WHAT CAN CONTEXT DO FOR TRUST IN MANETS?

Eymen Ben Bnina, Olivier Camp, Chùng Tiến Nguyễn
Department of Computer Science, ESEO, Angers, France

Hella Kaffel Ben Ayed
Faculty of Science, Tunis, Tunisia

Keywords: Trust models, context, ad hoc networks.

Abstract: The global performance of a mobile ad hoc networks (manet) greatly depends on, both, the cooperation of participating nodes and the environment in which the nodes evolve. The willingness of a node to cooperate can be illustrated by the trust assessed to the node. Yet, existing trust models, designed for reliable wired networks, do not take into consideration possible communication failures between client and server. We believe, that in the case of ad hoc networks such factors should be considered when computing trust. In this article, we show how an interaction can be decomoposed in three separate phases : two communication phases for transporting the request to the server and the response back to the client, and one execution phase which represents the actual execution of the service by the server. We propose to define the communication environment using contextual attributes and to consider this context when assessing trust to a server. We discuss the possible uses of context in the field of trust computation in manets and define contextual attributes that seem important to consider when modelling and computing trust.

1 INTRODUCTION

A distributed system is a system consisting of nodes which cooperate together provide users with services such as web services, data exchange and sharing of software. Such an environment is said to be fully distributed if there is no central component controlling the whole system. Besides centralised distributed systems, in which one or more central servers control the entire system, we can find fully distributed systems in which all nodes are equals and are, together, in charge of controlling the system. **M**obile **a**d hoc **net**works (manets), also called spontaneous networks, are an example of such fully distributed systems.

Manets are IP networks made up of a collection of mobile nodes communicating via radio links. They do not rely on any predefined infrastructure or centralised administration to operate. Nodes in a manet may, at any time, leave, enter or roam within the network. The resulting dynamic nature of the network's topology, along with the unreliability of the wireless links, thus require for manet services to be highly adaptable. For instance, in the case of routing, the lack of a network infrastructure implies that the service is provided in a peer-to-peer fashion and that all the nodes need to act as collaborating routers, to provide multi-hop routes between any source and destination. Moreover, the availability of an individual central node cannot be guaranteed at all times. Therefore, services cannot rely on a point of centralisation and should be provided in a distributed and adaptive manner.

Manets provide an easier way, in comparison with classical, infrastucture-based networks, to aggregate large amounts of resources while maintaining a low system maintenance cost and are thus an interesting solution when setting up dynamic and flexible applications. However, these systems also bring up some new problems. Indeed, in such an environment, when a node needs to use a given service, it may not know with which quality of service each server is able to provide the service. This is particularly true in an open network in which a node does not know to which extent the other nodes are willing to cooperate.

One among the solutions that have been suggested for such problems, is to control whether a node is, or not, authorised to enter the network. This way, it can be considered that the nodes in the network have all the approval of a trusted central entity and may also themselves be trusted; according to a policy defined

Ben Bnina E., Camp O., Tiếń Nguyên C. and Kaffel Ben Ayed H. (2008).
WHAT CAN CONTEXT DO FOR TRUST IN MANETS?.
In *Proceedings of the Tenth International Conference on Enterprise Information Systems - SAIC*, pages 133-141
DOI: 10.5220/0001701001330141

by the trusted central entity. Yet, such limitation on the number of cooperating nodes may greatly degrade the performances of the network. Another solution is to calculate the trust that the client assigns to each server and, based on this, decide whether or not to cooperate with the service provider. Several works have tackled the problem of computational trust and different models have been proposed.

The existing trust models are well adapted to stable and reliable networks, but most are not suitable for the inherently dynamic nature of manets. Indeed, it is no point determining the most trustworthy server for a given service, if the unreliability of the network does not allow communication with this server. For this reason, we believe that the factors influencing the stability of the network should also be considered during the evaluation of trust.

