N2N: A Layer Two Peer-to-Peer VPN

Luca Deri[1], Richard Andrews[2]

ntop.org, Pisa, Italy[1]
Symstream Technologies, Melbourne, Australia[2]
{deri, andrews}@ntop.org

Abstract. The Internet was originally designed as a flat data network delivering a multitude of protocols and services between equal peers. Currently, after an explosive growth fostered by enormous and heterogeneous economic interests, it has become a constrained network severely enforcing client-server communication where addressing plans, packet routing, security policies and users' reachability are almost entirely managed and limited by access providers. From the user's perspective, the Internet is not an open transport system, but rather a telephony-like communication medium for content consumption.

This paper describes the design and implementation of a new type of peer-to-peer virtual private network that can allow users to overcome some of these limitations. N2N users can create and manage their own secure and geographically distributed overlay network without the need for central administration, typical of most virtual private network systems.

Keywords: Virtual private network, peer-to-peer, network overlay.

1. Motivation and Scope of Work

Irony pervades many pages of history, and computing history is no exception. Once personal computing had won the market battle against mainframe-based computing, the commercial evolution of the Internet in the nineties stepped the computing world back to a substantially rigid client-server scheme. While it is true that the today's Internet serves as a good transport system for supplying a plethora of data interchange services, virtually all of them are delivered by a client-server model, whether they are centralised or distributed, pay-per-use or virtually free [1].
Direct, free and private interoperation between domestic Internet users - for example a direct file transfer or a point-to-point server-less message exchange - is generally prevented by the market-driven characteristics of typical corporate and domestic Internet access systems. These tend to mask the user's IP identity and limit peer accessibility:

- In many cases, end users do not have any control over their connection configuration, which is totally managed by Internet service providers (ISPs).

- ADSL, fibre or cable Internet users are generally hidden through chains of NAT [35] devices that may even start at the customer's premises with the ubiquitous home access gateway. In some cases, a domestic user connection has a private, i.e. non-public, IP address, which prevents an external peer initiating a direct IP session. This kind of access is generally acceptable to a domestic content consumer, given the client-server nature of most Internet services. In this scenario, a NAT-ed access appears indistinguishable in performance with respect to a "premium service" offering permanent, public IP assignment.
- Firewalls outside the user's control, greatly reduce the possibility of a user being contacted by a direct session initiated elsewhere. Furthermore these devices limit the protocols that can be used in a direct information transfer between geographically distributed peers.

All of these facts lead to the conclusion that for the vast majority of users, the Internet is a severely constrained IP transport system that hinders visibility and data exchange. Even when users are visible, dynamic addressing techniques and roaming issues prevent them from being consistently addressed by a unique name and having equal capabilities regardless of the access connection that they are using. The consequence is that the increasing success of social networks and communities is restricted to selected service models. Users can for instance exchange files only through a specialised application (e.g. Microsoft Messenger) instead of pushing files onto a user's public folders.
User experience is restricted to a consumer-provider model. Even user-to-user communications are defined within the bounds of users connecting to a common server. Partial solutions for the set of problems outlined above are already offered by many peer-to-peer (P2P) applications, but they approach the problem at the application level, rather than at network level. In general, these solutions rely upon distributed hash tables (DHTs) [2] for setting up a so-called network overlay among peers. This overlay network is in turn used as a communication means for specific, overlay-network-aware services typically file sharing and instant messaging.

P2P has become a truly disruptive approach that has changed the way the Internet is used: it has allowed users to create an application-based closed network in which data can be exchanged even with the limitations of the "closed" Internet such as firewalls, dynamic IPs and NAT [36]. Usually P2P is limited to a specific service (e.g. file sharing) rather than attacking the problem of generic IP communications through firewall restrictions. True IP users target an IP address and a service (e.g. http://www.google.com) for datagram exchange, while P2P users target an application token (e.g. song xyz or skype user abc). The fact that P2P applications are able to cross most firewalls is not perceived by their end-users as a security hole but rather as a desirable property. In fact people do not care about IP configuration, but they do care about permanent service availability regardless of the connection type (cable, wifi, phone)

and physical location (e.g. at home, on the street or at work).

