A Trust Framework for Peer-to-Peer Interaction in Ad Hoc Networks

Javesh Boodnah
School of Electronic Eng. and Computer Science
Queen Mary, University of London
London, United Kingdom
jave@ieee.org

Stefan Poslad
School of Electronic Eng. and Computer Science
Queen Mary, University of London
London, United Kingdom
stefan.poslad@elec.qmul.ac.uk

Abstract—As mobile devices are increasingly adopted by a wider public for more diverse applications, the question of who to trust while on the move becomes a crucial one. When operating within an ad-hoc network, a trust framework is the logical solution to provide ways to distinguish the most trustworthy neighbours. This aids in achieving reliable transfer of information, promotes use of local services and encourages sharing of content and resources. This paper proposes such a scheme to set up a trust framework, based on aggregating trust and the trust decision process around the most trustworthy nodes. Furthermore, the impact of the trust framework is analysed by observing the evolution of trust over time, the overhead such a framework generates and its effectiveness in stopping non-cooperation or malicious intent.

Keywords-ad-hoc networks; trust; framework; malicious nodes; non-cooperation

I. INTRODUCTION

Increasing mobility together with the need to access ICT services requires some access to previously unknown local services. These local services may not be networked in a fixed topology. Various scenarios can be depicted ranging from academic base stations without access to a backbone wireless/wired carrier to disaster recovery networks in remote areas.

While the application of ad hoc networking in each specific case will have its particular requirements, very often, a case can be made that dynamically generated heads mimic real life depictions where “leaders” typically emerge in most scenarios to guide the rest. One example of such an occurrence is at trade conferences and exhibitions where organisers, exhibit and floor managers can all act as trustworthy heads around which trust clusters can form.

The work presented here demonstrates the formalisation of a trust model on a pure ad hoc network [9]. This is defined as one where joining entities are previously unknown to one another and create a self-starting network. The ad hoc network differs from other distributed systems in that its member nodes undertake to provide routing themselves. There are no dedicated routing or access nodes. The implication is that trust has to be built from a neutral stand-point. One can argue that sometimes in distributed systems including ad hoc networks, requesting entities can define concepts akin to trust in order to justify whether or not to trust. For example, a system may define a quality of service that a peer must provide [2]. Indeed some researchers define trust as a “metaphor” that is the direct product of how interactions within a system evolve [8]. The building of trust within the trust model therefore takes place as a direct consequence of the lessons learned from observing, and collecting recommended observations of, the behaviour of an entity with respect to a predefined concept.

Researchers in the domain of mobile ad hoc networking (MANET) have realised that being able to freely share content with anyone has underlying trust issues.

This means that in order to enable and sustain reliable networking, a trust relationship has to be formed between interacting entities. Such a relationship is acquired over time as the entity’s experience of its surroundings increases. Without such a relationship, much chaos is likely to ensue as entities have no way of determining whom or what to depend on in order to promote their own agendas.

The notion of trust, traditionally confined to social and human networks, was first comprehensively covered in the domain of computing, by Stephen Marsh [1]. In its simplistic form and for the purpose of this paper, trust can be seen as “an inherent property in which one autonomous component cannot completely control another autonomous component but which may need to rely on or one another or require some cooperation from it” [2].

There are distinctions to be made between direct trust in which an entity utilises past interactions and knowledge to formulate an expectation of future behaviour from its object; and indirect trust where the same entity relies on other entities in order to formulate said expectation based on their being part of the same “normative institution” [2] or on their recommendations. The latter generally leads to what is known as a reputation system. A commonly quoted example is eBay’s feedback system [4], similar to many other opinion websites across the web. However, all these reputation systems, operating through recommending entities, require the presence of a centralised and trusted server to compute, store and display valid reputation information. This centralised authority is vital to such reputation systems but cannot be readily assumed to be present in the case of mobile devices. The reputation system needs to be decentralised [3, 5] such that it is accessible to entities on the move that may not have access to central repositories.

The simplest way to decentralise is to locate the reputation system on every single entity in the system [6].
However, this may not be the most efficient way in terms of additional computational workload and overheads. As demonstrated previously [7], a simple system of clustering nodes with one another and assigning tasks to certain supernodes poses some advantages while maintaining the integrity of the system. The use of clusters as a basis for a trust model will inherit similar benefits. From a trust perspective, clustering also enhances the likelihood of interaction and sharing of recommendations, thus providing more relevant and up to date reputation information.

