CRATOS
Center for Research on the Applications of Telematics
to Organisations and Society

# Internet-based Secure Virtual Networks

*L. Delgrossi, D. Ferrari*

{ldgrossi, dferrari}@pc.unicatt.it

Quaderni del CRATOS

*Serie di Telematica*

CTR-T97-003

Università Cattolica del Sacro Cuore – Piacenza, Italy

**Abstract**

In previous works, we have introduced the notion of *supranet*. Supranets are virtual networks - private to a group of users - that can be built on top of a physical network (e.g., the Internet) by any user of such a network making use of an appropriate software toolkit. We expect supranets to be used in the future by many different types of groups, each with different requirements in terms of security. This paper discusses the ideas on which supranets are based as well as the security issues to be considered when designing supranets on top of the Internet.

## 1. Introduction

In previous works [1], [2], and [3], we have introduced the notion of *supranet*. Supranets are virtual networks - private to a group of users - that can be built on top of a physical network (e.g., the Internet) by any user of such a network making use of an appropriate software toolkit.

Supranets have been designed to offer groups of users an extended set of services: for instance, when creating a supranet, a user may fully control the topology of the network and the communication paths along which the messages are sent, restrict access to the network services to group members only, dictate rules of behaviour and impose sanctions to those members who violate these rules. In short, supranets are created, exploited, managed and deleted by their users, and this allows these users to tailor the communication environment they use to their specific needs. We feel that this ability will represent an important addition to the services offered by today's networks.

Supranet members should feel they are part of a secure private environment from which non-members are excluded. This calls for the provision of security mechanisms that protect supranet communications from the outside world. On the other side, supranet members are likely to be organised into a number of user classes, each corresponding to a predefined set of privileges; also, it may be necessary to provide the means to protect some or even all of the conversations among supranet members. This means that appropriate security mechanisms have to be provided inside a supranet as well.

We expect supranets to be used in the future by many different types of groups, each with different requirements in terms of security. In [1], we have described a business scenario with a supranet connecting a broker with his clients; in [2], we have presented a number of potential supranet applications, including a supranet-based auction sale. Since it is necessary to meet very different needs, one of our design goals has been to allow the users to choose the security level they is feel more appropriate in each case. This can be done, for instance, by specifying appropriate options at supranet construction time.

This paper discusses the ideas on which supranets are based as well as the security issues to be considered when designing supranets on top of the Internet. Section 2 describes the fundamental principles and requirements on which the design of virtual networks is based. Section 3 presents some important issues in supranet design. The design of supranet security mechanisms is discussed in more detail in Section 4. Finally, Section 5 concludes this paper by summarising the main results.

## 2. Internet-based Virtual Networks and their Requirements

Virtual private networks (VPNs) have existed for a number of years. The typical VPN is a network built for a large corporation or institution, uninational or multinational, by a telecommunications company. A VPN is usually a cheaper surrogate of a private network (PN): instead of setting up an *ad-hoc* network for organisation X, a telecommunications company allocates some of the resources of its existing network (or networks) to X and organises them so that they will give X the illusion of having its own PN, while the physical network is actually shared among other organisations and other users. A VPN is able better to approximate the ideal if, besides providing addressing and routing characteristics similar to those of a PN, it protects its users from intrusions, leaks, wiretappings, impersonations, and other breeches of security to the extent they would be protected against these threats in a PN.

To our knowledge, the Internet does not, at the present time, have anything like this, nor has such a facility been proposed for it yet. We have investigated the feasibility of a VPN implementation on top of the Internet, and obtained positive results. However, much more important than these results is the realisation that that the approach to building a VPN in an Internet-like environment ought to be totally different from the one traditionally followed in the context of telecommunication networks.

