Towards Attack-Tolerant Networks: Concurrent Multipath Routing and the Butterfly Network

Edward L. Platt
University of Michigan
elplatt@umich.edu

Daniel M. Romero
University of Michigan
drom@umich.edu

ABSTRACT

Networks with single points of failure are particularly susceptible to targeted attacks. In communication networks, these types of faults can leave users vulnerable to censorship and targeted surveillance, even when cryptography is utilized. Centralized networks have single points of failure by definition, leading to a growing popularity in decentralized architectures and protocols. However, centralized network structure can arise even when protocols are decentralized. While based on decentralized protocols, the Internet and World-Wide Web have been shown both theoretically and historically to be highly susceptible to adversarial faults, in part due to emergent, structural centralization. Existing network trust models, such as webs of trust, fail to adequately address network structure. We describe a novel, adversarial fault-tolerant, concurrent multipath routing algorithm for the decentralized butterfly network topology. We also develop a partial trust model that makes it possible to quantify the adversarial fault tolerance of a network, while also making more realistic transitivity assumptions than webs of trust. When network topology can be dictated, these results can be used to create scalable, attack-tolerant infrastructures. More generally, our results provide a formalism for evaluating the effects of network structure on adversarial fault tolerance.

CCS Concepts

•Security and privacy → Formal methods and theory of security; •Networks → Network architectures; •Software and its engineering → Peer-to-peer architectures;

Keywords

Butterfly Network; Fault Tolerance; Adversarial Faults; Multipath Routing; Censorship; Decentralization

1. INTRODUCTION

WWW '17 April 3–7, 2017, Perth, Australia

DOI: 10.1145/1235

Large-scale communication networks, exemplified by the Internet, have become ubiquitous, and are now crucial infrastructure for individuals, communities, and organizations around the world. As with any critical infrastructure, the cost of a failure can be immense, so methods for tolerating various kind of faults are an important and ongoing area of research [46, 1, 40]. In fact, the decentralized design of the Internet was motivated by the need to withstand nuclear strikes [6]. *Adversarial faults*, in which an adversary can strategically target attacks, are some of the most difficult to protect against. Such attacks are often used for censorship and targeted surveilance. Both theoretical results and recent events (described below) have demonstrated that the Internet is surprisingly vulnerable to adversarial faults. Decentralization remains a promising approach to building resilient networks, but further work is needed to understand the relationship between decentralized network structure and adversarial fault tolerance.

Analysis of the Internet's router network has shown that while it is remarkably resilient against random faults, it is highly susceptible to adversarial faults [1]. These results have been attributed to the scale-free structure of the Internet's router network [4, 5]. In scale-free networks and other networks with heavy-tail degree distributions, random failures are highly likely to affect only low-degree nodes, thus having little effect. However, Adversarial faults target the few high-degree nodes, and therefore remove a large number of edges with each fault. So while the *protocols* of the Internet are decentralized, the *network structure* is somewhat centralized. In other words, the protocols of the Internet do not *require* centralization, but centralization may still emerge from the sociotechnical processes that create its network structure.

The Internet's vulnerability to censorship and other targeted attacks has been confirmed by the success of several such attacks. For example, in 2008, YouTube suffered a worldwide outage for several hours when a service provider in Pakistan advertised false routing information [19]. The action (known as a *black hole attack*) was intended to censor YouTube within Pakistan only, but resulted in a worldwide cascading failure. Such vulnerabilities are not limited to any one system or protocol, but result from the network structure itself. With strong theoretical and historical evidence that network structure can create vulnerabilities, methods for analyzing structural vulnerabilities and for designing fault tolerant networks are needed. This paper presents several contributions towards advancing those goals.

We consider a setting in which a source node (Alice) in a network attempts to route a message to a target node (Bob) by forwarding it through the links of the network. We assume that some nodes in the network may be compromised by an attacker (Mal). Compromised nodes may behave incorrectly by blocking, altering, or incorrectly routing messages. We assume that Mal has full knowledge of the network structure, but has limited resources and thus can only compromise a fixed number of nodes. We also assume that nodes within a fixed network distance of either Alice or Bob are *trusted* and cannot be compromised. This assumption, which we call *partial trust transitivity*, is a weaker (more realistic) form of the transitive trust assumption used by the popular web of trust approach [45, 15].

Under the above assumptions, we show how to evaluate the influence of network structure on attack-tolerance, and propose a method for achieving a high-level of attack-tolerance on the butterfly network topology. The butterfly topology is popular in parallel processing [26] and peer-to-peer [31, 25] applications, due to its regular structure and high connectivity. While it may seem questionable to dictate the topology of trust relationships, there are many applications where this occurs, e.g., overlay networks [31, 25], formal organizations [34], government-regulated cellular networks [42], and call tree notification systems [36]. Furthermore, we stress that trust is not static, it changes over time (ableit slowly). In this sense, we approach trust as deliberate infrastructure. Since centralized network structure is inherently vulnerable to attack, the ability to deliberately structure trust relationships is a necessary component of any attack-tolerance scheme.

We first describe how fault tolerance techniques can be adapted and evaluated in a network setting with partial trust. Generally, faults in network paths can be correlated, preventing the appliation of standard fault tolerance techniques [3, 41], which assume independent faults. By using *independent paths*, which have no untrusted nodes in common, we model communication across a complex network by a number of redundant and independent channels. Fault-tolerance schemes can then be applied when redundant messages are sent in parallel, a technique known as *concurrent multipath routing* [46, 37, 21]. The receiver can then use the redundant messages to detect and/or correct errors. We formally evaluate the effects of network structure on attack-resistance and show that the probability of undetected errors decreases exponentially with the number of independent paths between source and destination, even when no individual path is entirely trusted.

We also propose a novel concurrent multipath routing algorithm for the butterfly topology. The algorithm identifies independent paths, which when combined with the fault-tolerant concurrent multipath routing scheme above, achieves a high level of adversarial fault tolerance on the butterfly topology.

