On Curbing Virus Propagation

A. Orlitsky L.A. Shepp

AT&T Bell Laboratories

March 23, 1996

Abstract

Let A and B be two sets of computers (not necessarily inanimate). Each computer in A needs to interact once with each computer in B. The order of interaction is not specified. Each computer might be infected by a virus, hence interactions are carried out using interfaces. Conceptually, an interface has two sides and no virus can move from one side of an interface to the other. This allows for possible contagion-free interactions. Others have shown that for $|A|, |B| \geq 3$, the minimal number of interfaces needed is exactly $\left\lceil \frac{2}{3}\min(|A|,|B|) + \frac{1}{2}\max(|A|,|B|) \right\rceil$.

We consider a modification of this problem. In addition to the interactions above, each computer in A (not necessarily inanimate) wishes to interact with all other computers in A. We show that for $|A| \geq 2$ and $|B| \geq 1$ at least $\left\lceil \frac{2}{3}|A| + \frac{1}{2}|B| \right\rceil$ interfaces are needed and give an algorithm that uses exactly that many. Hence, in the original problem, all computers in the smaller of the sets A and B can interact with each other using no additional interfaces. We also consider the case of a single set A where each computer needs to interact with all others. We prove that for $|A| \geq 3$, the number of interfaces needed is precisely $\left\lceil \frac{2}{3}|A| \right\rceil$.

1 Introduction

Consider two sets of computers, A and B. Each computer in A needs to interact once with each computer in B as depicted in Figure 1. The order of interaction is not specified. Each computer might be infected by a virus, hence interactions are carried out using interfaces.

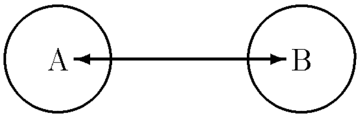

Figure 1: Each computer in A interacts with each computer in B.

Conceptually, an interface has two sides, representing its software/hardware parts that contact computers or other interface sides. When an interface side contacts a computer or another interface side, it gets infected by all viruses infecting that computer or interface side. Once an interface side is infected by a virus it stays infected by that virus. However no virus can move from one side of an interface to the other. This allows for possible contagion-free interactions.

One way to achieve all interactions with no contagion is to assign an interface to each computer. Throughout the interactions each computer contacts the same interface side, hence each interface has at most one infected side and only noninfected interface sides contact each other. This requires $|A|+|B|$ interfaces. Letting one computer interact without an interface, the number can be reduced to $|A|+|B|-1$. We show that when $|A|=3$ and $|B|=1$, all three interactions can be performed using just two interfaces. Let $A=\{a_1,a_2,a_3\}$ and $B=\{b\}$, and denote the interfaces by i_1 and i_2.

(1) a_1 interacts with b using both interfaces: a_1 contacts one side of i_1, b contacts one side of i_2, and the remaining sides of i_1 and i_2 contact each other. Following this interaction, each of i_1 and i_2 has one possibly infected side and one noninfected side.

(2) a_2 interacts with b using i_2. b keeps the same interface side while a_2 contacts the previously noninfected side.

(3) a_3 interacts with b using both interfaces: b keeps the same side of i_2 while a_3 contacts the noninfected side of i_1; the possibly infected sides of i_1 and i_2 contact each other.

These interactions are illustrated in Figure 2. Computers are represented by squares, interfaces by rectangles. A bar over an interface name (e.g., $\overline{i_1}$ in the third interaction) indicates that the interface's original orientation was reversed. A solid interface side indicates that following the interaction shown this side might be infected.

The general problem is to determine the minimum number of interfaces needed for arbitrary sizes of A and B. It was first exposed by M. Gardner [Ga 77, Ga 78] who also described the example above. Reluctant to mention computers and interfaces, his article represented

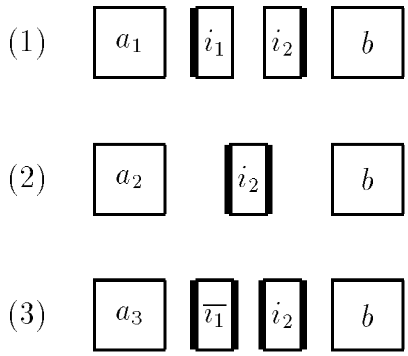

Figure 2: Computers a_1, a_2, and a_3 interact with computer b using interfaces i_1 and i_2.

