DrATC: Dynamic routing Algorithm based on Trust Characteristics

Davide Ferraris[1] and Lorenzo Monti[2]

[1] Network, Information and Computer Security Lab, University of Malaga, Malaga, Spain
ferraris@uma.es

[2] Cubit Innovation Labs, Via Mario Giuntini, 25, Cascina, Pisa, Italy
lorenzo.monti@cubitlab.com

Abstract. In this paper, we propose a dynamic routing algorithm that leverages various trust characteristics to determine the most trusted path in a network. Trust, a multifaceted concept, encompasses attributes such as direct and indirect experiences, transitivity, directionality, context-dependence, and more. Our approach allows the routing protocol to selectively incorporate these characteristics to enhance the decision-making process. For instance, in scenarios prioritizing direct trust, nodes route packets based solely on direct interactions with their neighbors. In more complex scenarios, both direct and indirect trust are considered, utilizing recommendations from trusted nodes to establish trust with previously un-contacted nodes. We also explore the use of alternative routes based on specific trust values, ensuring sensitive data is routed through the most trustworthy paths. By integrating these trust metrics, the proposed algorithm dynamically adapts to varying network conditions and requirements, improving the overall reliability and security of the data transmission. Our experimental results demonstrate the algorithm's effectiveness in selecting trusted paths and highlight the importance of context and adaptability in trust-based routing. This work contributes to the field by providing a flexible and robust framework for incorporating trust into dynamic routing decisions, paving the way for more secure and reliable network communication.

Keywords: Trust · Network · Routing

1 Introduction

Since the beginning of the human history, it has always been a problem to decide which route to choose either was for hunting or deliver commercial goods. This problem has been transferred in the routing protocols when the first packets were sent to the internet. However, the common denominator in order to choose the path has always been one: trust.

In fact, either we are hunting, deliver a "real" packet or a TCP packet, we need to trust the receiver or the intermediary in order to interact with it. However, if we consider fundamental routing protocols that are still used today in many applications, sometimes trust has been just considered as a default characteristic. For example, for Dijkstra algorithm, it is only important to choose the shortest path from an origin node to a destination node in a network. It is true that the metric chosen can be the distance, time, cost, but we have always to trust the nodes. What happens if one of the nodes in the middle of the transmission wants to behave maliciously? The packet will be lost or worst. Because sometimes, the shortest path is not the best one. This is similar to the children story where the hero has to choose between the right path in a forest where the shortest one was represented by evil trees, fog and bright eyes in the darkness and the longest path was represented by light, peaceful animals and marvelous trees.

Thus, our point is, trust cannot be left out of the equation when choosing a "path". However, trust is difficult to define [7]. There is not a standard definition of it because it is multi disciplinary. Moreover, trust has many characteristics that usually are against each others [1]. Thus, in this paper we will define the characteristics of trust and then we will present a dynamic routing algorithm that is based on the same characteristics.

Internet routing protocols, such as the Border Gateway Protocol (BGP), Open Shortest Path First (OSPF), and Intermediate System to Intermediate System (IS-IS), form the backbone of the internet's infrastructure [14]. They are the algorithms and rules that determine how data packets are directed from one router to another, guiding them along the most efficient and reliable path. These protocols are the unsung heroes of the digital age, silently managing the intricate web of connections that allow us to send emails, stream videos, make online purchases, and access countless online services.

However, our task is design each routing algorithm as they will only be based on the trust levels among the nodes. Such nodes can be an Internet of Things (IoT) device belonging to a cluster, sensors in a common network, basically everything that can be connected. Thus, we will only focus on trust and general nodes. We will represent a decentralized network where each node can have enough computational power to compute a trust value according to the different characteristics [26].

The paper is composed as follows. In Section 2, we discuss the related work. In Section 3, we present the trust characteristics. In Section 4, we present the dynamic routing algorithm based on trust characteristics. In Section 5, we will propose three examples showing how the algorithm work and then, in Section 6, a use case presenting on how the trusted routing algorithm works. Finally, in Section 7, we conclude and present the future work.

2 Related works

In this section, we will firstly present definitions of trust, then we will discuss about which algorithm exists in routing and finally we discuss about existing works about trusted routing.

2.1 Trust

Trust is a multifaceted concept with varied definitions across disciplines [18]. For this reason, trust is a foundational concept in many fields, including sociology, psychology, and computer science [3]. In the context of computer science, trust refers to the confidence in the reliability, integrity, and security of entities within a system, such as nodes in a network, software components, or entire systems [8]. The concept of trust is particularly critical in decentralized and distributed systems, where direct oversight and control are limited.

