Efficient Search Algorithms using Autonomous Mobile Sensor Nodes

Seokhoon Yoon and Chunming Qiao
Department of Computer Science and Engineering
State University of New York at Buffalo

Abstract

In this paper, we propose search algorithms using multiple autonomous and cooperative mobile sensor nodes (MSN). Our goal is to minimize the total search time and the travel distance of MSNs in a given search area while enabling cooperation among the MSNs and tolerating possible failures of one or more MSNs. We first describe a Lane Based Search (LBS) strategy in which each mobile sensor node is in charge of a sub-area called a lane. We then propose and analyze three distributed synchronization schemes, namely *Alternating Column Synchronization (ACS)*, *Strict Line Synchronization (SLS)*, and *X Line Synchronization (XS)* which are essential for co-operation and fault tolerance to MSN failures. Simulations were performed to evaluate their performances in terms of the total search time and the average travel distance. We show through simulation results and numerical analysis that X synchronization with appropriate parameters outperforms alternating column synchronization and strict line synchronization.

I. Introduction

One of the important civilian or military tasks is to survey a large area in order to search for targets or simply collect certain data (possibly image) utilizing a fleet of a small number of mobile sensor nodes (or simply MSN). The need for using multiple MSNs comes from the fact that the area is often much larger than the sensing range (and even the data communication range) of any single mobile node. In fact, even with multiple MSNs, the applications we consider require that the MSNs to move from their initially deployed locations to other locations within the area so the entire area can be covered. The MSNs may need to periodically communicate with their control and command center (CCC) or base station by sending CCC collected data periodically and receiving control information. The objectives may include minimizing the total mission completion time, traveling distance, and energy consumption by either all or individual nodes.

The proposed architecture uses a set of MSNs that coordinate and cooperate with each other under autonomous and distributed control. A useful feature is its resilience against a failure of an MSN. Another useful feature is that it supports heterogeneous MSNs with different sensing, communication and processing capabilities, which are often deployed in a mission, due to practical constraints that limit the energy, processing power and size, as well as cost of each MSN. In this paper, one "Lead" MSN , or L-MSN, in addition to a set of "Member" MSNs, or M-MSNs, will be deployed. (The term MSN will refer to both L-MSN and M-MSN.)

Among the deployed MSNs, an M-MSN is mainly used for data collection and has limited processing and communication capability when compared to the L-MSNs. In particular, each M-MSN has a relatively small data transmission range when compared to the subarea it needs to search, especially for high bit rate transfers of images for example.

Meanwhile, the L-MSN performs some data compression and analsys, and then transmits the results to a remote command and control center (CCC) for processing and human interpretation. The L-MSN has either a long-range radio (an additional dedicate radio), or the same radio as M-MSNs' but through relaying or cooperative transmission by another L-MSN, to communicate with the CCC. It sends the M-MSNs command and control signals either generated by itself or received from the CCC in order to direct the M-MSNs to dynamically adapt their operations during the mission, with respect to, for example, what type of data to collect and where (which subarea) to collect the data, and how to communicate the data (including, e.g., encoding methods to use, and the future rendezvous time and place with other M-MSNs and itself).

When communicating with each other, both types of MSNs have a relatively small communication range when compared to the sub-area each of them has to cover, especially for high bit rate transfers of images for example. As a result, when two MSNs are each performing their search operations for example, they may be too far for them

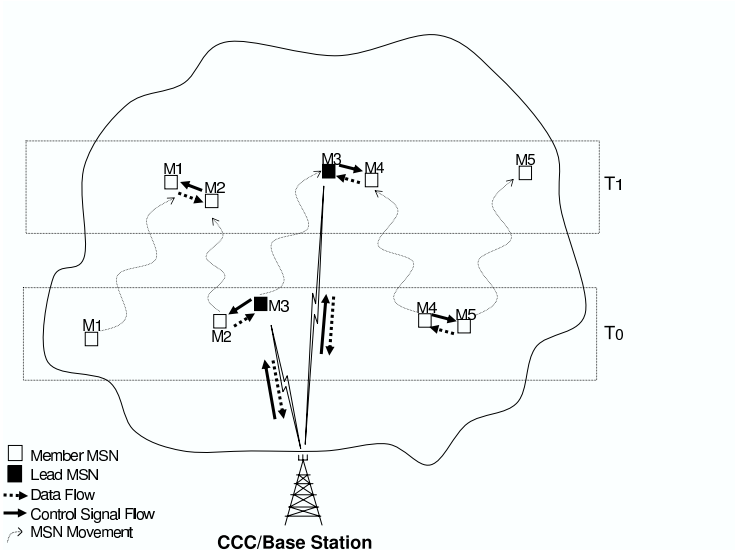


Fig. 1. A snapshot of a search operation

to communicate with each other. In other words, an M-MSN has to move close enough to another M-MSN or the L-MSN in order to transfer data between them. In this respect, they form a network where nodes are not always connected to each other (such a network is a special case of the delay-tolerant network (DTN) or intermittently connected network (ICN)).

A snapshot of the search operation of a team with a L-MSN and M-MSNs is shown in Fig. 1. At time T_0, M_2 and M_3 can communicate, but, due to limitation of communication range, communication is not possible between M_1 and M_2. M_1 collects data at time T_0, and sends the data to M_2 at time T_1. Afterwards, the data will be eventually delivered to the L-MSN in the middle of the team. As shown in the Fig. 1, the control signal flows in the direction opposite to that of the data.

Under the above architecture, for a given area and a limited number of MSNs, a major challenge is to successfully search the area within the shortest period of time with the shortest travel distance of MSNs.

To address the challenge under the configuration with multiple MSNs above, we first propose a load-balancing search approach called Lane Based Search (LBS), which divides the area width-wise into lanes, one for each MSN. The width of the area is assumed to be so large relative to the sensing and communication ranges of the MSNs that even the width of a lane is still larger than the sensing and/or communication ranges of each MSN. Accordingly, each MSN needs to move not only forward only but also left and right in order to search the entire lane and/or be able to communicate with the neighboring MSNs.

We also propose several "synchronization" schemes, including the so-called Strict-Line (SL), Alternating Columns (AC), and X synchronization schemes for the MSNs to exchange their data and perform other cooperative operations including failure detection and recovery. We determine the appropriate synchronization timeout period values to be used, and failure recovery procedures for each of these schemes. We also evaluate their performance in terms of total search time and average moving distance, and discuss the tradeoffs involved in these schemes.

There have been many studies on how to use MSNs for movement assisted sensor deployment [1], [2], how to use MSNs to increase coverage [4] or maintain coverage after failure of some sensors by relocating the surviving sensors [5], and how to use mobile nodes to collect sensor data [6]. However, none of these studies addresses the unique requirements of a search operation in a large area with multiple cooperative MSNs.

There have also been studies on individual MSN technologies based on UAVs and autonomous vehicles [7]–[9], as well as studies on team behaviors [10]–[14]. In particular, the study in [13] proposed the use of a Ranger node (similar to a "mother ship"), carrying a set of "scout" node for surveillance. More specifically, the Ranger node

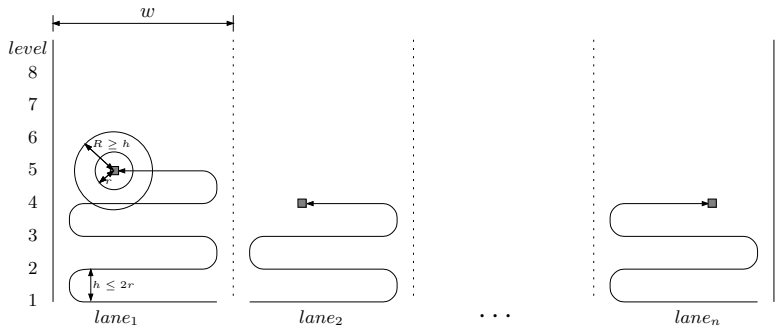


Fig. 2. Lane Based Search

would travel to a relatively small search area (office or lab environment) and deploy the scout nodes. However, none of these studies proposed protocols with the main objective being to minimize the mission completion time, let alone synchronization schemes in such a context.

A major contribution of the work is the proposed architecture that is practically useful due to its support for the use of heterogeneous, cost-effective MSNs with limited sensing and communication ranges, and its support for failure-resilience. Another major contribution is the proposed time-efficient protocol with appropriate synchronization schemes, and their performance evaluation.

The rest of the paper is organized as follows. Section 2 introduces a basic strategy called Lane Based Search for load-balancing. Section 3 describes three synchronization schemes. Section 4 devises appropriate timeout period for the schemes. Section 5 and Section 6 present the numerical analysis of the total search time and the average travel distance of MSNs for three schemes. Section 7 evaluates the performance of schemes through simulations. Finally, Section 8 concludes the paper.

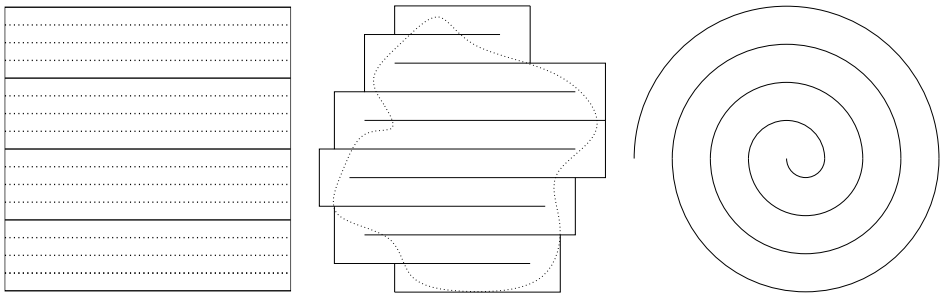
II. LANE BASED SEARCH STRATEGY

One of the objectives of the Lane Based Search (LBS) strategy is to achieve load balancing among MSNs. With LBS, a search area of H x W units is partitioned into n sub areas of equal size, where n is the number of MSNs. Each sub area, called a lane hereafter, is H x w units, where $w = \frac{W}{n}$. As shown in Fig. 2, all MSNs (including the L-MSN) are responsible for searching their designated lane. If the width of lane (w) is less than the sensing range of MSNs, r, MSNs move only forward in the search direction. For a large search area, however, w can be much larger than r. In that case, a MSN has to move horizontally as well as forward (as a snake does) as illustrated in Fig. 2. The search area can thus be considered to have multiple levels length-wise, and the height of each level, h, has to be less than or equal to $2r$ to avoid any sensing coverage hole. In addition, since the width of a lane can also be larger than MSNs' transmission range R (which may or may not be larger than r), an MSN has to move to the borders of its lane in order to communicate with its neighboring MSNs for relaying data and control signals. Note that the above assumptions on w, R and r are especially applicable to target identification using UWB sensor/radar based imaging and UWB radios for communications that can generate a large amount of data that can be transferred at a high bit rate at a relatively short communication range. In addition, the architecture and concept developed in this paper is general enough to be adaptable to other scenarios as well.

LBS can be applicable to cases with areas of various shapes. For example, Fig. 3 shows possible shapes of a search area. In Fig. 3(a), a rectangular shaped area is partitioned into multiple subareas (which are also in rectangular shape) each of which has multiple lanes represented as dotted lines. As shown in Fig. 3(b), an irregular shaped area can also be partitioned into multiple rectangular shaped subareas. If the last position of the search target is known, a search team may move along spiral shaped lanes as shown Fig. 3(c).

III. SYNCHRONIZATION SCHEMES

In order for MSNs to relay data and control signals, detect an MSN failure, and cooperate with other MSNs, synchronization at RPs among MSNs is required due to the differences in MSN's speeds and MSN's limited data transmission range. Synchronization among neighboring MSNs (or sync peers) occurs at RPs, which are agreed



(a) Rectangular (b) Irregular (c) Spiral

Fig. 3. Shapes of a Search area

upon locations predetermined by the synchronization scheme used. During synchronization, following operations are carried out.

