Norsk Regnesentral
NORWEGIAN COMPUTING CENTER

Note

Hikernet

Peer-to-Peer Messaging in an Ad-Hoc Network

Note no DART/10/04

Authors Wolfgang Leister

Date December 2004

About the authors

Wolfgang Leister is chief research scientist in the DART department. His research interests include distributed systems, multimedia, and computer graphics.

Norsk Regnesentral

Norsk Regnesentral (Norwegian Computing Center, NR) is a private, independent, non-profit foundation established in 1952. NR carries out contract research and development projects in the areas of information and communication technology and applied statistical modeling. The clients are a broad range of industrial, commercial and public service organizations in the national as well as the international market. Our scientific and technical capabilities are further developed in co-operation with The Research Council of Norway and key customers. The results of our projects may take the form of reports, software, prototypes, and short courses. A proof of the confidence and appreciation our clients have for us is given by the fact that most of our new contracts are signed with previous customers.

Norsk Regnesentral
Norwegian Computing Center
Postboks 114, Blindern
NO-0314 Oslo, Norway

Besøksadresse
Office address
Gaustadalléen 23
NO-0371 Oslo, Norway

Telefon · telephone
(+47) 22 85 25 00
Telefaks · telefax
(+47) 22 69 76 60

Internett · internet
www.nr.no
E-post · e-mail
nr@nr.no

Title	**Hikernet**
Authors	**Wolfgang Leister**
Date	December
Year	2004
Publication number	DART/10/04

Abstract

This document describes the proposal of the messaging service HikerNet enabling electronic communication in areas without ordinary networking access. The transport of the messages is based on small devices that are carried around, and which can exchange messages at close range based on peer-to-peer connections in an ad-hoc network. This document outlines how the HikerNet is designed, and discusses the properties and limitations of the service, as well as possible extensions.

Keywords	Ad-hoc networks, Peer-to-Peer
Target group	
Availability	public
Project number	
Research field	Multimedia-Multichannel
Number of pages	8

HikerNet

Wolfgang Leister*

Abstract
This document describes the proposal of the messaging service HikerNet enabling electronic communication in areas without ordinary networking access. The transport of the messages is based on small devices that are carried around, and which can exchange messages at close range based on peer-to-peer connections in an ad-hoc network. This document outlines how the HikerNet is designed, and discusses the properties and limitations of the service, as well as possible extensions.

1 Introduction

In densely populated areas our civilisation has achieved the ability to transfer messages in nearly real time to everybody within the reach of a network. However, recipients outside the area of the infrastructure cannot be reached easily.

To build up a networking infrastructure is expensive, and for many areas it might not be worth the investment, could be impossible, or the necessary funding are not available, e.g., sparsely populated areas, developing countries, on sea, deserts, etc.

Ad-hoc P2P networks can be used to build a network infrastructure in the cases above, or when the infrastructure has been destroyed, but can also be used to build an alternative networking infrastructure to the existing networks.

We observe that even sparsely populated areas are populated by humans, who are hiking within an area. The idea of this proposal is to use these hikers as bearers of electronic messages.

We illustrate this with the following example: When hiking in a mountain area, there is usually no coverage for GSM or other networks. In the lack of a networking infrastructure, the sender of a message could write a postcard and put it in the mail box of a cabin. Some other hiker would take the postcards to the next cabin, and so on, until the postcard eventually reaches a post office, from where it is transported to the recipient the ordinary way.

We use similar mechanisms in order to transport electronic messages. One distinction is that electronic messages can be replicated while being transported. Using these mechanisms, our goal is to create a network, that is affordable and that is able to transport messages within reasonable time, reliability and security.

2 Architecture

2.1 Basic operations

The proposed message forwarding service will be referred to as the **HikerNet**. It is based on (mobile) terminals attached to user transport nodes (H-nodes), which carry the user's incoming and outgoing messages. The HikerNet infrastructure also consists of inner transport nodes (N-nodes), which carry and exchange messages for all users. The structure of three distinct parts (terminal, H-node, N-node) is of a conceptual nature; all three parts can be physically integrated into one device. The N-nodes implement the transport system, which works independently and automatically, and are beyond the reach of the users.