First of all, the need for a VPN is not confined to the world of large organisation: the only reason VPNs have been set up exclusively for such organisations is that smaller groups could never afford the steep prices of VPNs. The Internet is much less expensive; in particular, it makes VPNs affordable even to associations, clubs, small businesses, and so on, since packet switching technology exempts the network from assigning dedicated physical resources to each VPN. We believe that the usefulness of VPNs for all of those groups of individuals, small organisations, and mixtures of both will be recognised as soon as inexpensive VPNs start being regarded as feasible. In fact, extranets and distributed intranets are examples of VPNs whose desirability is no longer doubted by anyone.

Another observation we have made is that the traditional model of VPNs and VPN creation suffers from at least the following problems when transferred to an Internet-like context:

- many of the groups that would benefit from the availability of inexpensive VPNs have dynamically (and sometimes rapidly) changing membership, objectives, organisation, and requirements; some of them also have relatively short lifetimes; asking a telecommunications company or another service provider to make all the changes that are required is neither cheap nor, in some cases, fast enough;
- the Internet and Internet-like networks are fundamentally anarchic systems; there are no central authorities, no centralised controls, no laws, whereas groups need VPNs over which they can have full authority and control;
- in the Internet philosophy there are no privileges, no services that are reserved to certain classes of users; if VPNs can be built on top of the Internet so inexpensively that all users can afford them, then all users should be allowed to build their own VPNs.

These arguments suggest that there ought to be a VPN creation, management, and deletion service available to all Internet users for their direct use. The service could consist of a software toolkit, which could be free or licensable at a reasonable price. Users who do not wish to use it themselves could always, of course, get a third party to do it for them. The user or set of users who sets up a VPN for a group would retain full authority and full control over the resulting VPN. We call this individual or collective entity the *creator*, and allow its role to be transmitted to others as many times as necessary during the lifetime of the VPN.

Not all virtual networks satisfy the needs for controlled and secure communication that most groups requiring a VPN have. For instance, the Mbone is a virtual network, but cannot fulfil the role of our VPNs. What are the specific requirements of these VPNs?

We have examined a large number of different groups, and found that all their requirements fall into the following six types:

- Membership: only "members" should be allowed to access the network, i.e., to make use of the available services. Restricted membership implies the existence of an admission control policy, possibly enforced by a central authority or committee ruling the network. Members may belong to one of several classes, differing from the others in privileges and obligations.
- Topology: the paths (consisting of nodes and links) that connect members in the network may be explicitly designed and tailored to specific user needs. This provides the ability to decide what paths should messages follow for security, control, or other reasons. It is desirable that the network's topology be dynamically adjustable based on new needs or convenience.
- Capacity: as for paths, it should be also possible for the network designer to define the capacity of the resources involved in the communications. Such resources include the bandwidth of the links, as well as the CPU power and memory size of each node. Resource management functions that enable a dynamic adjustment of the resources' capacity are also desirable.
- Connectivity: this requirement consists of the ability to control which services a member is allowed to benefit from and which other members he can reach and communicate with. This allows the network authority to "hide" services from a member, e.g., when he does not possess appropriate privileges, or to hide the presence of other members, e.g., for confidentiality reasons. Ideally, it should be possible to dynamically establish or tear down connectivity among members.
- Multicast: this is the ability to send messages to a subgroup of the members in a secure manner, so that the messages cannot reach a member which is not included in the subgroup. As an alternative, a message reaching an undesired destination should be useless, e.g. indecipherable, for that destination.
- Security: communications among members may need to be kept private. In such cases, both the intrusion of outsiders and the eavesdropping by other members must be avoided. Network users may need to be guaranteed on the identity of their interlocutors. Different applications may have different requirements: some may request services that guarantee the authentication or the anonymity of the sender, others may need mechanisms that help avoid the denial or the repudiation of service.

We felt that VPNs that satisfy requirements of these types and can be created, managed, and destroyed (using a suitable toolkit) by any user of the Internet-like network on top of which they are built needed to be identified by a new term. We chose the term *supranet*, which, based on the Latin preposition "supra", i.e., "on top of", refers to the overlay characteristics of all VPNs, hence also of those indicated by the term.