In this paper, we present a brief review of some existing trust models and discuss how researchers have proposed different approaches for computing trust. We stress the fact that most models do not explicitely take situational information into account when computing trust and explain why this is penalising in unreliable and dynamic contexts such as manets. Thus, we discuss the notion of context and present how it can be used in manets during trust computation.

The remain of this paper is organised as follows: section 2 gives a brief survey of some existing trust models, and points out approaches proposed by researchers for computing trust. Section 3 presents the reasons for which, in our opinion, context should be considered when computing trust in manets and gives an overview of existing definitions for context. Section 4 gives our perception of context in manets and proposes a set of contextual attributes that, we think, should be considered when studying the result of past interactions when computing trust. We conclude in section 5 and present our future works.

2 BACKGROUND

Trust is a concept which is frequently used in our daily lives. In fact, it often guides the social interactions with other individuals. Trust has been studied in various fields (philosophy, economics, sociology, psychology and, more recently, computer science) thus leading to the existence of several definitions (Duma et al., 2005)(Grandison, 2003) and a lack of coherence among researchers. (McKnight and Chervany., 1996). For us, the most appropriate definition is that proposed by Gambetta in (Gambetta, 1988): "Trust (or symmetrically, distrust) is a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both before he can monitor such action (or independently of his capacity ever to be able to monitor it) and in a context in which it affects its own action". This definition stresses the facts that: trust is uncertain, trust is subjective and trust depends on context.

Several trust models have been described in literature. The differences between them mainly concern : trust modeling, trust management and decision making. While trust modeling deals with the representational and computational aspects of trust values, trust management focuses on the collection of evidence and risk evaluation.Concerning decision making it is actually a part of trust management. In this survey, we will concentrate only on trust modeling.

The model defined in (Marsh, 1994) is based only on direct experiences between trustor and trustee. In this model, the author introduces three types of trust :

- **Dispositional Trust** T_x refers to the trust of an agent x regardless of the possible cooperation partner and the situation.
- **General Trust** $T_x(y)$ describes the trust of x on y regardless of the situation.
- **Situational Trust** $T_x(y,\beta)$ describes the trust of agent x on agent y in situation β. The values of trust belong to $[-1,1[$. These trust values illustrate the fact that full distrust exists but not total trust.

Abdul-Rahman and Hailes (Abdul-Rahman and Hailes, 2000) present a trust model based on the sociological characteristics of trust. Their model supports the following properties of social trust: **a)** Trust is context-dependent. **b)** Trust supports negative and positive degrees of belief of an agent's trustworthiness. **c)** Trust is based on prior experiences. Agents are able to identify repeated experiences with similar contexts and with the same agents. **d)** Reputational information is exchanged between agents through recommendations. **e)** Trust is not transitive - all evaluations of recommendations take into account the source of the recommendation. **f)** Trust is subjective -different agents may have different perceptions of the same agent's trustworthiness. **g)** Trust is dynamic and non-monotonic -further experiences and recommendations increase or decrease the level of trust in another agent. **h)** Only Interpersonal Trust (the trust one agent has in another agent directly in a specific context) is supported.

This model uses direct trust and recommendor trust for computing the global trust value assigned by an agent to the trustworthiness of another. The direct trust that a given trustor agent assigns to another

trustee agent *A*, relatively to a given context *C*, is represented by $t(A,c,td)$ where td is the degree of trust assigned to agent *A* and may take one of the following four values: "very trustworthy", "trustworthy", "untrustworthy", "very untrustworthy". In this model, recommendor trust represents the belief of agent *A* concerning the fact that agent *B* is trustworthy, to a certain degree, for giving recommendations concerning other agents relatively to a given context *c*. Recommendor trust is represented by $tr(B,c,rtd)$ where rtd is a semantic distance between *A*'s perception and the recommendations it received.