The case for ad hoc networks can be made in instances where conventional networking is not available. Conventionally, mobile devices operate by locating the nearest available trusted wireless point in order to access a known wired backbone. This is typical of a wide area network and the topology involved is fairly simple. However, despite advances being made in lean computing and power conservation on mobile devices, these resources remain constrained and often nodes in a network find it preferable to operate on low energy in order to access local services. A mundane example would be a smart phone unsuccessfully trying to achieve a 3G connection in a patchy coverage area, thus draining its battery when local services may well have been provided via other neighbouring nodes over an application on Bluetooth. Furthermore, attempting to address unreliability by repeated retransmissions in a low energy network wastes precious bandwidth.

Surimposing trust on the ad hoc network therefore reinforces the transfer of information and aids in the delivery of dependable local services. One of the ways in which the success of a trust framework can be measured is via the quality of service that it can provide for members adopting it. For the purpose of this paper, the trust model presented utilises the success rate of networking requests as one of the criteria set for achieving quality of service. The behaviour of the model with respect to entity feedback provided and its ability to reflect changing circumstances (thus presenting an altering trust landscape) are investigated.

II. Survey of Ad Hoc Trust Models

In order to model a trust framework, there are several approaches that are required relating to the conceptual and mathematical models and the confidence building brought about through software development. Several trust models have been formulated for the ad hoc networks. Some of them are detailed below.

Schweitzer et al. proposed a method for building trust relationships and making the entities autonomous so that they were able to make decisions without referring to a central network [10]. K Ren et al. proposed a distributed trust approach which claimed to build well-established trust reputation systems without relying on any predefined assumption. Resilience towards nodes’ dynamically leaving/joining and scalability were also aims of this project [11].

The work by Liu et al proposed a trust model that is based on the update of trust levels throughout a given mobile ad hoc network by the use of Intrusion Detection Systems (IDSs) which were installed on all nodes operating in the network [12]. The focus of Buchegger and Le Boudec [13] was on the robustness on the method and its ability to detect false recommendations from any given node, be they erroneous accusations or undeserved praise. The motivation was that by solely relying on first hand observations a node does not make optimum use of all available information [13].

A mechanism by Rebahi et al [14] was developed to detect malicious packet forwarding attacks. In fact, the authors perceived that trust in ad hoc networks started by detecting misbehaviours. They made some use of reputation which they defined as simply “the perception that a node has of another’s intention and norms.” The framework by Pirzada & McDonald tried to address the issue of distributed authorities being normally required in trust evaluations by using a modified version of the Dynamic Source Routing (DSR) protocol to generate trust information passively [15].

A model was proposed by Rahman and Hailes which was a general reputation-based framework that revolved around using recommendations to decrease perceived risk within a network [16]. A framework by Liu and Issarny [17] is yet another mechanism, incorporating the dimensions of time and context, for forming, evolving and propagating reputation. It also claims effectiveness in distinguishing between truth-telling and lying agents.

These models all operate on the assumption that nodes do not have access to a central repository in order to generate trust information, that is, they conform to the definition of pure ad hoc networks. In the scenarios above, trust is typically depicted as reputation – the amalgamation of trust ratings from various sources on a particular entity. These aim to generate trust information by observation in order to distinguish trustworthy nodes from those that are not. The decentralised aspect of the framework is the recurrent theme in the above examples – this is key in order to apply any prospective model to an ad hoc network. Where the models depicted above come lacking are that they fail to recognise that it is possible to “improve the odds” of a given relationship being trustworthy by exploiting existing information during the bootstrapping phase. For example, this could be by identifying natural leaders within a specific peer group and ensure that other nodes gravitate around them.

There is also a lack of implementation of proper bootstrapping mechanisms – instead the models place reliable on a priori information before interactions take place. This is an assumption that poses some restriction on the general viability of the models in all situations. Admittedly, most trust frameworks require tailoring to their particular application, so the manifestation of such a restriction would be rare.

There is also limited evidence of the effect that having the trust framework on top of normal networking entails.
While trust frameworks are typically not expected to be as demanding in resources such as cryptographic methods, they nevertheless do use up some resources and this can affect the quality of service achievable by the network.

The trust framework proposed below attempts to address these findings.

III. FRANTIC: A DECENTRALISED TRUST FRAMEWORK

The proposed model here is based on a new FRamework for Ad-hoc Networking Trust using Interacting Clusters (FRANTIC). For a trust relationship to be established, the following are required: a truster, a trustee, a given action and a context in which to exert that trust. The model is based on clusters that interact with one another via cluster heads, effectively leading to a super cluster superimposed on the normal layer of interacting nodes. Intra-cluster communication is between nodes and the cluster-head.