Unfortunately P2P has been used predominantly at the application level, and the above-listed beneficial properties of P2P protocols are limited to solving particular application-level problems. Based on these considerations, the authors decided to exploit P2P principles in order to interconnect network resources that otherwise would not be reachable due to network configuration and security restrictions [3].

N2N (network to network) is a novel layer-two over layer-three P2P virtual private network (VPN) application developed by the authors, which allows users to exploit properties typical of P2P applications at the network level instead of application level. This means that users can gain unrestricted IP visibility and be reachable with the same address regardless of their current network environment. In a nutshell, as OpenVPN moved SSL [11] from application (e.g. used to implement the https protocol) to network protocol, N2N moves P2P from application to network level.

2. The Design of N2N

A virtual private network [4] is a secure logical network that is tunnelled though another network. VPNs are often used for implementing secure point-to-point communications through the public Internet. Therefore they usually feature user authentication and content encryption. Network administrators use VPNs for securely and permanently connecting remote sites through the Internet without the need for expensive leased lines. Mobile users and telecommuters use VPNs to connect to their private office. The key elements of VPNs are (a) encryption (which protects sensitive data while travelling on the public Internet) and (b) encapsulation (which allows transport between cooperating tunnel end-points).

Permanent VPNs are often implemented at layer two with protocols such as 802.1q [5], and at layer three with BGP/MPLS [6] and are often static in terms of topology and peers. Semi-permanent VPNs such as those used by mobile users can be based on standard protocols such as PPTP [7] and L2TP over IPSEC [8] or de-facto products such as Cisco VPN and OpenVPN [9]. VPNs are implemented either using complex (to implement, administer and use) protocols such as IPSEC [10] or using SSL/TLS [11] that were originally designed to securely interconnect applications such as web servers with browsers.
Regardless of the VPN type, the key concept is that network administrators configure the VPN and users must use the setups specified by administrators in order to use it. VPN servers must be accessible by means of a public IP address such that the client can reach the VPN server. VPNs therefore form a star-topology with the service located at the publically reachable nexus. Server reachability can be an issue as many

VPN systems use non-TCP/UDP protocol, or use privileged low (< 1024) ports (e.g. ISAKMP used by IPSEC uses port 500) that are often blocked by firewalls. This means that VPN clients can be unusable from many places like public hotspots, hotels and many GPRS connections. User mobility and remote access often do not work with a VPN model.

The above limitations of the current VPN systems have been the driving force for the authors for the design of N2N. In a nutshell we ask: "Is it possible to have decentralised, network-administrator-free, secure and permanent network access with a single/uniform address regardless of the current user's location, local IP address and network type?".
The authors designed N2N to give N2N users the ability to create dynamic private networks. As happens with community networks, users should be able to create their own overlay network which other users are invited to join. With VPN the network administrator chooses who may join the VPN and what interactions will be tolerated. N2N is somewhat similar to Hamachi [12], a popular application mostly used for creating private networks on which to play games. With N2N users can choose their IP address and the encryption keys, whereas with Hamachi this is not possible and all the security is delegated to Hamachi, making the whole solution weak from a security point of view.

The main design features of N2N are:

- N2N is an encrypted layer two private network using a P2P protocol. Each N2N node has a name and a common encryption key pre-shared among the users that have been invited to join the network (community).
- Encryption is performed at edge nodes using open ciphers with user-defined encryption keys. This differs from popular applications like Skype and Hamachi where the traffic is encrypted by the application with no control by the application user. Skype developers can decode Skype traffic [13], which gives users a false sense of security. This does not happen with N2N where only users holding the private keys can decrypt the traffic.
- Each N2N user can simultaneously belong to multiple communities. Users will have an encryption key, MAC and IP address for each N2N community.
- Like most P2P protocols, N2N has one or more supernodes and several edge nodes. Supernodes are used to introduce edge nodes and to cross symmetric NAT. N2N packets are encrypted/decrypted only by edge nodes and supernodes forward packets based on a clear-text packet header without inspecting the packet payload. This is a core differentiator of N2N.
- N2N can cross NAT and firewalls in the reverse traffic direction (i.e. from outside to inside) so N2N nodes become directly reachable from the community even if running on a private network.
- N2N communities are meant to be self-contained, but it is possible to route

traffic across N2N communities. Packet forwarding through N2N is disabled by default, as this can be a security flaw. N2N users can enable it if necessary but doing so requires explicit user awareness.