In the next section, we briefly discuss some issues in supranet design, with the only exception of the security-related issues, which are the subject of the subsequent section.

## 3. Supranet Design Issues

Supranets are exploited by groups of users connected to a physical network (e.g., the Internet). In general, it is necessary to modify the structure of the underlying physical network to embed support for supranet services. In the Internet architecture, this can be achieved by inserting a "supranet layer" between the traditional transport and network layers, as sketched in Figure 1. Future physical networks, we believe, are going to be supranet-capable by design.

The solution in Figure 1 requires the addition of a supranet layer on top of IP, and the installation of its modified or supplemented version on supranet hosts and routers; it does not require any changes to the applications, to the socket code or to the transport protocols, except for the simple indication that a message is destined to a given supranet rather than to the Internet.

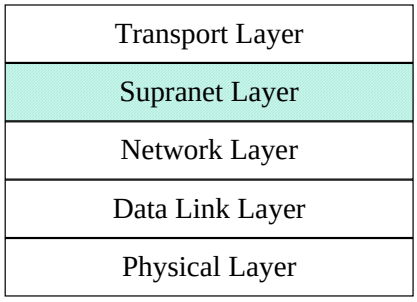


Figure 1: Position of the Supranet Layer in the Internet Protocol Stack.

In a supranet, it is possible to run all the existing applications that can be executed over the physical network on top of which the supranet was built. However, in order to fully exploit the functions provided by the supranet layer, it may be necessary to build new applications or to modify the source code of the existing ones. In this section, we anticipate the main issues to be considered when designing supranets. Please note that the goal at this stage is to discuss these issues and the design decisions that have been taken so far, rather than to provide details to be used as the basis of an implementation of the system.

Note also that many of these decisions coincide with, or are similar to, those that are found in other types of virtual or overlay networks. Supranets are novel mainly because they can be built by any user with an easy-to-use toolkit, and because they can be built on top of the Internet or Internet-like networks, where virtual networks are new.

### *Address Space*

A supranet has its own address space. Supranet addresses are assigned to supranet hosts and routers by the creator. These virtual network components are to be mapped onto physical components; a one-to-one mapping will usually be preferred for the sake of simplicity. Thus, a physical host that is also a supranet host has two addresses, a physical one and a virtual one. Furthermore, a supranet is endowed with its own name space and needs a centralised or distributed name server that translates supranet names (e.g., supranet URLs) into supranet addresses.

### *Routing*

The degree of topological and connectivity control normally desired in a supranet makes it essential for all routes to be fixed. Having fixed routes in a supranet means that, for instance, to reach supranet host 'E' from supranet host 'A', supranet packets always visit supranet routers 'B', 'C' and 'D' in this order (see Figure 2).

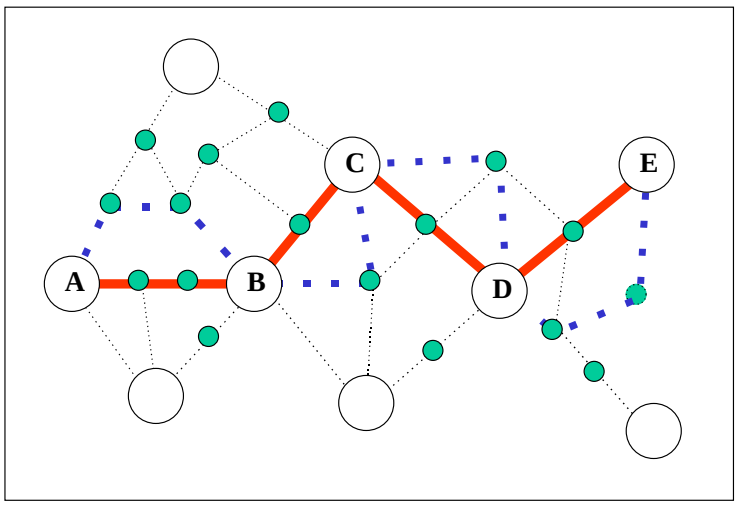


Figure 2: A supranet route connecting hosts ‘A’ and ‘E’ through ‘B’, ‘C’, and ‘D’.

