Isolated Template Immunity

Arthur M. Farley and Andrzej Proskurowski

CIS-TR-92-01
January 1992

Department of Computer and Information Science
University of Oregon

Isolated Template Immunity

Arthur M. Farley and Andrzej Proskurowski
Computer and Information Science, University of Oregon
Eugene, OR 97403

January 9, 1992

Abstract

We introduce the notion of isolated template immunity in communication networks. A template is simply a connected graph. A network's topology is template immune to a set of templates T if it remains connected under removal of any imbedding of an element of T. A network's operational protocol is template immune to a set of templates T if its topology is template immune and its protocol guarantees that operating sites can communicate. Isolated template immunity to a set of templates T allows multiple failures such that one imbedded template does not contain vertices or neighbors of another imbedded template. We discuss network topologies and associated protocols represented as routing tables or calling procedures, that are isolated template immune to template sets consisting of bounded length paths.

1 Introduction

The reliability of a communication network is characterized by the set of communication tasks that can be completed in the presence of certain sets of failures in the network. In this paper, we will focus on site-to-site communication tasks in store-and-forward network architectures. Each such communication task is characterized by a *message* containing the information to be communicated, an *originator*, being a site that creates the message, and a *destination*, being the site to which the originator wishes to convey the message. In a store-and-forward network, a message is transferred between two sites by a series of one or more calls. A site can place a call only to those sites to which it is directly connected; thus, if a site is to communicate with a non-neighboring site, the message must be forwarded along a route connecting the two sites.

We model a communication network by a combination of topology and protocol specifications. A network's topology is represented by an undirected graph $G = (V, E)$, whose vertices V correspond to sites of the network and whose edges E, represented as pairs of vertices (u, v), correspond to communication lines of the network. We assume the edges are undirected and correspond to lines that can be used in either direction. We will refer to the graph representing a network's topology as the network's graph.

The notions of neighborhoods, paths, and connectivity will be important in establishing our results. Two vertices are *neighbors* in a graph G if they are connected by an edge of G. We denote the set of neighbors of a vertex v, the *neighborhood* of v in G, by $N(v)$. The order of $N(v)$ is called the *degree* of v, $deg(v)$. A *path* between two vertices v_1 and v_k in a graph consists of a sequence of vertices $(v_1, ..., v_k)$, such that for each i, $0 < i < k$, the pair (v_i, v_{i+1}) is an edge of the graph. We define the *length* of a path to be the number of vertices in the path. We will refer to the general path on n vertices (of length n) as P_n. By our definition, P_1 is a single vertex.

When considering the reliability of a network, we will be concerned with connectivity properties of its graph. Two vertices are *connected* in a graph G if there exists a path between them in G. A graph is connected if and only if every pair of vertices in G is connected. Removal of certain elements of a graph can disconnect the graph. A *separator* of a graph G is a set of vertices, which, when removed from G, will yield a graph that is not connected. A *minimal separator* S of G is a separator such that no proper subset of S is a separator of G.

A set of network failures will be modeled by removal of the corresponding vertices or edges from a network's graph. For example, in the case of a site failure, the corresponding vertex and all edges incident to that vertex are removed from the graph. We refer to the graph remaining after removal of failed elements as the *operative graph*; the sites remaining are termed *operative sites.* Obviously, two operative sites can no longer successfully communicate if their associated vertices are not connected in a network's operative graph. However, two operative sites may not be able to communicate even though their vertices are connected in the operative graph if the network's operational protocol is not sufficiently resilient to encountered failures.

A network's communication protocol will be expressed in terms of routing tables and calling procedures associated with sites of the network. The essential communication primitive is the *call*, placed by one site, the *sender*, to a neighboring site, the *receiver.* A *routing table* for a site indicates, for each other site as destination, one or more possible calls that the site is to place. A *calling procedure* uses the routing table to determine which neighboring site, if any, to call next with a message that a given site has received or originated.

We characterize the reliability of a communication network by which pairs of operative sites still can communicate in the presence of a given set of failures. We will say that a network is *immune* to a given set of failures if and only if all pairs of operative sites can still communicate in the operative graph under the network's protocol. We are interested in designing efficient, immune networks. By *efficient*, we mean networks with graphs that have small size-to-order ratio (*i.e.*, the ratio of the number of edges to the number of vertices) among all networks immune to specified sets of failures.

Our presentation proceeds as follows: In Section 2 we introduce and illustrate by simple example the notions of templated immunity and isolated templated immunity, as well as a class of benchmark graphs, the k-trees. We then introduce topologies immune to isolated templated failures consisting of paths in Section 3; in Section 4, we present protocols proving those topologies immune.

2 Restricted Element Failures

In this section, we discuss a network's robustness with respect to possible multiple, but restricted, occurrences of prespecified configurations of element failures. We will introduce the notion of template immunity and then extend it to isolated template immunity.

2.1 Template Immunity

A *template* is a connected graph. An *imbedding* of a template t in graph G is a one-to-one association of all vertices of t with a subset of vertices in G, such that if there exists an edge between two vertices of t then there exists an edge between the corresponding vertices in G. A *templated failure* will be a failure of elements of a communication network that corresponds to an imbedding of a template in the network's graph. A templated failure will be said to be based upon a set of templates T if the subgraph associated with the failure corresponds to one of the templates in T. We say a templated failure *covers* the associated vertices in the graph.

A graph G is *template immune* to a set T of templates if and only if it remains connected after introduction of any possible templated failure based upon T in G. We can state the following lemma regarding graphs that are template immune, based upon definitions given above:

Lemma 2.1 *A graph G is template immune to a set of templates T (i.e., G is in Immune(T)) if and only if no templated failure based upon T in G can cover any minimal separator of G.* ∎

The class of graphs template immune to a set T of templates is referred to by Immune(T). For example, if the set T consists of a single element, the single vertex graph P_1, then Immune($\{P_1\}$) is the class of 2-connected graphs (*cf.* [4]). Minimum size examples of such graphs are the cycles. The *cycle* C_n is the unique connected graph on n ($n > 1$) vertices, such that every vertex has degree 2.

In a previous investigation of template immunity, we restricted our attention to templates consisting of a node and its k neighbors [3]. Such graphs are called *stars* and are denoted as $K_{1,k}$ graphs. The distinguished node and its neighbors are called the *center* and the *tips* of the star, respectively.