In order to compute the trust that an agent *A* assigns to another agent *B*, the authors in (Castelfranchi and Falcone, 1998) propose a trust model based on a cognitive approach. They assert that the reasons that make *A* ask for a service from *B* result from a set of mental beliefs. For that, their model is based on the following three beliefs: **1)** Competence belief: *A* believes that *B* can actually do the task; **2)**Dependence belief: *A* believes that *B* is the necessary or the best agent to rely on for doing this task; **3)**Disposition belief: *A* believes that *B* actually will do the task. The latter belief is articulated by two other beliefs which are: **i)**Willingness belief: *A* believes that *B* has decided and will perform action α that is related to the task**; ii)**Persistence belief: *A* believes that *B* is stable in its intentions of doing α.

Thus, trust in this model is represented as a set of mental attitudes which allow agent *A* to believe that another agent *B* will respond to its requests.

The authors in (Yu and Singh, 2001) present a model which doesn't combine information relative to direct and indirect interactions. Recommendations are only used if an agent has never interacted directly with the agent for which trust is computed. An example of such a situation would be if the trustee has only recently joined the network. Each node stores the information concerning direct interactions as a set of values that reflect the quality of these interactions. This model only considers the most recent experiences and defines an upper and lower thresholds that define the limit between what are considered QoS ascribed to trustworthy agents, QoS with no clear classification and QoS ascribed to non trustworthy agents. By applying the Dempster-Shafer theory of evidence on the stored information an agent is able to compute the probability with which the trustee belongs to one of the above three groups.

In (Nguyen and Camp, 2007), the authors propose a model that uses, both, direct experiences and recommendations. They are given as input to a Bayesian network, and the computing of trust values is based on a probabilistic approach using Bayes' theorem of conditional probability. This model considers that the future behavior of an agent depends on its past behaviors and bayesian networks are very efficient for manipulating the associated conditional probabilities.

In this model, the trust assigned by agent *A* to a server *S* for providing service $S*$ with a quality q is defined as the probability that *S* will provide $S*$ to *A* with the given quality q and is calculated considering the results of the past interactions with service $S*$ provided by *S*.

In this section, we have reviewed some of the existing trust models and pointed out approaches that researchers rely on for computing trust. However these models do not use situational information which we believe very important in dynamic environments. In the following section, we will discuss why contextual information, specifically in the case of manets, is important when computing trust. We will also show how such information may change the trust an agent has in a server for completing a given task.

3 WHY IS CONTEXT IMPORTANT FOR TRUST IN MANETS ?

For their computing needs, agents in a manet often have to rely on services provided by others. The fully distributed nature of a manet does not allow for a trusted central entity to manage these interactions and this thus has to be taken in charge of by each agent. The server providing the required service may be out of the client's emission range. In such a case the client must rely on intermediate nodes to transmit its request, and the server should also use a hop by hop approach to reply to the client. For the client to consider its interaction successful, the following three steps must be accomplished successfully : The request reaches the server, the server executes the request, the server's response reaches the client. This is represented in figure 1.

On the contrary, if the interaction fails, client *C* cannot determine which one of the steps, detailed

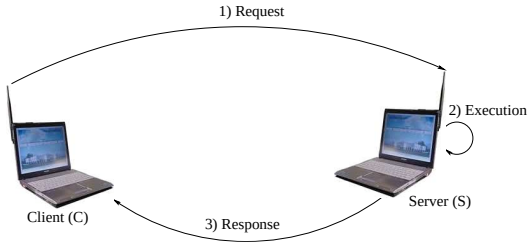

Figure 1: Interaction between agents in a manet.

above, has failed. Moreover, *C* does not have a global vision of the network and thus does not know the route taken by its request, nor does it know the intermediate nodes that helped in the transmission of the response. In fact, the only information the client holds concerning the routes to and from the server are those found in its own routing table - ie ; the first hop and the number of hops to destination. This, of course is only true if we do not search for such information in the implementation dependent data manipulated by the routing protocol. We choose to only consider the routing table to stay independent from any specific routing protocol.

According to Gambetta's definition of trust, the trust *C* has that it will obtain a reply to its request to *S* depends on: the trust it has that the request will reach *S*, the trust it has that *S* will correctly execute the service and the trust it has that the reply will be transmitted back.