- Data aggregation: The process in which M-MSNs communicate with their neighbors to deliver data (of neighbors and themselves) to the L-MSN.
- Control signal dissemination: The process in which control signals from the L-MSN or the base station are forwarded to all MSNs.
- MSN failure detection: MSN failure detection consists of two steps: the timeout event and failure confirmation. If an MSN's waiting time for its sync peer becomes greater than a pre-determined timeout period, it has a timeout event. Following the timeout, the MSN cooperates with other MSNs
- Failure recovery: Failure recovery follows failure confirmation. One of MSNs is chosen as a recovery node, and goes back up to the last synchronization position to cover the failed MSN's area. Further, MSNs in the search team perform reallocation of lanes to cover the remaining area without any coverage hole.
- Reallocation of lanes: If the number of the MSNs is changed, then the size and number of lanes are also changed accordingly, and MSNs are assigned to new lanes.
- Rejoining of a recovery node: After finishing the recovery for a failed MSN, the recovery node catches up with other MSNs and rejoins them.

In this paper, we propose three different distributed synchronization schemes. Each synchronization scheme has different failure detection and recovery algorithms. We present the details of the failure and recovery algorithm in the following subsections with the description of each synchronization scheme.

A. Alternating Column Synchronization(ACS)

In ACS, an MSN (shown as a square in Fig. 4) meets a given sync peer (one of its neighboring MSNs) at every other level as shown in Fig. 4. If M_i (or $MSN\ i$) has an RP with M_{i-1} at the left border of its lane at $level\ j-1$, it will have another RP with M_{i+1} at the right border of its lane at $level\ j$, and those two RPs are represented as $RP^{j-1}_{i-1,i}$ and $RP^{j}_{i,i+1}$ respectively (RPs are represented as shaded circles in Fig. 4).

In ACS, we assume that the radius of the transmission range of each MSN is at least twice longer than the sensing range, i.e., $R \geq 2h$. Also recall that $h \leq 2r$ so $R \geq h$. However, both R and r can still be much smaller than w.

If a MSN failure occurs, the MSN at a lower level among the two neighboring MSNs of the failed MSN will serve as a recovery MSN, and goes back at most one level to cover the failed MSN's area. Afterwards, the recovery MSN rejoins the other surviving MSNs, and the team continues the search operation.

1) ACS algorithm: In ACS, all MSNs (except for M_1 and M_n) have an RP at every level as shown in Fig 4. If an MSN (e.g. M_{i-1}) meets one of its neighbor (e.g. M_{i-2})at $level\ j$, it meets its *other* neighbor (e.g. M_i) at $level\ j+1$. The MSN which arrives at an RP earlier (e.g. M_{i-2} at $level\ j$) than its sync peer (e.g. M_{i-1}) waits for the neighbor (e.g. M_{i-1}) until it arrives at the RP or a timeout occurs. If the sync peer (e.g. M_{i-1}) reaches the RP before a timeout, they perform synchronization. During synchronization, the MSN farther from the L-MSN (e.g. M_{i-2}) forwards data (which it has generated and received from the other sync peer) to the MSN closer to

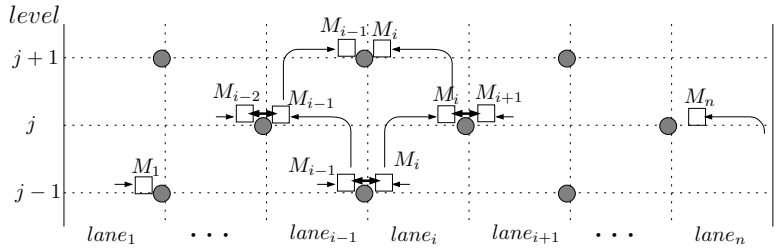


Fig. 4. Alternating Column Synchronization

the L-MSN (e.g. M_{i-1}), while the control signals are also forwarded in the opposite direction (i.e. from M_{i-1} to M_{i-2}). On the other hand, if an MSN (e.g. M_{i-2}) has a timeout, it moves toward the failed MSN's (e.g. M_{i-1}) other sync peer (e.g. M_i) in order to confirm the failure. Note that since the height of each level is at most R , as long as there is at most one MSN failure at a time, the two neighboring MSNs of the failed MSN will eventually be in radio range of each other to confirm the failure of their common neighbor. Following failure confirmation, the MSN at a lower level (e.g. M_{i-2}) goes back at most one level to cover the failed MSN's area. Afterwards, the MSN rejoins the other surviving MSNs, and the team continues the search operation.

2) MSN Failure Detection and Recovery in ACS: When M_k $(1 < k < n)$ fails at an arbitrary $level\ j$, both of its neighbors M_{k-1} and M_{k+1} (or sync peers) will have a timeout at different moments at either $level\ j$ and $level\ j+1$ or $level\ j+1$ and $level\ j$ respectively. As an example in Fig. 5(a), M_3 has failed, and M_2 and M_4 have a timeout event in Fig. 5(b) and Fig. 5(c) respectively. Immediately after having a timeout , neighboring nodes of the failed MSN move to meet each other along the cross line over the lane of the failed MSN up to a distance of $2w$. The maximum level difference between the neighbors of the failed MSN is only 1, and hence the MSN with an earlier timeout definitely meets the other sync peer of the failed MSN within a distance of $2w$. If either one of those MSNs sees M_k while moving to each other, the timeout event is false (they may have had an inappropriate timeout period or M_k has traveled with a much lower speed than expected). In that case, they go back to their RP and wait for M_k for a normal synchronization. If the two neighbors only see each other (e.g., M_2 and M_4 as shown in Fig. 5(d)), they confirm the failure of M_k (M_3), and re-calculate the size of lanes because there are only $n-1$ active nodes. The MSN which is at a lower level (M_2 in Fig. 5(d)) is chosen as a recovery node, and the recovery node covers the failed MSN's area at the current level. In this case, there is no need for a recovery node to move levels back, because $level\ j-1$ must have already been covered by the failed MSN. The recovery node only needs to cover $level\ j$ of the failed MSN's lane. The neighbors also forward the information about the change of lanes and the failure to other MSNs. In Fig. 5(f), there are only 3 lanes for 3 MSNs, and MSNs have new RPs represented as white circles.

Note that if M_1 (or M_n) fails, then M_2 or (M_{n-1}) will have a timeout event, and can immediately move towards M_3 (or M_{n-2}) to inform it and the rest of the MSNs of the failure. Afterwards, M_2 (or M_{n-1}) will go one level below for recovery.

The frequent synchronization of ACS, and hence the large aggregate synchronization delay over the operation can degrade the performance in terms of the total search time. However, ACS has a quick recovery process due to the fact that a recovery node needs to go back to at most one level for covering the area of failed MSN.

B. Strict Line Synchronization(SLS)

In SLS, synchronization among MSNs occurs at RPs on designated sync lines. As shown in Fig. 6, a search team has a sync line after every K levels during a search mission. The search team first completes a search in a section (which is the area between two consecutive sync lines), and then performs synchronization before it continuing the search. The synchronization process has two phases: the data aggregation phase and the control signal dissemination phase. In the data aggregation phase, an MSN first visits the RP further from the L-MSN in order to meet one of its sync peers. After the MSN meets one of its sync peer (which is further from the L-MSN than the MSN itself) and receives data from that sync peer, the MSN visits its other RP closer to the L-MSN to meet its other sync peer to forward the data. When the two center MSNs (one of which is the L-MSN) meet, the data aggregation phase is

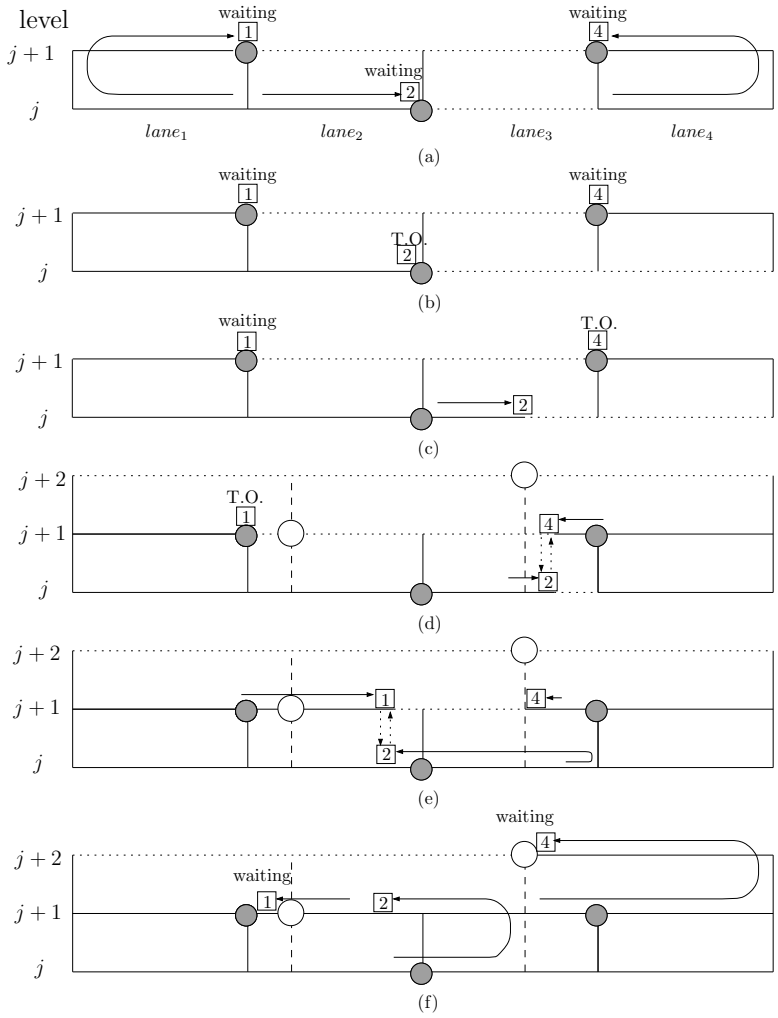


Fig. 5. ACS: Failure Detection and Recovery

completed. The data aggregation phase is followed by the control signal dissemination phase in which MSNs meet their sync peers in the reverse order in order to disseminate control signals from the L-MSN.

In SLS, if an MSN fails to reach a sync line, its two neighbors will have a timeout event and meet to confirm the failure. Then, one of them will serve as a recovery MSN, and go back up to K levels to cover the failed MSN. Meanwhile, the remaining MSNs adjust their lane assignment to avoid any sensing coverage hole and continue the search operation. Eventually, after finishing the recovery, the recovery MSN will catch up with the other surviving MSN with a straight line movement and join the the search operation.

1) SLS algorithm at a sync line: The search team is divided into two groups: the left group and the right group. The left group comprises MSNs that have IDs less than or equal to $\lceil \frac{n+1}{2} - 1 \rceil$. Other MSNs belong to the right group. Here n is the number of MSNs. There are two center nodes in the middle of the search team, each of which belongs to a different group. As shown in Fig. 7(a), there are only $n-1$ RPs at a sync line.

