ROLE BASED CROSS-LAYER COMMUNITIES ON WMN

João Paulo Barraca, Susana Sargento, Rui Aguiar
Instituto de Telecomunicações, Campus Universitário de Santiago, Aveiro, Portugal
Department of Electronics, Telecomunications and Informatics, Universidade de Aveiro, Aveiro, Portugal
jpbarraca@av.it.pt, ssargento@det.ua.pt, ruilaa@det.ua.pt

Keywords: Communities, cross-layer, cooperation, access control, mesh networks

Abstract: The community notion can be exploited as a rational concept leading users to cooperate in sharing resources on Wireless Mesh Networks. We propose a novel concept for self-organizing networks, where multiple entities (network elements or users) collaborate to achieve common goals, and in particular, to establish the basic connectivity and service delivery infrastructures. The resulting architecture is based on wireless mesh communications, with different entities taking different roles in the communities in a cross-layer approach. These communities can collaborate, leading to increasingly complex and geographically extended scenarios.

1. INTRODUCTION

Wireless mesh networks are able to provide broadband access using an adaptable wireless infrastructure, directly tied to user interaction. Mesh networks exploit ad-hoc networks concepts where self-organization, cooperation and distributed operation are vital concepts. Its usage scenarios are usually related to the provision of broadband network access to distant rural areas or in dense urban environments [Borcoci, 2007], being actively exploited by several companies.

In rural areas, far from any wired network access, meshed operation using technologies such as 802.16 can cover areas as large as 1000km^2, providing reasonable broadband access to geographically distant users. On the other hand, in metropolitan environments, meshed technologies enable the deployment of cost effective networks with added benefits in terms of its adaptability to user density, self-organization and self-healing capabilities, as well as the capacity of favouring content creation by users. This last benefit is expected to be the centre of next generation networks where the user focuses on direct user interaction rather than content provision by large content providers.

A fully distributed architecture enables cost effective scalability at the cost of lower predictability as stated in [Dressle, 2006]. Mesh networks can evolve by combining the thousands of access points (AP) existing in metropolitan areas into a single adaptable network. In terms of adaptability and resilience this is the ideal move; however, the approach leads to a large degree of heterogeneity in access conditions as well as decrease in reliability. Furthermore, it is not expected that both users and operators deploying wireless equipment let the mesh operate in a fully distributed (uncontrolled) manner, retaining some control either by personal or monetary reasons.

To address the management of these environments in a distributed and efficient way, this paper proposes a community-oriented architecture, developed inside the IST-WIP [IST-WIP] project. The community approach envisioned considers that communities are formed across different layers with the possibility of having different objectives, e.g., wireless resource sharing, routing support, exchange of specific distributed application-layer services. In this sense, we propose a cross-layer approach for the communities' creation and management: the communication between several nodes requires a cross-layer interaction and agreements between different types of communities, where this interaction is addressed in a similar way across layers. We also describe the benefits and the main impact, on a community driven architecture, and the application of the generalized community concept.

This paper is organized as follows. Section 2 presents a simplified vision of the mesh network architecture. Section 3 discusses the communities' concepts, the roles and rules applied to the community elements, and the cross-layer approach. The management process of the communities is addressed in section 4 for intra-community, and in section 5 for inter-community interactions. Section 6 describes some interesting business models achieved with this approach, and section 7 presents the final conclusions.

2. WIP ARCHITECTURE

The WIP global network is structured based on a wireless backhaul, optimized for high performance forwarding between fixed access points that provide overall connectivity to highly mobile terminals organized into spontaneous sub-networks. The wireless backhaul will make use of sophisticated techniques for high performance transmission (directional antennas, multiple radios), and technologies such as 802.16. For practical reasons, mobile terminals need to live with more traditional wireless solutions such as omni-directional antennas and standard 802.11a/b/g wireless equipments on off-the-shelf laptops.

