ISR TECHNICAL REPORT 2009-3

Path Optimization Techniques for Trusted Routing in Tactical Mobile Ad-Hoc Networks

Kiran K. Somasundaram, John S. Baras

Path Optimization Techniques for Trusted Routing in Tactical Mobile Ad-Hoc Networks

An interplay between Semirings

Kiran K. Somasundaram, John S. Baras
Institute of Systems Research, University of Maryland, College Park, MD 20742

December 15 2008

1 Introduction

Over the past decade a lot of research effort has been invested in trying to understand security principles for Mobile Ad-Hoc Networks (MANETs). Among the various problems that the research community has focussed, a particularly interesting problem is that of trusted routing in MANETs. Unlike in traditional networks like the Internet, the information flow in a MANET is not restricted to a certain class of routing nodes. The MANETs are different in this sense, because every node acts as a potential relay node. Thus for the proper functioning of any higher layer protocols in MANETs it is of paramount importance that the mobile ad hoc nodes co-operate in routing and forwarding. In many practical scenarios where MANETs are deployed, it might be advantageous for individual nodes not to participate in this forwarding game. Thus one might expect under an unsupervised or unmonitored scenario, the network essentially breaks into components with similar interest groups. This raises the major issue of connectivity in a MANET which is a primitive requirement for the routing layer. Thus securing the network from self-interest groups is inherently coupled with the security of the routing layer. This particular observation has spawned several interesting works in the recent past. Amongst the several ideas developed, a particularly rich literature has developed in the area of reputation inspired routing. The fundamental inspiration for this work is to break the self-interest components by creating incentives for the nodes to co-operate. For an introduction we refer the reader to [14] and [15]. For a more recent survey of the reputation systems in Ad-Hoc Networks refer to [1]. In most of these works in the literature these reputation systems are designed to operate at every node with no fixed infrastructure. Under this strict assumption each MANET node is forced to perform the so called ““*self-policing*‘” [15]. However we propose an alternate means of providing these nodes with a Trust-Support-System which we refer to as the *Sentinel Sub-Network* (SSN). We provide insights on the design and construction of this SSN. We adhere to a bottom-up approach to justify our construction of the SSN architecture. Further in this work we propose several routing schemes/algorithms in the presence of such an SSN. The wide gamut of routing algorithms that we present provides the network administrations in the framework of *Community Networking* a valuable trade-off analysis tool between security and performance.

This paper is organised as follows. In Section 2 we give a brief summary of the reputation systems used in MANETs. In Section 3 we detail the design principles and construction of a Sentinel SubNetwork. In Sections 5 and 6 we provided distributed algorithms that solve the trust routing problem in a multi-criteria setting.

2 Trust/Reputation Systems

In this section we present the key principles and terminology used in the Trust/Reputation literature. The inspiration to use Reputation Systems to aid proper operation of the MANET is detailed in the work [15]. It is based on the fundamental assumption that nodes have a natural incentive to only consume, but not contribute to the services in the system. Further the authors of [15] point out that this misbehaviour could be due to greediness or intent to vandalise of the system. It has been identified by [11] that the goal of reputation systems is to

1. To provide information to distinguish between a trustworthy and untrustworthy principal.

2. To encourage principals to act in a trustworthy manner.

3. To discourage untrustworthy principals from participating in the service the reputation system is present to protect.

[15] uses these goals in the context of MANETs to construct the fundamental modules of a reputation system. The authors propose that a MANET

reputation system should consist of

1. Monitoring and Detection Modules
2. Reputation Control System
3. Response Modules

The goal of monitoring is to observe nodes in the system and detect their deviation from the agreed protocols. The Reputation control system updates the so- called direct-trust and propagates information vectors to construct the indirect-trust metrics. For a detailed exposition on the ontology of these trust terminology refer the reader to [6] and [17]. The response modules should consist of protocols which isolate/penalize aberrant nodes.

The aforementioned principles form the back-bone for several systems proposed in the recent literature on MANET security. Techniques such as Intrusion Detection Systems (Monitoring Mechanisms) [16] have been used to provide information vectors to the reputation control systems. Several Reputation Update/Control systems have also appeared in the MANET security literature such as [2], [9] and [10]. There are also other protocols such as SPROUT [5] which makes use of the reputation systems to route amid colluding attackers. We find that the literature on Reputation Systems is large and do not claim a comprehensive citation of it.

We find most of the reputation mechanisms and response routing are build from the *self-policing* principle. However for tactical MANETs this constraint can be relaxed. In the following section we present the arguments for a having a SSN and give some design principles for building it.

3 Sentinel Sub-Network (SSN)

We claim that to the best of our knowledge this form of trust-support-system is fundamentally different from the ones that are presented in the literature. The basic idea is to logically decouple the Reputation System from other Network functionalities. For tactical MANETs this is valid design principle because we can always have a trusted-core. We argue that if such a trusted-core of even *low capacity* cannot be realized in hostile environments, then design efforts for a still higher capacity reliable secure overall network will not bear any fruits. Thus logically this appears as a subnetwork that lives in the original network. This is show in figure.1. In the forthcoming subsections we state design constraints for building the SSN. We show that the logical constraints on the SSN translate to certain constraints on the communication graph topology of the network.

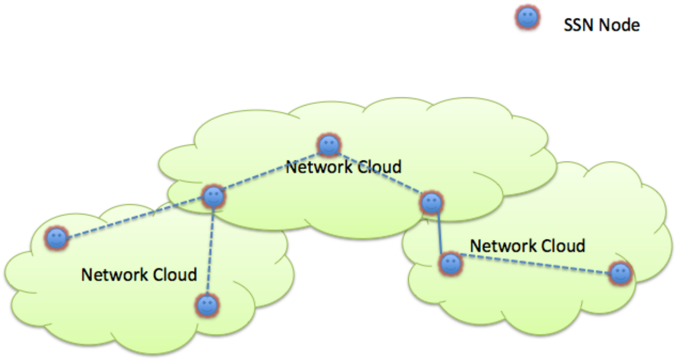


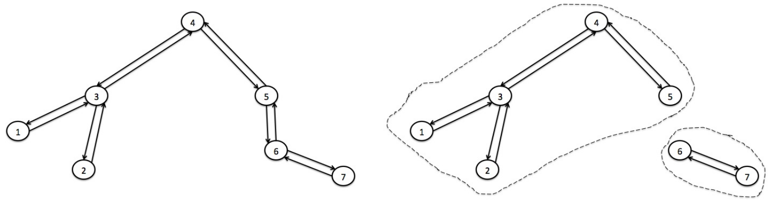
Figure 1: Logical Representation of the SSN

3.1 Modelling Logical SSN

As shown in Fig.2 the Logical Layer of the SSN can be modelled as a Directed Graph $G_{SSN}(V_{SSN}, A)$. Here V_{SSN} is the set of logical layer nodes of the SSN. It is assumed that these nodes have the computation machinery to perform the first two fundamental modules of a trust reputation system explained in section 2. This includes monitoring and the trust update/control systems. Since the goal of the SSN is to decouple the trust functionality from other network functionalities, the response mechanisms such as trust-based routing is carried out by other layers of the network. A denotes the arc set of the logical graph, which captures the trust relations among the SSN nodes. $i.e \quad \forall(i, j) \in A, \quad t_{SSN}(i, j)$ represents that trust that the sentinel node i has on sentinel node j. It should be noted here trust can be asymmetric and hence it is possible for $t_{SSN}(i, j) \neq t_{SSN}(j, i)$. In the following section we discuss the possible graph structure of the logical SSN.