A cluster is an aggregation of nodes within a particular geographical area. For the purpose of this framework, it is assumed that any node may not belong to more than one cluster. This is done to simplify the computing of reputation information from nodes as this is done by cluster heads. It also reduces the risk of collusion by malevolent nodes, although such a risk is not addressed in this particular paper and is the subject of further work.

Because of the tasks imparted to cluster heads, one of the main challenges is at bootstrapping stage where careful election of cluster heads needs to take place. Current research has focused on implementing a cluster head election process from otherwise indistinguishable nodes but very often such cluster heads can be naturally found in existing scenarios and FRANTIC aims to capitalise on such availability.

In FRANTIC, all nodes are trustors, with their respective neighbours being their trustees. Each node holds an individual trust relationship, defined by a trust metric, with its immediate neighbours (on a one-hop basis only). These trust relationships are asymmetric. This means that if a certain node A’s trust in a node B about performing an action is 0.5 (where 0.5 is the trust metric), this does not automatically imply that node B has the same amount of trust about node A.

The trust metric evolves over time due to two factors: firstly it may rise or fall depending on new data received by the truster and secondly, it may undergo a period of natural “decay” over time if data regarding a certain entity is not received over time. The “decay” promotes cooperation within the network as nodes are eager to gain a good reputation.

In FRANTIC, the trust metric is generated from a concept akin to trust as proposed in section 1. In other words, an action is set as the yardstick via which trust is generated. All nodes are observed by their peers in relation to their success or failure in carrying out this specific action. The trust metric is therefore a measure of the reputation of a given node since the observation of nodes is reported to the cluster-head which then proceeds to assign a value to the trustworthiness of a particular node based on those observations.

The scenario depicted by the model here is that of a large exhibition centre where users are spread out over a relatively large area but at the same time are bound within a confined space (the perimeter of the exhibition centre). Because of the way exhibition centres are laid out, there is the natural tendency for clusters to be formed based on people’s location. These clusters will generally be either specific stands where people congregate or areas of the exhibition centre relative to the interest of a certain section of the visiting population (e.g. car technology for male visitors and beauty products for female visitors – these can be subdivided into further segments).

There are two options for bootstrapping: a cluster-head can be naturally assumed for a given cluster (in the scenario above, this can be the exhibition manager of a particular stand, a large trustworthy client or even exhibition centre hosts), or an election can take place in order to generate a cluster head. In that case, a bootstrapping period is agreed between the nodes that found the initial ad hoc network and at the end of the period, the node with the highest reputation is the cluster head. While this may attract collusion between nodes in a cluster and possible malevolence, the resources and time involved may not make this a viable proposition for a would-be rogue node(s).

The operation of the cluster is as follows: every time a recommendation is received by the cluster-head regarding a member node, the cluster-head computes a new reputation and stores it on its database. This database is then circulated to cluster members and other cluster-heads within the network periodically. This ensures that nodes are aware of the most reputable members of the network at all times. The algorithms involved in the process are as follows.

Initially, when all nodes are deemed to be equal, the reputation for a given node $X$ is given by:

$$R_X = \Sigma(T_{XY})/\delta,$$  \hspace{1cm} (1)

where $Y$ is any node which produces a recommendation for $X$. $\delta$ is the number of nodes that provided a recommendation. This statistical average provides the reputation for $X$. This computation is done by the cluster-head which collects recommendations $T$ from all member nodes.
nodes. $T$ itself is calculated at node level based on the observation of the success or failure of the specific action detailed above.

It should be noted that the maximum value of $T$ is rounded off at 1 and the minimum value is rounded off at zero. This means that any trust value that happens to be $\geq 1$ is then rebased to 1 just like any trust value that is $\leq 0$ is rebased to 0.

Further to the above algorithm for reputation as the mean of recommendations at the initial stage, a new algorithm is then applied for subsequent reputation calculations. This is because in order to make the model fair and more resilient, a higher importance is given to nodes that themselves have a higher reputation value. This means that the reputation is now a weighted average of the recommendations provided to the cluster head.

$$R_X = \left[ \Sigma (T_{XY1} \times I_{Y1} + T_{XY2} \times I_{Y2} + \ldots + T_{XYn} \times I_{Yn}) \right] / \beta \quad (2)$$

where $I_Y$ is the importance of $Y$ and is simply defined as the actual reputation of $Y$ at that point in time and $\beta$ is the sum of all importances from $Y_1$ to $Y_n$.