When considering failures of star subgraphs, we assumed that a failure at a selected node causes the removal of that node and k of its neighbors (or all of its neighbors, if there are less than k neighbors). This corresponds to the template set $T = \{K_{1,j} \mid 1 \leq j \leq k\}$.

In [3], we considered several classes of graphs immune to single star failures, discussing their trade-offs as to size-to-order ratios, diameters, and after-failure diameters. The graph of smallest size for given order n which remains connected after an arbitrary star failure is the cycle C_n. Such a graph has n edges and diameter of $n/2$. After failure, the operative graph is a path with either $n-3$ or $n-2$ (when $k=1$) nodes and diameter of $n-4$ or $n-3$. A *double star* consists of two stars $K_{1,(n/2)-1}$ with respective tip nodes made adjacent in a one-to-one manner. It has $3n/2$ edges and diameter 3. After a star failure, the operative graph has either $n-3$ remaining nodes (or $n-2$ when $k=1$) and diameter of 4 when a tip node corresponds to the center of the failed star, or $n-(k+1)$ remaining nodes and diameter of 2 when one of the two centers corresponds to the center of the star.

These two classes of graphs, cycles and double stars, while having relatively few edges, seem to be at opposite extremes of the trade-off between number of operative sites and diameter of the remaining operative network. A complete graph on n nodes, K_n, is immune to *any* single templated failure but requires many edges for the benefit derived.

2.2 Isolated Template Immunity

An imbedding of elements from a set of templates T in a graph G is an *isolated imbedding* if no two of the imbedded templates have associated vertices in G that are identical or adjacent. In other words, around each failure there is a buffer of operative sites; thus, the failures are topologically isolated from each other. A graph G is *isolated template immune* to a set T of templates if and only if it remains connected after removal of any possible isolated imbedding of elements of T in G. The class of graphs that are isolated immune to a set of templates T will be denoted by IsoImmune(T).

Lemma 2.2 *A graph G is in IsoImmune(T) if and only if no isolated imbedding of templated failures based upon T in G can cover a minimal separator of G.* ∎

If a graph is to be immune to isolated imbeddings of a particular set T of templates, Lemma 2.2 gives a general, minimal condition that must be satisfied. However, this condition is not particularly useful in design of the corresponding immune networks; it simply restates the definition in terms of separators. The notion of separator in a graph is obviously relevant. If we could rephrase our lemma as sufficient (if not necessary) conditions on the relationship between elements of the template set T and separators of a graph, we would have some principles to guide our search for feasible designs.

As noted above, when templated failures are isolated there is a buffer of operative sites between them . As we imbed one failure, we guarantee that certain other vertices can not be covered by other failures when failures are isolated. We will say that a given templated failure *immunizes* those other vertices in the network's graph. If the graph consisting of a single vertex is included in the set of templates, then a templated failure is only guaranteed to immunize neighboring vertices. In cases where no template is a single vertex, vertices other than neighbors may be immunized by a given templated failure (*i.e.,* there might not be "space left" to imbed a minimum diameter template). The immunization of nearby vertices suggests the following design principle, which is a sufficient condition for isolated immunity to a set of templates:

> Given a set of templates T, a graph in which any element of T that covers a vertex of a minimal separator also immunizes another vertex of the separator is immune to isolated failures from T.

One way to implement this principle is to consider network topologies in which every separator is a connected graph having more vertices than any template in T. By the following lemma, such graphs are in IsoImmune(T).

Lemma 2.3 *If every separator in a graph G is a connected subgraph having more vertices than any element of a set of templates T, then G is in IsoImmune(T).*

Proof: (By contradiction.) Assume there exists an isolated imbedding of elements of T that disconnects G. This implies that some separator S is completely covered by the imbedding. Yet, no single templated failure can cover S, as S has more vertices than any template in T. Furthermore, a template that covers part of S must immunize at least one other vertex of S, as S is a connected subgraph of G. Thus, any imbedding of templates that disconnects G can not be isolated, contradicting our assumption. ■

This principle should guide us in designing classes of efficient networks that are immune to isolated imbeddings of certain sets of templates. There is a number of ways to define efficiency. One is to choose a suitable "benchmark" class, an infinite family of graphs immune to such failures, and then define sparser, immune families. Then, we can measure efficiency in terms of the improvement in the size-to-order ratios between the sparse and benchmark graphs.

We will consider the family of graphs known as k-trees to be reasonable benchmarks for our designs. A *k-tree* can be defined recursively as either a k-complete graph (*i.e.*, a graph on k vertices, such that every vertex has every other vertex as neighbor), or as a k-tree to which a new vertex has been added by connecting it to every vertex of a k-complete subgraph. Later in this paper, we will use the known property of k-trees which is that they do not have a subgraph homeomorphic to the complete graph with k+2 vertices, *i.e.*, K_{k+2} (see, for instance, [7]). Another known property of k-trees is that every minimal separator in a k-tree induces a k-complete graph ([7]). For a given set of templates T in which the maximum number of vertices is $k-1$, a k-tree is immune to isolated imbeddings of elements of T, by Lemma 2.3. A k-tree has size-to-order ratio equal to k. That ratio will provide us with a target to improve upon in our search for efficient designs.

That a network's topology is template immune is necessary, but not sufficient, for the network to be template immune to a given set of templates. For a network to be template immune, its communication protocol also must guarantee that every pair of operative sites can still communicate.

In the remainder of the paper, we present results on network topologies that are isolated template immune to a series of template sets. After that, we will complete the network designs by presentation of operational protocols that realize the immune performance in those topologies.

3 Isolated Template Immune Topologies

In this section, we will define classes of graphs that are isolated immune to template sets containing paths of bounded lengths. How can we define an infinite class of graphs that have a certain, desired structure in their separators? A general technique we employ is that of "growing" the desired graphs from a base graph, successively adding one or more vertices in a particular configuration and connecting them to certain subsets of vertices in the current graph [5,6]. We will require that the base graph be isolated template immune and that each addition operation maintain the immunity. The recursive definition of k-trees above is an instance of a definition of this form.

A related technique will be to take two instances of immune graphs and "glue" them together by identifying a certain subset of vertices from one graph with a subset in the other. For example, a 2-tree can be defined as being either a triangle (completely connected set of 3 vertices) or two 2-trees glued together by identifying a P_2 from each graph. The identified end vertices of the composite edge become a separator in the resultant graph. That separator must satisfy certain constraints if we are to maintain isolated immunity.