3.2 Graph Topology of Logical SSN

The next fundamental question that we try to address is the topology of G_{SSN}. In a situation when the problem at hand lends itself to a construction of the trusted core (this would be a typical situation in tactical MANETs) it is indeed possible to have a G_{SSN} that is path-connected. However for other practical scenarios, such as in community networking, it might be the case that the G_{SSN} is not path connected and can be broken into self-interest partitions. Both scenarios are shown in 2. These forms of graph topologies induce two distinct reputation systems in the SSN.



(a) Connected Logical SSN (b) Disconnected Logical SSN

Figure 2: Logical Representation of the SSN G_{SSN}

3.3 Local Trust vs Grand Trust SSN

In the case when the G_{SSN} is partitioned, there is no trust relation among the sentinel nodes across partitions. Any trust system on the nodes of such a G_{SSN} has information vectors limited to its partition. We address such a trust system as a Local Trust SSN. It should be mentioned here that within a partition the sentinel nodes can perform message passing to obtain trust estimates of the participating MANET stations. In the case when the G_{SSN} is path-connected, the trust system can obtain estimates of every participating MANET station through message passing. This form of trust system is called Grand Trust SSN. We do not address any particular algorithm for monitoring nodes and performing trust updates. Any of the reputation systems addressed in literature can be used with our framework. We attempt to only answer the question of how the graph topology limits the message passing for the reputation systems. This modelling is in the same lines with the trust modelling of [17]. In this work the semiring algorithms of [17] are however information-limited to the self-interest partitions.

Though the notion of Grand Trust is appealing, it should be mentioned that in most of the practical scenarios it would be expensive to create a Grand Trust SSN. With this observation we proceed to develop response algorithms that can work within the framework of a Local Trust SSN. These algorithms are discussed in Section.5.

3.4 Trust SSN and Network Functionality

In the logical SSN construction so far we have assumed that the reputation mechanism is decoupled from the other network functionalities. However for reliable information transfer in the network, the reputation system must feed information vectors to the network functions. This is logically represented in Figure 3. It shows that the reputation system residing in the logical SSN feeds the routing layer of the network. It should be noted that every router is fed by only one SSN node. This is because we assume that the router modules have no trust computation capabilities. If more than one SSN nodes feeds into a router module, then there is no means by which it can combine the trust estimates. The routing layer would be typically biased by these unambiguous estimates in performing its routing functions. Any ambiguity in the trust estimates should be resolved by the SSN layer.

3.5 Realization the SSN

In this concluding section of SSN construction we attempt to give some insights where the logical SSN layer is to be realized. One means is to have dedicated low power stations to perform the monitoring and trust updates. Another viable option is to install the SSN layer in certain nodes which are assumed to be pre-trusted. Both assumptions are very much valid in the context of tactical MANETs in which the network managers have the ability to create heterogeneous stations. We refer the stations on which the SSN layer is installed as the **Sentinel Stations**. The nature of this construction however creates some constraints on the physical proximity of the sentinel stations. The Sentinel Stations should be chosen in such a manner that every station in the tactical MANET is in the radio range of at least one Sentinel Station. And from the justification in sub-section. 3.4 it is assumed that only one Sentinel Station feeds the MANET station of interest. Such a tactical MANET with Sentinel Nodes is shown in Figure 4

These sentinels may or may not be within the radio range of each other.

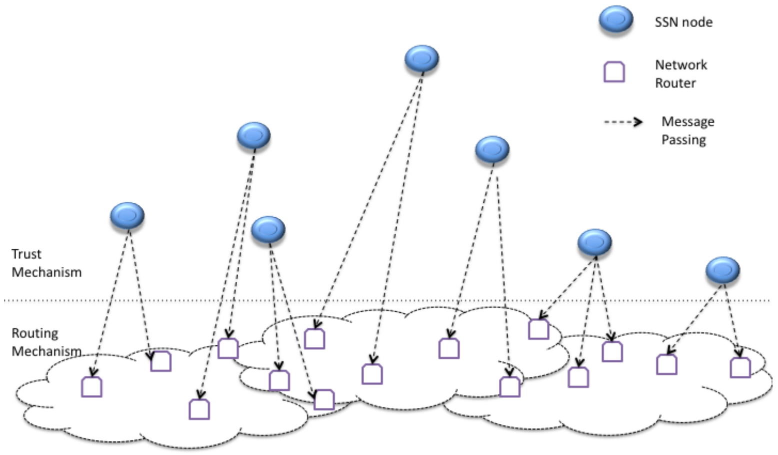


Figure 3: Relationship between Logical SSN and Network Functionality(Routing)

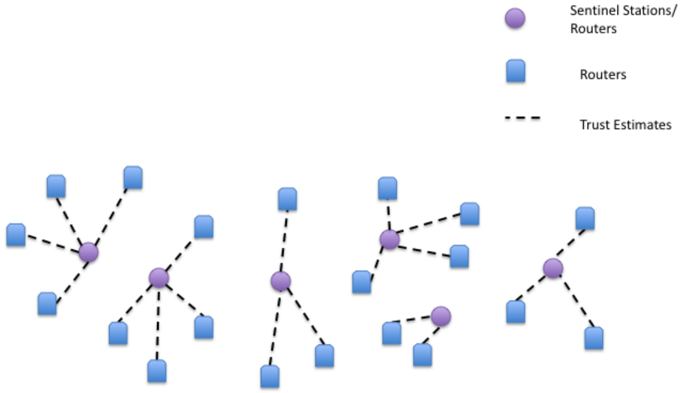


Figure 4: Tactical MANET with SSN

If the sentinel network is partitioned this would correspond the case of Local Trust SSN in the logical SSN framework. Once we have such a realization, the next design question is to develop response routing algorithms that work with certain metrics on performance and security . In the forthcoming sections we develop these metrics for the response algorithms.

4 Trusted Routing as an Multi-Objective Optimization Problem

In this problem formulation the SSN monitors behavior of the stations (in the presence of greedy/malicious stations) or the quality of the links (in the presence of jammers) and updates trust values. Further the SSN assumed to work with local trust, avoiding the usual assumption of a globally trusted principal. This is consistent with the premise that the logical SSN in most cases is partitioned into self-interest groups. We propose algorithms to solve the trusted routing problem distributively with these local trust estimates. Further we require the Sentinel Network to disseminate the trust information only in the local neighbourhood. By this, we mean that every station has access only to the trust values of its neighbouring stations or adjacent links. Such a condition makes the construction of the sentinel network very inexpensive. Under these conditions, it is possible for the sentinel network to perform local monitoring and distributed trust evaluation. In most cases, this is more of a necessity because the sentinel networks are usually low capacity networks.

Routing Decisions

If trust evaluations from the SSN is available to the routing layer, the routing decisions can be biased by this metric. The routing decisions essentially correspond to selecting paths that satisfy certain performance metric and trust metric. A reasonable performance criterion is the packet delay along a path. When there are multiple source-destination pairs, the delays are dependent on the routes chosen by each of these pairs.

However for the trust metric of the path, there is no universally-defined notion referred to in the current research literature. In light of this problem, we attempts to define a reasonable metric for the trust of a path which adheres to a generic rationality.

