Decontamination of an Arbitrary Network from Multiple Black Viruses

J. Cai*, P. Flocchini†, N. Santoro*
* School of Computer Science, Carleton University, {jie.cai,santoro}@carleton.ca
† EECS, University of Ottawa, pflocchi@uottawa.ca

Abstract

In networks supporting mobile agents, a particular threat is that posed by the presence of a black virus (BV), a harmful entity capable of destroying any arriving agent, and of spreading clones to neighbouring sites. A BV can only be destroyed if at the same site there is an anti-viral agent; but, if the BV is already active when encountering the agent, it spreads clones to all the neighbouring sites. The problem of locating and permanently eliminating all such harmful presences has been studied both in special classes of regular network topologies, and in networks of arbitrary and unknown topology; however, in all these studies, it was assumed that there was a single BV in the system. In this paper we consider the case of multiple BVs. We present fully distributed decontamination procedures that operate with no a priori knowledge of the network topology, nor of the number and location of the black viruses.

1 Introduction

Consider a networked environment supporting mobile agents; malicious entities may be present in the system possibly damaging the network’s sites and harming the mobile agents (e.g., see [16]). In the distributed computing literature one such type of malicious entity, the black hole, has been extensively studied (e.g., [1, 6, 7, 8, 10, 11, 9, 15, 18, 20]). A black hole is a node infected by a process which destroys any incoming agent without leaving any detectable trace of the destruction. The primary concern has been on locating the position of the black hole so that it can be deactivated. Another type of harmful mobile entity, the intruder, has also been studied from a theoretical point of view, focusing mostly on intruder capture (also known as graph decontamination and connected graph search) (e.g.,[2, 12, 13, 14, 17, 19]). In such a problem, an extraneous mobile agent, the intruder, moves through the network infecting the visited sites; the task is to decontaminate the network using a team of system agents avoiding recontamination. The main difference between these two types of malicious entities is that a black hole is a stationary presence which is harmful to the agents but not to the network sites, while the intruder does not cause any harm to system agents but, being mobile, is harmful to the network sites.

Recently some investigations have started to focus on an harmful entity, called black virus (BV), that combines some of the destructive power of black holes with the mobility feature of the intruders [4, 5]. Like a black hole, a BV destroys any agent arriving at the network site where it resides. Special system agents, called cleaners, are capable of disabling the virus: when a cleaner is at a site when a BV arrives, the BV is disabled without any further consequence; on the other hand, if a cleaner arrives at a BV site, the BV is disabled but it destroys the cleaner, and clones of the BV will spread to all the neighbouring sites. The objective is to use a team of system cleaning agents to permanently deactivate all BVs and their clones from the network. This task, called black virus decontamination (BVD), is clearly dangerous for the system agents performing the deactivation, (since any agent arriving at a node where an instance of the BV resides will be destroyed), as well as for the nodes where the BV will spread to. The objective is to design algorithmic strategies that would enable the team of system agents, injected in the system at the same network site, to move in the network so that, within finite time, any presence of the BVs is permanently removed. The goal of a solution protocol is to minimize the spread of the BVs, i.e. the number of node infections by the BVs, as well as the number of agent lost in the process, the casualties. The clones can have the same harmful capabilities of the original BVs (fertile) or be unable to produce further clones (sterile).

The BVD problem has been studied so far exclusively in presence of a single black virus. Solutions have been proposed for networks with special topologies, namely grids, tori, and hypercubes [5], as well as in arbitrary topologies [3, 4]. In this paper, we provide solutions for the most general case when an arbitrary number of BVs is initially present and their number and location are unknown to the agents; our solutions, for sterile and for fertile clones, are fully decentralized and work in asynchronous networks of arbitrary unknown topology.
2 Model and Terminology

The topology of the network is modelled as a simple undirected graph $G = (V,E)$ with $n = |V|$ nodes (or sites) and $m = |E|$ edges (or links). We denote by $E(v) \subseteq E$ the set of edges incident on $v \in V$, by $N(v) \subseteq V$ the set of its neighbours, by $d(v) = |E(v)|$ its degree, and by $\triangle(G)$ (or simply $\triangle$) the maximum degree in $G$. Every node $v$ has a distinct identity $id(v)$. The links incident to a node $v$ are labelled with distinct arbitrary port numbers from a set $l_v$ (w.l.g. let $l_v = \{1,2,3,...,d(v)\}$). A team $A = \{a_1,\ldots,a_k\}$ of mobile system agents, called cleaners, is injected in the network at a node $h$, the home base. Each agent $a \in A$ is a computational entity with its own local memory and a unique identifier $id(a)$ from some totally ordered set, and it can move from node to neighbouring node. The agents can have different roles (i.e., different states), but they all operate according to the same protocol. More than one agent can be at the same node at the same time. Communication among the agents is face-to-face: two (or more) agents can communicate only when at the same node, and there are no a priori restrictions on the amount of exchanged information.

