

Generic Reliability Trust Model

Glenn Mahoney
Wendy Myrvold
Gholamali C. Shoja

Department of Computer Science, University of Victoria, Canada V8W 3P6
Email: {gmahoney,wendym,gshoja}@cs.uvic.ca

Abstract—Economic and social activity is increasingly reflected in operations on digital objects and network-mediated interactions between digital entities. Trust is a prerequisite for many of these interactions, particularly if items of value are to be exchanged. In this paper the probabilistic model and computational approaches found in some network reliability models are applied to modelling computational trust. The result is a new generalized trust model called the Generic Reliability Trust Model or GRTM and a new transitive trust metric called Hop-Count Limited Transitive Trust (HLTT). A conservative approximation heuristic is defined which leads to more practical algorithm performance. Results from a JAVA-based implementation, utilizing an XML-based trust-graph representation and a random power-law trust graph generator, demonstrate potential for application to large ad-hoc trust networks.

Keywords: computational trust, network reliability, identity and trust management, trust technologies, technologies for building trust.

I. INTRODUCTION

Human trust is a complex and context sensitive interaction of risk, value, experience, expectation, uncertainty, social relationships, and individual human qualities [3], [15]. Humans operating in a physical world use a variety of interpersonal skills and social arrangements to create and evaluate trust in others. We have an intuitive notion of what is meant by the word trust and live in a rich social web that sustains our ability to trust other people and organizations. This trust enables us to interact in a range of situations. The role of trust in human/social systems is to guide and support decisions where the outcomes depend on the good behaviour of others and where results cannot be completely controlled [3], [9], [15]. For this paper, trust is defined as follows:

Definition 1.1: Trust is one's reasonable expectation of a positive outcome in a situation where there is less than full control over the actions of the participants.

In a network-mediated virtual world, where interaction is represented by the exchange of messages between digital entities, trust is a fundamental challenge; identity is suspect, the exchange medium is suspect, and there is often no history of interaction or expectation of future interaction between particular entities [20]. Nevertheless, trust remains a requirement for maintenance of cooperative social groups and online economic activity [3]. In the virtual world of the online auction site eBay, buyers and sellers do not know each other, with 98.9% of seller-buyer pairs conducting less than five transactions over a five-month period [19, Section 4]. Yet, after being created in 1995, by 2002 eBay had 61.7 million

registered users, 638 million listed items, and facilitated \$14.9 billion dollars (US) in gross sales [7]. These results were supported by providing good-enough trust to an effectively anonymous community of ad-hoc buyers and sellers –

“The key to eBay's success is trust. Trust between the buyers and sellers who make up the eBay community. And trust between the user and eBay, the company.” – eBay Web Site [8]

Generalized trust reasoning capabilities should enable parties – individuals, corporations, human and software agents – to operate in more situations in a world of information-based, network-mediated relationships. Some trust between a larger set of parties enables an organization to perform its function with greater scale (interact with more parties) and less friction. These capabilities could be used by various parties to help decide with whom and how much information to share. As well, these decisions could be based on multiple sources of trust-related information.

Computational trust is formal trust definitions or rules for representing and evaluating trust-like relationships implementable in software [15]. This emerging field of research applies computational models to trust decisions within virtual environments. For the rest of this paper, the word trust should be interpreted as computational trust.

The specific focus of this paper is on trust mechanisms supporting ad-hoc, network-mediated interaction within a large pool of potential interactors, such as Internet users. In this application domain, the parties have little direct knowledge or formal relationships or explicit contracts with each other. They cannot utilize human trust due to a lack of physical (face-to-face) presence or human relationships. Some trust-related mechanisms suitable for this domain are available or emerging. For example, we can authenticate an identity and secure message contents through cryptography, evaluate past behaviour through reputations, and create some expectation of future interaction through community membership [6], [9], [19]. Many of these solutions are limited or application-specific, and do not support the more general notion of trust used for this paper. Validating the identity of another party and having a secure channel to communicate with that party does not mean you can trust them, they may be a crook. Online communities and associated reputations are often isolated, application-specific domains. Our goal is to define a general representation of trust between a set of entities and a trust

metric providing a means to evaluate some measure of the trust between particular entity pairs.