A as a set of surgeons, the elements of B were patients, and the interfaces were pairs of surgical gloves.

Hajnal and Lovász [HL 78] showed that $\left\lceil \frac{2}{3}\min(|A|,|B|) + \frac{1}{2}\max(|A|,|B|) - \frac{1}{3} \right\rceil$ interfaces are needed and described an algorithm that uses just one additional interface. In their article, A was a set of infected rabbits, B was a set of radioactive plates, and the interfaces where membranes blocking radiation and viruses.

Vardi [Va 89a] closed the gap showing that, for $|A|, |B| \geq 2$, the minimal number of interfaces is exactly[1] $\left\lceil \frac{2}{3}\min(|A|,|B|) + \frac{1}{2}\max(|A|,|B|) \right\rceil$. The reader is referred to the title of that paper for the interpretation there.

In the next section we consider a deviation of that problem. In addition to the interactions above, each computer in A wishes to interact with all other computers in A. This is depicted in Figure 3. We show that for $|A| \geq 2$ and $|B| \geq 1$ at least $\left\lceil \frac{2}{3}|A| + \frac{1}{2}|B| \right\rceil$ interfaces are needed and give an algorithm that uses exactly that many. With this hindsight, it is easy to see that in the original problem the computers in the smaller of the sets A and B can interact with each other at no additional cost.

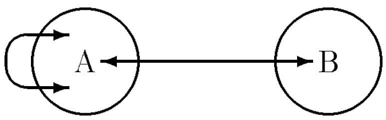

Figure 3: Each computer in A interacts with all other computers in A and all computers in B.

[1] Except for $|A| = |B| = 2$ where two interfaces suffice.

Almost as straight forward is the case of a single set A where each computer needs to interact with all others (see Figure 4). We consider this case in Section 3 and prove that for $|A| \geq 3$ the number of interfaces needed is precisely $\left\lceil \frac{2}{3}|A| \right\rceil$. We believe that this version of the problem is more relevant to computer networks. However we realize that some readers, disagreeing with that assessment, might think that this section was written tongue in cheek.

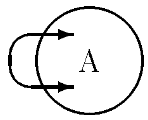

Figure 4: Each computer in A interacts with all other computers in A.

Still more general versions of the problem, with arbitrary interaction graphs, were formulated and partially solved by Vardi [Va 89b]. Finally, we caution the reader that these results are theoretical in nature. Practical objections may arise, so computer interactions may be restricted to account for probabilistic considerations as in [Ei 89].

2 Two interacting sets

Our proofs use the basic techniques of [HL 78], pursued a little further. We first describe their lower-bound method. It is convenient to assume that each computer is indeed infected by one virus and that all viruses are different.

Given a sequence of interactions and the interfaces used in each, [HL 78] created the following directed *interaction graph*. Its vertex set is $A \bigcup B$. Each edge represents an interface. Each end of the edge corresponds to one side of the interface. If a side of the interface is clean and becomes infected with a single virus then the corresponding end of the edge is connected to the vertex (computer) corresponding to that virus. Note that the side may later get infected with other viruses. If a side is clean and becomes infected with more than one virus simultaneously, or if it never gets infected, the corresponding end of the edge is connected to an arbitrary vertex. If one side is infected before the other then the edge is directed from the side that was infected first to the other side (whether or not it eventually gets infected). Otherwise, both sides get infected for the first time simultaneously or neither side is infected. In both cases, the edge is arbitrarily directed.

A *connected component* of the interaction graph is a subset of the vertices satisfying: (1)

no edge connects a vertex in the component to a vertex outside the component (2) there is a path (not necessarily directed) between any two vertices in the component.

The following Lemma is implicit in [HL 78].

Lemma 1 Let $s, t \in A \bigcup B$. If $s \to t$ is a connected component of G then all interactions sx must precede in time all interactions ty where $y \neq s$.

Proof: Suppose the interaction ty is done strictly before sx. The interface used by t in ty must be $s \to t$. Since the edge is from s to t, the s side is infected no later than the ty interaction. Since y is not s and no other interface is ever infected with s alone, the s side of $s \to t$ must have been infected by s prior to the interaction st. Hence interaction st must be done with additional additional interfaces between y and the s side of $s \to t$. Consider the interface contacting the s side of $s \to t$. This side cannot be clean and cannot be infected by s alone (in both cases there would be another edge hitting s). Therefore it must be infected by a virus different from s. During the interaction the s side of $s \to t$ gets infected by that virus and sx cannot be done. □

A connected component is *simple* if it contains no self or multiple (parallel) edges. Lemma 1 can be used to prove the next lemma. The proof is similar to that of Lemma 3 hence omitted.

