch to out *best* found solution (line 2). We next check to see if there are remaining points to be covered in $\mathbf{A}$, if no we update *best* and go one step backward along the current search branch; otherwise we continue (line 7-31). To further improve the solution efficiency, our second enhancement selects the location in each iteration by sorting all of C_L in descending order based on the maximum number of fresh vertices that a visual sensor placed at the corresponding location can cover and the produced facing direction is handled in a similar way. In each rank of the sorted list (where all the considered visual sensors can cover the same number of fresh vertices), if there are more than one candidate locations on that rank, another sorted function will be called to further sort these candidate locations by ascending order based on traffic load bottleneck number (line 8-16). This can further reduce the chances to select infeasible solution options so as to quickly find better quality candidate location options. A similar enhancement approach is used when selecting the facing direction of a visual sensor. We also ensure in the algorithm, that all the candidate locations in the list have a communication path to the base-station (line 20). The next enhancement is defined

by the following constraint:

$$\frac{F}{F_c^1} + depth < best$$

where F denotes the remaining fresh vertices ready to be covered, F_c^1 represents the number of covered vertices by the first visual sensor of current set S, (where we choose F_c^1 as denominator due to that since all the candidate locations has been sorted before choosing by descending order, the number of fresh vertices covered by the first visual sensor is the maximum compared with others, which can guarantee the best solution will not be pruned). (line 21-25) The function of this constraint is to bound the currently estimated branch that the number of candidate locations to cover the remaining fresh vertices plus the *depth* must be smaller than the currently found *best* solution, otherwise the branch will be pruned. An update is done by removing the candidate position in **L** where visual sensors have already been deployed and removing the accordingly covered vertices from **A**. Next, we recursively call the enhanced DFS search to explore the remaining search branches (line 26). A minimized set of $|S| = n$, where S is optimal is returned in best as shown in line 32.

3.5 PERFORMANCE EVALUATION

We conduct extensive simulations to evaluate our solution using a customized simulator implemented by JavaScript, which can emulate the 3D deployment of wireless visual sensor nodes in a virtual environment. Table 3.1 outlines the default parameter settings that are utilized in the performance evaluation. For comparison, we implement a baseline algorithm that randomly selects locations in **L** to deploy visual sensors without considering traffic load balance. Once a location is selected, the facing direction of the visual sensor is adjusted to greedily cover the maximum number of fresh grid vertices in **A**. If the visual sensor cannot cover any fresh vertices, it will be removed and another random location will be selected, and the candidate location must satisfy the network connectivity requirement

as well. This process will continue until all the grid vertices in **A** have been covered. For each setting, we run the baseline algorithm for 100 times and show the average of the results with an error bar to indicate the minimum and maximum values.

Figure 3.3 shows the number of visual sensors deployed by different approaches with various 3D indoor space. As expected, that with the size of the 3D indoor space increasing, more visual sensors are required to cover the indoor space. However, our greedy algorithm produces a 62.5% reduction on the number of visual sensors compared with the baseline algorithm. The Enhanced-DFS can further outperform Greedy algorithm by up to 17%. It is worth nothing that although we run the simulation on a standard PC with configurations listed as: i7 processor, 16GB RAM, 1TB SSD and set a short time limitation (1 *hour*) for the enhanced DFS solution in our evaluation, it can still successfully return the optimal results for the case with the length of 3D indoor space equal to 100 and 200. We thus conjecture that our Enhanced-DFS solution can return optimal results for more cases if it runs on a local or cloud server that has much more computation power and more time is allowed. In Figure 3.4, we show the trade-off between the number of visual sensors required to cover the entire 3D indoor space and the bottleneck of traffic load. As less number of sensor nodes can cause fewer options of forwarding paths, which leads to larger bottleneck of traffic loads as illustrated in Figure 3.4. Meanwhile, the greedy heuristic algorithm and enhanced-DFS algorithm tremendersly outperform the random algorithm which is the base line by 73%. In Figure 3.5, we further examine the performance of our optimal load balance algorithm by comparing it with the solution that uses the best tree topology that can be found to minimize the traffic bottleneck, where both approaches are working on the network toplogies generated by our greedy heuristic algorithm. It is clear to see that our approach can further reduce the bottleneck of traffic load by up to 20%, further demonstrating its optimality on minimizing the traffic load bottleneck.

Testing Parameters	
Parameter	Default Value
3D indoor space	$Length \times 60 \times 100$
A	Same as 3D indoor space
L	Top half of walls and ceiling
g_A	25
g_L	25
g_D	15 degrees
FOV	50 degrees
Aspect Ratio	1.7778
Near Field	1
Far Field	100