In Figure 2, large circles represent supranet (and Internet) routers; small circles are Internet routers only. Note that paths AB, BC, CD and DE are supranet (i.e., virtual) links; to each link there correspond multiple physical paths.

When the network on top of which the supranet is created is, like the Internet, connectionless at the network layer, router ‘B’ may be reached from host ‘A’ through different physical routes (hence, different Internet routers) for different packets. When, on the other hand, the underlying network is connection-oriented at the network layer (or offers both types of service, as integrated-services networks ought to [5]), also the physical route will be fixed. In all cases, the supranet need not be connection-oriented, as the appropriate static routing tables generated by the creator and stored in supranet routers will be sufficient for the purposes of topology, connectivity and multicast control.

The supranet header, which in the case represented in Figure 2, is assembled by ‘A’ and examined by ‘B’, ‘C’, ‘D’ and ‘E’, must include the supranet addresses of the source (A) and the destination (E). In the Internet, supranet packets will travel between supranet hosts and supranet routers encapsulated into IP packets. Tunnelling is therefore an important technique in the construction of supranets.

### *Multicasting*

As supranets provide a controlled communications environment, it has to be possible to achieve control also over multicast forms of communications. To this purpose, the supranet creator is allowed to specify a number of multicast groups, and to assign a supranet multicast address to each of them. Since supranet membership is restricted, it is possible in general to build multicast trees that define the paths from each member of the multicast group to the others. When building such paths, appropriate mechanisms exist that can be used to minimise the overall group traffic. Every change in group membership requires a modification of all the trees pertaining to the multicast group. Additional security mechanisms, such as private keys for each group, will be used to provide security services at the multicast group level.

### *Rules and Sanctions*

In human societies, groups and associations are governed by a number of rules. In a similar way, the operation of a supranet is governed by rules dictated by its creator. The construction of a supranet requires a number of choices concerning necessity and freedom; such choices are to be made by the creator. Some actions or prohibitions will be made compulsory by the absence of alternatives. In other cases, the creator will choose to provide alternatives, set rules, and let users decide whether they wish to follow the rules or not. Those who do not will incur into well-publicised sanctions. Obviously, in order for sanctions to work, user behaviour must be monitored; actions that are both non-monitorable and desired have therefore to be made compulsory if this is economically possible. Rules are concerned with admission, access rights, use of supranet resources, relationships among members, etiquette, and so on. For instance, leaking protected information to the outside will generally be prohibited.

### *Supranet Modifications*

The creator's initial specifications for all user requirements cannot be assumed to be perfect or immutable. Changes will have to be made to them during the supranet lifetime due to the addition of new members, the departure of old members, the discovery of new needs or of mistakes in the previous specifications, and so on. The creator will have to be provided with tools facilitating all reasonable modifications of the requirements on which the supranet has been based.

In summary, functions to be included in the "supranet layer" include: definition of the address space for the supranet (both unicast and multicast addresses), creation and management of supranet-level routing tables; assembling supranet packets and forwarding them to the next supranet node on the way to the final destination; support for multicast communications; checkpoints, filters, and other forms of control to detect users' misbehaviours. Besides these functions, the supranet layer includes security-related mechanisms that are presented in the next section.

## 4. Security Mechanisms

This section completes the discussion on the main issues to be considered when designing supranets (and supranet tools) by presenting a possible approach to the design of supranet security mechanisms. The scheme presented below meets all the requirements imposed by the design goals introduced in the above sections. However, since this scheme lacks generality we expect that, in the future, it will be necessary to adapt it so that it can accommodate other encoding algorithms and single key cryptographic techniques[1]. We feel that this scheme is adequate for the purpose of the current experiments we are conducting.

