Attack-Tolerance in Structured Networks via Multipath Routing Draft: 2016-02-16

EDWARD L. PLATT, University of Michigan
DANIEL M. ROMERO, University of Michigan

TODO

CCS Concepts:•**TODO**→ **TODO;**

Additional Key Words and Phrases: censorship, decentralization, fault tolerance, multipath, networks, peer-to-peer, trust

TODO: Fix bibtex encoding.

1. INTRODUCTION

Communication networks, expemplified by the Internet, have become ubiquitous and critical infrastructural for communities, organizations, and markets. 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. The key question is: what are the techniques and network structures that can be used to create robust, fault-tolerant systems?

Many complex networks, including the Internet and World-Wide Web, exhibit scale-free structure [Barabsi and Albert 1999; Barabsi and others 2009]. While scale-free networks tolerate random faults well, they are highly susceptible to adversarial faults, i.e. attacks [Albert et al. 2000]. For example, in 2008, YouTube suffered a worldwide outage for several hours when a service provider in Pakistan advertised false routing information [Hunter 2008]. 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 an attribute of complex communication networks themselves. General techniques for understanding and mitigating these vulnerabilities are needed.

In this paper, we formally evaluate the effects of network structure on attack-resistance. We also propose an efficient, fault-tolerant network architecture. Our analysis focuses on *structured networks*, in which links between nodes are constrained to a particular architecture. The proposed fault-tolerant architecture utilizes *multipath routing*, in which many possible paths exist betweeen two nodes on a communication network. In our case, multiple paths allow multiple copies of a message to be sent, which can be used for error-detection and error-correction by the receiver.

Our main contributions are:

— We present a formal *partial trust model*, which makes weaker transitivity assumptions than previous web-of-trust models, and use this model to quantify the effects of network structure on fault tolerance;

This work is supported by TODO.
Authors addresses: E.L. Platt and D.M. Romero, School of Information; email: {elplatt,drom}@umich.edu.

EC'16, July 24–28, 2016, Maastricht, The Netherlands. ACM 978-1-4503-3936-0/16/07.

http://dx.doi.org/10.1145/XXXXXXX.XXXXXXX

— We show that the probability of successfully detecting an adversarial fault increases exponentially with the number of disjoint untrusted paths between the neighborhoods of the sender and receiver, and that this value depends on network structure;
— We present a scalable, efficient, and attack-tolerant multipath routing algorithm on the butterfly network topology.

Paper organization. TODO

2. BACKGROUND AND RELATED WORK

In *structured networks*, 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 and distributed computing architectures [Kshemkalyani and Singhal 2008]. 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 [Lua et al. 2005].

Many distributed consensus protocols (such as those used by crypto-currencies) are designed to provide tolerance against arbitrary or adversarial faults. Byzantine agreement protocols [Castro et al. 1999; Lamport et al. 1982] provide tolerance against arbitrary faults (including attacks) under some circumstances, but are limited to small networks due to poor scalability. Proof-of-work [Dwork and Naor 1993; Nakamoto 2008] and proof-of-stake [King and Nadal 2012] provide better scalability, but wasteful of computational and energy resources. Federated Byzantine Agreement (FBA) [Mazires 2015] is scalable and optimal, guaranteeing that the protocol only fails when success is impossible, although it does not provide a way to evaluate the probability of failure. All of these protocols rely on the assumption that their cryptography cannot be compromised. While cryptography can be extremely resistant to technological attacks, it can still be compromised through coercion (e.g., legal action). In additon, the fault-tolerance of FBA depends on redundancy in the network structure but leaves an unanswerd question: how should a network be structured to achieve fault tolerance? This paper addresses both of these issues by analyzing the role of network structure on adversarial faults, without relying on assumptions of cryptographic integrity.