3.1 IsoImmune($\{P_1\}$)

We have previously described a class of networks that is immune to isolated imbeddings of single vertex P_1 failures. In that case, we were able to match the efficiency of our benchmark graphs, 2-trees, but could not better it. In [2], we establish that any 2-tree is in IsoImmune($\{P_1\}$) and define routing tables and calling procedures sufficient to realize immune performance in the corresponding network. We will propose an alternative approache to defining routing tables later in the paper. It is easy to understand the immunity of 2-trees to isolated vertex failures. In a 2-tree, every minimal separator induces a P_2; any single vertex failure immunizes the other vertex of any separator of which it is part. Therefore, such graphs are in IsoImmune($\{P_1\}$) by Lemma 2.3.

In [2], we also note that the *prism* (see Figure 1(a)), consisting of two triangles (*i.e.*, 3-complete graphs) interconnected in a one-to-one fashion,

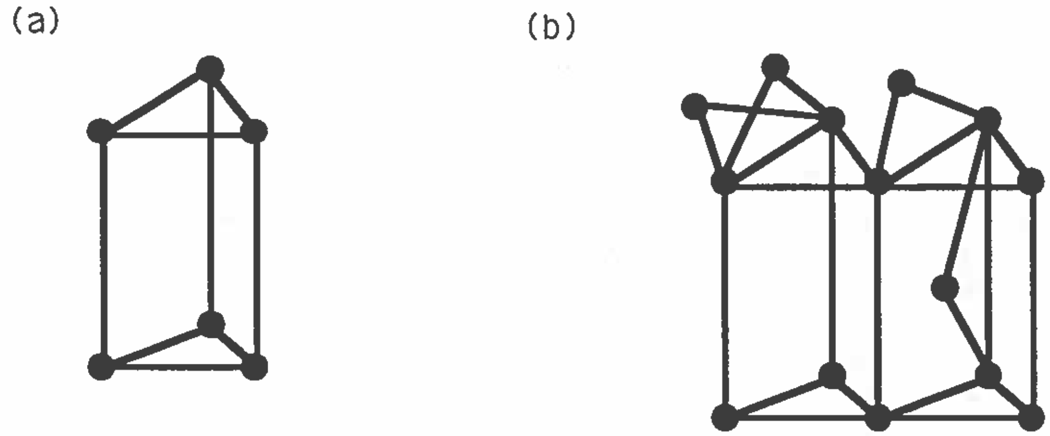

Figure 1: The prism graph (a) and a prismatic 2-tree (b)

is in IsoImmune($\{P_1\}$), as well. Let us define an extended class of graphs immune to isolated vertex failures by using the P_2-gluing operation discussed above in the context of 2-trees. A *prismatic 2-tree* is either a triangle, a prism, or the result of gluing two prismatic 2-trees together by identifying a P_2 from each graph (see Figure 1(b)). It is straightforward to see that a prismatic 2-tree is a topology that is immune to isolated failures of single vertices, as all minimal separators again contain a pair of neighbors. Prismatic 2-trees preserve the benchmark size-to-order ratio of 2-trees. It remains an open question whether they are in the class of *minimum* IsoImmune($\{P_1\}$) graphs (*i.e.*, those with fewest edges for given number of vertices) and, if the answer is affirmative, whether there are other graphs in the class of such graphs.

3.2 IsoImmune($\{P_2\}$)

We will call P_2 a *dipole*. A graph G that is immune to isolated dipole failures (*i.e.*, in IsoImmune($\{P_2\}$)) we call an *IDFI* graph. The minimum number of vertices in a separator of an *IDFI* graph is 2. If a two-vertex separator induces an edge in G, then G can be disconnected by a single dipole failure. Thus, we must consider graphs in which any two-vertex separator consists of non-adjacent vertices. The cycle graphs C_n for $n > 2$ are minimum graphs satisfying this property. While they guarantee immunity to single P_2 failures, more structure will be needed in networks that insure immunity to isolated instances of such failures.

Our benchmark class of *IDFI* graphs are 3-trees, which have minimal separators such that at least one vertex is immunized by any dipole failure. 3-trees have size-to-order ratio of 3.

Two (non-adjacent) vertices with identical neighborhoods we call *twins*. A minimal separator that is a pair of twins is useful in the following sense:

Lemma 3.1 *Only one vertex of a pair of twins can fail, given isolated P_2 failures.*

Proof: Any P_2 failure that fails one element of a pair of twins can not fail the other twin, since it fails one of the pair's common neighbors. Failure of such a vertex immunizes the other vertex of the pair. ∎

This lemma provides a basis for construction of a class of efficient *IDFI* graphs. By Lemma 2.2, if we can construct graphs whose minimal separators always contain a pair of twins, they will be in IsoImmune($\{P_2\}$).

We start with the simplest graph containing a pair of twin vertices, the cycle C_4. The two minimal separators of C_4 are pairs of twins. We continue by adding vertices in a manner preserving immunity, connecting each new vertex to both vertices of a pair of twins. We will call such graphs *twin graphs* (an example of a twin graph is shown in Figure 2(a)). We remark that twin graphs are *partial 3-trees*, *i.e.*, subgraphs of 3-trees, but in general they are not partial 2-trees (see, for instance [1]).

Theorem 3.1 *A twin graph is an IDFI graph.*

Proof: By induction on the number of new vertices added to the initial C_4 in a construction process, we show that every minimal separator contains a twin pair. Our condition is true initially, as the two separators of the C_4 are twin pairs. As we add a new vertex, all existing separators remain intact, except for those that separate vertices of the twin pair to which we connected the new vertex. The new vertex simply augments those separators which, by inductive assumption, contain twin pairs. The only new separator created consists of the two vertices to which we connected the new vertex;

