Group Management in Peer-to-Peer VPNs

Florian Fuchs
Betreuer: Benedikt Elser
Seminar Future Internet WS09/10
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik
Technische Universität München
Email: f.fuchs@mytum.de

***Abstract*— The importance of Virtual Private Networks (VPNs) is still increasing at a progressive rate. Apart from the use in companies for intranets and so called home offices, VPNs are already used in the private area for file-sharing or for the secure access to home networks. Besides a centralized organization of the network, which companies usually prefer, there is also a decentralized approach through a peer-to-peer (P2P) network imaginable. P2P networks are highly scalable, extremely flexible and lack a static infrastructure like servers which makes them affordable. Peer-to-peer VPNs are therefore an easy way for private users to build their own private network. Yet with peer-to-peer networks new kinds of problems arise. The administrative tasks, in centralized networks performed by servers, have to be handled by the peers themselves. This implicates designing an identification and authentication process as well as maintaining established connections and handling the frequently changing amount of users.**

This proceeding presents and compares different group styles and current implementations in regard to how they address these problems. There are theoretical concepts of groups and applications which attempt to provide an user-friendly solution. The focus of the proceeding is mainly identifying the characteristics and the limits of the various group styles and presenting the related challenges one might face when realizing a specific style. The proceeding concludes with a short discussion about the anonymity problem which occurs when more than one VPN is set up.

***Index Terms*— P2P, VPN, group management**

I. Introduction

A virtual private network is a secure logical network which manages tunneled connections between different parties through another, usually insecure, network like the internet [1]. The purpose is to provide exclusive services for the members of the private network. For companies an apparent adoption of this idea is an intranet. Employees all over the world have access to one common site and thus to the provided applications. This enables so called home offices where the employee has no more need to be present at the office while still having the ability to use all internal services of the company over the network connection.

Besides the use of VPNs in companies there are multiple possibilities for using them in the private area. Imagine sharing photos, music or videos through just storing them in a public folder, instant messaging without the need of an external service provider or playing computer games with friends using a local machine as server.

The impulse for using a peer-to-peer network instead of the classical client-server architecture is normally the amount of disposable resources distributed over the totality of its members. Peers release available resources or provide services for other peers and in exchange use resources from these peers. A common goal usually unites all peers, for example sharing files in large P2P networks like Gnutella.

Nonetheless there is a big difference between the employment of virtual private networks based on an existing infrastructure or on a peer-to-peer network. A VPN of a company for instance is highly centralized, meaning there is a static organization which provides the administrative services needed for establishing and maintaining the network. Contrary, in the private area there is normally not an adequate infrastructure and the members of a network might change more frequently. Both, the non-availability of a central server and the demanding administrative effort, have to be faced when designing a peer-to-peer VPN.

How to identify users for instance appears as a new challenge due to the lack of a central entity where peers can easily register themselves. New, until now unknown, users want to become a member and for security reasons there has to be an authentication process. What happens when active members quit their membership since they may have performed tasks for the group and provided a part of the collective knowledge?

All mentioned problems can be summarized as the management of a group, while the group denotes all current members of the decentralized network. In the following proceeding these problems will be addressed. The first part describes general characteristics and challenges of peer-to-peer VPNs while in a second part different group styles will be introduced. In the third part different implementations of peer-to-peer VPNs will be presented and how the group management is handled in each of them. As examples serve the SocialVPN application [2], the IgorVPN application [3], the Layer Two Peer-to-Peer VPN approach by Deri et al. [1] and the ELA approach by Aoyagi et al. [4]. Finally, there is a short discussion about the anonymity problem which is highly related to decentralized networks.

II. Characteristics of P2P VPNs

There are different types of networks depending on the degree of decentralization. Segmenting the types in three

parts, namely the centralized, the semi-decentralized and the decentralized network seems easy and works fine for the presented concepts. The main difference between them is the support through an external infrastructure. The centralized network was already mentioned in the introduction when talking about intranets and is not of our interest now. The distinction between the semi-decentralized and the decentralized network is quite useful since even the smallest support from an external entity might make things a lot easier. The terms decentralized network and peer-to-peer network will be used synonymously. Furthermore challenges related to decentralized networks can be classified in two categories.

Firstly, users of decentralized networks have to discover each other and subsequently a connection between them has to be established. In general, there is no need to know the identity of the other users in private networks as long as the members trust each other. In centralized networks a user trusts the superior authority and the authority itself is the contact point for new users. In contrast, in a decentralized network both, locating and the authentication of the members, have to be organized somehow and by someone. There may be more than one single response point for the users of a network as we will see later in detail.

