Davis Social Links: Leveraging Social Networks for Future Internet Communication

Lerone Banks, Prantik Bhattacharyya, Matthew Spear, Shyhtsun Felix Wu
Department of Computer Science
University of California, Davis
{ldbanks, pbhattacharyya, mjspear}@ucdavis.edu, wu@cs.ucdavis.edu

Abstract

In this paper, we present a social network based network communication architecture, Davis Social Links (DSL). DSL uses the trust and relationships inherent to human social networks to provide an enhanced communication architecture for future Internet designs. We begin with a conceptual discussion of how future network architectures can leverage social networks. Next, we describe the DSL architecture and how it provides to end-users control over their reachability within the network. We conclude with a discussion of the capability to manage dynamic communities within DSL.

1. Introduction

The simple yet powerful host-to-host communication protocols driving the current Internet architecture suffers from a lack of mechanisms to provide end-to-end protection. This shortcoming results in SPAM, DDoS attacks, and compromised machines. The primary cause of these vulnerabilities is the concept of **routable identity**, where routing information is encoded in the identities of communicating hosts. With routable identity, the only information that a host needs to contact another is the target host's identity. Once this identity (and consequently its routable identity) is revealed, a host has no control over which other hosts can use it and no power to revoke access to itself (or its identity) if it is being abused. An email address, which allows any host to contact the owner of the email address, is a great example. The few ways an **owner** can prevent being contacted with the email address are either the impractical option of getting a new email address or the inefficient option of manually assigning message senders to black/white lists. This leaves the owner of a routable identity vulnerable to unwanted communication.

The concept of relationships between users (or nodes) and their trust are key components of network communication. Online Social Networks (OSNs) are the only networking applications that explicitly utilize relationships between nodes to route information e.g. in Facebook, the concept of any-to-any communication (although turned on by default) **is not an inherent feature of the system the way it is for email and IP networks**. Instead, messages are routed between two nodes based on the existence of social link paths that represent a series of revocable relationships between the nodes on the path.

In this paper, we present a network architecture that utilizes social relationships to enhance network commuication. The key contribution of the architecture is that it provides a mechanism for mitigating the vulnerabilities due to routable identity by providing a means for message receipients to control their received messages. An intuitive description and the core building blocks of the architecture of DSL are presented in section 2. 'Receiver Controllability', 'Trust Maintenance' and 'Dynamic Communities', the key features of DSL are discussed in section 3. We conclude in section 4 with a brief outline of our ongoing effort to realize the concepts of DSL through a Facebook application.

2. Davis Social Link Model

In [1] we introduced Davis Social Links (DSL) as an architecture for decentralized peer-to-peer communication. The goal was to identify the core components of a robust general communication system that can provide protection against the inherent vulnerabilities of Internet based communication. The key updates described in this paper are the use of policies to control attribute propagation, simplification of the route discovery process, improved concepts about trust and its maintenance and the concept of dynamic community.

Due to space limitations, we omit a survey of the work related to the many ares covered by DSL though some of them have been previously discussed in [1]. We begin this section with an intuitive description of the architecture.

2.1. Case Study and Motivation

Under the current Internet infrastructure, the exact path from source to destination is chosen independent of the semantic properties of the applications. For instance, in a typical enterprise, a new employee Alice sends a message directly to Bob. If Bob does not know Alice, Bob may decide that the message is spam from a questionable source. However, if Bob receives the exact same message forwarded from a second-line manager, Chris, then value of the message is much higher. In this simple example, Bob and Chris have a trust relationship (or 'social link'). If the routing of the message follows a particular sequence of social links, then every party in this communication transaction will have critical semantic information to support this transaction more effectively. Under such a model, Alice will need to find a valid social path before she can communicate with Bob. The preceding example describes a manual process that should be an automatic network function. DSL automates provides this functionality by utilizing the social path as a method of information/communication access control.