Table 3.1. Simulation parameter settings

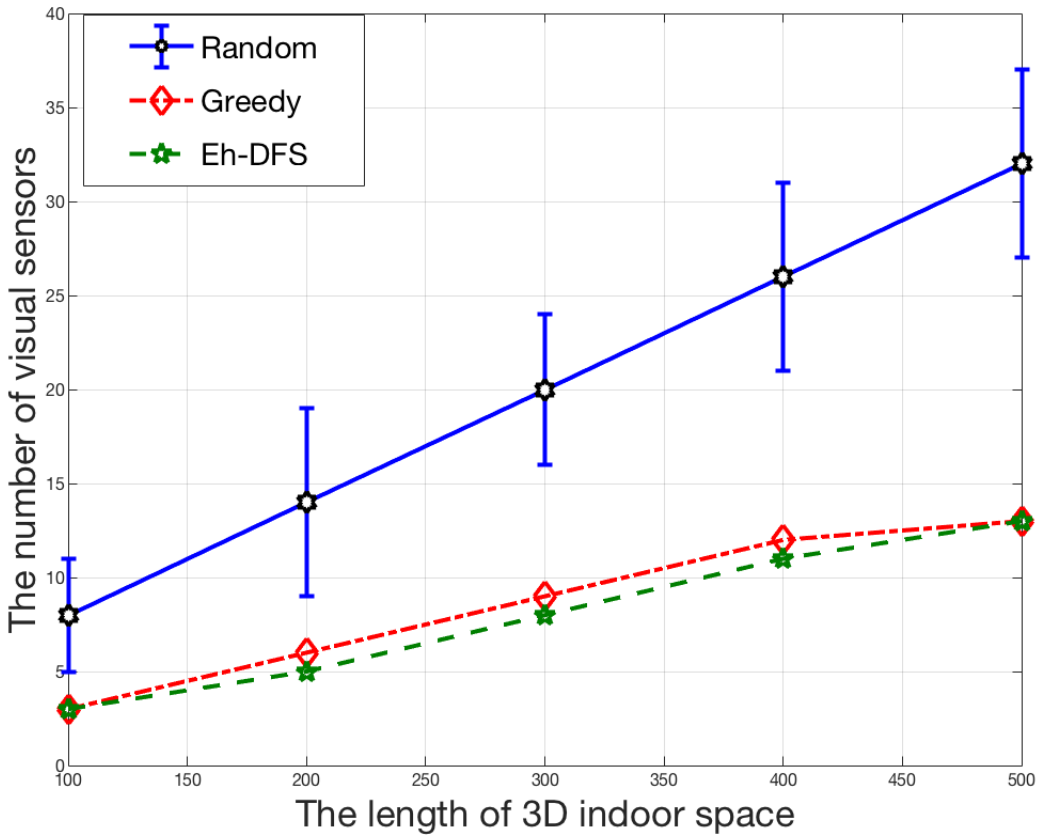

Figure 3.3. Variation of Indoor Space

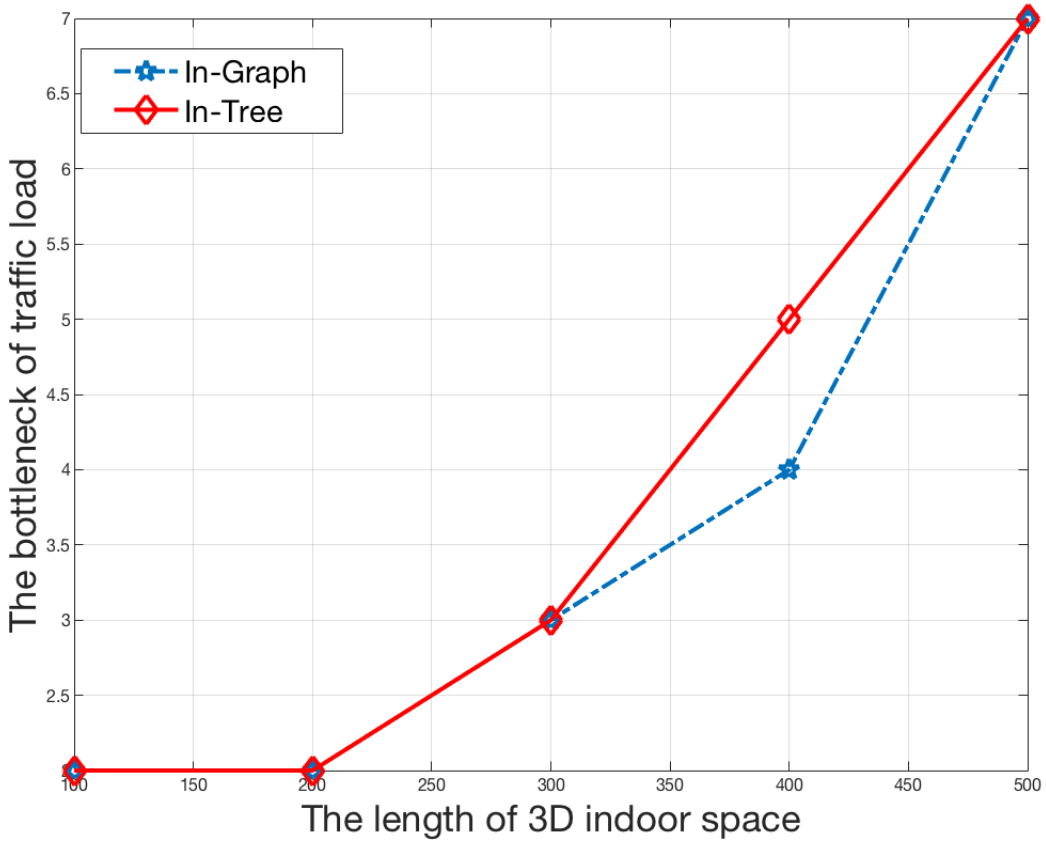

Figure 3.4. Variation of Traffic Bottleneck

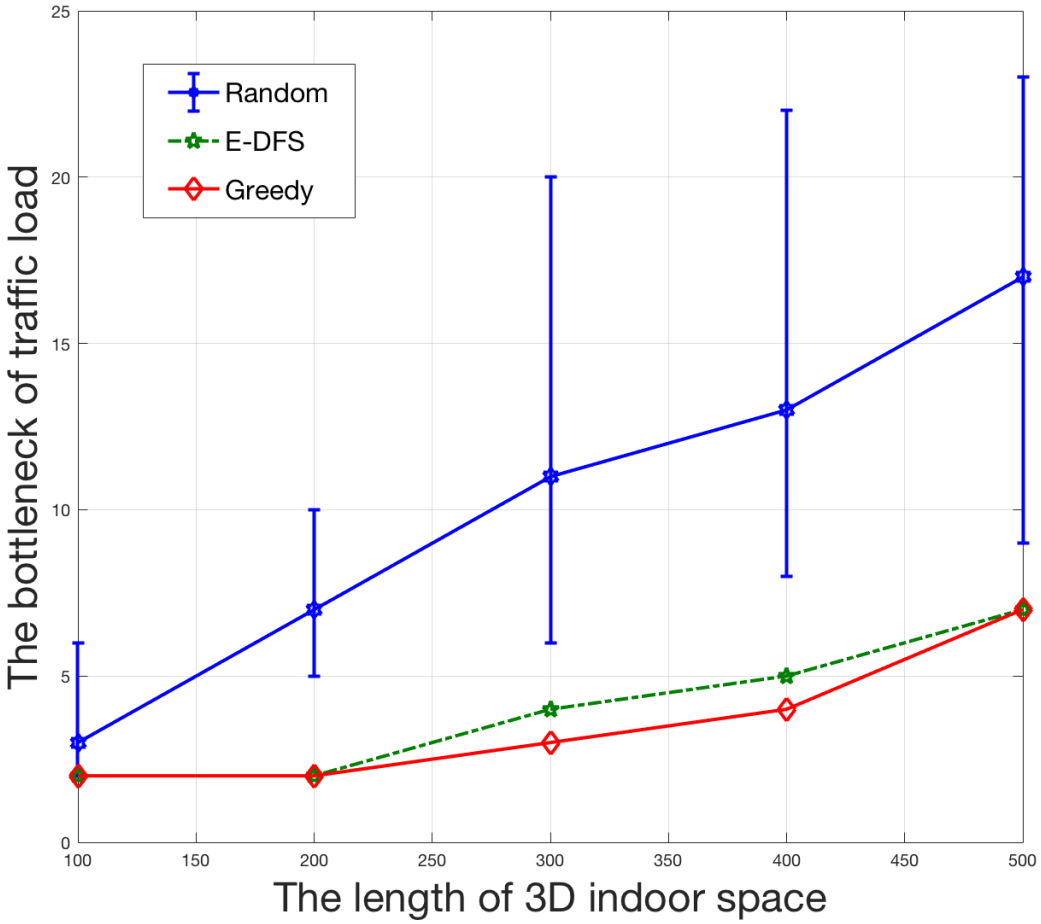

Figure 3.5. Traffic Load Variations in Different Structures

CHAPTER 4

LINK LAYER SCHEDULING: A CROSS-LAYER DESIGN APPROACH

4.1 BACKGROUND

Wireless visual sensor networks (WVSNs) play a prominent role in a wide range of applications such as environmental monitoring, military, industrial, medical, and security Akyildiz et al. (2002); Huang and Tseng (2005) where a typical sensing environment is a 3D indoor space with visual sensors attached to walls and/or ceilings to monitor activities of targets within the sensing field. For example, the newly emerged shopping applications such as Amazon go and Alibaba Tao Cafe [1], where customers may purchase products without being checked out by a cashier or using a self-checkout station. All the operations such as product counting, user account matching, and payment processing are achieved through visual sensors with deep learning algorithms and data fusion technologies. However, all the visual sensors are wired connected for the current version. The wireless version is needed to address the unexpected situation such as power outage, emergent moment. Dislike traditional sensors such as temperature or light sensors often having ominidirectional sensing range, fan-shape and pyramid-shape are often considered for the sensing range of a visual sensor in 2D and 3D Xu et al. (2015), respectively, which renders the deployment solutions for traditional sensors and 2D sensing field incapable of solving the WVSN deployment problem for 3D indoor space monitoring Deif and Gadallah (2013); Han et al. (2012); Raha et al. (2012); Horster and Lienhart (2006); Morsly et al. (2011); Zhang et al. (2009); Newell et al. (2010); Wang et al. (2019); Brown et al. (2017).

Moreover, as all the visual streaming data generated from each visual sensor are

[1] https://syncedreview.com/2018/01/22/amazon-go-vs-alibaba-tao-cafe-staffless-shop-showdown/

required to be collected back to the base station for further processing, this also differs from the data aggregation approaches often used in traditional wireless sensor networks Krishnamachari et al. (2002); Fasolo et al. (2007). If the transmission path for each visual sensor cannot be carefully handled, a sensor can be easily overloaded. Furthermore, if each sensor is powered by battery (e.g., during a power outage or an emergency moment), the network lifetime can be significantly shortened due to the quickly draining off of the battery power. To this end, we note that most of existing solutions for traffic load balance in Wireless Sensor Networks (WSNs) are modeled by tree topology rooted at the base station such as Soro and Heinzelman (2005, 2009). However, using tree topology may eliminate the opportunities that some sensor nodes in the network may connect to multiple sensors closer to the base station to further balance the traffic loads, and sub-optimal results may be yielded through such tree topology based strategies. Therefore, it is necessary to consider the full network topology to maximize the benefit of load balance.

In addition, to make sensor nodes share the wireless media fairly and effectively, a Link layer scheduler is usually employed at the MAC protocol level to ensure efficient communications in the network. Yet due to the extremely large size of captured visual data in WVSNs compared with scalar data in WSNs, the communication scheduling solutions proposed for WSNs are often found insufficient to solve the problem in WVSNs Kacimi et al. (2013); Lipare et al. (2019). Therefore, the deployment of WVSNs must not only guarantee the 3D indoor area coverage and connectivity between visual sensors and the base station, but also ensure that the scheduling of data collection and communication is organized fairly and efficiently in order to minimize the communication overhead, shorten the total latency, and prolong the lifetime of the entire network.

To address all these aforementioned issues, in this chapter we take a cross-layer design approach, striving to provide a holistic solution across visual sensor coverage, end-to-end streaming data delivery, network traffic load balance and wireless link scheduling. In particular, we first model the wireless visual sensor deployment and scheduling as a cross-layer

optimization problem. We then use a divide and conquer strategy to further study two subproblems, i.e., for a given network topology, how to achieve optimal traffic load balance and then optimal link scheduling, respectively. For the former subproblem, we work on the full network topology, transforming the traffic load balance problem into a generalized max-flow problem and propose an efficient algorithm accordingly. For the latter, we customize the TDMA MAC protocol by developing a busy link first edge coloring algorithm that can well afford heterogeneous loads among different links to further optimize the link scheduling. With these building blocks, we further develop SEED, a SchEduling-aware Enhanced Depth-first search algorithm that can effectively prune sub-optimal searching space and yield optimal solution for the original cross-layer optimization problem if given enough time.

We have conducted extensive simulations to carefully examine our proposed solution. The results show that our solution can greatly reduce the overall communication latency up to 39% compare to standard TDMA solution, and achieve excellent load balance by reducing up to 20% of load on the bottleneck compared to the strategy with tree topology while minimizing the number of deployed visual sensors up to 64% compared to the baseline algorithm.

The remainder of the chapter is organized as follows. In Section II, we present the related works. Section III provides our problem formulation. In Section IV, we first discuss our two sub-solutions for full network topology based load balance as well as cross-layer link scheduling, respectively, and then use them as building blocks to further develop SEED, a SchEduling-aware Enhanced Depth-first search algorithm that is capable to yield optimal solution. Evaluation results are presented and discussed in Section V.

4.2 RELATED WORK

WSNs have renovated many research areas by providing reliable and scalable technologies over the past decade. As one of the most recent advances of WSNs, a fast-growing interest is also seen in WVSNs Tavli et al. (2012). Although there have been numerous

studies to explore and address the deployment problem in WSNs Raha et al. (2012); Akshay et al. (2010); Li et al. (2003), the unique features of wireless visual sensors and the resulting differences on sensing range, device cost, data collection pattern, etc. render these previously proposed solutions cannot be directly applied to solve the problem in WVSNs Megerian et al. (2005).

To this end, a Depth First Search algorithm was developed to address both sensing coverage and network connectivity issues in Sheramin et al. (2010). Chow et al. Chow et al. (2007) proposed a simple model to provide maximum angular coverage in WVSNs by generating a minimum set of sensors to cover all objects of interests. In paper TYen (2014), the authors proposed a novel approach to sensor deployment problem, where a PTZ camera is used in a WVSN. The similar work to ours is proposed in paper Yen (2013), where a WVSN is used for sensor coverage problem. A greedy heuristic approach FoVIC has been proposed where the objective is to cover the maximum number of uncovered nodes within the area via a mathematical model. However, these pioneering works mainly considered 2D deployment scenarios, which become less effective and inaccurate to solve the problems in 3D indoor monitoring.

On the other hand, the authors in Kacimi et al. (2013) proposed a heuristic algorithm which combines load balancing with transmission power control to address the traffic load balancing problem in WSNs. In Lipare et al. (2019), the Grey Wolf Optimization approach for energy-efficient clustering and routing has been applied to tackle the load balancing problem in WSNs. However, these aforementioned works are mainly focused on WSNs. TDMA MAC protocol has been employed by Gandham et al. (2008) to address the problem of link scheduling. Two phases were consisted in their link scheduling algorithm. The authors attempt to assign a transmission direction along each edge, so that the hidden terminal problem can be avoided. Moreover, reversing the direction of transmission along every edge leads to another feasible direction of transmission has been demonstrated by their work. Pantazis et al. (2009) proposed a TDMA scheduling scheme for energy efficiency

which achieves high levels of power conservation while reduces the end-to-end transmission delay. They claim that only one sleep interval for the end-to-end transmission from the sensors to the appropriate gateway need to be sacrificed as delay when using their WakeUp intervals solution. A multi-objective TDMA scheduling problem for many-to-one wireless sensor networks was presented by Mao et al. (2007), which is used to save this part of energy and further improve the time performance. An effective optimization framework has been proposed which combine genetic algorithm and particle swarm optimization algorithm together to enhance the searching ability. They demonstrated that the proposed hybrid algorithm outperform these two algorithms on total time or total energy for data collection. However, the aforementioned existing solutions are all dedicated for WSNs, that all deal with scalar data instead of visual streaming data (i.e., the larger size of visual data). Therefore, the solutions for WSNs are incapable for solving the problem in WVSNs. Based on our knowledge, this is the first work addressing scheduling problem in WVSNs.

4.3 PROBLEM STATEMENT

We consider a 3D indoor space, where some 3D areas are required to be fully covered by visual sensors. Only certain locations can be used to deploy the visual sensors such as ceiling and/or the upper half of the wall. After a visual sensor is deployed, its facing direction will be adjusted to a certain direction and then stay there during the whole monitoring period. Our first objective is to use minimum number of visual sensors to fully cover all the required 3D indoor space. Beyond this basic requirement, there are other issues that may also need to be considered, e.g., the 3D indoor space may not be covered by WiFi, or the WVSN is required to communicate out of the WiFi band to minimize the interference with the normal WiFi traffic. Therefore, the deployment must ensure that the entire network is fully connected by its own wireless communication to the base station. Another issue is that since the deployment of the WVSNs may not be considered during the building design and construction phase, which means that the continue power supply may not be available to

the visual sensors, the deployment then needs to consider the network lifetime if the visual sensors can only be powered by batteries. Furthermore, the traffic load on each visual sensor and the number of time-slots used to transmit visual data back to the base station must also be minimized and well scheduled in order to shorten the transmission delay and prolong the overall lifetime of the entire network.

