# Paper Title*

*Note: Sub-titles are not captured in Xplore and should not be used


1st Given Name Surname
*dept. name of organization (of Aff.)*
*name of organization (of Aff.)*
City, Country
email address

2nd Given Name Surname
*dept. name of organization (of Aff.)*
*name of organization (of Aff.)*
City, Country
email address

3rd Given Name Surname
*dept. name of organization (of Aff.)*
*name of organization (of Aff.)*
City, Country
email address

4th Given Name Surname
*dept. name of organization (of Aff.)*
*name of organization (of Aff.)*
City, Country
email address

5th Given Name Surname
*dept. name of organization (of Aff.)*
*name of organization (of Aff.)*
City, Country
email address

6th Given Name Surname
*dept. name of organization (of Aff.)*
*name of organization (of Aff.)*
City, Country
email address



***Abstract*—This document is a model and instructions for LaTeX. This and the IEEEtran.cls file define the components of your paper [title, text, heads, etc.]. *CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.**



***Index Terms*—component, formatting, style, styling, insert**


## I. Introduction

This document is a model and instructions for LaTeX. Please observe the conference page limits.

## II. Ease of Use

### A. Maintaining the Integrity of the Specifications

The IEEEtran class file is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

## III. Prepare Your Paper Before Styling

Before you begin to format your paper, first write and save the content as a separate text file. Complete all content and organizational editing before formatting. Please note sections III-A–III-E below for more information on proofreading, spelling and grammar.

Keep your text and graphic files separate until after the text has been formatted and styled. Do not number text heads—LaTeX will do that for you.


Identify applicable funding agency here. If none, delete this.


### A. Abbreviations and Acronyms

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, ac, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

### B. Units

- Use either SI (MKS) or CGS as primary units. (SI units are encouraged.) English units may be used as secondary units (in parentheses). An exception would be the use of English units as identifiers in trade, such as “3.5-inch disk drive”.
- Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.
- Do not mix complete spellings and abbreviations of units: “Wb/$m^2$” or “webers per square meter”, not “webers/$m^2$”. Spell out units when they appear in text: “. . . a few henries”, not “. . . a few H”.
- Use a zero before decimal points: “0.25”, not “.25”. Use “$cm^3$”, not “cc”.)

### C. Equations

Number equations consecutively. To make your equations more compact, you may use the solidus ( / ), the exp function, or appropriate exponents. Italicize Roman symbols for quantities and variables, but not Greek symbols. Use a long dash rather than a hyphen for a minus sign. Punctuate equations with commas or periods when they are part of a sentence, as in:

$$a + b = \gamma \tag{1}$$

Be sure that the symbols in your equation have been defined before or immediately following the equation. Use “(1)”, not

"Eq. (1)" or "equation (1)", except at the beginning of a sentence: "Equation (1) is . . ."

### D. LaTeX-Specific Advice

Please use "soft" (e.g., `\eqref{Eq}`) cross references instead of "hard" references (e.g., `(1)`). That will make it possible to combine sections, add equations, or change the order of figures or citations without having to go through the file line by line.

Please don't use the `{eqnarray}` equation environment. Use `{align}` or `{IEEEeqnarray}` instead. The `{eqnarray}` environment leaves unsightly spaces around relation symbols.

Please note that the `{subequations}` environment in LaTeX will increment the main equation counter even when there are no equation numbers displayed. If you forget that, you might write an article in which the equation numbers skip from (17) to (20), causing the copy editors to wonder if you've discovered a new method of counting.

BibTeX does not work by magic. It doesn't get the bibliographic data from thin air but from .bib files. If you use BibTeX to produce a bibliography you must send the .bib files.

LaTeX can't read your mind. If you assign the same label to a subsubsection and a table, you might find that Table I has been cross referenced as Table IV-B3.

LaTeX does not have precognitive abilities. If you put a `\label` command before the command that updates the counter it's supposed to be using, the label will pick up the last counter to be cross referenced instead. In particular, a `\label` command should not go before the caption of a figure or a table.

Do not use `\nonumber` inside the `{array}` environment. It will not stop equation numbers inside `{array}` (there won't be any anyway) and it might stop a wanted equation number in the surrounding equation.