Once such a trust metric is defined, we find that the routing problem can be treated as a Multi-criteria Optimization Problem on a graph. The trust metric could be posed as *Hard Constraints* or could be *Soft-Coupled* into the cost function of the optimization problems. A generic version that includes both is explained in the forthcoming sections.

5 Distributed Multi-criteria Routing

In this section we develop mathematical expressions for the performance and trust metrics of a path. We observe that two semiring structures naturally arise from the definition of our metrics. We show that these two semirings can be effectively combined in a multi-criteria formulation lending itself to the required distributed solution.

5.1 Mathematical Formulation

The given Ad-Hoc Network is modeled as Communication Graph $G(V, E)$, where V is the set of vertices representing the stations and E is the edge set which connects those stations/vertices that are within radio range of each other. The radio communication is essentially symmetric and hence G is an undirected graph. Let $\mathcal{P}_{SD}$ denote the set of paths in G from source S to destination $D \neq S$. Then $\mathcal{P} = \cup_{S \neq D} \mathcal{P}_{SD}$. Let $\mathcal{N}(i)$ denote the neighbours of i (*i.e.* those nodes that are in the radio range of i).

5.2 Modeling Node Trust

In models with greedy relay nodes, it is assumed that the sentinel network assigns a trust score to every node the in the network. The trust scores are then securely flooded to all the neighbours of interest. This trust score is represented as $t(i) \quad \forall i \in V$. Further it is assumed that $t(i)$ is flooded to only $\mathcal{N}(i)$.

5.3 Modeling Edge Trust

In models with jammers, it is assumed that the sentinel network assigns a trust score to every link/edge indicating its susceptibility to jammers. This trust score is represented as $t(i, j) \quad \forall (i, j) \in E$. Again each $t(i, j)$ is flooded to only $\mathcal{N}(i)$ and $\mathcal{N}(j)$. It should be noted here that $t(i, j) \neq t(j, i)$ in general. The trust relations on the links form a directed graph with asymmetric trust weights on the links.
The algorithms described in the forthcoming sections work seamlessly with

both Node Trust (symmetric) and Edge Trust (asymmetric) cases. For simplicity of exposition, only the Node Trust case is explained in detail. However in places where there is a distinction between the two problems, we mention that difference.

5.4 Trust of a Path

In this section we define a trust metric for a path. Each path is an ordered sequence of station identifiers or alternatively an ordered sequence of links. The trust metric of a path should be defined as a composition of the trust values defined on the these nodes/edges. For the sake of explanation consider a trust score for each station which ranges from 0 to 1. Higher trust values correspond to better behaved stations. Figure 5 shows a typical path $(i_1, i_2, i_3, i_4, i_5, i_6, i_7)$. There are certain trust scores specified for each station along the path. In particular the relaying station i_5 has a trust value of 0.3, which is to be considered pretty low for the sake of this example.

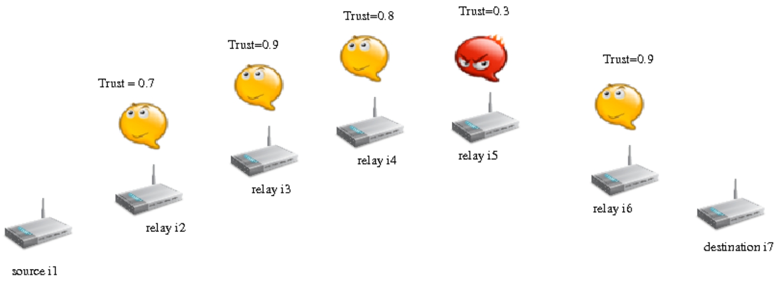


Figure 5: A typical path

In defining a trust metric, its reasonable to adhere to the adage that the strength of a path(chain) is limited by the strength of the weakest link in the path. The rationale behind this assumption is that the trust value of a path cannot be greater than trust value of any of the stations along the path. Thus the trust value of a path is lesser than or equal to the trust value of the stations/links along the path.

$$\begin{aligned} & Trust(path) \leq Trust(i) \quad \forall i \in (path) \\ \Rightarrow \qquad & Trust(path) \leq \min_{i \in (path)} Trust(i) \end{aligned}$$

In the context of node trust, it is defined as follows

$$\forall p \in \mathcal{P}, \quad t(p) \leq \min_{i \in p} t(i)$$

and similarly for edge trust, it is defined as

$$\forall p \in \mathcal{P}, \quad t(p) \leq \min_{(i,j)\in p} t(i,j) \tag{1}$$

Throughout the formulation we assume that trust values are non-negative.

5.5 Average Delay along a Path

One of the reasonable performance metric to choose is the congestion delay along a path. Other metrics such as the hop count can also be used in place of the delay metric in this formulation. Our algorithms are independent to the exact metric chosen. The only structure that we assume is that the congestion metric should have an additive property along the path. It should be mentioned here that this gives rise to one of the semirings in our optimization algorithms.

In typical multi-hop wireless networks, the delay that each packet suffers is primarily due to the queue buildup at the MAC layer. Let us suppose the average delay for the queue at node i is given by $d(i) \quad \forall i \in V$. The delay of a path is then given by

$$d(p) = \sum_{i\in p} d(i) \quad \forall p \in \mathcal{P} \tag{2}$$

Again in the light of having a distributed algorithm, it is assumed that the delay information is exchanged only in the local neighbourhood. This exchange can be monitored and attested by the sentinel network to possibly avoid malicious information dissemination.

5.6 Route Selection - A multi-objective Optimization

In the framework, the routing logic is to choose paths in the graph G which have high trust values and low delay values. This work confines to the effort of selecting a single path as a route profile for given source-destination pair of traffic. For a given path $p \in P_{SD}$, the delay and trust objective are denoted by $d(p)$ and $t(p)$ respectively. While the delays follow the natural order of the reals, the order on trust can be abstract. However for this solution, it is assumed that there is a model map, that transforms the delay space and abstract ordered space of trusts to $\mathcal{R}^q \quad q \leq 2$. For the example shown in Figure 5, the *model map* is just the identity map. It should be noted however, as long as a meaningful order is defined on the delay-trust product space, it is very possible to use our algorithms in the forth-coming sections on that abstract order.

Notation	Definition	Name
$x \leq y$	$x_i \leq y_i \quad i = 1, 2, .., Q$	Weak component-wise order
$x < y$	$x_i \leq y_i \quad i = 1, 2, .., Q$ and $x \neq y$	Component-wise order
$x \ll y$	$x_i < y_i \quad i = 1, 2, .., Q$	Strict component-wise order
$x \leq_{lex} y$	$x_k < y_k \quad or \quad x = y \quad k = \min\{i : x_i \neq y_i\}$	Lexicographic component-wise order
$x \leq_{MO} y$	$\max_i x_i \leq \max_i y_i$	Max order

Table 1: Table of Orders in $\mathcal{R}^Q$

The route controller then essentially tries to solve the **Multi-criteria Optimization Problem (MCOP)** of the following class

$$(P_{SD}, f, X)/\theta/(\mathcal{R}^Q, \preceq) \qquad Q \leq 2$$

where P_{SD} is the set of feasible paths, f is a vector valued objective function and X is abstract delay-trust product objective-space. $\theta : X \rightarrow \mathcal{R}^Q$ is the model map that transforms the objective-space into $\mathcal{R}^Q$. $\preceq$ is the any order on $\mathcal{R}^Q$. We define various model maps and order relations, where the MCOP can be solved distributively. The orders that would be considered are tabulated in Table.1 for vectors x and $y \in \mathcal{R}^Q$.