The agents have no a priori knowledge of the network $G$ nor of its size, and they are provided with limited visibility: that is, an agent at node $v$ can perceive the 2-neighbourhood $N^2(v)$ of $v$, including the node identities and the edge labels. Some of the nodes in the network are infected by a black virus (BV), a process endowed with reactive capabilities for destruction and spreading. A BV can be deactivated only by a system agent called cleaner arriving at the same node; however, in the deactivation process, the cleaner loses its antiviral power (it is destroyed) and the BV releases clones that spread to the neighbouring nodes. If a BV clone arrives at a node with no cleaner, it infects the node and becomes resident there; if instead it reaches a node already occupied by a cleaner, the clone is deactivated before it can cause any harm. The clones can have the same harmful capabilities of the original BV (fertile) or they may be unable to produce further clones (sterile).

We assume that, at the same node, multiple BVs (clones or original) are merged; i.e., at any time at each node there is at most one BV, with an original BV taking precedence over clones. In the system there are no global clocks, and the duration of any activity (e.g., processing, communication, moving) by all the entities is finite but unpredictable; in other words, the system is asynchronous and any needed synchronization must be achieved by the agents’ protocol.

The Black Viruses Decontamination problem (BVD) consists of removing the original BVs and all their clones from the network using the team of cleaners. A protocol defining the actions of the cleaners solves BVD in $G$ if, regardless of the location of the home base and of the number and locations of the BVs, within finite time the network is free of BVs and at least one cleaner survives. A protocol solves BVD if it solves it in any network $G$. The cost measure of a solution protocol is $P$ in a network $G = (V,E)$, is the spread$(P,G)$, that is the maximum number of nodes contaminated by the BVs and their clones when executing $P$ in $G$ in the worst case (i.e., over all possible initial locations of the BVs and of the home base, and all possible execution delays). Note that, since each instance of the BV (original or clone) has to be eventually removed and since each removal requires the destruction of at least one cleaner, the spread also measures the number of cleaner casualties.

A solution protocol is monotone if no recontamination occurs in any of its executions; that is, once a node is visited by a cleaner, it will not be (re)contaminated by a BV clone. A solution protocol $P$ is sequential if it prescribes the nodes to be explored one a time, and the exploration of a new node to be started only after the exploration of the previous node has been completed. An important property is the following:

**Theorem 2.1.** [5] Let $P$ be a solution protocol.
Then there exists a monotone sequential solution protocol $Q$ such that, for any asynchronous network $G$, spread$(P,G) \geq$ spread$(Q,G)$.

Hence, the quest for spread-efficient solution protocols can be limited to monotone sequential protocols.

Let $\mathcal{P}$ be the set of all monotone sequential solution protocols. Consider a protocol $P \in \mathcal{P}$, and a graph $G = (V,E) \in \mathcal{G}$. Since $P$ is sequential, the nodes of $G$ are visited for the first time one at a time, starting from the home base. Let $P(G,h) = [x_0,x_1,x_2,...,x_{n-1}]$ be the resulting ordered sequence in a BV-free execution of $P$ in $G$ starting from $h = x_0$. We call residual degree of $x_i$ in $P(G,h)$ the number of neighbours of $x_i$ following it in the sequence $P(G,h)$; i.e., $\rho(x_i) = |\{x_j \in V(x_i) : n > j > i\}|$. We say that $\rho(P(G,h)) = \max_{n>0} \{\rho(x_i)\}$ is the residual degree of the entire sequence $\pi_P[h]$.

3 The Case of Sterile Clones

3.1 Basic Bounds

A simple upper bound on the spread of monotone protocols is the following.