Several definitions of trust have been proposed in the literature, each emphasizing different aspects of the concept. We describe three of the many of them:

- Marsh [18]: Marsh introduced the idea of formalizing trust as a computational concept, defining it as a value that can be used to predict the future behavior of an entity based on past interactions. This approach laid the groundwork for incorporating trust into automated decision-making systems.
- Gambetta [11]: Trust is a means to reduce the complexity of interactions in an uncertain environment, serving as a mechanism to manage the uncertainty associated with the actions of others. Gambetta's definition underscores the role of trust in simplifying complex, uncertain interactions.
- Jøsang [15]: Trust is the subjective probability by which an entity believes that another entity will perform a particular action on which its welfare depends. This definition highlights the probabilistic and subjective nature of trust.

In computer science, trust is often modeled and quantified to enhance the security, reliability, and performance of systems [13]. Trust models are used in various domains, including network security, distributed systems, and e-commerce [9].

Trust models in computer science typically involve the following components [10]:

- **Direct Trust**: Derived from direct interactions and experiences with an entity. For example, if a node in a network consistently forwards data packets correctly, it gains a higher direct trust score.
- **Indirect Trust (Reputation)**: Based on recommendations or observations from other entities. If multiple nodes report positive interactions with a particular node, that node's reputation (indirect trust) increases.
- **Hybrid Trust**: Combines both direct and indirect trust to form a more comprehensive trust evaluation. This approach helps mitigate the limitations of relying solely on one type of trust metric. Trust metrics can be computed using various mathematical and probabilistic methods, including Bayesian networks, fuzzy logic, and weighted averages.

However, several applications of trust have been developed in computer science.

One of them has been performed for Ad-Hoc and Sensor Networks. In these decentralized networks, trust-based routing algorithms are used to ensure secure and reliable communication. For example, Ilyas et al. [14] proposed a trust-based routing framework for ad-hoc networks that isolates malicious nodes by evaluating trust metrics.

Trust has also been implemented in Peer-to-Peer (P2P) Networks [4]. Here, trust models help identify reliable peers and mitigate the risk of malicious behavior. Systems like BitTorrent use reputation systems to encourage cooperative behavior among peers.

Trust is even more important considering also the final users in E-Commerce systems where trust plays a crucial role in online transactions [15]. Systems like eBay and Amazon use trust and reputation systems to build consumer confidence and reduce the likelihood of fraud.

Moreover, we can state that in Cloud Computing, trust management in cloud environments ensures that users can rely on cloud service providers to handle their data securely and reliably. Trust models assess the trustworthiness of different cloud services based on factors like service history and security practices [22].

However, we believe that trust can play a crucial role if it is mainly considered in routing algorithms. For this reason, we will describe now what are routing algorithms and then the existing routing algorithms that consider trust partially.

2.2 Routing Algorithms

Routing algorithms are essential components in network systems, responsible for determining the optimal paths for data transmission from source to destination [14]. Traditional routing algorithms can be broadly classified into two categories: static and dynamic.

On one hand, Static Routing Algorithms involve predefined routes that do not change unless manually reconfigured. Examples include algorithms used in simple, small-scale networks where changes are infrequent. Dijkstra's algorithm is one of them [6]. This is a graph search algorithm used to find the shortest path from a starting node to all other nodes in a weighted graph. It uses a priority queue to repeatedly select the node with the smallest known distance, updates the shortest paths to its neighboring nodes, and marks it as visited. The process continues until the shortest paths to all nodes are determined. This algorithm guarantees the shortest path in graphs with non-negative edge weights [2].

On the other hand, Dynamic Routing Algorithms adapt to network conditions in real-time, responding to changes in network topology, traffic load, and link failures [14]. Common examples include:

- Distance Vector Routing (DVR): Each router maintains a table (vector) of the minimum distance to every other router. The Bellman-Ford algorithm is a foundational approach in DVR.
- Link State Routing (LSR): Routers have complete network topology information and independently compute the shortest path to every other router using algorithms like Dijkstra's, but in a dynamic way.

- Path Vector Protocols: Used in inter-domain routing, such as the Border Gateway Protocol (BGP), which maintains path information that gets updated as routes change.

2.3 Trusted Routing Algorithms

Trusted routing algorithms incorporate trust metrics into the route selection process to enhance security, reliability, and performance. These algorithms extend traditional routing methods by integrating trust evaluations of nodes and links, considering factors such as past behavior, recommendations, and security credentials.

Trust-Based Routing in Ad-Hoc Networks is useful. In fact, Ad-hoc networks, due to their decentralized nature and lack of fixed infrastructure, particularly benefit from trust-based routing. Pirzada et al. [25] proposed a trust-based routing framework for ad-hoc networks, where trust metrics are used to identify and isolate malicious nodes, thereby improving network security and reliability.

In Wireless Sensor Networks (WSNs), trust-based routing algorithms aim to ensure data integrity and network longevity. Khan et al. [17] presented a trust-aware routing protocol that evaluates trustworthiness based on direct and indirect observations, thus enhancing the resilience of the network against attacks and failures.

Some approaches integrate trust mechanisms into existing routing protocols. For instance, Perkins [24] enhanced the Ad hoc On-Demand Distance Vector (AODV) protocol by incorporating a trust model that assesses node reliability based on historical interactions, thereby improving route selection and network performance.

