ONTOLOGICAL ENGINEERING APPROACH TOWARDS TRUST ORIENTED SECURITY FRAMEWORK FOR ADHOC NETWORKS

Amandeep Verma[1] and Manpreet Singh Gujral[2]

[1]Punjabi University Regional Centre for IT & Mgmt., Mohali, INDIA
[2]University College of Engineering, Punjabi University, Patiala, INDIA

ABSTRACT

Conventionally, user authentication and access control mechanisms would be almost enough, to handle security for stand-alone computers and small networks. Ad hoc networks are illustrated by multi-hop wireless connectivity and recurrently changing network topology which have made them infrastructure less. Adding trust to the existing security infrastructures would improvise the security of these environments. Describing trust relations and their sub-components using ontologies, creates a methodology and mechanism in order to efficiently design and engineer trust networks. This is going to be used as a service for providing trust for ad hoc network at any level i.e. routing, authentication or access control. A trust oriented security framework for adhoc network using ontological engineering approach is proposed by modeling ad hoc network, the OLSR (Optimized Link State Routing) protocol and trust model as OWL (Ontology Web language) ontologies, which are integrated using Jena. In this model, a trustor can calculate its trust about trustee and use the calculated trust values to make decisions depending on the context of the application or interaction about granting or rejecting it. A number of experiments with a possible implementation of suggested framework are performed to make out the characteristics of the trust model and its effect on the ad hoc network operations.

KEYWORDS: *Ad hoc Networks, Ontology, Trust.*

1. INTRODUCTION

A Mobile Ad hoc Network is a network of mobile nodes operating in ad hoc mode. The network infrastructure is dynamically changing, and the links are wireless with less capacity and more prone to errors. This is mainly because of that open environments lack a central control and users in them are not predetermined [5] [6] Adding trust to the existing security infrastructures would enhance the security of these environments.

Ontology [4] defines a common vocabulary for researchers who need to share information in a domain. Each ontology O contains a set of concepts (classes) C and a set of properties P. A class is a collection of individuals and a property is a collection of relationships between individuals (and data). A property that relates an individual to another individual is called objectproperty and a property that maps an individual to a data literal is called datatype property. The ontologies lay the ground rules for modeling a domain by defining the basic terms and relations that make up the vocabulary of topic area. These ground rules serve to guide system builders in fleshing out knowledge bases, building services that operate on knowledge bases, and combining knowledge bases and services to create larger systems. The OWL

DOI : 10.5121/ijcnc.2012.4312

(Ontology Web language), which is recommended by W3C (World Wide Web Consortium), is used to describe the ontologies in present study using Protégé [15] open source ontology editor.

This paper introduced a trust oriented security framework for ad hoc networks by representing the ad hoc network, the OLSR (Optimized Link State Routing) protocol and the trust model as Ontologies. These ontologies are integrated using Jena and the result of the integration is used as framework for the implementation of trust oriented security framework. The paper is intended for the researchers having interest in building the trust oriented ad hoc networks.

The paper is organized as follows. The section 2 presents the review of literature to demonstrate the impact of trust on adhoc network, the usage of ontology for trust in some application areas and the ontologies proposed with reference to ad hoc networks. The section 3 gives the tabular or graphical and textual description of the involved concepts, the properties linking objects representing relationships and the other relevant. The section 4 is about the possible implementation and results of the framework. The section 5 presents the discussions. The section 6 concludes the paper and Section 7 is about the limitations and future work.

2. Review Of Literature

Today, ontologies are finding their way into a wide variety of applications. In addition to the Semantic Web, they are also applied to knowledge management, content and document management, information and model integration, etc. Trust ontologies specifically focus on issues such as types of trust based on how trust is evaluated and how trust ontology can be used for selection based on security and trust [1]. The functional ontology [13] facilitates us to comprehend a functional perceptive system which classifies functional structures of an object from its behavioural and structural model. The review of literature involves the study of the usage and effectiveness of the inclusion of trust in the operation of the ad hoc network. The usages of trust ontologies in various application areas were studied. The combination of ontologies and ad hoc network was also explored.

2.1 Trust in Ad hoc Networks

The effect of trust inclusion in ad hoc network has positive impacts. A number of studies support this viewpoint. A trust evaluation based security solution in ad hoc network [14] where Trust is based on experience statistics. It defends block hole, denial of services, routing table overflow, energy-consummation attacks. In Trust based Adaptive On Demand Adhoc Routing Protocol [7], trust is based upon a node has on its neighbor, different trust level defined and security is applied accordingly. Highly secure, save node’s power and even the time for communication is less. A trusted routing solution [10] in mobile ad hoc networks presents a model for computing, distributing and updating trust and proved very good against colluding malicious nodes

2.2 Trust Ontologies in Literature

A number of trust ontologies have been proposed for a variety of application domains in an effort to organize and formalize trust concepts and relationships. Web Services Trust Ontology (WSTO) – that models the context of a trust-based interaction [9] and enables the participants to describe semantically their trust requirements. A computational trust model based on the ontology structure [5], considering the semantic relations among pervasive elements and especially among trust categories is proposed. A novel semantic service trust organization [11] that uses an ontological approach to model service trust is also proposed.

2.3 Ontologies for Ad hoc Networks

A functional ontology [12] for reputation routing mechanisms base on node behavior is proposed. In this ontology, the functional structures and concepts that compose the reputation routing mechanism are identified. This is all about the reputation based routing decisions. Ontology of MANET (Mobile Ad hoc Network) attributes [3] including device security and performance characteristics can be leveraged to efficiently and effectively make dynamic configuration decisions for managing a MANET was shown.

2.4 Tools for Ontologies

The description of various tools for Ontology like OntoEdit OilED, OntoView, OntoManager and TextToOnto quoted in the survey [2] [8].

3. THE ONTOLOGY

Ontology consists of set of concepts (classes) and set of properties (relationships). The properties can be object properties or data properties. These properties have types and some restrictions. There may be individuals, most specific instances of the class. The concepts, properties and restrictions, individuals of all ontologies are described as follows.

3.1 The Trust Ontology

3.1.1 The Concepts

Classes are the focus of most ontologies. Classes describe concepts in the domain. A class can have subclasses that represent concepts that are more specific than the superclass. A subclass of a class represents a concept that is a "kind of" the concept that the superclass represents. The words concept(s) and class(es) are synonym for this paper, so can be used in the text, interchangeably.

The main class hierarchy, consisting five concepts involved in the said ontology are shown in the Table 1. Thing is abstract superclass for all classes.

Table 1: Main Concepts or Classes

Concepts	Description
*Participants_and_*	The participating nodes and their roles in trust evaluation
Trust_Evaluation_Types	The type, value and influencing factors while evaluation of trust about the desired trust
Operations	The listing of various operations involves in the evaluation of trust
Trust	The compositional structure of the trust
Decision_Makers	The concepts involved in decision making on the basis of trust

The Participants_and_ Roles class is about the description of the concepts that are participants and their roles in the trust evaluation process, is shown in the Figure 1. As it is a network, so the participants are the nodes and their roles may get changed for every other evaluation. The subclasses are Nodes and Role. The subclass Node--This shows the node entity. An active node type at any instance falls into any of the following category – Source_Node, Neighbors or Target_Node. Among these Trustor is a Source_Node, Trustee is Target_Node. The Neighbors are Direct Neighbor, Neighbor of Neighbor or Others. All of these categories are disjoint to each other. The other subclass Role -- This is for the Role of Nodes in current trust evaluation process. Source_Node is TrustEvaluator, Target_Node is TrustSubject and Neighbors are Recommenders.