A few rule-based defence mechanisms are also built into FRANTIC for this scenario in order to guard against malevolent and selfish nodes. The following rules are implemented by the cluster-head (the constants can always be modified to make the framework more lenient or harsher towards misbehaving nodes).

When the reputation of a node falls below 0.25. This is the set threshold for early corrective action. It has arbitrarily been set at 0.25 as this level is mid-way between a node that is completely untrustworthy (0) and one that is neither trustworthy nor untrustworthy (0.5). What failing to stay at or over a reputation of 0.25 involves in FRANTIC is a temporary ban for the node. In practical terms, this means that the node_id of said node is dropped from the reputation table that is sent out in reply to member nodes. The effect this has is that this node will then be barred from using the network as it will not be considered as being part of the cluster and therefore part of the network. This is a temporary ban from the cluster head and it only lasts for a set period of $n$ seconds. After the ban, the reputation assigned to the node is the one which it had prior to its reputation falling below 0.25. If this is not available, then the new reputation is set as 0.25 and the reputation broadcast, with the node being given the chance to redeem itself by contributing to the benefit of the network.

When the reputation of a node falls below 0.25 for a second time. This time, the node_id is deleted from all tables and the node is therefore ejected from the network and its node_id is broadcast to all adjacent cluster heads to be added to a special table for barred nodes.

When the reputation of a node falls below 0.1 within defined time period ‘$n$’. This is very rare and either depicts a very selfish node or a node that has been dropping packets voluntarily out of malice. The same procedure as point 2 is employed, with no chance for the node to redeem itself.

When a node issues false recommendations, whether highly favourable or highly negative for another node. In order to cater for such eventualities, the cluster head keeps a record of the variation, $\lambda$, of actual trust values from individual member nodes from overall reputation.

The standard deviation is also calculated as $\sigma$, using the following equation:

$$\sigma = \sqrt{\left( \frac{1}{N} \right) \left( \lambda_{X1}^2 + \lambda_{X2}^2 + \ldots + \lambda_{Xn}^2 \right)} \quad (3)$$

where $N$ is the number of recommendations obtained for $X$ (in this case: 5) and sqrt. denotes the square root and A to E are individual member nodes.

If any $\lambda_X$ value is found to be more than 1 standard deviation away from the mean value $R_X$, then the node is marked on the blacklist table. This is valid for all recommendations which that particular node may make. If the difference in value exceeds the standard deviation again for a second recommendation, then the node is barred from the network.

In order to ascertain that the reputation calculation and distribution worked, it was necessary to simulate it within a network environment. Furthermore, the relevance of the application of the rules in order to mitigate non-cooperation and malicious behaviour needed to be investigated.

IV. EXPERIMENTS

The experiments were set up in ns-2 [18], a discrete event simulator. It provides support for simulating routing protocols over wired and wireless networks. It was used in the experiments to randomise the routing of information within the clusters. Ns-2 is a low-level simulator typically used at the network layer of the OSI model. However, its choice was appropriate in this case as it provided a controlled environment in which the routing could be observed alongside the application layer of a trust framework. The objectives of the experiments were to determine whether:

1. the application of the framework resulted in the generation of trust and reputation information, based on a specific targeted action such as the successful delivery of information (either routing information or distributed content)
2. the trust framework introduced appreciable overhead into the system and the trade-off in performance was matched by the increased reliability and robustness. It should be noted that the trust management process itself does not introduce appreciable computing overhead for the node itself, given that such computation is done at the cluster head. The overhead is mostly in the distribution of the reputation via recommendations and cyclic updates from the cluster head.
3. the framework was resilient to instances of non-cooperation and malicious behaviour.

The experiments were to model the distribution of information at the selected exhibition centre scenario based on the cluster mechanism described previously. The
distribution of information was mimicked by the transfer of data packets. These operated as an abstraction for the real information.

Objective 1 was met by performing experiments on a single cluster only in order to observe the reputation mechanism at work. Packets were routed through nodes organised in a clustered formation. A 6-node configuration was used per cluster. This was an arbitrary selection – determining the ideal cluster size is the subject of further work. Initially, a single cluster was used in order to verify the framework was able to perform trust maintenance whereby nodes were able to be dropped by the network for poor performance and then allowed to redeem themselves. The trust information was generated by all nodes and measured and calculated at the end of 100s cycles and plotted.