The advantage of DSL architecture is its ability to effectively defend our communication infrastructure against many large-scale attacks that the current Internet architecture is not designed to handle. For instance, under DSL, a spammer needs to find a valid social path (i.e. a sequence of social links) to reach each of it's victims. This requirement for communication provides the capability of the victim to de-prioritize or even remove certain incoming social links being used by the spammer (we describe this process in the section 3.2). Defending networks (i.e., the infrastructure plus the nodes/hosts/routers) against large-scale attacks such as worm, DDoS, spam, phishing, and botnet is a primary motivation for the development of DSL.

2.2. Core Building Blocks

2.2.1 The Social Graph

The first component of our model is the social graph with assumptions that it exhibits standard small-world graph properties as outlined in [6]. A formal mathematical definition for a social graph can be found in [5].

2.2.2 Attributes

An attribute is a free-form bit string associated with a user. Attributes have no semantic/structural meaning from the point of view of DSL. They are keywords representing the characteristics of a user. It is possible for the keywords themselves to have some application layer meaning, but the DSL communication layer is unaware of their semantics. In DSL, there are two types of attributes:

Profile Attributes: user-defined/maintained keywords and their associated policies that represent the user within the network. Each node v in G maintains its own user-defined/configured set of profile attributes, ${K_v}^{PAtt}$.

Friendly Attributes: profile attributes of other users in the network for which a user satisfies the policy associated with the attribute. The set of friendly attributes, ${K_v}^{FAtt}$ for a node v represents routing control information for v and is determined by the characteristics of v as described in the policy subsection.

It is important to note the distinction between profile attributes and identity. Identity is a globally unique property of an entity that distinguishes it from other entities while a profile attribute is simply a loose representation not intended to denote uniqueness.

2.2.3 Policy

The set of profile attributes of a user v, represented by ${K_v}^{PAtt}$, is the collection of all individual profile attributes, k, each of which are associated with three different values: D represents hop count, T represents trust value, C represents community attributes. This triplet defines the policy of a profile attribute, k, as $Policy(k)$. Stated formally:

$$\forall k \in {K_v}^{PAtt},\ \exists\ Policy(k) = \text{[D,T,C]} \qquad (1)$$

Hop count, D, is a non-negative integer for the maximum path length from the owner of k to the query-issuing user in the social graph G. T is the minimum trust value that each subsequent node must have for each preceding node on the path from the query-issuing user to the owner of k. The set of community attributes, C, is the set of profile attributes that every node on the path from the query-issuing user to the owner of k must own in order to use the attribute k to access the owner of k.

For $Policy(k)$, each member of the policy triplet represents a criteria that must be met by any node attempting to use k to access the owner(s) of k. It must be noted that many nodes may share the attribute k. The policy associated with each instance of k and the characteristics of the user that is querying for k will determine the set of all owners of k that are returned in the query result representing all possible potential receivers.

2.3. Higher-level Concepts

2.3.1 Profile Attribute Access

Because current OSNs are centralized our model uses a central server (referred to as an Oracle) to maintain the social graph and individual node data. Though our model is analogous to current architectures that rely on central gateways and DNS for network traffic, our architecture is not dependent on centralization but uses it as an abstraction in order

to focus on developing the primary features of DSL. Under the DSL model, the Oracle verifies that a querying node satisfies the policies of query keywords for a particular node.

For example: u, a user in the OSN, controls and maintains the following set of attributes or keywords, K_u = {Professor, Computer Science, UC Davis, Soccer}. These keywords can come from his OSN user profile or any other source e.g. network affiliation, community subscriptions (a discussion on this topic is outside the scope of this paper). For each member k of K_u, user u assigns a policy, $Policy(k)$ e.g. , $Policy$('Professor') = [D $\leq$ 5, T $\geq$ 0.2, C = {'Computer Science','Professor'}].

The set of nodes that can access node u with k is denoted by W. An element of W is denoted by w. Thus, $w \in W | k \in {K_w}^{FAtt}$ if for w, $d[w]_u = 5$, where $d[w]_u$ is the topological distance between users w and u in G, $T_{fz} \geq 0.2$ for all pairs of nodes, f and z, that are on a path from w to u and {'Computer Science','Professor'} $\subset {K_y}^{PAtt}$ for each node y on the path from w to u. If user u creates similar policies for the rest of the members of K_u, then the resultant set of keywords and their policies forms the set ${K_u}^{PAtt}$.