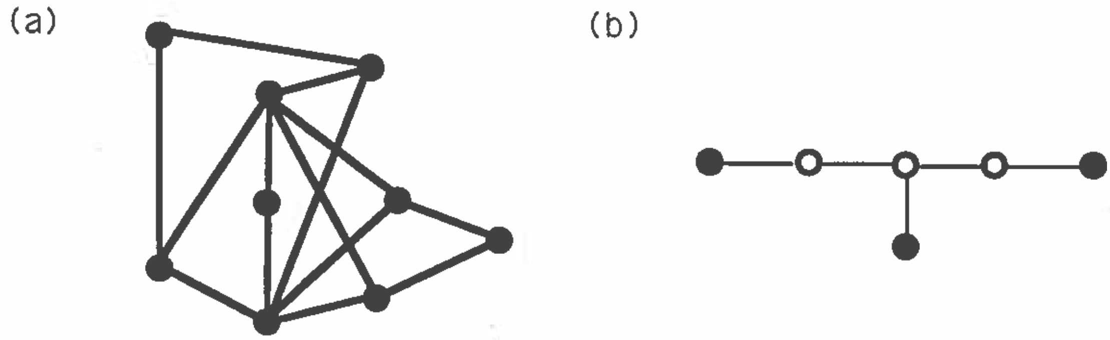

Figure 2: A twin graph (a) and its representative tree (b)

by construction, these are twins. Thus, by Lemmas 2.2 and 3.1, any twin graph is an *IDFI* graph. ∎

The recursive construction of twin graphs implies that they exhibit a tree-like structure. We call a twin pair of vertices to which a new vertex is connected during the construction process a *committed twin pair.* Thus when a new vertex is added, it is either connected to an existing, or committed twin pair or creates a new such pair.

Lemma 3.2 *Given a twin graph G, any vertex of degree larger than 2 is in a unique, committed twin pair.*

Proof: Consider a vertex u of G such that $deg(u) > 2$. Let G' be a subgraph of G corresponding to the time instant in the recursive construction of G immediately after the third neighbor w of u has been added. By definition, w was then made adjacent to another vertex v of G' such that, in G', $N(u) = N(v)$. Since no vertex in $V(G')$ other than u and v is adjacent to w, no other vertex is a twin of u in G'. Since u has degree 3 in G', no vertex added later (*i.e.*, in $V(G-G')$) can be made adjacent to all neighbors of u, as only two of them could be the twin pair to which a new vertex is made adjacent. Thus, $\{u, v\}$ is the unique twin pair containing u in G. ∎

Given the uniqueness of committed twin pairs, we can define the *representative tree* of a twin graph with internal vertices representing committed twin pairs of G, external vertices (leaves) representing the degree 2 vertices of G, and edges reflecting the definitional relation between added vertices and their adjacent twin pairs. Since both vertices of a committed twin pair were either in the original C_4 or connected to the same twin pair in the construction process, the above defines the edges between committed twin pairs, as well. The representative tree of a twin graph is given in Figure 2(b), with nodes representing committed pairs shown as circles. We will exploit the notion of representative tree in Section 4 when defining an immune protocol for twin graphs.

The number of edges in a twin graph having n vertices is $2n$, for the size-to-order ratio of 2, which is an improvement over our benchmark class of 3-trees. By relaxing our condition on neighborhoods of vertices in a separator, we will be able to construct *IDFI* graphs with even fewer edges for a given number of vertices.

Lemma 3.3 *Let x and y be two vertices in a graph G. If the sets $X = N(x)-N(y)$ and $Y = N(y)-N(x)$ are such that either both are empty, only one is empty, or every vertex in X is connected to every vertex in Y and vice versa, then x and y can not both fail by the introduction of isolated P_2 failures in G.*

Proof: If both X and Y are empty, we have the situation that exists in twin graphs. When only one set is empty, we can assume without loss of generality that X contains a vertex u. If the dipole (x, u) fails, this immunizes all other neighbors of x, which (since we assume $Y = N(y) - N(x) = \emptyset$) includes all neighbors of y. Therefore, y can not fail by introduction of isolated dipole failures in G, as a dipole failure involving y must involve one of its neighbors. A third case remains. Let u be an element of X and v an element of Y. A failure of the dipole (x, u) would immunize all vertices in Y (since they are adjacent to u) and also all other vertices in $N(y)$ (since they are adjacent to x), thus protecting y from failure, as above. Similarly, a failure of dipole (y, v) immunizes all vertices in $N(x)$, indirectly protecting x. So, there would be no way to fail both elements of $\{x, y\}$ when dipole failures are isolated. ∎

The above result can be used to construct *IDFI* graphs with fewer edges than twin graphs. We again start with a C_4. We add 3 new vertices to a current graph, connecting them to a twin pair, x and y, as follows: one new

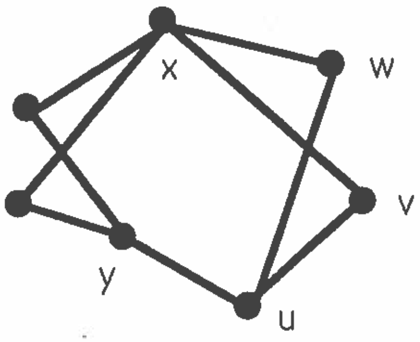

Figure 3: An augmentation in a wing graph

vertex, u, is connected to y, and two other new vertices, v and w, to x; u is connected to both v and w. This construction operation is illustrated in Figure 3. We call such graphs *wing graphs*. Wing graphs have the asymptotic size-to-order ratio of 5/3, as each vertex addition operation adds 3 new vertices and 5 new edges. We will prove that wing graphs are *IDFI* graphs, thus improving upon the size-to-order ratio of 2 associated with twin graphs.

Theorem 3.2 *A wing graph is an IDFI graph.*

Proof: We show that every separator of a wing graph contains two non-adjacent vertices that satisfy the condition on neighborhoods presented in Lemma 3.3. If the condition on neighborhoods is met, then the graph is in *IDFI* graph, by Lemma 2.2.