Secondly, once a network is established, it has to be maintained. New users probably want to join the network and this raises the issue by who and in which manner new users are approved or denied. What happens in a situation where a member should be excluded from the network? Since there is usually neither a trained administrator nor a distinctive intention of the users to manage the peer-to-peer network, the administrative tasks should be minimized somehow.

Due to the lack of a centralized technical infrastructure in peer-to-peer networks, there are normally no fix costs. Therefore a P2P network is usually really cheap to maintain and fast to establish. The potential members just need to identify each other and then establish a connection, assuming there is a proper software providing this service. For a P2P VPN the minimum amount of members is obviously two whereas theoretically no limit for the amount of participants exists. In practice huge numbers of peers do matter due to the data overhead which will probably result when the number of peers increases [5], [6]. This shows the urgent need for an adequate management of the peers in a group. This flexibility makes a P2P network indeed easily scalable, a fact large P2P networks like BitTorrent or Gnutella are using [7]. It is important to notice that the VPNs intend to establish whole networks and not only supporting single services like file-sharing.

In a company the potential members of a VPN are easy to identify, the distribution of keys seems obvious and usually there is already a technical infrastructure which can be used for the network. Technical infrastructure suggests not only the presence of servers and available IP addresses yet furthermore technical workforce with the necessary knowledge.

The downside of using P2P connections in VPNs is their vulnerability. There is not only a risk from outside the network like in centralized networks yet also from the inside. One peer might have a hidden intention and, depending on the organization of the group, also the power to impair other peers [8]. This might leave a feeling of insecurity between the members of a group. Hence, the solution of this challenge is essential when one considers using a P2P network as fundament for a VPN. Thus the group management has to include mechanisms which are able to handle these threats.

There are already theoretical models about trust management [8]. The different approaches range from ranking other members to applying the context of the situation to every trust decision. Trust can thereby be represented by a simple decision or by different values on a whole scale of trustability. The trust decisions can furthermore be kept secretly or spread into the group while adopting the trust decision of another member also implies that one trusts this member totally. There is also the overall question how much every node in the network knows about the other nodes. If only direct neighbors are known and the whole communication is handled by them, the trust management should be really simple. Yet with a growing number of connections the required knowledge about others increases and the management activities may use unjustified resources. This can be countered through simply collecting information about bad behavior, namely creating a black-list, and distributing the information over the network. The decision should always be based on the success of a previous action. The question by whom the outgoing of an action is determined is indeed still unsolved. Should the decision be an automatically proceeding process or a process requiring user interaction? Implementation of different approaches of trust management depend extremely on the characteristics of the group and the decision should be made individually in every situation.

While the trust management can be classified as an internal problem of groups, peer-to-peer VPNs also face several impediments from the technical infrastructure. Since the IPv6 standard is not fully implemented yet, applications have to handle connections using the older and with a smaller address space equipped IPv4 standard with all related problems. Most computers are behind firewalls now and already in a local network managed by a router which handles the network adress translation (NAT). The routers may block due to security issues every incoming connections while allowing outgoing. Even though there are already technical solutions to these problems, the group management part of an application plays a major role [9].

Most users of P2P VPNs are no network administrators and the configuration of an application should therefore be as easy as possible. For that reason most applications use zero configuration networking (zeroconf) where the network is set up automatically. Once the network is deployed, the user should also not be overstrained with decisions affecting the group management. In the following section different group types will be presented and some of them require the user to be active. The implementor should be aware that not every user wants to be that active in a group.

III. Group styles

When defining group styles, the point of view has to be determined. This paragraph cares mainly about the management part of a group. This means how the group can be described as an organizational entity. Hence the access to the group and how it can be controlled seems to be a promising approach due to the fact that also the leaving of a member, especially when the member is active in a hierarchy, has an influence to the approval process of new members.

In this part different groups styles will be presented, based on the work of Salzeder [3]. Beginning with the easiest one, occurring problems will be identified and the need for more robust ones will emerge. The first challenge a group of peers which decide to create a P2P VPN might face is how to seperate themselves from other users of the same network they are all in.

A. Paradise

To start with the simplest style, the peers build a group by simply choosing a name for that group or more clearly an unique identifier. A new user becomes a member of the group by adopting the name. This fact is in the context of VPNs impractical and in general strongly idealized. Only in a network where every user trusts everyone else and all users stand to the rules the so-called ”Paradise” seems to fit. Figure 1 shows a group as a part of a set of peers. While in this group all users are connected among themselves the single peers interact with different other peers outside the group. A more expedient way needs security against unwanted peers for the group.