The scheme makes use of cryptography and *asymmetric* keys to provide an appropriate security level. This means that, different keys are to be used for the encryption and decryption of a single message. Usually, one of these two keys is *private*, i.e., known by its owner only, and the other is *public*, i.e., known by all members in the group. Messages encrypted with a private key are to be decrypted with the correspondent public key and vice versa.

Before discussing how asymmetric encryption is applied, let us take a look at the header of a supranet packet. The supranet header includes, among others, the following relevant fields:

- *SupranetProtVers*: indicates which protocol version is being used
- *SupranetIdentifier (Sid)*: indicates which supranet the packet belongs to
- *UpperProtocol*: the identifier of the upper layer protocol
- *SendAddress*: the supranet address of the sender
- *DestAddress*: the supranet address of the destination
- *Checksum*: the checksum value calculated on the packet
- *Bitmap*: data structure that indicates a supranet user's permissions

Note that the length in bytes of some of these fields may change from supranet to supranet as it is determined by the creator's choices at supranet construction time. For example, a supranet with a strict limit of 100 on the maximum number of hosts and routers may be implemented with an 8-bit virtual address space, e.g., the *SendAddress* and *DestAddress* fields require no more than 8-bit each.

### *Restricted Network Access*

A supranet carries all the traffic generated by the communications among its users. Only members in the group have access to the supranet, while non-members are excluded. From the security point of view, it would be desirable that the physical links used for data transmission be entirely dedicated to the traffic generated by supranet members. Potentially, this would make it more difficult for an outsider to intercept, decode, modify, and forge data packets. For example, some large corporations make use of infrastructures consisting of a Private Network (PN) built with dedicated links, thus achieving a high level of protection. The drawback of this scheme is its very high cost.

Supranets, on the other hand, rely on an underlying physical network to carry the traffic generated by their users. The same physical network may simultaneously carry the traffic associated with more supranets; also, the physical network may in general carry non-supranet traffic. For example, consider

[1] To generalise the scheme a mechanism similar to the *security association* field used in the Authentication Header specifications [6] could be considered.

a typical scenario in which a number of supranets are built on top of the Internet. Since the traffic associated with different groups is conveyed over the same physical links, it is necessary to provide the means to associate each single packet with its correspondent virtual network and to set an adequate level of protection among the different communication environments.

In our design, data packets belonging to any supranets are identified by the presence of the supranet header; the *SupranetIdentifier* (*Sid*) field included in the supranet header is used to associate each packet with its correspondent supranet. From the point of view of a supranet member, there is no difference between outsiders who are members in a different supranet and those who do not belong to any supranet: all of them are non-members with respect to which protection is needed.

To provide an adequate level of protection in this kind of environment, it would be possible to encrypt all packets travelling on supranet with a *supranet private key*. The correspondent supranet public key would be known by all supranet members, e.g., distributed off-line by the supranet creator at the time each member joins the group. Note that in no cases the *SupranetProtVers* and *Sid* fields should be encrypted, because they are necessary to identify the supranet and, with this scheme, to determine which supranet public key has to be used for the decryption. The scheme provides adequate protection among logically different communication environments. However, it has some clear disadvantages:

- the supranet public key would likely be a weak key, as it would be known by a potentially large number of users and it is used very frequently,
- the encryption and decryption of the entire packet to be executed for all packets may significantly affect overall performances, and
- not all groups and not all members in the same group need always to protect all of their messages.

Most importantly, as explained in the following sections, the scheme that we have chosen to achieve sender authentication and confidentiality within a supranet works just as well for protection against the intrusion of non-members. For all these reasons, the supranet key has not been adopted.

### *Sender Authentication*

Let us now consider again the simple scenario in Figure 2, where user A sends a message to user D along the A-B-C-D path. In a supranet, the message sent by A is encapsulated into an IP packet before it leaves host A. The same happens at supranet routers B and C. The IP packets generated at A, B, and C are different one from the others because their *SendAddress* and *DestAddress* fields indicate a different sender and destination address (A/B over the A-B path, B/C over the B-C path, and so on) and because segmentation at the IP level may be needed along some of the paths.