Our proof will be by induction on the number of vertex additions performed in the construction process. The initial C_4 meets our condition, as discussed above. Assume a wing graph satisfies our condition on neighborhoods after k addition operations. Let us consider the graph produced by the $(k+1)$th operation. Adding the new vertices, u, v, w to a twin pair $\{x, y\}$ (as in Figure 2(b)), creates several new separators in the graph. Consider the new separator (u, x). $N(u) - N(x) = \{y\}$, while $N(x) - N(u)$ consists of the neighbors of x other than v and w. As x and y were twins in the prior graph, y is adjacent to all of those other neighbors of x. Thus, our condition on neighborhoods is met for that separator. The set $\{y, v, w\}$ separates u from the rest of the graph. Since v and w form a twin pair, our condition is met again. The two new separators, separating x and y, respectively, from the rest of the graph, extend (by inclusion of u or v and w) former separators already satisfying the condition. Finally, the previous twin pair $\{x, y\}$ itself, which may have been a separator, satisfies the conditions of Lemma 3.3, as our construction directly reflects the condition. Thus, all of the new separators, and so the new graph, satisfy our condition. ∎

3.3 IsoImmune($\{P_1, P_2\}$) Networks

We have defined prismatic 2-trees and twin graphs, which are immune to isolated instances of isolated single vertex (P_1) and dipole (P_2) failures, respectively. Here, we consider networks in which both of these failures may occur. Let the *wheel graph* W_n on n vertices be the graph having $n-1$ vertices connected in a cycle C_{n-1}, called the *rim*, and a single vertex, called the *hub*, made adjacent to all vertices on the rim (*cf.* Figure 4(a)). By definition, W_n has $2n-2$ edges, its size-to-order ratio of 2 being better than the ratio of 3 associated with our benchmark 3-tree graphs.

Theorem 3.3 *The class of wheel graphs is exactly the class of minimum size graphs in IsoImmune($\{P_1, P_2\}$).*

Proof: It is easy to see that W_n is in IsoImmune($\{P_1, P_2\}$). No single failure can disconnect the graph. If the hub is involved in a failure, only that one (isolated) failure can occur; failure of the hub vertex immunizes all other vertices. If the hub is not involved, any failure immunizes the hub; all operative vertices remain connected to each other through the hub.

To prove that the wheel graphs constitute exactly the class of minimum size IsoImmune($\{P_1, P_2\}$) graphs, we first establish that the minimum size for a graph in this class is $2n-2$, for graphs of order n. We see that W_4 is isomorphic with the complete graph on 4 vertices and is the unique minimum size immune graph of order 4. Assume that G has the minimum order among all immune graphs on $n > 4$ vertices with fewer than $2n-2$ edges. G can have no vertex of degree 2 since adjacent neighbors of such a vertex can fail in a P_2 failure and non-adjacent neighbors can fail in two isolated P_1 failures. Thus G must have a vertex u of degree 3 since, if all vertices were of degree at least 4, G would have at least $2n$ edges. The neighborhood of u in G contains a path, lest two isolated P_1 and P_2 failures disconnect u from G.

Without loss of generality, assume the path is (x, y, z). We claim that the graph G' formed by removing vertex u from G and adding the edge (x, z) is also in IsoImmune(P_1, P_2). This is so because any isolated imbedding of templates in G' that would disconnect G' would also disconnect G (if (x, z) is not an edge in G, then its failure in G' can be simulated by isolated failures of x and z in G). Thus, G' is in IsoImmune(P_1, P_2). G' has $n-1$ vertices and fewer than $2n-2-2 = 2(n-1)-2$ edges, as 3 edges were deleted from G and at most one added. Therefore, n is not the minimum order for an immune graph having few edges. By contradiction, no such graph exists.

Now we show that no other graphs are minimum IsoImmune($\{P_1, P_2\}$). Our argument is similar to that above. By inspection, W_4 is the only graph on 4 vertices that is in the class. Let k be the least k ($k > 4$), such that W_k is not the only graph with k vertices and $2k-2$ edges in IsoImmune($\{P_1, P_2\}$). Let G be one of these other graphs having k vertices. G must contain at least one vertex u of degree 3, as argued above. The neighborhood of u contains a P_3, call it (x, y, z), as above. We remove vertex u and add an edge between the two non-adjacent vertices x and z of the P_3. This reduces the number of vertices by 1 and number of edges by 2. The resultant graph G' is in IsoImmune($\{P_1, P_2\}$), as argued above. By our assumption, G' must be W_{k-1}, as k is the least number of vertices for which a minimum size member of IsoImmune($\{P_1, P_2\}$) is not a wheel.

There are two cases to consider for the position of the added edge (x, z) in G'. The triangle (x, y, z) resulting from the reduction must involve the hub of W_{k-1} (since $k > 4$) and two adjacent vertices of degree 3. However, for $k > 5$, the added edge (x, z) could not be incident to the hub, as the original graph G would not be immune (isolated failures of x and a dipole involving z would disconnect G). Thus, the removed vertex u was a rim vertex of the wheel graph W_k (by inspection, this is also true for $k = 5$). Therefore, G was a wheel graph. This contradicts our assumption that there exists a minimum order IsoImmune($\{P_1, P_2\}$) graph that is not a wheel. ■

3.4 IsoImmune($\{P_1, P_2, ..., P_k\}$)

We can generalize our notion of wheel graphs to provide a class of graphs that are immune to isolated imbeddings of longer path failures. We define the

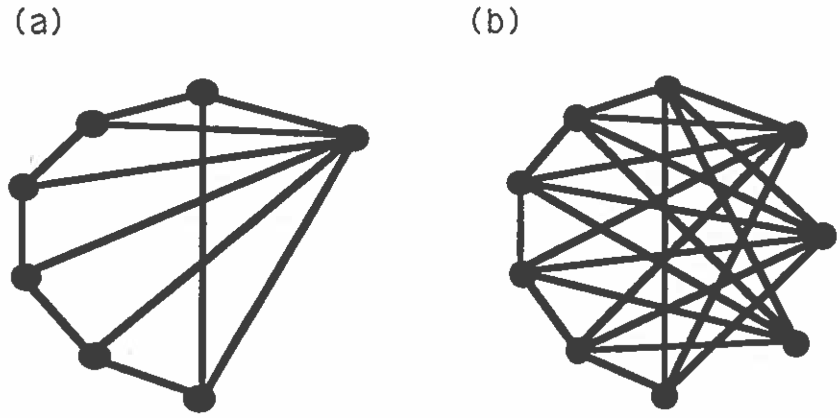

Figure 4: The wheel W_7 (a) and the 3-centered wheel graph ${}_3W_{10}$ (b)

class of multi-centered wheels, as follows. A *k-centered wheel* on n vertices ${}_kW_n$ ($n > k+1$) consists of a hub, being a set of k independent (*i.e.*, mutually non-adjacent) vertices, and a rim, being all other $n-k$ vertices connected in a cycle, such that each vertex on the rim is also adjacent to every vertex of the hub. Figure 4(b) presents a ${}_3W_{10}$ graph.