Our main contributions are:

- We show that *independent paths* with *partial trust transitivity* enable standard fault tolerance techniques to be applied to adversarial faults in complex communication networks. We find that the probability of detecting adversarial faults approaches 1 exponentially as the number of independent paths between sender and receiver increases;
- We present a scalable, efficient, and attack-tolerant concurrent multipath routing algorithm on the butterfly network topology.

This paper is organized as follows. Section 2. reviews background and related work. Section 3. discusses the desired properties of attack-tolerant network infrastructure. Section 4. describes adversarial fault tolerance on structured networks. Section 5. describes a routing algorithm for multipath fault tolerance on the butterfly network topology. Section 6. discusses the our results. And Section 7. concludes.

2. BACKGROUND AND RELATED WORK

There has been considerable work on trust in network security. Both centralized and decentralized approaches are commonly used to create trust infrastructures. Centralized approaches such as *public key infrastructure* (PKI) suffer from a number of vulnerabilities [14], which stem largely from the single points of failure inherent to centralization. The well-known and widely-used *web of trust* approach [45, 15] is a decentralized alternative. In a web of trust, individuals choose who they trust initially. Trust is then extended to new individuals if they are vouched for by a currently-trusted individual, making it possible to extend their web of trust to a large number of nodes. However, the web of trust approach assumes that trust can be extended transitively, which is unrealistic [10]. Previous work on evaluating network structure has focused on authentication protocols, showing that independent paths can reduce an adversary's ability to impersonate a target [28]. Other work has shown that identifying independent paths in arbitrary networks is NP-hard and provided approximation algorithms [38]. Our work complements these by explicitly quantifying the effects of partial trust transitivity, and providing a routing algorithm to efficiently construct independent paths on a structured network.

Many distributed consensus protocols (such as those used by cryptocurrencies) are designed to provide tolerance against arbitrary or adversarial faults. Byzantine agreement protocols [27, 8] provide tolerance against arbitrary faults (including attacks) under some circumstances, but are limited to small networks due to poor scalability. Proof-of-work [13, 35] and proof-of-stake [22] provide better scalability, but are wasteful of computational and energy resources. Federated Byzantine Agreement (FBA) [33] is scalable, allows for flexible trust, and is highly fault-tolerant on networks meeting a set of requirements. However, FBA does not provide a method for evaluating the fault tolerance properties of different network structures or calculating how the failure probability for a particular network.

Most existing adversarial fault tolerance schemes focus on end-to-end, protocol-based, cryptographic solutions [15], without addressing single points of failure in-between (i.e., *man-in-the-middle attacks*). There are however, a few notable fault tolerance schemes that focus on network structure. Fiat and Saia described a scheme that combines the butterfly topology with expander graphs to create a highly censorship-resistant content-addressable network [16], although this scheme exhibits poor scaling due to a very high level of data replication. Perhaps the most mature structural solution is the Freenet collaboration [11]. Freenet uses secret sharing [39, 7] and small-world routing [44, 23] to create a

content-addressable network with a high level of both confidentiality and censorship resistance. Freenet guarantees that data is stored redundantly, but still allows for centralized network structure and thus single points of failure as data travels from its origin to the redundant storage locations.

Multipath routing protocols send redundant information over multiple paths when routing a message through a network, in contrast to traditional *unipath* routing, which uses a single path. Multipath routing can have many benefits, including reduced congestion, increased throughput, and more reliability [37]. Many of these routing protocols offer increased confidentiality [46]. Some approaches utilize redundant paths as backups for increased fault tolerance [2], and some specifically protect against adversarial faults [24, 20, 30]. Most work on multipath routing has been motivated by applications related to wireless sensor networks (WSNs), and have thus focused on ad hoc, unstructured networks, often having a central base station. The method of Liu et al. [29] routes multiple messages first to random peers and then to a central base station, with the network edges constrained by sensors' physical location. The butterfly routing algorithm we present takes a conceptually similar approach in a different network setting.

Our proposed routing algorithm makes use of a *structured network*, in which link structure is predetermined. Such networks can be designed to have favorable structural and routing properties, at the expense of complicating the addition or removal of nodes. Structured networks have been a popular tool in parallel processing architectures [26]. More recently, peer-to-peer systems based on distributed hash tables have used structured "overlay" networks to map table keys to local TCP/IP routes [31, 25].

3. ATTACK-TOLERANT INFRASTRUCTURE

In this section, we describe the functional properties required of an attack-tolerant network, which serve as guiding principles in later sections. We pay special attention to a property we call *stabilizing asymmetry*.

3.1 Functional Properties

Scalability: For large scale networks it is important that the infrastructure allows for the network to grow while remaining functional. In practice, people, devices, and connections, have limited capabilities and these limitations need to be considered as part of the design of the infrastructure.

Decentralization: Systems having single points of failure are less tolerant against faults at those points. The existence of such points not only increases the likelihood that an attack will succeed, but also incentivizes attack by presenting effective targets. In order to minimize single points of failure, attack tolerant infrastructures must use decentralized protocols and decentralized network structures.

Stabilizing asymmetry: In the context of international conflict, *asymmetric conflicts* are a special case that makes it possible for the less powerful party to have an advantage over the more powerful party [32]. In asymmetric conflicts, the same level of resource expenditure yields different results for different parties; the attacker's resources are either more or less effective than the defender's. We call the latter case stabilizing asymmetry, because it reduces an attacker's power relative to their target. With this in mind, an attack-resistant infrastructure will benefit from a high level of stabilizing asymmetry.

3.2 Structural Properties

We are specifically concerned with network structure-based approaches to fault tolerance. In networks, specific structural properties are required to achieve the functional properties described above.

Sparsity and low diameter: To achieve scalability, networks must be *sparse* and have a *low diameter*. In practical settings, humans and devices have an upper limit on the number of connections they can maintain (e.g., Dunbar's number [12]). In sparse networks, the number of links grows slowly as the network grows in size, allowing the network to scale without exceeding the nodes' capacity for links. Similarly, low-diameter guarantees that as a network grows, a short path will still exist between any pair of nodes. While low diameter guarantees a path exists, paths are only useful if an efficient *routing* algorithm exists to find them.