For detailed exposition of Multi-criteria Optimization terminology we refer the reader can refer to [4]. To justify the model maps that our work considers, a brief introduction to semirings is presented in the next subsection.

5.7 Semirings - The Delay and Trust Semirings

We present a discussion of the semiring structure which is applicable to the trusted routing problem. For a detailed survey of the applications of semirings we refer the reader to [13]. For specific applications we also suggest the works of [8], [7] and [18].

A semiring is an algebraic structure $(S, \oplus, \otimes)$ which satisfies the following axioms

(A1) (S,$\oplus$) is a commutative semigroup with a neutral element $\textcircled{0}$

$$\begin{aligned} a \oplus b &= b \oplus a \\ a \oplus (b \oplus c) &= (a \oplus b) \oplus c \\ a \oplus \textcircled{0} &= a \end{aligned}$$

(A2) (S,$\otimes$) is a semigroup with a neutral element $\textcircled{1}$ *and* $\textcircled{0}$ *as an absorbing element*

$$\begin{aligned} a \otimes (b \otimes c) &= (a \otimes b) \otimes c \\ a \otimes ① &= a \\ a \otimes ⓪ &= ⓪ \end{aligned}$$

(A3) $\otimes$ *distributes over* $\oplus$

$$\begin{aligned} a \otimes (b \oplus c) &= (a \otimes b) \oplus (a \otimes c) \\ (a \oplus b) \otimes c &= (a \otimes c) \oplus (b \otimes c) \end{aligned}$$

It should be noted that the functions which have this semiring structure lend themselves to distributed computation/evaluation by the virtue of the distribution property(A3) which henceforth will be referred to as **Semiring Distribution**. Of these semiring structures, the one that is particularly useful for optimization is the **Ordered Semiring**. Here the $\oplus$ is the supremum or infimum operator and $(S, \otimes, \preceq)$ is an ordered semigroup. An ordered semigroup is a semigroup with an order relation which is monotone with respect to $\otimes$.

$$a \preceq b \quad and \quad a' \preceq b' \quad \Rightarrow \quad a \otimes a' \preceq b \otimes b'$$

For the problem at hand, it is natural to deal with a graph whose vertices or edges are labelled with elements from S, which can be considered to be some form of weight/cost. These can corresponds to the delay on the nodes or the trust values on the nodes/edges. In both cases, the optimization problem would attempt to find the path, which has the optimal (smallest or largest) aggregate weight/cost in a given semiring.

The delay optimization Eqn.2 corresponds to the $(\mathcal{R}_+ \cup \{0\}, \min, +)$ semiring, which is called the **Delay Semiring**. Correspondingly the **Trust Semiring** is $(-\mathcal{R}_+ \cup 0, \min, \max)$ semiring. This is because the Eqn.1 suggests that the trust optimization should be

$$\begin{aligned} & \max_{p \in \mathcal{P}_{SD}} t(p) \\ \Rightarrow \quad & \max_{p \in \mathcal{P}_{SD}} \min_{i \in p} t(i) \\ \Rightarrow \quad & \min_{p \in \mathcal{P}_{SD}} \max_{i \in p} -t(i) \end{aligned}$$

It is clear that the two objectives delay and trust are different semirings with a common min operator. The following sections suggest ways in which these two semirings can be combined in the spirit of **Distributed multi-criteria optimization**. Notions of Pareto Optimality, Lexicographic Optimality, Max-Ordering and Approximation Semirings will be considered.

6 Distributed Multi-criteria Optimization Algorithms

Given the delay and trust semiring, let us define the objective function $f : \mathcal{P}_{SD} \rightarrow R^2$

$$f(p) = (d(p), -t(p)) \quad \forall p \in \mathcal{P}_{SD}$$

With this bi-objective function, we define various model maps and develop distributed algorithms to solve the multi-objective optimization problem.

The rest of this section is organized as follows. In subsection 6.1 we detail the Pareto optimal trusted routing problem and provide a distributed algorithm to obtain all the Pareto paths. In subsection 6.2 we present the solution to the lexicographic ordering of the trusted routing problem. In subsections 6.4 and 6.5 we develop interesting scalarized version of the multi-objective problem. In all the above cases we present distributed algorithms to solve the corresponding problems.

6.1 Pareto Optimal Routing

Consider the Pareto Class $(\mathcal{P}_{\mathcal{SD}}, f, R^2/id/R^2, <)$, where $<$ is the component-wise order defined in Table.1 and the model map is the identity map.
Pareto Optimal Path : A path $p^* \in \mathcal{P}_{SD}$ is Pareto optimal $\exists$ no path $p \in \mathcal{P}_{SD} \neq p*$ such that $f(p) < f(p^*)$.
To compute the Pareto Optimal Paths, we develop an extended version of *Haimes - ϵ constraint* method. For material on this method refer to [19] and [3]. The basic idea of this method is to convert all but one of the objectives into constraints and solve the constrained single-objective problem for various constraints. This constrained optimization problem might have more than one global minimizer. Hence from this solution set, the path which is optimal with respect to other objective is chosen. Thus our version of the corresponds to solving the following two subproblems for $\epsilon \leq 0$.
SUBPROBLEM 1(ϵ)

$$\begin{array}{lll} \min\limits_{p \in P_{SD}} & \sum\limits_{i \in p} d(i) & ...(SubPb_1(\epsilon)) \\ \textit{such that} & \max\limits_{i \in p} -t(i) \leq \epsilon & \end{array}$$

SUBPROBLEM 2(ϵ)
If $P^*(\epsilon)$ denote the set of optimal paths for problem $SubPb_1(\epsilon)$.

$$\begin{aligned} p^* &= arg \max_{p \in P^*(\epsilon)} t(p) \quad (\cdots SubPb_2(\epsilon)) \\ &= arg \max_{p \in P^*(\epsilon)} \min_{i \in p} t(i) \end{aligned}$$

The problems $SubPb_1(\epsilon) \quad and \quad SubPb_2(\epsilon) \quad \forall \epsilon \leq 0$ obtains all the Pareto Optimal paths for a given graph. The proof of Pareto optimality of our algorithm is Appendix.A

Hard Constrained Routing / Path Exclusion

We observe that an interesting routing principle arises from the Pareto Optimal Routing Algorithms. The subproblems $SubPb_1(\epsilon)$ and $SubPb_2(\epsilon)$ give rise to paths which we refer to as the **Trust-Hard Constraint Paths**. This is a useful routing tool, because it optimizes over over a constraint that satisfies *"All feasible paths should have a trust value of above $-\epsilon$"*. This is exactly the situation the network operator might be interested in if she wants certain security requirements on the routing paths. In other words, the network operator might tend to exclude paths which have very poor trust values.

Distributed Solution

This section proposes an algorithm to solve SUBPROBLEMS 1 and 2, using the Semiring Distribution and Distributed Exclusion, which needs message passing only among neighbour nodes. That is every node $i \in V$ passes messages to only $\mathcal{N}(i)$. The assumption is that the delay value of the optimal path to destination D from any node i is securely exchanged between the neighbour nodes. The corresponding trust values of the paths are also exchanged.