Theorem 3.4 *The ${}_kW_n$ graph is in IsoImmune($\{P_1, P_2, ..., P_{2k-2}\}$), for $n > 3k-2$ and $k > 2$.*

Proof: We have k vertices in the hub and more than $2k-2$ vertices on the rim. To disconnect a vertex on the rim, we must fail all hub vertices plus at least two vertices on the rim, all in one failure. To disconnect a vertex in the hub requires that all vertices on the rim fail, also in one failure. Either of these situations would require a path of length $2k-1$. If a failure involves any vertex of the hub, all remaining vertices on the rim are immunized. If a failure involves a vertex on the rim, then all remaining vertices of the hub plus neighbors of the failure on the rim are immunized. As such, we see that a single failure from the set of available templates cannot disconnect the graph and immunizes other elements sufficient to guarantee immunity for the graph. ■

The number of edges in a ${}_kW_n$ graph is $n-k$ on the rim, plus $(n-k)k$ connecting the rim to the hub, for a total of $(k+1)(n-k)$. This indicates

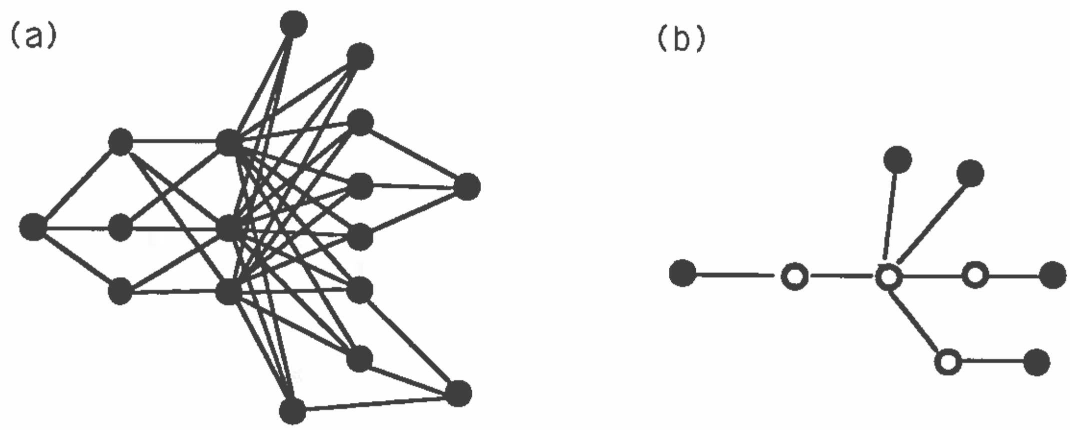

Figure 5: A 3-sibling graph (a) and its tree representation (b)

a size-to-order ratio of $k+1$ for networks in IsoImmune($\{P_1, P_2, ..., P_{2k-2}\}$), for $k > 2$. This is significantly less than the size-to-order ratio of $2k-1$ associated with our benchmark $(2k-1)$-trees, for paths of length $2k-2$.

3.5 IsoImmune($\{P_2, ..., P_k\}$) Networks

In this section, we discuss networks that are immune to isolated path failures up to some length limit k, this time without the possibility of single vertex failures. Without single vertex failures, immunization can reach beyond direct neighbors, as we have seen before. The notion of twin pair used in Section 4.2 can be generalized to that of an *s-sibling set*: a set of s independent (*i.e.*, mutually non-adjacent) vertices having identical neighborhoods of order at least s. By an iterative construction procedure, we can form tree-like graph structures, having s-sibling sets as separators.

An s-sibling tree on n vertices (for $n \geq 2s$) is either the complete bipartite graph on $2s$ vertices ($n = 2s$) or is formed by connecting a new vertex v to an s-sibling set of an s-sibling tree having $n-1$ vertices ($n > 2s$). Each s-sibling tree on n vertices has $s(n-s)$ edges. Figure 5 presents an example of a 3-sibling tree together with a tree representation reflecting the structure of such a graph (committed 3-sibling sets are represented by open vertices). The recursive definition leads to an easy inductive proof of the following lemma.

Lemma 3.4 *Every separator of an s-sibling tree contains an s-sibling set.* ∎

If we are to provide immunity to failures of paths up to length k, all vertices of a sibling set separator must not be covered by a single failure. Thus, the number of siblings in a separator must be greater than $(k+1)/2$.

Theorem 3.5 *An s-sibling tree on n vertices is in IsoImmune*$(\{P_2, ..., P_k\})$, *for any s greater than* $(k+1)/2$.

Proof: By Lemma 4.3, every separator contains an s-sibling set and thus has more than $(k+1)/2$ independent vertices. Hence, no separator can be covered by a single P_k failure. Each failure immunizes at least one vertex of any sibling set involved in the failure. Thus, at least one vertex of every separator is operational in any set of isolated P_k failures. ∎

The s-sibling trees are efficient designs for IsoImmune$(\{P_2, ..., P_k\})$ in that they only require a size-to-order ratio of $(k+1)/2$, while our benchmark class of $(k+1)$-trees would require a ratio of $k+1$.

4 Isolated Template Immune Protocols

In the preceding sections, we have defined topologies that are immune to isolated failures drawn from several different template sets. To complete definition of an immune network, we must specify a protocol that realizes message transfers among the remaining operative sets. Communication protocols will be defined in terms of routing tables and calling procedures associated with sites of the network. We assume that each site can perform a primitive operation *call*(x, m), sending a message m to a neighboring site x. This operation can succeed, with the effect of x receiving message m, or fail due to failure of x or the line to x, with x receiving no message.

To realize immunity to possible failures, each site must have more than one call it can place for each possible destination of the message. When it is sufficient to make at most one alternate call, we can define two routing table entries, *preferred*(x, y) and *alternate*(x, y), each indicating a neighboring site that x is to call given y as ultimate destination. Our basic calling protocol $\mathcal{P}$ for site x given message m and site y as destination (assuming x is not y) is defined then simply as follows:

if *call*(*preferred*(x, y), m) fails **then** *call*(*alternate*(x, y), m).

We will create the necessary preferred and alternate routing tables during the iterative construction process of the immune 2-tree and twin graph topologies, for which protocol $\mathcal{P}$ applies without modification. We then provide immune protocols for the other classes of immune graphs defined above.