Uniform centrality: There are many ways to measure node centrality in networks [17]. The more uniform these measures are across nodes, the more decentralized a network is. Centrality is minimized in *vertex transitive* networks, for which an edge-preserving map always exists from any node to any other node. In other words, all nodes occupy structurally indistinguishable positions in the network.

Redundancy: In a network, redundancy refers to the existence of multiple non-overlapping paths between nodes or components. Redundancy can help reduce single points of failure and decrease centralization. One measure of redundancy is given by the ratio of edges to nodes [6]. A single point of failure occurs when a node holds a uniquely central position. When alternative paths are added to bypass central nodes, the network becomes more redundant.

4. TRUST AND FAULT TOLERANCE

Within the field of *fault tolerance*, many techniques have been developed for building reliable systems out of unreliable components [3, 41]. We will make use of standard fault tolerance terminology, summarized here. A *fault* occurs when one component of a system behaves incorrectly (e.g., a routing node blocking or altering a message). The result of that fault (e.g., a recipient receiving conflicting messages) is an *error* state. If the error is undetected or corrected to the wrong value, the system is said to have experienced a *failure* (e.g., an altered message is accepted as authentic). We are concerned in particular with *adversarial faults*, which (as opposed to random faults) are chosen strategically to maximize the likelihood of a failure.

4.1 Partial Trust Model

An important challenge in large-scale, secure communication is the ability to communicate reliably and securely with remote parties for whom no direct, trusted link exists.

The commonly-used web of trust approach [45, 15]. extends trust infinitely transitively, to friends of friends, and friends of friends of friends, and so on. However, the assumption of infinitely transitive trust is unrealistic [10], and does not allow for the analysis of network structure.

We can allow for more realistic transitivity assumptions and incorporate network structure into a trust model with one key insight: even if no single path between a sender and receiver is fully trusted, multiple copies of a message can be sent along different paths and compared by the receiver to detect and correct errors. However, in the presence of an attacker, there is only benefit in sending an additional message along a path if that path does not share untrusted single points of failure with any of the existing message paths (otherwise, the adversary can compromise both messages by causing a single fault). Following [38], we call paths that do not share any untrusted single points of failure *independent*. The maximum number of independent paths represents the effective redundancy that can be utilized by any redundancy-based fault tolerance scheme. We now propose a method for quantifying the effective redundancy of a network as a function of its structure and level of trust transitivity. For the purpose of analyzing adversarial fault tolerance, this method allows a communication network to be modeled as a set of virtual links that directly connect each node pair, with each virtual link providing some level of redundancy.

We now specify the *partial trust model*. We assume an undirected graph $G = (V, E)$, although the model can easily be extended to directed multigraphs. Vertices representing commiunicating agents, and with edges representing mutually trusted communication links. Let $v \in V$ be an arbitrary sender (Alice) and $w \in V$ be an arbitrary receiver (Bob). We assume the presence of an adversary (Mal) who knows the full structure of the network, and who can compromise a fixed number of agents, gaining complete control of their behavior, as long as those agents are not trusted by either Alice or Bob. We define a *trust radius* h and that nodes u and u' trust each other if their distance is less than h. For a given node u, we call the set of trusted nodes its *trusted neighborhood* $T_h(u)$, and all nodes at exactly distance h the *trust boundary* $B_h(u)$:

$$T_h(u) = \{u \mid d(u, u') < h\} \quad (1)$$

$$B_h(u) = \{u \mid d(u, u') = h\}. \quad (2)$$

The trust boundary B_h plays an important role because these nodes are not trusted by u, but if compromised can entirely isolate u from the rest of the network. These trust assumptions imply that when Alice sends a message to Bob, Mal can only cause faults in the set of nodes outside both of their trusted neighborhoods: $V \setminus (T_h(v) \cup T_h(w))$. We refer to this set of nodes as the *untrusted region*.

Having defined the assumptions of the partial trust model, we now quantify the effective redundancy between Alice and Bob when trust radius h is assumed. This quantity, $\delta_{v,w,h}$ is exactly the max-flow/min-cut of the graph after Alice's and Bob's trusted neighborhoods have been collapsed into single source/sink vertices. Each trust boundary forms a cut of the network and places an upper bound on the min-cut:

$$\delta_{v,w,h} \leq \min\left(\mid B_h(v) \mid, \mid B_h(w) \mid\right). \quad (3)$$

Equality holds when there are no bottlenecks within the untrusted region, an indication that the network is decentralized. The redundancy of the entire graph can be characterized by the minimum over all vertex pairs:

$$\delta_h(V) \equiv \min_{v,w \in V} \delta_{v,w,h}. \quad (4)$$

Thus, for any pair of nodes in the network, at least δ_h independent, redundant paths can be constructed between them. While δ_h is a purely structural property of the graph and may have additional applications, we are interested in this quantity because it places an upper bound on the effectiveness of any redundancy-based fault tolerance scheme. As a function of h, δ_h describes a network's ability to create redundancy from trust. Combined with a *routing algorithm* that tells nodes how to forward messages to guarantee they take independent paths, this redundancy can be used to achieve fault tolerance. For attack-tolerant networks, it is thus desirable for δ_h to increase quickly as a function of h. Furthermore, by applying the partial trust model, an adversarial fault tolerance scheme can treat each pair of nodes as being directly connected by a virtual link with least δ_h redundant channels. We will now describe such a scheme and evaluate its fault tolerance properties.