The need to cross NAT and firewall devices motivated the use of P2P principles for interconnecting N2N nodes. During the design phase, the authors analysed several popular P2P protocols [14] ranging from proprietary (e.g. Skype SDK) to open (e.g. BitTorrent [15]) protocols. Unfortunately most protocols have been created for file sharing and are not suitable for N2N because PDUs (Protocol Data Units) have been designed to carry file information (e.g. name, length, type, attributes such as MP3 tags) and perform distributed file searches. Even though existing P2P protocols were not immediately usable for N2N without modification, some concepts already present in other P2P architectures [16] have been utilised as is explained in the following chapter. In addition to the properties listed so far, N2N presents further differences from other approaches [28] [29] [30]:

- Unlike most P2P overlay networks such as Chord [25] and Pastry [26] that are affected by the problem or locating objects/peers in a limited number of overlay hops, in N2N this is not a problem as, by design, peers are reachable either directly or in one hop when passing through the N2N community. This design choice has dramatically simplified peer lookup and membership information without requiring complex algorithms for information bookkeeping [27].
- N2N node membership is rather static. Nodes usually register with a N2N community and stick with it as long as the node is operational. In other networks such as Gnutella or Napster the membership change rate is much higher and can lead to issues as the network topology might need to be changed in order to handle new members.
- N2N nodes themselves do not store, cache, replicate or manage any content. This is because N2N connects network peers rather than content and consumers.
- The goal of N2N is not to share files but rather to allow peers to communicate generically in a secure way and to locate each other by consistent addressing regardless of their physical network location. Data sharing is accomplished by higher layer protocols as the Internet design intended.

3. N2N Architecture and Implementation

Edge nodes run on a host that can be placed in a private or public LAN. Supernodes are used to introduce edge nodes and relay packets to an edge behind symmetrical NAT (which prevents it from being directly reachable by peers). With symmetric NAT all requests from the same internal IP:port to a specific IP destination:port are mapped to an external IP:port. If the same internal host sends a request with the same source address and port to a different destination, a different external mapping is used.

This prevents the arrival of packets from any other remote socket.

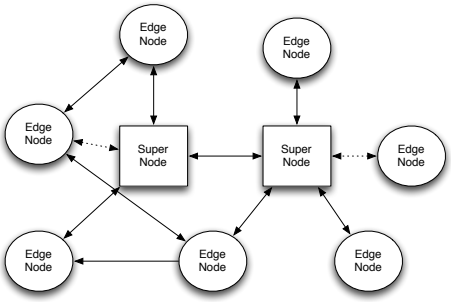

Fig. 1. N2N architecture

Edge nodes have a list of supernodes to which they register at startup. Supernodes temporarily store information about the edge nodes; such information needs to be periodically refreshed by the edge node. Edge nodes register to the first available supernode; registration to additional supernodes happens if the supernode to which the edge node is registered becomes unresponsive.

As N2N is a layer two VPN, edge nodes are identified uniquely by a 6 byte MAC address and a 16 byte community name. Edge nodes use TAP [17] devices that act as virtual ethernet devices implemented in the operating system kernel. In TAP devices, the ethernet driver is implemented in a user-space application. N2N provides such a driver implementation, which encapsulates encrypted ethernet frames within UDP packets as in Figure 2.

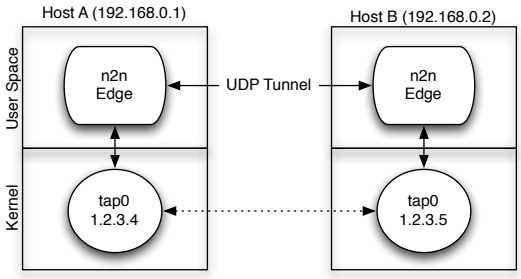

Fig. 2. N2N node communications

The kernel sees the TAP device as the path through which to send ethernet frames (based on LAN routing). UDP capsules arriving on the IP side are decrypted and the ethernet frames injected into the kernel as if they had arrived from an ethernet adapter. The use of TAP devices reduces the design of the architecture to familiar LAN concepts. The use of UDP encapsulation simplifies firewall and NAT traversal

compared to other encapsulations such as GRE [18] that are often blocked by firewalls.