Synchronization consists of two phases: the data aggregation phase, and the control signal dissemination phase. In the data aggregation phase, M_1 moves to $RP_{1,2}$ upon arrival at a sync line. For MSNs with ID greater than 1 (i.e. $i > 1$), M_i moves to $RP_{i-1,i}$ (Here we assume that M_i belongs to the left group.) If the MSN belongs to the right group, it first moves to $RP_{i,i+1}$, and has the opposite moving direction. For example, M_2 first visits $RP_{1,2}$ as shown Fig. 7(a). Note that, in Fig. 7(a), all MSNs except M_2 and M_6 have already arrived at the sync line. If $RP_{i-1,i}$ is already occupied by M_{i-1}, then M_i receives data from M_{i-1} and moves to $RP_{i,i+1}$. In other words, M_{i-1} forces M_i to move toward the center of the team (e.g., M_1 and M_2 force M_2 and M_3 to move toward

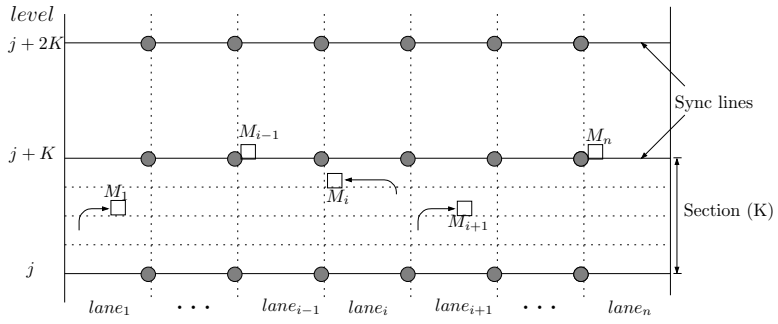


Fig. 6. Strict Line Synchronization

the center as shown Fig. 7(b) and Fig. 7(c) respectively). If $RP_{i-1,i}$ is not occupied, M_i waits for M_{i-1} until the timeout period expires or M_{i-1} arrives. In this way the two center nodes meet at an RP eventually (e.g., M_3 and M_4 in Fig. 7(g)), which means all MSNs, except for the nodes at the edges, have met both of its neighbors and relayed data. When the two center nodes meet at an RP, the L-MSN (which is one of the center nodes) receives data of all MSNs and completes the data aggregation phase. In the control signal dissemination phase that follows the data aggregation phase, M_i receives a control signal from M_{i+1} at $RP_{i,i+1}$, and moves toward $RP_{i-1,i}$ to forward the signal. Immediately after forwarding the control signal (without waiting for the completion of the dissemination phase), M_i continues the search operation up to the next sync line.

2) MSN Failure Detection and Recovery in SLS: Suppose M_k belongs to the right group and it fails to arrive at a sync line (for example, suppose M_6 fails before it arrives at a sync line as shown Fig. 7(a)). The failure causes both of M_k's neighbors, M_{k-1} and M_{k+1}, to timeout at $RP_{k-1,k}$ and $RP_{k,k+1}$ respectively (possibly simultaneously). As an example, in Fig. 7(d), M_5 and M_7 have a timeout event at $RP_{5,6}$ and $RP_{6,7}$ respectively. Following a timeout, two neighbors move toward the lane assigned to the failed MSN. When M_k's neighbors meet (M_5 and M_7 in this example), they confirm the failure, and the outer MSN among them, M_{k+1} (M_7), is chosen as a recovery node, and the inner node M_{k-1} (M_5) moves toward M_{k-2} (M_4) and informs it of the failure. As shown in Fig. 7(f), the recovery node M_7 goes back up to the last sync line to cover the area of the failed MSN. Note that other MSNs can also have a timeout event like M_4 in Fig. 7(e). The timeout causes $M4$ moves toward M_5 to see M_5 or M_6, because M_4 does not know whether M_5 has failed or M_5 has been waiting for M_6. When M_4 and M_5 meet in Fig. 7(f), M_4 learns of M_6's failure, and then it also forwards this information to M_3 in Fig. 7(g). When the L-MSN (M_3) is informed of the failure, it performs reallocation of lanes (which results in $n-2$ lanes) and forwards this change to the search team members (e.g., the search team has only 5 lanes in Fig. 7(h)). The remaining $n-2$ MSNs in the search team do not wait for the recovery node to return back to the sync line. Instead, they continue the search operation. After the recovery node finishes recovery, it catches up with the team with linear movement and rejoins the team by sending a rejoin request message to other MSNs. The rejoining process occurs at a sync line where the recovery node arrives earlier than other MSNs. After the recovery node rejoins the team, the team performs reallocation of lanes and has $n-1$ lanes. Note that the recovery node knows the position of the L-MSN, because the L-MSN periodically broadcasts control messages containing its location with a high transmission power.

SLS may have a lower synchronization overhead than ACS because, in SLS, the search team does not synchronize at every level. This can reduce the total search time. However, in SLS, it may take a longer time to detect and recover from an MSN failure. Also note that, unlike ACS in which only one MSN failure can be handled at a time, SLS can handle multiple simultaneous MSN failures, because, in SLS, all surviving MSNs are on the same sync line during synchronization. In other words, after having a timeout, each surviving MSN will eventually meet its surviving neighbors on a sync line by moving until it meets another MSN or reaches the left border of the lane of M_1 or the right border of the lane of M_n.

C. X Synchronization(XS)

In XS, the MSNs in a search team perform a search individually until they reach an area called *sync section*. In a sync section, MSNs perform synchronization including data aggregation (DA) and control signal dissemination

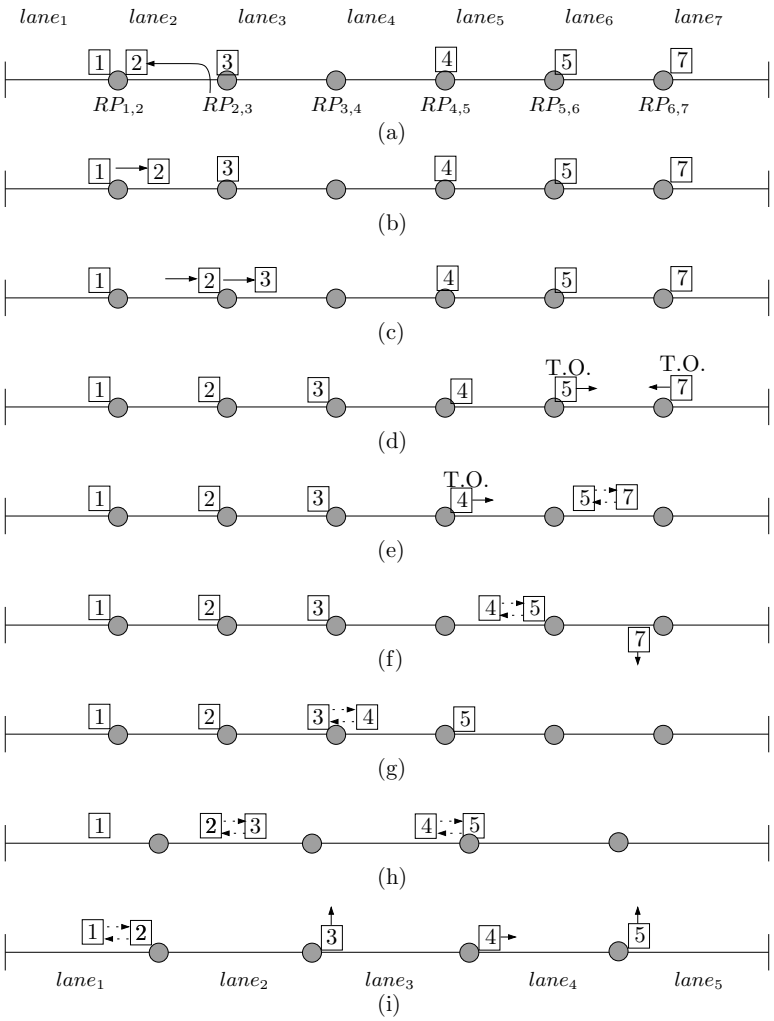


Fig. 7. SLS: Failure Detection and Recovery at a sync line

(CSD). After finishing synchronization in a sync section, the MSNs continue the search in the following non-sync section as shown Fig. 8. A sync section is set after every K level during the search operation, and has n levels as its height as shown in Fig. 8. The size of a non-sync section, K is a system parameter that dictates the frequency of the synchronization of the team during the search mission.

The synchronization process consists of two phases: the data aggregation (DA) and the control signal dissemination (CSD) phase. The DA phase precedes the CSD phase in a sync section. The DA phase and the CSD phase occur along the DA segment and CSD segment shown in Fig. 8 respectively. In other words, as shown in Fig. 8, data collected by MSNs and control signals from the L-MSN are forwarded along the DA segment and the CSD segment respectively.

For the XS algorithm, the search team is logically divided into two groups: the left group and the right group. The MSNs with ID less than or equal to $\lfloor \frac{N+1}{2} \rfloor$ belong to the left group. Otherwise, MSNs belong to the right group. Two MSNs in the middle of the team are refereed as center MSNs, each of which belongs to a different group. Note that at least one of the center MSNs is the L-MSN.

To facilitate the presentation, for the time being, we assume that $R \geq h$ where R and h are the data transmission range and the height of each level respectively. We will discuss how this constraint can be removed at the end of this section. In addition, how the search team can recover from multiple MSN failures will be also discussed. In the following discussion, we will denote the RP at $level\ j$ for $MSN\ i$ (or M_i) and $MSN\ i+1$ (or M_{i+1}) by $RP^j_{i,i+1}$. We also assume that the search team has n MSNs where n is any even number larger than 2.

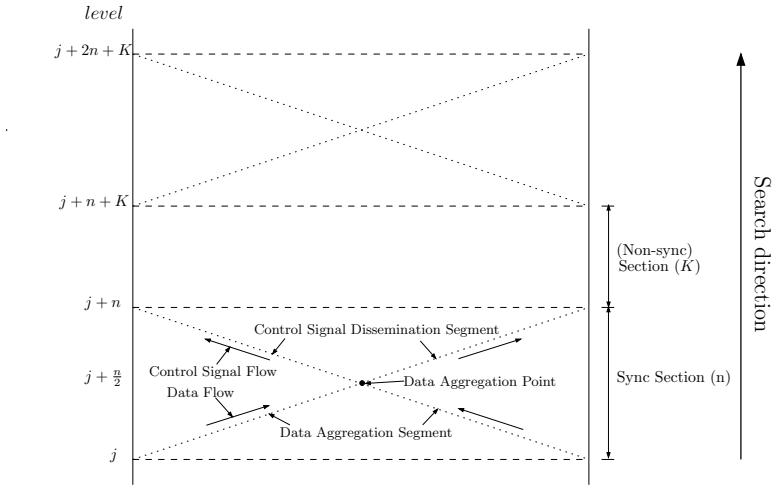


Fig. 8. X Synchronization

1) XS algorithm at a sync section: The algorithms of DA and CSD for the right group are described in this section (while those for the left group are similarly). Fig. 9 illustrates the synchronization process. Suppose the sync section begins at $level\ j$ as shown in Fig. 9.

The MSNs in the group use a slightly different algorithm depending on their location in the search team as follows.