The Authentication Header (AH) proposed in [6] can be used to provide sender authentication at the IP layer. However, AH has a strong dependence on the IP packet structure, e.g., on the sender and destination fields of the IP header. Thus, in our scenario, it would be necessary to update at every hop not only the IP header, but also the correspondent authentication header. Most importantly, the AH scheme could be used to authenticate sender A along the A-B path, sender B along the B-C path, and so on, but it would be insufficient to provide end-to-end sender authentication, that is, to authenticate sender A to the final destination D.

For these reasons, we decided not to adopt the AH scheme and to use a mechanism based on asymmetric key cryptography instead. To authenticate his messages, sender A encrypts a portion of the supranet header with his private key ($K_{priv}(A)$). The receiver uses the public key of the sender ($K_{pub}(A)$) for the decryption. Since A is the only owner of his private key, any messages encrypted with this key must have been necessarily generated by A.

But which part of the supranet header should be encrypted with the private key of the sender ? First, we have to observe that the *SendAddress* field of the supranet header should be in clear text, otherwise a destination would not know which key to use for the decryption. The *DestAddress* field of the supranet header could be encrypted with the private key of the sender, but this would not be very practical because this field is used at every intermediate host to compute the next hop on the path to the final destination. Also, since supranet addresses are virtual addresses that can be interpreted by means of appropriate tables only[2], we felt that a higher level of protection was normally not necessary in this

[2] We assume supranet routing tables are stored in secure storage areas at each node. Should this assumption for some reasons not hold, it would be wise to encrypt the *DestAddress* field of the supranet header at the price of a presumably not very significant loss in performance.

case. So, it is sufficient for the sender to encrypt the *Checksum* field of the supranet header; this also allows us to meet the data integrity requirements.

### *Data Integrity*

The value of the *Checksum* field of the supranet header, which is computed over the whole packet, should be encrypted with the private key of the sender. This guarantees that the contents of the packet are not manipulated after the sender has generated it and that packets are not forged by an outsider. Should the *Checksum* field not cover the whole packet, but for example only the header, it would be possible for an intruder to leave the header unmodified and substitute the payload, so that it would look like the new contents would have been written by A. With checksum covering the whole packet, even if an outsider modifies the payload and then re-computes the value of the *Checksum* field, he cannot encrypt this value with the private key of A.

An advantage of this scheme is that it allows the receiver automatically to detect the transmission of corrupted packets. However, a drawback is that this is normally done at a higher layer also, e.g., by the TCP protocol, thus resulting in a duplication of functions and efforts. Also, should the user explicitly ask for an unreliable service, e.g., by selecting the UDP protocol[3] at the transport layer, the supranet layer would still need to compute the value of the *Checksum* field. This would be an obvious limitation in the case of transmission of digital audio and video streams, that can tolerate a certain amount of errors, but can hardly tolerate any transmission delays. A possible alternative would be to calculate the checksum of only a portion of the payload, in the attempt to reduce the computational time required to calculate the checksum value. In this case, however, it would be necessary to impose a minimum packet size for supranet packets.

In conclusion, the detection and correction of corrupted packets are left to the higher layer protocols. At the supranet layer, for those messages that are encrypted for sender authentication purposes, a mechanism for the detection of corrupted packets is available at no extra cost. The supranet layer does not attempt to recover, e.g., by asking for packet retransmission, from the data integrity errors it detects.

### *Confidentiality*

When a message - sent by A to D - needs to be kept confidential, A must encrypt it with the public key of the receiver, in this case D ($K_{pub}(D)$). This way, only D will be able to decrypt the message by using his private key ($K_{priv}(D)$). The encryption of the payload occurs before the computation of the checksum value for the packet. In multicast conversations, group-level public and private keys can be used. This scheme protects both from external intruders and from other members of the same supranet. For this reason, we felt there was no need for additional supranet private and public keys to protect all supranet messages from non-members.