Lemma 2 [HL 78] If every computer in A interacts with every computer in B then

1. A and B cannot both contain simple connected components of size 2.
2. There are at most two simple connected components of size 2 having one element in A and the other in B. □

Each computer must infect at least one clean interface hence every vertex in the interaction graph is connected to at least one edge (no *isolated* vertices). A simple connected component of size two has two vertices and one edge. In all others, the number of edges is at least two thirds the number of vertices. Hajnal and Lovász deduced that in one of the sets A and B, all elements (except, possibly, for 2) are in connected components with at least two thirds as many edges as vertices. In the other set, all elements are in connected components with at least half as many edges as vertices. Hence there are at least $2 + \frac{2}{3}(\min(|A|,|B|) - 2) + \frac{1}{2}(\max(|A|,|B|) - 2) = \frac{2}{3}\min(|A|,|B|) + \frac{1}{2}\max(|A|,|B|) - \frac{1}{3}$ edges, or interfaces, in any interaction graph for A and B.

In the current case, each computer in A needs to interact with all other computers in A hence we can prove a slightly stronger result.

Lemma 3 If each computer in A interacts with all other computers in A and all computers in B then

1. If A contains a simple connected component of size two then there are no other simple connected components of size two in the graph.

2. There are at most two simple connected components of size two having one element in A and the other in B.

Proof:

1. Suppose that there were two simple connected components of size two. One, $a_1 \to a_2$ with endpoints in A, the other $s \to t$. Then Lemma 1 applied to $a_1 \to a_2$ implies that the interaction $a_1 t$ must precede $a_2 s$. But Lemma 1 applied to $s \to t$ implies that the interaction $a_2 s$ must precede $a_2 t$.

2. Suppose that there are more than two such components. Then at least two must have direction. Say, $a_1 \to b_1$ and $a_2 \to b_2$. The argument above shows that the interactions $a_1 b_2$ and $b_1 a_2$ cannot be performed. □

With this, we can prove the lower bound for almost all cases.

Theorem 1 Let $|A| \geq 2$ and $|B| \geq 1$. The minimal number of interfaces needed for each computer in A to interact with all other computers in A and all computers in B is $\left\lceil \frac{2}{3}|A| + \frac{1}{2}|B| \right\rceil$.

Proof: Lemma 3 shows that at least $\left\lceil \frac{2}{3}|A| + \frac{1}{2}|B| - \frac{1}{3} \right\rceil$ interfaces are needed. This is the desired lower bound for all values of $|A|$ and $|B|$ except for $|A| \equiv 2 \pmod 3$ and $|B| \equiv 0 \pmod 2$ and for $|A| \equiv 1 \pmod 3$ and $|B| \equiv 1 \pmod 2$. Every algorithm for given $|A|, |B|$ can also serve for $|A| - 1, |B| + 1$. Therefore, we only need to prove the lower bound for $|A| \equiv 1 \pmod 3$ and $|B| \equiv 1 \pmod 2$. The proof is similar to that of Lemma 6 and therefore omitted here. The upper bound is proved next. □

Lemma 4 Let $|A| \geq 2$ and $|B| \geq 1$. $\left\lceil \frac{2}{3}|A| + \frac{1}{2}|B| \right\rceil$ interfaces suffice for each computer in A to interact with all other computers in A and all computers in B.

Proof: We use the notation of [HL 78]. Participating computers are denoted by P's and interfaces by I's. Each interaction consists of two computers separated by several interfaces. We describe this as a sequence. For example, $P_1\ I_1\ I_3\ P_2$ means that P_1 contacts the left side of I_1, the right side of I_1 contacts the left side of I_3, and the right side of I_3 contacts P_2.

If I is an interface then $\bar{I}$ denotes the inverted interface. The left side of I is the right side of $\bar{I}$ and vice versa. Hence it is safe to begin with $P_1\ I_1\ I_2\ P_2$ followed by $P_3\ \bar{I}_1\ P_1$ as P_3 interacts with the right side of I_1 which is still clean (it only contacted the clean left side of I_2).