### E. Some Common Mistakes

- The word "data" is plural, not singular.
- The subscript for the permeability of vacuum $\mu_0$, and other common scientific constants, is zero with subscript formatting, not a lowercase letter "o".
- In American English, commas, semicolons, periods, question and exclamation marks are located within quotation marks only when a complete thought or name is cited, such as a title or full quotation. When quotation marks are used, instead of a bold or italic typeface, to highlight a word or phrase, punctuation should appear outside of the quotation marks. A parenthetical phrase or statement at the end of a sentence is punctuated outside of the closing parenthesis (like this). (A parenthetical sentence is punctuated within the parentheses.)
- A graph within a graph is an "inset", not an "insert". The word alternatively is preferred to the word "alternately" (unless you really mean something that alternates).
- Do not use the word "essentially" to mean "approximately" or "effectively".
- In your paper title, if the words "that uses" can accurately replace the word "using", capitalize the "u"; if not, keep using lower-cased.
- Be aware of the different meanings of the homophones "affect" and "effect", "complement" and "compliment", "discreet" and "discrete", "principal" and "principle".
- Do not confuse "imply" and "infer".
- The prefix "non" is not a word; it should be joined to the word it modifies, usually without a hyphen.
- There is no period after the "et" in the Latin abbreviation "et al.".
- The abbreviation "i.e." means "that is", and the abbreviation "e.g." means "for example".

An excellent style manual for science writers is [7].

### F. Authors and Affiliations

**The class file is designed for, but not limited to, six authors.** A minimum of one author is required for all conference articles. Author names should be listed starting from left to right and then moving down to the next line. This is the author sequence that will be used in future citations and by indexing services. Names should not be listed in columns nor group by affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization).

### G. Identify the Headings

Headings, or heads, are organizational devices that guide the reader through your paper. There are two types: component heads and text heads.

Component heads identify the different components of your paper and are not topically subordinate to each other. Examples include Acknowledgments and References and, for these, the correct style to use is "Heading 5". Use "figure caption" for your Figure captions, and "table head" for your table title. Run-in heads, such as "Abstract", will require you to apply a style (in this case, italic) in addition to the style provided by the drop down menu to differentiate the head from the text.

Text heads organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced.

### H. Figures and Tables

*a) Positioning Figures and Tables:* Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation "Fig. 1", even at the beginning of a sentence.

TABLE I
TABLE TYPE STYLES

| Table Head | Table Column Head | | |
|---|---|---|---|
| | *Table column subhead* | *Subhead* | *Subhead* |
| copy | More table copy[a] | | |

[a]Sample of a Table footnote.

Fig. 1. Example of a figure caption.

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity "Magnetization", or "Magnetization, M", not just "M". If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write "Magnetization (A/m)" or "Magnetization {A[m(1)]}", not just "A/m". Do not label axes with a ratio of quantities and units. For example, write "Temperature (K)", not "Temperature/K".

## IV. BLACK VIRUS DISINFECTION IN TRIPLE LOOPS

### A. Introduction

In this chapter, we discuss parallel strategy on BVD problem in chordal ring. A chordal ring is a circulant graph with $d_1 = 1$, i.e., it is an augmented ring, and will be denoted by $C_n(1, d_2, ..., d_k)$. More specifically, in chordal ring each node is directly connected to the nodes at distance $d_i$ and $n - d_i$ by additional links called chords. The link connecting two nodes is labeled by the distance that separate these two nodes on the ring. For convenience, if we say the agents or the clones move along chord $i$, then they actually move along $d_i$. If we say the agents or the clones move $i$, then they actually move along chord $d_x$ with its length equal to $i$. Let us donated by $d$ the half degree of the chordal ring (for example, for the chordal ring structure $C_n(1, 2, 4, 5)$, $d = 4$), by $l$ the length of the longest chord of the chordal ring (for example, for the chordal ring structure $C_n(1, 2, 4, 5)$, $l = 5$). In order to simplify the process, we assume that all the nodes in the network are marked with a number: the starting point is marked 0, then the second node is marked 1... Our goal is to minimize the time to complete the whole decontamination process and at the same time the casualties (number of agents destroyed by the BV) In order to do that, we propose parallel strategy for decontaminating the chordal ring. In the elimination phase, we propose two strategies to surround the neighbours of the clones sequentially and parallelly. In the parallel case, an tricky situation might happen: supposing that the sites of the two clones are connected, in this case, after these two clones are triggered, one of their clones spread to another sites and since the agents sent to destroy them die, these two sites are empty when the second round clones arrive, which make our decontamination invalid. In order to solve this problem, we make an assumption when we talk about parallel strategy in elimination phase in chordal rings: when a BV is trigger at $T_i$, it take negligible time for its clones to move to all its neighbours, for example, at $T_i$.