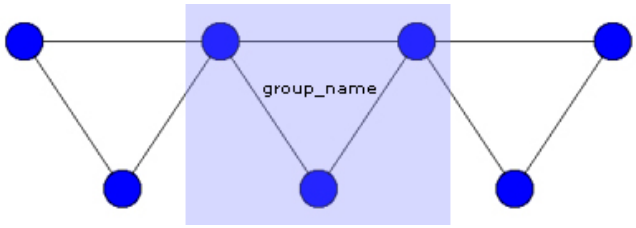

Fig. 1. Paradise: Forming a group

B. Password protection

Let’s ignore internal threats for the time being and focus on security against other users. The apparent solution for securing the network against unwanted members is to define a password for the group. Therefore the group is still identified by the group name yet the password is needed to enter it. The password can either be set by the founder of the group or in collaboration between all potential members. The security issue about the access to the group seems fixed. New users need only the password to become a member of the group as Figure 2 depicts and use the password to identify themselves.

Once a peer has the password it appears difficult to exclude or ban a member from the group. A new password has to be set and spread between all members except the one which should be excluded. Since there is usually no central entity in a P2P network the delivery of the new passwords is the duty of several members. All these members need to be aware which member is designated to be banned. Otherwise this member might be in the position of getting the new password through which the whole action fails. In conclusion, a password protection seems only helpful when the group has a static consistency and is not durable. For this reason the type is also called a ”temporary group”.

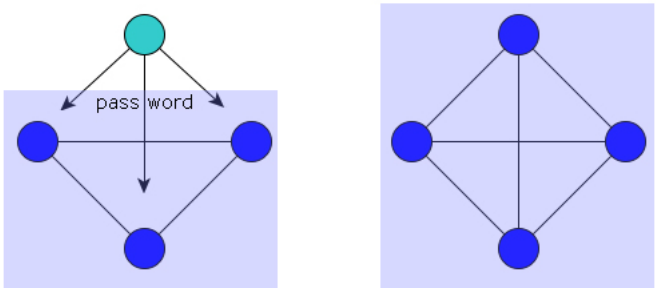

Fig. 2. Password protection: Using a password to enter the group

C. Monarchy

One might think about defining one or more entities inside the group performing tasks like setting new passwords, respectively banning members from the group. Assuming one member is assigned for this kind of tasks the group is called ”Monarchy”. The rights of the monarch might vary depending on the responsibility. If the monarch also maintains the right of approving new members to the group even the password is not needed anymore. How such a situation may look like is indicated by Figure 3. Still two challenges remain unsettled. At first, what happens if the monarch leaves the group? Is there a hierarchy which member inherits the position? And if so, how is it designed? Secondly, hypothesize the monarch itself is the one to be banned from the group. The latter seems unsolvable since there would be a need for complex control mechanisms.

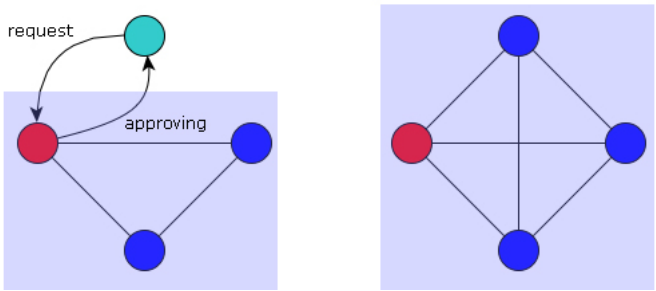

Fig. 3. Monarchy: admission process

D. Supporters

Two solutions might address these challenges. Next to the monarchy a whole hierarchy of responsibilities could

be implemented. Yet this seems to fix only the succession problem and is therefore not otherwise specified. Imagine instead separating the members of a group in different sub-groups. The easiest case are two groups, divided through the distribution of rights. One group contains users as they are also in the prior types and the other group contains members called ”supporters”. Figure 4 illustrates an example of two sub-groups. How many connections between the members exist depends on the purpose of the group. Nonetheless it appears helpful if all supporters keep connections among themselves. A non-supporter needs theoretically only a single connection to a supporter to keep connected to the group.

Supporters are similar to monarchs apart from the fact that there are several supporters while there is only one monarch. The rights of one supporter may also vary depending on the group and it seems useful that every supporter maintains the same rights. Rights could be approving new users, banning existing members or appointing new supporters. When setting up a group with supporters problems like how many supporters are needed to provide support at any time and how to become a supporter have to be solved. Yet the central question of this style is whether a supporter has independent rights or does his decisions rely on the approval of other supporters?