However, we found lack of trust considerations in all of these methods. In fact, we believe that in order to fully consider trust in routing algorithms, we should consider that trust metrics can be computed using various methods, including:

- Direct Trust: Based on direct interactions and experiences.
- Indirect Trust: Derived from recommendations or observations from other nodes.
- Hybrid Approaches: Combine both direct and indirect trust metrics to form a comprehensive trust evaluation.
- Other Trust Characteristics: see the next section.

For this reason, we will now present the characteristics of trust and around them we will build the trusted algorithm. In order, to dynamic choose them when computing a routing path.

3 Trust Characteristics

In this paper, we propose a routing algorithm enhancing by trust considering its characteristics.

We have identified several characteristics proposed by different authors during the years. We summarize them in Table 1. In the first column there is the trust characteristic and in the second one the paper mentioning such characteristic.

Table 1. Characteristics of trust

Direct	[3]
Indirect	[1]
Transitive	[5, 27]
Directed	[27]
Dynamic	[12, 4, 23]
Context-dependent	[1, 20]
Local	[1, 5]
Global	[1]
Specific	[16, 19]
General	[16, 19]
Asymmetric	[21]
Subjective	[12, 27]
Objective	[1]
Composite-property	[12, 27]
Measurable	[27]

Now, we specify trust characteristics describing the meaning of each of them:

1. **Direct.** Trust is based on the direct experience. We can also say that trust depends on past history [3].
2. **Indirect.** Usually, if direct experience is absent, we can start computing a trust value considering the recommendation of other entities [1]. This is the basis of systems based on reputation.
3. **Transitive.** Trust can be also considered as transitive [5]. In fact, trust can be conditionally transferable, as there is the possibility to transmit/receive trust information through a path of recommendations [27].
4. **Directed.** Trust is directed. It means that we have an oriented relationship between different entities [27]. Thus, it is possible that if an entity A trusts an entity B, the opposite can be not the same (i.e., B distrusts A).

5. **Dynamic.** Trust change over time, it can increase or decrease due a several actions. Chang [4] states that "trust builds with time". In fact, an entity could trust another entity for a determined context in a specific moment, but this can change positively or negatively in a future moment [23]. Moreover, as Grandison stated [12] "trust must be able to adapt to the context in which a trust decision has been made and can change according to different contexts".
6. **Context-dependent.** As we mentioned before, trust is strictly connected to the context. "In general, trust is a subjective belief about an entity in a particular context [27]." and more specifically "where the trust of a node i in a node j varies from one context to another [1]".
7. **Local.** Trust can be **local** [1] because it depends on a couple of entities (i.e., Alice and Bob) and if we consider other two couples (i.e., Alice and Charlie, and Bob and Charlie), it is possible that Alice distrust Charlie, even if Bob trusts Charlie [5].
8. **Global.** As Abdelghani stated "trust also called reputation means that every node has a unique trust value in the network which can be known by all other nodes [1]".
9. **Specific.** On the one hand, we can state that trust can be specific [19, 16]. This happens because an entity can trust another entity only for a specific purpose or service.
10. **General.** On the other hand, trust can be considered as general [19, 16]. In this case, the an entity A trusts an entity B independently from the purpose or a specific context.
11. **Asymmetric**. This means that two entities tied by a relationship may differently trust each other. It means that even if A trusts B, this does not imply that B trusts A [21].
12. **Subjective.** Trust is subjective because it is related to a personal opinion based on different factors (i.e., past experience) and these factors can be differently important for different entities [12]. In fact, trust is perceived in a dissimilar way for each individual in a particular context [27].
13. **Objective.** Trust can be also considered **objective** "such as when trust is computed based on Quality of Service (QoS) properties of a device [1]". Furthermore, an objective parameter to compute trust is also known as **reputation**. Connected to *indirect.*
14. **Composite-property.** Trust can be composed of different attributes. For example as Grandison [12] stated it can be composed of "reliability, dependability, honesty, truthfulness, security, competence, and timeliness". Thus, compositionality is an important aspect for trust computations [27] and every attribute could have different weight.
15. **Measurable.** Finally, trust is measurable. In fact, "trust values can be used to represent the different degrees of trust an entity may have in another. [27]." This characteristic is the basis for the computation of a final trust value during trust management.

The aforementioned characteristics and their relationships are explained in Figure 1.