Let $\mathbf{A}$ represents the areas that need to be monitored in the given 3D indoor environment. We use $\mathbf{L}$ to denote the candidate location set (e.g., ceiling and upper half of walls) where the wireless visual sensor can be deployed. A tuple (L, D) represents the visual sensor that is deployed at the location $L \in \mathbf{L}$ with D as the facing direction. We divide the continuous space in $\mathbf{A}$ and $\mathbf{L}$ into discrete grids, and use g_A to denote the granularity of the grids used in $\mathbf{A}$ and g_L the granularity in $\mathbf{L}$. In addition, we also divide the surface of the facing direction sphere into grids (e.g., like the longitudes and latitudes divide the surface of the earth) and use g_D to denote the granularity.

We assume that, time is divided into equal intervals referred to as time-frames, and each time-frame can be further split into slots of identical length referred to as time-slots. We use *slot* to denote one single time slot. The considered wireless visual sensor network can be deemed as a graph $G = (V, E)$. Each $v \in V$ denotes a wireless visual sensor, and every $e \in E$ represents the communication link between two sensors or between a sensor to the base station. R_{max} denotes the maximum communication range achievable for each wireless visual sensor node. Our objectives are thus to find the deployment plan $P = \{(L_1, D_1), (L_2, D_2), ...\}$, and the contention-free link layer scheduler. As visual data streaming is often collected in a round by round manner in WVSNs, we let the data generated at node v_i be r_i packets per round, for $i = 1, 2, ..., |V|$, and v_0 represents the base station. Let t_0 denotes the time that a data collection round starts.

Define a link schedule as $Schedule = \{(l_{x_1,y_1}, t_1)\ , (l_{x_2,y_2}, t_2), ..., (l_{x_k,y_k}, t_k)\}, t_0 \leq t_1 \leq ... \leq t_k$, where (l_{x_i,y_i}, t_i) means that the link at edge (x_i, y_i) will be activated at time t_i to send a packet from sensor x_i to y_i. The deployment and traffic scheduling together can thus

be formulated as a cross-layer optimization problem to find a deployment plan P and link schedule $Scheduler$, subject to the following constrains:

(1) Area Coverage Constraint:

$$\forall a \in \mathbf{A}, \exists (L, D) \in P, \text{ such that } a \in cover(L, D, R_S) \ ,$$

where R_S is the sensing range of a visual sensor and $cover(L, D, R_S)$ represents the 3D space covered by the visual sensor at (L, D).

(2) Interference-free Constraint:

$$\begin{aligned} &\forall (l_{x_i,y_i}, t_i), (l_{x_j,y_j}, t_j) \in Scheduler, \text{ if } t_i = t_j, \\ &\text{then, } |x_i, x_j| > R_{max}, |x_i, y_j| > R_{max}, |y_i, x_j| > R_{max}, \\ &|y_i, y_j| > R_{max}, x_i \neq x_j, x_i \neq y_j, y_i \neq x_j, y_i \neq y_j. \end{aligned}$$

(3) Traffic Source Constraint:

$$r_i + \sum_{(L_{x_j,y_j},t_j) \in Schedule} I_{[y_j = v_i]} = \sum_{(l_{x_j,y_j},t_j) \in Schedule} I_{[x_j = v_i]};$$

(4) Traffic Destination Constraint:

$$\sum_{(l_{x_j,y_j},t_j) \in Schedule} I_{[y_j = v_0]} = \sum_{1 \leq i \leq |V|} r_i;$$

where $I_{[\cdot]}$ is the indicator function. Constraints (3) and (4) focus on end-to-end traffic, which follow that when the data collection round finishes, each sensor must send out all its data packets and the base station must receive all of them. In order to minimize the visual data

streaming delay of the entire WVSN, an utility function is established as bellow:

$$U(G) = \frac{1}{t_k}$$

The rationale therein is that maximizing $U(G)$ can minimize the number of slots in a frame, which can potentially increase the throughput (i.e., allow better streaming quality) or decrease the latency to send all streaming data to the base station (i.e., minimize the streaming delay). Additionally, maximizing $U(G)$ can improve the reliability as in practice some saved time slots can be added back to the end of the schedule and used to rescue the occasionally lost packets.

Our objectives are thus to (a) minimize $|P|$, (b) minimize the traffic bottleneck, i.e., the load of the wireless communication link with the maximum traffic load among all the wireless communication links $\min\{\max_{e_i \in E} \sum_{(l_{x_j,y_j},t_j) \in Scheduler} I_{[l_{x_j,y_j}=e_i]}\}$, (c) maximize the utility of visual data streaming $\max\{U(G)\}$. In practice, these objectives may conflict with each other and not achieve optimal simultaneously. In order to maximize the benefits, we adopt a linear combination to merge them into one objective function, which in practice can provide more flexibility to users according to their preferences (e.g., users can give 30% of weight to minimize traffic bottleneck and 70% of weight to maximize the utility of visual data steaming). Therefore, we use α and β to represent the weight of $\max_{e_i \in E} \sum_{(l_{x_j,y_j},t_j) \in Scheduler} I_{[l_{x_j,y_j}=e_i]}$ and $U(G)$, respectively, where $\alpha + \beta = 1$. In this combined objective function, our goal is to minimize the combination of total number of visual sensors, traffic bottleneck, and total time-slots, where total time-slots can be denoted with the reciprocal of $U(G)$. Then, the objective function can be presented as

$$\min\{|P| + \alpha \max_{e_i \in E} \sum_{(l_{x_j,y_j},t_j) \in Scheduler} I_{[l_{x_j,y_j}=e_i]} + \beta \frac{1}{U(G)}\}.$$

Since the first priority of this optimization is to minimize the total number of visual sensors,

the number of traffic loads and time slots have been normalized into range 0 to 1. Let m, n represent $\max_{e_i \in E} \sum_{(l_{x_j,y_j},t_j) \in Scheduler} I_{[l_{x_j,y_j}=e_i]}$ and $\frac{1}{U(G)}$, respectively. Then, we have $m_{normalized} = \frac{m-m_{min}}{m_{max}-m_{min}}$, and $n_{normalized} = \frac{n-n_{min}}{n_{max}-n_{min}}$. Therefore, the summation of $\alpha m_{normalized} + \beta n_{normalized}$ should be in range 0 to 1 as well.

4.4 SOLUTION DESIGN

We use a divide and conquer strategy to tackle the formulated cross layer optimization problem. Specifically, we first propose a solution to achieve optimal traffic load balancing in a given network topology, which is discussed in the previous chapter. We then present a collision-free scheduling solution named Busy Link First algorithm which can also obtain the optimal result for a given traffic load. Finally, we use them as building blocks to further solve the original cross layer optimization problem.

4.4.1 Busy Link First Algorithm

For a given topology, we can apply the Edge-Coloring algorithm to group together the communication links together that can carry out transmission simultaneously. Edge-coloring of a graph is an assignment of "colors" to the edges of the graph so that no two incident edges have the same color. For example, Figure 4.1 shows an edge coloring of a graph by the colors red, blue, and green. The edges in the graph can be deemed as wireless communication links, and the vertices can be wireless visual sensors in the WVSN. Therefore, since edge AC, BE and DG have been colored by red, the wireless communications between AC, BE and DG can be performed simultaneously, the same situation for AB and CF. However, the communication between BD has to occupy additional time interval. As such, the links with same color can transmit messages simultaneously without collision, which serves as fundamental base of collision-free traffic scheduling algorithm.

When we deal with scalar data captured by normal wireless sensor, it can be transferred within single time-slot for each group of communication links with same color due

to the smaller size of data, even the accumulated data can be forwarded to the next hop within single time-slot. In such cases, the traditional TDMA solution can be directly applied, and the total number of time-slots used in each round is equal to the number of groups of communication links (i.e., the number of colors).

However, in WVSNs, we collect visual data which is much larger than scalar data in size, and some wireless visual sensors have to serve as relay node to forward visual data for other sensor nodes. Therefore, the captured visual data or accumulated data cannot be transmitted to next hop within single time-slot in each round. In order to collect all the captured visual data to the base station in each round, multiple time-slots have to be applied to some groups of communication links.

Moreover, we note that the time-slots used by each group of communication links may not be equal, where the number of time-slots for each group is decided by the busiest communication link in the group. For instance, without loss of generality, assuming in Figure 4.1 each visual sensor generates 1 unit of visual data in each round, and 1 unit of visual data can be transmitted to next hop within 1 time-slot. Figure 4.2 shows the network topology with traffic load attached beside each communication link to illustrate how much traffic may go through the corresponding link. From Figure 4.2, we know that communication links CF, BE, and DG transmit 1 unit of visual data in each round (i.e., 1 unit/round), BD and AC transmit 2 unit/round, and AB transmits 4 unit/round. According to Figure 4.1 which indicates the same network topology as figure 4.2, we know that the total communication links need to be divided into 3 groups in order to achieve conflict-free transmission based on the edge-coloring solution. Since identical time-slots have to be assigned to each group in traditional TDMA, 12 time-slots are needed to collect all the visual data back to the base station in each round, where 12 time-slots are calculated by 4 unit/round times 3 groups, where 4 unit/round has to be chosen as the number of time-slots for each group of communication links due to the traffic load of the most busy link is 4 unit/round. However, according to Figure 4.1 and 4.2, since we know the maximum traffic load within each group of

communication links as 2 unit/round in red group, 4 unit/round in green, and 2 unit/round in blue group, respectively, then it is easy to calculate how much time-slots will be wasted if the traditional TDMA solution has been applied (i.e., 4 time-slots are wasted in each round as 2 time-slot in red + 2 time-slots in blue = 4 time-slots). However, if we know the exact traffic load for each communication link, the total time-slots used to collect all the visual data back to the base station in each round can be dramatically decreased, because we can customize the time-slots for each group of communication links according to the maximum traffic load in each group. This can be achieved by retrieving the exact traffic load information from the network layer in our cross-layer design approach. As a result, when we apply color assignment solution provided in Figure 4.1, 8 time-slots are needed to collect all the data in each round. The group of communication links with red color occupies 2 time-slots, the group with blue color needs 2 time-slot, and group with green color uses 4 time-slots, respectively.

It is worth nothing that the color assignment solution for each network topology is not unique. For instance, Figure 4.1 and 4.3 are showing two color assignment solutions for the same network topology, and there may exist more color assignment solutions for this network topology. Since multiple color assignments exist for each network topology, one nature question is whether they are the same or some one is better than others in terms of shortened delay when performing data collection. According to color assignment solution presented in Figure 4.3, only 7 time-slots are needed to achieve the same goal which is even better than the solution showing in Figure 4.1 (8 time-slots). Therefore, different color assignment solutions may have diverse performance in terms of shortened delay. And our goal is thus to explore one of the most efficient color assignment solutions among all the combinations of communication links. To this end, we develop Busy Link First Algorithm to optimally address this problem. The Busy Link First algorithm sorts all the communication links in descending order according to their traffic load. It then picks the most busy and available communication link to assign the color first, which can generate the optimal solution

in terms of minimizing the total number of time-slots as shown by the below theorem.