- M_n: M_n searches its lane until it reaches its first RP with M_{n-1} at $level\ (j+1)$ (i.e. $RP_{n-1,n}^{j+1}$) in a sync section. At $RP_{n-1,n}^{j+1}$, M_n sends data it has collected since the previous DA segment to M_{n-1}. Then, M_n continues the search until it reaches $RP_{n-1,n}^{n+j-1}$ to synchronize with M_{n-1} again and receive from M_{n-1} control signals which M_{n-1} has received from its other sync peer. After receiving control signals, M_n continues searching its lane.
- M_i ($\frac{n}{2}+1 < i < n$): An intermediate MSN, M_i searches its lane until it reaches $RP_{i,i+1}^{n-i+j}$ at $level\ (n-i+j)$. At $RP_{i,i+1}^{n-i+j}$, M_i receives from M_{i+1} the aggregated data from M_{i+1} to M_n. Then, as shown in Fig. 9, M_i moves toward its second RP at the next $level\ (n-i+j+1)$, $RP_{i-1,i}^{n-i+j+1}$, and forwards to M_{i-1} the data it has collected by its sensor and by M_{i+1} through M_n. After forwarding data, M_i continues the search up to $RP_{i-1,i}^{i+j-1}$, and meets M_{i-1}, and then receives control signals. M_i forwards the control signal to M_{i+1} at $RP_{i,i+1}^{i+j}$ and continues the search.
- $M_{\frac{n}{2}+1}$ (center MSN): Suppose $M_{\frac{n}{2}+1}$ is the L-MSN. After it receives the aggregated data from $M_{\frac{n}{2}+2}$ at $RP_{\frac{n}{2}+1,\frac{n}{2}+2}^{\frac{n}{2}+j-1}$, it moves toward $RP_{\frac{n}{2},\frac{n}{2}+1}^{\frac{n}{2}+j}$ to synchronize with the other center MSN, $M_{\frac{n}{2}}$. When $M_{\frac{n}{2}+1}$ meets $M_{\frac{n}{2}}$ and receives the data aggregated from all MSNs, the DA phase is completed. $M_{\frac{n}{2}+1}$ communicates with CCC if necessary and then initiates the CSD phase to disseminate the control signal or commands from CCC or itself.

2) MSN Failure Detection and Recovery in XS: If M_{i-1} has a timeout while waiting for M_i at an RP, it will try to confirm with M_{i+1} that M_i has failed. Note that the failure confirmation process is required, because the timeout at M_{i-1} does not always indicate the failure of M_i; it is possible that M_i has been waiting for M_{i+1} at another RP. Following the failure confirmation, an MSN is chosen as a recovery MSN, and remaining MSNs adjust their lanes to avoid any sensing coverage hole and to re-balance the load.

Depending on the location of the failure, there are two cases of an MSN failure: MSN Failure before a DA segment, and MSN Failure after a DA segment.

MSN Failure before the DA segment

Suppose M_i ($\frac{n}{2} < i < n$) fails to arrive at a DA segment in a sync section which begins at $level\ j$. The failure of M_i causes timeout events (possibly at the same time but could be different times) of M_{i-1} and M_{i+1} at RPs at

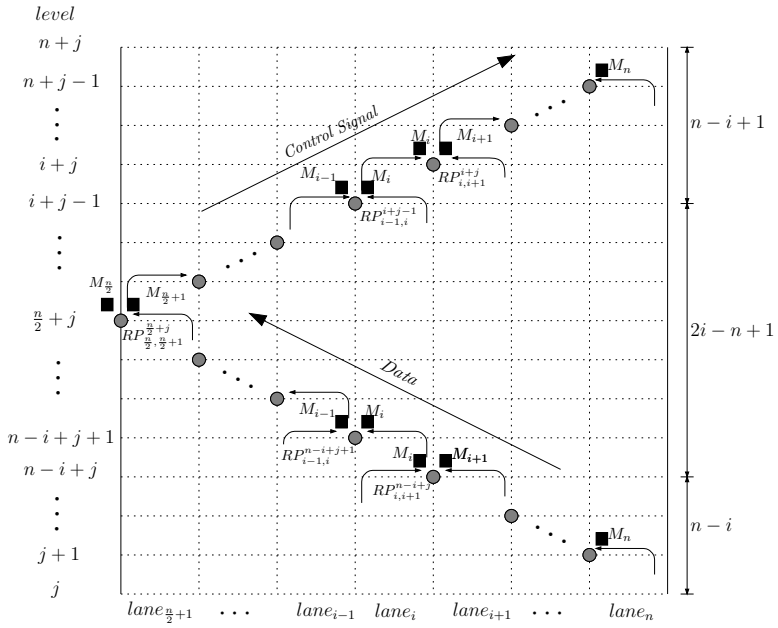


Fig. 9. XS: Synchronization of the right group

$level\ (n-i+j)$ ($RP^{n-i+j}_{i-1,i}$) and $level\ (n-i+j+1)$, $RP^{n-i+j+1}_{i,i+1}$, respectively. Following the timeout, M_{i-1} and M_{i+1} move toward the lane assigned to M_i as shown in Fig. 10(a). If M_{i-1} or M_{i+1} sees (i.e. can communicate with) M_i during their movement, the timeout is a false alarm (which is possible due to too small timeout period). In such a case it returns to its RP, and waits for M_i for normal synchronization. If M_{i-1} and M_{i+1} see each other but not M_i, as shown in Fig. 10(a), they confirm the failure of M_i and initiate a recovery process. After the confirmation, M_{i+1}, which is at a lower level, is chosen as a recovery MSN, and, as shown in Fig. 10(b), goes back to cover the lane assigned to M_i up to the previous DA segment. Meanwhile, the MSN at a higher level, M_{i-1}, moves toward its other sync peer, M_{i-2}, and informs the latter (which may or may not have a timeout event) of the failure as shown in Fig. 10(b), and continues the search until it reaches the CSD segment. The information about the failure is eventually delivered to the L-MSN at $level\ (\frac{n}{2}+j)$. On receiving failure notification, the L-MSN recalculates the width of lanes and forwards a control signal containing this information to its neighbors. M_{i-1} receives the control signal at $level\ i+j-2$ as shown in Fig. 10(c). To deal with the fact that both M_i and M_{i+1} are absent from the CSD process, M_{i-1} moves up and right as in climbing a staircase towards $RP^{i+j+1}_{i,i+1}$ to forward the control signal to M_{i+2} (which may have a timeout and also been moving downward as shown in Fig. 10(d). Immediately after forwarding control signals, M_{i-1} moves toward its newly assigned lane, and continues the search operation until the next DA segment. Note that after the recovery MSN (M_{i+1}) finishes the recovery, it will catch up with other MSNs by moving straight in the search direction. Also note that the L-MSN periodically broadcasts its location with a high transmission power, which makes it possible for the recovery MSN to detect the signal when it follows along the lane of the L-MSN. The process for the recovery MSN to rejoin other MSNs occurs in the first sync section encountered. For the L-MSN to reallocate the lanes are similar to the failure detection and recovery process described above and hence, will be omitted.

Note that if M_1 or M_n fails, its neighbor (M_2 or M_{n-1}) timeouts at $level\ j+1$. In this case, M_2 (or M_{n-1}) does not need failure confirmation (because it is not possible that M_1 or M_n is waiting for another MSN), and immediately moves toward the other neighbor, M_3 (or M_{n-2}) in order to inform it of the failure. Then, M_2 (or M_{n-1}) becomes a recovery MSN, and goes back to cover the failed MSN.

If a center MSN, $M_{\frac{n}{2}}$ (or $M_{\frac{n}{2}+1}$) fails, $M_{\frac{n}{2}-1}$ and $M_{\frac{n}{2}+1}$ (or $M_{\frac{n}{2}}$ and $M_{\frac{n}{2}+2}$) have a timeout and meet to confirm the failure. After confirmation of the failure, the remaining center MSN, $M_{\frac{n}{2}+1}$ (or $M_{\frac{n}{2}}$) forwards the control signal both $M_{\frac{n}{2}-2}$ and $M_{\frac{n}{2}+2}$ (or $M_{\frac{n}{2}-1}$ and $M_{\frac{n}{2}+3}$).

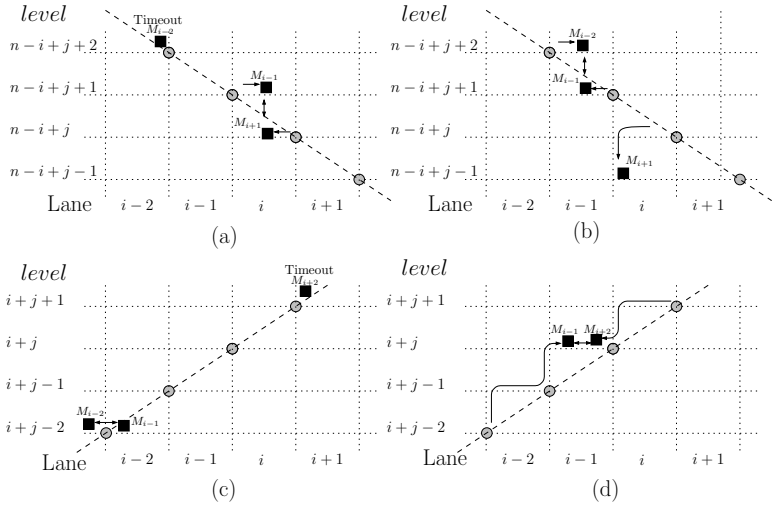


Fig. 10. MSN failure detection and recovery

MSN Failure after a DA segment

It is possible that M_i fails between a DA segment and a following CSD segment after it successfully completes its operation at the DA segment. In this case, M_{i-1} and M_{i+1} have a timeout at $RP_{i-1,i}^{i+j-1}$ and $RP_{i,i+1}^{i+j}$ respectively during CSD. Following the timeout, M_{i-1} and M_{i+1} meet and confirm the failure. The MSN at lower level, M_{i-1}, is chosen as a recovery MSN, and goes back to cover the lane of M_i. M_{i+1} recalculates the lane assignment so that M_{i+1} through M_n can cover lanes from $lane\ i-1$ to $lane\ n$ from this moment on, and informs M_{i+2} of the failure and the reassigned lane, which in turn informs M_{i+3} and so on. At the next DA segment M_{i+1} meets M_{i+2} to forward data and inform the latter of the failure of M_i in the last sync section. The information about the failure is delivered to the L-MSN, and the L-MSN performs reallocation of lanes accordingly among all surviving MSNs.

So far, we have assumed that only one MSN failure occurs at a time. However, XS can be extended to detect and recover multiple simultaneous MSN failures. For example, in Fig. 9, if two consecutive MSNs, M_{i-1} and M_i, have failed before they reach the DA segment, M_{i-2} and M_{i+1} will have a timeout event, and they will meet eventually while moving toward the lanes of M_{i-1} and M_i respectively with staircase movement along RPs on the DA segment. Note that non-consecutive MSN failures can also be detected. If M_{i-2} and M_i have failed in Fig. 9, M_{i-1} and M_{i+1} will first detect the failure of M_i, and then, M_{i-3} and M_{i-1} will also detect the failure of M_{i-2} when they meet.

Recall that we have an assumption of $R \geq h$. This constraint is needed for two neighboring MSNs of the failed MSN to communicate each other when they are one level apart. However, this constraint can be relieved by allowing MSNs to move along the DA or CSD segment represented in a dotted line in Fig. 10.

IV. MOBILE SENSOR FAILURE TIMEOUT CALCULATION

In order for MSNs to detect a MSN failure a timeout mechanism is used. A proper timeout value is necessary to avoid a waste of time waiting for a failed sensor with a too large timeout value or a needless recovery process with a premature timeout.

The estimated arrival time distribution of an MSN at the RP is used to decide a timeout period for the MSN.

In order to estimate the movement time distribution of MSNs for some distance, we first obtain the movement time distribution of an MSN for a distance d for some value d, and use this distribution to estimate the distribution of the movement time for a longer distance. On the other hand, to calculate the delay, the sync algorithm should be considered, because MSNs' delay at their RPs are co-related to other MSNs' delay at other RPs. In other words, for the accurate calculation of the delay of an MSN, the knowledge of all other MSNs's travel time and delay is necessary, which makes the calculation time consuming. To solve this problem we propose simplified ways of calculating the estimated arrival time distribution.

After obtaining the arrival time distribution, we calculate a timeout value within which a peer sensor arrives at the next RP with a given probability of P_o.

A. Timeout Calculation in ACS

To facilitate the description, let's define a few random variables and constants:

- X_i^l: The arrival time of M_i at the RP at $level_l$;
- $T_i^{l,m}$: The movement time of M_i between two RPs at $level_l$ and $level_m$;
- D_i^l: The delay (including waiting time for a sync peer and synchronization delay) of M_i at the RP at $level_l$;
- S_i^l: The time synchronization is completed between M_i and its sync peer at $level_l$;
- $d_{i,j}$: The synchronization delay between M_i and M_j.