2.3.2 DSL Routing

Routing in DSL is a two-step process. First, the sending node issues a query for a path to a set of nodes owning a set of keywords. The Oracle then responds with a set of paths to the targets for which the sending node satisfies the policies of the query keywords.

Querying: Each node v in G can query for it's set of friendly keywords, ${K_v}^{FAtt}$. Theoretically, a user can query for any set of keywords but a query is successful in returning a list of recipients only if the query keyword belongs to the set of friendly attributes for the querying node. A querying node can get this set from the Oracle. Note that set, ${K_v}^{FAtt}$, does not indicate which users v can access, only that she can access a set of other users with the set ${K_v}^{FAtt}$.

Query Matching: Given a query Q consisting of some set of keywords, Q_k, issued by node v in G. The Oracle determines the result set by finding the set of users, $\text{Receivers}_{\text{Potential}}$, for which $Q_k \subset {K_u}^{PAtt}$ for $u \in V$. Then for each user, u in $\text{Receivers}_{\text{Potential}}$, checks whether v and each node on the path from u to v satisfies the policy for each keyword in Q_k. $Policy(k)_u$ denotes the policy for keyword $k \in {K_u}^{PAtt}$ for user u. A satisfactory result leads to the inclusion of u in the final set of potentially accessible receivers, $\text{Receivers}_{\text{Final}}$ from source v.

3. Key Features

We discuss the key features of DSL: 'Receiver Controllability', 'Trust Maintenance' and 'Dynamic Communities' in this section.

3.1. Receiver Controllability

OSNs inherently provide mechanisms for limiting access to users and determining authorization of message senders via social connection information. In DSL, we use this property of social networks to allow a user to control its reachability within the network via trust and reputation derived from previous interactions and social distance. The result is that routing paths are determined by destination nodes through their keyword policies. By controlling the depth of the social graph at which keywords can be used and providing a threshold for trust, a user can implicitly grant access to users beyond those to which it is directly connected while maintaining control over their reachability within the network. Consequently, the social paths that exist between users, and not routable identities, are the main factor in the routing of messages between users. As a result, users gain control over the sources of received messages without having to use mechanisms such as black/white lists to grant explicit access to every potential source in the social graph. If a user is receiving too many bad messages via one of its neighbors, it can reduce the trust value of that neighbor or sever the link forcing message senders to find more trustworthy paths. Alternatively, the user can change the policy of or remove the keywords that allow bad messages. These options represent a distinct advantage over dropping and getting a new routable identity because problem paths are reduced or eliminated and benign paths function normally while requiring minimum action from the user.

3.2. Trust Maintenance

DSL integrates trust as a factor in reachability. Reputation systems (e.g. *ebay.com*) take a global approach to reputation. [2] shows that any global trust system is exploitable. Thus, we utilize the concept of trust similar to the manifestation of trust in human relationships, where connected users (friends) have an individual concept of trust in their friendship. DSL provides a mechanism where nodes can explicitly maintain trust values for their neighbors and can use these trust values as a means of controlling the traffic that reaches them. DSL utilizes a 'personalized reputation model,' where trust is based on an individual perspective, creating a trust system that is provably non-exploitable as compared to global reputation models.

Motivated by [4], our model consists of a trust structure, $T_{uz} = \langle g_{uz}, b_{uz} \rangle$ where g_{uz} is the number of good interactions and b_{uz} is the number of bad interactions that node u has had with node z. A user maintains the trust values for its direct neighbors by giving feedback about the quality of interactions to produce a relative reputation value R_{zu} that represents the reputation value of node z according to node u. Changes in reputation based on a received message are

propagated along the reverse path of the message. Analysis from [4] shows this is a powerful mechanism to exclude misbehaving nodes from participation in the network. Under this model, given some trust value, t contained in the policy of some profile attribute $PA_u \in {K_u}^{PAtt}$ for a node u, some node z can use PA_u to access u if $R_{zu} \geq t$ if z is directly connected to u, otherwise z must have a friend f where $R_{zf} \geq t$ and $R_{fu} \geq t$.