Theorem 3. *On any given network topology, Busy Link First algorithm can return the optimal solution to minimize the total number of required time-slots.*

Proof. Suppose there are totally p groups of communication links generated from the edge-coloring solution. Let L_i represents the communication link with the maximum traffic load in group i, where $1 \leq i \leq p$. Then the final solution getting from Busy Link First algorithm is $S^* = \{L_p^*, L_{p-1}^*, ..., L_i^*, ..., L_1^*\}$, where the items in S^* are in descending order. Assuming that S' exists which is better than S^* in terms of total time-slots used. Therefore, within S', $\exists$ L'_m and $1 \leq m \leq p$, where $L'_m < L_i^*$, which means that L_i^* can be replaced by L'_m the communication link with smaller traffic load. According to Busy Link First Algorithm which is greedy algorithm, L_i^* has been chosen in group i due to L_i^* is conflict with members in group n (i.e., the links with larger traffic load and directly connecting with L_i^* are already added into group n), where $i < n \leq p$. If L_i^* can be replaced by L'_m, L_i^* has to be added into group n, it makes L_i^* conflicted with neighbor link who is already added into that group, which leads the contradiction. Therefore, the solution generated by our algorithm is optimal. □

Algorithm 5 summarizes our Busy Link First Algorithm, which takes a set of communication links with traffic load on each link. Then, according to the traffic load of each communication link, the communication link set is sorted in descending order (line 1). Within the while loop, the busiest and available edge is chosen and assigned a free color (line 4-10). For each set of communication links with the same color, the algorithm finds out the busiest link within each color and put the link into a set B (line 12-16). Thereafter, the number of time-slots for each set of communication links (i.e., the communication links have been assigned with same color) is assigned according to the traffic load of the busiest link within the corresponding set. Finally, the optimal time-slots assignment solution S is returned.