The wireless backhaul presents some similarities with the current Internet: access points are fixed end-points and its role recalls the functions of the core Internet. In fact, multiple accesses to current Internet can be pruned in this backhaul. However, for spontaneous sub-network formation of mobile terminals, we also need components designed from scratch to deal with mobility and self-organization.

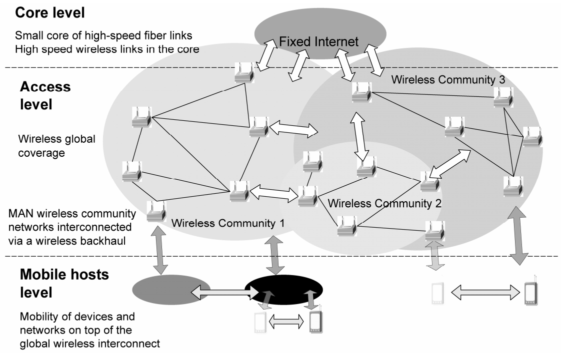

Figure 1: Topological WIP architecture

The nodes in the WIP architecture contribute to the operation of the network by participating in traffic forwarding and by providing some local resources and services. Their contribution may depend on the state of local resources and, fundamentally, on the node' willingness to share them. Their operation is autonomic, i.e. they operate without explicit human intervention (e.g. nodes may choose frequency bands, time schedules and topologically valid addresses…).

In order for end-users to decide to participate and contribute to the wide coverage of this radio internet, it is important that it is designed so as to be an attractive alternative to the wired Internet for private or group communications. Towards this end, the WIP project is focused on developing approaches concerning mobility management and routing avoiding the restrictions imposed by the current Internet architecture. Moreover, it aims at building an architecture that enables the spontaneous creation of several types of communities (existing or novel ones) allowing them to self-organize and customize the network functionality to suit their own needs. Also, this community notion can be used in the creation and efficient operation of a mesh network. First, as a general concept, the notion of a community could be used for efficiently implementing aspects of the network (or lower) layers' functionality. Additionally, carefully designed communities could provide adequate incentives to users (and nodes) to participate in the WIP network sharing their available resources (their wireless access points, their Internet access), which in most cases is necessary for ensuring a wide coverage of this type of wireless networks.

3. COMMUNITIES

In our generalized community concept, communities are defined as a set of entities that collaborate or cooperate. The concept is traditionally used to denominate a group of users united by a common interest, and collaborating towards a predefined objective. Examples of such collaborating communities can range from the web based encyclopedias [Wikipedia], forums where users exchange ideas, or comment movies [IMDB], to file sharing networks [Emule] or distributed computing services [SETI@Home]. A seldom considered collaboration based on communities is related to the network elements themselves: some routing protocols [Haas, 2002] already had similar collaboration concepts, exploiting the ability of the network to organize and cluster. However, currently at this level there is no formal description of the participation rules and interaction is somewhat limited to a rigid task (routing, QoS, monitoring or charging).

The next sub-sections present the roles (and its rules) we apply to community entities, and the cross-layer approach.

3.1 Roles and Rules

Rules are vital to communities as they define the behaviour nodes should follow, resources they can use, and the configuration parameters they should apply. The agreement on the same set of rules will create a coherent environment, even if members have different resources or provide different services. All communications between nodes in the Internet share a “common context” with rules and roles (server-client). In our approach, the rules composing a role do not define complete communication mechanisms (such as IP, TCP or 802.11). Instead they define the policies governing the underlying mechanisms as well as the available services.

Examples of rules would be the wireless channel to use, the key and mechanism to cipher packets, the bandwidth to use for P2P file sharing, or the permissions regarding a particular service.

Taking as an example a neighbourhood community grouping all APs and terminals in a city, this community contains different equipments operating autonomously and with different resources. The rules applied to each community member may suffer some changes. In this scenario, as an example, APs can have a rule stating the possibility of selecting the wireless channel; on the other hand, laptops can be forbidden to choose the wireless channel due to their reduced knowledge of the spectrum, or participating in routing due to their higher instability. However, they can support enhanced services like processing and storage, which may be unavailable to routers due to lack of resources.