### B. Shadowed Exploration

**Initialization** The chordal ring is a complete symmetrical structure, so we can randomly choose a node $x_0$ as the start node. Initially, we place $2d$ agents at node 0, 1, ..., $2d - 1$ (first round). Agents residing in nodes from 0 to $d - 1$ are in shadowing group, while from $d$ to $2d - 1$ are in exploring group. If none of the agent is destroyed, we then place $d$ agents at nodes 0,1, ..., $d - 1$ (second round). If not, we can easily know the location of the BV and employ agents to surround the new formed BVs, then destroy them permanently. Only the agents employed in the first round move in the exploring phase. The agents employed in the second round remain dormant, guarding the nodes to guarantee the monotone.

**Route of the agent in exploring phase** We separate the time of exploring phase into two part: moving time and notice time. In the whole phase of exploring phase, they are arranged as below: $T_{move\_1}$, $T_{notice\_1}$, $T_{notice\_1'}$, $T_{notice\_1''}$, $T_{move\_2}$, $T_{notice\_2}$, $T_{notice\_2'}$, $T_{notice\_2''}$ ...More specifically, every cycle contains one unit of time for moving and three units of time for notice. We discuss why we arrange the time as above and what exactly the agents do in the notice time later.

In chord ring $C_n(1, d_2, \ldots, d_k)$, all the agents in the array move along their longest chord $d_k$ in $T_{move\_i}$. That is, agents move along $d_k$ in $T = 1 + 4t$ $(t \in \mathbb{N})$. An example of how agents move in chordal ring $C_n(1, 2, 4, 5)$ at $T_{move\_i}$ in exploring phase is shown in figure **??**. For our convenience, in some case, we consider the chordal ring as arranged in rows of $d_k$ where the last node of a row is connected to the first node of the following row and the last node is connected to the first. Depending on the size of the chordal, the last row could be incomplete. So in this matrix, moving down a column corresponding to using the longest chord $d_k$. In the matrix, we also mark the number of nodes.

**Three Jump NotificationTechnique** In the model of BV decontamination exploring the network sequentially, explore agent moves forward for one step. If the node is safe, it move back to the leader agent, and then move forward to the safe


(a) Arrangement of agents at $T_{move\_i}$

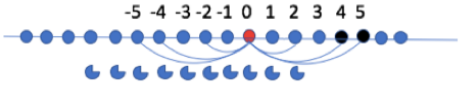

(b) Arrangement of agents at $T_{move\_i+1}$

Fig. 2. Arrangement of agents when moving

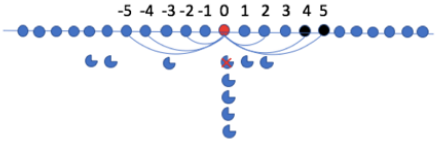
node together. In this case, if the leader agent does not meet the leader in next $T$ after it moves forward, the leader agent learns that the original BV resides in the next node so it stop moving. But in our model, we employ $2d$ agents in the exploring phase, if they are not properly informed when one agent is destroyed by BV, in their next step of moving, some of them may be destroyed by the new formed BVs. In order to avoid the casualties, we propose $Three\,Jump\,Notification\,Technique$ to properly notice the agents who will move to the new formed BVs. For convenience, let us take the node where the original BV resides as the original of the one-dimensional coordinate system. In a chordal ring $C_n(1, d_2, \ldots, d_k)$, if the original BV is triggered, the clones of it spread to nodes whose coordinates are $-d_k$, $-d_{k-1}$, …, $-d_2$, $-1$, $1$, $d_2$, …, $d_{k-1}$, $d_k$. Obviously the nodes whose coordinates are $1$, $d_2$, …, $d_{k-1}$, $d_k$ may become the BV (Possible BV Nodes) now, our goal is to notify the agents which will move to these nodes to stop (Risky Agent). The coordinates of them are $1 - d_k$, $d_2 - d_k$, …, $d_{k-1} - d_k$, $d_k - d_k$ (which is exactly the coordinate of the original BV) respectively. It is obvious that not all of the nodes from $1$ to $d_k$ become BV nodes after the triggering because there might be some agents already there, but since notifying all the $Risky\,Agents$(RAs) does not add more cost comparing to notifying some of them, in our strategy, we notify all of the RAs. Let us donate one of the Possible BV Nodes by $d_i$, and in our $Three\,Jump\,Notification\,Technique$, agent residing in node $-d_i$ (Notification Agent) is employed to notify the agent residing in node $-d_k + |d_i|$ (RA) who will move to the BV node.