Multipath routing uses multiple paths when routing a message through a network, in contrsast to traditional *unipath* routing, which uses a single path. Multipath routing can have many benefits, including reduced congestion, increased throughput, and more reliability [Qadir et al. 2015]. Many of these routing protocols offer better security [Zin et al. 2015]. Some approaches utilize redundant paths as backups for increased fault tolerance [Alrajeh et al. 2013], and some specificially protect against adversarial faults [Khalil et al. 2010; Kohno et al. 2012; Lou and Kwon 2006]. The method of Liu et al. [Liu et al. 2012] routes multiple messages first to random peers and then to their final destination. The butterfly algorithm we present takes a conceptually similar approach. 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 trust model presented in this paper provides a more general way to evaluate the effect of network structure on adversarial fault-tolerance.

3. ATTACK-TOLERANT NETWORKS INFRASTRUCTURE

In this section, we describe the functional proprties required of an attack-tolerant network infrastructure and how those properties translate into constraints on network structure. We pay special attention to a property we call *stabilizing asymmetry*.

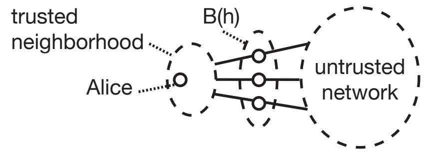

Fig. 1. TODO

3.1. Functional Properties

In most cases, infrastructure must remain functional as it grows; it must be *scalable*. Systems having single points of failure are less tolerant against faults at those points, which both raises the likelihood of a failure and creates targets where attackers might be able to focus their resources efficiently. Attack-tolerant infrastructure must minimize single points of failure; it must be *decentralized*. In some, but not all, cases it is also desirable that an infrastructure provides *secrecy*, protecting the contents of a message or the identity of the sender.

We add one additional property, which we call *stabilizing asymmetry*. In the context of international conflict, [Mack 1975] observed that power imbalance usually determines the outcome of a conflict (with the more powerful side winning) except in the special case of *asymmetric conflicts*. In asymmetric conflicts, the same level of resource expenditure yields different results for different parties. There are two possible cases: the attacker's resources are either more or less effective than the defender's. We call the latter case stabilizing asymmetry, because it lowers the incentive to attack. With this in mind, an attack-resistant infrastructure will benefit from a high level of stabilizing asymmetry.

3.2. Structural Properties

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. 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.

Redundancy in a network can help reduce single points of failure as well as increase fault-tolerance. One measure of redundancy is given by the ratio of edges to nodes [Baran and others 1964]. A single points of failure occur when a node holds a uniquely central position, and can be eliminated by adding alternative paths around the node, which increases the network's redundancy. Similarly, redundancy in a network can enable fault-tolerance techniques that reduce the effectiveness of attacks and creating stabilizing asymmetry. When secrecy is desired, redundancy can help by allowing messages to be divided into parts and sent along different paths so even if one path is compromised, the whole message is not revealed.

Networks with *vertex transitivity* have the greatest protection agasint single points of failure. In vertex transitive networks, for any pair of nodes, an edge-preserving map always exists from one node to the other. In other words, all nodes occupy structurally indistinguishable positions in the network.