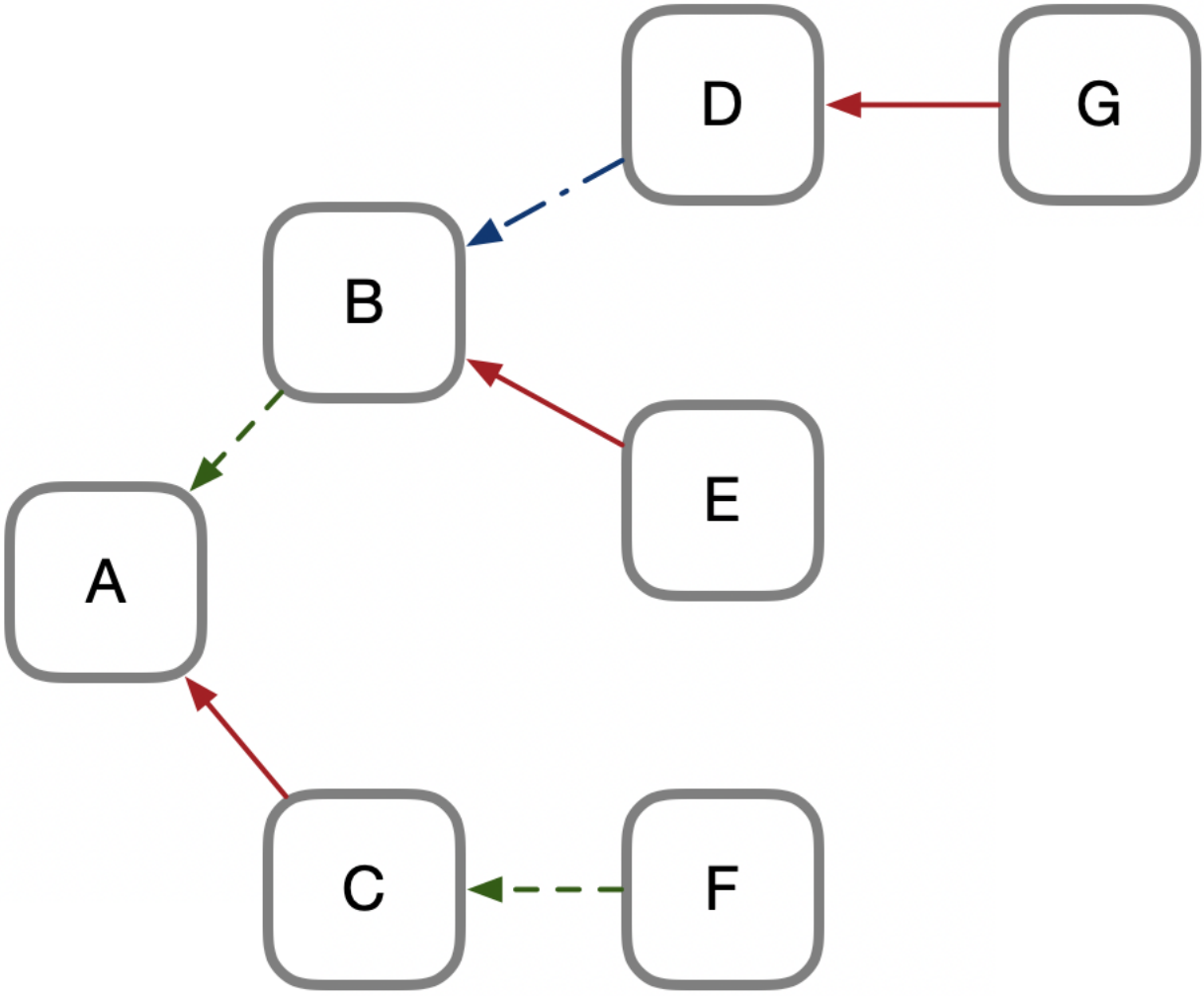

Figure 4.1. Example of Edge Coloring

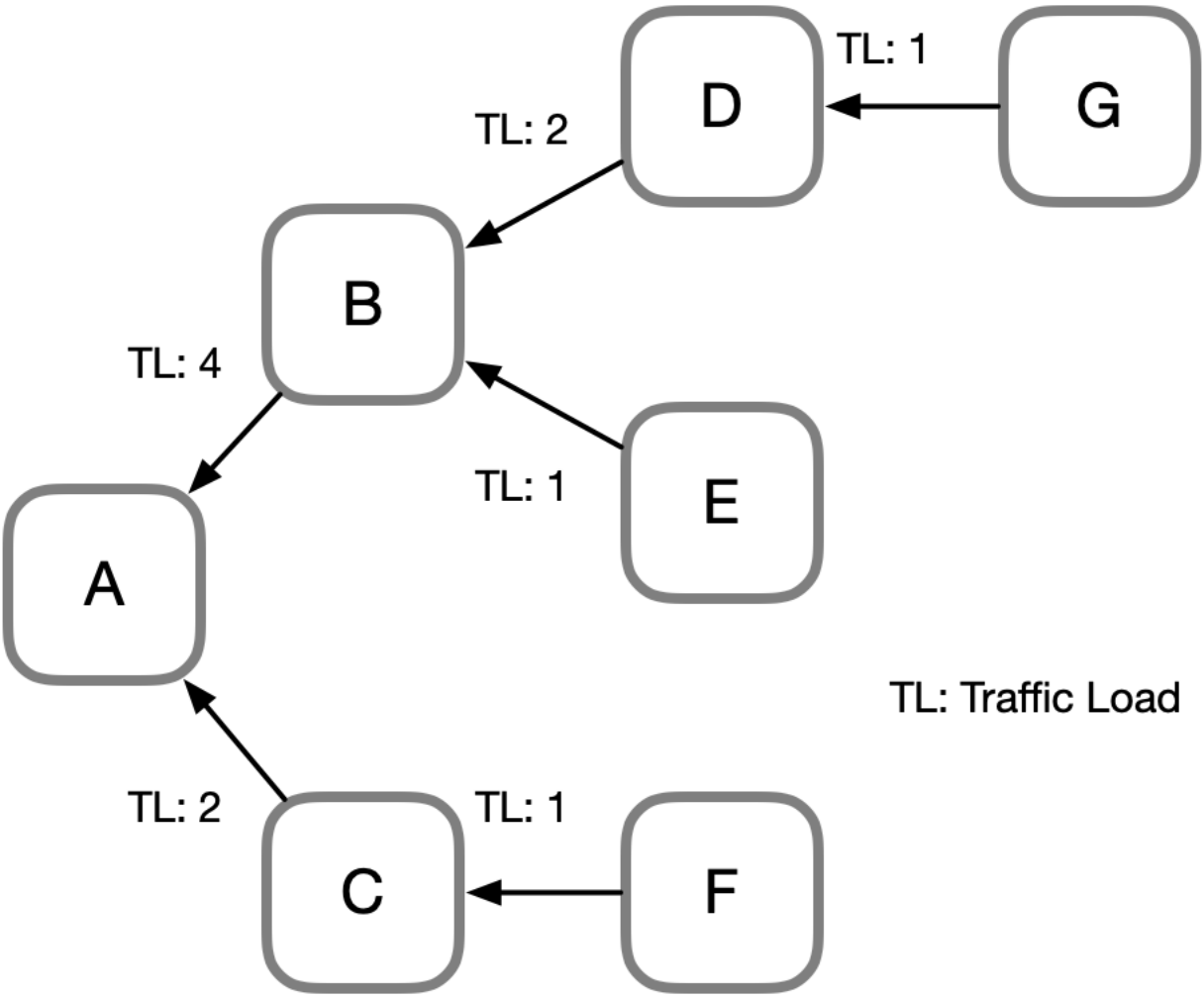

Figure 4.2. Network Topology with Traffic Load

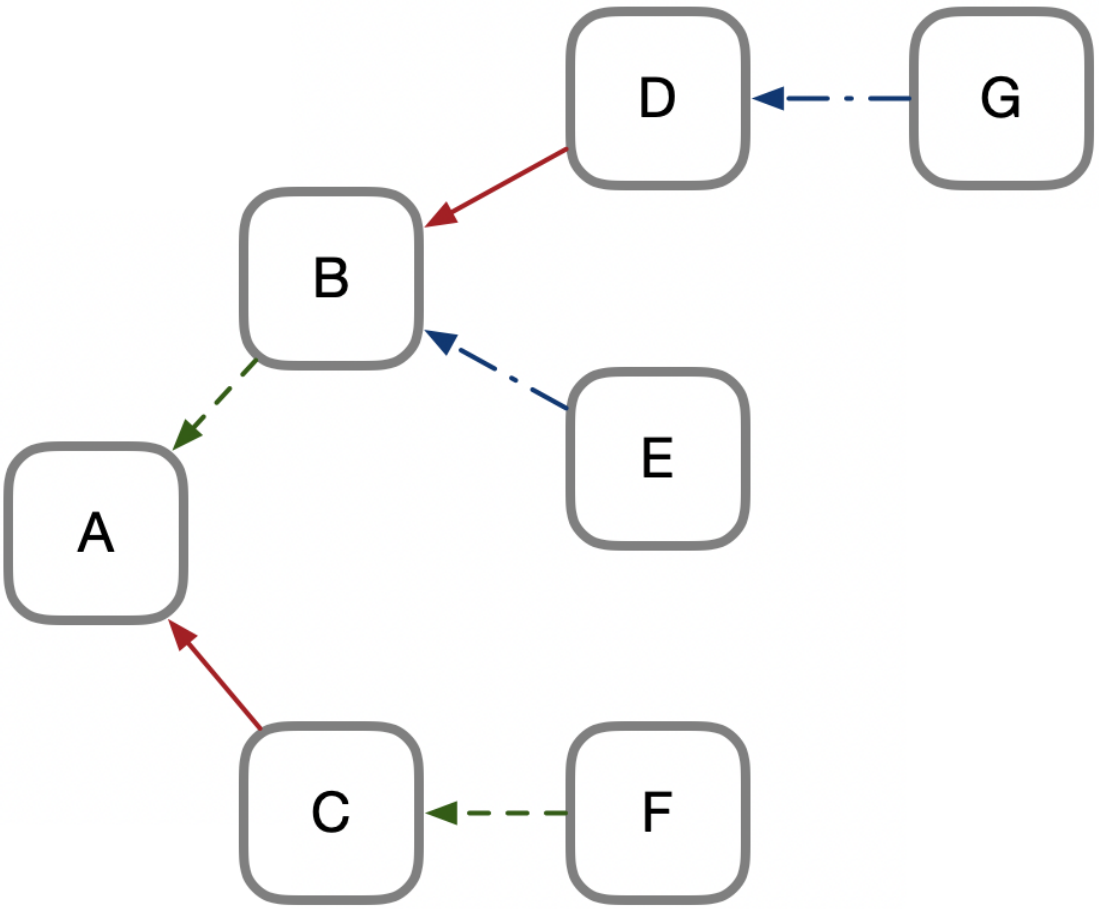

Figure 4.3. Example of Different Edge Coloring

Algorithm 5 Busy Link First Algorithm

Input: A graph **G**. A set of communication links with traffic load as edge set **E**(**G**)
Output: Time-slots assignment solution S

```
1: Sort E(G) according to traffic load in descending order.
2: Let U := E(G)
3: while U ≠ ∅ do
4:    Let (u, v) be most busy edge in U.
5:    Let F[1 : k] be a maximal fan of u starting at F[1] = v.
6:    Let c be a color that is free on u and d be a color that is free on F[K].
7:    Invert the cd_u path.
8:    Let w ∈ V(G) be such that w ∈ F, F' = [F[1]...w] is a fan and d is free on w.
9:    Rotate F' and set c(u, w) = d.
10:   U := U − {(u, v)}
11: end while
12: for color c in all assigned colors do
13:    for edge e in E(G) do
14:       if color of e is c and traffic load is larger than S(c) then
15:          update S(c)
16:       end if
17:    end for
18: end for
19: return S
```

4.4.2 SchEduling-aware Enhanced Depth-first search Algorithm

With the Busy Link First algorithm as building block, we propose a visual sensor deployment solution named SchEduling-aware Enhanced Depth-first search algorithm (SEED) which can obtain optimal solution if given enough time. Dislike Traditional DFS algorithms that exhaustively evaluate each search branch, SEED algorithm uses branch pruning to cut the candidate location searching branches that are guaranteed not able to improve the currently found best solution, so that the explored solution space can be significantly reduced. Algorithm 6 summarizes SEED algorithm, where *depth* denotes the number of used visual sensors to cover the space in the current search branch, and *best* is the currently found best solution. Our first enhancement is to determine its feasibility by comparing the current search branch to our *best* found solution (line 2). We next make a decision that whether we need to update *best* or trim the current branch. If the current solution is better than the *best*, the final solution set $S_{min_{result}}$ is reset to be empty, and the Busy Link First algo-

rithm is applied to calculate the minimum number of time slots under the current network topology. Meanwhile, the currently *best* found solution is added into $S_{min_{result}}$ (line 6-11). To further improve the efficiency of SEED, out second enhancement selects the location in each iteration by sorting all of remaining $L \in \mathbf{L}$ in descending order based on the maximum number of fresh vertices that a visual sensor placed at the corresponding location can cover and the produced facing direction is handled in a similar way (line 13-21). This can further reduce the chances to select infeasible solution options so as to quickly find better quality candidate location options. We also ensure in the algorithm, that all the candidate locations in the list have a communication path to the base station (line 24). The next enhancement is defined by the following constraint: $\frac{F}{F_c^1} + depth < best$ where F denotes the remaining fresh vertices ready to be covered, F_c^1 represents the number of covered vertices by the first visual sensor of current solution, (where we choose F_c^1 as denominator due to that since all the candidate location has been sorted before choosing by descending order, the number of fresh vertices covered by the first visual sensor is the maximum compared with others, which can guarantee the best solution will not be pruned). (line 25-30) The function of this constraint is to bound the currently estimated branch that the number of candidate locations to cover the remaining fresh vertices plus the *depth* must be smaller than the currently found *best* solution, otherwise the branch will be pruned.

4.5 PERFORMANCE EVALUATION

An extensive simulations are conducted to evaluate our solutions using a customized simulator implemented by JavaScript, which can emulate the 3D deployment of wireless visual sensor nodes in a virtual environment. Table I outlines the default parameters settings that are utilized in the performance evaluation.

For comparison, we implement a baseline algorithm that randomly selects locations in $\mathbf{L}$ to deploy video sensors. Once a location is selected, the facing direction of the visual sensor is adjusted to greedily cover the maximum number of fresh grid points (i.e., those

Algorithm 6 SchEduling-aware Enhanced Depth-first search Algorithm

Input: **L**, **A**, D
Output: Set of minimized deployment solutions $S_{min_{result}}$
Initialize: List of **A**, D, **L** and *best*

```
1: DFS(depth)
2: if depth ≥ best then
3:   return
4: end if
5: if A == 0 then
6:   if depth < best then
7:     clear S_min_result
8:   end if
9:   update best
10:  min_result.timeSlots ← BusyLinkFirst
11:  add min_result → S_min_result
12: else
13:  for all C_L in L do
14:    compute all face ∈ D for C_L
15:    select face with max F_c
16:  end for
17:  sort tempList by descending order according F_c
18:  for all rank (i.e., all C_L with the same max F_c) in tempList do
19:    resort each rank by ascending order
20:  end for
21:  store tempList → Queue
22:  while Queue ≠ ∅ do
23:    C_L ← Dequeue
24:    if C_L within R_max of any deployed visual sensor then
25:      if lowerBound(F) + depth ≥ best then
26:        break
27:      end if
28:      record C_L* and F_c* then remove them
29:      DFS(depth+1)
30:      add F_c* back to A
31:    end if
32:  end while
33:  add all removed C_L* back
34: end if
35: return S_min_result
```

Testing Parameters	
Parameter	Default Value
3D indoor space	$Length \times 60 \times 100$
A	Same as 3D indoor space
L	Top half of walls and ceiling
g_A	25
g_L	25
g_D	15 degrees
FOV	50 degrees
Aspect Ratio	1.7778
Near Field	1
Far Field	100

Table 4.1. Simulation parameter settings

points that have not been covered by previously deployed visual sensors). The visual sensor will be removed if it cannot cover any fresh points, and another random location will be selected. This process will continue until all the grid points in **A** have been covered. We run the baseline algorithm 100 times to show the average results. In this random algorithm, the connectivity has not been considered. Otherwise, the result may not be able to returned, especially for longer corridor cases. The evaluation results indicate that our solution SEED can outperform the random algorithm by 64%.

In this paper, the original TDMA method has been implemented as baseline that equal number of time-slots have been assigned to each group of communication links. And the time-slot number is equivalent to the traffic load of the most busy communication link within the entire network. Figure 4.4 shows the number of time-slots utilized by different approaches with various 3D indoor space. As expected, more time-slots are required to transmit all the raw data back to base station, as long as the size of 3D indoor space increasing. Moreover, our customized TDMA without_sorting algorithm produces a 25% reduction on the number of time-slots compared with the baseline algorithm. The Busy Link First algorithm can further outperform the customized TDMA without_sorting algorithm by 29%.

It is worth nothing that we run the simulation on a standard PC with configurations listed as: i7 processor, 16GB RAM, 1TB SSD and set a short time limitation (1 *hour*) for

Corridor Length	Random Algorithm (Average)	SEED
100	8	3
200	14	5
300	20	8
400	26	11
500	32	13
600	38	17
700	45	19
800	50	23

Table 4.2. Total number of visual sensors used to cover the entire 3D indoor space

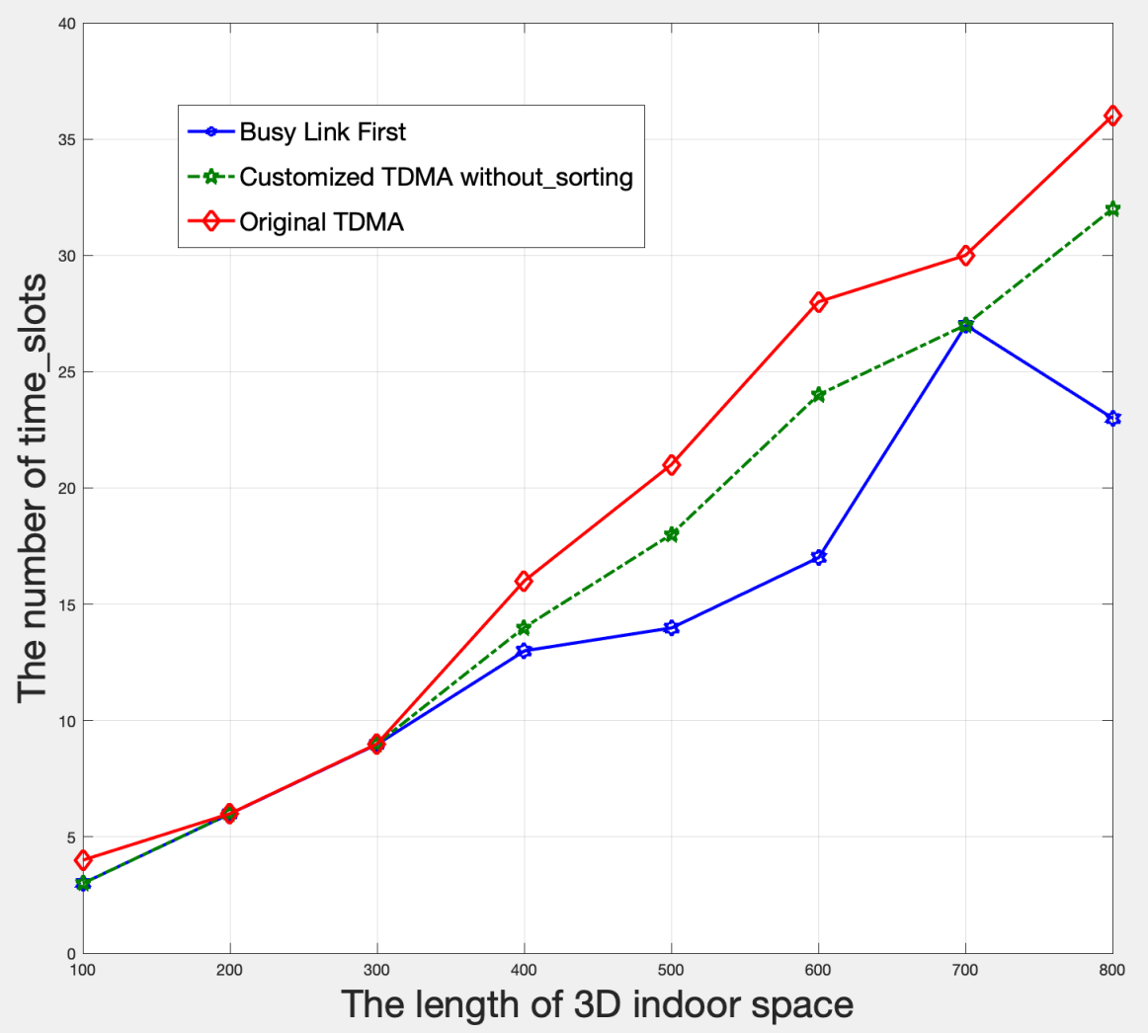

Figure 4.4. Comparison among three TDMA solutions

SEED solution in our evaluation, it can still successfully return the optimal results for the case with the length of 3D indoor space equal to 100 and 200. We thus conjecture that our Enhanced-DFS solution can return optimal results for more cases if it runs on a local or cloud server that has much more computation power and more time is allowed. Since the set with minimum number of visual sensors to fully cover the monitored area is not unique, the SEED has ability to dig out multiple sets for each corridor length. Figure 4.5 demonstrates the upper bound and lower bound of each corridor length in the aspect of total time-slots. The results indicate that the difference between upper bound and lower bound can be up to 46%. Similarly, Figure 4.6 shown that the gap between upper bound and lower bound can be reach to 62% in terms of traffic load balancing.