**Theorem 3.1.** Let $G$ be an arbitrary graph with $n_{BV}$ black viruses producing sterile clones. Any monotone decontamination protocol $G$ incurs at most $\Delta(G) \cdot n_{BV}$ casualties.
Proof. By definition, every monotone protocol $P$ is designed so that an explored node is never recontaminated. This means that $P$ must guarantees that a BV contaminates only its non-explored neighbours; thus, in every execution of $P$ in $G$, a BV contaminates with clones at most $\Delta(G) - 1$ nodes; since these clones are sterile, each will cause only one casualty; by adding the casualty done for the BV node itself, the claim follows.

This upperbound can be tight, as shown below.

**Theorem 3.2.** There is an infinite number of graphs $G$ with $n_{BV}$ black viruses producing sterile clones where every monotone decontamination protocol incurs at least $\Delta(G) n_{BV}$ casualties.

To prove this theorem, consider the following infinite family of regular graphs (i.e., where all nodes have the same degree). A graph $G(d, k)$, $d \geq 2$ and $k \geq 0$, in this family is composed of $2k$ complete graphs of dimension $d - 1$, $G_1, G_2, ..., G_{2k}$ plus two single nodes $x$ and $y$; there is a matching between the nodes of $G_i$ and those of $G_{i+1}$ for $i < 2k$; there is an edge between $x$ and $y$; there is an edge between $x$ and all nodes of $G_1$, and an edge between $y$ and all nodes of $G_{2k}$; see Figure 1 for an example. It is to verify that $G(d, k)$ so defined has $n = 2k(d - 1) + 2$ nodes and is $d$-regular.

![Figure 1: G(5, k)](image)

**Theorem 3.3.** To decontaminate $G(k, d)$ from $n_{BV} \leq k$ black viruses producing sterile clones, the number of casualties required in the worst case is $\text{spread}(G(k, d)) = d n_{BV}$.

Proof. Consider any monotone sequential decontamination protocol $P \in \mathcal{P}$ starting from $y$ as the homebase. Consider the situation when there is a single BV in each of the clusters $G_1, G_3, ..., G_{n_{BV}-1}$. In each of those clusters, the choice of which node is a BV is made by the adversary according to the following rule: when a cluster is visited for the first time, that node is declared a BV. This means that if cluster $G_{2i+1}$ is visited for the first time the entire cluster will be infected. Furthermore, if the exploration came from $G_{2i}$ (or $y$ if $i = 0$), then also the neighboring node in $G_{2i+2}$ (or $x$ if $i = k - 1$) will be contaminated; similarly, if the exploration came from $G_{2i+2}$ (or $x$ if $i = k - 1$), then also the neighboring node in $G_{3i}$ (or $y$ if $i = 0$) will be contaminated. In other words, each BV will contaminate $d - 1$ nodes, and there is no overlap between these contamination; thus in total, there will be $n_{BV}(d - 1)$ contaminations by sterile clones. Adding the $n_{BV}$ casualties needed to clear the BVs, the theorem follow.

Notice that, in this proof, the adversary was able to obtain the maximum possible number of casualties by placing the BVs separate from each other. Indeed the distance between the BVs can play a role on the total number of casualties.

### 3.2 Decontamination Protocol

The strategy of the proposed decontamination protocol follows two phases: shadowed exploration and elimination.