3.3. Trust

Scale and indirect communication

Source-to-network

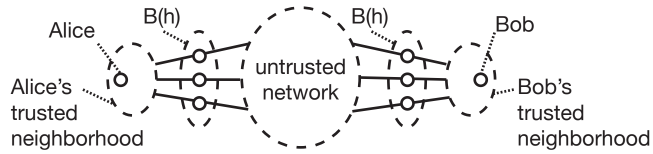

Fig. 2. TODO

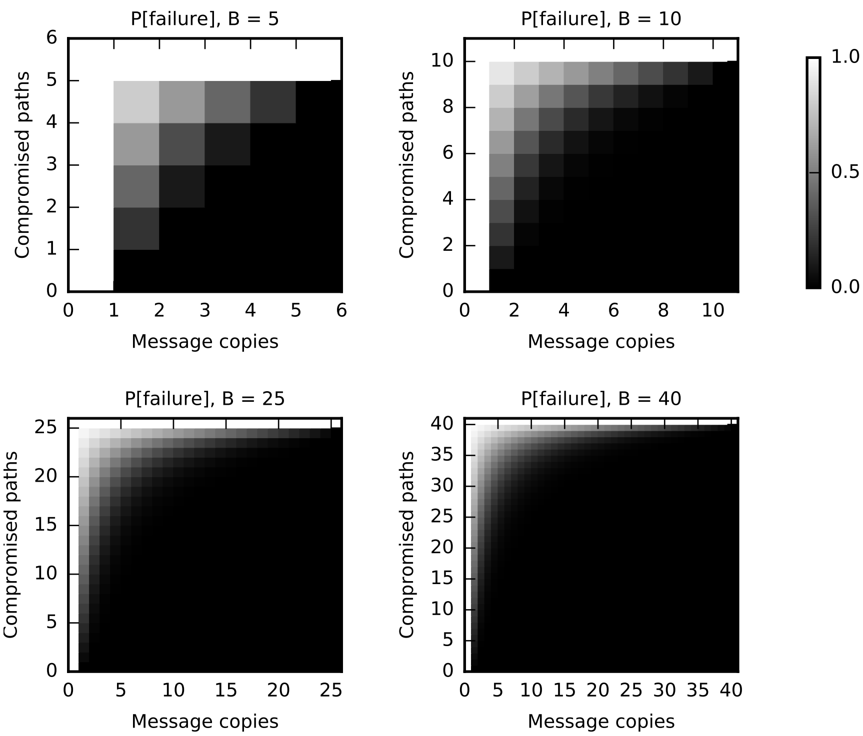

Fig. 3. TODO

Source-to-destination

4. MULTIPATH FAULT TOLERANCE

Fault tolerance with partial trust and redundancy.

4.1. Fault Tolerance

Results

5. MULTIPATH ROUTING ON THE BUTTERFLY TOPOLOGY

We now describe a Concurrent Multipath Routing scheme on a specific family of networks: the butterfly network []. Several variations on the butterfly network exist. Specifically, we utilize the wrap-around butterfly. We denote the m-dimensional di-

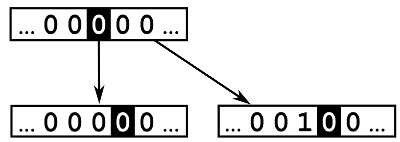

Fig. 4. Schematic illustration of the two types of edges in a directed butterfly network. The node (l, z) is shown as the bit string z with a square around the lth bit.

rected wrap-around butterfly as $\mathrm{wBF}(m)$:

$$\begin{aligned} \mathrm{wBF}(m) &= (V, E_\downarrow \cup E_\rightarrow) & (1)\\ V &= \mathbb{Z}_m \times \mathbb{Z}_{2^m} & (2)\\ E_\downarrow &= \{((l, z), (l+1 (\mathrm{mod}\ m), z) \\ &\quad | \, l \in \mathbb{Z}_d, z \in \mathbb{Z}_{2^m}\} & (3)\\ E_\rightarrow &= \{(l, z), (l+1 (\mathrm{mod}\ m), z \oplus 2^l) \\ &\quad | \, l \in \mathbb{Z}_d, z \in \mathbb{Z}_{2^m}\}, & (4) \end{aligned}$$

where $\oplus$ represents the bitwise XOR operator. Each node is associated with a level l and an m-bit integer z. There are two types of edges, shown in Figure 4. 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 bitstring equal to z with the lth bit flipped.

The wrap-around butterfly network has a number of properties making it useful for multipath routing:

Vertex transitivity. The problem of finding a route between arbitrary nodes v and w can be reduced to finding a route from node $(0, 0)$ to some $\tilde{w}$.
Logarithmic diameter. For any two nodes, the length of the shortest path between them is logorithmic in the size of the network.
Constant degree. In practical applications, each communication link requires additional resources, such as physical infrastructure or entries in a routing table. A constant degree limits the number of such resources needed as the network grows in size.