As we mentioned in the previous section, the objectives of minimizing the traffic load and minimizing the total time-slots may not be optimized simultaneously. Therefore, the linear combination has been applied to maximize the total benefit from these two objectives. Surprisingly, the evaluation results indicate that these two objectives can be optimized at the same time in our testing cases, when we set the α and β combinations as $\alpha = x\{x|0.1, 0.2, ..., 0.9\}$, $\beta = y\{y|0.9, 0.8, ..., 0.1\}$ in Table III. According to the experiment results in Table III, we conclude that our solution is insensitive to the linear combination weight. Furthermore, the results of minimized tradeoff combinations have been demonstrated in Figure 4.7. In Figure 4.7, we demonstrate that the upper bound and lower bound of total time-slots within various corridor length when the traffic load as minimum has been fixed. Figure 4.8 shown that the objective of minimizing the total time-slots may not be achieved automatically while we optimized the objective of minimizing the traffic load, which further indicates the effectiveness of our solution. The similar scenario for objective of minimizing the traffic load while fixed the total time-slots as minimum is demonstrated in Figure 4.9.

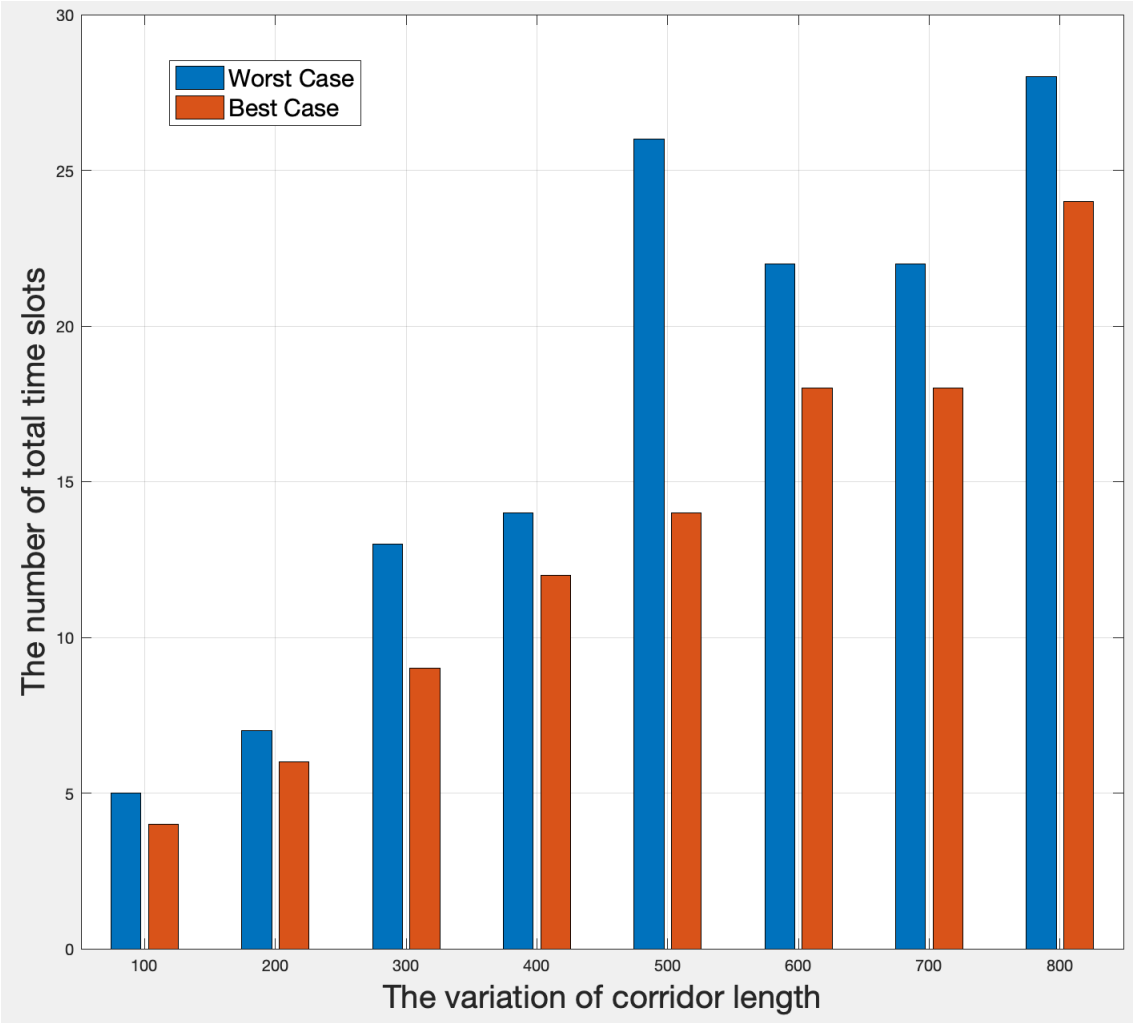

Figure 4.5. Total time-slots boundary within various corridor length

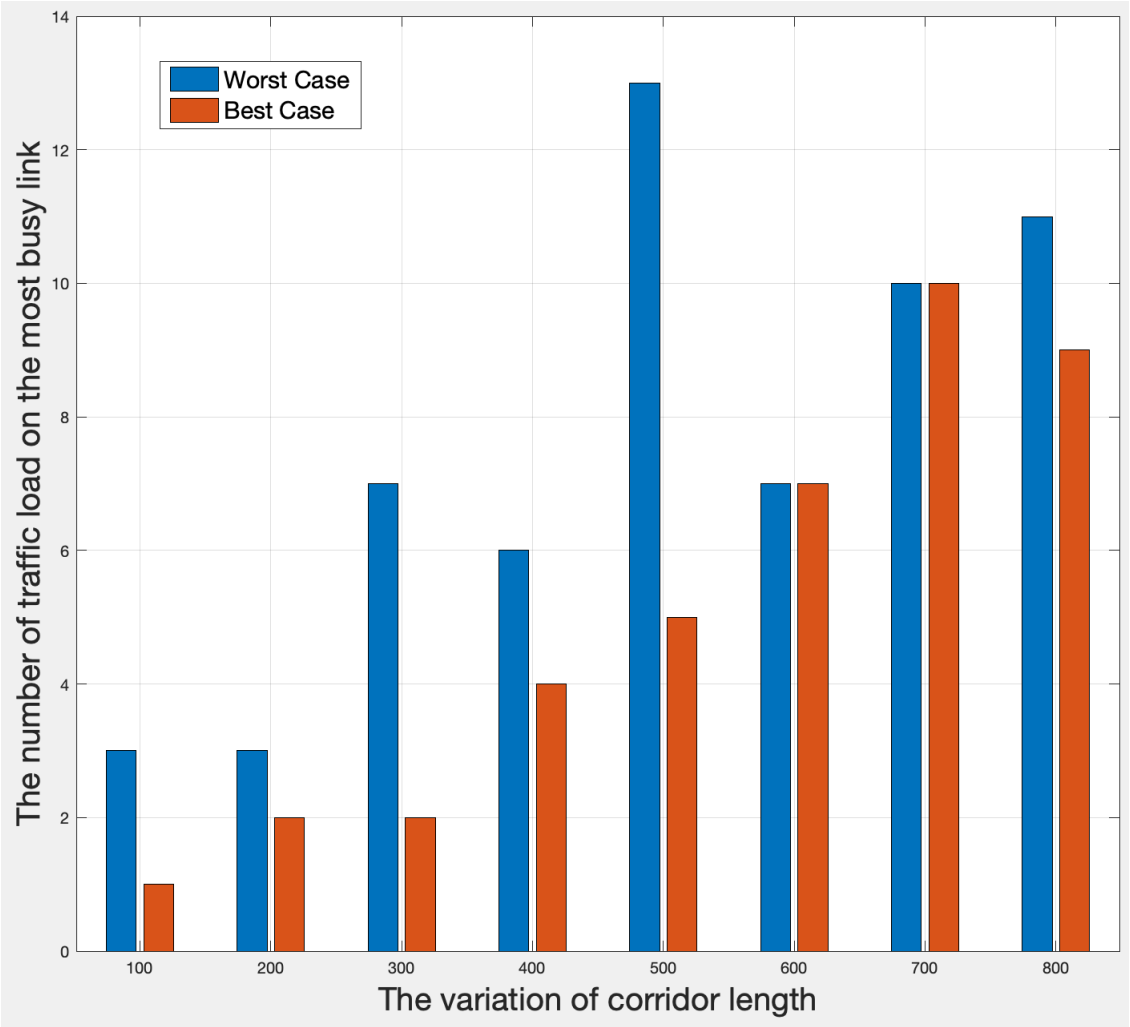

Figure 4.6. Traffic load boundary within various corridor length

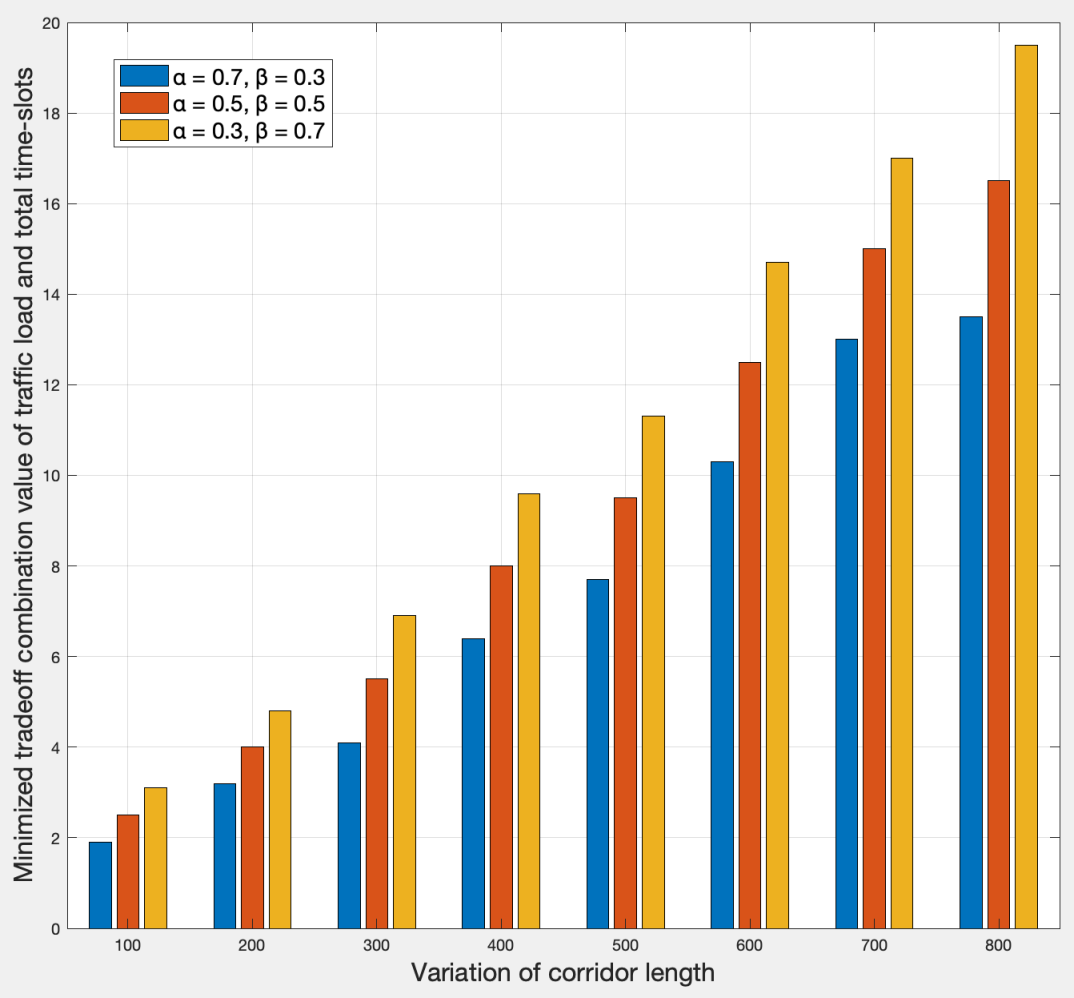

Figure 4.7. Minimized tradeoff combination value of traffic load and total time-slots. The weight of α is 0.7, 0.5, or 0.3, and the weight of $\beta = 1$ - α.