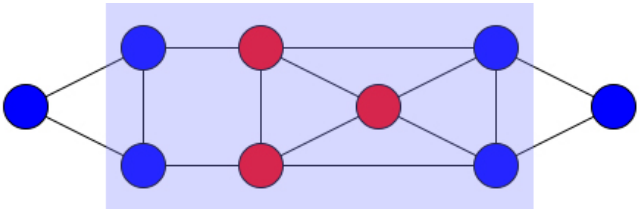

Fig. 4. Supporters: red circles indicate the sub-group supporters

E. Voting

As already discussed, decisions can be made independently or in common. If voters make a decision together, it is not as important to differentiate between supporters or normal members. One of the central issues of this style, called ”Voting”, is instead how the base should be designed since in networks not all users are available at all times. One apparent example is the time difference between regions. If one decides to count only online users there might be a situation where a small group of members gets unjustified power over other users. In contrast, if one requires too many members for a valid decision there might be a lack of decisions when not enough members are available.

Nevertheless this style is extremely flexible. Decisions can be made on a relative or an absolute basis. Relative means that a determined minimum of approvals are needed to accredit a task. Figure 5 shows a group controlled by supporters with a voting mechanism. A peer wants to become a member and therefore sends a membership request to a supporter of the group. The supporter forwards this request to other supporters and counts the responses. Once a minimum of approvals or denials is counted, the original supporter gives a reply to the potential new member. In the example the applicant has no majority and the access to the group is therefore denied [10].

As already mentioned this system is flexible and this example serves only as one of many. Yet it reveals one more problem, the need for a lot of traffic, especially if there are no supporters. The more democratic the voting system is, the more messages need to be sent between various members. Hence, the whole system gets slow and complex if not just one member manages the process. This might happen since in a large group not every member maintains a direct connection to every other member.

In the paper by Saxena et al. [7] about Threshold Cryptography using certificates for access control is proposed. A threshold results if some of the members play against the rules of the group. Therefore the cryptographic keys are distributed among the members of the group and decisions are made by mutual consent. Since every member possesses only a share of the key a specified minimum of group members is needed to sign messages for the group by contributing their share of the key. This prevents a single member from cheating.

This system can also be used for the access to a group. New members ask trusted members of the group for a part of the shared secret. Every member which approves admission sends this part to the potential member. Once the applicant has enough parts he is able to reconstruct the membership certificate. Using the certificate the new member is able to sign messages and prove membership to the rest of the group. Nonetheless in this method the question how to exclude a present member is also apparent. The approach does not provide a solution for this kind of problem yet.

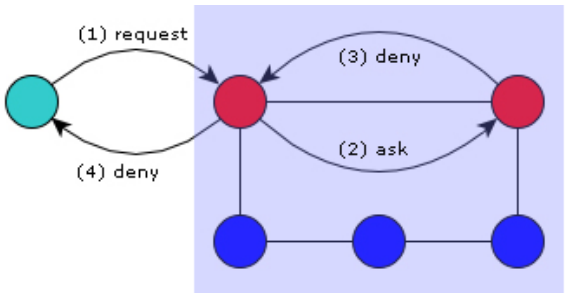

Fig. 5. Voting: Denying a request based on supporter voting

F. Anarchy

As seen in the last section, a member may have the intention of cheating. Therefore the group management should support mechanisms to prevent situations where risks for other members emerge. In a real network it will probably not be possible to suppress all misbehaviors. Large amounts of members also intensify the problem since the complexity increases according to the group strength. With cheating every violation of group rules by one or multiple group members is denoted.

A violation can be as simple as the deliberate rejection of a connection to another honest group member. Figure 6 shows

the situation where the peer labeled with an ”A” blocks every connection to the peer labeled ”B”. From a technical point of view this is easily realizable through an packet filter on the ethernet level. Nevertheless it destroys a part of the group structure and it seems hard for other members to detect this disruption of the network. Other peers might still rely on the existence of the connection while it is already blocked for a long time.

We call this situation ”Anarchy” yet it is obviously more naturally emerging than an proposed group style. It is proposed since it may arise very fast. As already mentioned peer-to-peer networks are characterized through a frequent change in memberships. For this reason new connections between peers are established and existing connections terminate generally with a higher frequency. Thus the group has usually a dynamic character and the behavior is hard to predict.

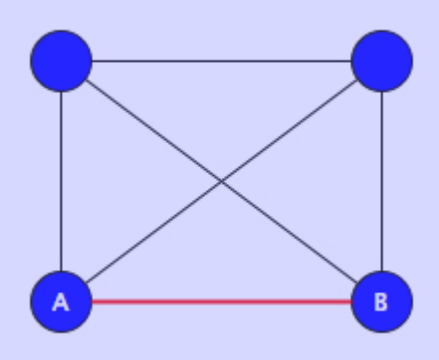

Fig. 6. Anarchy: Suppression of a connection between two peers

G. Web of trust

Until now the trust decision was either made completely by every peer itself or by a higher authority, namely a monarch or a member of the supporter sub-group. It is also imaginable that the authentication process is designed in a way that either the user itself or another peer certifies other peers. Therefore a peer has a group of other peers he trusts completely. This means he trusts in their decisions concerning, for instance, the approval of new group members. This adds transitivity to the trust model and is called a ”Web of Trust” [11] . Figure 7 depicts this graphically. Peers A and B trust the other peers and accept their certification of peer C. For that reason they establish a connection although they have never authenticated peer C by themselves. The certified connection is displayed by the dotted line.