We describe each algorithm as a sequence of stages. The stages have to be followed in the given order but interactions in a given stage can be performed in arbitrary order.

We describe three algorithms:

1. For $|A| \equiv 1 \pmod 3$ and $|B| \equiv 0 \pmod 2$.

2. For $|A| \equiv 2 \pmod 3$ and $|B| \equiv 1 \pmod 2$.

3. For $|A| \equiv 0 \pmod 3$ and $|B| \equiv 0 \pmod 2$.

The first algorithm can be used for $|A| \equiv 0 \pmod 3$, $|B| \equiv 1 \pmod 2$. The third algorithm can be used for $|A| \equiv 2 \pmod 3$, $|B| \equiv 0 \pmod 2$ and for $|A| \equiv 1 \pmod 3$, $|B| \equiv 1 \pmod 2$.

$|A| \equiv 1 \pmod 3$, $|B| \equiv 0 \pmod 2$.

Let $|A| = 3K+1$ and $|B| = 2L$ where $K, L \geq 1$. We describe an algorithm that uses $2K+L+1 = \left\lceil \frac{2}{3}|A| + \frac{1}{2}|B| \right\rceil$ interfaces. The algorithm has four stages. The elements of A are denoted $P_0, P_1, \ldots, P_{3K}$. The elements of B are denoted $Q_1, \ldots, Q_{2L}$. The $2K+L+1$ interfaces are I_{3k+1} and I_{3k+2} for $0 \leq k \leq K-1$, J_{2l+1} for $0 \leq l \leq L-1$, and I_0.

1. For all $0 \leq j, k \leq K-1$ and all $0 \leq l \leq L-1$ do:
   $P_0\ I_0\ I_{3k+1}\ P_{3k+1}$, $P_0\ I_0\ I_{3k+2}\ P_{3k+2}$, $P_0\ I_0\ J_{2l+1}\ Q_{2l+1}$,
   $P_{3j+1}\ \bar{I}_{3j+1}\ I_{3k+2}\ P_{3k+2}$, $P_{3k+1}\ \bar{I}_{3k+1}\ J_{2l+1}\ Q_{2l+1}$, $P_{3k+2}\ \bar{I}_{3k+2}\ J_{2l+1}\ Q_{2l+1}$.

2. For all $0 \leq j, k \leq K-1$ and all $0 \leq l \leq L-1$ do :
   $P_{3j+1}\ \bar{I}_{3j+1}\ \bar{I}_0\ \bar{I}_{3k+2}\ P_{3k+3}$, $P_{3k+3}\ I_{3k+2}\ I_0\ J_{2l+1}\ Q_{2l+1}$.

3. For all $0 \leq k \leq K-1$ and all $0 \leq l \leq L-1$ do :
   $P_{3k+1}\ \bar{I}_{3k+1}\ \bar{I}_0\ \bar{J}_{2l+1}\ Q_{2l+2}$.

4. For all $0 \le j, k \le K-1$ and all $0 \le l \le L-1$ do :
   $P_0\ \bar{I}_0\ \bar{I}_{3k+2}\ P_{3k+3}$, $\quad P_0\ \bar{I}_0\ \bar{J}_{2l+1}\ Q_{2l+2}$, $\quad P_{3j+2}\ I_{3j+1}\ \bar{I}_{3k+2}\ P_{3k+3}$,
   $P_{3k+2}\ I_{3k+1}\ \bar{J}_{2l+1}\ Q_{2l+2}$, $\quad P_{3k+3}\ I_{3k+2}\ \bar{J}_{2l+1}\ Q_{2l+2}$.

$|A| \equiv 2 \pmod 3$, $|B| \equiv 1 \pmod 2$.
Let $|A| = 3K+2$ and $|B| = 2L+1$ where $K, L \ge 0$. We describe an algorithm that uses $2K+L+2 = \left\lceil \frac{2}{3}|A| + \frac{1}{2}|B| \right\rceil$ interfaces. The algorithm has six stages. The elements of A are denoted $P_1, P_2, \ldots, P_{3K+2}$. The elements of B are denoted $Q_1, \ldots, Q_{2L+1}$. The $2K+L+2$ interfaces are I_{3k+1} and I_{3k+2} for $0 \le k \le K$ and J_{2l+1} for $0 \le l \le L-1$.