This paper generalizes and extends the seed of the model in the Maurer Confidence Valuation (MCV) trust metric [17]. The probabilistic model and computational framework of Network Reliability is applied to create the Generic Reliability Trust Model or GRTM. Using the GRTM framework, a new trust metric is defined: Hop-count Limited Transitive Trust (HLTT). The potential for application of this new trust model and metric will be demonstrated through a practical approximation heuristic for the metric and performance results from a JAVA implementation.

The remainder of the paper is organized as follows. Section II presents an abstract trust model and a brief survey of existing models. Section III presents the new GRTM model and the HLTT metric. Section IV presents general computational algorithms for GRTM, an approximation heuristic, and experimental implementation results. Section VI concludes with a brief comparison of GRTM/HLTT and the surveyed models, and suggestions for future research. More details of this research are presented in [14].

II. ABSTRACT TRUST MODEL

This section presents an *abstract trust model* that identifies common aspects of trust models. This abstract model describes important objects and relationships required to represent and evaluate trust in various situations. Its purpose in this paper is to provide a conceptual framework motivating new generalized trust models and supporting comparison between some existing models and the new GRTM model.

Entities are the subject objects of trust relationships. They are the sources, targets, and intermediaries – people, agents, software objects, or systems – which are the subjects of, participate in, or provide supporting information for trust decisions. The *local* or *source entity* is the entity attempting to reason about its subjective trust of another entity, called the *remote* or *target entity*. The local entity may use information supplied by third parties referred to as *intermediate* or *recommender entities*.

A *trust value* is some measure or quantification assigned by a local entity to its belief in the trustworthiness of another entity. A few possible types of trust value include boolean, confidence, or discrete values such as {no-trust, partial, complete} [2]. The trust value often indicates the expectation of a successful interaction, through which some desired outcome will be achieved. Marsh [15] provides an extensive discussion of trust values. For example, using confidence-based trust values, high trust is a value from 0.75 to 0.9, and low trust is from 0 to 0.25.

Trust is subject-matter specific; that Sally trusts Tom with her lawn mower does not mean she also trusts him with her car. Another way of expressing this often implied aspect of trust is found in the following expansion from [4] where X stands for the subject-matter:

“A trusts B” is shorthand for “A trusts B about X under certain conditions.”

Direct trust is some entity’s independent belief in the trustworthiness of another entity. Direct trust, and trust in general, is not symmetric; that Sally trusts Bob does not imply that Bob trusts Sally. *Indirect trust* is some entity’s belief about the trustworthiness of another entity which is derived from the beliefs of other entities. Indirect trust, and trust generally, is not always transitive; that Sally trusts Bob, and Bob trusts Tom, does not always imply that Sally trusts Tom. A *recommendation* is a statement of direct trust about a remote entity made by an intermediate entity. *Trust evidence* is the direct trust of some set of entities. A *trust situation* is a subset of trust evidence representing some subject-matter.

Trust roots, also called *seeds of trust*, are the positive assumptions about specific entities made by all entities in some community. The label “authority” is often applied to an entity that is the subject of a trust root. Models with formally modelled trust roots can be termed *centralized* and those without can be termed *distributed*. A distributed trust model can be used to represent a centralized model by having some direct trust common across all entities.

A *trust model* is some definition of entities, trust values, trust subject-matter, direct trust, indirect trust, and trust roots. A *trust metric* within some trust model is a function that computes a trust value from a trust situation. It defines how some local entity can utilize trust evidence and indirect trust to reach a conclusion about the trustworthiness of some remote entity in a specific situation. Transitivity is the primary means for deriving indirect trust. The conditions under which transitivity is allowed are a defining aspect of a trust metric. Trust metrics can be characterized by their underlying theoretical models. *Arithmetic type metrics* process evidence using simple arithmetic operations; for example, computing the average value of input trust values. *Chain of Proof (COP) type metrics* perform a boolean validity test using a chain of evidence; if each link in the chain is valid and correctly linked to the next, then the chain as a whole is valid. A *probabilistic type metric* utilizes some probability-based measure, such as risk or confidence, and processes trust evidence using some probability-preserving operations.

The subjectivity and scalability of trust are determined by the specification of some trust model and trust metric. If the metric does not support any transitivity and there are no trust roots, then all trust will be strictly localized or personal, based solely on a local entity’s beliefs. It is for this reason that indirect trust utilizing so-called “*weak ties*” (the bridge individuals through which smaller social clusters are connected) is so important in social networks. These weak ties establish connections with other social clusters and enable wide spread communication in larger social networks [10].