In ACS, an MSN has an RP at every level. Let M_i have an RP at the right border at $level_i$, then the arrival time of M_i at the RP, X_i^l, is

$$X_i^l = T_i^{l-2,l-1} + D_i^{l-1} + T_i^{l-1,l} + S_i^{l-2} \tag{1}$$

Note that S_i^{l-2} is already obtained in a previous synchronization. The synchronization delay, D_i^l, depends on which synchronization peer arrives at the RP earlier. That is,

$$D_i^{l-1} = \begin{cases} X_{i-1}^{l-1} - (T_i^{l-2,l-1} + S_i^{l-2}) + d_{i,i-1} & \text{if } X_{i-1}^{l-1} - (T_i^{l-2,l-1} + S_i^{l-2}) > 0; \\ d_{i,i-1} & \text{otherwise.} \end{cases} \tag{2}$$

From 1 and 2, the arrival time can be rewritten as

$$X_i^l = T_i^{l-1,l} + max(X_{i-1}^{l-1}, T_i^{l-2,l-1} + S_i^{l-2}) + d_{i,i-1} \tag{3}$$

As shown in 3, the arrival time of M_i at $level_l$ is depend on the arrival times of M_1,...,M_{i-1} at $level_{l-i}, ..., level_{l-1}$ respectively.

Calculating the distribution function from the result above will be time consuming (Note that there can be up to $N-1$ random variables and max operations which require $(N-1)$-tuple integration). To simplify the calculation, we assume that the movement time distributions of MSNs are the same for an equal distance, and MSNs farther from M_i arrives at RPs earlier than MSNs closer to M_i, and stay at the RPs for a constant delay d_{sync}.

Whether an MSN is the right-hand side sync peer (RSP) or the left-hand side sync peer (LSP) affects the way of calculation. Let random variable $\mathbf{t_i^y}$ be M_i's travel time for a distance y. Then, the estimated arrival time distribution of M_i at an RP at $level\ l$ is

$$X_i^l = \begin{cases} S_1^{l-(i+1)} + t^{2(w+h)+(i-1)(w+h)} + (i-1)d_{sync} & \text{if } M_i \text{ is L.S.P;} \\ S_n^{l-(n-i+2)} + t^{2(w+h)+(n-i)(w+h)} + (n-i)d_{sync} & \text{if } M_i \text{ is R.S.P;} \end{cases} \tag{4}$$

Recall that $\mathbf{t^y}$ is a random variable that represents the travel time of an MSN for a distance y. In order to estimate the travel time distribution of an MSN for a given distance y (e.g. between two consecutive RPs of the two MSNs), we first obtain the travel time distribution of an MSN for a distance d for some value d through simulations. More specifically, Let random variable $\mathbf{s}$ be the travel time of an MSN for a distance d (which is much less than w), and the probability density function (p.d.f) of $\mathbf{s}$ be given as $f_s(s)$. Let $m = \frac{y}{d}$, then $\mathbf{t^y}$ can be represented as the sum of $\mathbf{s}$.

$$\mathbf{t^y} = \mathbf{s_1} + \mathbf{s_2} + \cdots + \mathbf{s_m} \tag{5}$$

Because $\mathbf{s_i}$ is i.i.d., the p.d.f of $\mathbf{t^y}$ is the convolution of $f_{s_i}(t^y)$ where $1 \leq i \leq m$. Further, according to the Central Limit Theorem, the distribution of $\mathbf{t^y}$ approaches a normal distribution if m is large enough. More specifically, let μ and σ^2 represent the mean and variance of $\mathbf{s}$ respectively, then $\mathbf{t^y}$ will approximately follow a normal distribution with the mean, $m\mu$, and the variance, $m\sigma^2$.

Let D be the total synchronization delay, then, the estimated arrival time $\mathbf{x_i}$ also follows a normal distribution with ($m\mu + D$, $m\sigma^2$) as its mean and variance. From the distribution function of $\mathbf{x_i}$, an arrival time, t_0, that corresponds to a given probability P_0 is chosen as the candidate of timeout period (i.e. $P_0 = P\{\mathbf{x} \leq t_0\}$). The timeout value, TO is chosen as $TO = \beta t_0$ where β (≥ 1) is a design parameter to add some safety margin against premature timeout.

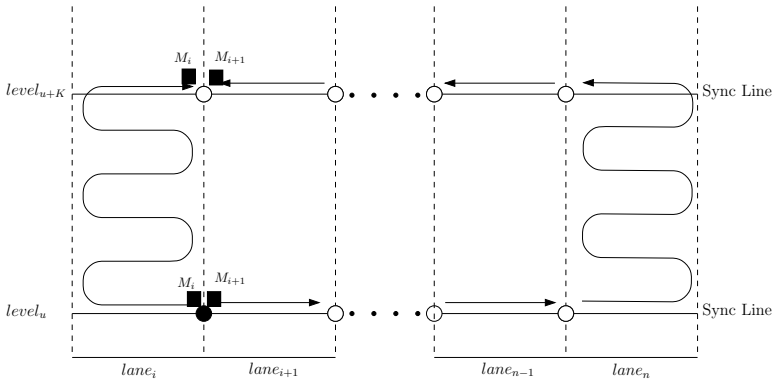


Fig. 11. Timeout Calculation in Strict Line Synchronization

B. Timeout Calculation in SLS

Suppose M_i belongs to the right group of the search team, and M_i and M_{i+1} meet in a dissemination phase at $level\ u$ as shown Fig. 11, and MS_i forwards a control message to MS_{i+1}. Before M_i and M_{i+1} meet and synchronize at the next synch line at $level\ u+K$, three processes should occur in order as shown in Fig. 11. First, the control signal is forwarded to M_n. Second, M_n performs the search until it reaches the next synchronization line. Third, sensing data of MSNs are forwarded to M_{i+1}. In other words, M_i waits for M_{i+1} until data aggregation is completed for the sensors from M_n to M_{i+1}. Therefore, the arrival time of M_{i+1} can be considered as the sum of the time needed to complete these three processes.

On the other hand, the arrival time of M_i can be directly obtained from it travel time distribution, because no prior synchronization is needed.

Note that the distance between two neighboring RPs is w, and there are K levels between two consecutive sync lines. From these observation, we obtain the estimated arrival time of M_i, $\mathbf{x_i}$ be at the RP on the next sync line. That is,

$$\mathbf{x_i} = \begin{cases} t^{K(w+h)+w(n-i-1)} + 2(n-i-1)d_{sync} & \text{if } M_i \text{ is L.S.P and } i > \lceil \frac{n+1}{2} - 1 \rceil; \\ t^{K(w+h)+w(i-1)} + 2(i-1)d_{sync} & \text{if } M_i \text{ is R.S.P and } i \leq \lceil \frac{n+1}{2} - 1 \rceil; \\ t^{K(w+h)} & \text{otherwise.} \end{cases} \tag{6}$$

Once the estimated arrival time,$\mathbf{x_i}$, is obtained, the appropriate timeout period is also calculated in a similar way to the case of ACS.

C. Timeout Calculation in XS

Let random variable $\mathbf{x_i}$ be the estimated arrival time of M_i at the next RP, and let $d_{p,q}$ $(1 \leq p, q \leq n, q = p+1)$ represent the sync delay which is the amount of time elapsed from the arrival of the earlier MSN (among M_p and M_q) to the completion time of the synchronization between M_p and M_q.

Suppose that two arbitrary MSNs, M_{i-1} and M_i ,where $\lfloor \frac{n+1}{2} \rfloor < i-1, i \leq n$, synchronize with each other at an RP at $level\ u$ on a CSD segment as shown in Fig. 12, and they calculate the timeout period for each other for the next RP at $level\ (u + 2(n-i+1) + K)$, at the next DA segment.

We first estimate the arrival time of M_i. Between this moment of synchronization (of M_{i-1} and M_i at $RP^u_{i-1,i}$) and the next synchronization at $RP^{u+2(n-i+1)+K}_{i-1,i}$ (note that the position of M_i in the team and K determine the level of the next RP), three processes should occur in order as shown in Fig. 12. First, the control signal is forwarded to M_n at $level\ (u+n-i)$ on the CSD segment. Second, M_n performs a search to its next RP at $level$ $(u+n-i+K+2)$. Third, data collected by MSNs are forwarded to M_i at $level\ (u+2(n-i)+K+1)$, and then, M_i moves toward to $RP^{u+2(n-i+1)+K}_{i-1,i}$ to forward data to M_{i-1}. Therefore, the arrival time of M_i can be considered as the sum of the time needed to complete these three processes.

In the CSD and DA phase, the data or control signals are forwarded through $2(n-i)$ RPs. Note that the distance between two RPs is $w+h$.

Therefore, the arrival time of M_i at $RP_{i-1,i}^{u+2(n-i+1)+K}$, $\mathbf{x_i}$, can be represented as

$$\begin{aligned} \mathbf{x_i} &= \mathbf{t_i^{(w+h)}} + \cdots + \mathbf{t_{n-1}^{(w+h)}} + \mathbf{t_{n-1}^{(w+h)}} + \cdots + \mathbf{t_i^{(w+h)}} && (7) \\ &+ \mathbf{t_n^{(w+h)(K+2)}} && (8) \\ &+ d_{i,i+1} + \cdots + d_{n-1,n} + d_{n,n-1} + \cdots + d_{i+1,i} && (9) \end{aligned}$$

where the terms at the first line represent the sum of travel time of M_j ($i \le j \le n-1$) during DA and CSD, and the second line represents the search time of M_n for $K+2$ levels, and the terms at the third line represent the sum of sync delay among MSNs during DA and CSD.

We also observe that, when M_j ($i \le j \le n$) arrives at $RP_{j-1,j}$ on the DA segment, most probably M_{j-1} has already arrived at the RP. This is because, for M_j to arrive at the RP, $2(n-j)$ synchronization operations should precede at other RPs, while the arrival of M_{j-1} does not require any preceding synchronization. (Note that M_j and M_{j-1} have an equal distance to travel). For the same reason, when $M_{j'}$ ($i \le j' < n$) arrives at $RP_{j',j'+1}$ on the CSD segment, most probably $M_{j'+1}$ has already arrived at the RP. Under these observations, we assume that an MSN (the left-hand side MSN during CSD and the right-hand side MSN during DA) can begin synchronization with its neighboring MSN immediately after it arrives at an RP (without waiting for its neighbor). In addition, for simplicity, we assume that the time to transmit data and control signals between two sync peers is some constant value d_{sync}. From the above observation and assumption, we have $d_{p,q} = d_{sync}$ (where $1 \le p, q \le n$ and $q = p+1$).

Further, we assume that all MSNs' speed variations have the same statistical properties. Then, the total travel time in (7) and (8) can be regarded as the travel time of one MSN for the distance of sum of travel distance in (7) and (8).

Then, setting the last synchronization time between M_{i-1} and M_i to 0, (7) - (9) becomes,

$$\mathbf{x_i} = \mathbf{t_i^{(K+2n-2i+2)(w+h)}} + 2d_{sync}(n-i) \tag{10}$$

On the other hand, the arrival time of M_{i-1} ($\mathbf{x_{i-1}}$) can be directly obtained from MSNs' travel time distribution.

$$\mathbf{x_{i-1}} = \mathbf{t^{(w+h)(2(n-i+1)+K)}} \tag{11}$$

This is because no prior synchronization is required for M_{i-1} to arrive at $RP_{i-1,i}^{u+2(n-i+1)+K}$.