When an edge node is started, it registers with the first configured supernode and sends an acknowledged registration packet. If the acknowledgement is not received, the edge node retries then move on to another supernode. Registrations are sent periodically in order to refresh the edge registration and make sure that any firewalls between the edge and the supernode do not block the connection due to inactivity. This means the supernode can relay packets to the edge node as long as the firewall stays open. This is the technique used by N2N to traverse NAT in the reverse direction. Each supernode keeps a list of paths to each edge node keyed by {community, MAC}. N2N provides layer-2 broadcast via the supernodes, which act to forward broadcast and multicast packets to all edge nodes in the community. As edge nodes receive remote packets they also build a list of {MAC, UDP socket} for peers in the community, and send registration requests directly to the peers. Figure 3 shows how this can lead to a NAT traversal in the reverse traffic direction.

Peer registration provides a mechanism for edges to form direct connections thereby removing the supernode from the path. If the sender edge node receives an acknowledgement for a register message previously sent directly to a remote node, then the nodes can reach each other directly. If one of the peers is behind symmetric NAT, the act of sending a registration request directly to the other peer opens a return path through the firewall. If both peers are behind symmetric NAT, direct connectivity is not possible. As happens with ARP [21], dynamic peer registrations expire if not renewed. Note that:

- The N2N community name is conceptually similar to the 802.1q VLAN ID.
- Dynamic peer registration may fail, e.g. due to firewalling. In this case packets can use asymmetric routing, e.g. A to B via S but direct from B to A.

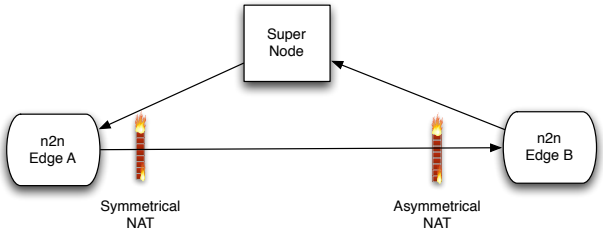

Fig. 3. N2N communications through NAT

N2N uses Twofish [19] as its encryption algorithm. The authors chose this symmetric key block cipher as it is fast, unpatented and its source is uncopyrighted and licence-free. Each N2N community has a shared key that is used to encrypt/decrypt N2N

packet payloads. If a supernode is compromised, injected traffic will be discarded as supernodes do not ever know community keys. layer-2 frames are also compressed using the Lempel-Ziv-Oberhumer (LZO) [20] algorithm that, like Twofish, is fast, efficient and available under the GNU GPL license. The N2N packet header is not compressed (nor encrypted) which allows supernodes to forward packets.
MAC address duplication within N2N can lead to the same problems that occur in ethernet networks so care must be taken to avoid conflicts. Edge nodes can have dynamic IP addresses by means of a DHCP server attached to an edge node in a community. As N2N TAP interfaces behave like real (e.g. ethernet) interfaces, it is also possible to run other services such as those defined by the IETF Zeroconf Working Group including, but not limited to, multicast DNS (mDNS) and DNS Service discovery [22]. Unlike most P2P systems, a node naming scheme [33] for locating peers is redundant in N2N.

4. N2N Evaluation and Testing

As N2N is a technology that is designed to interconnect heterogeneous computers, the authors ported it to three common platforms: Linux, MacOS X including BSD variants, and Windows. From the software point of view the code base is the same. Platform-specific code was needed for supporting the various TAP APIs. The authors acknowledge that the OpenVPN project has done a significant amount of work, so that today it is relatively simple to use TAP devices in a multivendor environment.
In the test setup, the supernode was installed on a Linux PC with a public IP address although this is not really necessary as long as the supernode UDP port is publically reachable. N2N nodes were placed behind several types of NAT/firewall devices, including symmetric NAT and multi-NAT (i.e. cascaded NAT devices) that are often used by ISPs. Edge nodes have been used on all above listed platforms in order to evaluate interoperability. As soon as the edge application starts, the node is part of the N2N community and can communicate with remote N2N peers. Several protocols were tested successfully, ranging from SSH to dynamic FTP.
As N2N uses TAP devices, it is possible to run popular tools such as tcpdump and wireshark on the TAP interfaces. Edge-to-supernode traffic is both encrypted and compressed, but traffic at the TAP interface is clear-text.
All existing IP-based applications can run on N2N without any change or recompilation. Even multicast works within N2N communities as long as the routing tables are correctly configured. Nevertheless the full power of N2N is unleashed when using community-based applications such as Retroshare [23] or I Hear You [24], the latter being a P2P VoIP application.