An *identity* is a label used to refer to an entity. Using this simple definition, a person’s name is their identity. *Identity trust* is some measure of the confidence in the one-to-one mapping between an entity and a identity; that an entity is who s/he says that s/he is and that a given identity used by multiple parties refers to the same entity. *Identity authentication* is the verification of the identity of an entity [9].

A *trust mechanism* is the combination of a trust model and metric. A *trust system* is some operational trust mechanism including provisions for handling trust evidence, application interfaces for obtaining trust metric results, some identity authentication capability, and may also include mechanisms for managing trust roots. Table I presents a short list of real or proposed trust mechanisms, described using the abstract trust model framework.

III. GENERIC RELIABILITY TRUST MODEL

Reliability theory is the study of the performance of a system of failure-prone elements. Network reliability is concerned with the reliability of computer communications networks, specifically the ability of a network to carry out a desired operation [5]. The following network reliability definitions are taken from [5].

A *probabilistic directed graph* $G = (V(G), E(G))$ consists of a vertex set $V(G)$ and the set of arcs $E(G)$ where each arc in $E(G)$ corresponds to an ordered pair of vertices from $V(G)$. It is assumed that each arc e has an associated probability p_e of being operational. A *state* of G is a subset $S \subseteq E(G)$, interpreted to mean that all arcs in S are operational and all arcs in $E - S$ have failed. So a state S corresponds to the subgraph G_S of G given by $V(G_S) = V(G)$ and $E(G_S) = S$.

To describe a specific reliability model it is necessary to provide rules, here called *operational criteria*, for distinguishing between operational and non-operational states. For example, given a directed graph G , a *directed path* of length k is composed of an alternating sequence of vertices and arcs of the form $v_0, e_1 = (v_0, v_1), v_1, e_2 = (v_1, v_2), \dots, v_{k-1}, e_k = (v_{k-1}, v_k), v_k$. An *s, t-path* is a directed path which starts with s and ends with t . Vertex s is *connected to* t if there is an *s, t-path* in G . For a directed *s, t-reliability* model, the network is defined to be operational if and only if there is an operational *s, t-path* in the network. Some possible definitions of operational are given in Table III [21].

TABLE II
TRADITIONAL NETWORK RELIABILITY OPERATIONAL CRITERIA

Reliability Measure	Operational Criteria on state S
<i>Two-terminal</i>	there exists an <i>s, t-path</i> in G_S
<i>Source-to-K-terminal</i>	$\forall t \in K \subseteq V(G)$, there exists an <i>s, t-path</i> in G_S
<i>Source-to-all-terminal</i>	$\forall t \in V(G) - s$, there exists an <i>s, t-path</i> in G_S

The *reliability* of a network G , denoted $Rel(G)$, is the probability of obtaining an operational state. Assuming that arc failures are independent, $Rel(G)$ can be defined using state-space enumeration as follows [5, p. 9]:

$$Rel(G) = \sum_{\substack{S \subseteq E(G); \\ S \text{ is operational}}} \prod_{e \in S} p_e \cdot \prod_{e \in E(G) - S} (1 - p_e), \quad (1)$$

where p_e is the probability that
that the edge e is operational.

A *generic reliability problem* is to determine the probability of having an operational state given a probabilistic directed graph G and the operational criteria. If every superset of a operational state is operational, then the system is *coherent*. A useful property of all of the operational criteria to be defined in this paper is that the operational states are coherent. In general, let $OP(G)$ denote the set of all operational states for some network reliability model. A *minimal operational state of a coherent system* is a state $S \in OP(G)$ where for all $a \in S$, $S - \{a\} \notin OP(G)$. For coherent systems, there exist exact algorithms for the reliability problem which need only consider minimal operational states [5, p. 11].

The remainder of this section defines the Generic Reliability Trust Model as a generalized reliability problem. A *trust graph* is a labeled directed multigraph $G = (V(G), E(G))$, with the vertex set $V(G)$ representing entities and the set of arcs E consisting of ordered pairs of vertices (u, v) representing u 's trust in v . Each arc $e = (u, v)$ is labeled with a *trust label* of the form $\langle l, c \rangle$ where,

- $l \geq 0$ is an integer representing a level for the trust
- where the specific meaning is defined by
- a particular trust metric, but
- for any given level l , trust graphs can have at
- most one arc (u, v) labelled with l , and
- c is a confidence value, $c \in [0, 1]$.