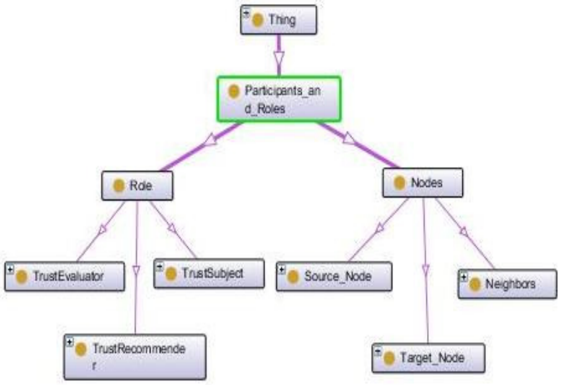


Figure 1: Participants and Roles

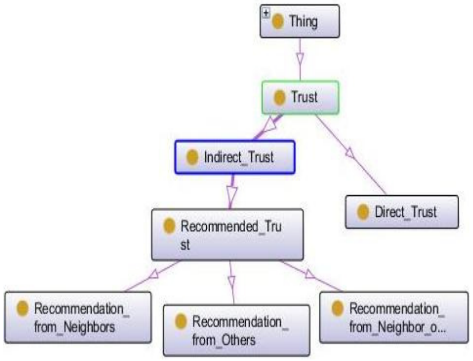

Thing
Trust
Indirect_Trust
Direct_Trust
Recommended_Trust
Recommendation_from_Neighbors
Recommendation_from_Others
Recommendation_from_Neighbor_o...

Figure 2: Trust Class Hierarchy

The compositional structure of Trust is shown in Figure 2. It is a combination of Direct_trust and Indirect_trust. The Indirect_trust is the Recommended_trust. The recommendations are from Direct_neighbors, Neighbors_of_Neighbors and Others. It has Qualitative or Quantitative value depending on the Trust_Value. The Trust_Evaluation_Types class hierarchy which is type, value and influencing factors while evaluation of trust about the desired trust is shown in Figure 3. The first subclass Trust_Type -- This is to specify the type of Recommended_trust. It is of either Global_trust i.e. involve recommendations from all active nodes of the network or Local_trust involves recommendation only from Direct_neighbors. The second subclass Trust_Value -- The value of Trust may be Qualitative or Quantitative. The Qualitative has values – Unknown, No_trust, Low_trust, Normal_trust, High_trust or Highest_Trust. The Quantitative has Minimum and Maximum value as dat properties for possible trust values. The third subclass RecommendedTrustCombinationType --- is for the parameters that to considers while combining the recommendations. If recommendations from any Node have the same effect then as per AllNeighborsSame all recommendations have the same WeightLevels i.e. another subclass under this category. To give different WeightLevels to Direct_Neighbors, Neighbor_of_Neighbor and Others-- DifferentLevels type is used.

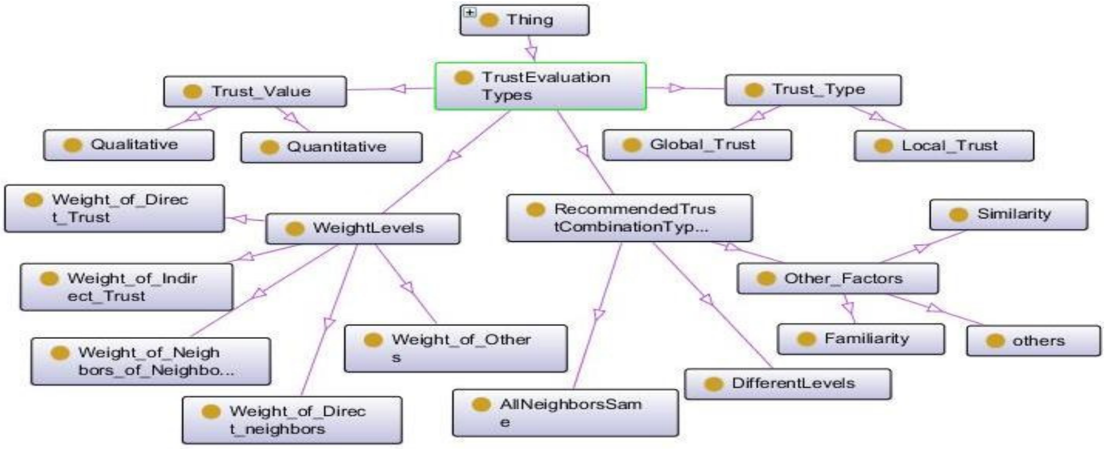


Figure 3: Trust Evaluation Types and their subclasses

The Operations class, listing of various operations involves in the evaluation of trust, is shown in the Figure-4. The operations are – Context_Determination, Trust_assessment and Trust_combination. The Context_Determination is used to get Context at any instance. The Trust_assessment is used for Mapping and/or Normalization the trust value. The Trust_combination operation consists two parts- one is used to combine recommendations obtained from different type of recommenders i.e. Trust_combination_for_Recommendations in order to get Recommended_Trust which will give the Indirect_trust. These are either recommendation for Global_trust by Trust_combination_for_Global_trust or for the evaluation of Local_trust by Trust_combination_for_Local_Trust. In addition to recommendations the Other_Factors factors like Similarity and/or Familarity or others may be used for effective calculation of Indirect_trust. The other operation for Trust_combination is i.e. Trust_combination_for_Direct_and_Indirect_Trust combines the Direct_trust and Indirect_trust to yield the Trust at that instance.

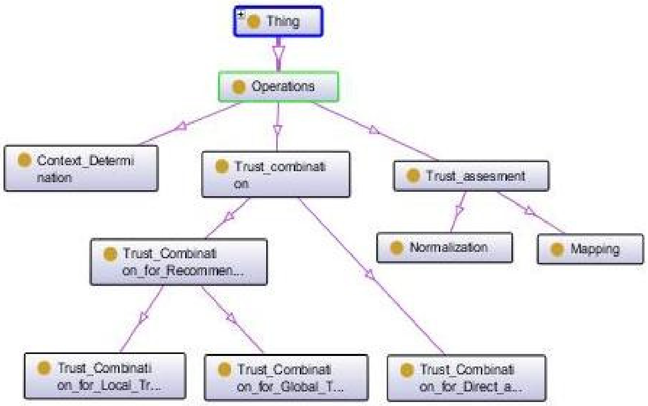

Thing
Operations
Context_Determi nation
Trust_combinati on
Trust_assesment
Normalization
Mapping
Trust_Combinati on_for_Recommen...
Trust_Combinati on_for_Local_Tr...
Trust_Combinati on_for_Global_T...
Trust_Combinati on_for_Direct_a...

Figure 4: Operations for Trust Evaluation

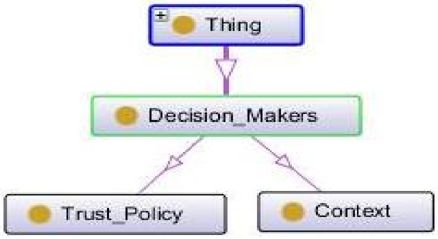
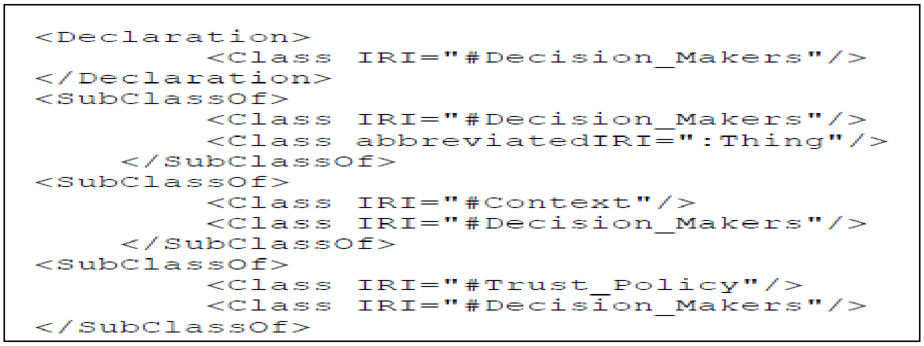


Figure 5: Decision Making Classes

The Decision_Makers involves the Trust_Policy and Context concepts shown in Figure 5. The context --- This refers the possible Context---No_Risk, Low_Risk, Moderate_Risk, High_Risky and Highets_Risk, that an application has to execute in the environment. The trust policies are defined for every possible value of the Context. A code segment to represent the concepts implementation is shown in Figure 6.



```
<Declaration>
        <Class IRI="#Decision_Makers"/>
</Declaration>
<SubClassOf>
        <Class IRI="#Decision_Makers"/>
        <Class abbreviatedIRI=":Thing"/>
    </SubClassOf>
<SubClassOf>
        <Class IRI="#Context"/>
        <Class IRI="#Decision_Makers"/>
    </SubClassOf>
<SubClassOf>
        <Class IRI="#Trust_Policy"/>
        <Class IRI="#Decision_Makers"/>
</SubClassOf>
```

Figure 6: Code Segment of Class declarations

3.1.2 The Properties

OWL properties represent relationships. These relationships are binary in nature. Properties link individuals from the domain to individuals from the range. In other words, they describe relationships between an individual and data values.