1. For all $0 \le k \le K$ and all $0 \le l \le L-1$ let P_{3k+1}, P_{3k+2}, and Q_{2l+1} use interface I_{3k+1}, I_{3k+2}, and J_{2l+1} respectively. Then perform all required interactions between them.

2. For all $0 \le j, k \le K-1$ and all $0 \le l \le L-1$ do:
   $P_{3j+1}\ I_{3j+1}\ I_{3K+2}\ \bar{I}_{3k+2}\ P_{3k+3}$, $\quad P_{3k+3}\ I_{3k+2}\ \bar{I}_{3K+2}\ I_{3K+1}\ P_{3K+1}$,
   $P_{3k+3}\ I_{3k+2}\ \bar{I}_{3K+2}\ J_{2l+1}\ Q_{2l+1}$.

3. For all $0 \le k \le K-1$ and all $0 \le l \le L-1$ do:
   $P_{3k+1}\ I_{3k+1}\ I_{3K+2}\ \bar{J}_{2l+1}\ Q_{2l+2}$, $\quad P_{3K+1}\ \bar{I}_{3K+1}\ I_{3K+2}\ \bar{J}_{2l+1}\ Q_{2l+2}$.

4. $P_{3K+1}\ \bar{I}_{3K+1}\ Q_{2L+1}$.

5. For all $0 \le k \le K-1$ do:
   $P_{3k+1}\ I_{3k+1}\ I_{3K+2}\ \bar{I}_{3K+1}\ Q_{2L+1}$.

6. Let P_{3K+2} use interface I_{3K+2}, let Q_{2L+1} use interface I_{3K+1} and or all $0 \le k \le K-1$ and all $0 \le l \le L-1$ let P_{3k+2}, P_{3k+3}, and Q_{2l+2} use interfaces I_{3k+1}, I_{3k+2}, and J_{2l+1} respectively. There is only one way each can use an interface. Then perform all remaining interaction.

$|A| \equiv 0 \pmod 3$ and $|B| \equiv 0 \pmod 2$.
Let $|A| = 3K$ and $|B| = 2L$ where $K, L \ge 1$. We describe an algorithm that uses $2K+L = \left\lceil \frac{2}{3}|A| + \frac{1}{2}|B| \right\rceil$ interfaces. The algorithm has eight stages. The elements of A are denoted $P_1, P_2, \ldots, P_{3K+3}$. The elements of B are denoted $Q_1, \ldots, Q_{2L+2}$. The $2K+L$ interfaces are I_{3k+1} and I_{3k+2} for $0 \le k \le K-1$, and J_{2l+1} for $0 \le l \le L-1$.

1. For all $0 \leq k \leq K-1$ and all $0 \leq l \leq L-1$ Let P_{3k+1} use interface I_{3k+1}, P_{3k+2} use interface I_{3k+2}, and Q_{2l+1} use interface J_{2l+1}. Then do all required interactions between P_{3k+1}, P_{3k+2}, and Q_{2l+1}.

2. For all $1 \leq j, k \leq K-1$ and all $0 \leq l \leq L-1$ do:
   $P_2\ \bar{I}_1\ \bar{I}_{3k+2}\ P_{3k+3}, \qquad P_{3k+1}\ \bar{I}_1\ \bar{I}_{3j+2}\ P_{3j+3}, \qquad P_{3k+3}\ I_{3k+2}\ I_1\ J_{2l+1}\ Q_{2l+1}.$

3. $P_3\ J_1\ Q_1$.

4. For all $1 \leq k \leq K-1$ and all $1 \leq l \leq L-1$ do:
   $P_2\ \bar{I}_2\ \bar{I}_1\ \bar{J}_1\ P_3, \qquad P_3\ J_1\ I_1\ I_{3k+1}\ P_{3k+1}, \qquad P_3\ J_1\ I_1\ J_{2l+1}\ Q_{2l+1}.$

5. For all $1 \leq k \leq K-1$ and all $1 \leq l \leq L-1$ do:
   $P_2\ \bar{I}_2\ \bar{I}_1\ \bar{J}_{2l+1}\ Q_{2l+2}, \qquad P_{3k+1}\ \bar{I}_{3k+1}\ \bar{I}_1\ \bar{J}_{2l+1}\ Q_{2l+2},$

6. $P_2\ \bar{I}_2\ Q_2$.