We show that $Notification\,Agent$ are able to meet $Risky\,Agent$ in three steps: $-d_i \xrightarrow{move|d_i|} 0 \xrightarrow{move\,along\,chord\,k} -d_k \xrightarrow{move\,|d_i|} -d_k + |d_i|$. In this case, the notifying route of the $Notification\,Agent$ whose coordinate is $-d_k$ is $-d_k \rightarrow 0 \rightarrow -d_k \rightarrow 0$. We would make some modification in the $Surrounding\,and\,Elimination$ phase, but now let us assume it still follow the route above. The whole process of $Three\,Jump\,Notification\,Technique$ in chordal ring $C_n(1, 2, 4, 5)$ is shown in figure **??**.

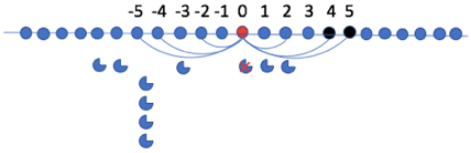
If the original BV residing in the red node, then once an agent moves to it, the agent and the BV are destroyed but the clones of the BV spread to all its neighbours. According to our technique, nodes whose coordinates are 1, 2, 4, 5 become



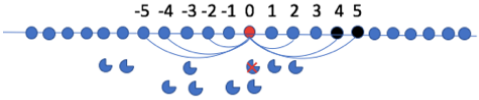

(a) Arrangement of agents at $T_{move_i}$




(b) Arrangement of agents at $T_{notification_i}$




(c) Arrangement of agents at $T_{notification'_i}$




(d) Arrangement of agents at $T_{notification''_i}$

Fig. 3. The whole process of the Three Notifying Technique in chordal ring $C_n(1, 2, 4, 5)$

$Possible\,BV\,Nodes$; agents residing in nodes $-4$, $-3$, $-1$, $0$ are the $Risky\,Agents$; agents residing in nodes $-5$, $-4$, $-2$, $-1$ are the $Notice\,Agents$. The routes for agents residing in nodes $-5$, $-4$, $-2$, $-1$ are $-5 \rightarrow 0 \rightarrow -5 \rightarrow 0$; $-4 \rightarrow 0 \rightarrow -5 \rightarrow -1$; $-2 \rightarrow 0 \rightarrow -5 \rightarrow -3$; $-1 \rightarrow 0 \rightarrow -5 \rightarrow -4$ respectively.

**Safe Exploring with Three Jump Notifying Technique**

After the initialization, agents employed in the first round move as the route we introduced in "Initialization" . When one of the agents is destroyed at $T - move\_i$, then $Three\,Jump\,Notification\,Technique$ begins. In "Route of the agent in exploring phase", we say that in the exploring phase every cycle contains one unit of time for moving and three units of time for notifying. Actually, the three units of time for notice are reserved for $Three\,Jump\,Notice\,Technique$, even though it only executes when one of the agents is destroyed. In another word, before encountering a BV, all the agents just stay where they are in the notice time. After executing the $Three\,Jump\,Notice\,Technique$, the $Notice\,Agents$ move back to where they are before the notification. For example, in the example in "Three Jump Notification Technique", Notification Agent residing in node $-1$ moves back to node $-4$ following the reverse route in the notice phase which is $-1 \rightarrow -5 \rightarrow 0 \rightarrow -4$.

### C. Surrounding and Elimination

In this section, we introduce the process of eliminating the BVs after the original BV is triggered. For

the purpose of saving the number of agents, we prefer to chase the $Keep\,Moving$ agents, but it is not necessary to complete the process, for example, you can carry enough number of agents so you can proceed the $Surrounding\,and\,Elimination$ immediately. As we mentioned before, we first chase the $Keep\,Moving$ agents then begin the $Surrounding\,and\,Elimination$. We propose two methods: the first one is to surround the new formed BVs sequentially, and the second one is to surround the new formed BVs parallelly. The former cost more time but save the number of agents in some cases; while the latter save time but uses more agents comparing to former method. Here we are confronted with two tradeoffs: whether to chase the $Keep\,Moving$ agents and whether to surround the new formed BVs parallelly. In the following section we first introduce these two techniques and in next chapter we discuss the tradeoffs between them.

### *D. notification Moving Agents Technique*

**Overview of the notification Moving Agents Technique**

When the $Shadowed\,Exploring$ ends, it is possible that some of the agents in the array are not informed and do not realize the existence of the BV, so they keep moving following the routes in $Shadowed\,Exploring$ phase but it is obvious that they would not encounter any BV. In order to reduce waste, we employ the agent who receives the clone from chord $d_k$ ($Coordinate\,Agent$) to notice the other $Keep\,Moving$ agents to move back to their position when the BV is triggered. Now we introduce how we choose the $Coordinate\,Agent$ and how the $Coordinate\,Agent$ notify the other agents.