Note that, whether an MSN is the right-hand side sync peer (RSP) or the left-hand side sync peer (LSP) and whether it belongs to the right group or the left group will affect the way of calculation. So far, we have assumed that M_i and M_{i+1} are in the right group. In general, let $r = \lfloor \frac{n+1}{2} \rfloor$, then $\mathbf{x_i}$ for the arrival time of an arbitrary MSN, M_i ($1 \le i \le n$) at an RP on the DA segment is,

$$\mathbf{x_i} = \begin{cases} \mathbf{t^{(K+2n-2i+2)(w+h)}} + 2d_{sync}(n-i) & \text{if } i > r \text{ and } M_i \text{ is R.S.P;} \\ \mathbf{t^{(K+2n-2i)(w+h)}} & \text{if } i > r \text{ and } M_i \text{ is L.S.P;} \\ \mathbf{t^{(K+2i)(w+h)}} + 2d_{sync}(i-1) & \text{if } i \le r \text{ and } M_i \text{ is L.S.P;} \\ \mathbf{t^{(K+2i-2)(w+h)}} & \text{if } i \le r \text{ and } M_i \text{ is R.S.P.} \end{cases} \tag{12}$$

With the obtained estimated arrival time,$\mathbf{x_i}$, the timeout period can be calculated in a similar way to the case of ACS.

The random variable $\mathbf{x'}$ representing a timeout period for an arbitrary MSN, M_i, at an RP on CSD segment can be represented in similar way, that is

$$\mathbf{x'} = \begin{cases} \mathbf{t^{(2i-n-2)(w+h)}} & \text{if } i > r \text{ and } M_i \text{ is the R.S.P;} \\ \mathbf{t^{2(i-r)(w+h)}} + d_{sync}(2i-2r-1) & \text{if } i > r \text{ and } M_i \text{ is the L.S.P;} \\ \mathbf{t^{(n-2i)(w+h)}} & \text{if } i \le r \text{ and } M_i \text{ is the L.S.P;} \\ \mathbf{t^{2(r-i+1)(w+h)}} + d_{sync}(2r-2i+1) & \text{if } i \le r \text{ and } M_i \text{ is the R.S.P.} \end{cases} \tag{13}$$

Note that the size of K does not affect the timeout periods at RPs on the CSD segment.

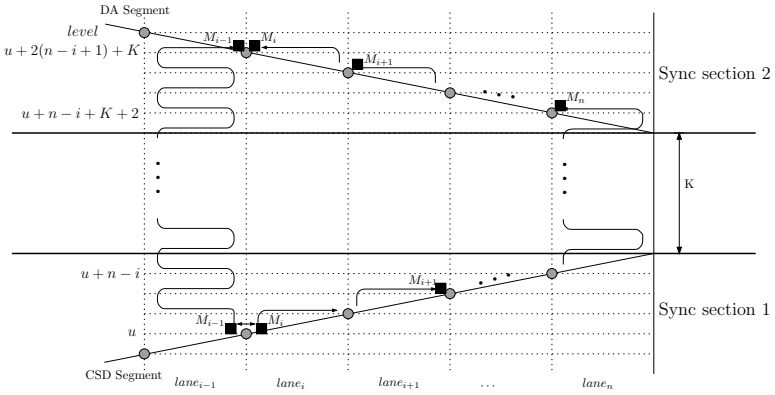


Fig. 12. Illustration for timeout period calculation for the right group

V. NUMERICAL ANALYSIS OF THE TOTAL SEARCH TIME

In this section, we present a mathematical model to approximate the total search time of each scheme in the presence of no MSN failure. The performance of each scheme in the presence of one MSN failure is also evaluated through simulation.

In order to estimate the search time of MSNs, two factors that contribute to the search time are considered: the movement time of MSNs and the synchronization overhead delay. To calculate the movement time, we first obtain movement time distribution for a distance d for some value of d, and use this distribution to estimate the distribution of the movement time for a longer distance. On the other hand, the way of estimating the synchronization delay depends on the scheme used. The final result is the mean of the total search time including the movement time and synchronization delay.

Note that, in order to provide a full sensing coverage, it is sufficient for a MSN to move to a position r away from its border lines. However, to simplify our analysis, we assume that w is much larger than r and hence, each MSN needs to move exactly w units each time it moves horizontally from one of its border line to the other border line.

A. Case of No MSN Failure

1) Alternating Column Synchronization: Let random variable $\mathbf{t}$ represent the movement time of an MSN for a distance d for some value of d, and have the probability density function (p.d.f) $f_t(t)$. In ACS, an MSN has an RP at every level, and the distance between two consecutive RPs is $w+h$ (recall that w and h are the width of a lane and the height of a level respectively). Let random variable $\mathbf{t}'$ be an MSN's movement time for the distance of $w+h$ and let $m = \frac{w+h}{d}$, then the path can be partitioned into m segments of d units each. Accordingly,

$$\mathbf{t}' = \mathbf{t_1} + \mathbf{t_2} + \cdots + \mathbf{t_m} \tag{14}$$

The p.d.f of $\mathbf{t}'$ is the convolution of $f_{t_i}(t_i)$ where $1 \leq i \leq m$.

$$f_{t'}(t') = f_{t_1}(t') * f_{t_2}(t') * \cdots * f_{t_m}(t') \tag{15}$$

In order to obtain the movement time distribution of the search team for the distance $w+h$, the movement time distribution of each individual MSN and the synchronization delay are considered. More specifically, every MSN has to synchronize with one of its neighbors at every level. A slow MSN keeps its neighbors from moving beyond their respective RPs. Therefore the speed of the slowest MSN at a level will determine the time taken for the search team to finish the level. We calculate the expected speed of the slowest MSN, and use it to estimate the total search time.

Note that the search team has n MSNs, and the movement time of each MSN at a level can be represented as a random variable $\mathbf{t}'_\mathbf{i}$ from (14) for $1 \leq i \leq n$. Further, the $\mathbf{t}'_\mathbf{i}$ is i.i.d. (independent, identically distributed). Let

random variable $\mathbf{z}$ be the slowest MSN's movement time between two levels, i.e., $\mathbf{z}$ is the n_{th} random variable in the *order statistics problem* [15].

Then the p.d.f of Z is

$$f_z(z) = nF_{t'}^{n-1}(z)f_{t'}(z) \tag{16}$$

and, the average movement time of the slowest node at a level is

$$E[\mathbf{z}] = \int_{-\infty}^{\infty} zf_z(z)dz \tag{17}$$

$$= \int_{-\infty}^{\infty} nzF_{t'}^{n-1}(z)f_{t'}(z)dz \tag{18}$$

According to the Central Limit Theorem, $f_{t'}(t')$ in (15) approaches a normal distribution. Let μ and σ^2 be the mean and variance of $\mathbf{t}$ respectively, then the mean and variance of $\mathbf{t}'$ will be $m\mu$ and $m\sigma^2$ respectively. From this, the right hand side of (18) becomes

$$n\int_{-\infty}^{\infty} z(\int_{-\infty}^{z} \frac{1}{\sqrt{2\pi m\sigma^2}}e^{-(x-m\mu)^2/2m\sigma^2}dx)^{n-1}f_{t'}(z)dz \tag{19}$$

The approximate total search time T_{ACS} is the product of the sum of movement and sync time of the slowest node at each level and the total number of levels, L, that is,

$$T_{ACS} = (E[\mathbf{z}] + T_{sync}) * L \tag{20}$$

where T_{sync} is the synchronization delay between two MSNs at an RP. In ACS, T_{sync} only includes d_{sync} which is the time to exchange data and control signals by two neighboring MSNs at an RP. For simplicity, we assume that d_{sync} is some constant value.

2) Strict Line Synchronization: Let random variable $\mathbf{s}$ be an MSN's movement time for a section (recall that a section is defined as the area between two consecutive sync lines) and f_s be the p.d.f of $\mathbf{s}$. The distance between two sync lines is $p = hK + w(K+1)$. Let $m' = \frac{p}{d}$, then we have, similar to (14),

$$\mathbf{s} = \mathbf{t_1} + \mathbf{t_2} + \cdots + \mathbf{t_{m'}} \tag{21}$$

The synchronization completion time at a sync line depends on the arrival of the slowest MSN. In other words, MSNs that have arrived earlier at a sync line wait for the slowest MSN before they complete the synchronization. From this observation, we first calculate the expected arrival time of the slowest MSN, and then calculate the expected synchronization delay after the arrival of the slowest MSN.

Let random variable $\mathbf{z}'$ represent the arrival time of the slowest MSN. The expected value of $\mathbf{z}'$ is, similar to (18),

$$E[\mathbf{z}'] = n\int_{-\infty}^{\infty} z'F_s(z')^{n-1}f_s(z')dz' \tag{22}$$

At a sync line, an MSN needs to move a distance of $(w - R)$ to communicate with one of its neighbor. Let random variable $\mathbf{u}$ represent the movement time for the distance of $(w-R)$, and let $m'' = \frac{w-R}{d}$. Then, the expected value of $\mathbf{u}$, $E[\mathbf{u}]$ can be calculated simply based on its p.d.f.

For simplicity we assume that all MSNs start a search simultaneously at the previous sync line and hence, they have an equal probability, $\frac{1}{n}$, of being the slowest. If M_1 or M_n is the slowest, after it arrives at the sync line, the data has to be forwarded to the center through $\frac{n}{2}$ lanes. If a center center ($M_{\frac{n}{2}}$ or $M_{\frac{n}{2}+1}$ assuming there are even number of MSNs) is the slowest, the data has to be forwarded through only one lane. Thus the expected number of lanes through which the data from the slowest MSN has to be forwarded at a sync line is $\sum_{i=1}^{\frac{n}{2}} \frac{1}{n}2i = \frac{n+2}{4}$. In the control signal dissemination phase, the control signal from the center MSN is always forwarded through $\frac{n}{2} - 1$ lanes. Therefore, the average synchronization delay T_{sync} at a sync line is

$$T_{sync} = (\frac{n+2}{4} + \frac{n}{2} - 1)(E[\mathbf{u}] + d_{sync}) \quad (23)$$

$$= \frac{3n-2}{4}(E[\mathbf{u}] + d_{sync}) \quad (24)$$

The approximate total search time T_{sls} is the product of the expected delay on a section and the number of sections.

$$T_{SLS} = (E[\mathbf{z}'] + T_{sync}) * \frac{L}{K} \quad (25)$$

3) X Synchronization: Suppose two center MSNs meet at a DA point at $level\ j$, and complete the DA phase. From this moment to the completion time of the next DA phase, as shown 13, three processes should be completed by both the left and the right group. First, the control signal is forwarded to the M_1 (or M_n) at $level\ (j + \frac{n}{2} - 1)$. Then, M_1 (or M_n) performs a search until it reaches its next RP at $level\ (j + \frac{n}{2} + K + 1)$. Finally, data are forwarded to the center MSNs, and the center MSNs meet at the next DA point at $level\ (j + n + K)$. Therefore, the search time for a d-section will be the arrival time (at the next DA point) of the slowest center MSN.

For the completion of the CSD phase and the DA phase, the data or the control signals are forwarded through $n - 2$ RPs. The distance between two RPs is $w + h$, and an MSN needs to move for $w + h - R$ to communicate its sync peer. Therefore, the total travel distance for CSD and DA is $(n - 2)(w + h - R)$. For the search of M_1 (or M_n), M_1 (or M_n) travels for a distance of $(K + 2)(w + h) - R$. Then, the sum of the travel distance for CSD and DA and the distance for the search of M_1 (or M_n), P, is

$$P = (n - 2)(w + h - R) + (K + 2)(w + h) - R \quad (26)$$

Let random variable $\mathbf{u}$ be an MSN's travel time for the distance of P. $\mathbf{u}$ can be represented as the sum of $\mathbf{s}$. Also let $\mathbf{v}$ be the a center MSN's arrival time at the next DA point. Then, $\mathbf{v}$ is

$$\mathbf{v} = \mathbf{u} + d_{sync} \times (n - 2) \quad (27)$$

Note that the right group and the left group independently perform the three processes. Therefore, DA can be completed after both the left center MSN and the right center MSN arrive at the DA point.