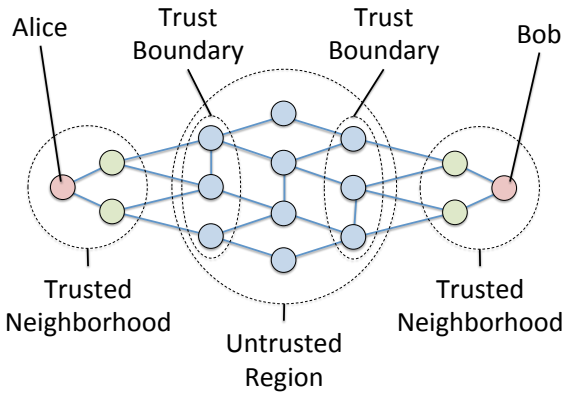

Figure 1: Illustration of a trusted communication network and the network properties used by the *partial trust model*. Edges represent mutually trusted communication links. The sender (Alice, v) and receiver (Bob, w) trust all nodes less than the *trust radius* h hops away. These nodes form their *trusted neighborhoods* $T_h(v)$ and $T_h(w)$. We assume that all faults occur in the remaining nodes: the *untrusted region*. The untrusted nodes in contact with the trusted neighborhoods for the *trust boundaries* $B_h(v)$ and $B_h(w)$, which (in the absence of central bottlenecks) determine the *effective redundancy* δ_h provided by the network. Alice and Bob can be modelled as connected by a direct link with at least δ_h redundant channels.

4.2 Multipath Fault Tolerance

Once we have determined a network's effective redundancy, we can apply redundancy-based fault tolerance techniques, by sending multiple copies of a message (*concurrent multipath routing*). We model our sender (Alice) and receiver (Bob) as communicating over a direct link with δ_h redundant channels. The partial trust model allows us to make this simplifying assumption for analyzing a fault tolerance scheme, but implementing such a scheme will require a method for constructing specific network paths. We will

return to the question of constructing paths in the next section. For now, we concern ourselves with the question: given that the network provides δ_h redundant channels between Alice and Bob, what is the probability that an adversary (Mal) causes an undetectable error?

Let us first assume that Alice sends a message copy over each available channel. When Bob receives the messages, there are several possibilities. If some of the messages are missing (Alice can include the number of messages as part of the message) or if some of the messages disagree, Bob knows that some of the messages were either blocked or altered, and he has successfully detected an error. Bob has not accepted any compromised information, and can request retransmission, so no failure has occurred. If Bob receives all of the messages, and they all agree, he accepts the message. If the messages agree, there are two possible cases. The first case is that Mal has not compromised any of the messages, and Bob has correctly accepted them, so no failure has occurred. The second case is that Mal has compromised *all* of the messages, so Bob has accepted an erroneous message and a failure has occurred. In the present scenario, whether this occurs depends only on whether Mal has the resources to compromise all of the channels. In a more realistic, and more interesting, scenario, both Alice and Mal have limited resources and are not able to use or compromise all available channels.

In a more sophisticated multipath fault tolerance scheme, Alice randomly chooses $k \leq \delta_h$ channels and sends a copy of her message on each. We assume that Mal is capable of compromising $l \leq \delta_h$ channels. Having full knowledge of the network, Mal's best strategy is to identify a minimum node cut in the network and compromise nodes from that cut. With this strategy, each compromised node reduces effective redundancy by one, effectively compromising one of the virtual channels between Alice and Bob. Since Alice chooses channels randomly, all channels are equally likely to contain a message, so Mal can do no better than also choosing randomly. If $k > l$, at least one message will get through uncompromised and all errors are detectable. Otherwise, the probability of Mal producing an undetectable error is the probability that all of Alice's chosen channels are compromised:

$$p_f = \frac{l!(\delta_h - k)!}{\delta_h!(l-k)!}. \quad (5)$$

Letting $k = \alpha\delta_h$ and $l = \beta\delta_h$, then applying Stirling's approximation gives:

$$p_f \approx \frac{\sqrt{\beta(1-\alpha)}}{\sqrt{\beta-\alpha}} \left[\left(\frac{\beta-\alpha}{1-\alpha} \right)^{\alpha} \left(\frac{\beta}{\beta-\alpha} \right)^{\beta} (1-\alpha) \right]^{\delta_h} \quad (6)$$

Figure 2 shows the value of p_f as a function of k and l. Equation (6) shows that while p_f depends on the fractions of redundant paths actually utilized α and compromised β, it decreases exponentially with the effective redundancy δ_h. This result is significant because δ_h depends only on the network structure and the strength of trust transitivity, not on the amount of resources available to the sender or the adversary. In other words, this scheme exhibits a *stabilizing asymmetry*, senders can tolerate attacks from significantly more powerful adversaries, as long as the network structure provides large δ_h. By putting attackers at a disadvantage, attacks are not only tolerated but also disincentivized, further decreasing the likelihood of a successful attack.

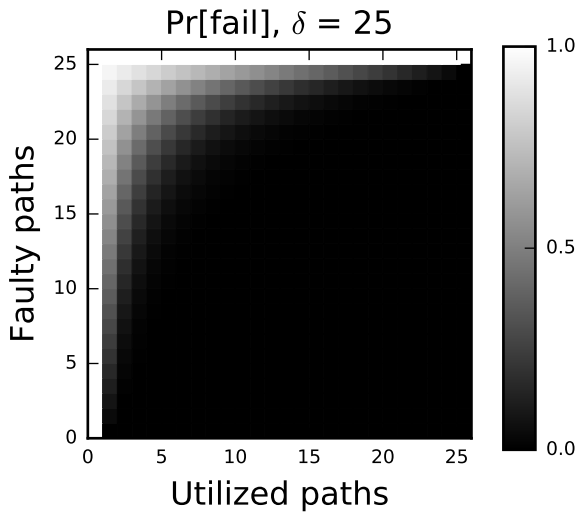

Figure 2: The probability of an undetectable error as a function of the number of message copies and the number of adversarial faults.

In order to derive the above results, we have assumed that Alice and all intermediary agents are able to identify specific, independent network paths that achieve the effective redundancy δ_h. We now proceed to describe a routing algorithm for doing so in the special case of the butterfly network topology.

5. MULTIPATH BUTTERFLY ROUTING

In previous sections, we showed that reliable communication across a network can be achieved even when any single message path might be compromised by an adversary, provided the network has sufficient redundancy, and provided the sender and intermediaries know how to route message copies along independent paths. In this section, we address the both requirements by proposing a novel routing algorithm for constructing independent paths on the butterfly network topology. This architecture and routing algorithm achieve an effective redundancy that increases exponentially with the trust radius, allowing a very high level of adversarial fault tolerance.