In a trust graph G , for each arc e in $E(G)$, the values of the associated label $\langle l, c \rangle$ can be referenced as l_e and c_e and the arc's operational probability is $p_e = c_e$. The level value generally represents an assessment of the competence of the entity to act as an intermediary; the specific meaning is defined by a metric. The confidence value c_e is the probability of a successful outcome from an interaction between entities u and v at trust level l_e as determined by u ; this represents u 's confidence in v and it is assumed that it is independent of any other confidence. It is assumed that G relates to a single subject-matter.

A *generic reliability trust metric* is a function which gives a trust value as the probability of obtaining an operational state. Computing the trust metric is a reliability problem; that is

$$Trust(G) = Rel(G).$$

Concrete members of this family of trust metrics are defined by specifying operational criteria associated with a general reliability problem. This process is demonstrated in the next section by the definition of a new metric within this framework.

A. Hop-Count Limited Transitive Trust (HLTT)

Hop-Count Limited Transitive Trust (HLTT) is a new trust metric that fits into the framework of a generic reliability trust metric. For a trust graph G and arc $e = (s, t)$ in $E(G)$, the trust level value l_e specifies the maximum *s, t-path* length representing some trust from s to t . In addition, the trust graphs are restricted to be digraphs.

TABLE I
SURVEY OF TRUST MECHANISMS

Trust Mechanism	Metric Type	Subject-matter	Evidence	Direct Trust	Indirect Trust	Trust Roots
X.509 PKI [12]	COP	public key authentication	digital certificates	local storage of trusted certificates	certificate signed by CA	Certificate Authority (CA)
PGP [23]	COP	public key authentication	digital certificates	local certificate key ring	certificate signed by "introducer"	application-specific
Trust Management [16]	COP and policy evaluation	access control	digital certificates and application-specific policy rules	local policies	signed certificate (credential)	application-specific
Distributed Trust [1]	arithmetic	categories	statements	local database of statements	recommendation protocol	application-specific
Network Flow Trust Metric [13]	Network-flow calculation to test for chain-of-proof sufficiency	public key authentication	digital certificates	local certificate	signed certificate	application-specific
Bayesian Network-Based Trust Model (BNTM) [22]	arithmetic	generalized, subject-matter adaption using local Bayesian network	transaction history (satisfaction counters)	local satisfaction counters and Bayesian network	recommendations obtained by satisfaction query	application-specific
Maurer Confidence Valuation [17]	probabilistic	public key authentication (a generalized model of a chain-of-proof system)	statements representing assertions about certificates	local certificate	signed certificate is a recommendation	application-specific
GRTM + HLTT (Section III)	probabilistic	generalized	statements represented as labeled arcs in a graph	local statement	remote statement	application-specific

Definition 3.1: Given a trust graph G and vertices s and t , an *HL-path* is an s, t -path P of length k , with the form $v_0, e_1, v_1, e_2, \dots, e_k, v_k$, such that for all $e_i \in P$, $l_{e_i} \geq k - i$.

For a trust graph G and vertices s and t in $V(G)$, *Hop-Count Limited Transitive Trust*, or $HLTT(s, t, G)$, is a generic reliability trust metric with the operational criteria that a state $S \subseteq E(G)$ is operational if and only if there exists an HL-path $\subseteq G_S$, from s to t .

A *derived statement* is a 3-tuple of the form $\langle u, v, l \rangle$ where u and v are entities and l is a non-negative integer representing a trust level. These statements provide an additional mechanism for defining operational criteria. Initial derived statements can be created from a trust graph G using the rule in Definition 3.2.

Definition 3.2: If (u, v) , labeled $\langle l, c \rangle$, is in $E(G)$ then $\langle u, v, l \rangle$ is a derived statement.

Additional derived statements are created by applying a set R of transitivity rules. For example, the set of HLTT transitivity rules are found in Definition 3.3.

Definition 3.3: HLTT Transitive Rules

(1) Let $i > 0$, $j \geq 0$, and $k = \min(i - 1, j)$.