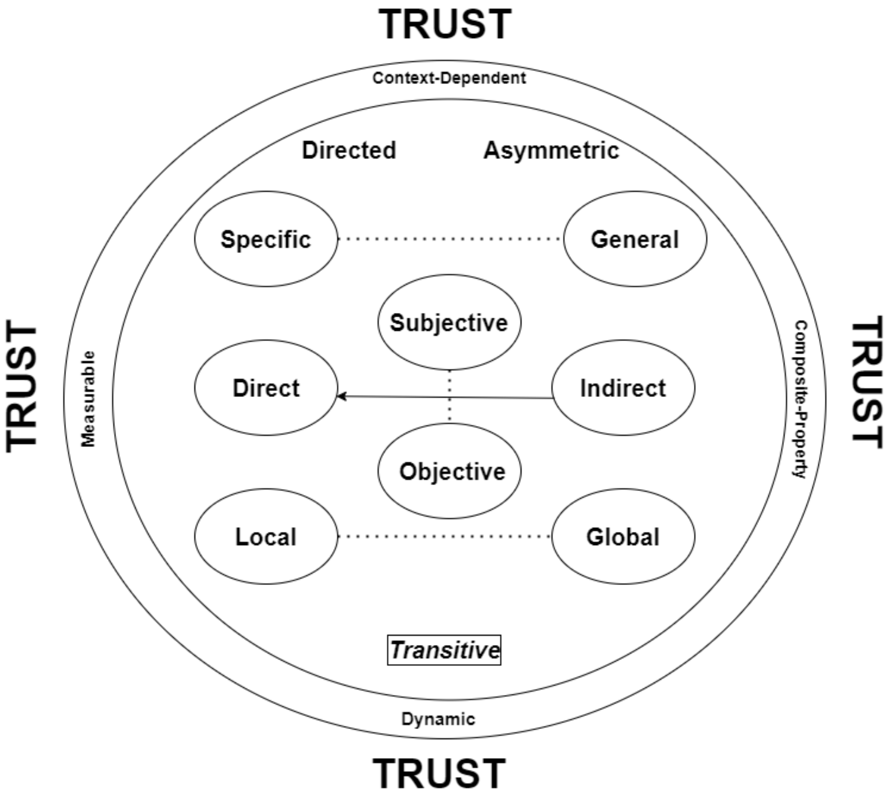

Fig. 1. Trust Characteristics and Their Relationships [9]

The outer circle indicates that the characteristics listed there are always present. The traits within the inner circle remain significant in all contexts (i.e., directed and asymmetric). *Transitive* is italicized because it is not always applicable and is placed in a separate rectangle. Additionally, three pairs of characteristics are connected by dotted lines, signifying that they are mutually exclusive. Specifically, trust can be either specific or general, subjective or objective, and local or global. However, trust can also simultaneously be specific, objective, and global.

At the center of the diagram is the pair: direct and indirect. An arrow extends from indirect to direct, indicating that indirect trust can sometimes lead to the formation of direct trust. This occurs when there is no prior direct knowledge (i.e., no past interactions), and an indirect parameter is needed to start building a trust value. This process is illustrated by the arrow.

4 DrATC: Dynamic routing Algorithm based on Trust Characteristics

As we mentioned before, dynamic routing algorithms are essential for maintaining efficient and reliable communication in networks, particularly in environments where network conditions and node trustworthiness can vary over time. By incorporating trust characteristics into dynamic routing algorithms, we can enhance the security and reliability of the network. This section outlines a general model for a dynamic routing protocol that leverages trust metrics to determine the most trusted paths for data transmission.

The trust-based dynamic routing algorithm integrates traditional routing metrics (i.e., hop count, latency) with trust characteristics to make more informed routing decisions. The algorithm continuously evaluates and updates trust levels of nodes based on their behavior and interactions, ensuring that the most trusted paths are chosen dynamically as network conditions evolve.

In Section 3, we have discussed about trust characteristics. For our algorithm, we want to take mainly into considerations two aspects. Direct trust and indirect trust. For the latter, we can also consider transitive trust.

More specifically:

- Direct Trust: Nodes evaluate direct trust based on past interactions. This involves recording and analyzing the outcomes of previous data exchanges, such as successful transmissions and detected security breaches.
- Indirect Trust: Nodes consider recommendations from other trusted nodes. This mechanism allows nodes to gather trust information about nodes they have not interacted with directly. If we consider transitive trust, nodes can infer trust relationships through chains of trusted nodes, allowing them to build trust with nodes beyond their immediate neighbors.

However, we will consider other important trust characteristics into the routing metrics, such as reliability, security, competence, and context-specific trust values. These metrics are weighted and combined to form a composite trust score for each node.

As trust is dynamics, we want to transfer this capability to the algorithm too. For this reason, trust levels are dynamically updated based on ongoing interactions and feedback from other nodes. This ensures that trust evaluations reflect the most current network conditions and node behaviors. Thus, the protocol adjusts trust scores in real-time, considering factors such as recent successful transmissions, detected anomalies, and recommendations from other nodes.

As also context is fundamental for trust considerations, we perform trust evaluations depending on the context, meaning trust scores can vary based on the type of data being transmitted and the specific requirements of the communication (i.e., higher security needed for sensitive data). Nodes dynamically adjust trust evaluations based on the context, ensuring that routing decisions align with the specific needs of the network at any given time.