7. For all $1 \leq k \leq K-1$ do:
   $P_{3k+1}\ \bar{I}_{3k+1}\ \bar{I}_1\ \bar{I}_2\ Q_2$.

8. Let P_1, P_3 and Q_2 use interfaces I_1, J_1, and I_2 respectively. And for all $1 \leq k \leq K-1$ and all $0 \leq l \leq L-1$ let P_{3k+2}, P_{3k+3}, and Q_{2l+2} use interfaces I_{3k+1}, I_{3k+2}, and J_{2l+1} respectively. Then perform all remaining interactions.

□

3 One interacting set

In this section we assume that there is only one set A. Each computer in A wants to interact with all others. We show that for $|A| \geq 3$ the minimal number of interfaces is precisely $\left\lceil \frac{2}{3}|A| \right\rceil$.

Theorem 2 For $|A| \geq 3$ the minimal number of interfaces needed is $\left\lceil \frac{2}{3}|A| \right\rceil$. □

Lemma 6 proves the lower bound for $|A| \geq 5$ and Lemma 7 provides the upper bound algorithm for $|A| \geq 4$. The cases $|A| = 3$ and $|A| = 4$ can be checked individually.

Throughout this section a we use Greek letters to denote computers. A dash represents a single edge pointing in either direction. The following lemma is used in proving the lower bound.

Lemma 5 Let $\rho - \sigma - \tau$ be a connected component in the interaction graph and let β be another computer. If all four sides of $\rho - \sigma$ and $\sigma - \tau$ are already infected, then not all interactions $\beta\rho$, $\beta\sigma$, and $\beta\tau$ can be done.

Proof: We assume that each of the four sides of $\rho - \sigma$ and $\tau - \sigma$ is infected by one virus (necessarily ρ, σ, or τ). The proof is simpler if they are infected by other viruses as well. There are two possibilities:

1. The first of the three interactions is $\beta\sigma$.
   Consider the interface used by σ. Without loss of generality it is $\rho - \sigma$. Since the ρ side of $\rho - \sigma$ is infected with ρ, it must contact another interface side. Since at most one interface side is ever infected with ρ alone, this interface side must have already been infected by a virus other than ρ. Hence the ρ side of $\rho - \sigma$ gets infected by a virus other than ρ and the interaction $\beta\rho$ cannot be performed.

2. The first of the three interactions is not $\beta\sigma$.
   Without loss of generality it is $\beta\rho$. Then the interface used by ρ is $\rho - \sigma$. As before, there must be an interface side contacting the σ side of $\rho - \sigma$. There are two cases:

   (a) The contacting side of that interface is the σ side of $\sigma - \tau$. Then for the reasons above, the τ side of $\sigma - \tau$ gets infected by a virus other than τ and $\beta\tau$ cannot be performed.

   (b) The contacting side is not the σ side of $\sigma - \tau$. Then this side is never infected by σ alone. Therefore it must have already been infected by a virus other than σ. Consequently, the σ side of $\rho - \sigma$ gets infected by a virus other than σ.

   The only interface left that can be infected by σ alone or τ alone is $\sigma - \tau$ and interactions $\beta\sigma$ and $\beta\tau$ are yet to be done. Without loss of generality assume that interaction $\beta\sigma$ precedes the interaction $\beta\tau$. Then the τ side of $\tau - \sigma$ gets infected in interaction $\beta\sigma$ and interaction $\beta\tau$ cannot be performed[2]. $\square$

Lemma 6 At least $\left\lceil \frac{2}{3}|A| \right\rceil$ interfaces are required for $|A| \geq 5$.

[2]This "argument at the end of Lemma 5" is used several times later on.

Proof: Part (1) of the proof of Lemma 3 can be used to show that the interaction graph has at most one simple connected component of size 2. Therefore there are at least $1+\frac{2}{3}(|A|-2) = \frac{2}{3}|A| - \frac{1}{3}$ edges. For $|A| \equiv 0 \pmod 3$ and for $|A| \equiv 1 \pmod 3$ this is the desired lower bound.

For $|A| \equiv 2 \pmod 3$, let $|A| = 3K + 2$ where $K \geq 1$. We show that at least $2K + 2 = \left\lceil \frac{2}{3}A \right\rceil$ edges are needed. There is essentially only one graph with $3K + 2$ vertices and at most $2K + 1$ edges having no isolated vertices and at most one simple connected component of size 2. This graph consists of the following connected components: $\alpha \rightarrow \beta$ and $\rho_i - \sigma_i - \tau_i$ for $1 \leq i \leq K$ (as before, a dash represents a single edge pointing in either direction). We show that not all required interactions can be performed in this graph.