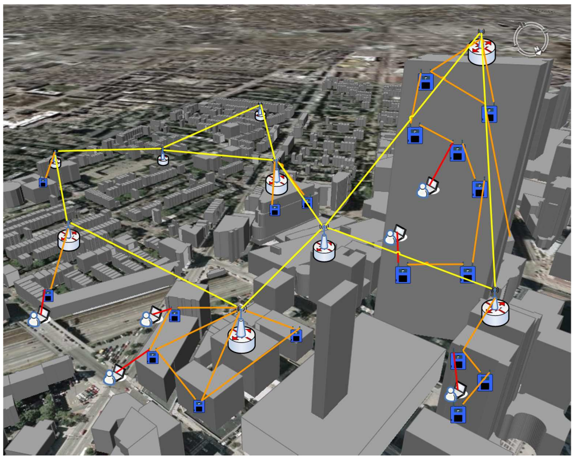

Figure 2: Heterogeneous neighbourhood community (from Google Earth)

All the set of rules which determine a functional entity compose a role. Depending on the interests of the members and resources available, members can always decide to act according to a given role. This will imply that the community will accept the member to perform such role, and the member will act accordingly to the rules composing the role. Roles can also be composed by no rules. A role with no rules is still useful as it may condition the access to resources, while not enforcing any specific behaviour.

Some roles can result in actions requiring higher trust than others (e.g., the role of a public storage will require high trust due to privacy requirements). A hierarchical environment is thus formed, requiring the members to correctly perform some roles in order to be eligible for more important roles. Also, trust information must be obtained from each interaction and later used in the process of role delegation.

3.2 Cross-layer Approach

Communities can coexist at different layers, contextualizing all types of communications; in this sense, the community mechanisms act vertically managing the control plane of the stack.

Also important on user driven networks, topology will influence user interactions as new users sharing the same interests become reachable. Conversely, users' interest will shape the formation and organization of network elements in order to optimize operation.

In a cross-layer community environment, access points and backhaul routers cluster in communities in order to optimize the scarce wireless resources, and users establish their upper layer communities above the created network infrastructure. Services related with media distribution or distributed processing could also create their own communities on top of the network support. Such behaviour is currently much exploited by communities exploited by viral marketing campaigns which promote upcoming movies, games or products.

From this vision, network and application/user level communities co-exist in the same scenario. For communication to be possible, user level communities require the existence of another community providing the actual physical delivery or a direct interface with the network stack. In Figure 3, users create an application community (A) sharing music. However, they will only be able to communicate using two different communities to route packets (B and C). In this case, for communication, it is formed a concatenation of multiple communities for transport support.

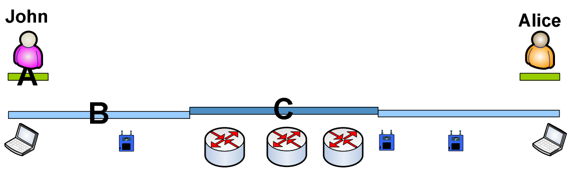

Figure 3 : Community interactions

The example can be managed considering interconnection between communities where some communities request delivery from other communities. The user community (A) asks to the routing community (B) to send a message to other participants of the user community. The B community accepts routing and performs the same request to C. Note that such exchanges may be confined to QoS parameters or traffic amounts and even require some form of payment/rewarding across communities. Also, some restrictions on authorizations can be applied. These concepts will be described in section 5.

4. COMMUNITY MANAGEMENT

The basis of the community management employed on our architecture is derived from solutions like RBAC [Sandhu, 1996] and more recently dRBAC [Freudenthal, 2002] and RT0 [Li, 2001]. All these solutions propose mechanisms to control the access to resources and services by the roles the entities possess. Entities create roles and delegate them to other entities when requested or required. Depending on the entity requesting the permissions, the role can be delegated with modifications from the original role. These modifications are used to further restrict the permissions of the requesting entity. It is important to notice that an entity can never delegate more permissions than the ones it already has.