4.1 Routing Tables for 2-Trees

We now consider immunity to isolated P_1 failures. In Section 3.1, we defined two schemes for constructing efficient IsoImmune($\{P_1\}$) graphs. Here, we define routing tables for 2-trees that are sufficient for immunity under the basic protocol $\mathcal{P}$. A similar protocol can be defined for prismatic 2-trees, but is more complex, requiring *two* alternate calls in some situations.

In [2], we discuss the creation of preferred and alternate routing tables when converting a spanning tree of sites into a 2-tree. Here, we define the routing tables in terms of the construction process for a 2-tree. Although a particular 2-tree can be realized by many different vertex addition sequences, we will consider here an arbitrarily chosen sequence as defining *the* iterative construction referred to below. The routing tables for sites of the network are created during the iterative construction of the 2-tree as follows:

In the base case of a 2-tree on two vertices, the corresponding sites are direct neighbors of each other. The preferred and alternate calls from each site for the other as destination are both to the other site. This will be the case for every pair of neighbors as we build the 2-tree and its routing tables. By this paradigm, a neighbor of a failed destination will call that site twice and the protocol will halt, albeit unsuccessfully.

As we add a new vertex w to a 2-tree network graph G, connecting it to two neighboring vertices x and y, we set w's preferred and alternate calls to be x and y, respectively, for all sites other than x and y as destination. Thus, for a vertex $s \notin \{x, y, w\}$, *preferred*(w, s) $= x$ and *alternate*(w, s) $= y$. As noted above, the preferred and alternate calls from w for x or y as destination will be both to the destination site. Similarly, x and y will place both preferred and alternate calls to w for w as destination.

We must now route calls from sites other than x and y with w as destination. First we consider the routing of all other sites s (if any) having both x and y as neighbors. These sites will call x as preferred and y as alternate for w as message destination. Formally, for vertex $s \in N(x) \cap N(y)$, *preferred*$(s, w) = x$ and *alternate*$(s, w) = y$.

To route calls to w from vertices of G adjacent only to one of its neighbors, x or y, we need to consider the iterative construction process of G. Without loss of generality, let us assume that in this process, when the vertex y was added it was made adjacent to x and its neighbor z. In any 2-tree G, neighbors of any vertex x induce an acyclic subgraph of G (otherwise, the cycle and three edges incident with x would induce a subgraph homeomorphic with K_4). Therefore, z together with vertices of G adjacent to x but not to y induce a tree. For such a vertex s, we set *preferred*$(s, w) = x$ and *alternate*$(s, w) = s'$, where s' is the neighbor of s on the path to z in the induced tree. Similarly, for a vertex s of G that is adjacent to y but not to x, *preferred*$(s, w) = y$ and *alternate*$(s, w) = s''$, where s'' is the neighbor of s on the path to z in the induced tree.

Finally, for sites outside neighborhoods of both x and y, we let the preferred and alternate calls for w as destination be the same as their preferred and alternate calls for x as destination. Thus, for vertex $s \notin N(x) \cup N(y)$, *preferred*$(s, w) =$ *preferred*(s, x) and *alternate*$(s, w) =$ *alternate*(s, x).

Lemma 4.1 *Protocol $\mathcal{P}$, in conjunction with the routing tables defined above, are sufficient to provide immunity to isolated P_1 failures in 2-trees.*

Proof: Our proof is by induction on the number of vertices added during the iterative construction of a 2-tree. Let G be the 2-tree existing after k vertex additions. When $k = 0$ (G is a 2-tree of 2 vertices) the protocol is trivially sufficient. We assume that it is sufficient in G for some positive k, and consider additions to the routing tables for the $(k+1)$th vertex addition of w adjacent to end vertices x and y of an edge in G. Any message from w will be able to reach either x or y when failures are isolated. By inductive assumption, these vertises can communicate with all other operative vertices. Also by inductive assumption, a message from any site s of G routed through x can get to an operational site in the neighborhood of x. If x is operational, the message will be routed through that vertex to w. Let us assume that x is no longer operational; thus, all of its neighbors, including y are immunized. If a call intended for w arrives at a neighbor of both x and y, our routing tables provide an immune call path through y. If it arrives at a vertex that is a neighbor of x but not of y, the routing table will direct the message along a path in the tree of neighbors of x toward a site that is also a neighbor of y; this provides the necessary call path through y to w.

Thus, using protocol $\mathcal{P}$, our routing tables are sufficient to realize immune performance. ∎

A vertex of a prismatic 2-tree might require three calls to realize immune communication when two of its neighbors in a prism fail. The two alternate call directions for each of the four vertices of a "glued" prism can be defined easily when the prism is used in the iterative construction process.

4.2 Routing Tables for Twin Graphs

We now consider the case of immunity to isolated P_2 failures. In Section 3.2, we have defined two schemes for constructing efficient *IDFI* graphs. Here, we will define routing tables for twin graphs. A protocol similar to the basic protocol $\mathcal{P}$ can be defined for wing graphs, but it will be more complex.

As with the 2-trees above, we will create routing tables such that our basic calling procedure $\mathcal{P}$ can complete message transfers in twin graphs successfully in the presence of isolated P_2 failures. We will update routing tables as we add each vertex to a twin graph, as done above for 2-trees.

For neighbors x and y, we will have $\textit{preferred}(x, y) = \textit{alternate}(x, y) = y$. Thus, if a site has failed and the message reaches an operative neighbor, that neighbor will try to call the site twice (failing both times) and the protocol will halt in failure.

In the initial C_4, a vertex with a message for its only non-neighbor as destination calls one neighbor as preferred and the other as alternate. Consider adding a new vertex w to the twin pair $\{x, y\}$; we must determine routing table information with w as originator of a message for existing sites as destination (other than $\{x, y\}$, since they are neighbors of w, with calls defined as above) and with all other existing sites as originator of a message for w as destination. For site s other than x and y, we define $\textit{preferred}(w, s) = x$ and $\textit{alternate}(w, s) = y$. For a neighbor s of x and y (other than w), we define $\textit{preferred}(s, w) = x$ and $\textit{alternate}(s, w) = y$. For site s not a neighbor of x, we define $\textit{preferred}(s, w) = \textit{preferred}(s, x)$ and $\textit{alternate}(s, w) = \textit{alternate}(s, x)$.