Consider the time immediately prior to the first interaction of β with any of ρ_i, σ_i, or τ_i. From Lemma 1, all interactions of α with any of ρ_i, σ_i, or τ_i, $1 \leq i \leq K$ have been performed beforehand. Therefore for all $1 \leq i \leq K$, the ρ_i side of interface $\rho_i - \sigma_i$ is infected by ρ_i and the τ_i side of interface $\tau_i - \sigma_i$ is infected by τ_i. Also, the σ_i side of at least one of $\rho_i - \sigma_i$ and $\tau_i - \sigma_i$ is infected by σ_i.

If for some $1 \leq i \leq K$, the σ_i sides of both $\rho_i - \sigma_i$ and $\tau_i - \sigma_i$ are infected then the last lemma says that the interactions $\beta\rho_i$, $\beta\sigma_i$, $\beta\tau_i$ cannot all be performed. Without loss of generality we assume that for all i, the σ_i side of $\rho_i - \sigma_i$ is infected with σ_i and the σ_i side of $\tau_i - \sigma_i$ is clean.

Consider the first interaction of β with any of ρ_i, σ_i, or τ_i. Without loss of generality, $i = 1$. There are three possibilities:

1. The first interaction is $\beta\rho_1$.

   The interface used by ρ_1 is $\rho_1 - \sigma_1$. As the σ_1 side is already infected, there must be an interface side contacting the σ_1 side of $\rho_1 - \sigma_1$.

   (a) If the contacting side is the σ_1 side of $\sigma_1 - \tau_1$ then the τ_1 side of $\sigma_1 - \tau_1$ gets infected and $\beta\tau_1$ cannot be done.

   (b) If the contacting side is any other, then (as it cannot be infected by σ_1 alone) the σ_1 side of $\rho_1 - \sigma_1$ gets infected with a virus different from σ_1. The only interface left that can be infected infected by τ alone or σ alone is $\tau - \sigma$. Although the σ side of $\tau - \sigma$ is clean, the argument at the end of Lemma 5 still works and the interactions $\beta\sigma_1$ and $\beta\tau_1$ cannot both be performed.

2. The first interaction is $\beta\sigma_1$.

   Then σ_1 uses either $\rho_1 - \sigma_1$ or $\tau_1 - \sigma_1$. If he uses $\rho_1 - \sigma_1$ then the ρ_1 side of $\rho_1 - \sigma_1$ gets infected with a virus other than ρ_1 and $\beta\rho_1$ cannot be done. Similarly if σ_1 uses $\tau_1 - \sigma_1$.

3. The first interaction is $\beta\tau_1$.

   τ_1 must use $\tau_1 - \sigma_1$ where, we assumed, the τ_1 side is infected and the σ_1 side is clean. There must be an interface side contacting the σ_1 side of $\tau_1 - \sigma_1$ (else, this side is first infected with β). There are three possibilities:

   (a) The contacting interface side is clean. Necessarily it is the σ_i side of an interface $\tau_i - \sigma_i$ where $i \neq 1$. Then the τ_i side of that interface gets infected with a virus other than τ_i and the interaction $\beta\tau_i$ cannot be done.

   (b) The contacting interface side is the σ_1 side of $\rho_1 - \sigma_1$. Then the ρ_1 side gets infected with a virus other than ρ_1 and the interaction $\beta\rho_1$ cannot be done.

   (c) The contacting interface side is infected by a virus other than σ_1 (necessarily, by other viruses too). Then the only interface safe for σ_1 and ρ_1 is $\rho_1 - \sigma_1$ and the argument at the end of Lemma 5 shows that $\beta\rho_1$ and $\beta\sigma_1$ cannot be both done. $\square$

Lemma 7 $\left\lceil \frac{2}{3}|A| \right\rceil$ interfaces suffice for $|A| \geq 4$.

Proof: A different algorithm is used for each residue of $|A|$ mod 3. We describe the algorithm for $|A| \equiv 0 \pmod 3$ and for $|A| \equiv 1 \pmod 3$. The algorithm for $|A| \equiv 0 \pmod 3$ can be used for $|A| \equiv 2 \pmod 3$. The algorithms are described using the conventions of Lemma 4.