The Object Properties

To model the trust relations, for both direct trust and recommendation trust, some properties must be defined in the ontology. The Table 2 shows the core object properties. The object properties, in essence, are the relationships among concepts. Each object property may have a corresponding inverse property. If some property links individual 'a' to individual 'b' then its inverse property will link individual 'b' to individual 'a'.

Table 2: Object Properties

Object Property	**Description**
hasValues	This is about the concepts related with other concepts , in terms of concepts having values as other concepts
queryTrust	It is the initiative by the *Trustor* to query trust about the *Trustee*
trustEvaluation	This has the various relationships of the concepts involved in the process of trust evaluation
assesingTrust	This is the depiction of the ways how concepts interacts for assessment of the trust from the value obtained from trust evaluation
policyMaking	The relationships of the concepts involved in policy making and making decision on the basis of trust value and context of the interaction.

The narrative of object properties in terms of their Domain and Range, which are the concepts, related by that property is as follows. The Most of the properties are irreflexive, asymmetric and functional in nature. The object property hasValues has subproperties, narrates its domain and range shown in the Table-3.

Table 3: hasValue Object Property

Sub Property	**Description**
hasRole	*Nodes* to *Role.* The inverse property is isRole. This depicts the Role of each node in the network for trust evaluation.
hasRecommendations	It is from *Nodes* to *Recommended_Trust*. This indicates that a node has recommendations for *Trustor* about the *Trustee*.
hasComponents	It is from *Trust*. It represents the components of the *Trust*
hasTrustValue	From *Trustor* to *Trust.*
hasValueOf	*Source_Node* to *Direct_Trust*. The direct trust a *Trustor* has on *Trustee*

The subproperty hasRole is from Nodes to Role, to link each participating nodes in trust evaluation process with their role. The further subproperties ---hasRoleEvaluator is from Source_node to TrustEvaluator, hasRoleRecommender is from Neighbors to TrustRecommender, hasRoleSubject is from Target_Node to TrustSubject. The subproperty hasRecommendations has three sub properties i.e. recommendations from Direct_neighbors, Neighbors_of_Neighbors and Others The other subproperty hasComponents is with two subproperties – hasDirectTrust and hasIndirectTrust, the two components of the Trust. The subproperties of the object property trustEvaluation are briefed in the Table 4. The domain and range of each of the property, the inverse property, if it exists, is expressed.

Table 4: trustEvaluation Object Property

Sub Property	**Description**
sendRecommendation	*Recommended_Trust* to *Trust_combination_for_recommendation*.
sendTrust	sending the values of *Direct_Trust* and *Indirect_Trust* for trust combination
needValueOf	*RecommendedTrustCombinationType* to*WeightLevels.*
canUseValueOf	*Trust_combination* to *Other_Factors*. The inverse property is *canUsedBy*
useWeights	*Trust_combination_for_Recommendation* to *RecommendedTrustCombinationType*

The subproperty sendRecommendation has further subproperties--- sendDirectNeighborRecom is from Recommendation_from_Neighbors, the other subproperty is sendNeighborofNeighborRecom from Recommendation_from_Neighbor_of_Neighbor, and sendOthersRecom is from Recommendation_from_Others. The other subproperty – sendLocalRecommendation is from Recommendation_from_Neighbors to Trust_combination_for_Local_Trust. The sendTrust property is with two sub properties. The sendDirectTrust and sendIndirectTrust are for sending the values of Direct_Trust and Indirect_Ttrust to Trust_combination operation. The needValueOf has two subproperties ---- needDifferentValueOf is from DifferentLevels to Weight_of_Direct_neighbors, Weight_of_Neighbors_of_Neighbor and Weight_of_Others. The property assesingTrust consist subproperties involved in the assesment of trust either in qualitative or quantitative manner as per the requirement shown in Table 5.

Table 5: assesingTrust Object Property

SubProperty	**Description**
isMapping	It is from *Mapping* to *Qualitative*. Functional in behavior.
isNormalized	It is from *Normalization* to *Quantitative*.
givesValueOf	*Trust_assesment* to *Trust_Value.*
givesValueFor	*Trust_assesment* to *Trust.* The inverse property isGivenBy
sendValueFor	*Trust_combination* to *Trust_assesment*.

The is Mapping property maps the trust value obtained from trust combination to a qualitative value like Low Trust or High Trust as per the possible individuals defined for the Qualitative class. The property isNormalized is to normalize the value obtained from trust combination, so that it should be in defined range for that. This normalized value can be used directly if the trust is required in quantitative form or mapped to qualitative value. The subproperty givesValueOf has sub-property is givesQualValue is from Mapping to Qualitative. The other sub-property is givesQuantValue is from Normalization to Quantitative. The givesValueFor property establishes the fact that the value of Trust is obtained from Trust_assesment. The sendValueFor property is to show the linking between Trust_combination and Trust_assesment for passing the value from it to other for assessment. The policyMaking property has sub properties shown in Table 6 to depict the relationship among concepts to make the decision on the basis of the Context of the application, Trust value obtained and the Trust_Policy defined for the Context on the basis of Trust value.

Table 6: policyMaking Object Property

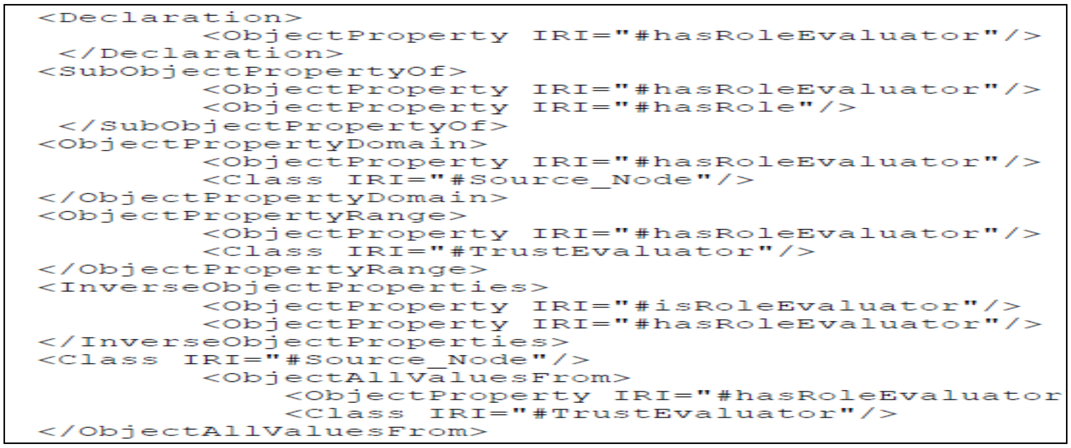
SubProperty	**Description**
isConsulting	*Trustor* to Trustor_Policy. The inverse property is consultedBy
isEffectedBy	*Trust_Policy* to *Context* The inverse property is hasEffect.
isValueOfContext	*Context* to *Context_Determination.* The inverse property is givesValueOfContext
useValueOf	*Trust_Policy* to *Trust*.

The property isConsulting is to show the relationship that Trustor consults the Trust_Policy in order to make out a decision. The property isEffectedBy is to demonstrate the fact that the Trust_Policy is effected by the Context i.e. every Context has different policy. The property isValueOfContext is to show the possible individuals as values for the output of Context_Determination operation. The last property useValueOf is to establish the relationship that Trust_Policy use the value of Trust in the process of making decision.