In the algorithm, we implement a Routing Decision Process (RDP) considering the following steps: Path Discovery, Path Evaluation and Path Selection. More specifically:

1. Path Discovery: Nodes initiate path discovery processes to identify potential routes to the destination. During this process, nodes exchange trust information and routing metrics.
2. Path Evaluation: Each potential path is evaluated based on its overall trust score, which is a composite of the trust scores of the intermediate nodes along the path.
3. Path Selection: The path with the highest trust score is selected for data transmission. This ensures that the chosen path not only meets traditional routing criteria but also maximizes trustworthiness.

However, as the algorithm is dynamic, feedback and learning are necessary. Thus, nodes continuously monitor the performance of the selected paths and provide feedback to update trust evaluations. This feedback loop allows the protocol to learn from network conditions and improve routing decisions over time. Nodes share their trust evaluations and experiences with other nodes, enhancing the overall trust awareness and cooperation within the network.

The flow related to the paths and related feedbacks is showed in Figure 2.

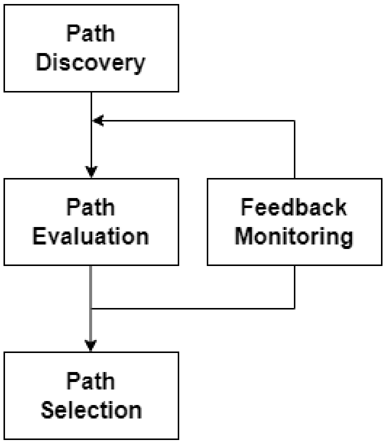

Fig. 2. Algorithm Flow and Feedback

4.1 General Operation of the Model

In this part, we describe the general behaviour of the algorithm. Firstly there is the initialization phase.

So, each node initializes its trust evaluations based on direct experiences and recommendations from other nodes. Nodes also set up mechanisms to dynamically update trust scores as interactions occur. Path Discovery and Trust Evaluation:

When a node needs to send data, it initiates a path discovery process, during which it gathers trust information about potential intermediate nodes. Nodes exchange routing and trust information, allowing the initiating node to evaluate the trustworthiness of each potential path.

The initiating node evaluates the trust scores of all potential paths and selects the one with the highest composite trust score. The selected path is used for data transmission, ensuring that the most trusted nodes are involved in the communication.

During and after data transmission, nodes monitor the performance of the selected path and provide feedback to update trust evaluations. Trust scores are dynamically adjusted based on the success or failure of the transmission, recommendations from other nodes, and any detected security issues.

The algorithm continuously adapts to changing network conditions and node behaviors. Trust evaluations are updated in real-time, ensuring that routing decisions remain optimal and secure.

5 Algorithm Utilization Scenarios

As we presented in the previous sections, trust has several characteristics. We want to enable our routing algorithm in order to choose among them in order to compute trust according to the selected one. For example, in this section we show a routing algorithm considering only direct trust, without consider transitive or indirect trust. In this case, each node will have to choose among different nodes only considering the direct experience.

In the second example, we will provide the possibility to enable both direct and indirect, in order to have the possibility to compute a trusted value also for nodes with no direct experience.

The third one, will consider the possibility that a trust value changes after receiving a feedback and dynamically change the more trusted route.

In all of these examples we will consider trust as directed as described in Section 3.

5.1 Scenario 1: Direct Trust-Based Routing

In this scenario, the routing protocol is designed to consider only direct trust. Each node in the network maintains a record of its direct interactions with neighboring nodes. The trust value is calculated based solely on these direct interactions. When a node needs to forward a packet, it selects the next hop based on the highest direct trust value. This ensures that the path chosen consists of nodes that have had positive direct interactions with their immediate neighbors.

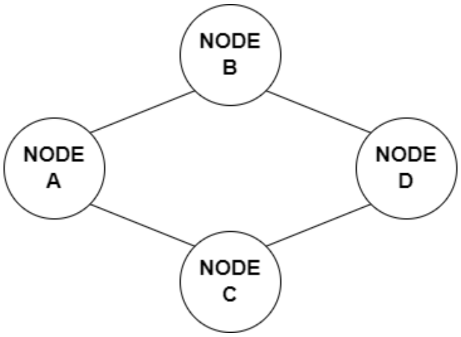

Fig. 3. Nodes Distribution in Scenario 1

We consider four different nodes and the need to be satisfied is that Node A needs to send packets to Node D.

In this scenario, we have the following situation also represented in Figure 3:

- Node A has direct trust values for Node B (0.9) and Node C (0.7).
- Node B has a direct trust value for Node D (0.8).
- Node C has a direct trust value for Node D (0.6).

According to these values, the routing decision will be the following:

Node A selects Node B as the next hop because the direct trust value (0.9) is higher than for Node C (0.7). Then, Node B forwards the packet to Node D based on its direct trust value (0.8).

In this case, there are no feedback changing the trust values.