In the shadowed exploration phase, starting from $h$, a sequence of nodes to be explored (exploration sequence) is constructed by the agents while exploring; initially, the sequence contains only the homebase. At each step, the agents select a target in the map of the graph constructed so far; initially the map contains only the two-hops neighbourhood $N^2(h)$ of the homebase. The target is chosen among the unexplored neighbours of the already explored nodes (the frontier), according to a greedy criterion: the selected node is one with minimum residual degree (recall that the residual degree of a node is the number of its unexplored neighbours); should there be multiple candidates, the one with the shortest distance from the last target is chosen (symmetry is broken arbitrarily). Once the target has been selected, some special cleaner agents (the shadows) move to occupy the explored neighbours of the target, so to protect them from BV clones. Once this operation is concluded, an agent (the explorer) moves to the target. In this process, special care must be taken when the explorer is moving to a target that is known to contain a BV clone (due to some earlier triggered BV). In fact, should this unexplored node also contain an original BV, not only the exploring agent would be eliminated, but also its clean neighbours, thus making the procedure non-monotone. To overcome this difficulty, the agents have to move in a doubly-cautious walk as follows: whenever exploring, from a node $v$, an unexplored node $u$ that contains a BV clone, the exploring agent (say $a$) is followed by a verifying agent (say $c$), whose function is to contribute to verify whether or not $u$ is a BV node. Observe that, upon its arrival, will destroy the BV clone (and possibly also an original BV resident there), and will be destroyed; hence, when $c$ arrives, it will find no trace of other entities (BV, clone, exploring agent); the verifying agent $c$ then returns to $v$. If $u$ is a BV node, the arrival of $a$ will send a BV clone to all of $u$’s
neighbours, including \( v \); should this happen, the clone will arrive at \( v \) before \( c \) returns. In other words, once \( c \) has departed for \( u \), within finite time, either a BV clone will arrive followed by \( c \), or just \( c \) will return. Thus, at node \( v \) it can be detected whether or not \( u \) was a BV node and the appropriate update to the map and to the other variables can be performed. If the target is not a BV node, the current map of the network is updated so to include the information acquired from visiting the target, and a new step of the exploration process takes place. If instead the target is a BV node, the explorer dies, the BV is deactivated, and BV clones move to all its neighbours. The clones arriving at the neighbours still unexplored (and thus unprotected by shadows), transform them into BV nodes; on the other hand, the clones arriving at already explored nodes are destroyed by the shadows located there. Once a BV node is found, causing BV clones to spread to the unprotected neighbouring nodes, the agents record in the map where all the clones of that BV have moved, and they continue with the shadowed exploration going to the successive target.

When the entire network has been visited, the elimination phase starts. Note that at this time, the map contains all the new BV nodes and their neighbours. In the elimination phase, each BV clone is decontaminated by an exploring agent moving there, once all its non-BV neighbours have been occupied by shadow agents.

Attention has to be made also when calculating the residual degrees of the nodes in the frontier (to choose the target node to explore). In fact, if there is an edge between a frontier node \( x \) and an unexplored node \( y \) which is in the clone list (i.e., \( y \in V^{-}_u \)) \( y \) is not considered in the calculation because, if \( x \) is chosen as the target and it turns out to be a BV, then the clone it would send to \( y \) is irrelevant (i.e., it does not increases the number of nodes to be decontaminated) because \( y \) contains already a BV (clone or original). We denote by \( \tilde{\rho} \) such a modified residual degree. The protocol MBV DECONTAMINATION STERILE, or MS for short, is shown in Figure 2.

3.3 Cost

The number of casualties incurred by the execution of the algorithm depends on the number and location of the BVs in the graph. We can calculate them precisely. Let \( \pi = \langle x_0, x_1, ..., x_{n-1} \rangle \), \( x_0 = h \), be the minimum residual degree sequence of the algorithm, computed as described above; and let \( \xi : V \rightarrow \{0, 1\} \) be such that \( \xi(x_i) = 1 \iff x_i \) is a BV. By construction we have:

\[ \text{Theorem 3.4. The number of casualties of Protocol MS is spread}(MS,G) = \sum_{0<i<n} (\tilde{\rho}(x_i, \pi) + 1)\xi(x_i) \]

MBV DECONTAMINATION STERILE

(* Initialization *)
All agents initially at home-base \( h \).
\( M = (V_M, E_M) := N^2(h); (* initial Map *) \)
\( V^o_\text{ex} := \{h\}; (* only the home base is explored *) \)
\( V^o_u := V_M \setminus \{h\}; (* initial unexplored nodes *) \)
\( V^o_u := \emptyset; (* initial unexplored nodes *) \)

\( \text{containing a BV or a BV clone} \)

\( F:\! = N(h); (* unexplored frontier *) \)
(* Iteration *)
while \( V^o_u \neq \emptyset \)
for all \( v \in Fr \) do
\[ r(v) := |\{ u \in V^o_u : (u,v) \in E_M \}|; (* residual degree of frontier *) \]
endfor
Choose \( v \in Fr \) such that \( r(v) \) is minimum;
(* selection of target \( v \) *)
\( N^o_{ex}(v) := \{ u \in V^o_{ex} : (u,v) \in E_M \}; (* explored neighbours of target *) \)
Position a shadow agent at each \( u \in N^o_{ex}(v) \);
if \( (v \in V^o_u) \) then
Move an exploring agent followed by a verifying agent to \( v \);
else
Move an exploring agent to \( v \);
endif
\( M := M \cup N^2(v); (* update map*) \)
\( V^o_{ex} := V^o_{ex} \cup \{v\}; (* update explored nodes*) \)
\( V^o_u := V_M \setminus V^o_{ex}; (* update unexplored nodes*) \)
if \( (v \text{ was a BV}) \) then
\( V^o_u := (V^o_u \setminus \{v\}) \cup (N(v) \setminus V^o_{ex}); (* update clone list *) \)
endif
\( Fr := \{ x \in V^o_u : \exists y \in V^o_{ex}, (x,y) \in E_M \}; (* update frontier*) \)
endwhile