Table 7: Restrictions on Object Properties

Restricted Class	**Restricted Property with Value (Class)**
Neighbors	hasRoleRecommender only *TrustRecommender*
Source_Node	hasRoleEvaluator only *TrustEvaluator*
Trustor	hasTrustValue only *Trust*
Target_Node	hasRoleSubject only *TrustSubject*
Context_Determination	givesValueOfContext only *Context*, givesValueOfContext some *Context*
Trust_Policy	isEffectedBy some *Context*

The object properties listed above has restrictions. Some of these restrictions are listed in Table 7. The 'only' is a restriction, implies that the only possible value, it is known as universal restriction. The restriction 'some' means at least one value of that type, it is an existential restriction. A code segment to exemplify the implementation of object properties is shown in Figure 7.



```
<Declaration>
        <ObjectProperty IRI="#hasRoleEvaluator"/>
 </Declaration>
<SubObjectPropertyOf>
        <ObjectProperty IRI="#hasRoleEvaluator"/>
        <ObjectProperty IRI="#hasRole"/>
 </SubObjectPropertyOf>
<ObjectPropertyDomain>
        <ObjectProperty IRI="#hasRoleEvaluator"/>
        <Class IRI="#Source_Node"/>
</ObjectPropertyDomain>
<ObjectPropertyRange>
        <ObjectProperty IRI="#hasRoleEvaluator"/>
        <Class IRI="#TrustEvaluator"/>
</ObjectPropertyRange>
<InverseObjectProperties>
        <ObjectProperty IRI="#isRoleEvaluator"/>
        <ObjectProperty IRI="#hasRoleEvaluator"/>
</InverseObjectProperties>
<Class IRI="#Source_Node"/>
        <ObjectAllValuesFrom>
            <ObjectProperty IRI="#hasRoleEvaluator
            <Class IRI="#TrustEvaluator"/>
</ObjectAllValuesFrom>
```

Figure 7: Code Segment of hasRoleEvaluator Object Property

The Data Properties

The data properties used in the ontology--- hasMinValue and hasMaxValue is from Quantitative class to build in data type Integer. This is to specify the range of permissible values for trust. The data property hasTrust to show the type of the value obtained from Trust_combination

3.1.3 The Individuals

Individual instances are the most specific concepts represented in a knowledge base. The Figure 8 shows the individuals belonging to Context, Trust and Trust Policy.

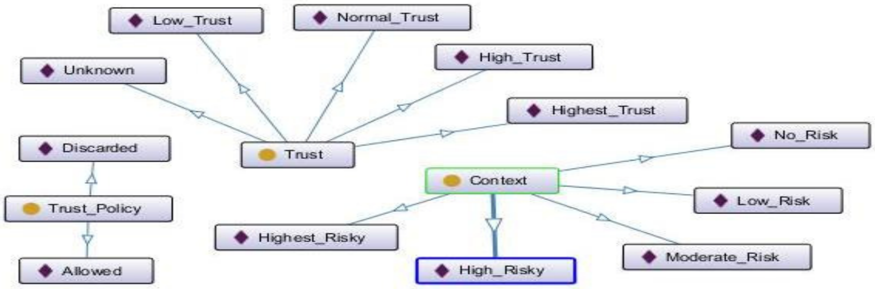
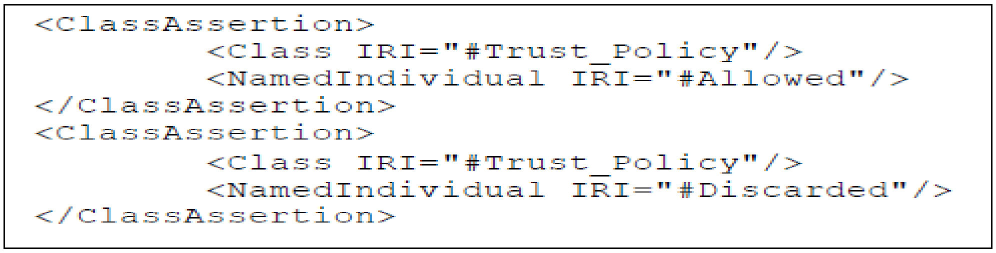


Figure 8: Concepts with their Individuals

The individuals of Qualitative are Unknown, Low Trust, Normal Trust, High Trust and Highest Trust. The possible individuals of Context are No Risk, Low Risk, Moderate Risk, High Risk and Highest Risk. The context value depends on the risk involved from the security point of view of the application in execution. The possible individuals of Trust_Policy are Allowed or Discarded. A possible implementation for Trust_Policy individuals is given in Figure 9.



```
<ClassAssertion>
        <Class IRI="#Trust_Policy"/>
        <NamedIndividual IRI="#Allowed"/>
</ClassAssertion>
<ClassAssertion>
        <Class IRI="#Trust_Policy"/>
        <NamedIndividual IRI="#Discarded"/>
</ClassAssertion>
```

Figure 9: Code Segment for Individual declarations

3.2 The OLSR Ontology

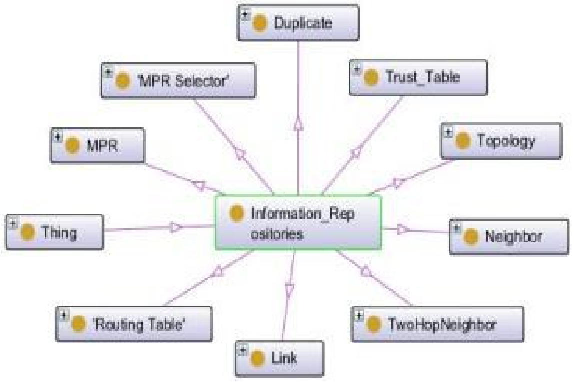
3.2.1. The Concepts

The main class hierarchy, consisting six concepts involved in the trust oriented OLSR ontology and the brief description about each class is given in Table 8.

Table 8: Main Concepts of OLSR Ontology

Concepts	**Description**
Node	This is the basic representation of a machine i.e. represented by identity
Object	This is the basic entity in the ad hoc network
Attributes	It is about the various attributes of the objects
Information Repositories	to represent the information repositories used for the operation of the ad hoc network
Packet	The description of the Packet in such type of network.
Operations	The operations performed while in operation

The various subclasses under Information_Repositories are shown in the Figure 10. These are the classes used to keep the information required by the node for their operation in the network. The Attributes class with components representing attributes of the node is depicted in Figure 11



.

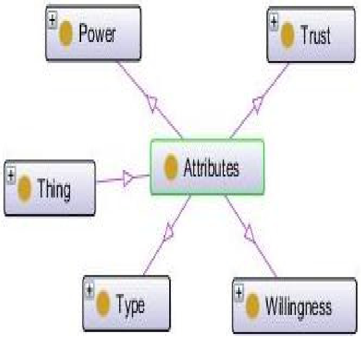
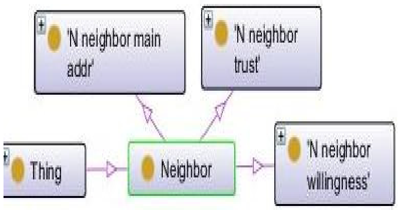


Figure 10: Concepts in Information Repositories

Figure 11: Concepts in Attributes

The Neighbor in the Figure 12 is used to store the information about the direct neighbors of node. The neighbors address, trust in that neighbor and the willingness of the neighbor to participate are the components of this class. The TwohopNeighbors with subclasses for the address of twohop neighbor, the trust on that neighbor, the address of the neighbor via it is twohop are the main components of this class shown in Figure 13.



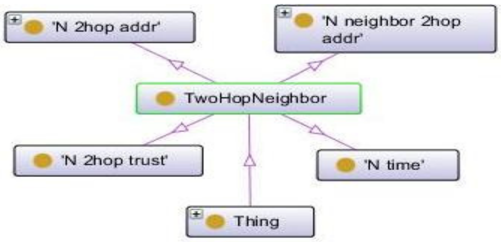
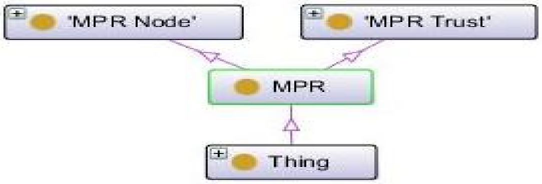


Figure 12: Structure of Neighbor

Figure 13: Structure of Twohop Neighbor

The MPR is Multi Point Relay used by the neighbor for routing purposes with its address and the trust on it is shown in Figure 14. The MPR Selector is the nodes address with its validity selects the holding node as its MPR presented in Figure 15.