If $\langle u, v, i \rangle$ and $\langle v, x, j \rangle$ are derived statements then $\langle u, x, k \rangle$ is a derived statement.

(2) Let $i > 0$.

If $\langle u, v, i \rangle$ is derived statement then $\langle u, v, i - 1 \rangle$ is a derived statement.

Thus, an alternate definition of operational criteria for $HLTT(s, t, G)$, that does not use the HL-path concept, is that the state $S \subseteq E(G)$ is operational if and only if the derived statement $\langle s, t, 0 \rangle$ exists in the reflexive, transitive closure of the initial statements corresponding to arcs in S under the HLTT Transitivity Rules.

This use of derived statements can be generalized for multiple trust metrics within the GRTM framework. Given some set of transitivity rules R for derived statements, the trust graph G , and vertices s and t , a *generic two-terminal reliability trust metric* is a generic reliability trust metric with the operational criteria that the state $S \subseteq E(G)$ is operational if and only if the derived statement $\langle s, t, 0 \rangle$ exists in the reflexive, transitive closure of the initial statements corresponding to arcs in S under R .

As an example application of the HLTT metric based on Figure 1, some support for trust between Alice and Bob is indicated by the ability to derive the statement $\langle Alice, Bob, 0 \rangle$

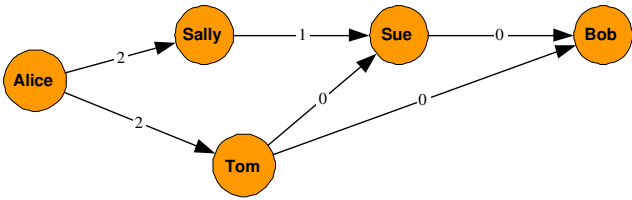


Fig. 1. Trust Graph between Alice and Bob with HLTT semantics

from the initial statements represented by this graph. Calculating a confidence value using the inclusion-exclusion approach involves enumerating all of the operational states and then applying Equation 2. Definition 3.3 produces the following minimal operational states for trust from Alice to Bob:

$$\{(Alice, Sally), (Sally, Sue), (Sue, Bob)\}$$

$$\{(Alice, Tom), (Tom, Bob)\}$$

Assuming $p_e = 0.8$ the resulting trust metric value is 0.82. Observe that the level zero label on the arc (Tom, Sue) means there is no HL-path from Alice to Bob through Tom and Sue, that Alice does not trust Tom to act as an recommender.

IV. GENERAL ALGORITHMS

In general, computing reliability is a #P-complete problem [5]. For a graph G with m arcs, complete state enumeration would generate or consider 2^m states representing all possible subsets of arcs in $E(G)$. Two general algorithms for calculating an exact value for a variety of reliability problems are based on inclusion-exclusion and the factoring theorem [5], [11], [21]. They require exponential time and, in the case of inclusion-exclusion, possibly exponential memory.

The generalized inclusion-exclusion algorithm to calculate reliability has two phases:

- 1) Search – determine all minimal operational states, and then
- 2) Inclusion-Exclusion – apply inclusion-exclusion to all combinations of these sets.

From [5, section 2.4.2], given the trust graph G and a set T of all minimal operational states from $OP(G)$, the inclusion-exclusion formula for calculating the reliability of coherent systems, $Rel(G)$, is given by the calculation:

$$\sum_{\substack{S \subseteq T, \\ \text{for } k \geq 1.}} (-1)^{k+1} \prod_{e \in T_1 \cup T_2 \cup \dots \cup T_k} p_e. \quad (2)$$

The other general computational approach is based on factoring. Here we consider the reliability of the graph G by looking at two subproblems after selecting an arc e in $E(G)$. The first subproblem is $(G * e)$ where e is assumed always to be operational. The other subproblem is $(G - e)$ where e is assumed to have failed. From [5], the *Factoring Theorem* states that for any reliability measure Rel of a coherent system,

$$Rel(G) = p_e Rel(G * e) + (1 - p_e) Rel(G - e) \quad (3)$$

A. Approximation Heuristic

Given that computing reliability is a #P-complete problem [5], and thus is not practical for larger graphs, a heuristic algorithm is described here to calculate a conservative approximation with practical performance. The *discard-inclusion-exclusion* algorithm modifies both phases of the the generic inclusion-exclusion algorithm. It takes as input three parameters:

DesiredConfidence is the minimum desired metric value, as determined by some user or within some application context,

ProbabilityFloor is the minimum probability of an operational state for it to be considered, and

MaxStates is the maximum number of operational states to be considered in the inclusion-exclusion phase.