Figure 2: MBV DECONTAMINATION STERILE (MS)

From this, it immediately follows that:

**Theorem 3.5.** Let \( G \) be an arbitrary graph contaminated by \( n_{BV} \) black viruses which produce sterile clones. Decontamination of \( G \) starting from \( h \) can be performed with \( \text{spread}(G) \leq (\tilde{\rho}(G, h) + 1) n_{BV} \)

That is, each BV will cause a number of additional casualties which is at most the minimal residual degree of the graph.

4 The case of Fertile Clones

4.1 Lower Bounds

In the case of fertile clones, the initial location of the BVs can have catastrophic consequences for the
cost of decontamination. Let $V_{BV}$ be the set of BV nodes. The removal of $V_{BV}$ and their incident edges from $G$ can create a set of disjoint connected subgraphs $G_0, G_1, ..., G_{k-1}$; let $G_0$ be the subgraph containing the homebase $h$; then each subgraph, except for $G_0$, will be eventually contaminated. We call such a condition a separation (see Figure 3).

Figure 3: An example of separation; the dark lines denote nodes with a BV.

**Theorem 4.1.** Decontamination of $G$ requires at least $|V_{BV}| + \sum_{0 \leq i < k-1} |V_i|$ casualties, where $V_i$ is the set of vertices of $G_i$.

Even if initially there is no separation (i.e. $k = 1$), the graph can break into separate components during the execution of a decontamination protocol. Consider a sequential monotone decontamination protocol $P$ and consider its execution in $G$ at time $t$. Let $V_{BV}(t)$ be the set of BV nodes at time $t$; the removal of $V_{BV}(t)$ and their incident edges from $G$ will create a set of disjoint connected subgraphs $G_0(t), G_1(t), ..., G_{k-1}(t)$; let $G_0(t)$ be the subgraph containing the homebase; due to monotonicity, $G_0(t)$ is the explored portion of the graph at this time. If $k > 1$, we call this condition one of dynamic separation. Then, with a reasoning similar to that of the proof of Theorem 4.1, we have the following.

**Theorem 4.2.** The number $\text{spread}(P, G)$ of casualties of protocol $P$ in $G$ at time $t$ is at least $\text{spread}(P, G) = \text{spread}(P, G, t) + |V_{BV}(t)| + \sum_{0 \leq i < k-1} |V_i(t)|$, where $V_i(t)$ is the set of vertices of $G_i(t)$, and $\text{spread}(P, G, t)$ is the number of casualties incurred by $P$ up to time $t$.

Note that the agents might not be able to detect if a condition of dynamic separation exists at a given time, because they do not know the location of the unexplored original BVs. On the other hand, they can detect if a dynamic separation has been caused by the clones. As mentioned, separation at any time (initial or dynamic) has a negative effect on the number of casualties because all the non-BV nodes inside those components will become infected. Another important observation is that, even if there is no initial separation and the execution might not cause dynamic separations, there are networks where no protocol can avoid $O(n)$ casualties, even if the number of BVs is small. Consider the graph $R$ where the homebase is connected to a ring of $2n_{BV}$ nodes, each connected to a (possibly distinct) node of a connected graph $G'$ of $n - (2n_{BV} + 1)$ nodes; see Figure 4 where $n_{BV} = 5$.

Figure 4: Five appropriately placed BVs can contaminate all nodes but $h$.