While N2N's throughput is similar to other VPN implementations such as OpenVPN and PPTP as they share the basic building blocks, its major advantages compared to

those technologies are:

- The ability to create a private network without a central control point.
- Direct packet exchange, which increases network efficiency and reduces latency.
- Applications see N2N as just another ethernet LAN.
- The code base is extremely small with no dependencies on external libraries (e.g. OpenSSL) or proprietary software, so it can be embedded into small devices and appliances.

N2N is a major step ahead compared to application layer P2P networks because it is transparent and usable by all applications without any N2N awareness. If N2N had been implemented at a higher layer, it would have been usable only by applications aware of it; as happens with most P2P applications.

N2N may appear as a way to circumvent many management mechanisms for security and privacy, and allow subcultures to share information free of monitoring. In this respect N2N might appear more as a problem rather than a solution to connectivity issues. The authors have not designed N2N to be a tool for defending users against network surveillance, but rather as utilitarian network overlays through which information can flow directly, securely and reliably.

4.1 Comparing N2N and Mobile IP

Mobile IP [33] is a scheme to allow global dynamic routing of IP packets to a static IP address when the host holding the static address is mobile. This is done by a series of routing tricks involving NAT and tunneling. It requires a central holder of the static IP address to be present to relay all packets destined for the mobile host. The presence of a packet relay node presents scalability and reliability problems. The types of problems to expect can be found extrapolating the lessons learned from GPRS, where the GGSN forms a bottleneck to traffic and a single-point of failure. N2N by contrast drastically alleviates these problems for participating communities of hosts. When a member of an N2N community changes its public IP address, all other members begin utilising the new address on ARP refresh and direct peer-to-peer communication is transparently restored without the need for a packet relay node such as the home agent in Mobile IP. Mobile IP is designed for use with telco-provided services where the care-of IP address assigned to a roaming device is routable and not firewalled. As a result Mobile IP is of little value when the mobile host changes its IP address to that of a host on a LAN which is isolated from the public internet by a firewall. N2N by contrast will pierce the firewall and transparently provide peer-to-peer communications once more. For the symmetric NAT case, N2N reduces to a similar situation as Mobile IP where the supernode must act as a packet relay. The peer-to-peer nature of N2N should deliver far better interactive performance due to the

drastically reduced round-trip time and the removal of a queueing point which exists at the home agent in Mobile IP. The shortening of the packet path reduces the average round-trip time as there are fewer hops to cover. The removal of the packet relay reduces the variance of round-trip time which - in Mobile IP - is influenced by momentary load associated with packet forwarding for other nodes.

Community in the N2N sense refers to a set of participating hosts which elect to be part of the community by maintaining registrations with the community supernodes. The uses of such communities are many. The pattern describes the internal communications of most small businesses, peer groups, affiliations, etc. N2N provides a LAN extension to a mobile community. No matter where a participating host roams to and what transport it uses, it remains a member of the N2N L2 network. Traditional road-warrior setups depend on client-server models with the mobile host being the client. Network security and resource access rely heavily on this model making it difficult to provide true peer-to-peer applications. Indeed the availability and uptake of true peer-to-peer applications may be hindered by the difficulty in providing such applications across network boundaries. N2N provides an enabling technology for true peer-to-peer IP communications models such as push-to-talk type conferencing, information synchronisation by push (rather than pull-from-server) or by broadcast.

Being a direct peer-to-peer technology N2N scales much better than solutions such as Mobile IP which rely on a single point of packet aggregation and forwarding [32]. The N2N model makes use of the supernodes only when required - typically at the time when a host must be contacted but its location is unknown. The supernode has minimal participation in packet delivery and as such is not a limiting factor on host-to-host round-trip time or throughput.