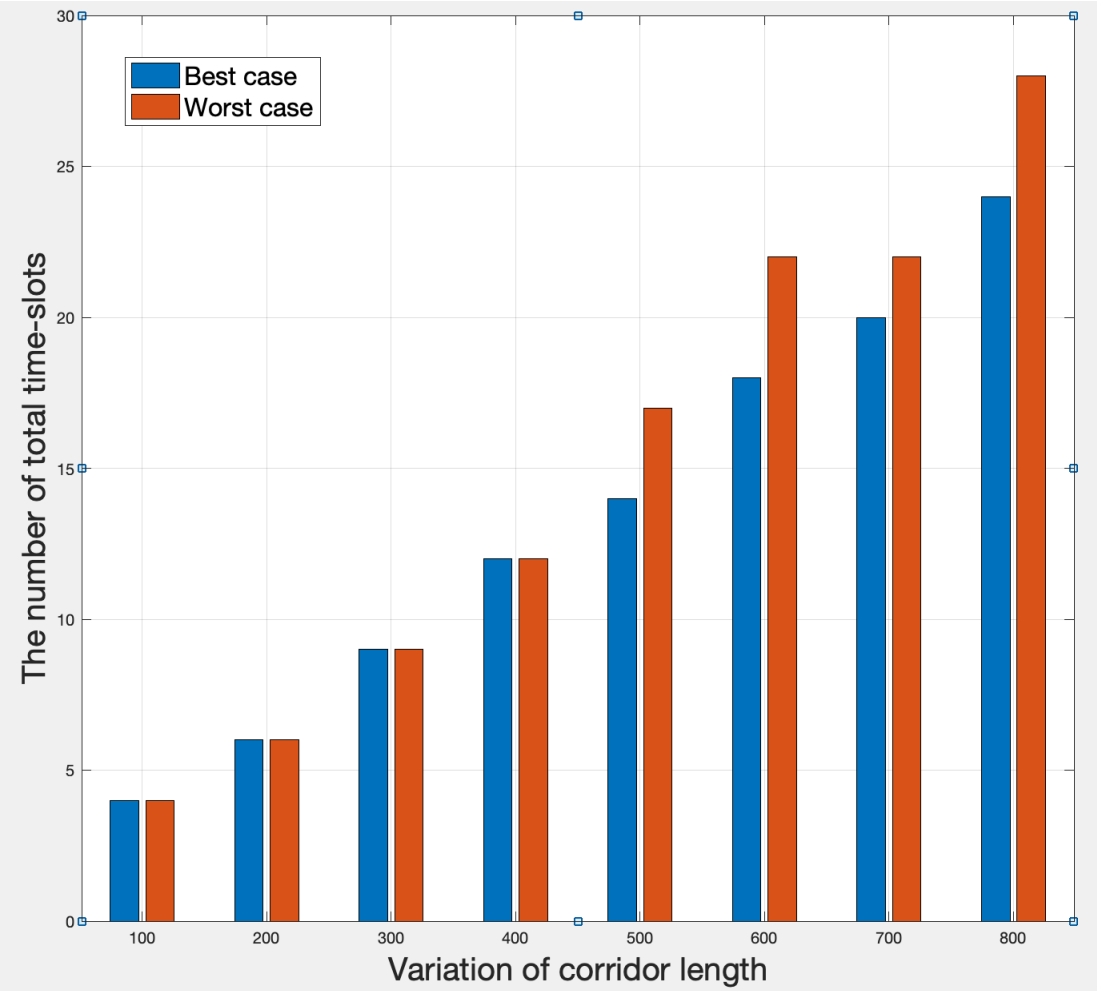

Figure 4.8. Total time-slots boundary within various corridor length while fixed the traffic load as minimum

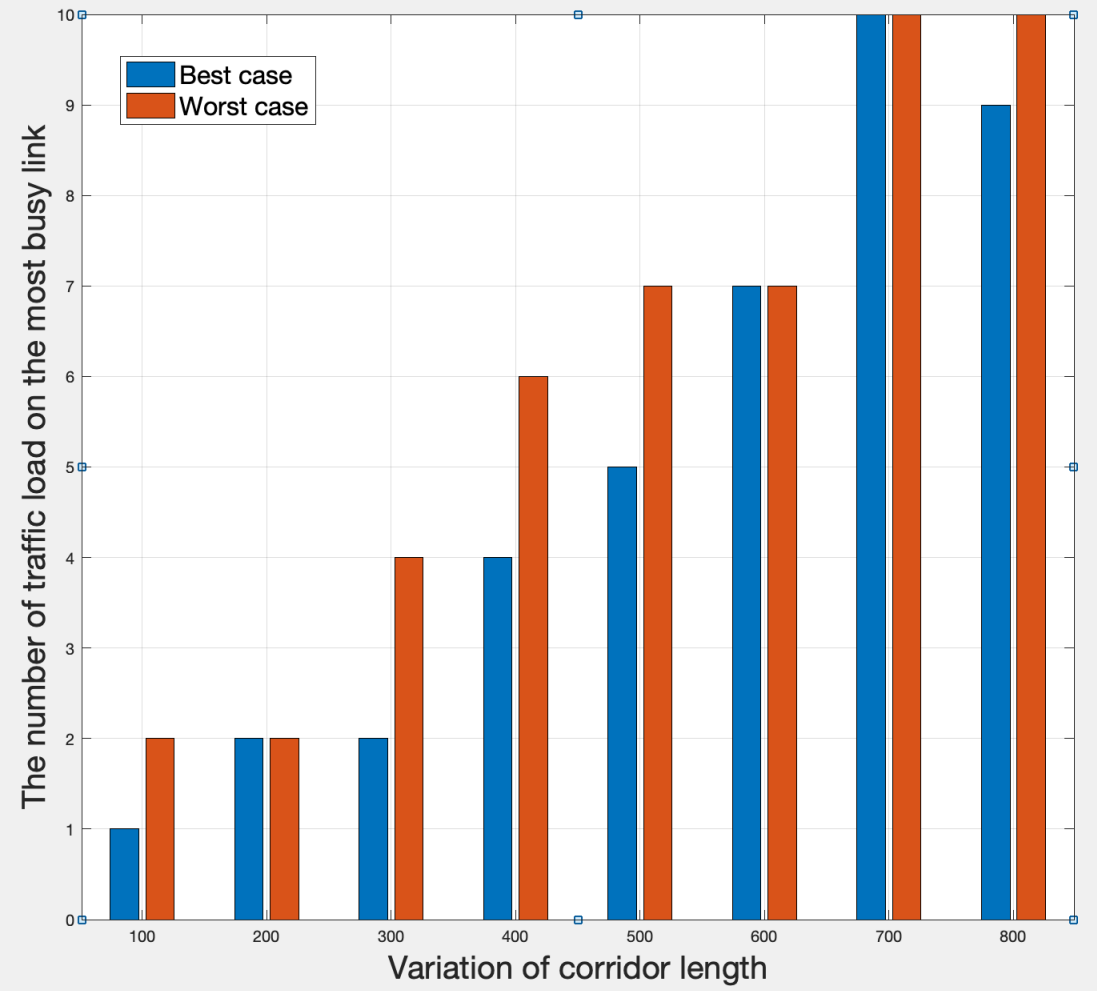

Figure 4.9. Traffic load boundary within various corridor length while fixed the total time-slots as minimum

	100	200	300	400	500	600	700	800
Traffic Bottleneck (Chosen)	1	2	2	4	5	7	10	9
Min Traffic Bottleneck	1	2	2	4	5	7	10	9
Total Time-slots (Chosen)	4	6	9	12	14	18	20	24
Min Total Time-slots	4	6	9	12	14	18	20	24
$\alpha = 0.1$ $\beta = 0.9$	3.7	5.6	8.3	11.2	13.1	16.9	19	22.5
$\alpha = 0.2$ $\beta = 0.8$	3.4	5.2	7.6	10.4	12.2	15.8	18	21
$\alpha = 0.3$ $\beta = 0.7$	3.1	4.8	6.9	9.6	11.3	14.7	17	19.5
$\alpha = 0.4$ $\beta = 0.6$	2.8	4.4	6.2	8.8	10.4	13.6	16	18
$\alpha = 0.5$ $\beta = 0.5$	2.5	4	5.5	8	9.5	12.5	15	16.5
$\alpha = 0.6$ $\beta = 0.4$	2.2	3.6	4.8	7.2	8.6	11.4	14	15
$\alpha = 0.7$ $\beta = 0.3$	1.9	3.2	4.1	6.4	7.7	10.3	13	13.5
$\alpha = 0.8$ $\beta = 0.2$	1.6	2.8	3.4	5.6	6.8	9.2	12	12
$\alpha = 0.9$ $\beta = 0.1$	1.3	2.4	2.7	4.8	5.9	8.1	11	10.5

Table 4.3. Trade-off Between Traffic Load and Total Time-slots

CHAPTER 5

CONCLUSION AND FUTURE WORKS

In this dissertation, we propose a cross-layer design approach to address deployment, traffic balancing, and link layer scheduling problems on application layer, network layer and link layer, respectively. Firstly, we studied the 2-angular-coverage problem on the Application Layer for WVSNs and developed a scheme to effectively monitor a 3D indoor space from different perspectives using visual sensor networks. For each perspective, we ensured that all the points inside the interest areas were covered. In each candidate location for a visual sensor, we discretize the area to effectively deploy 2-angular-coverage visual sensors. By iteratively selecting the facing direction choices for each visual sensor, our enhanced-DFS algorithm improved the coverage of the space and provided feasible solutions for networks with small values of M.