Also let $\mathbf{y}$ the arrival time of the slowest center MSN between the center MSNs. Then, we have,

$$\mathbf{y} = max(\mathbf{v}, \mathbf{v}) \quad (28)$$

Note that the p.d.f of $\mathbf{v}$, $f_v(v)$, can be obtained from the p.d.f of $\mathbf{u}$ and (27). Also let $f_y(y)$ be the p.d.f of $\mathbf{y}$. Then, from (28), we have

$$f_y(y) = 2F_v(y)f_v(y) \quad (29)$$

Let $C = \frac{1}{\sqrt{2\pi m\sigma^2}}$, then the expected arrival time of the slowest center MSN is

$$E[y] = \int_{-\infty}^{\infty} y f_y(y) dy \quad (30)$$

$$= 2\int_{-\infty}^{\infty} y F_v(y) f_v(y) dy \quad (31)$$

$$= 2C^2 \int_{-\infty}^{\infty} y(\int_{-\infty}^{y} e^{-\frac{(x-m\mu)^2}{2m\sigma^2}} dx) e^{-\frac{(y-m\mu)^2}{2m\sigma^2}} dy \quad (32)$$

The two center MSNs also need the time d_{sync} to exchange data. Therefore, the expected search time for a d-section is $E[y] + d_{sync}$. The expected total search time, T_{XS}, is the product of the search time for a d-section and the total number of d-sections, that is,

$$T_{XS} = (E[y] + d_{sync}) \times \frac{L}{n + K} \quad (33)$$

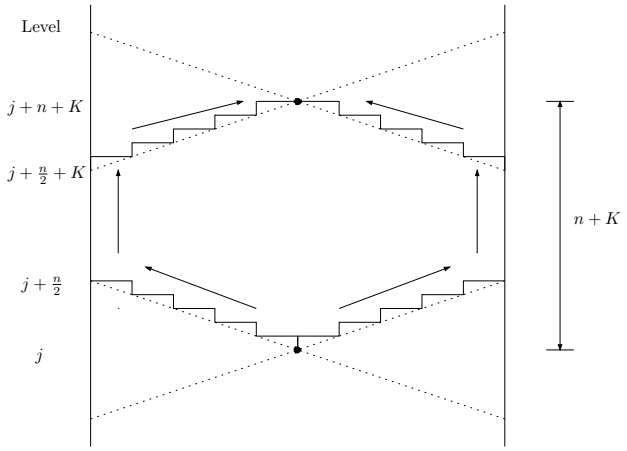


Fig. 13. Calculation of the search time in a d-section

B. Case with an MSN Failure

1) Alternating Column Scheme: Suppose one MSN out of n MSNs fails at an arbitrary $level\ f$. After the recovery of the failure, the $n-1$ MSNs perform the search. Therefore, the search area can be considered to be partitioned into two zones according to the number of MSNs performing the search. Let $Zone\ A$, and $Zone\ B$ be the area which is searched by n, and $n-1$ MSNs respectively. Let T_1^n and T_1^{n-1} be the search time of the team for one level with n MSNs and $n-1$ MSNs respectively.

Then the total search time with a failure at $level\ f$, T_f becomes

$$T_f = fT_1^n + (L-f)T_1^{n-1} \tag{34}$$

We assume that the failure can occur at any level between $level\ s$ and $level\ t$ with a uniform probability. Then, the expected search time with a failure at an arbitrary level, $\overline{T}_{XS}$ becomes

$$\overline{T}_{XS} = \frac{\sum_{f=s}^{t} E[T_f]}{s-t+1} \tag{35}$$

2) Strict Line Synchronization and X Synchronization: We describe both cases of SLS and XS, because a same way of calculation can be applied to both cases only with the difference of the size of a section. (Note that a section is used in SLS while a d-section is used in XS). Suppose a search team originally has n MSNs and K-level non-sync section, and the lane width of a lane is w. Suppose one MSN out of n MSNs fails at an arbitrary $level\ f$. During the recovery of a recovery MSN, the $n-2$ MSNs perform the search. After the recovery MSN rejoins the search team, the team performs the search with $n-1$ members. Therefore, the search area can be considered to be partitioned into three zones according to the number of MSNs performing the search. Let $Zone\ A$, $Zone\ B$, $Zone\ C$ be the area which is searched by n, $n-2$, and $n-1$ MSNs respectively. For simplicity, we assume that the search area has exactly g number of d-sections. There are two cases to be considered to calculate the the total search time.

- Case 1: After the recovery MSN rejoins the search team, the search team completes the mission;
- Case 2: Without the recovery MSN rejoining the team, other MSNs reach the end of the search area;

case1 - After the recovery MSN rejoins the search team, the search team completes the mission

Let random variable T^i represent the search time for a d-section (a section for the case of SLS) with i MSNs, and D_f be the time elapsed from the arrival of the team at a DA segment to the detection of the failure. Also let p, q be the number of sections of $Zone\ A$ and $Zone\ B$ respectively. Then the total search time, T_f becomes

$$T_f = pT^n + qT^{n-2} + (g-p-q)T^{n-1} + D_f \tag{36}$$

and the expected search time is

$$E[T_f] = pE[T^n] + qE[T^{n-2}] + (g-p-q)E[T^{n-1}] + E[D_f] \tag{37}$$

Note that $p = min\{p : p \times (n+K) \geq f\}$.

The value of q depends on when the recovery MSN rejoins other MSNs. For simplicity, we assume that both failure detection and reallocation of lanes occur at the level of DA point (For SLS, the level of a sync line). Then, the recovery MSN moves a distance of $(K+n)h$ (Kh for SLS) to go back to the last sync section. Then, for the search of the failed MSN's lane, it moves a distance of $(K+n)(w+h)$ ($K(w+h)$ for SLS). Then it moves linearly a distance of $qh(K+n-2)$ (qhK for SLS) to catch up with the search team. Let random variable R_t be the time taken for the recovery MSN to move a distance of $(K+n)h + (K+n)(w+h)$ ($Kh + K(w+h)$ for SLS) and $E[R_t]$ be the expected value of R_t. Also let R_f be the time needed for the recovery MSN to move a distance of $h(K+n-2)$ (hK for SLS). Then, we have

$$q = min\{q : qT^{n-2} \geq E[R_t] + qE[R_f]\} \tag{38}$$

Therefore,

$$q = min\{q : q \geq \frac{E[R_t]}{E[T^{n-2}] - E[R_f]}\} \tag{39}$$

Note that $E[T^n]$, $E[T^{n-1}]$, and $E[T^{n-2}]$ can be obtained using (32), and the p.d.fs of R_t and R_f can also be obtained. $E[R_t]$ and $E[R_f]$ are calculated based on their p.d.f. To estimate D_f, we simply subtract the expected search time of the team for a section from the average timeout period of MSNs, i.e., $E[D_f] = \frac{\sum_{i=1}^{n} TO_i}{n} - E[T^n]$.

Therefore, the expected search time of the team with a failure at $level\ f$ can be calculated from (37).

The failure can occur at any level between $level\ s$ and $level\ t$ with a uniform probability. Then, the expected search time with a failure at an arbitrary level, $\overline{T}_{XS}$ becomes

$$\overline{T}_{XS} = \frac{\sum_{f=s}^{t} E[T_f]}{s-t+1} \tag{40}$$

3) case2 - Without the recovery MSN rejoining the team, the team reaches the end of the search area: In this case, there are two sub cases: the case the recovery MSN first arrives at the end of the search area (or destination) and the case the remaining MSNs first arrive at the destination. If the remaining MSNs arrive at the destination earlier than the recovery MSN, the search time is the arrival time of the recovery MSN. Otherwise, the arrival time of the remaining MSNs is the search time.

Let $A = E[R_t] + (g-p)E[R_f]$ and $B = (q-p)E[T^{n-2}]$. Then, $E[T_f]$ is

$$E[T_f] = pE[T^n] + max(A, B) + E[D_f] \tag{41}$$

The expected search time with a failure at an arbitrary level can be calculated in a similar way to the $case\ 1$.

VI. NUMERICAL ANALYSIS OF THE AVERAGE TRAVEL DISTANCE

A. Case of No MSN Failure

1) Alternating Column and X Synchronization: When there is no MSN failure, the average travel distance of MSNs in ACS is equal to that of MSNs in XS, because MSNs in ACS or XS follow the exactly same path determined in the LBS unless an MSN failure occurs.

The average travel distance of MSNs can be obtained from the total travel distance of the team. At every level, the travel distance for horizontal movements of all MSNs is $w \times n$. In order to move to the next level, each MSN moves a distance of h. Thus, the total travel distance of MSNs is $wnL + h(L-1)n$. Therefore, the expected travel distance of MSNs with a failure at an arbitrary level, S_{XS} or S_{ACS} is

$$S_{XS} = S_{ACS} = L(w+h) - h \tag{42}$$

2) Strict Line Synchronization: An MSN moves an average distance of $K(w+h)$ for a search a section. On a sync line the MSN moves a distance of $2(w-R)$ for data aggregation and control signal dissemination. Therefore, the distance S an MSN travels for a section is

$$S = K(w+h) + 2(w-R) \tag{43}$$

Then, the average travel distance of MSNs, S_{SLS} for the search area is

$$S_{SLS} = (K(w+h) + 2(w-R))\frac{L}{K} \tag{44}$$

B. Case with an MSN Failure

1) Alternating Column Synchronization: Suppose one MSN fails at $level\ f$. Before the failure detection, the total travel distance of the MSNs is $nf(w+h)$. After the failure detection, the total travel distance is $(n-1)(L-f)(\frac{wn}{n-1}+h)$. (Note that the width of a lane becomes larger).

Therefore, the average travel distance of MSNs with a failure at $level\ f$, S_f is

$$S_f = f(w+h) + \frac{n-1}{n}(L-f)(\frac{wn}{n-1}+h) \tag{45}$$

Then, the expected distance of MSNs with a failure at an arbitrary level between s and t, S_{ACS}, becomes

$$S_{ACS} = \frac{\sum_{f=s}^{t} S_f}{n(s-t+1)} \tag{46}$$

2) Strict Line Synchronization: In order to calculate the average distance of MSNs, we first calculate the total distance of MSNs. Let S^i be the total travel distance of a team with i MSNs for searching a d-section. Then, we have

$$S^i = i(K(w+h) + 2(w-R)) \tag{47}$$

Let D_A, D_B, and D_C represent the travel distance of MSNs for the search for $Zone\ A$, $Zone\ B$, $Zone\ C$ respectively.