Most of the existing trust models only concentrate on the value of trust assigned to *S* in properly executing the service and choose a collaborator among the servers with the highest such value of trust. We believe that such an approach is insufficient, in the case of manets, and that the correct transmission of request and response should also be considered. However, due to the lack of information available concerning the exact participants and their behaviours in both these transmissions, we consider that an experience/recommendation based trust model is not adequate for the routing service. Instead, we prefer to consider that the proper transport of request and response both depend on the environment through which those messages are exchanged and propose to consider context relevant information when computing trust and deciding of a service provider.

Many definitions have been given for the concept of context. While in (Salber et al., 1999) authors qualify context as an environment or situation, many researchers use the definition given in (Abowd et al., 1999): "Any information that can be used to characterise the situation of entities (i.e whether a person, place or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves". In (Schilit et al., 1994), the authors propose to organise the concept of context in three categories:

- **User Context:** user profile, location, people nearby, social situation, activity, health conditions, agenda settings, etc.;
- **Execution Context:** network traffic, status of the device, availability of resources, communication costs, nearby resources, etc.;
- **Environment Context:** weather, light, noise level, temperature, time, etc.

This categorisation fits well with our vision of an interaction in a manet. Indeed, the routing context influencing the data exchanges between client and server clearly belongs to the environment context. Also, the execution context reflects situational factors that may influence the actual execution of the service by the server. As for user context, we will see that it can be used to describe the user profiles of both partners of the interaction.

The trust models presented in (Castelfranchi and Falcone, 1998), (Marsh, 1994), (Yu and Singh, 2001) and (Nguyen and Camp, 2007) are not context aware and compute trust values independently from context. Concerning the model presented in (Abdul-Rahman and Hailes, 2000), authors consider trust in the presence of virtual communities and leave context management open to let developers define their own context. In our opinion, contextual information should be recorded together with every experience and considered during the trust computation phase. In this work, we point out the contextual informations that, in our opinion, should be considered and discuss on how the context can be used when computing trust in ad hoc networks.

This section has presented the main reason for which we consider a trust model for manets should be context-aware and has briefly presented works defining the concept of context. We must now isolate the contextual attributes that may be interesting when considering the mobile, unstable and open environment of manets.

4 WHICH CONTEXT IS IMPORTANT FOR TRUST IN MANETS ?

Even though, to the best of our knowledge, few trust models take into account the routing context (Toivonen et al., 2006), some consider basic contextual information when collecting experiences (direct or contained in recommendations). For instance, the model proposed in (Nguyen and Camp, 2007) records the time at which each experience occurs and gives less consideration to older interactions than it does to more recent ones. We believe this information is particularly relevant in the case of a manet in which agents can have varying efficiencies according to their battery power or even suddenly disappear from the network.

Another contextual attribute, usually implicitely,

taken in charge by some models is the number of interactions that have occurred between client and server when trust in the server is calculated. Indeed, the trust values computed by probability based models gain precision as the number of interactions with the server increases. Now, the number of interactions could be considered as an information relative to the execution of a service by a given server and thus be part of the execution context; This is a first step towards context awareness.

Even though the above two contextual attributes seem important, they have no influence on the transmission of messages between client and server. We should now decide which contextual attributes are relevant to these steps of the interaction.

The routing context associated to an experience should contain representations for the factors that may influence the quality of exchanges between client and server.

4.1 Hop Count

Works presented in (Hekmat and Mieghem, 2003) show that the probability of success of a communication between agents is highly dependent on the distance, in terms of hop count, between the participants; the higher the hop count, the lower the probability of success. For this reason we consider the hop count between client and server to be a crucial information for caracterising the context of an interaction. This information is freely available in the routing table, for any node with which a communication is possible. However, the client can only retrieve, the distance to the server, from its routing table; It cannot determine the length of the return trip taken by the server's response. Indeed, existing routing algorithms, do not necessarily use the same route from one node to another, as they do than they do when returning. Nevertheless, we can assume that the return trip is similar, in terms of number of hops and therefore that the length of the route from the client to the server also characterises the travelling of the response.