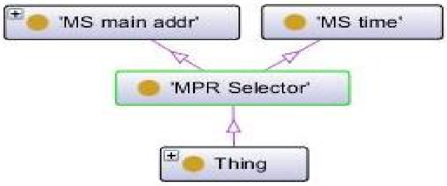


Figure 14: Composition of MPR

Figure 15: Composition of MPR Selector

The Duplicate class is used to avoid the reprocessing of packets that are already processed is shown in Figure 16. The address of the sender node its sequence number and the retransmitted state are the components of this class. In order to keep the topology information at any stage the Topology concept holds the required attributes or components is shown in Figure 17.

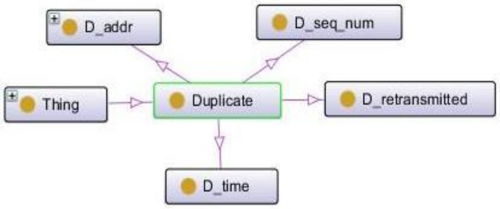


Figure 16: The Duplicate Class

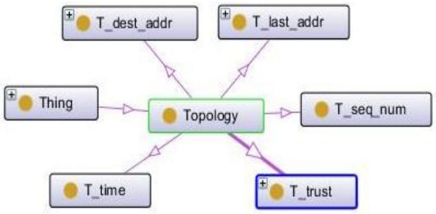


Figure 17: Topology Class and its components

The Link Class to store the information about the links with the trust value on that link is shown in Figure 18. The Trust Table is additional component in the present study which is not in traditional OLSR have the information about the trust of the source node on the other nodes is shown in Figure 19. The source node may not have the trust values for some nodes in the network.

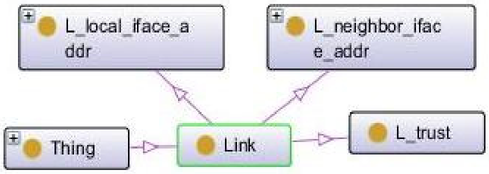


Figure 18: The Link Class

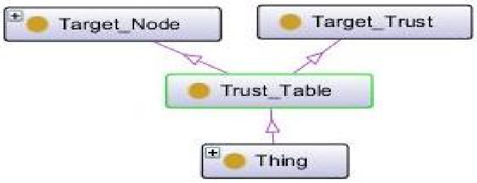


Figure 19: Trust Table Class

The Routing Table class having the information used for routing of the network with the address of the destination, its distance in terms of hop from the source node and the address of the next node to which packet is to routed is shown in Figure 20.

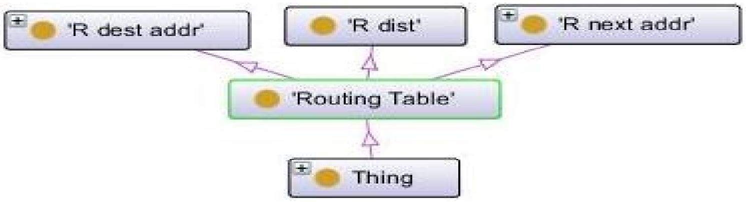


Figure 20: The Routing Table

The trust class with the choices of having either Qualitative or Quantitative value is shown in Figure 21. The same choices are also available for the willingness class shown in Figure 22.

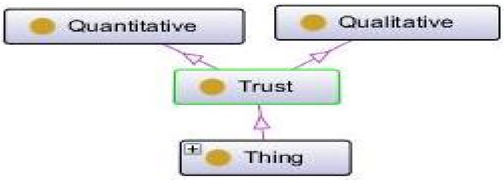


Figure 21: The Trust Class

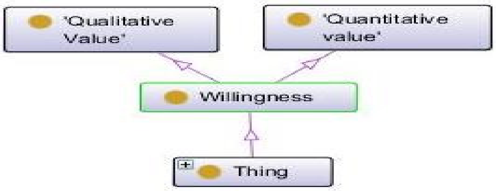


Figure 22: The Quantitative Class

In adhoc network all nodes are represented by an instance of the Object class as per OLSR protocol, is shown in Figure 23. The instances of this class have the all information necessary for their operation. Most of these are defined earlier. The other sub concepts, Battery is used to show the status of Battery at a given instant. The Type is used to identify the type of node for current operation. The possible values of it are – Source, destination or Intermediate.

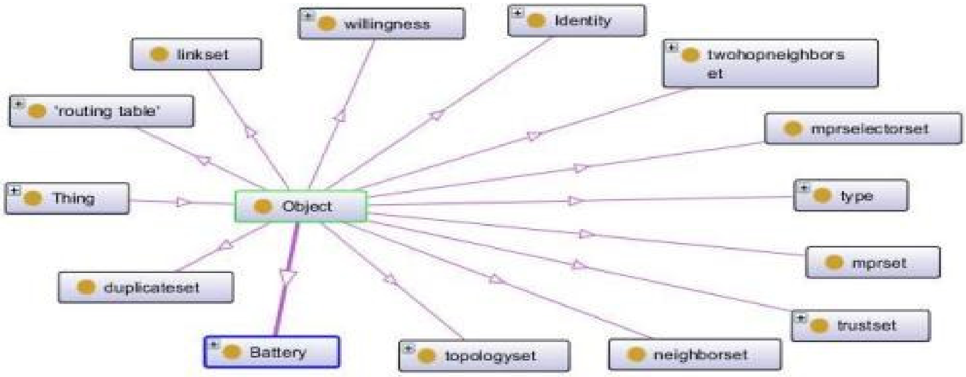


Figure 23: The Object Class

The possible operations while a node is in an operating environment are shown in Figure 24. The trust component is introduced in most of the operations for processing. In addition to the traditional operation, the trust message gets introduced to request and reply of the trust about a node.

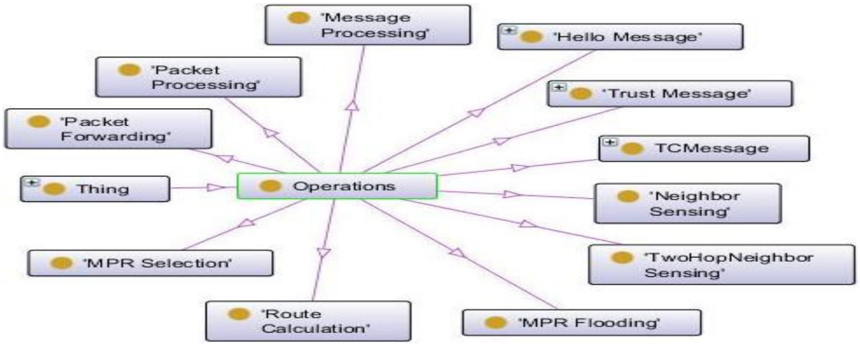


Figure 24: The Operations class Hierarchy

The structure of the Packet in terms of its header, message and its header with their attributes or components depicts in the Figure 25.

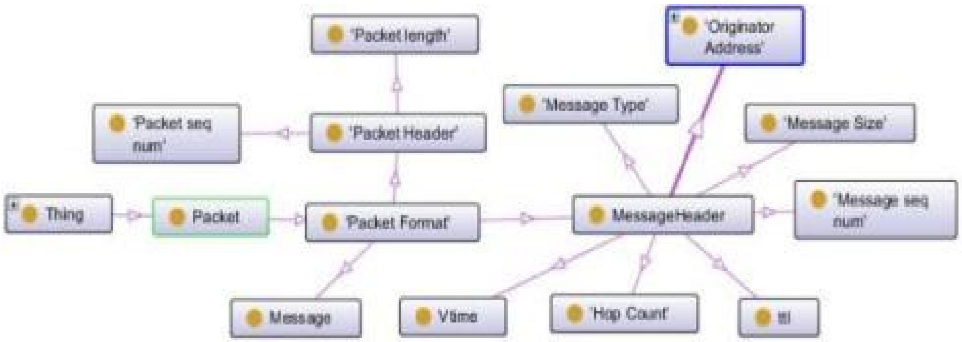


Figure 25: The Packet Class

The classes described have restriction on their objects. Most of the object values are restricted by universal quantification as the objects of these classes have well defined types for their values.