4.1 Community Bootstrap

Entities bootstrap communities by creating a community description (called doctrine) with a set of roles and eventually some general rules and attributes. These rules may restrict the dimension of the community, its physical location or impose any requirement on participation. The entity creating the community will have a special function as it owns all primitive roles. After new members arrive, these roles can be delegated. Depending on the purpose of the community, management can be distributed only if the owner allows such operation. This is expressed by the right of delegation. If no right of delegation is ever given to members, only the owner has the capability of authorizing new entities to join or enrol new roles.

Due to the dynamic behaviour of mesh networks, where some nodes are mobile, a distributed approach is required, and it is important to delegate the assignment right to other trustworthy entities as it increases the scalability and resilience of the network. If a community is to emulate a service or network provider, it is expected that only one or a small number of entities to actually have the capability to delegate new roles.

Because delegation chains need to be verified until the first issuer, it is desirable to have short delegation chains. These top nodes are first assigned by the creator of the community and should be chosen taking into account its attributes on processing capabilities, stability, low mobility, and above all, trust.

4.2 Clustering and redundancy

When delegating a top role to an entity, the community creator will also transmit information about some other entities sharing the same role and assignment permissions. With this information, the enrolling entity, following a cluster-based approach, will exchange information about the amount of delegations given and the percentage of computation, memory and wireless resources available.

Delegations requests are then load balanced between the several entities using this information. Such mechanism will actually balance the delegation tree and avoid the creation of long delegation chains.

In the occurrence of a delegation request which should have a negative response, the answer is sent directly not requiring forwarding to the other entities.

Other aspect considered is the support for delegation redundancy. In the case the top issuer of the delegation chain is unreachable, it will be harder for active entities to validate the delegation chain. Without this validation, it will be impossible to assure the delegation is still valid.

Entities may try to obtain same delegations from different issuers following different paths. The result will be the same delegation will be verifiable following different paths. Due to overhead constrains, this operation should be preformed only when the community is idle.

4.3 Community membership

The *Member* role is the one every entity must enrol in order to participate on the community. When a node wishes to join a given community it must locate a member and issue a request to join. If the member has been delegated the capability of assignment for this role, and the candidate follows the requirements of the *Member* role, the member may allow it to participate. Otherwise the member may indicate other entity (such as the issuer of its *Member* delegation) for the candidate to contact. Figure 4 depicts a sequence chart occurring when Joana and Peter want to join the community *C*. Maria was already authorized by John, which is the community creator. Notice that this example could be performed through different types of entities, running lower layer protocols, such as APs.

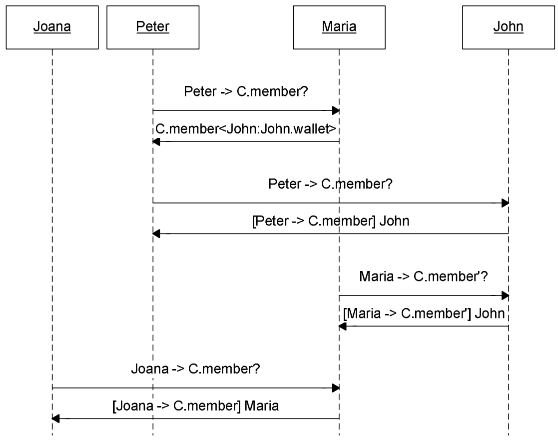

Figure 4: Joining a community through delegation

All members store the delegations provided to them on their *Wallet*. As proposed in [Freudenthal, 2002], this *Wallet* acts as a repository for the delegations issued. Entities may also publish their delegations to the rest of the network creating a distributed *Wallet.* Entities are capable of searching other's *Wallets* or enumerate proofs by a set of parameters. Moreover, when asked for a proof, any entity can either provide the proof, or return a pointer to another *Wallet* containing the required proof. This process continues until the proof is found or the entity returns no pointer.