The following graph depicts how the reputation of a single node is viewed by its surroundings. A plot of reputation versus time shows the evolution of the trust information based on the action of the node.

![Graph showing the reputation of node 1 against time](image)

Figure 2. A plot of the reputation of node 1 against time

In this experiment, the node is initially made to behave as a source, router and sink in order to make sure it is working properly and at the 5th cycle is made to start dropping packets in order to lower its reputation. This happens up until the point the node passes the 0.25 threshold value for the cluster head, at which point it receives a ban. As per the model, this is only a temporary ban that lasts for only one cycle. When the node is reintegrated into the network, its value is reset to 0.25. In its first cycle after being reintegrated, this node experiences no traffic. However, this may not necessarily mean that it is a sign of non-cooperation. It could well be because of its low reputation value, other nodes may not be keen on using it as a router. This is why in this special case, the cluster head does not apply the decay principle as the node has only come back from a ban. Besides, in order to apply a trust decay decrease, the cluster head would require the difference in reputations from the previous interaction which it does not have.

The next two cycles allow the node to redeem itself by increasing its workload and therefore increasing its own reputation. The last two cycles however, it does not experience any traffic and this time the cluster head ages its reputation by factoring in the decay principle and its reputation value is seen to fall.

In order to meet objectives 2 and 3, a second set of experiments was laid out which involved multiple clusters and the following variables were adopted:

- Simulation time: 1200s, Cycle time: 100s
- Number of nodes: 42 (7 clusters based on prevalent configuration)
- Simulation area: 1000m x 1000m, Pause time: 100s
- Movement: Random Waypoint Model, Nodal range: 250m
- Capacity: 2Mbps, Application: Constant Bit Rate (CBR), Speed: 10 m/s

The simulation was carried out as required above for 1200s with varying loads of traffic through various cycles. The graph below pertains to the traffic through one node only within a cluster. Data was analysed at the end of each cycle as opposed to on the fly.

![Graph showing the overheads generated at a single node](image)

Figure 3. The overheads generated at a single node

As far as the node is concerned, it can be seen that the overhead added by the trust mode by virtue of its reputation mechanism is not appreciably higher than the routing overhead.

In comparison, the following graph gives an indication of the routing vs. reputation overhead for the cluster head. As it can be seen, the reputation overhead in this case highly outweighs the routing overhead. This may seem like a gross disadvantage of being a cluster head. However, in order to set things in perspective, the average reputation overhead of all the member nodes is then plotted in comparison. The overall reputation overhead of the cluster head is lower than the overhead of its member nodes. This is to be expected as the cluster head has other duties to perform and does not involve itself in routing, other than to other cluster heads.

In order to simulate threat scenarios, the same simulation set-up is maintained with the difference being that a variable pause time is set and nodes are allowed to move according
to the random waypoint model i.e. at a constant speed determined randomly using a uniform distribution between 10 to 20 m/s.

A varying number of malicious nodes will be implemented which drop most of the data packets they are sent, typically anywhere between 80% (lower band) to 100% (higher band). This is so as not to make detection of the malicious nodes too obvious. If the malicious nodes drop all packets they are sent (routing and data packets included), then they will be weeded out of the network by the cluster head very quickly as their reputation will rapidly fall below threshold levels within a cycle or two.

![Overhead at Cluster](image)

Figure 4. Comparing overheads

![Malicious Nodes](image)

Figure 5. A trust audit: banning malicious nodes

Figure 5 shows that it takes roughly about 6 cycles for all nodes to be properly eliminated from the network. It should be noted that it has been assumed that there is no collusion between nodes and that the malicious nodes do not drop all packets. However, these are extreme situations; malicious nodes will often drop all packets indiscriminately resulting in a quicker ban from the network.

V. CONCLUSION AND FURTHER WORK

It transpires that FRANTIC is very effective in meeting stated objectives, i.e. to generate, distribute and maintain trust information. The end result is having a trust-connected network that is able to determine trustworthiness of its members at any given point in time. This makes it highly attractive for applications such as file sharing and exchange of other information over a reliable platform.

Further work is centred around determining optimal cluster size settings, election processes and other scenarios, including those with gateways to backbone infrastructures. Work is also being done on implementing ways to mitigate collusion. An avenue being pursued involves the use of Byzantine type fault tolerant protocols.

Different scenarios and applications call for a tailoring of the trust framework to fit each application such that there can never be a universal one – an adaptive one can provide sufficient trustworthiness based on the context in which it is required.

REFERENCES