The heuristic applies the following four tests to reduce or limit the number of operational states enumerated or considered:

- 1) In the search phase, discard a candidate operational state C if $Prob(C) < ProbabilityFloor$.
- 2) In the search phase, if any operational state S is discovered where $Prob(S) \geq DesiredConfidence$, then the search can be stopped and $Prob(S)$ returned as the approximate metric value.
- 3) After the search phase, if the number of states exceeds *MaxStates* then those where $Prob(S)$ is less or equal the average of $Prob()$ can be discarded until *MaxStates* remain.
- 4) During the inclusion-exclusion phase, given subsets of operational states T are considered in increasing order of the number of states i in T from which they are composed. The result is a lower bound on $Prob(T)$ if i is even. If this lower bound is greater or equal to *DesiredConfidence* return it as the approximate metric value.

The resulting metric value using this heuristic will be less than or equal to the exact value. If any of the heuristic tests are used to discard states or stop the algorithm early then the metric is an a conservative approximation, otherwise it is exact.

B. Experimental Results

A JAVA implementation was created for trust metrics using the GRTM framework defined in Section III. The results presented in this section are based on an implementation of the discard-inclusion-exclusion heuristic for the HLTT metric defined in Section III-A. These results were computed using simulated trust graphs. A *power law random graph* (PLRG) is a randomly generated graph in which the degrees of the vertices follow a power law distribution. The input graphs used were PLRGs. The characteristic equation of a PLRG, and the basis of the PLOD (power law out degree) graph generation algorithm used, is as follows [18]:

$$P(\text{degree}=k) \sim \alpha k^{-\beta} \quad (4)$$

TABLE III
HEURISTIC PARAMETER SETTINGS

Scenario	ProbFloor	MaxStates	DesiredConf
early-discard	0.4	18	0.8 *
soln-reduction	0.1	4	0.8
late-lower-bound	0.1	18	0.8
combination	0.4	8	0.8

For the trust graphs generated the parameter values used were $\alpha=0.7$, $\beta=0.8$, and for every arc e in $E(G)$, l_e (the trust level) was four and c_e (arc confidence) was 0.8. Four sets, or scenarios, of heuristic parameter settings were used to highlight the impact of each of the tests in the heuristic:

early-discard - restrict the enumeration of operational states and do not utilize the lower-bound test in the inclusion-exclusion phase.

soln-reduction - allow more operational states than early-discard, but prune them before the inclusion-exclusion phase.

late-lower-bound - allow more operational states than the other scenarios to focus on the lower-bound test in the inclusion-exclusion phase.

combination - use a combination of the other three scenarios.

The JAVA implementation was run using the scenario parameters shown in Table III with input trust graphs ranging in size from 10 to 5000 vertices. The elapsed time to compute the metric using each of these scenarios is seen in Figure 2.

As seen in the elapsed time, the algorithm demonstrates practical performance in calculating a trust metric for trust graphs of over 1000 vertices. This meets a practical objective of handling graphs composed of hundreds of entities within one second. All of the test scenarios performed well except *late-lower-bound*. This was due to the tendency of the input graphs to support a large number of operational states S with low $Prob(S)$ requiring significant computation in the inclusion-exclusion step.

V. COMPARISON

This section compares the proposed Generic Reliability Trust Model (GRTM) and Hop-count Limited Transitive Trust (HLTT) metric with the trust mechanisms in Table I. An important distinction between GRTM and all the other mechanisms is that GRTM is a trust model by the definition in Section II and thus represents a general framework for specifying a trust metric, while all the others are trust mechanisms with both a model and a metric. They are more directly comparable to the combination of GRTM and HLTT. A more detailed analysis is available in [14].