5.2 Scenario 2: Direct and Indirect Trust-Based Routing

In this scenario, the routing protocol is enhanced to consider both direct and indirect trust. Nodes can rely on recommendations from trusted neighbors to establish trust values for nodes with which they have no direct interactions. This approach allows the network to be more flexible and robust, especially in dynamic environments where direct interactions may be limited.

In this case we have six nodes and the need is for Node A to send packets to Node F.

In this scenario, we have the following situation also represented in Figure 4:

- Node A has direct trust values for Node B (0.8) and Node D (0.7).
- Node B has a direct trust value for Node C (0.3).
- Node D has a direct trust value for Node E (0.9).
- Node C and Node E have a direct trust value for Node F and they are (0.85) for C and (0.85) for E.

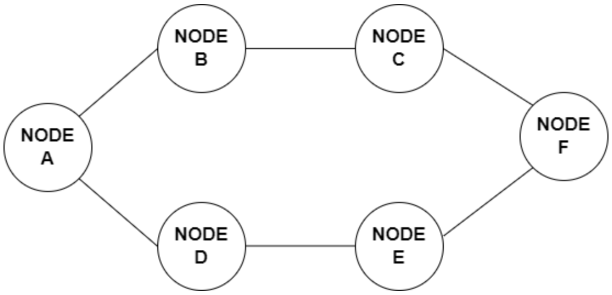

Fig. 4. Nodes Distribution in Scenario 2

According to these values, the routing decision will be the following:

Node A selects Node B as the next hop because the direct trust value (0.8) is higher than for Node D (0.7). However, the indirect trust value communicate by B to A according to the following step C is dramatically lower than the indirect trust value communicated by D to A according to Node E: (0.3) vs (0.9). So, the algorithm prefers to proceed from A through D insteaf of B. Then, the final values from C and E to F are the same (0.85). Thus, the final path will be A -> D -> E -> F.

5.3 Scenario 3: Adapting Routes Based on Feedback

In this scenario, we consider also the dynamic feedback mechanism to adjust trust levels in real-time. In this scenario, a node initially trusted for routing behaves maliciously, compromising the trust relationship. Our protocol must then adapt by recalculating and finding a new trusted path based on updated trust values.

Let's consider a network with nodes A, B, C, D, and E. The initial trust values between these nodes are as follows:

- Node A to Node B: 0.9
- Node A to Node C: 0.7
- Node B to Node D: 0.8
- Node C to Node D: 0.85
- Node D to Node E: 0.9
- Node B to Node E: 0.75
- Node C to Node E: 0.8

The goal is to find the most trusted path from Node A to Node E. Initially, the protocol selects the path based on the highest trust values. The most trusted path is A -> B (0.9), B -> D (0.8) and D -> E (0.9).

The total trust value for this path is 0.9 * 0.8 * 0.9 = 0.648.

However, suppose Node A receives feedback indicating that Node B has behaved maliciously, and its trust value drops to 0.4. The updated trust values are:

- Node A to Node B: 0.4
- Node A to Node C: 0.7
- Node B to Node D: 0.8
- Node C to Node D: 0.85
- Node D to Node E: 0.9
- Node B to Node E: 0.75
- Node C to Node E: 0.8

With the updated values, the protocol must recalculate the path from Node A to Node E, prioritizing nodes with higher trust values. The recalculated path is: A -> C (0.7), C -> D (0.85) and D -> E (0.9)

The total trust value for this new path is 0.7 * 0.85 * 0.9 = 0.5355.

Although the new path's total trust value is lower than the initial path, it avoids the malicious behavior of Node B, ensuring a more secure route. This dynamic adjustment mechanism helps maintain the integrity and security of the data transmission, even in the face of changing network behaviors.

In all of these examples, trust is considered directed, meaning that the trust relationship from one node to another is not necessarily reciprocal. This ensures that the routing decisions are based on the specific trust values in the given direction, optimizing the trustworthiness of the chosen path.

6 Use Case: Trust-Based Routing in a Network

In the previous section, we have presented three scenarios in which we have showed how the algorithm works for direct and indirect trust and what happens when a trust value change over time. In this section, we present a more complex use case considering different trust characteristics to find the most trusted path from point A to point B incorporating a variety of trust metrics into the routing algorithm.

In this use case, node A wants to send data to node B. The network comprises several intermediate nodes (e.g., C, D, E, F, etc.), each with varying levels of trustworthiness based on different trust characteristics. Node A must select the most trusted path to ensure data integrity and security.

The different options are the following:

1. Direct Trust: Node A has previous interactions with node C and has a history-based trust value for node C.
2. Indirect Trust: Node A has no direct interaction with node D but receives recommendations from node C about D.
3. Transitive Trust: Trust can be passed along a chain, e.g., if A trusts C and C trusts D, then A can consider trusting D through C.
4. Directed Trust: Trust is not reciprocal, meaning A might trust C, but C might not necessarily trust A.
5. Dynamic Trust: Trust levels change over time; A may update its trust value for C based on recent interactions.