There is a *Subtle Rationality* that should be mentioned here. The aggregate delay at node i includes the delay d_i at the node. However the aggregate trust cannot include t_i. This makes sense because, from every node's perspective it would have maximal trust on itself. Thus the trust aggregate accounts for the trust up to its neighbours only. This rationality is shown in Figure.6. It should be noted here that this exchange about aggregate trust value can be used as an input vector for the sentinel network's trust mechanism.

Distributed Extended Haimes Algorithm

From a computational perspective, it would be infeasible to solve the problems $SubPb_1(\epsilon) \quad SubPb_2(\epsilon) \forall \epsilon \leq 0$. However since this path optimization problem works with finitely many paths and we can resort to numerical means which adhere to the commensurability. Such approaches for distributed flow problems on graphs is discussed in detail in [12]. The basic premise here is that the finite set of numerals in the problem should be commensurable. This means, all the numerals in the given problem should have a structure wherein, they can be expressed as integral multiple of a finite quanta. We make is this assumption for most of the algorithms in this work.

We present the *Extended Haimes* procedure. In the first phase of the procedure, we develop a distributed algorithm to compute the trust-quanta. This algorithm is carried out at each node $i \in V$. There are two versions of this algorithm for Edge-Trust and Node-Trust respectively.

Algorithm 1 Edge-Trust - Compute Trust-Quanta δ^t

```
INITIALIZE:
$\delta_i^t(0) \leftarrow \max\{\delta \quad |\forall j \in \mathcal{N}(i), \quad t(i,j) = k_j\delta \texttt{ where } k_j's \texttt{ are integers}\}$
repeat
  $\delta_i^t(n+1) \leftarrow \max\{\delta \; |\forall j \in \mathcal{N}(i) \cup i, \; \delta_j^t(n) = k_j\delta \texttt{ where } k_j's \texttt{ are integers}\}$
until $\delta_i^t(n)$ converges
```

Algorithm 2 Node-Trust - Compute Trust-Quanta δ^t

```
INITIALIZE:
$\delta_i^t(0) \leftarrow \max\{\delta \quad |\forall j \in \mathcal{N}(i), \quad t(j) = k_j\delta \texttt{ where } k_j's \texttt{ are integers}\}$
repeat
  $\delta_i^t(n+1) \leftarrow \max\{\delta \; |\forall j \in \mathcal{N}(i) \cup i, \; \delta_j^t(n) = k_j\delta \texttt{ where } k_j's \texttt{ are integers}\}$
until $\delta_i^t(n)$ converges
```

When the algorithms Alg.1 and Alg.2 converge at every node, they converge to a common value $\delta_i^t = \delta^t \quad \forall i \in V$. The proof for this convergence is given in Appendix.C. This value δ^t is used to bootstrap the following algorithms. The constraint value ϵ is iteratively decremented as $\epsilon \leftarrow \epsilon - \delta^t$ starting from $\epsilon \leftarrow 0$.

Once the subgraph $G'(\epsilon)$ is obtained for either the node or edge trust

Algorithm 3 Node Trust-Graph Reduction: Procedure to form Reduced graph $G'(\epsilon)$ $\epsilon \leq 0$

```
for $j \in \mathcal{N}(i)$ do
  if $t(j) < -\epsilon$ then
    Node $j$ and its incident edges are excluded
  end if
end for
```

Algorithm 4 Edge Trust-Graph Reduction: Procedure to form Reduced graph $G'(\epsilon)$ $\epsilon \leq 0$

```
for $j \in \mathcal{N}(i)$ do
  if $t(i,j) < -\epsilon$ then
    (i,j) is excluded
  end if
end for
```

problem, the algorithms Alg.5 and Alg.6 are performed sequentially.

Algorithm 5 Delay-Bellman-Ford to reach destination D on $G'(\epsilon)$

```
repeat
  $SP_i^{n+1}(D) = d(i) + \min_{k \in \mathcal{N}(i)} SP_k^n(D)$
until $SP_i^n(D)$ converges
```

where $SP_i^n(D)$ is the minimum delay to reach destination D in n steps. This algorithm outputs the path set $P^*(\epsilon)$.

Algorithm 6 Extract Pareto Paths from $P*(\epsilon)$ on $G'(\epsilon)$

```
.
$p^* = arg \max_{p \in P*(\epsilon)} t(p)$
```

It is easy to check that the set $P^*(\epsilon)$ is polynomially bounded in non-trivial instances of the problem. Hence any optimization algorithm will be polynomially bounded for Alg.6.

Algorithms *Graph Reduction* and *Delay-Bellman-Ford* are carried out $\forall\epsilon \leq 0$. This corresponds to solving SUBPROBLEM 1 ($SubPb_1(\epsilon)$). The last algorithm of Pareto-Path-Extraction solves SUBPROBLEM 2($SubPb_2(\epsilon)$). These three algorithms when carried out $\forall\epsilon \leq 0$, yield all the Pareto Paths. The proof that these three algorithms solve $SubPb_1(\epsilon)$ and $SubPb_2(\epsilon)$ is shown in Appendix.B

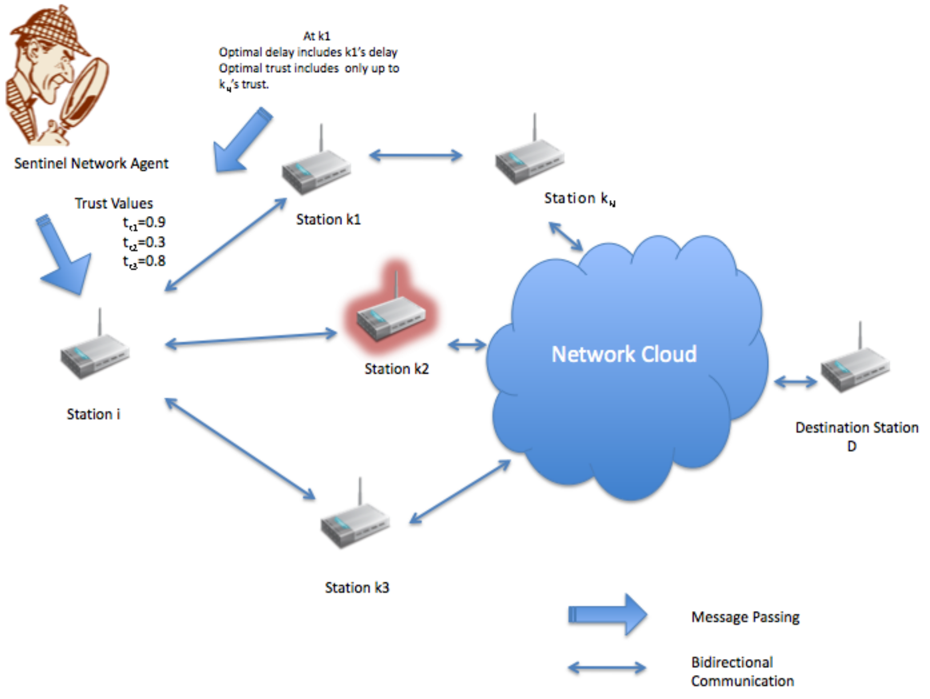


Figure 6: Subtle Rationality

6.2 Lexicographic Optimal / Biased Routing

The obvious shortcoming of using Pareto Optimality is that the number of paths optimal in the Pareto sense is large. There is still an issue of selecting certain paths that might be desirable in some sense. One such popular approach is Lexicographic Ordering. This method assumes that one metric is superior to other and tries to optimize with respect to the superior metric. Only if two or more feasible solutions are equally optimal in the superior metric, the other metric is considered. For the problem of trust routing, the superior bias can be assigned to either the trust metric or the delay metric. Based on this, the $\leq_{lex}$ defined in Table.1 is correspondingly modified. Mathematically this MCOP class can be represented as $(\mathcal{P}_{SD}, f, X/id/R^2_+, \leq_{lex})$. It should be noted that a Lexicographic optimal path is a Pareto optimal path (Chapter 6 of [4]).