The general approach used by GRTM of representing trust as a graph and implementing metrics through graph algorithms has been applied before to varying degrees. In MCV, a graph is used to depict the various examples. In the Network Flow mechanism, certificate-based trust is modelled as a graph, with keys as vertices, and two types of certificates as arcs. In this

mechanism, the flow calculation was not used to determine trust, but to determine boolean sufficiency of evidence. Finally, causal graphs associated with the Bayesian Network mechanism are used to model subject-matter dependencies from a single entity's perspective.

GRTM is a more generalized model than those surveyed, with a flexible trust model supporting multiple trust metrics. It is a significant generalization and extension of the MCV model which provided the initial seed. MCV is a more complex metric than HLTT; it has a more complex operational state and lacks a practical algorithm. GRTM is capable of representing any of the COP-type metrics, with the exception of the Trust Management policy evaluation. Unlike the arithmetic metrics presented, GRTM offers a mature underlying reliability model capable of expressing highly nuanced confidence assessments. Finally, Network Flow does not allow for entity-specific assessments of direct trust; rather all evidence is represented in a graph and values for use in the flow calculation are assigned based on distance from some source and target entities. This is a weaker notion of trust based on the number and diversity of entities making a statement as compared to GRTM's computation of source to target confidence based on multiple sources of entity-specific assessments.

VI. CONCLUSIONS AND FUTURE WORK

This paper presents a new computational trust model and metric, GRTM and HLTT. This work contributes theoretical support to this form of trust modelling and the presented algorithms and implementation demonstrate the potential for practical application. These applications of GRTM, with or without HLTT, will likely involve larger network-based communities with ad-hoc, semi-formal interactions. Examples include friend-of-a-friend (FOF) networks, some social-agent systems, or ad-hoc networks. This general application domain can be thought of as the *mushy middle* between valueless interactions on one side, and formal (e.g. contract-based) high-value interactions on the other. A few areas for future research include the following:

- Develop algorithms; apply additional results from Network Reliability research and provide additional characterization of algorithm performance and suitability.
- Extend GRTM to handle multiple subject-matters and distrust.
- Create or utilize some form of standardized representation, exchange or recommendation protocol.
- Integrate GRTM+HLTT with some application.

REFERENCES

- [1] A. Abdul-Rahman and S. Hailes, "A distributed trust model," in *Proceedings of the 1997 Workshop on New Security Paradigms*. ACM Press, 1997, pp. 48–60.
- [2] J. Callas, L. Donnerhacke, H. Finney, and R. Thayer, "Rfc2440: Openpgp message format," IETF Web Site, Internet Engineering Task Force (IETF), Network Working Group, Nov 1998, accessed on June 1, 2004, URL: <http://www.ietf.org/rfc/rfc2440.txt>.

Heuristic Performance

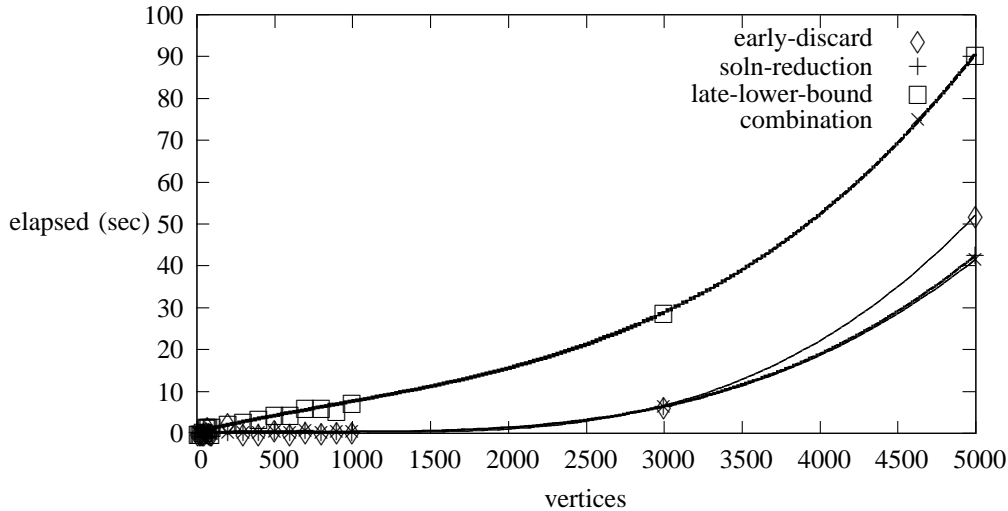


Fig. 2. Time of trust metric calculation using four sets of heuristic parameters.