We now describe a routing scheme that provides 2^h redundant paths between the h-hop trusted neighborhoods of any two nodes in an m-bit wrap-around butterfly network. Utilizing vertex transitivity, we label the source node as $(0, 0)$ and denote the destination node as $w = (l_w, z_w)$.

Let s be an integer such that $0 \leq s < 2^h$. Let $v_s^{(t)} = (l^{(t)}, z^{(t)})$ be the tth node in the path labeled by s. For convenience, we will omit the subscript s. We define two partitionings of the integers in $\mathbb{Z}_{2^m}$: one having the lowest h bits matching the bits of s, and one having the h bits preceeding the destination level l_w matching s:

$$\begin{aligned} S_s &= \{z \in \mathbb{Z}_{2^m} | \forall i \in \mathbb{Z}_h z_i = s_i\} & (5)\\ R_s &= \{z \in \mathbb{Z}_{2^m} | \forall i \in \mathbb{Z}_h z_{(l_w - h + i)} = s_i\}. & (6) \end{aligned}$$

Note that if $r \neq s$, then $S_s \cap S_r = R_s \cap R_r = \emptyset$.

We now construct a path such that between trusted neighborhoods $z^{(t)}$ is always in S_s, R_s, or both, guaranteeing that the path does not overlap with the other paths $v_r^{(t)}$. Routing proceeds in stages, with the level l increasing by 1 at each hop. In Stage 1 ($0 \leq t < h$), down or down-right edges are chosen such that the 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

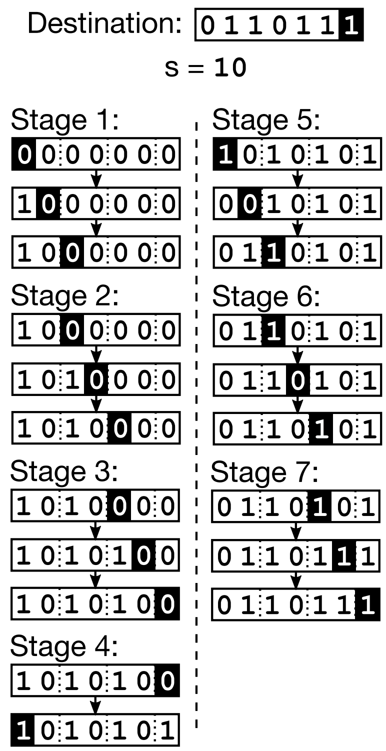

Fig. 5.

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

In Stage 2 ($h \leq t < l_w - h$), edges are chosen to make the tth bit of $z^{(t+1)}$ match the tth bit of z_w.

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

In Stage 4 ($l_w \leq t < m$), as in stage 2, paths are chosen such that the tth bit of $z^{(t+1)}$ matches z_w. After Stage 4, all bits of $z^{(m)}$ are equal to those of z_w except for the first h and the h preceeding index l_w. $z^{(m)}$ is also in both S_s and R_s.

At this point, we define $\tau = t - m$. In Stage 5 ($0 \leq \tau < h$), the first h bits of $z^{(t)}$ are set to match z_w, potentially removing $z^{(t)}$ from S_s.

In Stage 6 ($h \leq \tau < l_w - h$), all down edges are chosen, incrementing the level without any effect on $z^{(t)}$. At the end of Stage 6, $z^{(m+l_w-h)}$ is still in R_s and $v^{(m+l_w-h)}$ is now within the trusted neighborhood of w.

In the seventh, and final stage ($l_w - h \leq \tau < l_w$), the h bits of $z^{(t)}$ preceeding index l_w are set to match z_w. After this stage, $v^{(m+l_w)} = w$ and routing is complete.

6. DISCUSSION

Secrecy

Creation of the network. Improves on web of trust. Gives framework for determining where to build trust.