Delay Biased Routing

As the name suggests, the superior criterion is delay. The problem can also be solved distributively using a delay semiring $(R^2_+, \oplus_d, \otimes)$. The semiring operations are defined as follows. Let $(d1, t1)$ and $(d2, t2)$ be two sets of delay-trust pairs. Then

$$\begin{aligned} (d1,t1) \oplus_d (d2,t2) &= \begin{cases} (d1,t1) & if \quad d1 < d2 \\ (d2,t2) & if \quad d2 < d1 \\ (d1, max(t1,t2) & if \quad d1 = d2 \end{cases} \\ (d1,t1) \otimes (d2,t2) &= (d1+d2, max(t1,t2)) \end{aligned}$$

It is trivial to check that indeed the operations form a semiring in R^2_+. This form of routing can be used in a networks where the routers are willing to sacrifice security for performance.

Trust Biased Routing

In this form of routing, the trust metric is considered the superior criterion. To solve this problem the trust semiring $(R^2_+, \oplus_t, \otimes)$ is defined.

$$\begin{aligned} (d1,t1) \oplus_t (d2,t2) &= \begin{cases} (d1,t1) & if \quad t1 > t2 \\ (d2,t2) & if \quad t2 > t1 \\ (min(d1,d2),t1) & if \quad t1 = t2 \end{cases} \\ (d1,t1) \otimes (d2,t2) &= (d1+d2, max(t1,t2)) \end{aligned}$$

Again its easy to check that these operators constitute a semiring in R^2_+. Analogous to the previous case this routing sacrifices performance to security.

Distributed Solution

To solve biased routing problem at every node $i \in V$, the following algorithm is carried out. As in the previous algorithms there is only bounded message passing in the local neighbourhood. At each node, there are semiring elements $a_{ik} = (d_i, t_k), \quad \forall k \in \mathcal{N}(i)$ which are in the delay-trust pair. Every node i is assumed to have access to semiring elements $X^n_k(D) \quad \forall k \in \mathcal{N}(i)$, which represents the (delay-trust) metric for the optimal paths to reach destination D in n steps. As explained in Section.6.1 the information is exchanged securely with the aid of the sentinel network under the *subtle rationality* mentioned.

Algorithm 7 *A*2.Procedure Lexicographic Optimal Path to reach destination D

```
repeat
    $X_i^{n+1}(D) = \bigoplus_{k \in \mathcal{N}(i)} a_{ik} \otimes X_k^n(D)$
until $X_i^n(D)$ converges
```

where $\bigoplus = \oplus_t$ or $\bigoplus = \oplus_d$ for trust and delay biased routing respectively. This semiring iteration converges under the conditions given [13]. The proof that the algorithm converges to the lexicographic optimal path is given in Appendix.D.

6.3 Scalarization Methods

Among the several methods to obtain the Pareto Points, scalarization is another popular method to solve for the Pareto Optimal solutions for Convex problems. For path problems, the objective space is discrete and scalarization in most cases does not yield all Pareto solutions. Nevertheless, the scalarization techniques yield paths which are desired in another sense. The first scalarization is called Max-Order Optimality. The second scalarization is the standard convex combination of the objective functions. For the latter case, to develop a distributed solution using semiring distribution, the weighted scalarized cost function is suitably approximated. The two scalarization methods are described in the follow sections

6.4 Max-Order Optimality / Conservative Routing

In order to bring the delay-trust optimization problem in the common spirit of a minimization problem consider the transformed function $t_N(p) = C_t - t(p) \forall p \in \mathcal{P}_{sd}$. The constant C_t indicates the *Relative Importance* of the trust metric in max-ordering problem. The max-ordering problem corresponds to the class $(\mathcal{P}_{\mathcal{SD}}, f, R_+^2/max/R_+, <)$, where $<$ is the natural order in the reals. Thus the max-ordering problem corresponds to

$$\min_{p \in \mathcal{P}_{SD}} max(d(p), t_N(p))$$

This problem tries to select paths which are optimal in the worst-case sense of trust and delay. Thus it is a conservative means of routing wherein the cost of the path is governed by the worst-case value of its trust and delay. From an optimization point of view, this appears as a simplification of the cost function. However it should be noted that there is no evident reduction of the complexity of the path problem. The following algorithms distributively solves the conservative routing problem. Each of these algorithms run at each of the nodes i and needs only local information for its computation.

Conservative Routing Algorithm

As in the *Distributed Extended Haimes Algorithm* we invoke the commensurabilty techniques to compute a feasible step size. In this case we use the commensurabilty of both the trust and delay values.

Algorithm 8 Edge-Trust - Compute Delay-Trust-Quanta δ

```
INITIALIZE: $\delta_i(0) \leftarrow \max\{\delta \ |\forall j \in \mathcal{N}(i),\ t(i,j) = k_j\delta \wedge d(i) = l\delta$ where $l, k_j's$ are integers$\}$
repeat
  $\delta_i(n+1) \leftarrow \max\{\delta \ |\forall j \in \mathcal{N}(i) \cup i,\ \delta_j(n) = k_j\delta$ where $k_j's$ are integers$\}$
until $\delta_i(n)$ converges
```

Algorithm 9 Node-Trust - Compute Delay-Trust-Quanta δ

```
INITIALIZE: $\delta_i(0) \leftarrow \max\{\delta \ |\forall j \in \mathcal{N}(i),\ t(j) = k_j\delta \wedge d(i) = l\delta$ where $l, k_j's$ are integers$\}$
repeat
  $\delta_i(n+1) \leftarrow \max\{\delta \ |\forall j \in \mathcal{N}(i) \cup i,\ \delta_j(n) = k_j\delta$ where $k_js$ are integers$\}$
until $\delta_i(n)$ converges
```

Once δ converges on every node then ϵ is incremented as $\epsilon \leftarrow \epsilon + \delta$ starting from $\epsilon = 0$.

Algorithm 10 Edge Trust: Procedure Graph Reduction $G'(\epsilon)$

```
for $j \in \mathcal{N}(i)$ do
  if $t(i,j) < C_t - \epsilon$ then
    (i,j) is excluded
  end if
end for
```

Algorithm 11 Node Trust: Procedure Graph Reduction $G'(\epsilon)$

```
for $j \in \mathcal{N}(i)$ do
  if $t(j) < C_t - \epsilon$ then
    Node $j$ and its incident edges are excluded
  end if
end for
```

The Alg.12 assumes that in a reduced graph $G'(\epsilon)$ the optimal delay path is computed using an Shortest path algorithm, say Bellman-Ford. This optimal path is denoted as $p' * (\epsilon)$. If there exists no path from S to D, the delay of the path is assumed to ∞

Algorithm 12 Procedure Max-Ordering Optimal Path to reach Destination D

```
$\epsilon \leftarrow 0$
repeat
  $\epsilon \leftarrow \epsilon + \delta$
  Construct Sub-graph $G'(\epsilon)$ by calling Graph-Reduction Procedure
until $d(p'*) \leq \epsilon$
Output $p'*$
```

The validity of the algorithms and the proofs of convergence are discussed in Appendix.E.