**The Process of the notification Phase of the Coordination Agent**

We can see that the relative position of agents does not change when they keep moving along the longest chord. For example, agents residing in nodes 1, 5, 10 (noted as $A1$, $A5$, $A10$) move along the longest chord and then $A5$ moves forward for one step. The relative position of them would be exactly the same as the situation where all of them remain dormant except $A5$ moves forward for one step. So now we discuss how to notify all the $Keep\,Moving$ agents assuming they are dormant and then make some modification to fit the scenario where the $Coordinate\,Agent$ notifies the $Keep\,Moving$ agents. The problem we need to solve is that given a range which the agents are in and a $Coordinate\,Agent$ located in this range, let the $Coordinate\,Agent$ to notify all the agents in this range. In a chord ring $C_n(1, d_2, \ldots, d_k)$, the $Coordinate\,Agent$ sets a $Notification\,Window$ using the modular arithmetic. Let us assume the number of the node where the $Coordinate_Agent$ resides in is $x$, the number of the nodes where should be marked the $Beginning\,Flag$ and the $End\,Flag$ are $y$ and $z$.

The relations between $x$, $y$, $z$ should be as follow:

- $y$ is the biggest number which satisfies that it is smaller than $x$ and that $y$ mod $d_k$ =0;
- $z$ is the smallest number which satisfies that it is bigger than $x$ and that $z$ mod $d_k$ = $d_{k-1}$.

When the agent is chosen to be the $Coordinate\,Agent$, it computes its $Notification\,Window$. When we mention marking a flag, it does not mean that the agent has to move to the node to do that but only needs to remember the positions of the two flags in its memory. Also, after the position of the notice window is set, it remains stable. More specifically, when the $Coordinate\,Agent$ moves, he does not set another notice window using its new position. The $Coordinate\,Agent$ moves step by step to every possible position and notifies the agents to go back if there is one until it realizes that it just passes the $End\,Flag$. After that, he moves along the longest chord anticlockwise to the node marked a $Beginning\,Flag$ and continues moving again step by step to notify agents until it arrives its relative departing node. For example, if the $Coordinate\,Agent$ starts to notify other from node $x$, then its relative departing node is $x' = x + t \times d_k$ ($t \in \mathbb{N}$).

We make some modifications to let the solution fit the real scenario where the agents move along the longest chord:

- The Beginning Flag and the End Flag move along the longest chord also to keep the relative position the same.
- When one agent $A1$ moves to a node where there is an agent $A2$ knowing the position of the original BV, $A1$ would be informed and directly moves along the longest chord to its own position.
- Let us assume that the time when the original BV is triggered is $T_{move\_i}$ ($T\_trigger$), then the $Coordinate\,Agent$ should remember the $T_{trigger}$ and informs the agents he encounters of it. The agent $A1$ who encounters the $Coordinate\,Agent$ should remember the time when they encounter ($T_{notice\_now}$) and stop moving until next $T_{move}$ when it will meet another agent $A2$. Then $A1$ moves along $d_k$ anticlockwise for $T_{move\_now} - T_{trigger}$ times while $A2$ moves for $T_{move\_now} - T_{trigger} + 1$ times.
- When arriving its relative departing position at $T_{move\_a}$, the $Coordinate\,Agent$ knows that it has finished the task and moves along $d_k$ for $T_{move\_a} - T_{trigger} + 1$ times to its position when the original BV is triggered.
- Let us assume that the time when the BV is triggered is $T_{move\_i}$, the $Coordinate\,Agent$ starts to move step by step to notify other agents after $T_{move\_i+2}$, because we want to ensure the security of the $Coordinate\,Agent$. If it starts to notify other agents at $T_{move\_i+1}$, it might encounter a BV. Also, it should be ensured that the $Keep\,Moving$ agents in the exploring group are in the $Notice\,Window$ of the $Coordinate\,Agent$, or the $Coordinate\,Agent$ would never encounter them. We would talk about how to ensure this in next section.