6. Context-Dependent Trust: Trust values vary depending on the context (e.g., trust for data transmission might differ from trust for control messages).
7. Local Trust: Trust relationships are specific to pairs of nodes; A might trust C differently than it trusts D.
8. Global Trust: Each node has a reputation score known to all nodes in the network.
9. Specific Trust: A may trust C for routing control messages but not for data transmission.
10. General Trust: A may generally trust D without specific context.
11. Asymmetric Trust: Trust between A and C may not be mutual.
12. Subjective Trust: Trust evaluations are subjective and based on A's perspective.
13. Objective Trust**: Trust is computed using objective metrics like QoS or uptime.
14. Composite-Property Trust: Trust evaluations consider multiple attributes like reliability, competence, and security.
15. Measurable Trust: Trust values are quantifiable, allowing for comparison between paths.

The algorithm, will proceed with the steps presented in Section 4.

In the Path Discovery step, Node A gathers trust information about all intermediate nodes (C, D, E, F, etc.) from its direct interactions, recommendations from other nodes, and global reputation scores.

Then, during the Path Evaluation, we can have four different approaches. The first one is related to the Direct Trust computation, in which the algorithm calculate direct trust scores based on A's history with intermediate nodes. Another option is to consider Indirect Trust, thus there is incorporate recommendations from trusted nodes. Another possibility is to compute Transitive Trust scores using known trust relationships. Another option, is to consider only Context-Dependent Trust. In this case, there will be an adjustment of trust scores based on the specific context of the data transmission.

After choosing the approach, Node A evaluates possible paths to B considering the aggregated trust scores of intermediate nodes. Each path is assigned a trust score based on the composite trust values of the nodes in the path.

Now, there is the Path Selection part where the path with the highest trust score is selected as the most trusted path from A to B. This path should ideally maximize direct and indirect trust, consider transitive trust relationships, and account for the context and dynamics of trust values.

After the path is chosen, it is possible to proceed with Data Transmission. Thus, Node A sends the data to node B via the selected trusted path. During transmission, trust levels are monitored and adjusted dynamically based on the ongoing interactions and feedback.

An example is the following. Assume the following trust values:

- Direct Trust (A -> C): 0.8
- Indirect Trust (A -> D via C): 0.7 (C recommends D)

- Transitive Trust (A -> E via C and D): 0.6 (A trusts C, C trusts D, D trusts E)
- Context-Dependent Trust: Adjusted based on the type of data (i.e., more stringent for sensitive data)
- Composite Trust: Consider factors like reliability (0.8), security (0.9), and competence (0.7)

The trust calculation must be performed in order to calculate the trust score for each path:

- Path 1 (A -> C -> B): Trust Score = 0.8 (Direct Trust with C)
- Path 2 (A -> D -> B): Trust Score = 0.7 (Indirect Trust with D)
- Path 3 (A -> C -> D -> B): Trust Score = 0.7 * 0.8 = 0.56 (Transitive Trust via C and D)
- Path 4 (A -> E -> B): Trust Score = 0.6 (Transitive Trust via E)

Based on these calculations, Path 1 (A -> C -> B) would be chosen as the most trusted path.

In this example, Node A utilizes various trust characteristics to evaluate and select the most trusted path to Node B. By integrating direct, indirect, transitive, and other trust properties, the routing decision is optimized to ensure the highest level of trustworthiness, adapting dynamically to changes in the network. This approach highlights the complexity and importance of trust in secure and reliable network routing.

7 Conclusion and Future Work

In this paper, we proposed a dynamic routing algorithm that leverages trust characteristics to determine the most reliable paths in a network. By incorporating various trust metrics such as direct, indirect, and transitive trust, our approach allows for a nuanced evaluation of node relationships, ensuring that routing decisions are made based on comprehensive trust assessments. The flexibility to enable or disable specific trust characteristics demonstrates the adaptability of our protocol to different network scenarios and security requirements.

As a future work, we will improve the algorithm considering all the trust characteristics, defining it in a more rigorous way and applying it to complex scenarios. Moreover, we will propose an adaptation of BGP including trust characteristics.

Acknowledgments

This work has been supported by the EU project HORIZON-MSCA-2021-SE-01 under grant agreement No 101086308 (DUCA) and SecTwin 5.0 (TED2021-129830BI00) funded by the Ministry of Science and Innovation (Agencia Estatal de Investigación (AEI)/10.13039/501100011033) and by the the European Union "NextGenerationEU"/PRTR.

This work reflects only the authors view and the Research Executive Agency is not responsible for any use that may be made of the information it contains.