6.5 Scalarization with Weighted Sums / Approximation Semiring Relative Importance Routing

The weighted sums method to compute the Pareto efficient solutions is a common tool used for convex problems. However even for the discrete case of path problems, the weighted sum scalarization can yield some Pareto efficient paths. In path problems, the weighted scalarization computes the so called *Supported Pareto Paths*. The notion of supported efficient set is defined in [4]. The advantage of this method being, that the weighted cost function is the algorithm the flexibility of assigning importance to one metric over the other. The scalarization problem turns out to be

$$\min_{p \in \mathcal{P}_{SD}} d(p) - \mu t(p)$$

$$\min_{p \in \mathcal{P}_{SD}} \sum_{i \in p} d(i) - \mu \min_{i \in p} t(i)$$

Here the parameter μ is the *Relative Importance* factor for the trust metric. Larger values of μ bias the optimization problem to solve for paths which are relatively more optimal in the trust sense. It should be noted that cost function is not solvable by Semiring Distribution. In order to make it Semiring Solvable, the cost function is modified as follows

$$\min_{p \in \mathcal{P}_{SD}} \sum_{i \in p} d(i) - \mu \sum_{i \in p} e^{-Mt(i)}$$
$$\min_{p \in \mathcal{P}_{SD}} \sum_{i \in p} (d(i) - \mu e^{-Mt(i)})$$

This approximation is inspired from the large M behaviour of the exponentials.

$$\sum_i e^{-M*t(i)} \simeq e^{-M* \min_i (t(i))}$$

It should be also mentioned that the approximation can also be treated as the exponential penalty for the trust a path. This form of penalty function penalises those paths which have edges whose trust values are low. This alternate interpretation gives the designer the flexibility to design other distributed penalty functions that follow the trust rationale.

The approximation problem is semiring solvable under the $(R_+, min, +)$ semiring and converges under the conditions stated in [13].

7 Appendix

A Pareto Optimality of Extended Haimes Algorithm

Proposition A.1 *$SubPb_1(\epsilon)$ and $SubPb_2(\epsilon) \forall \epsilon \leq 0$ essentially solves for the Pareto Efficient paths.*

Proof Proof By Contraction.
Let p^* be the optimal solution obtained from $SubPb_1(\epsilon)$ and $SubPb_2(\epsilon) \quad \forall \epsilon \leq 0$. If $p*$ is not Pareto Optimal $\exists\ p \in \mathcal{P}_{SD}$ such that $f(p) < f(p^*)$. Since the order is component-wise order, this can correspond to two cases.
Case I . $d(p) < d(p*)$ and $-t(p) \leq -t(p^*)$. But $d(p^*)$ being optimal for

$SubPb_1(\epsilon) \forall \epsilon \leq 0$, this is not possible.
Case *II*. $d(p) = d(p^*)$ and $-t(p) < -t(p^*)$. But $t(p^*)$ being optimal for $SubPb_2(\epsilon) \forall \epsilon \leq 0$, this is not possible.
This contradicts the assumption and hence p^* is indeed Pareto Optimal.

B Algorithms that solve the Delay-Trust(ϵ) problem

This section shows how Graph-Reduction & Delay-Bellman-Ford algorithms sequentially solve $SubPb_1(\epsilon)$ and $SubPb_2(\epsilon)$. The proof explains the claimed *Interplay between the two semirings.* The proof is based on the observation is that the (R, max, min) semiring is closely related to a distributed edge exclusion problem. This observation is explained in the following sub-section.

B.1 Distributed Edge Exclusion - MaxMin Semiring

In problem $SubPb_1(\epsilon)$ the constrained of paths satisfy

$$-t(p) \leq \epsilon \quad \forall p \in \mathcal{P}_{SD}$$

Theorem B.1 *The above set of paths corresponds to the paths in the reduced graph $G'(\epsilon)$.*
$\mathcal{P}^c_{SD} = \mathcal{P}'_{SD}$

For the proof we consider the problem of Edge Trust. It should be noted that the proof for Node Trust follows trivially. The constraint then appears as

$$\begin{aligned} & \max_{(i,j)\in p} -t(i,j) \leq \epsilon \\ \Rightarrow \quad & \min_{(i,j)\in p} t(i,j) \geq -\epsilon \end{aligned}$$

this essentially eliminates all paths p which have an edge $(i,j) \in p$ such that $t(i,j) < -\epsilon$. Let $\mathcal{P}^c_{SD}$ denotes those paths which satisfy the constraint. *i.e* those paths p for which all the edges $(i,j) \in p$ satisfy $t(i,j) \geq -\epsilon$. Now consider a subgraph $G'(V, E')$ formed by eliminating all edges $(i,j) \in E$ whose $t(i,j) < -\epsilon$. The reduced edge-set is denoted by E'. Let $\mathcal{P}'_{SD}$ denote the set of all paths in G' from source S to destination D

Proof $\forall p \in \mathcal{P}'_{SD}$ the edges in $p \quad (i,j) \in E' \subseteq E$ and $t(i,j) \geq -\epsilon$.
$\Rightarrow p \in \mathcal{P}^c_{SD}$ (from the definition of P^c_{SD}).
$\Rightarrow \mathcal{P}'_{SD} \subseteq \mathcal{P}^c_{SD}$

Suppose $\mathcal{P}^c_{SD} \not\subseteq \mathcal{P}'_{SD}, \quad \exists p \in \mathcal{P}^c_{SD}$ such that $p \notin \mathcal{P}'_{SD}$.
Suppose $p = ((i_1, i_2), (i_2, i_3), ..., (i_{n-1}, i_n))$ is a sequence of edges in E. Each of these edges satisfy $t(t_k, t_{k+1}) \geq -\epsilon$. By definition, these edges would be contained in E'_{SD}. Thus the sequence of edges and the hence the path $p \in \mathcal{P}'_{SD}$.
$\Rightarrow \mathcal{P}^c_{SD} \subseteq \mathcal{P}'_{SD}$

$\Rightarrow \mathcal{P}^c_{SD} = \mathcal{P}'_{SD}$

In a similar manner for node trust, the paths of the subgraph $G'(\epsilon)$ formed by node and incident edge exclusion forms the constrained set of paths.

C Commensurabilty Techniques

This section explains the validity of the commensurability numeral calculations. These justify the quanta computation algorithms for both the Extended Haimes algorithm and the Conservative Routing algorithm.

Theorem C.1 *Algorithm "compute quanta" δ converges $\delta_i(n) \rightarrow_n \delta \quad \forall i \in V$*

Proof The sequence $\delta_i(n)$ is monotone non-increasing and is bounded from below by 0. Moreover the number of iterations for convergence is finite as the there are only finitely many different trust and delay values on the edges. Thus the sequence $\delta_i(n)$ converges.

Suppose $\delta_j(n)$ converges to different value. Say $\delta_i(n) < \delta_j(n)$. Suppose i and j are adjacent nodes, then by message passing both nodes achieve a common smaller value $\delta_i(n+1)$. If the same argument is carried out for every node, for a connected graph all the nodes achieve the common value δ.