**Election of the Coordination Agent**

We choose the agent who receives a BV clone from its chord $d_k$ to be the $Coordinate\,Agent$, which means when an agent receives a BV clone from its longest chord, then it realizes that it is chosen as the $Coordinate\,Agent$. We know that, the notice phase starts from $T_{move\_i+2}$ assuming the time when the BV is triggered is $T_{move\_i}$, which means

the notification Phase begins only after all the $Keep\,Moving$ agents move twice. At $T_{move\_i+2}$, all the $Keep\,Moving$ agents are in a notice window from nodes $d_k \times (move\_i + 1)\,to\,d_k \times (move\_i + 1) + d_k - 1$, and let us donate it by $Initial\,Notice\,Window$. The $Coordinate\,Agent$ would directly move to any node in $Initial\,Notice\,Window$. Now we propose three kinds of routes for the $Coordinate\,Agent$ to move to his destination:

The coordinates of the positions where the clones spread are: $x-d_k$ (which is the coordinate of the $Coordinate\,Agent$), $x - d_{k-1}$, $x - d_{k-2}$, …, $x - 1$, $x + 1$, $x + d_2$, …, $x + d_{k-1}$, $x + d_k$ supposing the coordinate of the original BV is $x$ and we are in a chordal ring $C_n(1, d_2, \ldots, d_k)$. Now we describe three scenarios:

- Scenario 1: The last agent of the $Exploring\,Group$ is destroyed by the BV and the positions of the clones satisfy: $x - d_{k-1} = x + 1$, $x - d_{k-2} = x + d_2$, …, $x - 1 = x + d_{k-1}$.
- Scenario 2: The last agent of the $Exploring\,Group$ is destroyed by the BV and the at least one pair of the positions of the clones does not satisfy: $x-d_{k-1} = x+1$, $x - d_{k-2} = x + d_2$, …, $x - 1 = x + d_{k-1}$.
- Scenario 3: One of the agents in the $Exploring\,Group$ except the last agent is destroyed by the BV.

In scenario 1, the $Coordinate\,Agent$ needs to move for 5 steps to reach its destination while in the other two scenarios, it only needs to move for 4 steps to arrive the destination. Now we propose the route for each scenario.

- For $CA$ in scenario 1: Let us donate by $y$ the coordinate of the node in the $Notice\,Window$ set by the original BV which does not receive any clone and his left neighbour receives a clone (the coordinate of it is $y - 1$). The $CA$ first moves to the original BV, then to node $y - 1$, finally to $y$. After that it only need to move along the chord $d_k$ for twice to reach its destination.
- For $CA$ in scenario 2: There is at least one pair of the positions of the clones does not satisfy the equations so there should be one node (assuming its coordinate is $z$) who receives a clone from the original BV but node $z+d_k$ is empty. The route now for the $CA$ is first move to the BV, then to node $z$, and then moves along the chord $d_k$ for twice to reach its destination.
- For $CA$ in scenario 3: The $CA$ here simply need to move for one step to its right neighbour and move along the chord $d_k$ for three times to reach its destination.

In any case, the $CA$ can reach its destination within 6 unit of time which is required for the $Keep\,Moving$ agents to move to the $Initial\,Notification\,Window$, so the $CA$ start to chase the $Keep\,Moving$ agents as we introduce from $T_{notify\_i+2}$.

An example of how the agents and the $CA$ move in chordal ring $C_n(1, 2, 7, 11)$is shown in 4.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |
| 33 | 34 | 35 | 36 | 37 | 38 | 39 | V | 41 | 42 | 43 |
| 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 |
| 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 | 65 |

Fig. 4. Arrangement of agent at $T_{move\_2}$ when the BV is triggered

Yellow nodes are connected to the original BV but guarded by agents while the grey nodes are the new formed BVs. The node marked $V$ is the original BV but now is clean. Agent residing in node 29 receives clone from chord $d_k$ so it knows it is the $CA$. During the notification time, agents residing in nodes 33, 38, 39 notify agents residing in nodes 36, 31, 30 respectively following the ?Three Jump Notice Technique? while the $CA$ moves to 28, 39, 50 and finally 61 following the route in scenario 3.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |
| 33 | 34 | 35 | 36 | 37 | 38 | 39 | V | 41 | 42 | 43 |
| 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 |
| 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 | 65 |
| 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 |

Fig. 5. Agents roles after $Three\,Jump\,Notification$ and choosing $CA$. (for convenience, we donate the $CA$ by a red spot, more specifically, node 50 is where $CA$ resides)

Agents in purple nodes would be noticed at $T_{move\_3}$ and move back. Agents in light green nodes are the $Keep\,Moving$ agents while agent in dark green nodes are informed to stop in $Three\,Jump\,Notification$. In the meantime, the $CA$ moves to node 28, 39, 50, and finally 61. It is obvious that the $CA$ can reach its destination before $T_{move_4}$, so it waits until $T_{notice_4}$ to start its notification phase.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |
| 33 | 34 | 35 | 36 | 37 | 38 | 39 | V | 41 | 42 | 43 |
| 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 |
| 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 | 65 |
| 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 |

Fig. 6. Arrangement of agent at $T_{move\_3}$. The $CA$ has arrived its destination)