Distributed algorithms already exist in the literature to efficiently locate objects on a distributed *Wallet*, having especial relevancy the ones based on Distributed Hash Tables. It should also be noticed that the same proof can be simultaneously mirrored at different *Wallets*. Such behaviour will enable both redundancy and faster location of the required information.

4.4 Resource Management

One important functionality for mesh networks is the possibility of adaptation to network conditions. Current proposals address this issue by providing solutions each managing a specific layer of the IP stack. Solutions based on zone routing [Haas, 2002] cluster route dissemination information taking in consideration the location of nodes. Others efficiently allocate the best wireless channel based on local estimation of interference [Ramachandran, 2006].

The community doctrine allows the description of formal rules stating the configurations to follow, but most importantly, they allow the real time negotiation of the best parameters with subsequent propagation of changes. This can be performed independently of the stack layer their resources refer to. Moreover, changes can be either global or affect only the entities belonging to the same role. One example is dynamic negotiation of the wireless parameters for backhaul transport, without changing the same parameters at the user part. Today this is possible, but not formally integrated and following a role based, and secure model.

In the scenarios we envision, management of resources can be performed using different methods. For a start, the creator of the community, having control over all roles and configurations issued, can force any rule into the community. Such action will result in a centralized management and should be only used when the community follows such management. Because members are free to create their own communities, if the community creator oppresses its members, it risks to loose control over the members.

In community with partially distributed management, entities can request their delegation issuers for a specific change either to a role or to the community. Because the specific attribute or rule can be defined either by any entity in the delegation chain, the request is propagated until it reaches the entity responsible. On a community with centralized management, this entity is the community creator.

If an offending rule or attribute is stated by a self-signed delegation (an entity somewhere in the middle of the delegation chain), this entity decides upon the request and either changes the role or denies the request.

If management is distributed among a list of entities, and the request targets a community attribute or rule, an election process takes place. The first managing node to receive the request checks if the request is valid. This can concern the verification of the number of elections requested by the entity, or any other constrain stated on the doctrine. Then, it requests all the entities with the management role to vote. These entities also forward the request to other entities they know and wait for a reply. Duplicate requests for the same election are obviously dropped. Each entity then takes a decision and replies with a signed response to the entity which sent or forwarded the request. Votes are kept at each forwarding entity and the accumulated result is propagated. Any entity can request the individual votes to check if the values reported are according to the votes received. The entity leading the election will then receive the result of all votes and issue a verdict to the requesting entity. It will also send the result to the voting entities which should apply it. In this phase it is still possible to request verification of the votes reported. Corruption of such information will result in severe punishment for the offending entity and cancelling of the election. If the election ran favourably to the request, roles are updated with new delegations being issued, replacing existing delegations.

The case of fully distributed management, the process occurs in the same manner, only it affects all entities. Because this process consumes much processing and bandwidth resources, the assignment of the voting right should be restricted. Alternatively the number of elections per entity should also be restricted. In small community with only a few entities, the process is efficient and capable of rapidly propagating changes to their neighbours.

5. INTER COMMUNITY AGREEMENTS

One of the main advantages of wireless mesh or ad-hoc technologies is their high adaptability to events, along with the fact that distributed algorithms can be deployed in order to manage the network in an autonomic manner. These networks are self-organized and self-managed, making use of contextualized management as defined in their community doctrine. Mesh communities are able to dynamically, and without user intervention, negotiate and trade services or perform peering agreements. The issues regarding distributed operation and self-management capabilities are related to resources and trust.

For a node to be able to properly decide if an agreement is to be established, it must possess knowledge on current community status, which thus may require a distributed knowledge base. Also it must have the permissions to actually perform such decision or take part on the negotiation. More resourceful peers will be responsible for actually storing and maintaining the information consistency.

Following the role based access model, interoperation between communities is expressed as the existence of a valid delegation. The result is all inter-community management is performed in the same manner. So, overlay communities are treated in a similar manner to neighbourhood communities. The main difference between the two is members of two neighbouring communities are only members of one community. If one community is overlaid on another, at least one entity belongs to both communities at the same time. Thus it may have different permissions inherent to the role of member.