- [3] L. Camp, H. Nissenbaum, and C. McGrath, "Trust: a collision of paradigms," in *Financial Cryptography. 5th International Conference, FC 2001. Proceedings*, ser. Lecture Notes in Computer Science, P. Syverson, Ed., vol. 2339, Program on Internet and Telecoms Convergence, Center for Technology, Policy, and Industrial Development (CTPID), MIT, Springer-Verlag, Germany, 2002, pp. 91–105.
- [4] B. Christianson and W. Harbison, "Why isn't trust transitive?" in *Security Protocols, International Workshop Proceedings*, ser. Lecture Notes in Computer Science, T. M. A. Lomas, Ed., vol. 1189. Cambridge, United Kingdom, April 10-12, 1996: Springer, 1997, pp. 171–176.
- [5] C. Colbourn, *The Combinatorics of Network Reliability*, J. Hopcroft, Ed. Oxford University Press, 1987.
- [6] W. Diffie and M. Hellman, "New directions in cryptography," *Information Theory, IEEE Transactions on*, vol. 22(6), 1967.
- [7] eBay Inc., "2002 annual report," eBay Web Site, San Jose, California, 2002.
- [8] eBay Inc., "Trust, safety, and privacy," eBay Web Site, San Jose, California, 2004.
- [9] T. Grandison and M. Sloman, "A survey of trust in internet applications," *IEEE Communications Surveys*, vol. 3(4), pp. 2–16, 2000.
- [10] M. Granovetter, "The strength of weak ties," *American Journal of Sociology*, vol. 78(6), pp. 1360–1380, 1973.
- [11] D. Harms, M. Kraetzl, C. Colbourn, and J. Devitt, *Network reliability: experiments with a symbolic algebra environment*. CRC Press, Inc., 1995.
- [12] "Rfc2459: Internet x.509 public key infrastructure certificate and crl profile," IETF Web Site, Internet Engineering Task Force (IETF), PKIX Working Group, January 1999, accessed on June 1, 2004, URL: <http://www.ietf.org/rfc/rfc2459.txt>.
- [13] R. Levien and A. Aiken, "Attack-resistant trust metrics for public key certification," in *Proceedings of the Seventh USENIX Security Symposium*. San Antonio: USENIX Assoc., 1998, pp. 229–241.
- [14] G. Mahoney, "A generalized trust model using network reliability," Master's thesis, University of Victoria, Department of Computer Science, December 2004.
- [15] S. Marsh, "Formalising trust as a computational concept," Ph.D. dissertation, University of Stirling, Department of Computing Science and Mathematics, April 1994, 214p.
- [16] J. L. Matt Blaze, Joan Feigenbaum, "Decentralized trust management," in *1996 IEEE Conference on Privacy and Security*, Oakland, CA, 1996.
- [17] U. Maurer, "Modelling a public-key infrastructure," in *Proceedings of 1996 European Symposium on Research in Computer Security (ESORICS'96)*, E. Bertino, Ed., vol. 1146. Rome: Springer-Verlag, 1996, pp. 325–350.
- [18] C. Palmer and J. Steffan, "Generating network topologies that obey power laws," in *Proceedings of the Global Internet Symposium, IEEE Globecom2000*. San Francisco: IEEE, 2000, pp. 434–438.
- [19] P. Resnick and R. Zeckhauser, "Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system," *Advances in Applied Microeconomics*, vol. 11, pp. 127–157, 2002.
- [20] P. Resnick, R. Zeckhauser, R. Friedman, and K. K., "Reputation systems," *Communications of the ACM*, vol. 43(12), pp. 45–48, 2000.
- [21] D. Shier, *Network reliability and algebraic structures*. Oxford University Press, 1991.
- [22] Y. Wang and J. Vassileva, "Bayesian network-based trust model," in *Proceedings of IEEE/WIC International Conference on Web Intelligence, 2003 (WI 2003)*, Halifax, Canada, October 13-17 2003, pp. 372–378.
- [23] P. Zimmermann, "Why OpenPGPs PKI is better than an X.509 PKI," OpenPGP Web Site, Feb 2001, accessed on June 1, 2004, URL: <http://www.openpgp.org/technical/whybetter.shtml>.