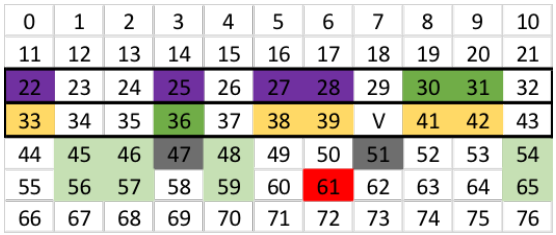



Fig. 7. Arrangement of agents at $T_{move_4}$. The $CA$ starts its notice phase

In notice phase, $CA$ starts to notify other $Keep\,Moving$ agents. First, it computes the $Notice\,Window$ which is from node 55 to node 65. Note that the $Notice\,Window$ would move along chord $d_k$ at every $T_{move}$. It moves to node 62 at $T_{notice\_4}$, node 63 at $T_{notice\_4?}$, node 64 at $T_{notice\_4??}$ and to node 75 at $T_{move\_5}$.

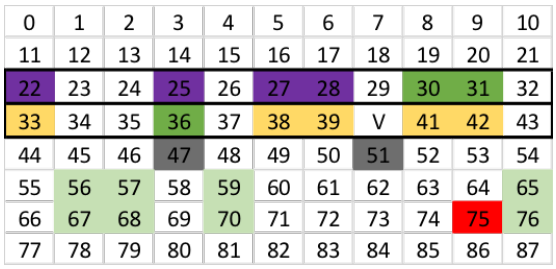



Fig. 8. Arrangement of agents at $T_{move_5}$.

Again, in the notification phase, $CA$ moves to node 76 at $T_{notice\_5}$. We can see that it encounters agent residing in node 76, so $CA$ informed it the $T_{trigger}$ which is $T_{move\_2}$. Agent residing in node 76 should remember $T_{notice\_now}$ which is $T_{notice\_5}$ and wait until next $T_{move}$ to inform agent ($Following\,Agent$) who resides in node 65 now but would move to node 76 next $T_{move}$. After encountering its $Following\,Agent$, it informs it to move back along chord $d_k$ for $T_{move\_now} - T_{trigger} + 1$ times which is $T_{move\_5} - T_{move\_2} + 1$ times while itself moves for $T_{move\_5}$- $T_{move\_2}$ times. At $T_{notice\_5?}$ when the $CA$ arrives at node 77, it knows that it just pass its $Ending\,Flag$ so it moves along the longest chord anticlockwise to its $Beginning\,Flag$ at $T_{notice\_5??}$.

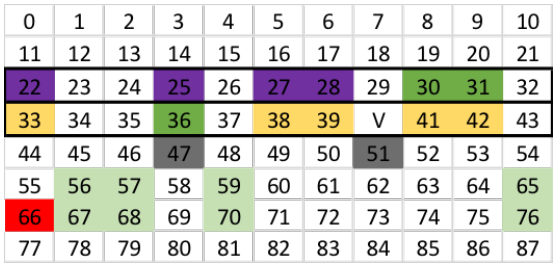



Fig. 9. Arrangement of agents at $T_{move_5''}$.

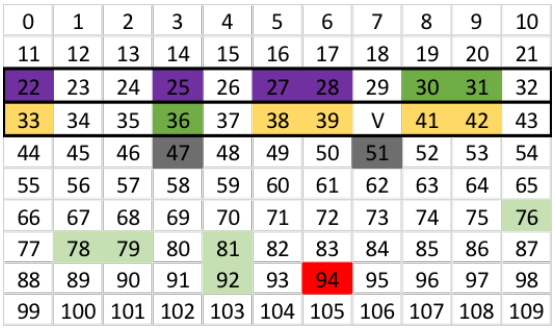



Fig. 10. Arrangement of agents at $T_{move_7''}$.

We could know that the $CA$ moves back to its relative original position at $T_{notice\_7??}$. It would wait until next $T_{move}$ to inform its $Following\,Agent$ of $T_{trigger}$ and moves back with it.