Theorem C.2 *The delay and trust of a path is commensurable in the quantum δ , the convergent value.*

Proof $\forall p \in \mathcal{P}_{SD} \quad d(p) = \sum_{i \in p} d(i)$. Here each $d(i)$ is commensurable in δ. Then so is their sum.

$\forall p \in \mathcal{P}_{SD} \quad t(p) = \min_{i \in p} d(i)$. Here each $t(i)$ is commensurable in δ. Then so is their minimum.

The previous theorems indicate that the delay and trust of the paths change by the quantum δ and no smaller than that. This suggests that this quantum can be used as a step size for searching for optimal paths.

D Proof of Lexicographic Optimality of Biased Routing Algorithms

As mentioned the conditions of convergence of the iteration of Biased Algorithms is discussed in detail in [13]. This section proves that the given semiring indeed extracts the lexicographic paths distributively. For the sake of exposition, only the Delay Biased Routing is considered. The corresponding proof for Trust Biased Routing follows trivially. The following distributed algorithm solves the conservative routing problem.

Proof At source S, let $p*$ be the path to reach D through composition of paths to reach D from its neighbours once the algorithm converges. Let us suppose $p*$ is not Lexicographically optimal for Delay Biased Routing. Then $\exists p \neq p*$ such that either
Case I . $d(p) < d(p*)$. This is not possible as all the sub-paths are optimal in the delay sense by definition of $\oplus_d$.
Case II. $d(p) = d(p*)$ and $t(p) > t(p*)$. This is not possible too, because the sub-paths are optimal in the trust sense if the delays are equivalent (from the definition of $\oplus_d$).

A similar proof follows for trust-biased routing.

E Proof for the Max-Ordering Optimality of Conservative Routing Algorithm

The inspiration for the algorithm to solve the Max-Ordering Optimization problem, comes from its corresponding Decsion problem.

Max-Ordering Delay-Trust Optimization Problem

$$\min_{p \in \mathcal{P}_{SD}} max(d(p), t_N(p))$$

Max-Ordering Delay-Trust Decision Problem $DP(\epsilon)$

"Given an $\epsilon > 0$ does there exist a path such that $max(d(p), t_N(p)) \leq \epsilon$". The smallest $\epsilon > 0$ that for which the decision problem is answered as a yes corresponds to the optimal value of the optimization problem. The corresponding path is the conserative routing path.

E.1 Validity of the Conserative Routing Algorithm

The conservative Routing Algorithm makes use of the above claim of the Decision Problem.

Theorem E.1 *There exists a sequence of Decision Problems $DP(\epsilon^n)$ which yeild a "yes" in a finite number of iterations.*

Proof Consider a sequence of ϵ^n is chosen as per the iteration $\epsilon^{n+1} = \epsilon^n + \delta$. This ϵ^n is a montone increasing sequence. This because $\delta > 0$ from the commensurability of numerals in the problem. So if the graph is SD-connected (there exists atleast one path from S to D) the sequence of Decision Problems $DP(\epsilon^n)$ must output a "yes" in a finite number of steps. Thus the theorem has been proved by showing the construction.

Theorem E.2 *The above sequence of decision problems $DP(\epsilon)$ are solved by the Conservative Routing Algorithms.*

Proof Algorithm 3 of the Conservative Routing Algorithm generates the sequence $\epsilon^{n+1} = \epsilon^n + \delta$.
The Decision Problem

$$\begin{array}{rl} & \max(d(p), t_N(p)) \leq \epsilon^n \\ \Rightarrow & d(p) \leq \epsilon^n \wedge t_N(p) \leq \epsilon^n \\ \Rightarrow & d(p) \leq \epsilon^n \wedge t(p) \geq C_t - \epsilon^n \end{array}$$

The second condition $t(p) \geq C_t - \epsilon^n$ is accounted by the Graph Reduction step of Alg.12 of Conservative Routing. (This is proved in B.1). The first condition $d(p) \leq \epsilon^n$ can be accounted/checked from computing the optimal delay path in the reduced graph(second condition). In the reduced graph if $d(p'*) = \epsilon^n$, then the algorithm terminates at a path which satisfies the second and first condition for the lowest possible value of ϵ^n. Thus it terminates at a conservative path.

References

[1] Azer MA El-Kassas SM Hassan AWF and El-Soudani MS. A survey of trust and reputation schemes in ad hoc networks. In *Third International Conference on Availability, Reliability and Security*, pages 881–886, 2008.

[2] Le Boudec JY Bucchegger S. Performance analysis of the confidant protocol (cooperation of nodes: Fairness in dynamic ad-hoc networks). In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, 2002.

[3] V. Chankong and Y.Y.Haimes. *Multiobjective Decision Making: Theory and Methodology.* Elsevier Science Publishing Co,Inc, 1983.

[4] Matthias Ehrgott. *Mutlicriteria Optimization.* Springer, 2000.

[5] Krishnamurthy SV Eriksson J, Faloutsos M. Routing amid colluding attackers. In *IEEE International Conference on Network Protocols*, pages 184–193, 2007.

[6] Baras JS Eschenauer L, Gligor V. *Security Protocols*, chapter On Trust Establishment in Mobile Ad-Hoc Networks. Lecture Notes in Computer Science. Springer, 2004.

[7] Loeliger H Kschischang FR, Frey BJ. Factor graphs and sum-product algorithm. *IEEE Transactions on Information Theory*, 46:489–519, 2001.

[8] Aji SM McEliece RJ. The generalized distributive law. *IEEE Transactions on Information Theory*, 46(2):325–343, 2000.

[9] Molva R Michiardi P. Core: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In *Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security: Advanced Communications and Multimedia Security*, pages 107–121, 2002.

[10] Sisalem D Rebahi Y, Mujica V. A reputation-based trust mechanism for ad hoc networks. In *Proceedings of the 10th IEEE Symposium on Computers and Communications*, pages 37–42, 2005.

[11] Zechhauser R Resnick P. Trust among strangers in internet transactions: Emperical analysis of ebay's reputation system. *Advances in Applied Microeconomics: The Economics of the Internet and E-Commerce*, 11:127–157, November 2002.

[12] R.T. Rockafellar. *Network Flows and Monotropic Optimization.* Athena Scientific, 1998.

[13] Gunter Rote. Path problems in graphs, 1989.

[14] Y G Le Boudec S Buchegger. Nodes bearing grudges: Towards routing security, fairness, and robustness in mobile ad hoc networks. In *Proceedings of Parallel, Distributed and Network-Based Processing*, pages 403–410, 2002.

[15] Y G Le Boudec S Buchegger. Self-policing mobile ad hoc networks by reputation systems. *IEEE Communications Magazine*, 43(7):101–107, July 2005.

[16] Uusitalo I Savola R. Towards node-level security management in self-organizing mobile ad hoc networks. In *Advanced International Conference on Telecommunications and Internaational Confernce on Internet and Web Application and Services*, pages 36–36, Feb 2006.

[17] Baras JS Theodorakopoulos G. On trust models and trust evaluation metrics for ad hoc networks. *IEEE Journal on Selected Areas in Communication*, 24(2):318–328, 2006.

[18] Poor V Verdu S. Abstract dynamic programming models under commutativiy conditions. *SIAM Journal on Control and Optimization*, 25(4):990–1006, 1987.

[19] L.S. Lasdon Y.Y Haimes and D.A. Wismer. On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems,Man and Cybernetics*, 1:296–297, 1971.