While transitivity knows no limits, it is advisable to employ it for only one level. That ensures that only direct neighbors are able to certify for the peer and no net of complex linkages arises. Nonetheless users of a Web of Trust should be aware that a chain is only as strong as its weakest member and that they may thus suffer from failings of their trusted neighbors. Once a user has the certification of one trusted member it is easy for him to get additional certifications. Following this pattern the dishonest member may reach uncontrollable influence in the group and finally impair other peers using his achieved power. Even more dangerous is a situation where this member acts as a trusted node for other peers. Then he is also able to introduce new unwanted member to the group.

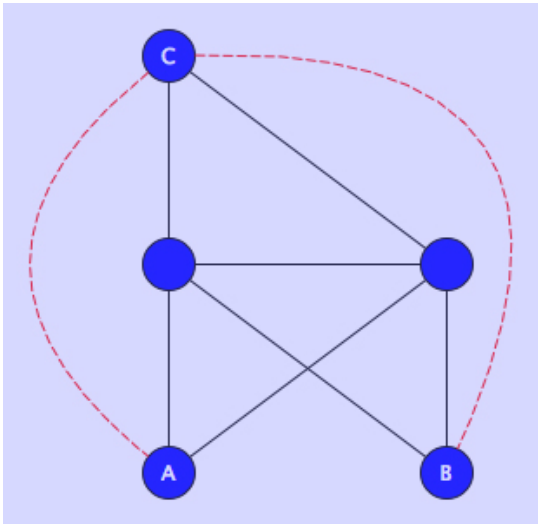

Fig. 7. Web of trust: Adding transitivity to the trust model

IV. Implementations

There are several different implementations of peer-to-peer VPNs next to the established, yet centralized, VPN applications like the Cisco VPN clients or OpenVPN. In the following, four examples will be presented which handle the raised challenges in different ways. The first example is the SocialVPN application [2], followed by the IgorVPN application [3]. Other promising approaches provide Deri et al. with the Layer Two Peer-to-Peer VPN [1] and Aoyagi et al. with the Everywhere Local Area network [4].

A. SocialVPN

The SocialVPN application [2] uses a social network as communication point to identify groups. It is therefore apparently not a fully decentralized P2P network and at the moment only the Facebok API serves as example for the support through a social network. For the selection of networks the ability to authenticate users, the possibility to query relationships and exchanging the cryptographic certificates is necessary. All this features are already provided by the Facebook API which made it the favorite option.

A group in the application is represented by the relationships between the user of the application and the peers on the other side of every relationship. Thus, the group is exactly tailored to every single user. Peer-to-peer networks, and especially P2P VPNs, can therefore be seen as an entity representing social structures and as a gate for fulfilling the needs members of the group have.

Using one group per user makes the SocialVPN application even more interesting since the center of an actual group is the user itself. Every user has its own group: the relationships

transferred from the social network. That derives from the fact that at every end-point of the network a virtual IP namespace exists managed by a virtual VPN router. The used technique for this is Brunet for the connectivity between the peers even behind NATs and IP over P2P (IPOP) which provides the fundament for the application to communicate with local IP addresses over a virtual network interface [9].

Which group the user of the application belongs to is defined by all relationships in the social network. At first it might look like the implementation of a Web of Trust. In a Web of Trust there would be only one group yet in the SocialVPN every user has its own group. An overlapping is theoretically only possible if two peers maintain exactly the same relationsships. Nevertheless the peers act in different virtual name spaces.

To establish a connection to another user obviously both have to use the same software. There is no chance to become a member of the group as long as there is no corresponding relationship in the social network. This seems to solve the security issue to the inside since the identification of users is shifted to the social network. It is implicated that the user trusts the users to which he maintains relationships in a social network and the social network is a convenient way of managing these relationships.

The security against breaking into a group is obviously only as strong as the security of the social network. The Facebook API is also used to exchange certificates in a secure way and it is therefore absolutly inevitable that the user trusts the social network. The established connections between the members in the network are then protected by IPsec and public key cryptography (PKI).

Every user of the application has also the possibility of blocking other users. This means that they are not able to use any services provided by the user even though they are in the group. The application supports this by a simple checkbox for every single member of the group.