4.2 N2N Scalability

N2N has been designed to be simple and address connectivity limitations present in most networks as explained in the previous sections. By no means has N2N been designed to implement a large overlay networks where thousand of nodes can join for a few reasons:

- Large overlays create a significant traffic load on supernodes which can lead to performance degradation.
- In order to optimise the overlay and increase the number of supernodes, some edge nodes (e.g. those that are not behind a symmetrical NAT) should become a hybrid edge-supernode as happens with most P2P applications. Yet this would impact on node performance.
- Efficient supernode selection based on criteria like round trip time and available bandwidth is necessary for a large overlay. However this would significantly

add complexity to the N2N code and produce extra node traffic in order to periodically calculate supernode section metrics.

That said, the authors are aware that the next N2N release should address some issues, including better scalability. In any case, this planned evolution will not upset the core N2N principles, which were designed to be simple, reliable, and usable for business activities.

4.3 Network Management and N2N

Most management protocols have not been designed to run across firewalls and NAT. However in some cases, network administrators are forced to use them in this scenario and often the solution is to setup the firewall with static access rules. This however is not a clean solution as it is rather simple to forge packets, in particular UDP-based protocols such as SNMP. However TCP-based used to administer remote hosts protocols such as VNC [34] and RDC (Remote Desktop Connection), are not suggested to run across firewalls as a protocol flaw could compromise the whole network.

Network management applications can take advantage of N2N for creating secure WAN management networks on which protocols can flow without having to take into account security and network constraints. Using different community names, administrators can add extra security by creating several overlays one for each group of homogeneous management resources, so that management traffic is further partitioned. Implementing the same partitioning scheme using traditional techniques such as VLAN or VPNs would have been much more difficult and in some cases (e.g. on WANs) probably not feasible at all.

5. Open Issues and Future Work

Although fully operational, the N2N development is not over. The authors are:

- Implementing N2N tunneling over other protocols such as HTTP and DNS in order to give users the ability to run N2N even on those partially open networks where only a few protocols such as email and web are allowed.
- Designing security mechanisms to avoid intruders and bugged edge nodes which could disable a community with node registration bombing.
- Enhancing supernode selection and registration algorithms, so that edge nodes dynamically select the fastest reachable supernode among the list of available supernodes [31].
- Evaluating the porting of N2N onto small devices such as Apple iPhone and Linux-based network devices including DreamBox and Android phones when

available.

In a nutshell the plan is to allow N2N to run on:

- Most partially-open networks and give mobile users the ability to have secure access with a fixed N2N IP address, if configured, regardless of their location, e.g. at the airport lounge, in a hotel room or at the office.
- Hostile networks where attackers can try to break N2N community security.
- Embedded portable devices that can be used for letting users access their own private network from all applications in a secure way, without having to cope with the limitations of most networks that often allow only reduced Internet access.

6. Final Remarks

This paper has described a novel type of P2P VPN that enables the creation of secure, private networks regardless of the peer locations, network access type, or operating system. Unlike the current VPN generation, N2N is fully decentralised and uses supernodes only at start-up or whenever peers are behind symmetric NATs that prevent direct peer communication. Existing applications do not need to be changed in any way to exploit N2N. The ability to tunnel N2N traffic over protocols such as HTTP and DNS makes it a very useful technology for allowing users to overcome restrictions in many networks. N2N users can be partitioned into networks and have a permanent, unique N2N IP address regardless of the current address of the device running N2N. This makes N2N suitable for creating overlay networks of users sharing homogeneous information. Finally, traffic encryption at the edge and not by any controlling entity, enables users to securely exchange sensitive information with much less risk of being intercepted or observed by unwanted peers.