From our point of view, hop count between client and server is the first contextual information to be considered in manets.

4.2 Mobility

The performance in terms of communication, of a manet is closely related to the capacity of its routing algorithm to adapt to the mobility of the nodes. Nevertheless, however efficient routing is in dealing with mobility, communication between very mobile nodes, or through a very dynamic network will be less reliable than between stationnary nodes, or in a stable environment. Finding out a metric for mobility in manets is a challenge. There exists few researchs that have focused on such a topic. In (Boleng et al., 2002), the authors have used link stability as a metric for mobility to show that mobility has a direct effect on end to end delay and data packet delivery ratio. Authors in (Ghassemian et al., 2005) have used a similar metric, deduced from the frequency of link state change and link connectivity duration. It thus seems natural to include information concerning mobility in the routing context. Moreover, intuitively, it seems to us generally more efficient for an agent to choose the most stable server in order to have the greatest probability that its requets reach the destination server.

What is mobility, how can it be defined and measured? Is it useful to consider the mobility of the entire network or should we only study that of the path between client and server? All these questions should be answered before considering mobility as a part of the routing context.

Mobility in ad hoc networks is a topic of research that has been considered through various angles : the specification of mobility aware routing protocols, the effect of mobility on the performances of routing protocols, the definition of mobility models for simulation, the definition of metrics to measure the mobility of a given simulated network. Here, our aim is, for any node of an operating network, to be able to determine both the mobility of any other node and the general mobility of the network.

This, of course, should be done using information held by each node concerning the others ; namely the routing table.

We can generally define mobility as the behavior of an object (entity, person, thing, etc) that has a changing position over time. However, in our case, it is important to point out that, what we refer to as mobility is in fact relative mobility. If two nodes move in the same direction and with equal speed, they can both be considered stable with respect to one another. However, if the other agents in the network are static then the two nodes have very high mobility relatively to the rest of the network. Thus, two measures of mobility may be considered: individual mobility of nodes and global mobility of the network. In our case, global mobility of the network should be considered because it gives an idea on the mobility of the surrounding environment. This general measure, however, does not reflect the individual mobility of each node because it is an average measure and thus stable nodes can not be discovered using it. Thus, in our opinion, it is also useful to consider the individual mobility of nodes. In such a case, it will be easier for

a client agent, given a set of servers with different individual mobility measure, to choose the one it judges most appropriate to respond to its needs. We propose to use the same metrics to measure both general network and individual node mobilities. In fact, we will consider that the global mobility of the environment is the average of the mobilities of all reachable nodes.

Before, examining the information that may reflect the mobility of a node or of the network as a whole, it should be noted that, since we do not consider that the nodes are equiped with a globally accessible location service (GPS for instance), a node is only able to partially capture its relative mobility with respect to other nodes and the mobility of the path between it and all reachable destination nodes. Moreover, the disappearence (respectively reappearence) of a node from the routing table can be the consequence of its relative mobility. Even though, such a situation may also be the result of the switching off or on of the device, we choose to consider that this information is a hint concerning this particular node's mobility .

To measure mobility of nodes, we choose to consider the following information gathered from the routing table: the appearances and disappearances of nodes in the routing table, the number of hops to all reachable nodes and the first hop to each destination. If we consider the situations depicted in figure 2, in which S_{t_i} represents the position of server S at time t_i ($i \in [0,3]$ and $t_i < t_{i+1}$), we notice that the movement of S between t_0 and t_1 can be read in C's routing table as a change in the distance to S; the movement between t_1 and t_2 is read by C as a change in the first hop of the route to S; and movement between t_2 and t_3 is seen as the disappearance of S from C's routing table.