The application is also able to support multiple computers of one user. A group may therefore contain various devices by every user. Virtual name spaces are used to alleviate the communication between users as much as possible. The name consists of the names of the machine, the users first and last name, the social network and ipop as suffix. An example would be *homepc.lisa.a.facebook.ipop* or *homepc.lisa.adams.facebook.ipop* depending on whether the use of an initial for the last name is already unique.

Figure 8 shows some aspects of the SocialVPN application graphically.

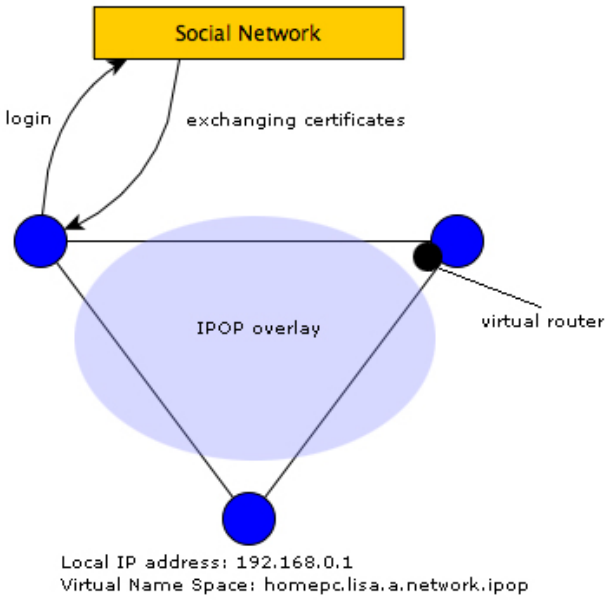

Fig. 8. SocialVPN

B. IgorVPN

The IgorVPN application [3] is a fully decentralized peer-to-peer VPN. Three different group management handler are implemented, namely for the described cases ”Paradise”, ”Temporary Group” and ”Monarchy”. Every type needs a group ID and has the optional element of a group name.

The identity keys of the group members are stored in a distributed hash queue (DHQ) which enables the communication between peers. In *Paradise* a new member needs only to be included in the DHQ to become a member of the group. Then the keys get regularly polled by the members of the group. Thereby all members are able to find everyone else in the group. The drawbacks of that simple mechanism were already described and the author recommends the use only for demonstration purposes.

In *Temporary groups* the group is identified by the group name, respectively the group ID, and a password. The password is set by the founder of the group and it is used to encrypt the entries in the DHQ. In every other aspect the temporary group is similar to a paradise group.

In *Monarchy* one peer has to found the group and acts therefore as the monarch. The only way to enter the group is by an invitation of the monarch. The invitation creates an entry in the subset through which the invitee is able to join the group. The monarch also has the exclusive right to ban members from the group. This is exerted by removing the entry from the subset. An implemented enhancement of the monarchy style is the possibility to assign trust points. This alleviates the occurring delay of distributing updates about banned members since returning members contact always the most trusted member before revealing their own identity to all group members.

C. N2N: Layer Two Peer-to-Peer VPN

The network-to-network (N2N) [1] approach is a fully decentralized peer-to-peer VPN as well. The application aims at providing full network support with all its advantages instead of application support like file-sharing. The members of the network are referred to as a community yet the meaning is the same as the term group used in this paper.

There are two types of members in the group, called Edge Nodes and Super Nodes, as described in Figure 9. This sounds

similar to the concept of supporters, presented as one of the group styles. The Super Node acts thereby as a mediator. Every Edge Node has to register itself at a Super Node. The Super Node has now the chance to manage the connections between Edge Nodes where NAT hinders a direct connection setup between the nodes. Even though a peer is protected by a NAT router he has the chance to get an arranged connection by the Super Node which registers him to the other peer. This detour also makes sense in a way that the trust issue is forwarded to the Super Node which may have more information about the members of the community than a single Edge Node.

A list of solid paths to Edge Nodes is saved and used to help asking Edge Nodes to communicate to other Edge Nodes. This enables the Super Nodes to act like a virtual network infrastructure. The original data is not sent through the super node since it has only an administrative character. This prevents this approach from excessive data overhead.

The N2N approach provides no information about the access control at the moment, except for the fact that there is an invitation process. Yet it shows how support from the inside of a group could be implemented and how supporters have to be designed to perform such support. Another feature of this approach is that every user can be a member of various groups since the application favors the establishment of connections more than the direct offering of services like storing data by every group member. Therefore each host maintains a virtual network interface for every network it is connected to.