3.2.2 The Object Properties

The hasNeighbor object property has three sub properties shown in Table 9. The properties are for the subclasses of the Neighbor class – N_neighbor_main_addr, N_neighbor_trust and N_neighbor_willingness respectively.

Table 9 : hasNeighbor Object Property

Sub Property	Description
hasNeighborAddress	functional property that Neighbor's address to the identity of node
hasNeighborTrust	The property mapping neighbor trust to an instance of Trust
hasNeighborWillingness	mapping from neighbors willingness to an instance of the Willingness

The has2hopNeighbor object Property has three sub properties – N_2hop_addr, N_2hop_trust and N-neighbor_2hop_addr shown in Table 10.

Table 10: has2hopNeighbor Object Property

Sub Property	Description
has2hopAddr	This maps the address of the 2 hop neighbor to its identity.
has2hopTrust	mapping of the trust in 2 hop neighbor to an instance of the Trust class
hasNeighbor2hopAddr	This is to map the address of the direct neighbor through which designated 2 hop neighbor is connected to source, to its identity.

The hasObjectProperty has sub properties about the object sub classes shown in the Table11.

Table 11: hasObjectProperty

Sub Property	Description
hasNeighborSet	To an instance of the Neighbor class
hasTwohopNeighborSet	To an instance of TwohopNeighbor class
hasMPRSet	To an instance of MPR class
hasMPRSelectorSet	To an instance of MPRSelector class
hasIdentity	To an instance of Identity of Node class
hasWillingness	To an instance of the Willingness class
hasDuplicate	To an instance of the Duplicate class
hasLink	To an instance of the Link class
hasTopology	To an instance of the Topology class
hasTrustSet	To an instance of the Trust Table class
hasRoutingTable	To an instance of the Routing Table class

Some other object properties with the description from Domain to range are given in the Table 12. In addition to these properties there are some other properties not listed here. Most of the properties listed in this heading are functional in nature. Many of the properties also have inverse properties but not explained for this ontology, as explained for Trust ontology, to avoid the repetitions.

Table 12: Some other Object Properties

Sub Property	Description
hasAddr	This is from D_addr of the Duplicate to the identity of Node
hasAdress	It is from MPR Node of MPR class to the identity of Node
hasTrust	It is from MPR Trust of MPR class to an instance of the Trust
hasAddress	This is from MS main addr of the MPRSelector to Identity of Node
hasOriginatorAddress	from Originator Address Message Header class to an Identity of Node
hasDestAddr	It is from R dest addr of Routing Table class to an identity of node
hasNextAddr	It is from R next addr of Routing Table to an instance of Identity of Node
hasTargetAddress	from the Target Node of Trust Table to an instance of Identity of Node
hasTargetTrust	from the Target Trust of Trust Table class to an instance of Trust class.

The following Table 13 gives the description of some of the data properties with the names of the Domain class and data type as range for these properties.

Table 13: Some Data Properties

Property	Description
Is retransmitted	This is a Boolean for D_retranmitted of Duplicate class
hasNodeIdentity	It is to int form Node class
hasIdentityvalue	It is to an int value from Identity of Node
hasMaxvalue	to int value indicating Maximum permissible value of Quantitative Trust
hasMinValue	to int value indicating Minimum permissible value of Quantitative Trust
hasMaxWillingness	to int value indicating Maximum value of Quantitative Willingness
hasMinWillingness	to int value indicating Minimum value of Quantitative Willingness

3.3 The Ad hoc Network Ontology

3.3.1 The Concepts

The Ad hoc network ontology has three main classes namely- Node, Network and Applications as shown in the Figure 26. The application class is to describe the characteristic and type of the application running in the network. The network class is to depict the network characteristics and the Node class is to define the attributes of the node in the network.

The Application class as shown in Figure 27 has two subclasses – Application Load and Application Type. The possible values for these are shown in 'The Individuals' section of this ontology.

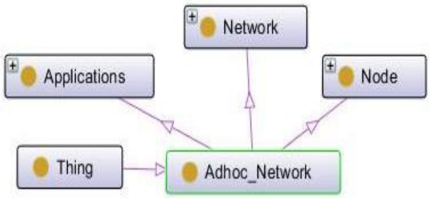


Figure 26: Ad hoc Network Classes

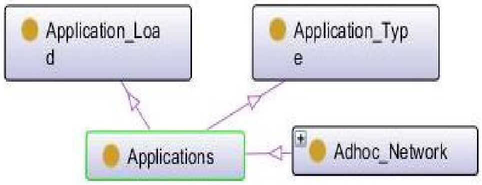


Figure 27: The Application Class

The network class as shown in the Figure 28 has number of subclasses to defines the attributes of the network like number of nodes, routing protocol used , placement of nodes, geographical area and many others.

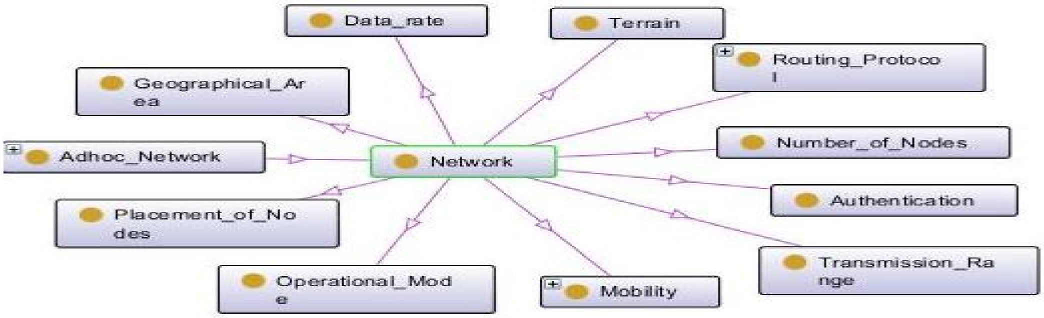
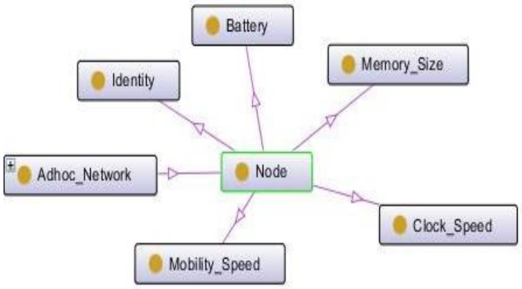


Figure 28: The Network Class

The node class has subclasses like identity, memory size, mobility speed, clock speed to define the characteristics of the node as shown in Figure 29. The mobility subclass of network has three subclasses- Trajectories, Direct manipulation and random to select a mobility pattern of the nodes in the ad hoc network is shown in Figure 30.



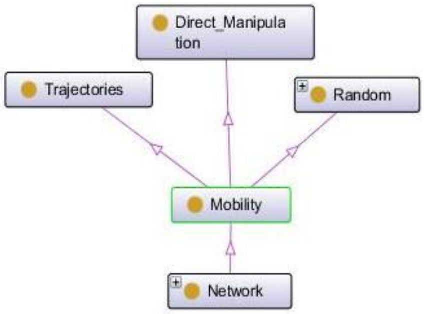


Figure 29: The Node Class

Figure 30: The Mobility Class

The Routing class is a subclass of the Network class. The possible types of Routing Protocols with possible values of the individuals for such type of protocols are shown in the following Figure 31.

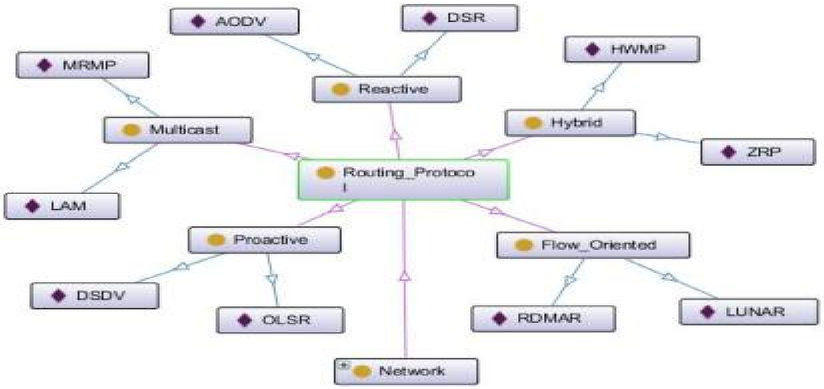


Figure 31: The Routing Protocol Class and its Individuals

3.3.2 The Individuals:

The individuals for the Application Load are Low, Medium and High to tell about the load of the application in run, is shown in Figure 32. The possible individual's for Application Type are shown in Figure 33. This list of individuals of the application types is not an exhaustive one. Many other individuals are possible for it.