The structure of the butterfly network is highly constrained, making it most suitable for applications where network structure can be designed or dictated. Examples of such networks include: overlay networks [31, 25], formal organizations [34], government-regulated cellular networks [42], and call tree notification systems [36]. For critical security applications in particular, the novel level of adversarial fault tolerance offered by this scheme is highly desirable, even if it requires new approaches to designing communication and trust networks. Lastly, we note that the partial trust model and multipath fault tolerance schemes of the previous section do not rely on any particular network topology or routing algorithm, and are not necessarily limited to such applications.

5.1 Butterfly Network Topology

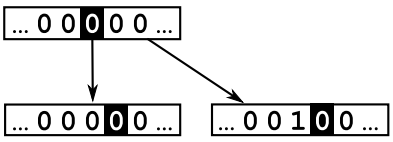

Figure 3: Schematic illustration of the two types of edges in a directed butterfly network. The node (l, z) is shown as the bit string z with a square around the lth bit. "Down" edges increment l, leaving z unchanged, while "down-right" edges increment l and invert the lth bit of z.

We choose the butterfly topology [26] because of several desirable properties (described below) and because its structure allows for relatively straightforward design and analysis of routing algorithms. While several variations on the butterfly network exist, we utilize the wrap-around butterfly. We denote the m-dimensional, directed wrap-around butterfly as a graph wBF(m):

$$\begin{aligned} \text{wBF}(m) &= (V, E_{\downarrow} \cup E_{\rightarrow}) && (7)\\ V &= \mathbb{Z}_m \times \mathbb{Z}_2^m && (8)\\ E_{\downarrow} &= \{((l,z),(l+1 \pmod m, z)\} && (9)\\ E_{\rightarrow} &= \{(l,z),(l+1 \pmod m, z \oplus 1_l\}, && (10) \end{aligned}$$

where $\mathbb{Z}_m$ is the set of integers modulo m, $\oplus$ represents componentwise addition modulo 2, and 1_l is a vector with a 1 in index l and 0 elsewhere. Each node is associated with a level l and an m-bit string z known as *the place-within-level*. There are two types of edges (shown in Figure 3). Down edges ($E_{\downarrow}$) connect nodes sharing the same z value in a cycle of increasing level l. Down-right edges ($E_{\rightarrow}$) also link to a node of level $l+1$, but one having the place-within-level equal to z with the lth bit inverted.

The wrap-around butterfly network is known to have several of the properties we desire for scalable, decentralized communication networks:

Vertex-transitivity: Because the wraparound butterfly is vertex transitive, it is maximally decentralized;

Small-diameter: For any two nodes, the length of the shortest path between them is $O(\log N)$, where N is the number of nodes in the network;

Sparsity: With a constant degree of 4, the wraparound butterfly is extremely sparse, and can scale indefinitely without node degree becoming a limitation;

Redundancy: Multiple paths exist between any two nodes. Specifically, we will prove below that the number of independent paths between two nodes increases exponentially with the trust radius h.

The structure of the butterfly network lends itself to a well-known (unipath) routing algorithm, which we later extend to the multipath case. The unipath algorithm first follows a down or down-right edge at every step, increasing the level l by 1 and cycling through the indices of the place-within-level. If the current node's place-within-level matches the destination node's at index l, a down edge is chosen and the place-within-level does not change. Otherwise, a down-right edge is chosen and the lth component of the place-within-level is flipped, after which it matches the destination. After m iterations of this, all levels have been visited and the place-within-level matches that of the destination. Simply following down (or up) edges will then increment (decrement) the level until the destination node is reached.

5.2 Multipath Routing Algorithm

We now present a routing algorithm to construct multiple independent paths between two nodes in a butterfly network. Independence requires that routes between two nodes v and w have no nodes in common, unless those nodes are less than distance h from either v or w. Informally, the algorithm achieves this by using the first h hops to reach as many distinct nodes as possible, then routing from each of those to a distinct intermediary node, and finally routing from the intermediary to the destination. By ensuring that the place-within-level differs for all paths (outside of the trusted neighborhoods of v and w) the algorithm guaranatees independence. As with the unipath routing algorithm, each of the multiple paths proceed using down and down-right edges, cycling through levels one at a time. However, we cycle through the levels twice, once to route from v to a particular path's intermediary node, and again to route from the intermediary to w. In the first cycle, path independence is guaranteed by ensuring that all paths differ in the first h bits of the place-within-level. Similarly, in the second cycle, independence is guaranteed by ensuring that all paths differ in the h bits of the place-within-level preceding the destination index (see Figure 4).

We now begin the formal specification of our multipath routing scheme for the wraparound butterfly network. Utilizing vertex transitivity, we label the source node as $(l^{(0)}, z^{(0)}) = (0,0)$ and denote the destination node as $w = (l_w, z_w)$, without loss of generality.

Let s be an h-bit binary string with s_i denoting the bit at index i. There are 2^h such strings. Let $v_s^{(t)} = (l^{(t)}, z^{(t)})$ be the node at position t in the path parameterized by s. For convenience, we will omit the subscript s when it is obvious from context. We define three distinct partitions of m-bit binary strings. Let $Q_{v^{(0)}}$ ($\overline{Q_{v^{(0)}}}$) be the set of m-bit strings in which the bits at all indices $h \le i < l_w - h$ match (do not all match) those of $z^{(0)}$. Note that $Q_{v^{(0)}}$ is trivially all m-bit strings if $l_w < 2h$. Let R_s ($\overline{R_s}$) be the set of m-bit strings with the lowest h bits all matching (not all matching) the bits of s. Let S_s ($\overline{S_s}$) be the set of m-bit strings with the h bits preceding index l_w all matching (not all matching) the bits of $\tilde{s}$, where $\tilde{s}$ is a cyclic permutation of s:

$$\tilde{s}_i = s_{(i+l_w) \bmod h}. \quad (11)$$

We will make use of the fact that:

$$s \neq s' \implies S_s \cap S_{s'} = R_s \cap R_{s'} = \emptyset. \quad (12)$$

Routes are constructed in 7 stages. The network topology dictates that $l^{(t+1)} = l^{(t)} + 1 \pmod m$, so we let $l = t \pmod m$. and that $z^{(t+1)}$ is equal to $z^{(t)}$ with or without the bit in index $l^{(t)}$ inverted, depending on whether the down or down-right edge was taken at step t.