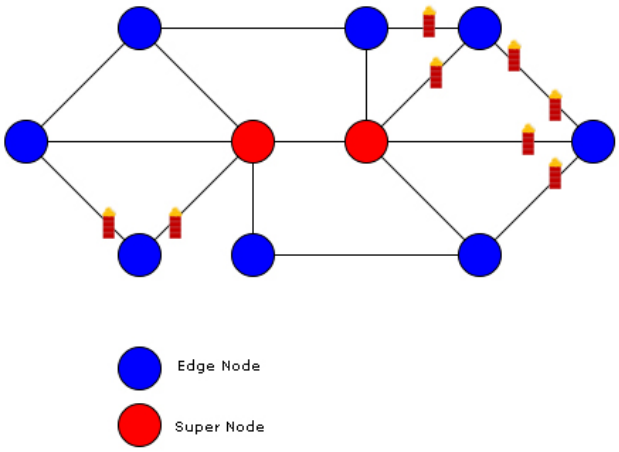

Fig. 9. N2N: Layer Two P2P VPN

D. Everywhere Local Area network

Like the IgorVPN application and the N2N approach the Everywhere Local Area network (ELA) is a fully decentralized peer-to-peer VPN as well. The approach also distinguishes its members in two groups yet with another differentiation. The two groups are called Core-Group and Edge-Group and the members are either denoted as Edge-Node or Core-Node depending on to which group they belong. The separation is based on wether the node is able to handle TCP and UDP connections or only TCP connections as tunnels. The use of UDP may be limited due to NAT routers or firewalls in the local network of the node.

To become a member of the network the new node must know at least one existing member of the group. After sending a request to this node the new peer gets classified and a randomly distributed IP address by the Network Pseudo Device which is a part of the ELA-VPN application.

While Core-Nodes are connected among one another, the Edge-Nodes maintain only one active connection to a Edge-Node. For backup there is a second connection to a different Core-Node yet it will be active only in the case the first Core-Node terminates its service. Hence, every Edge-Node has only one direct contact point to the inside of the group. This looks similar to the supporter group style with a full range of power for the core-nodes since they control the access to the rest of the group for the Edge-Nodes.

This concept is similar to the approach about information retrieval in third generation peer-to-peer architectures described in [6]. There a separation between leaf nodes and ultrapeers enables an efficient content distribution. Leaf nodes are connected to an ultranode while ultranodes are also connected among themselves. While lead nodes can only be a member of one group, called a club in the paper, an ultranode is used to connect the groups among each other.

In the ELA-VPN the whole routing process is based on the Core-Nodes which means that the Core-Nodes have to care about the destination of IP packets. The End-Node just forwards their packets to the Core-node.

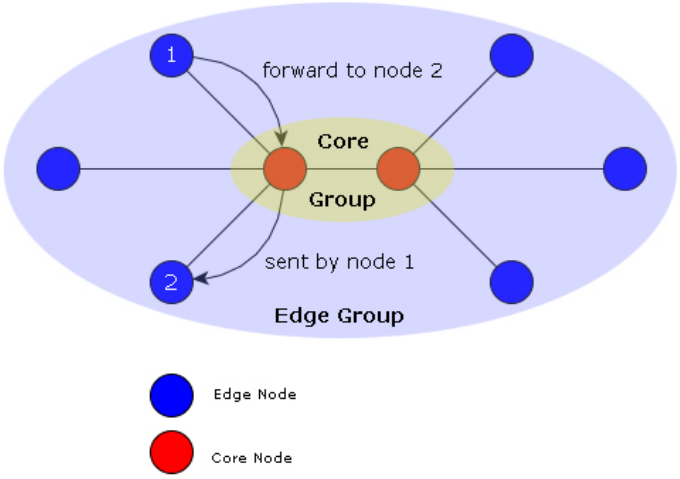

Fig. 10. Everywhere Local Area network

V. Anonymity

The prior section described several implementations of peer-to-peer VPNs. The focus of the presentation was on how they implement groups and how the groups are managed. Thereby the view was mainly defined by the existence of only one peer group and one virtual private network.

Imagine a peer is a member of more than one group like the situation described in Figure 11. The arising challenge is separating the traffic and the services provided by each single group. Otherwise there are high security risks for all members of all groups since they do not know anything about the members of the other groups. Therefore the group management of an application needs mechanisms to protect the user against the risks of this situation. As this is really hard most applications do not support multiple group membership yet.

In the SocialVPN [2] application this problem is solved quite elegantly. Since one user has only one group of relationships there is no need for a second group. At the moment the application supports only the Facebook API and it is not clear how it will be handled mixed up with other social network sites. Yet maintaining only one group even if it is mixed through multiple social networks seems a durable solution for this problem. Since every user manages his own local group the origin of a connection is only important for security issues.

The N2N application supports the membership in various groups as already mentioned. In the IgorVPN application the support is not implemented yet while there is no information at the moment how the ELA-VPN application handles multiple group membership.