### *E. Overview of the Elimination*

After all the agents move back to where they are when the BV is triggered, we start the $Surrounding\,and\,Elimination$. We are interested in destroying the BVs at one time. So we need to first guard all the neighbouring nodes of the new formed BVs. In order to avoid collision and efficiently leverage the agents, we allocate different Destination Tables to all the agents in the array to inform them where should they should move in different situations (e.g., when the first agent in the exploring team is destroyed, then every agent except the first agent have a distinct destination, when the second agent is destroyed, then every agent except that agent destroyed have a distinct destination. More specifically, for a Chord Ring with half degree $d$, every agent in the array carries a $Destination\,Table$ with $d-1$ destinations. If we need more agents, then we will give their $Destination\,Table$ to the last agent in the shadowing group, when the elimination begins, it clones enough number of agents and give the $Destination\,Table$ to them. Before moving to its destination, the agent computes the shortest route from its own position to its destination using Dijkstra Algorithm. There are two kinds of agent in the Elimination phase: surrounding agents who are responsible for guarding the neighbouring nodes of the BVs and destroying agents who move to the BVs after all the neighbouring nodes are guarded. We want the BVs to be destroyed at one time, so it is important that the destroying agents move to the BVs at the same time and only after all the neighboring nodes are guarded by agents. In fact, if the destroying agents know the longest time $t_{longest}$ to move to the destination taken by all the agents (including the destroying agents and the surrounding agents), then they move to the last node prior to the destination and wait until $t_{longest}$ to move to the BVs together. So in the $Destination\,Tables$ for the destroying agents, we also add an item which is the $t_{longest}$. Now we introduce how to compute the shortest routes and how we design the $Destination\,Tables$. Note that we design $Destination\,Tables$ for all the agents and allocate them to the agents before the exploring phase begins.

### F. Destination Table and Elimination

Supposing there are some BVs and agents in the chordal ring, it is obvious that the BV nodes are in the clockwise side of the agents. In order to use Dijkstra, first we need to map the chordal ring with BVs into a graph. We include nodes from the node containing the first agent to the node which is $d_k$ away from the last BV node, then delete the chords from the BV nodes to build the graph where we run Dijkstra Algorithm. Here is an example how we built the graph for running Dijkstra Algorithm. Below we show the situation when the third agent in the exploring group is destroyed by the BV (see 11). Only the chords of the original BV node are shown.

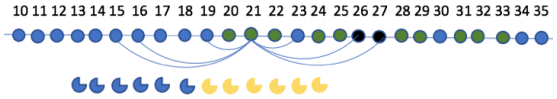



Fig. 11. Situation when the third agent in the exploring group is destroyed

The black node is the BV node while the green nodes need to be guarded. So in this case, we need 12 agents (10 surrounding agents and 2 destroying agents). We add nodes from 13 to 33 with their chords within this area and delete chords connected with the BV nodes to get the graph where we use Dijkstra Algorithm. Below is the graph we build. (see 12) For convenience, we show the all the nodes we included and the chords we delete.

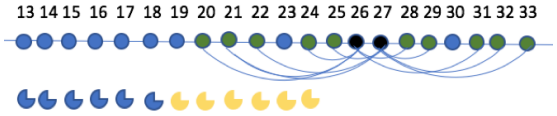



Fig. 12. The graph we build for Dijkstra Algorithm

Using the graph and Dijkstra Algorithm, we compute the routes from every agent to every node. Then we use enumeration to choose an allocation of every agents's destination satisfying:

- 1) the maximum length of the route should be minimum.
- 2) after the allocation, in every needed position there should be exactly one agent.

After we get the optimal allocation, we record every agent?s distinct destination and the position of the third agent in the exploring group in their $Destination\,Table$. Also, we add the length of the longest chord to every destroying agent. More specifically, here we talk about the situation when the third agent in the exploring group is destroyed. For every agent, the information we compute would be in the third part of its $Destination\,Table$ recording the position of the third exploring agent (for example, it connects this agent through chord $x$), the destination it should move if the third exploring is destroyed. For a destroying agent, there should be another item recording the length of the longest route in this part. Above we introduce how to design one part of $Destination\,Table$ of an agent, for every agent in chordal ring, it should hold a $Destination\,Table$ of $d-1$ parts, and every part contains 2 items (for surrounding agent) or 3 items (for destroying agent). After the one of the agent is destroyed, the agent can check their $Destination\,Table$ to get the information of their destination. Then using Dijkstra Algorithm they can compute the shortest route separately and starts to move.


## ACKNOWLEDGMENT

The preferred spelling of the word "acknowledgment" in America is without an "e" after the "g". Avoid the stilted expression "one of us (R. B. G.) thanks ...". Instead, try "R. B. G. thanks...". Put sponsor acknowledgments in the unnumbered footnote on the first page.


## REFERENCES

Please number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use "Ref. [3]" or "reference [3]" except at the beginning of a sentence: "Reference [3] was the first ..."

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors' names; do not use "et al.". Papers that have not been published, even if they have been submitted for publication, should be cited as "unpublished" [4]. Papers that have been accepted for publication should be cited as "in press" [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

## REFERENCES


[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, "Title of paper if known," unpublished.

[5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.