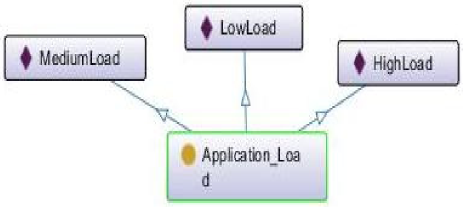


Figure 32: The Application load Individuals

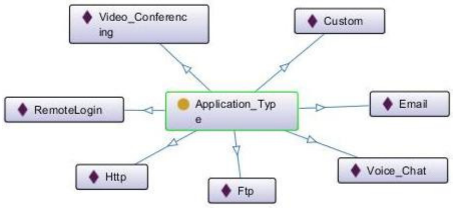


Figure 33: The Application Type individuals

4. IMPLEMENTATION AND RESULTS

These three ontologies – adhoc, OLSR and trust ontology are integrated using Jena-2.6.4. This integration of the ontologies provides a framework, serves as a template for building a trust oriented environment for ad hoc network. In order to validate this framework, a trust environment is implemented to make a decision whether to allow or discard the operation.

4.1 Experimental Setup

The maximum trust value is 10 and minimum trust value is 0, so if 't' is trust value $0 \leq t \leq 10$. The trust policy adopted to allow or discard an interaction on the basis of trust and the context of the application is as given in the following table 14.

Table 14: Trust policy

Context	Trust Value
No Risk	>0
Low Risk	>1
Medium Risk	>3
High Risk	>5
Highest Risk	>8

The nodes are populated with the random values for the trust about the other nodes. In addition to it the information repository of the node i.e. direct neighbors and two hop neighbors are also populated randomly. These direct neighbors and two hop neighbors are recommenders of the trust values, so to avoid any biasness about their selection, selected on random basis. The present study uses an ad hoc network with 100 nodes. The trust update Policy is that if the interaction is allowed then the trust of the source on the target gets increased by one, otherwise decreased by one.

4.2 Results

In order to study the effect on successful rate of interaction on the basis of trust by executing more applications, a hundred simulations of the network for the same number of applications made i.e. 100 simulations with 10 applications/interactions running on the network and then

100 simulations with 50 application running in the network and so on, the result gets averaged to have the final value as shown in Table 15.

Table 15: No. of application vs. Successful rate

No. of Appl. in one simulation	No Risk	Low Risk	Medium Risk	High Risk	Highest Risk
10	100	99.48	84.98	44.89	.5
50	100	99.60	84.93	47.90	1.8
100	100	99.79	83.15	51.03	3.9
200	100	99.4	87.92	56.62	7.5
500	100	99.93	91.34	64.08	10.01
1000	100	99.83	94.38	64.09	40.25

The data of the above table is represented graphically in following figure 34. It is inferred from the graph that as the number of applications increases the successful rate of applications increases, especially for high and highest risk applications. The reasons for this as the initial value(s) of the trust on other nodes are random values and do not satisfy the trust policy requirement of such high risk applications. As the network progress the trusts of some nodes get increased after their successful operation which may satisfy the trust policy requirements.

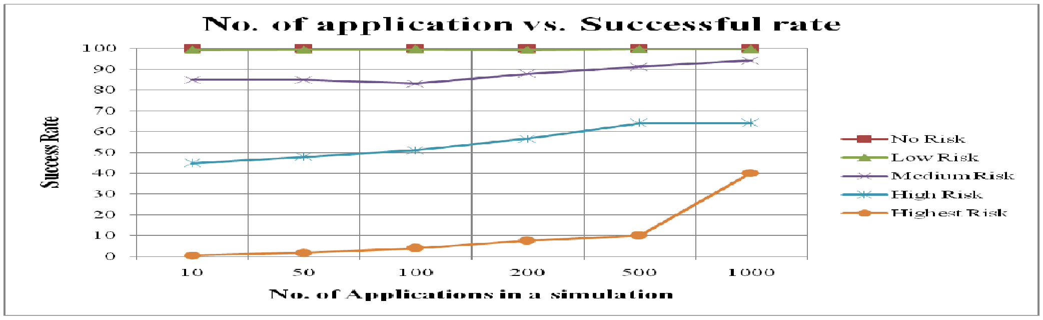


Figure 34: No of Applications Vs Successful rate

As inferred from the graph, there is always increase in trust value and there is no decrease in it. In order to show trust also decreased, in the table, the data is shown where the involved applications are of either medium risk, high risk or highest risk. This is to avoid the suppression of decrease in trust values due to the unsuccessful operations of High and Highest risk due to the increase in trust by No Risk and Low Risk application which have almost 100% successful rate.

Table 16: Risk and Successful Applications

No. of Applications in one simulation	Medium Risk	High Risk	Highest Risk
10	81.48	46.07	0
50	81.71	43.76	.3
100	81.25	43.11	.4
200	82.70	43.44	.2
500	80.84	42.75	.06
1000	79.68	43.12	.01

The graphical visulisation of the above tabulated data in table 16 for medium risk, high risk and highest risk are shown in Figures 35. If the network experiences only medium, high and highest risk applications then the success rate get changed and even decreased as the number of applications increased. The reason attributed to this conclusion is more unsuccessful application cause trust to get diminished, so as a result more unsuccessful applications. From these results it is inferred that trust is dynamic in nature.

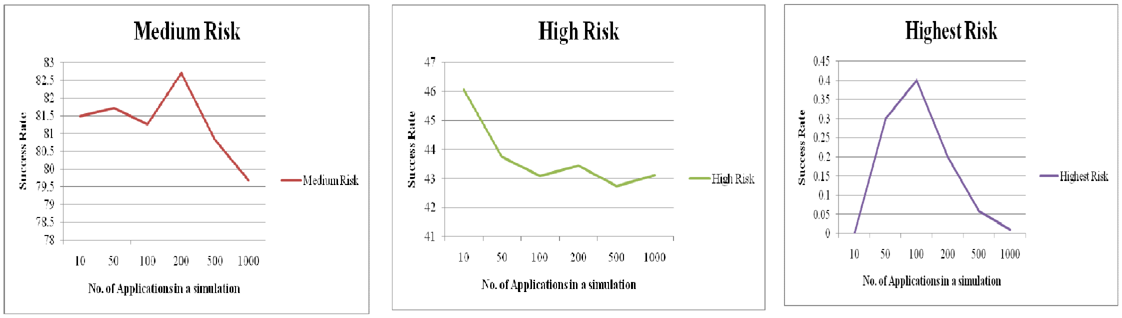


Figure 35: Dynamisim in Trust Values

Two types of trust- Global Trust, a trust value as a result of recommendations from the whole network, and Local Trust, a trust value as a result of the recommendations of direct neighbors are considered. In order to understand the effect of these for different types of contexts of application the results of the following setup are tabulated in Table 17. In addition to it effect of assigning different weights to the recommendation depending on the proximity of the recommender to the source node are also put into table.

Setups of the network: 100; Simulation for each network: 10; Number of Applications: 100

Table 17: Global Trust vs. Local Trust

Type Of Trust / Context	Global Trust		Local Trust
	Different Weights	Same Weights	Same Weights
No Risk	100	100	100
Low Risk	99.07	99.65	99.17
Medium Risk	81.65	80.00	81.84
High Risk	47.43	41.82	42.79
Highest Risk	.09	.07	.02

It is evident from the table that the successful rate of the applications differs significantly as a result of Global trust and Local Trust for High Risk and Highest Risk Application. It is also concluded from the following graphs shown in figure 37 that by assigning different weights the there is gain in successful rate for medium, high and highest risk applications. In the following figures the first point in each graph shows the usage of Global Trust with different weights, the second point shows the usage of Global Trust with same weights and the third point is the usage of Local Trust only and the weights are therefore same in this case.