References

1. Wafa Abdelghani, Corinne Amel Zayani, Ikram Amous, and Florence Sèdes. Trust management in social internet of things: a survey. In *Conference on e-Business, e-Services and e-Society*, pages 430–441. Springer, 2016.
2. Isaac Agudo, Carmen Fernandez-Gago, and Javier Lopez. A model for trust metrics analysis. In *International Conference on Trust, Privacy and Security in Digital Business*, pages 28–37. Springer, 2008.
3. Thomas Beth, Malte Borcherding, and Birgit Klein. Valuation of trust in open networks. In *European Symposium on Research in Computer Security*, pages 1–18. Springer, 1994.
4. Junsheng Chang, Huaimin Wang, and Yin Gang. A dynamic trust metric for p2p systems. In *2006 Fifth International Conference on Grid and Cooperative Computing Workshops*, pages 117–120. IEEE, 2006.
5. Bruce Christianson and William S Harbison. Why isn't trust transitive? In *International workshop on security protocols*, pages 171–176. Springer, 1996.
6. E Dijkstra. Dijkstra's algorithm. *nd http://en. wikipedia. org/wiki/Dijkstra_ algorithm (accessed 2007/10/12)*, 1959.
7. Carmen Fernandez-Gago, Francisco Moyano, and Javier Lopez. Modelling trust dynamics in the internet of things. *Information Sciences*, 396:72 – 82, 2017.
8. Davide Ferraris, Carmen Fernandez-Gago, and Javier Lopez. A trust by design framework for the internet of things. In *NTMS'2018 - Security Track (NTMS 2018 Security Track)*, Paris, France, February 2018.
9. Davide Ferraris, Carmen Fernandez-Gago, Rodrigo Roman, and Javier Lopez. A survey on iot trust model frameworks. *The Journal of Supercomputing*, 80(6):8259–8296, 2024.
10. Giancarlo Fortino, Lidia Fotia, Fabrizio Messina, Domenico Rosaci, and Giuseppe ML Sarné. Trust and reputation in the internet of things: State-of-the-art and research challenges. *IEEE Access*, 8:60117–60125, 2020.
11. Diego Gambetta et al. Can we trust trust. *Trust: Making and breaking cooperative relations*, 13:213–237, 2000.
12. Tyrone Grandison and Morris Sloman. A survey of trust in internet applications. *IEEE Communications Surveys & Tutorials*, 3(4):2–16, 2000.
13. Lance J Hoffman, Kim Lawson-Jenkins, and Jeremy Blum. Trust beyond security: an expanded trust model. *Communications of the ACM*, 49(7):94–101, 2006.
14. Muhammad Ilyas, Zahid Ullah, Fakhri Alam Khan, Muhammad Hasanain Chaudary, Muhammad Sheraz Arshed Malik, Zafar Zaheer, and Hamood Ur Rehman Durrani. Trust-based energy-efficient routing protocol for internet of things–based sensor networks. *International Journal of Distributed Sensor Networks*, 16(10):1550147720964358, 2020.
15. Audun Jøsang, Roslan Ismail, and Colin Boyd. A survey of trust and reputation systems for online service provision. *Decision support systems*, 43(2):618–644, 2007.
16. Peter Kenning. The influence of general trust and specific trust on buying behaviour. *International Journal of Retail & Distribution Management*, 36(6):461–476, 2008.
17. Tayyab Khan and Karan Singh. Tasrp: a trust aware secure routing protocol for wireless sensor networks. *International Journal of Innovative Computing and Applications*, 12(2-3):108–122, 2021.
18. Stephen Paul Marsh. *Formalising trust as a computational concept.* PhD thesis, Department of Computing Science and Mathematics, University of Stirling, 1994.

19. JL Morrow Jr, Mark H Hansen, and Allison W Pearson. The cognitive and affective antecedents of general trust within cooperative organizations. *Journal of managerial issues*, pages 48–64, 2004.
20. Francisco Moyano, Carmen Fernandez-Gago, and Javier Lopez. A conceptual framework for trust models. In *9th International Conference on Trust, Privacy and Security in Digital Business (TrustBus 2012*, volume 7449 of Lectures Notes in Computer Science, pages 93–104. Springer Verlag, Sep 2012.
21. Michele Nitti, Roberto Girau, and Luigi Atzori. Trustworthiness management in the social internet of things. *IEEE Transactions on knowledge and data engineering*, 26(5):1253–1266, 2014.
22. David Nuñez, Carmen Fernández-Gago, and Jesús Luna. Eliciting metrics for accountability of cloud systems. *Computers & Security*, 62:149–164, 2016.
23. Michalis Pavlidis. Designing for trust. In *CAiSE (Doctoral Consortium)*, pages 3–14, 2011.
24. Charles E Perkins. Ad hoc on-demand distance vector (aodv) routing, internet-draft. *draft-ietf-manet-aodv08. txt*, 2001.
25. Asad Amir Pirzada, Chris McDonald, et al. Establishing trust in pure ad-hoc networks. In *ACSC*, volume 4, page 1. Citeseer, 2004.
26. Rodrigo Roman, Pablo Najera, and Javier Lopez. Securing the internet of things. *Computer*, 44(9):51–58, 2011.
27. Zheng Yan and Silke Holtmanns. Trust modeling and management: from social trust to digital trust. *IGI Global*, pages 290–323, 2008.