The failure of MSN occurs in $Zone\ A$. The total travel distance of MSNs in $Zone\ A$, D_A is

$$D_A = n(K(w+h) + 2(w-R)) - (pK-f)(w+h) \tag{48}$$

$$\tag{49}$$

Only $n-2$ MSNs perform a search for $Zone\ B$. Therefore,

$$D_B = q(n-2)(K(\frac{wn}{n-2}+h) + 2(\frac{wn}{n-2}-R)) \tag{50}$$

The travel distance for the search of $zone\ C$ is

$$D_C = (g-p-q)(n-1)(K(\frac{wn}{n-1}+h) + 2(\frac{wn}{n-1}-R)) \tag{51}$$

The recovery MSN goes back to the last sync section for a recovery and then catches up with the team with a linear movement. Therefore, the distance the recovery MSN moves from its starting the recovery to rejoining the team, D_R is

$$D_R = K(w+h) + hK + qhK \tag{52}$$

$$= K(w+h(2+q)) \tag{53}$$

Then, the total distance, S_f is the sum of D_A, D_B, and D_C. That is,

$$S_f = D_A + D_B + D_C + D_R \tag{54}$$

The expected average distance of MSNs with a failure at an arbitrary level, S_{SLS}, becomes

$$S_{SLS} = \frac{\sum_{f=s}^{t} S_f}{n(s-t+1)} \tag{55}$$

3) X Synchronization: Let S^i be the total travel distance of a team with i MSNs for searching a section. Then, we have

$$S^i = iK(w+h) \tag{56}$$

Let D_A, D_B, and D_C be the travel distance of MSNs for searching $Zone\ A$, $Zone\ B$, $Zone\ C$ respectively. An MSN fails in $Zone\ A$. Therefore, D_A is

$$\begin{aligned} D_A &= np(K+n)(w+h) - (p(K+n)-f)(w+h) && (57)\\ &= (w+h)(p(K+n)(n-1)+f) && (58)\end{aligned}$$

In $Zone\ B$, the team has $n-2$ MSNs. Therefore,

$$D_B = q(n-2)(K+n-2)(\frac{wn}{n-2}+h) \tag{59}$$

Similarly, the distance for the search of $zone\ C$ is

$$D_C = (g-p-q)(n-1)(K+n-1)(\frac{wn}{n-1}+h) \tag{60}$$

The distance the recovery MSN moves from its starting the recovery to rejoining the team, D_R is

$$\begin{aligned} D_R &= (w+h)(K+n) + h(K+n) + qh(K+n) && (61)\\ &= (K+n)(w+h(2+q)) && (62)\end{aligned}$$

Then, the total distance, S_f is the sum of D_A, D_B, and D_C. That is,

$$S_f = D_A + D_B + D_C + D_R \tag{63}$$

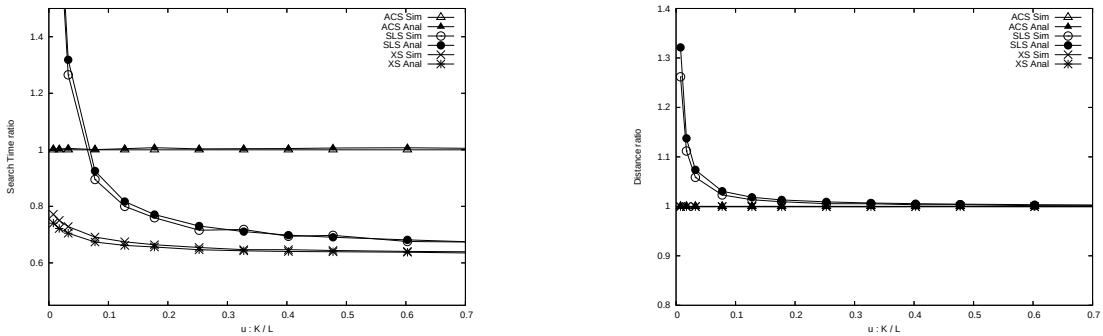
The expected average distance of MSNs with a failure at an arbitrary level, S_{XS}, can be obtained in a similar way to (46).

VII. Numerical Results

In this section, we perform simulations to evaluate the two synchronization schemes by varying the size of K and the number of MSN failures. The size of the search area is 10000m x 40000m. Each MSN has a wireless transmission range of 200m, and a sensing range of 50m, which results in $h = 100m$ and $L = 400$. (recall that L is the total number of levels). The speed variation of MSNs follows Gauss-Markov process model [16] which can represent a large range of mobility patterns from the random-walk to the constant velocity fluid-flow models. The initial speed of MSNs was set to 30m/s. The search team has 18 MSN members as a default value. The timeout period for a sync peer was set to the amount of time within which the sync peer arrives at the next RP with the probability of 0.999. The average travel distance of MSNs and the total search time i.e., the time elapsed between the departure of the search team and the completion of the search operation of the last MSN, were chosen as the performance metric.

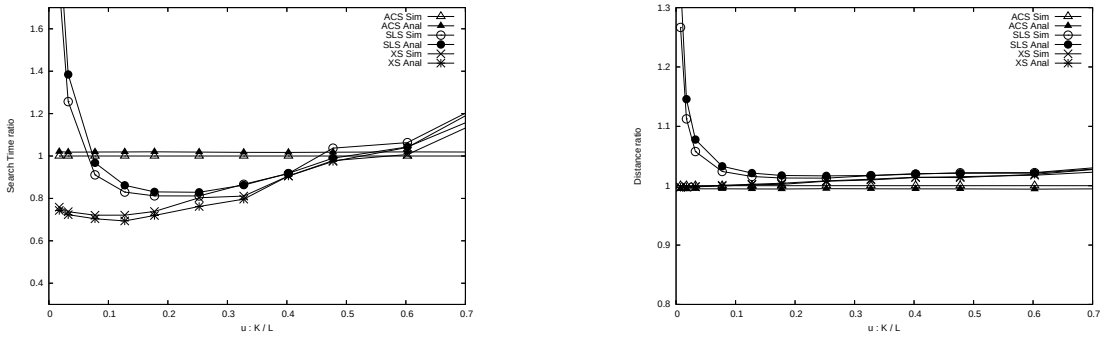
In Fig. 14 and Fig. 15, the Y axis represents the normalized search time when the total search time of ACS (obtained via simulation) is set to 1, and the X axis represents u (the ratio of K to L) used for SLS and XS(as such, u does not affect the search time for ACS).

As shown in Fig. 14 and Fig. 15, our numerical estimation of the performance of the synchronization schemes closely approximates the simulation results in the presence of no MSN failure and an MSN failure. Note that the ratio u has a significant effect on the search time of SLS. Initially, the search time of SLS decreases rapidly as u grows. However, the rate of the decrease of the search time becomes smaller and smaller as u grows. Specifically, the search time of SLS decreases with u due to a fewer and fewer sync operations, although the rate of the decrease becomes smaller and smaller because when u approaches 1, the search time approaches its lower bound which is the moving time of the MSNs. As u grows, the search time of XS also decreases, because a large value of u (or K) results infrequent synchronization. A larger value of u, however, will require a longer response time of search team against an event and a larger buffer size of MSNs.



(a) Search Time (b) Avg. Distance

Fig. 14. Varying size of K without a MSN failure



(a) Search Time (b) Avg. Distance

Fig. 15. Varying size of K with a MSN failure

As shown in Fig. 14(b), ACS and XS have the same average travel distance due to the exactly same path determined in the LBS when there is no MSN failure. The average travel distance of SLS is larger than that of ACS or XS, because of extra movements for synchronization.

Fig. 15 shows the results of the schemes in the presence of a failure. We assume that a failure can occur at any time, except at the beginning and the end of the search area during the search operation with a uniform probability. In this case, the search time of SLS decreases with small u initially as in the case of no failure, but as u grows larger than about 0.2, the search time of SLS increases, eventually exceeding the search time of ACS. The search time of XS decreases as u grows until u reaches about 0.12. As u grows beyond 0.12, however, the search time of XS increases, and also becomes larger than that of ACS. This result indicates that a large u (or K) also has a negative effect on the search time in the case of MSN failures. More specifically, when a failure occurs at an arbitrary position in the search area, the team detects the failure at the next synchronization line and the recovery node has to go back up to K levels while the remaining $n-2$ MSNs have to cover n lanes until the recovery node rejoins the team. It takes the recovery node a longer time a with larger K to rejoin the team, which results in a larger search time. In particular, the case $e = 1$ (or $K = L$) may be considered as equivalent to the case where each MSN independently searches its lane until it reaches the end of the lane. Clearly, if an MSN failed, then one of the remaining MSNs has to go back up to L levels and the search will not complete until that MSN returns back. Therefore, there exists an optimal range of u for minimizing the search time of SLS or XS.

As shown in Fig. 15(b), the MSN failure does not affect travel distance of MSNs as much as it affects the search time.

VIII. Conclusion and Future works

In this paper, we have studied search algorithms using autonomous MSNs. we first have proposed a load-balancing strategy called Lane Based Search (LBS). We then proposed three synchronization schemes, called Alternating Column Synchronization (ACS), Strict-Line Synchronization (SLS) schemes, and X Synchronization (XS). Through numerical analysis and simulation results we have shown that XS has a better performance in terms of the total search time and average travel distance with an appropriate K. Many other problems including cooperative transmission for communication remain open.

References

[1] A. Howard, M. J. Mataric, and G. S. Sukhatme, "An incremental deployment algorithm for mobile robot teams," in *Proceedings of IEEE/RSJ International Conference on Robotics Systems (IROS)*, October 2002.

[2] G. Wang, G. Cao, and T. L. Porta, "Movement-assisted sensor deployment," in *IEEE INFOCOM Conference Proceedings*, March 2004.

[3] J. Wu and S. Yang, "Smart: A scan-based movement-assisted sensor deployment method in wireless sensor networks," in *IEEE INFOCOM Conference Proceedings*, March 2005.

[4] B. Liu, P. B. O. Dousse, P. Nain, and D. Towsley, "Mobility improves coverage of sensor networks," in *ACM MobiHoc*, 2005.

[5] G. Wang, G. Cao, T. L. Porta, and W. Zhang, "Sensor relocation in mobile sensor networks," in *IEEE INFOCOM Conference Proceedings*, March 2005.

[6] D. Henkel, C. Dixon, J. Elston, and T. X. Brown, "A reliable sensor data collection network using unmanned aircraft," in *Proceedings of the second international workshop on Multi-hop ad hoc networks: from theory to reality (REALMAN 2006)*, Florence, Italy.

[7] (2005, October) Unmanned aircraft systems roadmap 2005-2030. [Online]. Available: http://www.fas.org/irp/program/collect/uav_roadmap2005.pdf

[8] R. Behringer, W. Travis, R. Daily, D. Bevly, W. Kubinger, and W. H. V. Fehlberg, "Rascal - an autonomous ground vehicle for desert driving in the darpa grand challenge 2005," in *Poceedings of IEEE Intelligent Transportation Systems*, September 2005, pp. 644–649.

[9] A. Bacha, C. Reinholtz, A. Wicks, M. Fleming, A. Naik, and M. A. N. Elder, "The darpa grand challenge: overview of the virginia tech vehicle and experience," in *Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems*, 2005.

[10] T. Balch and R. C. Arkin, "Behavior-based formation control for multiagent robot teams," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 926–939, December 1998.

[11] X. Cui, T. Hardin, R. K. Ragade, , and A. S. Elmaghraby, "A swarm-based fuzzy logic control mobile sensor network for hazardous contaminants localization," in *The IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2004)*, FortLauderdale, Florida, U.S.

[12] D. Payton, M. Daily, R. Estowski, M. Howard, and C. Lee, "Pheromone robotics," *Autonomous Robots*, vol. 11, no. 3, pp. 319–324, November 2001.

[13] P. E. Rybski, S. Stoeter, M. D. Erickson, M. L. Gini, D. F. Hougen, and N. Papanikolopoulos, "A team of robotic agents for surveillance," in *Fourth International Conference on Autonomous Agents (Agents 2000)*.

[14] M. A. Lewis and G. A. Bekey, "The behavioral self-organization of nanorobots using local rules," in *Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Raleigh, NC, July 1992.

[15] A. Papoulis and S. U. Pillai, *Probability, Random Variables and Stochastic Processes 4th Edition*. McGraw-Hill, pp. 245–246.

[16] B. Liang and Z. J. Haas, "Predictive distance-based mobility management for multidimensional pcs networks," *IEEE/ACM Transcations on Networking*, vol. 11, no. 5, pp. 718 – 732, October 2003.