$|A| \equiv 1 \pmod 3$:

Let $|A| = 3K + 1$ where $K \geq 1$. We describe an algorithm that uses $2K + 1 = \left\lceil \frac{2}{3}|A| \right\rceil$ interfaces. The algorithm has three stages that have to be performed in the specified order. Interactions in a single stage can be performed in arbitrary order. The participants are denoted $P_0, P_1, \ldots, P_{3K}$ and the $2K + 1$ interfaces are I_{3k+1} and I_{3k+2} for $0 \leq k \leq K - 1$, and I_0.

1. For all $0 \leq j, k \leq K - 1$ do:

   $P_0\ I_0\ I_{3k+1}\ P_{3k+1}, \qquad P_0\ I_0\ I_{3k+2}\ P_{3k+2}, \qquad P_{3j+1}\ \bar{I}_{3j+1}\ I_{3k+2}\ P_{3k+2}.$

2. For all $0 \le j,k \le K-1$ do:
   $P_{3j+1}\ \bar{I}_{3j+1}\ \bar{I}_0\ \bar{I}_{3k+2}\ P_{3k+3}$.

3. For all $0 \le j,k \le K-1$ do:
   $P_0\ \bar{I}_0\ \bar{I}_{3k+2}\ P_{3k+3}$, $\quad P_{3j+2}\ I_{3j+1}\ \bar{I}_{3k+2}\ P_{3k+3}$.

$|A| \equiv 0 \pmod 3$:
Let $|A| = 3K$ where $K \ge 2$. We describe an algorithm that uses $2K = \left\lceil \frac{2}{3}|A| \right\rceil$ interfaces. The participants are denoted by $P_1, P_2, \ldots, P_{3K}$ and the $2K$ interfaces are I_{3k+1} and I_{3k+2} for $0 \le k \le K-1$.

1. For all $1 \le j,k \le K-1$ do:
   $P_1\ I_1\ I_2\ P_2$, $\quad P_1\ I_1\ I_{3k+1}\ P_{3k+1}$, $\quad P_1\ I_1\ I_{3k+2}\ P_{3k+2}$, $\quad P_2\ \bar{I}_2\ I_{3k+1}\ P_{3k+1}$,
   $P_2\ \bar{I}_2\ I_{3k+2}\ P_{3k+2}$, $\quad P_{3j+1}\ \bar{I}_{3j+1}\ I_{3k+2}\ P_{3k+2}$.

2. For all $1 \le j,k \le K-1$ do:
   $P_2\ \bar{I}_2\ \bar{I}_1\ \bar{I}_{3k+2}\ P_{3k+3}$, $\quad P_{3j+1}\ \bar{I}_{3j+1}\ \bar{I}_1\ \bar{I}_{3k+2}\ P_{3k+3}$.

3. $P_2\ \bar{I}_2\ P_3$.

4. For all $1 \le k \le K-1$ do:
   $P_3\ I_2\ I_1\ I_{3k+1}\ P_{3k+1}$.

5. For all $1 \le j,k \le K-1$ do:
   $P_1\ \bar{I}_1\ \bar{I}_2\ P_3$, $\quad P_1\ \bar{I}_1\ \bar{I}_{3k+2}\ P_{3k+3}$, $\quad P_3\ I_2\ \bar{I}_{3k+1}\ P_{3k+2}$, $\quad P_3\ I_2\ \bar{I}_{3k+2}\ P_{3k+3}$,
   $P_{3j+2}\ I_{3j+1}\ \bar{I}_{3k+2}\ P_{3k+3}$.

$\square$

References

[Ei 89] B. Eisenberg, The number of partners and the probability of infection by the AIDS virus, to appear, Statistics in Medicine.

[Ga 77] M. Gardner, The doctors' dilemma, "Isaac Asimov's Science Fiction Magazine," Vol. 1, No. 1, 1977.

[Ga 78] M. Gardner, The doctors' dilemma, "aha! Insight," Scientific American and W. H. Freeman and Company, New York 1978.

[HL 78] A. Hajnal and L. Lovász, An algorithm to prevent the propagation of certain diseases at minimum cost, "Interfaces between computer science and operations research," edited by J.K. Lenstra, A.H.G. Rinnooy Kan, and P. Van Embde Boas, Matematisch Centrum, Amsterdam, 1978.

[Va 89a] I. Vardi, The condom problem, preprint.

[Va 89b] I. Vardi, private communication.