Secondly on the Network Layer, the general deployment problem of optimizing the placement and angular direction of each visual sensor was first formulated in a continuous space, and then converted into a discrete version where 3D grid lattices are applied so as to be able to achieve arbitrary approximation accuracy by adjusting their granularities. We also noted that tree topology is not sufficient to minimize the traffic load bottleneck and developed an optimal load balance algorithm by transforming the problem into a generalized max-flow problem and solving it efficiently. With this as building block, we further proposed a Greedy Heuristic and an Enhanced-DFS solutions to address the traffic-aware WVSN deployment problem, where if given enough time, the latter can return the optimal results. Our extensive simulations demonstrated that our Greedy algorithm produces a 62.5% reduction on the number of used visual sensors compared with the baseline algorithm. The Enhanced-DFS

can further outperform the Greedy algorithm by up to 17%. Additionally, our traffic-aware solution can further reduce the bottleneck of traffic load by up to 20% compared to the tree topology based approach.

Finally, we studied the scheduling aware WVSN deployment problem for 3D indoor monitoring on the Link Layer. We applied customized TDMA MAC protocol named Busy Link First which assisted by Directed Edge Coloring algorithm to tackle the scheduling problem in order to achieve collision-free wireless visual sensor network. Our extensive simulation demonstrated that our customized TDMA solution Busy Link First algorithm can further reduce the number of total time slots by 39% compared with the original TDMA solution. With this as a building block, we proposed a SchEduling-aware Enhanced Depth-first search algorithm to address WVSN deployment problem in 3D indoor area, where if given enough time, the optimal solutions can be returned. Beyond these goals, the evaluation results indicate that the objectives minimized traffic load and minimized total time-slots can be optimized simultaneously from our solution in the testing cases.

We plan to further fine-tune our solution based on the preliminary performance evaluation. For example, we are going to implement coverage rechecking step to eliminate the redundancy of deployed visual sensors from previous result on both algorithms. When we deploy new sensors into the interesting field, the area covered by previously deployed visual sensors may be covered again by newly mounted visual sensors. If that is the case, the previously deployed visual sensors can be removed in order to eliminate the redundancy (i.e., save the budget).

In the second research stage, the sequential search has been used in the Optimal Traffic Load Algorithm to find the minimized maximum traffic load by sequentially increasing the link capacity of Non-Virtual-Link. In small size of 3D space, our solution is efficient due to the size of traffic bottleneck is much smaller compare to the total number of wireless virtual sensors used in the solution. However, when the interested area is quite larger, more sensors are required. The traffic load may be reached to tens or even hundreds. Under this situation,

if more efficient searching algorithm is applied to replace the sequential search such as binary search, which can make our Optimal Traffic Load Algorithm even more efficient. In our next research step, the aforementioned strategy will be applied and tested to further improve our solution.

BIBLIOGRAPHY

BIBLIOGRAPHY

Akshay, N., M. P. Kumar, B. Harish, and S. Dhanorkar (2010), An efficient approach for sensor deployments in wireless sensor network, in *Emerging Trends in Robotics and Communication Technologies (INTERACT), 2010 International Conference on*, pp. 350–355, IEEE.

Akyildiz, I. F., W. Su, Y. Sankarasubramaniam, and E. Cayirci (2002), Wireless sensor networks: a survey, *Computer networks*, 38(4), 393–422.

Ammari, H. M. (2011), On the problem of k-coverage in 3d wireless sensor networks: A reuleaux tetrahedron-based approach, in *2011 Seventh International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pp. 389–394, IEEE.

Brown, T., Z. Wang, T. Shan, F. Wang, and J. Xue (2017), Obstacle-aware wireless video sensor network deployment for 3d indoor monitoring, in *GLOBECOM*, IEEE.

Chen, P.-Y. (), A survey of solutions to the coverage problems in wireless sensor networks.

Chow, K.-Y., K.-S. Lui, and E. Y. Lam (2007), Maximizing angle coverage in visual sensor networks, in *2007 IEEE International Conference on Communications*, pp. 3516–3521, IEEE.

Deif, D. S., and Y. Gadallah (2013), Classification of wireless sensor networks deployment techniques, *IEEE Communications Surveys & Tutorials*, 16(2), 834–855.

Fasolo, E., M. Rossi, J. Widmer, and M. Zorzi (2007), In-network aggregation techniques for wireless sensor networks: a survey, *IEEE Wireless Communications*, 14(2), 70–87.

Gandham, S., M. Dawande, and R. Prakash (2008), Link scheduling in wireless sensor networks: Distributed edge-coloring revisited, *Journal of Parallel and Distributed Computing*, 68(8), 1122–1134.

Han, K., L. Xiang, J. Luo, and Y. Liu (2012), Minimum-energy connected coverage in wireless sensor networks with omni-directional and directional features, in *Proceedings of the thirteenth ACM international symposium on Mobile Ad Hoc Networking and Computing*, pp. 85–94.

Hefeeda, M., and M. Bagheri (2007), Randomized k-coverage algorithms for dense sensor networks, in *IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications*, pp. 2376–2380, IEEE.

Horster, E., and R. Lienhart (2006), Approximating optimal visual sensor placement, in *2006 IEEE International Conference on Multimedia and Expo*, pp. 1257–1260, IEEE.

Huang, C.-F., and Y.-C. Tseng (2005), A survey of solutions to the coverage problems in wireless sensor networks., *Journal of Internet Technology*, 6(1), 1–8.

Kacimi, R., R. Dhaou, and A.-L. Beylot (2013), Load balancing techniques for lifetime maximizing in wireless sensor networks, *Ad hoc networks*, 11(8), 2172–2186.

Katsuma, R., Y. Murata, N. Shibata, K. Yasumoto, and M. Ito (2009), Extending k-coverage lifetime of wireless sensor networks using mobile sensor nodes, in *2009 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, pp. 48–54, IEEE.

Kouakou, M. T., K. Yasumoto, S. Yamamoto, and M. Ito (2012), Cost-efficient sensor deployment in indoor space with obstacles, in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium on a*, pp. 1–9, IEEE.

Krishnamachari, L., D. Estrin, and S. Wicker (2002), The impact of data aggregation in wireless sensor networks, in *Proceedings 22nd international conference on distributed computing systems workshops*, pp. 575–578, IEEE.

Li, X.-Y., P.-J. Wan, and O. Frieder (2003), Coverage in wireless ad hoc sensor networks, *Computers, IEEE Transactions on*, 52(6), 753–763.

Liang, C.-K., C.-H. Tsai, and M.-C. He (2011), On area coverage problems in directional sensor networks, in *Information Networking (ICOIN), 2011 International Conference on*, pp. 182–187, IEEE.

Lipare, A., D. R. Edla, and V. Kuppili (2019), Energy efficient load balancing approach for avoiding energy hole problem in wsn using grey wolf optimizer with novel fitness function, *Applied Soft Computing*, 84, 105,706.

Liu, L., H. Ma, and X. Zhang (2008), On directional k-coverage analysis of randomly deployed camera sensor networks, in *2008 IEEE International Conference on Communications*, pp. 2707–2711, IEEE.

Malek, S. M. B., M. M. Sadik, and A. Rahman (2016), On balanced k-coverage in visual sensor networks, *Journal of Network and Computer Applications*, 72, 72–86.

Mao, J., Z. Wu, and X. Wu (2007), A tdma scheduling scheme for many-to-one communications in wireless sensor networks, *Computer Communications*, 30(4), 863–872.

Megerian, S., F. Koushanfar, M. Potkonjak, and M. B. Srivastava (2005), Worst and best-case coverage in sensor networks, *Mobile Computing, IEEE Transactions on*, 4(1), 84–92.

Morsly, Y., N. Aouf, M. S. Djouadi, and M. Richardson (2011), Particle swarm optimization inspired probability algorithm for optimal camera network placement, *IEEE Sensors Journal*, 12(5), 1402–1412.

Munishwar, V. P., and N. B. Abu-Ghazaleh (2013), Coverage algorithms for visual sensor networks, *ACM Transactions on Sensor Networks (TOSN)*, 9(4), 45.

Newell, A., K. Akkaya, and E. Yildiz (2010), Providing multi-perspective event coverage in wireless multimedia sensor networks, in *IEEE Local Computer Network Conference*, pp. 464–471, IEEE.

Pantazis, N. A., D. J. Vergados, D. D. Vergados, and C. Douligeris (2009), Energy efficiency in wireless sensor networks using sleep mode tdma scheduling, *Ad Hoc Networks*, 7(2), 322–343.

Peng, J., J. Jingqi, W. Chengdong, and L. Hongchao (2013), A coverage-enhance scheduling algorithm for 3d directional sensor networks, in *Control and Decision Conference (CCDC), 2013 25th Chinese*, pp. 2888–2892, IEEE.

Raha, A., S. Maity, M. K. Naskar, O. Alfandi, and D. Hogrefe (2012), An optimal sensor deployment scheme to ensure multi level coverage and connectivity in wireless sensor networks, in *Wireless Communications and Mobile Computing Conference (IWCMC), 2012 8th International*, pp. 299–304, IEEE.

Sheramin, G., N. Ebrahimian, A. H. Navin, and M. K. Mirnia (2010), Connectivity issue in wireless sensor networks by using depth-first search and genetic algorithm, in *Computational Intelligence and Communication Networks (CICN), 2010 International Conference on*, pp. 377–381, IEEE.

Soro, S., and W. Heinzelman (2009), A survey of visual sensor networks, *Advances in multimedia*, 2009.

Soro, S., and W. B. Heinzelman (2005), On the coverage problem in video-based wireless sensor networks, in *2nd International Conference on Broadband Networks, 2005.*, pp. 932–939, Citeseer.

Tavli, B., K. Bicakci, R. Zilan, and J. M. Barcelo-Ordinas (2012), A survey of visual sensor network platforms, *Multimedia Tools and Applications*, 60(3), 689–726.

TYen, H.-H. (2014), Novel visual sensor deployment algorithm in ptz wireless visual sensor networks, in *Wireless and Mobile, 2014 IEEE Asia Pacific Conference on*, pp. 214–218, IEEE.

Wang, Z., F. Wang, T. Brown, J. Xue, and J. Zhang (2019), Traffic aware wireless visual sensor network deployment for 3d indoor monitoring, in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE.

Wu, M.-C., and W.-F. Lu (2013), On target coverage problem of angle rotatable directional sensor networks, in *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2013 Seventh International Conference on*, pp. 605–610, IEEE.

Xu, S., W. Lyu, and H. Li (2015), Optimizing coverage of 3d wireless multimedia sensor networks by means of deploying redundant sensors, *arXiv preprint arXiv:1509.04823*.

Yadav, R., S. Varma, N. Malaviya, et al. (2009), A survey of mac protocols for wireless sensor networks, *UbiCC journal*, 4(3), 827–833.

Yap, F. G., and H.-H. Yen (2014), A survey on sensor coverage and visual data capturing/processing/transmission in wireless visual sensor networks, *Sensors*, 14(2), 3506–3527.

Yen, H.-H. (2013), Efficient visual sensor coverage algorithm in wireless visual sensor networks, in *Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International*, pp. 1516–1521, IEEE.

Yildiz, E., K. Akkaya, E. Sisikoglu, and M. Y. Sir (2014), Optimal camera placement for providing angular coverage in wireless video sensor networks, *IEEE transactions on computers*, 63(7), 1812–1825.

Zhang, Y., S. Zheng, and S. Xiong (2009), A scheduling algorithm for tdma-based mac protocol in wireless sensor networks, in *2009 First International Workshop on Education Technology and Computer Science*, vol. 3, pp. 148–151, IEEE.

VITA

Zhonghui Wang

Zhonghui Wang is a Ph.D. candidate in Computer and Information Science Department of University of Mississippi. He got the Master Degree in University of Mississippi, MS, USA.