**Theorem 4.3.** $\text{spread}(R) = n - 1$

*Proof.* Consider any monotone sequential decontamination protocol $P$, and consider the following game between $P$ and an adversary that will choose $n_{BV}$ out of the $2n_{BV}$ ring nodes to be BVs. Let us call “free” a ring node where a decision has not been made yet by the adversary; initially all ring nodes are free. When the protocol chooses as a target a ring node $z$, if $z$ is free, the adversary will designate it as a BV node; then both is neighbours on the ring are no longer free and become “contaminated”. If $z$ is “contaminated”, it will also contaminate its two ring neighbours. Notice that in this process the adversary will designate as BV nodes at most $n_{BV}$ ring nodes. It is easy to see that, whatever is the decontamination sequence of $P$, any selected target is a BV, either because so designated by the adversary, or because contaminated by a previous target. \hfill \Box

### 4.2 Decontamination Protocol

In the following we call the BVs that exist in the system before the decontamination starts the original BVs, and the ones that are generated by triggering the original BVs the exposed BVs, meaning that their locations is uncovered (i.e., exposed) by agents’ exploration. As usual, the number and location of the BVs and the network topology are unknown.

The general strategy of the algorithm, shown in Figure 5, is to sequentially explore the network from the home base, using shadows to ensure monotonicity, and keeping track of the exposed BVs. Based on the most current knowledge on the graph (i.e., which nodes have been explored, and which nodes are infected by BVs), the next node to be explored is chosen from the frontier, initially excluding the exposed BV nodes, as the one with the minimum number of neighbouring unexplored nodes that are not exposed BV; the reason not to consider a neighbouring exposed BV (say node...
z) when computing the residual degree of a candidate x, is because if x is an original BV and is chosen as a target, its clone moving to z will merge with the BV already there; thus it will not add to the number of infected nodes; in other words, x can be explored without worrying about infecting node z because z has been infected already. Observe that, in this selection process, the target is never an exposed BV; it could however be an original BV. If this is the case, its unexplored neighbours become exposed BVs and they are properly recorded. After each new exploration, the agents update the residual degrees of all unexplored nodes including the exposed BVs; again, when computing the residual degree of a candidate of its degree, its unexplored neighbours are shadowing their explored neighbours (procedure CLEAN_0-DEGREE_EXPOSED_BV). Note that eliminating a zero-residual-degree exposed node (zero-RD) does not require any surrounding and does not create any new BV node. After all zero-RD nodes are cleaned, a new target is selected and the exploration process continues. The only deviation from this process occurs if the agents determine, from the current map, that they can surround and clean some exposed BVs safely, i.e., without contaminating any non-BV node. In this case, the agents perform the cleaning (Procedure CLEAN_SAFE_BV) and then resume the exploration process. The exploration process terminates when either there are no more unexplored nodes or the frontier is composed solely of exposed BVs, each of residual degree greater than zero. In the first case, the decontamination process is terminated. In the second case, one more step, called Secondary Step, is needed. Let \( V_{BV}(t) \) be the set of exposed BV nodes at this time; the removal of \( V_{BV}(t) \) and their incident edges from G will create a set of disjoint connected subgraphs \( G_0(t), G_1(t), ..., G_{k-1}(t) \) where the subgraph \( G_0(t) = G_{ex} \) is the currently explored graph and contains the homebase, while all other \( G_i(t) \) are still unexplored. We know, by Theorem 4.2 that all nodes of those unexplored \( G_i \) will become necessarily contaminated. Thus, we simply proceed to sequentially explore the unexplored nodes, protecting with shadows their explored neighbours. The agents clean up all regions completely surrounded by exposed BVs, causing all unexplored nodes to become contaminated, then cleaned up. This process continues until there are no more unexplored nodes. The algorithm is shown in Figure 5, where the pseudo-code of procedures CLEAN_0-DEGREE_EXPOSED_BV and CLEAN_SAFE_BV is straightforward and not shown.

**MBV Decontamination - Fertile**

(* Initialization *)

All agents initially at home-base \( h \).

\[
M = (V_M, E_M) := N^2(h); (* initial Map *)
\]

\[
V_{ex} := \{h\}; (* only the home base is explored *)
\]

\[
V_{un} := V \setminus M \setminus \{h\}; (* initial unexplored nodes *)
\]

\[
V_{un} := \emptyset; (* initial unexplored nodes with exposed BV *)
\]

\[
Fr := N(h); (* unexplored frontier *)
\]

Finished := false;

(* Iteration *)

while \( V_{un} \neq \emptyset \) (* Primary Step *)

\[
Fr^+ := Fr \setminus V_{un};
\]

forall \( v \in Fr^+ \)

\[
r(v) := |\{u \in V_{un}^+: (u, v) \in E_M\}|;
\]

Choose \( v \in Fr^+ \) such that \( r(v) \) is minimum;

\[
N_{ex}(v) := \{u \in V_{ex} : (u, v) \in E_M\};
\]