### *Sender Anonymity*

In some supranets, the sender of a message may desire to remain anonymous. A certain degree of anonymity with respect to non-members is implicitly provided by the fact that the supranet sender and destination addresses are virtual identifiers that can be related to the IP addresses of the sender and receiver only by using appropriate supranet tables, which are maintained in a secure storage area. This means that a supranet packet, even if it is intercepted by an outsider, does not contain sufficient information for the interceptor to understand who has sent it and who will receive it.

Also, it is not possible to make assumptions based on the IP addresses included into the header of the IP packet encapsulating the supranet packet, because it is in principle impossible to decide whether the sender is the actual sender of the packet or the last forwarding router, and whether the destination is the final destination or the next router. By detecting long series of supranet packets with a given supranet identifier (sid), it would be possible to guess whether a host belongs to a given supranet or not. However, the knowledge of which supranet number corresponds to a particular supranet would be required to be able to exploit this information.

To maintain anonymity within the supranet is more difficult. A possible solution is based on the introduction of an anonymity server (AS), associated with an anonymity public and private key

[3] The most recent version of UDP calculates the checksum over the entire packet (S. Pink), private communication, June 1997.

($K_{pub}$(AS), $K_{priv}$(AS)). If a sender desires to send an anonymous message, he encrypts it with the anonymity public key, indicates the final destination in an appropriate field, and then sends the message to the anonymity server. The anonymity server decrypts the message by means of the anonymity private key and forwards it to the real destination after having encrypted the payload with the destination's public key. When the message is received at the destination, the address of the sender contained in the packet header is that of the anonymity server, so that the destination has no information on who the real sender of the message is.

## 5. Conclusions

In previous works, we introduced the notion of supranets, i.e., Internet-based secure virtual networks, and described a number of potential application areas. This paper focused on the main design requirements to be considered when building a supranet. After the discussion of such typical networking issues as addressing, routing, and multicasting, we focused on security and showed how a set of mechanisms based on asymmetric keys cryptography can be used to address in an elegant and efficient way all the security issues that have been considered.

### Bibliography


[1] D. Ferrari and L. Delgrossi: "*Supranets*", CRATOS Technical Report CTR-T96-001, September 1996, submitted to the IEEE Internet Computing Journal.

[2] B. Rossi, L. Delgrossi, D. Ferrari: "*The Applications of a Toolkit for Virtual Network Creation and Management*", 4th IEEE Workshop on the Architecture and Implementation of High-Performance Communication Systems (HPCS'97), Chalkidiki, Greece, June 1997.

[3] L. Delgrossi and D. Ferrari: "*A Virtual Network Service for Integrated-Services Internetworks*", Proc. 7th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'97), St. Louis, Missouri, USA, May 1997.

[4] A. Bhimani: "*Securing the Commercial Internet*", Communications of the ACM, Vol. 39, No. 6, June 1996.

[5] D. Ferrari: "*Should an Integrated Services Internetwork be Connectionless or Connection-Oriented ?*", 6th International NOSSDAV Workshop, pp: 3-4, Zushi, Japan, April 1996.

[6] R. Atkinson: "IP Authentication *Header*", Internet RFC 1826, Proposed Standard, August 1995.

[7] R. Atkinson: "*IP Encapsulating Security Payload*", Internet RFC 1827, Proposed Standard, August 1995.

[8] S. Fotedar, M. Gerla, P. Crocetti, L. Fratta*: "ATM Virtual Private Networks*", Communications of the ACM, Vol. 38, N. 2, pp: 102-109, February 1995.

[9] J. M. Schneider, T. Preuss, P. S. Nielsen: "*Management of Virtual Private Networks for Integrated Broadband Communication*", Proceedings of the ACM SIGCOMM'93 Conference, pp: 224-237, September 1993.