3.3. Dynamic Communities

DSL inherently provides a powerful mechanism for the formation and management of dynamic communities. A concept recently added to the architecture of DSL, implicit dynamic communities differ from groups, communities, or networks currently supported by most OSNs in that they are formed without any formal registration and in order to become a member, a user must have a social relationship with a member of the community. This mechanism exists via profile and community attributes and is best illustrated with following examples.

Background: We consider the case study of online user interactions within Facebook groups, blogs and other message boards regarding the 'Proposition 8' ballot measure during the 2008 State of California election [3]. Due to open access, users left scathing comments on boards that held opposite views. The only response of the offended community was to remove the comments and block the pseudo-identity (which is easily defeated by a Sybil attack). Dynamic communities within DSL provide a solution to such situations.

Attribute-based Formation: Here, we show how to build an exclusive community among connected nodes. Given nodes A,B,C: node A owns the profile attributes 'Prop 8' and 'YES' both with the policies, $Policy$('Prop 8' | 'YES') = [$D \leq 2,\ T \geq 0,\ C = \{$'Prop 8', 'YES'$\}$] because it favors the proposition and nodes B and C own the profile attributes 'Prop 8' and 'NO' both with the policies, $Policy$('Prop 8' | 'NO') = [$D \leq 2,\ T \geq 0,\ C = \{$'Prop 8', 'NO'$\}$] signaling their opposition. As a result of each nodes choice of profile attributes, nodes B and C are able to form an 'Anti-Prop 8' community that is unreachable by node A via the attribute 'Prop 8' regardless of whether A is on social link paths to both B and C and A owns the attribute 'Prop 8'.

Relationship-based Formation: Next we demonstrate how to handle when a node becomes aware of the keywords of a community for which it is not a member. Continuing the previous example, node A adds the attribute 'NO' with $Policy$('NO') = [$D \leq 2,\ T \geq 0,\ C = \{$'Prop 8', 'NO'$\}$] to join the 'Anti-Prop 8' community and sends an offensive message to the community members. Upon realizing the path of the negative message to the community via his association with A, node B severs its friendship with node A. As a result, node A is removed from the 'Anti-Prop 8' community in spite of knowing the community keywords. This final example shows the necessity of friendship as a requirement for community membership. Node B does not have to sever it's tie with A in order to remove it from the community, reducing it's trust value for A would have worked as well while still maintaining the social connection.

These examples showcase a key advantage of DSL: **the ability of users to control the formation and their membership in dynamic communities by updating their profile attributes and social links.**

4. Conclusion and Future Work

The ultimate goal of the Davis Social Links project is to develop a dynamic, scalable, trust-based, decentralized communication system/architecture for large-scale networks (10 million $\sim$ 10 billion nodes) with considerations of security/robustness, manageability, high-performance, and mobility. In this paper we present significant conceptual advances towards realizing an architecture that uses ideas from online social networks to create an improved communication model suitable for the Internet of the future. Key contributions of this architecture are feasible methods for providing receiver controllability over received messages, the explicit integration of trust into network routing, and the ability to form dynamic communities, all via attribute policies and propagation.

Acknowledgments. We thank Xiaoming Lu, Shaozhi Ye, Juan Lang and Ankush Garg for their valuable advice and comments. This work was supported by NSF (award number 0832202).

References

[1] L. Banks, S. Ye, Y. Huang, and S. F. Wu. Davis social links: integrating social networks with internet routing. In *LSAD '07: Proceedings of the 2007 workshop on Large scale attack defense*, pages 121–128, New York, NY, USA, 2007. ACM.

[2] D. DeFigueiredo and E. Barr. Trustdavis: A non-exploitable online reputation system. In *CEC '05: Proceedings of the Seventh IEEE International Conference on E-Commerce Technology*, pages 274–283, Washington, DC, USA, 2005. IEEE Computer Society.

[3] S. of California. Voter's guide available at http://www.voterguide.sos.ca.gov/title-sum/prop8-title-sum.htm.

[4] M. Spear, J. Lang, X. Lu, N. Matloff, and S. F. Wu. Messagereaper: Using social behavior to reduce malicious activity in networks. Computer Science, UC Davis, Techincal Report CSE-2007-9.

[5] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

[6] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998.