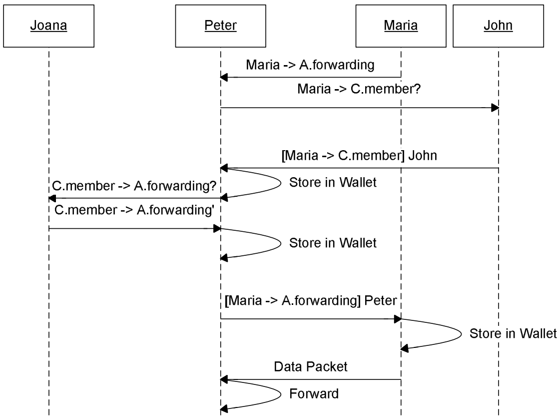

Figure 5: Forwarding of packets between different communities

In the example depicted in Figure 5, Maria and John are members of the community C while Joana and Peter are members of community A. When Maria asks Peter to forward a packet, first Peter checks if Maria is really a member of C. Then, because he has no delegation authorizing communication with other communities, it asks Joana about the delegation. In the request it asks if packets from a member of community C can be forwarded by a member of A. Joana replies with a delegation stating that any member of C can use A to forward packets. Also, this delegation can be further assigned to other members.

Several other communities may be available for the forwarding of traffic of A, and the one offering better conditions will be the one chosen. For example, another community F may have a role stating that, for community A, 5Mb/sec are available. However, for community C, only 2 Mb/sec are allowed. Therefore, only 2 Mb/sec can be actually used between A and C using F to forward. This is one example of negotiation and access control; other examples can include the time period of forwarding, the type of traffic, or any other attributes or restrictions applied to this community concept.

After all delegations are stored on Peter's *Wallet*, it forwards the packets. In this case, Joana, as the community creator (this is a simplified community with only one top node), could have limited to only authorizing forwarding during a certain time period or using a given bandwidth.

Again, notice that the names Joana, Peter, etc., may not be the user names, but names assigned to specific nodes in the network or even unique random identifiers.

6. BUSINESS OPORTUNITIES

Following the proposed communities' management mechanisms, new and novel business models can be built on top of the communities' concept.

First, the proposed mechanisms enable new nodes/users to join the community automatically and in a self-organized approach. This enables both the support of access control for new nodes in the network and for adaptable shared communication contexts. This approach can then be used to build micro operators, wireless and mesh based, with core nodes fixed and already established in the community, with possible network extensions through new nodes. For this business model to be possible, incentive mechanisms should be in place to enforce cooperation of new nodes. Moreover, through the communities' management and access control concepts proposed, only authorized users can access the community (network) and the services available, emulating the same behaviour of access to the operator services. Moreover, access constrains may take in consideration reputation or resources, and be applied to any layer of the communication stack. Finally, the inter-community management following a cross-layer approach enables the creation of a network operator through the support of interaction between different layer communities.

The delegation concept described can also have a large impact in the support of new business models. As an example, one community can sell network resources (e.g. bandwidth) to another community, which contains specific delegation roles to enable the reselling of these (or a set of) resources to different communities. This can be applied, for example, to federation agreements between inter-domain operators, which define service level agreements (SLAs) between each other (the selling of resources for traffic traversing their inter-domain connections), and even end-to-end agreements which contain end-to-end resources available for a specific set of services, through the reselling of the resources to the several domains on the end-to-end path. Another example is again the support of micro-operators, whose resources can be achieved through this delegation process.

7. CONCLUSIONS

This paper presents a community-based approach to the definition of next generation user-centric communications. Nodes and users are encouraged to cooperate at all levels, sharing their resources both at the application and communication layers. The social relationships existing between users will promote increased trust in local environments, allowing wireless techniques based on individually owned APs to become a trusted communication environment. Mesh networking is particularly adequate to this approach, allowing users to roam freely, while their APs establish long-term communication backbones.