Yet, this capture is only partial as the available information in the routing table (the existence of the destination node, the hop count and the first hop to destination) is not sufficient to detect all mobile nodes

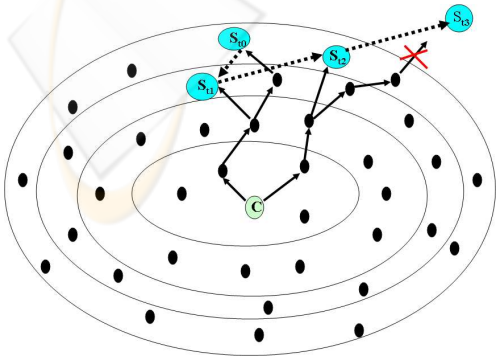

Figure 2: Detected mobility.

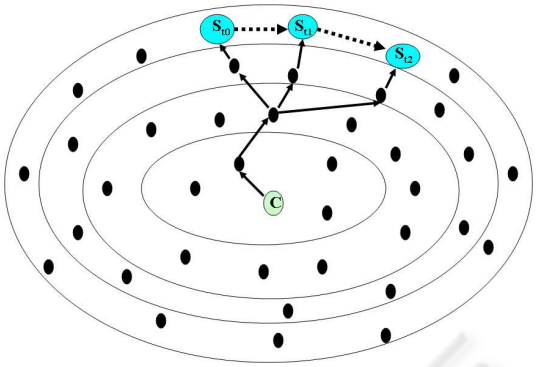

Figure 3: Undetectable mobility.

or routes. In fact, any change in a route, past the first hop, will be unoticeable if it does not modify the hop count. Figure 3 in which S_{t_0}, S_{t_1} and S_{t_2} represent ,respectively, the position of server S at times t_0, t_1 and t_2 ($t_0 < t_1 < t_2$) represents such a situation.

From C's point of view S is always 4 hops away and always has the same first hop to destination.It is thus not considered mobile. For these reasons, it would be useful, when studying mobility, to know all intermediate nodes between C and S. This information is available from the routing table if C and S are neighbours (in this case there are no intermediate nodes between C and S).Otherwise, the exact route can only be discovered by examining the routing tables on the intermediate nodes.

To estimate the mobility of the nodes in the network, we propose to consider the routing table at regular time intervals and, for each reachable destination node, to measure the following :

- Path Stability: The percentage of time during which each entity has been in the routing table since the begining of the time interval,
- Link Stability: The percentage of time during which each entry has been as a neighbour in the routing table since the begining of the time interval,
- Distance Stability: The average number change in hop count to each entry since the begining of the time interval,
- First Hop Stability: The average number of changes in the first hop since the begining of the time interval.

4.3 Density

Density in ad hoc networks is another information that may be considered when choosing an appropriate server with which to cooperate. On the one hand,

if the network is dense, there will be more possibilities to find the best route to each destination. On the other hand, in a scarce network, less potential routes will be available for each destination. The movement of the nodes may have, as an effect, to break links in a route. In such a case, the redundancy of routes in a dense network will often allow to find a new route to reach the destination node. On the contrary, such a movement might result, in interruption of all communications with the destination if node density is not as high.

Mobility and density are thus closely related to one another, and it is clear that some configurations are more favorable than others. For instance, it seems clear that a dense and static network will be more efficient in finding stable routes to destination nodes than a scarce and very mobile network. However, the case of stable and scarce networks and of mobile and dense networks need to be studied with attention.

As it is the case for mobility, different granularities can be given to the measure of density: the global density of the network, composed of all reachable nodes, can be considered or density can be calculated on a per hop basis.

We are now studying the effects of both mobility and density on the efficiency of communications in a manet in order to determine precisely how these contextual attributes should be measured and how they can be considered in trust computation. This will be the subject of a future paper.

4.4 Server and Client Profiles

Number of hops, mobility and density are contextual attributes that are part of the routing context; they influence the quality of the communication between client and server. In the classification described in section 4, we have chosen to consider the routing context as a part of the environment context. We will now discuss how the profiles of both client and server may also be considered as contextual information and included in what is defined by (Schilit et al., 1994) as the user context.