Position a shadow agent at each \( u \in N_{ex}(v) \);

Move an exploring agent to \( v \);

\[
M := M \cup N^2(v); (* update map *)
\]

\[
V_{ex} := V_{ex} \cup \{v\}; (* update explored nodes *)
\]

\[
V_{un} := V_M \setminus V_{ex}; (* update unexplored nodes *)
\]

\[
Fr := \{x \in V_{un} : \exists y \in V_{ex} : (x, y) \in E_M\};
\]

if \( (v \ was \ a \ BV) \) then

\[
V_{un} := (V_M \setminus \{v\}) \cup (N(v) \setminus V_{ex});
\]

CLEAN_0-DEGREE_EXPOSED_BV;

CLEAN_SAFE_BV;

\[
V_{un} := V_{un} \setminus V_{ex}^+;
\]

endwhile

while \( V_{un} \neq \emptyset \) (* Secondary Step *)

Choose \( v \in Fr^+ \) (* selection of target \( v \) *)

\[
N_{ex}(v) := \{u \in V_{ex} : (u, v) \in E_M\};
\]

Position a shadow agent at each \( u \in N_{ex}(v) \);

Move an exploring agent to \( v \);

\[
M := M \cup N^2(v); (* update map *)
\]

\[
V_{ex} := V_{ex} \cup \{v\}; (* update explored nodes *)
\]

\[
V_{un} := V_M \setminus V_{ex}; (* update unexplored nodes *)
\]

\[
Fr := \{x \in V_{un} : \exists y \in V_{ex} : (x, y) \in E_M\};
\]

endwhile

**Figure 5: MBV Decontamination - Fertile (MF)**

### 4.3 Cost

The cost of the proposed protocol MBV DECONTAMINATION - FERTILE, MF for short, depends on many factors; in particular, it depends on (1) the topology of the network and the fact that it is unknown and chosen by an adversary; and (2) the location and number of BV, unknown to the agents and under the choice of an adversary. We can however establish some general bounds on the number of casualties.

**Theorem 4.4.** Let the decontamination of \( G \) by MF do not require executing Secondary Step. Then

\[
\text{spread}(MF, G) \leq n_{BV} \cdot \Delta(G)
\]

**Proof.** If the Secondary Step is not executed, then all the exposed clones are cleaned in the Primary
Step: this is done by the procedures CLEAN_0-DEGREE_EXPOSED_BV and CLEAN_SAFE_BV, without contaminating non-BV nodes. In other words, only the original BVs have contaminated non-BV nodes; since a BV can only contaminate its neighbours, then each BV will contribute at most $\Delta(G) - 1$ casualties needed to clean the non-BV nodes contaminated by its clones, plus one casualty for itself; thus the claim follows.

In other words, if the Secondary Step is not required, the number of casualties is quite limited; indeed, it is asymptotically optimal in bounded-degree graphs.

**Theorem 4.5.** Let the decontamination of $G$ by MF require executing Secondary Step, starting at time $t$. Then $\text{spread}(MF,G) \leq n_{BV}(t) \cdot \Delta(G_{ex}(t)) + |V_{un}(t)|$ where $G_{ex}(t)$ is the explored graph at time $t$, $V_{un}(t)$ are the still unexplored nodes at time $t$, and $n_{BV}(t)$ is the number of original BVs in $G_{ex}(t)$.

**Proof.** Let $\text{spread}(MF,G,t)$ be the number of casualties incurred by MF up to time $t$. By Theorem 4.2, $\text{spread}(MF,G) = \text{spread}(MF,G,t) + |V_{un}(t)|$. But, by Theorem 4.4, $\text{spread}(MF,G,t) = \text{spread}(MF,G_{ex}(t)) \leq n_{BV}(t) \cdot \Delta(G_{ex}(t))$, and the claim follows.

If the Secondary Step is required, Theorem 4.5 states that the number of casualties depends on the size of the still unexplored area $V_{un}(t)$ at time $t$, which could be very high. However, the total number of casualties can be very high not just for the proposed protocol. Indeed, the adversary can always force every decontamination protocol to incur $O(n)$ casualties, even with a small number of BVs (Theorem 4.3).

**Acknowledgments.** This research is supported in part by NSERC under Discovery Grants programs, and by Prof. Flocchini’s University Research Chair.

**References**