Stage 1: **($0 \le t < h$)** Down or down-right edges are chosen such that the tth bit of $z^{(t+1)}$ is equal to the tth bit of s. Throughout Stage 1, all nodes are within the sender's trusted neighborhood. Throughout Stage 1,

	A	B	C	D
	h	$l_w - 2h$	h	$m - l_w$
start	$0\ldots$	$\ldots 0\ldots$	$\ldots 0\ldots$	$\ldots 0$
1.	s	$\ldots 0\ldots$	$\ldots 0\ldots$	$\ldots 0$
2.	s	$\ldots 1\ldots$	$\ldots 0\ldots$	$\ldots 0$
3.	s	$\ldots 1\ldots$	$\tilde{s}$	$\ldots 0$
4.	s	$\ldots 1\ldots$	$\tilde{s}$	$z_{w,D}$
5.	$z_{w,A}$	$\ldots 1\ldots$	$\tilde{s}$	$z_{w,D}$
6.	$z_{w,A}$	$z_{w,B}$	$\tilde{s}$	$z_{w,D}$
7.	$z_{w,A}$	$z_{w,B}$	$z_{w,C}$	$z_{w,D}$

Figure 4: Progression of place-within-level z as the multipath routing algorithm cycles through the levels of the butterfly network.

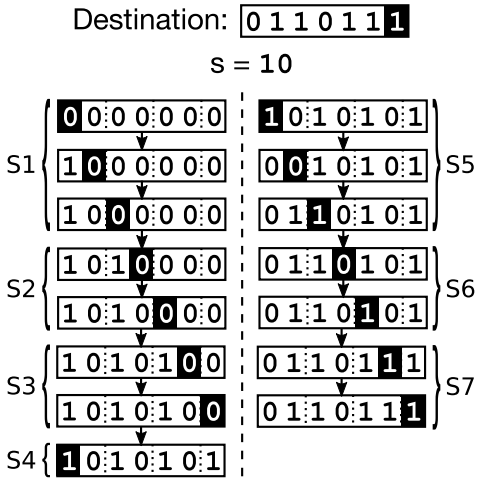

Figure 5: An example of one path as constructed by the proposed multipath routing algorithm. The path is shown for $s = 10_2$ and $w = (6, 0110111_2)$.

Table 1: Butterfly Multipath Routing Variables

NAME	VARIABLE
butterfly dimension	$m \in \mathbb{Z}_+$
node level	$l \in \mathbb{Z} : 0 \le l < m$
node place within level	$z \in \mathbb{Z}_2^m$
trust radius	$h \in \mathbb{Z} : 1 \le h \le \lfloor m/2 \rfloor$
path index	$s \in \mathbb{Z}_2^h$

$z^{(t)} \in Q_{v^{(0)}}$. At the end of Stage 1, $z^{(h)} \in S_s$, and $z^{(t)}$ will remain so until the level cycles to 0 at $t = m$.

Stage 2: $(h \le t < l_w - h)$ Edges are chosen to make the tth bit of $z^{(t+1)}$ the inverse of the tth bit of $z^{(0)}$. Note that this stage does not occur when $l_w < 2h$. If this stage occurs, then $z^{(t)} \in \overline{Q_{v^{(0)}}}$ until these levels are reached again in stage 6.

Stage 3: $(l_w - h \le t < l_w)$ The bits of $z^{(t)}$ are chosen to match $\tilde{s}$, such that after the stage is complete, $z^{(t)} \in R_s$.

Stage 4: $(l_w \le t < m)$ Paths are chosen such that the tth bit of $z^{(t+1)}$ matches that of the destination node z_w. This stage will not occur if $l_w > m - h$.

Stage 5: $(m \le t < m + h)$ There are two cases. If $2h < l_w < m-h$, then there is no overlap between the indices defining R_s and S_s. In this case, the first h bits of $z^{(t)}$ are set to match z_w. Otherwise there is some overlap between the indices defining R_s and S_s. In this case, the each of the first h bits of $z^{(t)}$ is either kept the same if $l_w - h \le l < l_w$, or set to the corresponding bit of z_w otherwise. In this stage and after, $z^{(t)}$ is no longer guaranteed to be in R_s. However, $z^{(t)}$ remains in S_s during and after this stage.

Stage 6: $(m + h \le t < m + l_w - h)$ In this stage, edges are chosen to set the bits of $z^{(t)}$ to their corresponding value in z_w. $z^{(t)} \in \overline{Q_{v^{(0)}}}$ throughout this stage, but not afterwards.

Stage 7: $(m + l_w - h \le t < m + l_w)$ The h bits of $z^{(t)}$ preceding index l_w are set to match z_w. All nodes in this stage are within h hops of w and thus in its trusted neighborhood. After this stage, $v^{(m+l_w)} = w$ and routing is complete.