Applications Distributed apps: storage, email, cryptocurrency, secure multiparty computation. Wireless Sensor Networks

7. CONCLUSION

ACKNOWLEDGMENTS

The authors would like to thank TODO

REFERENCES

Rka Albert, Hawoong Jeong, and Albert-Lszl Barabsi. 2000. Error and attack tolerance of complex networks. *nature* 406, 6794 (2000), 378–382.

Nabil Ali Alrajeh, Mohamad Souheil Alabed, and Mohamed Shaaban Elwahiby. 2013. Secure ant-based routing protocol for wireless sensor network. *International Journal of Distributed Sensor Networks* 2013 (2013).

Albert-Lszl Barabsi and Rka Albert. 1999. Emergence of scaling in random networks. *science* 286, 5439 (1999), 509–512.

Albert-Lszl Barabsi and others. 2009. Scale-free networks: a decade and beyond. *science* 325, 5939 (2009), 412.

Paul Baran and others. 1964. On distributed communications. *Volumes I-XI, RAND Corporation Research Documents, August* (1964), 637–648.

Miguel Castro, Barbara Liskov, and others. 1999. Practical Byzantine fault tolerance. In *OSDI*, Vol. 99. 173–186.

Cynthia Dwork and Moni Naor. 1993. Pricing via processing or combatting junk mail. In *Advances in CryptologyCRYPTO92*. Springer, 139–147.

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

Issa Khalil, Saurabh Bagchi, Cristina N Rotaru, and Ness B Shroff. 2010. UnMask: Utilizing neighbor monitoring for attack mitigation in multihop wireless sensor networks. *Ad Hoc Networks* 8, 2 (2010), 148–164.

Sunny King and Scott Nadal. 2012. *Ppcoin: Peer-to-peer crypto-currency with proof-of-stake*. August.

Eitaro Kohno, Tomoya Okazaki, Mario Takeuchi, Tomoyuki Ohta, Yoshiaki Kakuda, and Masaki Aida. 2012. Improvement of assurance including security for wireless sensor networks using dispersed data transmission. *J. Comput. System Sci.* 78, 6 (2012), 1703–1715.

Ajay D Kshemkalyani and Mukesh Singhal. 2008. *Distributed computing: principles, algorithms, and systems*. Cambridge University Press.

Leslie Lamport, Robert Shostak, and Marshall Pease. 1982. The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)* 4, 3 (1982), 382–401.

Anfeng Liu, Zhongming Zheng, Chao Zhang, Zhigang Chen, and Xuemin Shen. 2012. Secure and energy-efficient disjoint multipath routing for WSNs. *Vehicular Technology, IEEE Transactions on* 61, 7 (2012), 3255–3265.

Wenjing Lou and Younggoo Kwon. 2006. H-SPREAD: a hybrid multipath scheme for secure and reliable data collection in wireless sensor networks. *Vehicular Technology, IEEE Transactions on* 55, 4 (2006), 1320–1330.

Eng Keong Lua, Jon Crowcroft, Marcelo Pias, Ritu Sharma, and Sharon Lim. 2005. A survey and comparison of peer-to-peer overlay network schemes. *Communications Surveys & Tutorials, IEEE* 7, 2 (2005), 72–93.

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

David Mazires. 2015. *Stellar Consensus Protocol: A Federated Model for Internet-level Consensus*.

Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. *bitcoin.org* (2008), 28.

Junaid Qadir, Anwaar Ali, Kok-Lim Alvin Yau, Arjuna Sathiaseelan, and Jon Crowcroft. 2015. Exploiting the power of multiplicity: a holistic survey of network-layer multipath. *Communications Surveys & Tutorials, IEEE* 17, 4 (2015), 2176–2213.

Shazana Md Zin, Nor Badrul Anuar, Miss Laiha Mat Mat Kiah, and Ismail Ahmedy. 2015. Survey of secure multipath routing protocols for WSNs. *Journal of Network and Computer Applications* 55 (2015), 123–153.