References

1. A. Fuggetta, The Net is Flat, Cefriel Technical Report, http://alfonsofuggetta.org/mambo/images/stories/Documents/Papers/TheNetIsFlat.pdf, (2007).
2. P. Maymounkov and D. Mazières, Kademilia: A Peer-to-peer Information System Based on XOR Metric, 1st Intl. Workshop on Peer-to-Peer Systems, (2002).
3. L. Deri, Empowering peer-to-peer services, EFNIW Workshop, (2007).
4. B. Gleeson et al., IP Based Virtual Private Networks, RFC 2764, (2000).
5. A. McPherson and B. Dykes, VLAN Aggregation for Efficient IP Address Allocation, RFC 3069, (2001).
6. E. Rosen and Y. Rekhter, BGP/MPLS IP Virtual Private Networks, RFC 4364, (2006).
7. K. Hamzeh et al., Point-to-Point Tunneling Protocol (PPTP), RFC 2637, (1999).
8. B. Patel et al., Securing L2TP using IPsec, RFC 3193, (2001).

9. C. Hosner, Open VPN and the SSL VPN Revolution, Sans Institute, (2004).
10. S. Kent et al., Security Architecture for the Internet Protocol, RFC 2401, (1998).
11. T. Dierks and C. Allen, The TLS Protocol, RFC 2246, (1999).
12. LogMe In, Hamachi Security – an Overview, White Paper, (2007).
13. Skype Ltd, Skype Public API, https://developer.skype.com/, (2008).
14. K. Khan and A. Wierzbicki (Ed), Foundation of Peer-to-Peer Computing, Special Issue, Elsevier Journal of Computer Communication, Volume 31, Issue 2, (2008).
15. BitTorrent Protocol Specification, http://www.bittorrent.org/protocol.html, (2008).
16. H. Balakrishnan et al., Looking up data in P2P systems, Communications of the ACM, (2003).
17. M. Krasnyansky, Universal TUN/TAP Driver, http://vtun.sourceforge.net/, (2001).
18. S. Hanks et al., Generic Routing Encapsulation (GRE), RFC 1701, (1994).
19. S. Schneider et al., Twofish: A 128-Bit Block Cipher, Couterpane Labs, (1998).
20. M. Oberhumer, LZO Compression Library, (2005).
21. D. Plummer, An Ethernet Address Resolution Protocol, RFC 826, (1982).
22. S. Cheshire and D. Steinberg, Zero Configuration Networking: The Definitive Guide, O'Reilly Media, (2005).
23. The Retroshare Team, Retroshare, http://retroshare.sourceforge.net/, (2007).
24. M. Trotta, I Hear You, http://ihu.sourceforge.net/, (2008).
25. I. Stoica at al., Chord: A Scalable Peer-To-Peer Lookup Service for Internet Applications, Proceedings of ACM SIGCOMM, (2001).
26. A. Rowstron and P. Druschel, Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-peer Systems, Proc. of IFIP/ACM Middleware, (2001).
27. A. Gupta et al., One Hop Lookups for Peer-to-Peer Overlays, Ninth Workshop on Hot Topics in Operating Systems, (2003).
28. C. Wang and B. Li, Peer-to-Peer Overlay Networks: A Survey, Technical Report, Dept. of Computer Science, HKUST, (2003).
29. E. Keong Lua et al., A Survey and Comparison of Peer-to-Peer Overlay Network Schemes, IEEE Communications Surveys, (2005).
30. M. Castro et al., Exploiting Network Proximity in Peer-to-peer Overlay Networks, Technical Report MSR-TR-2002-82, Microsoft Research, (2002).
31. M. Jovanovic et al., Scalability Issues in Large Peer-to-peer Networks - A Case Study of Gnutella, Technical Report, Univ. of Cincinnati, (2001).
32. X. Li and C. Plaxton, On Name Resolution in Peer to Peer Networks, Proceedings of 2nd Intl. Workshop on Principles of Mobile Computing, (2002).
33. C. Perkins, IP Mobility Support, RFC 2202, (1996).
34. T. Richardson, The RFC Protocol, http://www.realvnc.com/docs/rfbproto.pdf, (2007).
35. J.D. Touch, Those pesky NATs, IEEE Internet Computing, July/August 2002, p. 96.
36. Information Sciences Institute, TetherNet, http://www.isi.edu/tethernet/, (2002).

Acknowledgment and Code Availability

The authors would like to thank Simone Benvenuti, Carlo Rogialli and Maria Teresa Allegro for their help and suggestions during N2N development.

N2N is distributed under the GNU GPL 3 licence and is available at the ntop home page http://www.ntop.org/n2n/ and other mirrors on the Internet.