Lemma 4.2 *Protocol $\mathcal{P}$, in conjunction with the routing tables defined above, are sufficient to provide immunity to isolated P_2 failures in twin graphs.*

Proof: Our proof is by induction on the number k of vertices added to the twin graph beyond the base C_4. Let G be a twin graph after k vertex additions. When $k=0$, we see that $\mathcal{P}$ and the routing tables are sufficient. Let k be some positive integer, and consider the results after addition of the $(k+1)$th vertex w to a twin pair $\{x, y\}$ of G. If the message transfer is between w,x,y, or another neighbor of that twin pair, obviously the transfer is immune. Any message from w will reach either x or y from where, by inductive assumption, all other operative vertices can be reached.
By our inductive assumption, the protocol and routing tables of G provide that a message from any other vertex of G through x can reach the neighborhood of x in the presence of isolated P_2 failures. By Lemma 3.2, once a twin pair becomes committed by addition of a third common neighbor, the pair will always have identical neighborhoods. Since, by Lemma 3.1, vertices of a twin pair can not fail simultaneously when P_2 failures are isolated, immunity is guaranteed for any message transfer in the augmented graph G. Construction of the routing tables implies that a message transfer in a twin graph G is reflected by a progress along the corresponding path in the representative tree of G, ensuring termination of the process. ∎

4.3 Immune Protocol in Wheels

When we considered a set of templates that consists of both single vertex and dipole failures, $\{P_1, P_2\}$, we found that wheel graphs are the minimum-size immune topology (Section 3.3). To complete our definition of an immune network, it remains to specify an immune communication protocol for sites of such a network.

There are two classes of sites to consider in a wheel graph, the hub site and rim sites. The hub site will have a different protocol than rim sites. Since it is adjacent to every other site, the hub site simply calls the destination as its preferred and alternate call.

For every rim site, its preferred call is to the hub site, for any other site as destination. As alternate routing for rim sites, there are two possible call

```
if originating the message
then if call(hub,m) fails
       then if call(alternate1,m) fails
              then call(alternate2,m)
else {forwarding the message}
       if received message from alternate1
       then if call(alternate2,m) fails
              then call(alternate1,m)
              else if call(alternate1,m) fails
                     then call(alternate2,m)
```

Figure 6: Protocol $\mathcal{P}'$ for rim sites of a wheel graph.

directions. Since a P_2 failure may have affected the hub and one of its rim neighbors, two alternate calls, *alternate1* and *alternate2*, must be available for every rim site. The alternate calls are to the site's two neighbors on the rim.

We define a new protocol $\mathcal{P}'$ for rim sites of a wheel graph in Figure 6. When a rim site originates a message transfer, it first calls its preferred destination, the hub (*preferred*$(x, y) = z$ for rim vertices x, y and a hub z). If the hub site is down, then the originator places a call to a neighbor on the rim. If that call fails and the receiver was not the destination, the rim site places a third call to its other neighbor on the rim. This third site must be operational since the hub site has failed.

When forwarding a message, a rim site must have received the message from a neighbor on the rim (the hub must be down). As its alternate action, the site calls its other neighbor on the rim. If that site is down and is not the destination, the forwarding site calls the neighbor on the rim from which the message was received, thereby returning the message to traverse the cycle (now reduced to a path of operational sites) in the opposite direction to the destination.

This new protocol provides the immune communication behavior made possible by the wheel graph topology.

4.4 Immune Protocols in Other Classes of Graphs

When we add paths of length up to $2k-2$ ($k > 2$) to our set of templates, we find that *k-centered wheels* ${}_kW_n$ provide a class of immune topologies. Immune communication protocols for the ${}_kW_n$-based networks are straightforward generalizations of the protocol defined above for wheels. Each site on the rim now has a list of k hub sites to call (instead of only one). A site on the rim, when originating a message, calls the message destination first, if it is a neighbor. Otherwise, it starts calling hub vertices until it succeeds (as it must by our proof Theorem 3.4). That hub site forwards any message directly to its final destination on the rim, as all hub sites can not fail when failures are isolated. The only difficulty remaining for the protocol is a hub site originating a message for another hub site. The originator will have to call at most $2k-1$ sites on the rim before finding one that is operative, which site can then forward the message directly to the hub site destination. This generalized protocol produces the immune communication behavior desired.

It remains to define immune communication protocols for k-sibling trees which provide immunity to sets of paths of length at most $2k$–2, not including the single vertex template. Consider a vertex x in an k-sibling tree trying to send a message to a vertex y. Either the two sites are neighbors and x can call y directly, or the neighborhood of x contains a unique k-sibling set in its neighborhood separating x from y. This is due to the tree-like structure of the sibling-set separators of the graphs (see Figure 5). Therefore, the following protocol provides the immune behavior desired: a site calls the destination of a message if it is a neighbor; otherwise, the site calls members of the unique, k-sibling set neighborhood lying between it and the ultimate recipient, until a call is successful. One call is guaranteed to be successful if failures are isolated in the network.

5 Conclusion

In this paper we have described efficient designs for networks that are immune to isolated occurrences of path failures of bounded lengths. Our designs include not only specifications of topology but also specifications of communication protocols that are sufficient to realize immune communication performance. While efficient, most of our designs are not known to be minimum. Open questions remain as to the determination of minimum designs for the cases considered here, other than the wheel graphs which are shown to be the minimum elements of IsoImmune($\{P_1, P_2\}$).

References

[1] Arnborg, S. and A. Proskurowski, Linear time algorithms for NP-hard problems restricted to partial k-trees, *Discrete Applied Mathematics 23* 1(1989), 11-24.

[2] Farley, A.M., Networks immune to isolated failures, *Networks 11* (1981), 255-268.

[3] Farley, A.M. and A. Proskurowski, Immunity to subgraph failures in communication networks, *Congressus Numerantium 62* (1988), 149-158.

[4] Harary, F., *Graph Theory*, Addison-Wesley, 1969.

[5] Klawe, M., D.G. Corneil and A. Proskurowski, Isomorphism testing in hookup classes, *SIAM Journal on Algebraic and Discrete Methods 3* 2(1982), 260-274.

[6] Proskurowski, A., Recursive graphs, recursive labelings, and shortest paths, *SIAM Journal on Computing 10* 2(1981), 391-397.

[7] Rose, D., On simple characterization of k-trees, *Discrete Mathematics 7* (1974), 317-322.