Two different challenges have to be kept apart in the context of anonymity. *Firstly*, looking at a specific member of the group, there are services which the peer offers to other members of the group. This services might be providing public folders for file-sharing, a web server or other applications like games to which other users can connect. Two different problems may arise in this context. The first is that one service is only for one group while another is only for a different group. In general, the services have different addressees. The second problem is that one service should be offered to both groups yet with different content. *Secondly*, members of the same group might be in the position of asking for services. Therefore, the user has to assure that this user is not able to observe all provided services yet only the ones which are intended for his group.

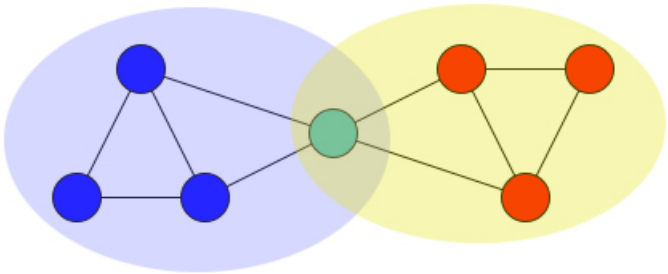

Fig. 11. Anonymity

VI. Conclusion

The contribution of this proceeding was the detailed presentation of different group styles and the evaluation of related implementations. As inspection depicted there are already sound applications which use a variety of techniques to solve the raised challenges. While most applications address all the related technical problems important group mechanisms remain unsettled. The SocialVPN application appears the most promising approach even though it is only a semi-decentralized approach. There is a clear authentication process supported by a trusted external entity and still the ability to block users. This seems an easy way for private users without network skills. The other presented concepts provide interesting approaches to the issued challenges related to group mechanisms. Also the Threshold Cryptography [7] as a concept to implement voting in a group appears promising. Yet an application which implements this technique in a robust technical environment is still missing. The development of peer-to-peer based virtual private networks seems still to be in its infancy. Most concepts are more a prototype for a specific problem than an ready-to-use application. Only the SocialVPN application seams already suitable for non-technical users.

References

[1] L. Deri and R. Andrews, "N2n: A layer two peer-to-peer vpn," in *AIMS '08: Proceedings of the 2nd international conference on Autonomous Infrastructure, Management and Security*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 53–64.

[2] R. J. Figueiredo, O. P. Boykin, P. St. Juste, and D. Wolinsky, "Social vpns: Integrating overlay and social networks for seamless p2p networking," June 2008. [Online]. Available: http://byron.acis.ufl.edu/papers/cops08.pdf

[3] M. J. Salzeder, "Using fully decentralized peer-to-peer technologies for virtual private networks," Diplomarbeit in Informatik, April 2009.

[4] S. Aoyagi, M. Takizawa, M. Saito, H. Aida, and H. Tokuda, "Ela: A fully distributed vpn system over peer-to-peer network," in *SAINT '05: Proceedings of the The 2005 Symposium on Applications and the Internet*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 89–92.

[5] D. Bin, W. Furong, and T. Yun, "Improvement of network load and fault-tolerant of p2p dht systems," in *Information Technology: Research and Education, 2006. ITRE '06. International Conference on*, Oct. 2006, pp. 187–190.

[6] A. Asvanund, R. Krishnan, M. D. Smith, and R. Telang, "Interest-Based Self-Organizing Peer-to-Peer Networks: A Club Economics Approach," *SSRN eLibrary*, 2004.

[7] N. Saxena, G. Tsudik, and J. H. Yi, "Threshold cryptography in p2p and manets: The case of access control," *Comput. Netw.*, vol. 51, no. 12, pp. 3632–3649, 2007.

[8] K. Aberer and Z. Despotovic, "Managing trust in a peer-2-peer information system," in *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*. New York, NY, USA: ACM, 2001, pp. 310–317.

[9] A. Ganguly, A. Agrawal, P. O. Boykin, and R. Figueiredo, "Ip over p2p: Enabling self-configuring virtual ip networks for grid computing," 2006, pp. 1–10. [Online]. Available: http://dx.doi.org/10.1109/IPDPS.2006.1639287

[10] T. Bocek, D. Peric, F. Hecht, D. Hausheer, and B. Stiller, "Towards A Decentralized Voting Mechanism for P2P Collaboration Systems," Department of Informatics, University of Zurich, Tech. Rep. ifi-2009.02, March 2009.

[11] A. Datta, M. Hauswirth, and K. Aberer, "Beyond "web of trust": enabling p2p e-commerce," in *E-Commerce, 2003. CEC 2003. IEEE International Conference on*, June 2003, pp. 303–312.