In a distributed system, nodes may be organised in virtual communities (Abdul-Rahman and Hailes, 2000) and prefer to cooperate with nodes in the same community or in partner communities. Our trust model defines two types of communities : *static communities* that reflect an underlying administrative organisation (we could for instance consider a ”student” community, an ”administrative” community, a ”teacher” community and a ”research” community in a university’s network) and *dynamic communities* which each node defines based on the results of the interactions it has had with the others and thus on the trust it has in them in providing a given service with a certain level of quality. Concerning dynamic communities, they can be used to regroup well behaving servers together or, contrarily, misbehaving servers ; for example, a node could dynamically define, both, the community of servers in which it has ”very high” trust for providing service $S*$ and the community of servers in which it has ”very low” trust in providing service $S*$. The trust to put on an interaction with a member of the community is defined by the dynamic community itself. Any node, whether server or client, may thus belong to several communities, and we consider that such information should be considered as profile information that may be useful in predicting the result of an interaction. In the case of static communities a trust policy can be defined to describe inter-community trust relation. The static communities to which belong the server and the client may also be considered as contextual information since they may affect the quality with which the service is provided

4.5 First Hop

The contextual information we have considered up to now in the routing context affect the quality of an interaction, however cooperative the nodes of the network are. In fact, mobility, density and the number of hops to a destination are factors that affect the quality of communication regardless of the willingness of the nodes to properly route traffic. Thus, the contextual attributes presented above do not take into account the presence of selfish or malicious nodes in the network. Such nodes may, by refusing to relay data, have a disastrous effect on communication.

Rather than considering routing as any other service and using the model to compute the trust in routers, we propose to also use situational information to take into account such malicious nodes.

Whenever two agents wish to communicate with one another, unless they are neighbours, they must rely on intermediate nodes and especially on a first hop. This information may seem very partial in the case of long routes. Yet, we believe that it is an important contextual attribute to be added to the routing context.

In this section we have presented contextual attributes that seem interesting to consider when computing trust. These contextual arguments can be organised according to the following taxonomy represented in figure 4. We have also showed in this section that certain context attributes (hop count, mobility and density) helped dealing with the general be-

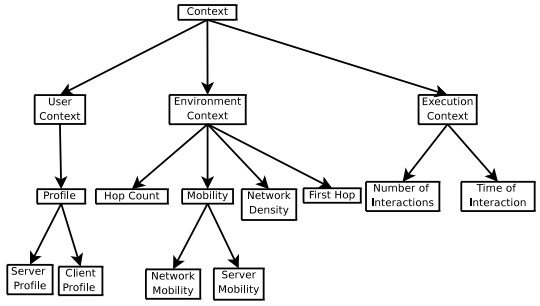

Figure 4: Taxonomy of context.

havior of manets when only in the presence of cooperative nodes, other attributes (profile informations) took advantage of a possible underlying organisation and others stills (hop count) may help in avoiding non cooperative or malicious agents.

Contextual information should be considered both, when recording experiences and during the trust computing process. An experience should thus carry the contextual information that reflects the context in which it has occurred. Moreover, to evaluate the trust we have in a server S that, in a specific context C, we will receive its response to a request, we should transpose the set of all previous experiences with S into context C. Through this transposition, we evaluate, for each experience, the quality it would have had if it had occurred in the current context. The set of transposed experiences can now be used as an input to the trust computing process, to calculate the trust assigned to S in context C.