THEOREM 1. *Given an m-bit wraparound butterfly network ($m \ge 0$), and a trust radius $h \ge 1$, for all node pairs (v, w), if there exists a node u that belongs to two distinct routes $v_s^{(\cdot)}$ and $v_{s'}^{(\cdot)}$ ($s \ne s'$), it implies that $u \in T_h(v) \cup T_h(w)$.*

PROOF. Nodes from two paths can only coincide if their levels are the same. Nodes which share a level must either be in the same stage, or 4 stages apart. Let (a,a') denote a pair of sub-paths corresponding to stage a of one path and stage a' of another. Excluding paths that intersect in their trusted neighborhoods, (1,1) and (7,7), we have reduced the list of possible intersections to the following cases: (2,2), (3,3), (4,4), (5,5), (6,6), (1,5), (2,6), and (3,7). Nodes in stages 2–4 belong to R_s so cannot overlap with any stage 2–4 nodes from another path, eliminating (2,2), (3,3), and (4,4). Similarly, nodes in stages 4–6 belong to a unique S_s, eliminating (5,5) and (6,6). Nodes in stage 1 belong to $Q_{v^{(0)}}$ while those in stage 5 belong in its complement, eliminating (1,5). Similarly, for all l in stage 2, $z^{(l)}$ is equal to $z^{(0)}$, while in stage 6, $z^{(l)}$ is the inverse, eliminating (2,6) This leaves only (3,7), a collision which can occur only for only one path (with s matching the first h bits of z_w), and which enters the trusted neighborhood in stage 3. For this single path, we can proceed directly from stage 2 to stage 7, eliminating the last possible collision. □

Thus, assuming the partial trust model with trust transitive for h hops, we can construct 2^h paths on a wraparound butterfly topology which do not intersect outside the trusted neighborhoods of the source and destination. Note that the node sequence $v_s^{(t)}$ can be calculated entirely from the source v, destination w, and path parameter s, meaning that with this information nodes are able to determine which neighbor to route a given message copy to. Furthermore, the existence of 2^h paths places a lower bound on the effective redundancy δ_h, showing that the decentralized, redundant,

structured networks such as the butterfly can have a very low probability of failure when faced with adversarial faults, even from a very powerful attacker.

6. DISCUSSION

While decentralized protocols have received much attention for their potential fault tolerance applications, a better understanding of how network structure contributes to fault tolerance is necessary. We have proposed a network-based scheme for adversarial fault tolerance on the butterfly topology, utilizing a novel *concurrent multipath routing* algorithm. We have also demonstrated how *partial trust transitivity*, in addition to being more realistic than infinite transitivity, enables the analysis of the relationship between trust, network structure, and fault tolerance.

Fault-tolerant network infrastructures have many direct applications. Areas such as cryptocurrency [33, 35, 22], secure multiparty computation [43, 9, 18], and wireless sensor networks [21] have immediate need for scalable, fault-tolerant infrastructures. Many Internet services (e.g., email, social networks, cloud storage) are still highly centralized and vulnerable to technical and non-technical (i.e. coercive) attacks. Fault tolerance using *both* decentralized protocols and decentralized network structures is one promising approach to securing these services.

We have focused specifically on adversarial faults that block or change messages (e.g., censorship). We have not specifically addressed *confidentiality*: how to keep those messages private. Existing cryptographic techniques for confidentiality are relatively mature compared to those for tolerating censorship. In fact, an attacker might use targeted attacks to achieve censorship as a fallback when surveillance isn't feasible. However, the techniques presented in this paper are entirely compatible with, and in some cases could enhance, existing confidentiality techniques. For example, the well-known *man-in-the-middle* attack exploits a privileged network position to attack otherwise secure cryptography, suggesting that structural approaches can complement cryptographic ones.

While our present proposal is specific to the butterfly topology, the multipath fault tolerance scheme could be applied to any network that has both sufficient redundancy and a routing algorithm to discover independent paths. For general networks, finding all such paths is NP-hard, but efficient, suboptimal algorithms exist [38]. However, our approach is best suited for structured networks. While the ability to control network topology is only achievable in some applications, those applications are nonetheless significant (e.g. overlay networks [31, 25], formal organizations [34], government-regulated cellular networks [42], and call tree notification systems [36]). Furthermore, when high levels of adversarial fault tolerance are required, designers of new sociotechnical systems might choose incorporate such networks. For example, a coalition of groups supporting free expression could use our results to construct a censorship-resistant communication network by cultivating the necessary trust relationships.

Our work suggests several directions for future work towards developing practical attack-tolerant communication infrastructure. The development of new multipath routing algorithms on other structured networks could achieve higher levels of redundancy. It is also desirable to identify dynamics that give rise to structured networks, and to evaluate whether our results can be generalized to unstructured or approximately structured networks. Finally, these results could be implemented to address specific applications, e.g., secure messaging, domain name resolution, or anonymous web browsing.

7. CONCLUSION

We have presented a novel multipath routing algorithm for the butterfly topology, as well as a scheme for using this algorithm to achieve a high level of attack-tolerance in a communication network, even when no single path is entirely trusted. Under this scheme, the probability of an adversary causing an undetectable error decreases exponentially with the network's effective redundancy. The effective redundancy, in the case of the butterfly topology, grows exponentially with the trust radius. Furthermore, a small increase in the number of messages sent can compensate for a large increase in the number of messages compromised by an adversary. We have also demonstrated how the assumption of partial trust transitivity can enable a formal analysis of the effects of network structure on attack-tolerance. These results are directly applicable to systems in which the link structure can be imposed by the designer, and more generally, provide a formalism that can be used more to evaluate the role of network structure, trust transitivity, and effective redundancy on adversarial fault tolerance.

8. ACKNOWLEDGMENTS

9. REFERENCES

[1] R. Albert, H. Jeong, and A.-L. Barabási. Error and attack tolerance of complex networks. *Nature*, 406(6794):378–382, 2000.

[2] N. A. Alrajeh, M. S. Alabed, and M. S. Elwahiby. Secure ant-based routing protocol for wireless sensor network. *Int J Distrib Sens N*, 2013, 2013.

[3] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE T Depend Secure*, 1(1):11–33, 2004.

[4] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[5] A.-L. Barabási and others. Scale-free networks: a decade and beyond. *Science*, 325(5939):412, 2009.

[6] P. Baran and others. On distributed communications. *Volumes I-XI, RAND Corporation Research Documents, August*, pages 637–648, 1964.

[7] G. R. Blakley. Safeguarding cryptographic keys. *P Natl Comp Conf*, 48:313–317, 1979.

[8] M. Castro, B. Liskov, and others. Practical Byzantine fault tolerance. In *OSDI*, volume 99, pages 173–186, 1999.