These multi-level communities can define their own policies and establish cooperation agreements with other communities – both at the same and at different levels of the communication stack.

Roles and delegations are of vital importance to community management, with all the organization being managed according to these concepts. Furthermore, such methods allow the creation of extended business models where, instead of the service providers, users and their relations are the driven forces for network operation.

Current work is focused on further refining the community architecture and the integration of QoS and mobility mechanisms. Results from prototype implementations and real world deployments, on metropolitan scenarios, will further contribute to the evaluation of the solutions proposed.

ACKNOWLEDGEMENTS

This work has been partially supported by the European Commission project IST-WIP under contract 27402.

REFERENCES

D. Banerjee, S. Saha, S. Sen, P. Dasgupta, 2005, "Reciprocal Resource Sharing in P2P Environments", *Proeedings. of the 4th International Conference on Autonomous Agents and Multi-Agent Systems* (AAMAS'05), Utrecht, The Netherlands, pp. 853-869.

E. Borcoci, E. Guainella, M. Katz, P. Neves, 2007, "WiMAX technology support for applications in environmental monitoring, fire prevention and telemedicine", *IEEE Mobile WiMAX 2007*, Orlando/USA, pp. 25-29.

L. Buttyan, J. Hubaux,, 2000, "Enforcing Service Availability in Mobile Ad-Hoc WANs", *In Proc. of the IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing.*

B. Cohen, 2003, "Incentives build robustness in Bittorrent", *In Workshop on Economics of Peer-to-Peer System.*

B. Dressler, 2006, "Self-Organization in Ad Hoc Networks: Overview and Classification," *University of Erlangen, Dept. of Computer Science 7, Technical Report.*

E. Freudenthal, T. Pesin, L. Port, E. Keenan, V. Karamcheti, 2002, "dRBAC: Distributed Role-based Access Control for Dynamic Coalition Environments", *In Proc. of the International Conference on Distributed Computing Systems*, Vienna.

P. Golle, K. Leyton-Brown, I. Mironov, and M. Lillibridge. 2001, "Incentives for sharing in peer-to-peer networks", *In Proc. of the Second International Workshop on Electronic Commerce*, pages 75—87.

Z. Haas, M. Pearlman, P. Samar, 2002, "The Zone Routing Protocol (ZRP) for Ad Hoc Networks", Internet *Draft, draft- ietf-manet-zone-zrp-04.txt.*

G. Hardin., 1968 "The Tragedy of the commons". *Science 162*, pp. 1243-1248.

Q. He, D. Wu, and P. Khosla. 2004, "SORI: a secure and objective reputation-based incentive scheme for ad hoc networks." *In IEEE Wireless Communications and Neworking Conference (WCNC 2004)*

D. Hughes, G. Coulson, and J. Walkerdine, Freeriding, 2005, "Gnutella Revisited: the Bell Tolls?", *IEEE Distributed Systems Online*, Vol. 6, No. 6.

N. Li, W. Winsborough, J. Mitchell. 2001, "Distributed credential chain discovery in trust management", *In Proc. of ACM Conf. on Computer and Communications Security*.

K. Ramachandran, E. Belding, K. Almeroth, M. Buddhikot, 2006, "Interference-Aware Channel Assignment in Multi-Radio Wireless Mesh Networks", *IEEE Infocom*, Barcelona, Spain.

R. Sandhu, E. Coyne, H. Feinstein, C. Youman, 1996, "Role-based access control models", *IEEE Computer*, 20(2):38–47.

B. Yu, M.P. Singh, K. Sycara, 2004 "Developing trust in large-scale peer-to-peer systems", *In Proc. of First IEEE Symposium on Multi-Agent Security and Survivability.*, pp. 1-10.

EMule Project: www.emule-project.net

IMDB - The Internet Movie Database (IMDb): www.imdb.com

IST-WIP - An All-Wireless Mobile Network Architecture: www.ist-wip.org

SETI@Home: setiathome.berkeley.edu

Wikipedia – The free encyclopaedia: www.wikipedia.com