5 CONCLUSIONS AND FUTURE WORKS

Trust is a concept that must be studied with attention in distributed environments based on the exchange of services between partners. In this paper we have briefly presented some of the models proposed by research for computing trust. These models do not explicitely take contextuals information into account when assessing trust. We have discussed that in the particular case of manets, the context, and particularly the routing context, should be considered in the trust computation process. We have identified a set of contextual attributes that may affect the result of an interaction with a service provider. We propose to use these contextual attibutes together with the other parameters that characterise each experience when assigning trust to a cooperation partner. These attributes may greatly influence the way in which a client agent perceives the quality of a given server. It may thus be more efficient to request a service form a provider with low trust in a favourable context than to one with higher trust in a disadvantegeous context. Yet, before considering the proposed context variables, an adequate metric, for measuring their values should be defined. Even though the metrics used to measure some of the proposed variables seem quite straightforward (this is the case for the number of hops between both partners), other factors are much more tricky to measure and the definition of a metric for these variables is not clear (this is the case for mobility). Moreover, while some variables may be considered individually, others are so closely related to each others that they should be considered together (this is the case for density and mobility). We are now studying the influence of the identified context variables on the quality of communications in order to define metrics for these attributes. A second step will be to study how these contextual attributes should be considered by a context-aware trust model for manets.

REFERENCES

Abdul-Rahman, A. and Hailes, S. (2000). Supporting trust in virtual communities. In *HICSS '00: Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 6*, page 6007, Washington, DC, USA. IEEE Computer Society.

Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M., and Steggles, P. (1999). Towards a better understanding of context and context-awareness. In *HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, pages 304–307, London, UK. Springer-Verlag.

Boleng, J., Navidi, W., and Camp, T. (June 2002.). Metrics to enable adaptive protocols for mobile ad hoc networks. In *in Proceedings of the International Conference on Wireless Networks (ICWN '02), Las Vegas, Nev, USA,*, pages 293–298.

Castelfranchi, C. and Falcone, R. (1998). Principles of trust for mas: Cognitive anatomy, social importance, and quantification. In *Proceeding of the International Conference on Multi-Agent Systems(ICMAS'98)*.

Duma, C., Shahmehri, N., and Caronni., G. (August 2005.). Dynamic trust metrics for peer-to-peer systems. In *In Proc. of 2nd IEEE Workshop on P2P Data Management, Security and Trust*.

Gambetta, D. (1988). *Can We Trust Trust?* Basil Blackwell.

Ghassemian, M., Friderikos, V., and Aghvami, A. H. (September 2005). On mobility metrics applied for ad hoc network protocol evaluation. In *The 7th IFIP International Conference on Mobile and Wireless Communications Networks (MWCN), Marrakech, Morocco*.

Grandison, T. W. (2003). *Trust management for internet applications*. PhD thesis, Departement of computing, University of London.

Hekmat, R. and Mieghem, P. V. (October 2003). Degree ditribution and hopcount in wireless ad hoc networks. In *Proceedings of the 11th IEEE International Conference on Networks (ICON 2003)*, pages 603–609, Sydney, Australia.

Marsh, S. P. (1994). *Formalising Trust as a Computational Concept*. PhD thesis, University of Stirling.

McKnight, D. H. and Chervany., N. L. (1996). The meanings of trust. Technical report, Carlson School of Management,University of Minnesota,United States.

Nguyen, C. T. and Camp, O. (2007). A bayesian network based trust model for improving collaboration in mobile ad hoc networks. *International IEEE Conference on Computer Sciences - RIVF(07)*.

Salber, D., Dey, A. K., and Abowd, G. D. (1999). The context toolkit: aiding the development of context-enabled applications. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 434–441, New York, NY, USA. ACM Press.

Schilit, B., Adams, N., and Want, R. (December 1994). Context-aware computing applications. In *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications*.

Toivonen, S., Lenzini, G., and Uusitalo, I. (2006). Context-aware trust evaluation functions for dynamic reconfigurable systems. In *Proceedings of the WWW'06 Workshop on Models of Trust for the Web (MTW'06)*, Edinburgh, Scotland, UK.

Yu, B. and Singh, M. P. (2001). Towards a probabilistic model of distributed reputation management. In *Proceedings of the Fourth Workshop on Deception, Fraud and Trust in Agent Societies, Montreal, Canada, pp. 125-137*.