[9] D. Chaum, C. Crépeau, and I. Damgard. Multiparty unconditionally secure protocols. In *STOC*, pages 11–19. ACM, 1988.

[10] B. Christianson and W. S. Harbison. Why isn't trust transitive? In *Security protocols*, pages 171–176. Springer, 1997.

[11] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. Freenet: A distributed anonymous information storage

and retrieval system. In *Designing Privacy Enhancing Technologies*, pages 46–66. Springer, 2001.

[12] R. I. Dunbar. Neocortex size as a constraint on group size in primates. *Journal of Human Evolution*, 22(6):469–493, 1992.

[13] C. Dwork and M. Naor. Pricing via processing or combatting junk mail. In *Advances in Cryptology*, pages 139–147. Springer, 1993.

[14] C. Ellison and B. Schneier. Ten risks of PKI: What you're not being told about public key infrastructure. *Comput Secur J*, 16(1):1–7, 2000.

[15] N. Ferguson and B. Schneier. *Practical cryptography*. Wiley, New York, 2003.

[16] A. Fiat and J. Saia. Censorship resistant peer-to-peer content addressable networks. In *SIAM SODA*, pages 94–103. ACM, 2002.

[17] L. C. Freeman. Centrality in social networks conceptual clarification. *Social networks*, 1(3):215–239, 1978.

[18] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. In *STOC*, pages 218–229. ACM, 1987.

[19] P. Hunter. Pakistan YouTube block exposes fundamental internet security weakness: Concern that pakistani action affected youtube access elsewhere in world. *Computer Fraud & Security*, 2008(4):10–11, 2008.

[20] I. Khalil, S. Bagchi, C. N. Rotaru, and N. B. Shroff. UnMask: Utilizing neighbor monitoring for attack mitigation in multihop wireless sensor networks. *Ad Hoc Networks*, 8(2):148–164, 2010.

[21] S. R. Khiani, C. Dethe, and V. Thakare. Comparative Analysis of Multipath Routing Techniques and Design of Secure Energy Aware Routing Algorithm for Wireless Sensor Network. *IJACR*, 3(3):374, 2013.

[22] S. King and S. Nadal. *Ppcoin: Peer-to-peer crypto-currency with proof-of-stake*. August, 2012.

[23] J. Kleinberg. The small-world phenomenon: An algorithmic perspective. In *STOC*, pages 163–170. ACM, 2000.

[24] E. Kohno, T. Okazaki, M. Takeuchi, T. Ohta, Y. Kakuda, and M. Aida. Improvement of assurance including security for wireless sensor networks using dispersed data transmission. *J Comp Sys Sci*, 78(6):1703–1715, 2012.

[25] D. Korzun and A. Gurtov. *Structured peer-to-peer systems: fundamentals of hierarchical organization, routing, scaling, and security*. Springer, New York, NY, 2013.

[26] A. D. Kshemkalyani and M. Singhal. *Distributed computing: principles, algorithms, and systems*. Cambridge University Press, 2008.

[27] L. Lamport, R. Shostak, and M. Pease. The Byzantine generals problem. *TOPLAS*, 4(3):382–401, 1982.

[28] R. Levien. Attack-resistant trust metrics. In *Computing with Social Trust*, pages 121–132. Springer, 2009.

[29] A. Liu, Z. Zheng, C. Zhang, Z. Chen, and X. Shen. Secure and energy-efficient disjoint multipath routing for WSNs. *IEEE T Veh Technol*, 61(7):3255–3265, 2012.

[30] W. Lou and Y. Kwon. H-SPREAD: a hybrid multipath scheme for secure and reliable data collection in wireless sensor networks. *IEEE T Veh Technol*, 55(4):1320–1330, 2006.

[31] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim. A survey and comparison of peer-to-peer overlay network schemes. *IEEE Commun Surv Tut*, 7(2):72–93, 2005.

[32] A. Mack. Why big nations lose small wars: The politics of asymmetric conflict. *World Politics*, 27(02):175–200, 1975.

[33] D. Mazières. *Stellar Consensus Protocol: A Federated Model for Internet-level Consensus*. 2015.

[34] L. B. Mohr. *Explaining organizational behavior*. Jossey-Bass, 1982.

[35] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *bitcoin.org*, page 28, 2008.

[36] J. V. Nickerson, B. Tversky, J. E. Corter, L. Yu, and D. Mason. Thinking with networks. In *CogSci*, volume 36, 2010.

[37] J. Qadir, A. Ali, K.-L. A. Yau, A. Sathiaseelan, and J. Crowcroft. Exploiting the power of multiplicity: a holistic survey of network-layer multipath. *IEEE Comm Surv Tut*, 17(4):2176–2213, 2015.

[38] M. K. Reiter and S. G. Stubblebine. Resilient authentication using path independence. *IEEE T Comput*, 47(12):1351–1362, 1998.

[39] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[40] J. P. Sterbenz, D. Hutchison, E. K. Çetinkaya, A. Jabbar, J. P. Rohrer, M. Schöller, and P. Smith. Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines. *Computer Networks*, 54(8):1245–1265, 2010.

[41] J. Von Neumann. Probabilistic logics and the synthesis of reliable organisms from unreliable components. *Automata studies*, 34:43–98, 1956.

[42] D. C. Walker. *Mass notification and crisis communications: Planning, preparedness, and systems*. CRC Press, 2012.

[43] A. C. Yao. Protocols for secure computations. In *SFCS*, pages 160–164. IEEE, 1982.

[44] H. Zhang, A. Goel, and R. Govindan. Using the small-world model to improve freenet performance. In *INFOCOM*, volume 3, pages 1228–1237. IEEE, 2002.

[45] P. R. Zimmermann. *The official PGP user's guide*. MIT press, 1995.

[46] S. M. Zin, N. B. Anuar, M. L. M. M. Kiah, and I. Ahmedy. Survey of secure multipath routing protocols for WSNs. *J Netw Comput Appl*, 55:123–153, 2015.