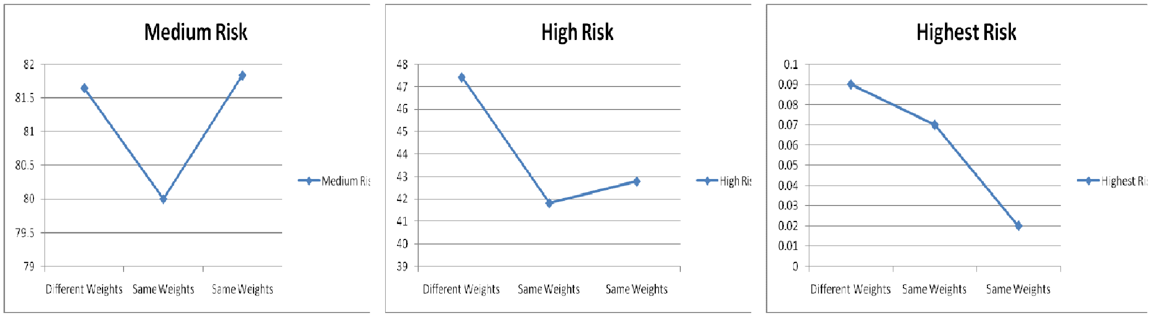


Figure 37: Global Trust Vs Local Trust

5. DISCUSSIONS

On the bases of above results the perspectives where the suggested approach and framework can be used as follows.

5.1 Application Execution

Any node in the network may have malicious behavior and this behavior lowers the value of trust on this node by others. Before the execution of any application on some other node, it is advisable to evaluate the trust on that node in such ad hoc environment. The applications that one node want to execute on/with other node may vary from highest risky to no risk at all. Even the same application may be of No Risk sometimes and may be of highest risk at other times depending on the contents like in emails. The proposed framework provides the feature that the context can be specified along with the application. Most of the earlier studies allows or discard the execution of any application without the consideration of the context. In such situations the security requirements of no risk application is the same as for the highest risk application. For obvious reasons the proposed framework suggests the reduction in security complexity by taking into consideration the context of the application in decision by the trust policy.

5.2 Routing Environment

For routing environment the interaction means forwarding of the packet. As far as routing environment for ad hoc environment is concerned, most of the attacks are due to the malicious nature of the nodes en route. In the present study, the proposed framework assumes the trust updation after the completion/ time out of the interaction. Nodes in an ad hoc network are able to observe the behavior of their 1-hop neighbors directly. So if any node drops the packets or forwards it to un-legitimate nodes, this behavior gets observed by all its 1-hop neighbors. In other words the immediate source of the packet get noticed this behavior and ultimately reduce the trust value for that node. So the persistent behavior of the node to drop the packet ultimately lowers its trust value. This decrease in trust value avoids its selection as node en route for future communications and also causes routing tables of the protocols to be modified to avoid any route through this node and try to find other routes excluding this node en route. Many studies use trust for routing computations. But this route selection is for all applications, irrespective of their contexts. Hence the same route is used for No Risk and Highest Risk applications. The cost spending on following the trusted route for No Risk applications is of no use and of least use for Low Risk applications. It is advisable to have different routes for the same destination depending on the context of the application.

5.3 Authentication

Authentication in network relies on the public key certificates signed by some trustable nodes. The idea to select or reject the authentication due to the trust value of the introducer without taking care of the degree of security required for that leads to inflexible constraint that must be satisfied. Generally, accept or reject depending on the threshold value of trust. This is discrete value. This is same for all. This type of threshold proves to be fatal for high risk applications because of low threshold value. On the other hand No Risk and Low risk application are not possible to execute on such node because authentication of that node gets failed because of high threshold value of trust. Therefore it is difficult to find the single optimum value of trust. The proposed framework is able to authenticate, moreover context can be used to have a set of threshold values rather than having single threshold value.

5. 4 Pick the Best

In order to pick the best among the available options, the criteria to select the most trusted one is the obvious choice. The most trusted one is more costly as compared to others in terms of money, complexity and many other related factors. Therefore best does not means the most secure. So to make the choice more cost effective and even secure the context is used to select among the options available. The proposed framework is also able to handle this type of perspective.

6. CONCLUSION

The objective of the ontology is to attain, to descript and to symbolize the knowledge of allied fields for modeling the trust for ad hoc networks to provide a common understanding of the fields, and then to give a clear definition of the vocabulary and the mutual relations between the vocabulary from the different levels.

7. LIMITATIONS AND FUTURE SCOPE

The proposed work is not able to handle the fuzziness, if exists, regarding the context associated with the operation in execution. Moreover, the uncertainty in trust value is not used in the work. The future work is to include the uncertainty and introduce the fuzziness in the contexts.

REFERENCES

[1] Anna C. Squicciarini, Elisa Bertino, Elena Ferrari and Indrakshi Ray, Achieving Privacy in Trust Negotiations with an Ontology-Based Approach, IEEE Transactions on Dependable And Secure Computing, Vol. 3, No. 1, pp. 1-18, 2006.

[2] Haase, P. & Sure, Y, "D3.1.1.b State of the Art on Ontology Evolution" http://citeseerx.ist.psu.edu, ,2004

[3] Mark E. Orwat, Timothy E. Levin, and Cynthia E. Irvine, "An Ontological Approach to Secure MANET Management", Proceedings of International Conference on Availability, Reliability and Security, pp. 787-794, 2008

[4] Mizoguchi R. "Tutorial on ontological engineering", http://www.ei.sanken.osaka-u.ac.jp/pub/miz/Part1-pdf2.pdf

[5] Mohsen Taherian, Rasool Jalili, Morteza Amini, "PTO: A Trust Ontology for Pervasive Environments", *Proceedings of IEEE* International Conference on Advanced Information Networking and Applications, pp. 301-306, (2008).

[6] Mohsen Taherian, Rasool Jalili, Morteza Amini, “A Semantic-Aware Ontology-Based Trust Model for Pervasive Computing Environments”, LNCS, Publisher: Springer-Verlag, Volume: 5060 , pp 47-59, 2008

[7] Rajiv K. Nekkanti and Chung-wei Lee, “Trust Based Adaptive on Demand Ad Hoc Routing Protocol”, Proceedings of the ACM Southeast Regional Conference, pp. 88-93, (2004)

[8] Rim Djedidi and Marie-Aude Aufaure, “ Ontology Evolution: State of the Art and Future Directions”, http://perso.ecp.fr/~aufaurema/Ontology-Evolution.pdf,

[9] Stefania Galizia, Alessio Gugliotta and John Domingue, “A Trust Based Methodology for Web Service Selection”, Proceedings of IEEE International conference on Semantic Computing, pp. 193-200, (2007)

[10] Tirthankar Ghosh, Niki Pissinou and Kami Sam Makki, “Towards Designing a Trusted Routing Solution in Mobile Ad Hoc Networks”, International Journal of Mobile Networks and Applications, pp. 985-995, (2005)

[11] Wanita Sherchan, Surya Nepal, Jonathon Hunklinger, Athman Bouguettaya, “A Trust Ontology for Semantic Services”, Proceedings of IEEE International Conference on Services Computing, pp. 313-320, (2010)

[12] Wei Guo, Zhong-Wei Xiong and Ren-Zuo Xu, “Functional Ontology of Routing Reputation for MANET”, Proceedings of International Conference on Wireless Communications, Networking and Mobile Computing, (Oct 2008), pp. 1-5

[13] Yoshinobu Kitamura, Toshinobu Sano and Riichiro Mizoguchi, “Functional Understanding based on an Ontology of Functional Concepts”, Proceedings of Sixth Pacific Rim International Conference on Artificial Intelligence, (August 2000), pp. 723 – 733.

[14] Z. Yan, P. Zhang and T. Virtanen, “Trust Evaluation Based Security Solution in Ad hoc Networks”, Technical Report, Nokia Research Center, (2003)

[15] Protege, http://protege.stanford.edu/download/download.html