The devices for the transport nodes (TN)[1] are supposed to be inexpensive, easy to carry, and accessible by ad-hoc communication using radio, or short-distance communication technologies (like IR, Bluetooth) or possibly be operated by inserting memory devices into a docking station. A TN device consists of memory, a processor, and a radio transmitter. The processor power and capabilities needed are rather modest, while the memory requirements are about some MBytes per TN. The TN devices are supposed to have enough memory to carry a reasonable number of messages between users.

*Norsk Regnesentral, Postboks 114 Blindern, NO-0314 Oslo, email: Wolfgang.Leister@nr.no

[1] TN can be H-nodes and/or N-nodes.

The TN can be integrated into communication devices like modern mobile phones without the need of additional hardware. The messages are propagated using ad-hoc communication and store-and-forward paradigms (peer-to-peer technique) between these TN devices, without the need of human attention.

The principles of message-propagation for the HikerNet are illustrated in Figure 1. The core of the HikerNet is built up as follows:

- The sender uses a mobile terminal to write a message in a MUA (mail user agent). Incoming messages can be read on this terminal. The senders and the receivers are connected to dedicated H-node, to which the message is transported immediately.
- A message is transferred with an ad-hoc connection to N-nodes, which happen to be within the reach of the H-node device.
- The messages are forwarded using a store-and forward method between N-nodes. The N-nodes automatically propagate (broadcast) messages when within reach of each other. After the transmission a copy of the message is kept on the sender N-node of this transmission, thus the message being replicated.
- Most of the physical displacement of messages arises from the TN devices being carried.
- H-nodes only receive messages addressed to this H-node. Since the user terminal accesses the H-node, a message has arrived its destiny when it has arrived at the H-node.
- The arrival of a message is acknowledged by sending an ACK-message to the HikerNet.[2] After an ACK-message arrives at a N-node, this node will stop with further distribution of the message, and keep the message ID for a while in order to avoid duplicates being forwarded later. ACK messages will be removed when the maximum time for forwarding has elapsed.
- The messages are propagated from a TN until an acknowledgement arrives, it is obvious that the message cannot arrive, or the message expires.
- Messages are identified by a message ID, using the MD5 sum of the message (not using the outer header, since its content can change during propagation). Additionally, the message ID is used for checking the consistency of a message.
- Messages are encrypted using public and private keys in order to provide reasonable privacy and security.

2.2 Extensions

Besides the core of the HikerNet we suggest the following extensions:

- We introduce *stationary N-nodes* into the HikerNet, which are not moving. These might be implemented with larger capacity, and serve as message hubs[3].
- We introduce *stationary relays* which represent N-nodes being present at several locations within an area. The manifestations of this N-node are interconnected by some infrastructure, e.g., the Internet, and implement the same state of this N-node at all its locations.
- We introduce *bridges*, which are stationary relays that connect several larger areas covered by HikerNet. However, messages are only sent to nodes that are in areas where the receiver has been in the vicinity lately, i.e. this part of the bridge has seen messages of this hiker. In order to trigger the bridges, keep-alive messages can be sent, which register a node at a bridge.
- We introduce *gateway*s from and to other messaging services, e.g. Internet mail. The gateways are H-nodes with well-known addresses that forward messages from other services to the HikerNet and vice versa. Address translation must be provided, e.g., using extra header information in the messages.[4] Note, that these gateways deliver from a global service, while the HikerNet is designed for smaller regions. Therefore, the same principle as in bridges applies. Note also, that filters must be employed in order to avoid spam.
- Additional use of broadcasting networks (radio, satellite, etc) would increase the connectivity of the HikerNet users, since radio waves can be sent over long distances. Broadcast messages are sent over a broadcast carousel[5] to all recipients. The broadcast carousel acts like an N-node with one-way properties. The HikerNet messages are encoded with relatively low bandwidth and multiplexed with other transmitted content. Broadcast messages can be received by both H-nodes and N-nodes. The

[2] These ACK-messages do not necessarily take the inverse route of the original message.

[3] The placement of stationary N-nodes might be at cabins.

[4] Using this the gateway functionality can be subject to billing.

[5] Carousel technology is, e.g., used for broadcast within Digital Television by DVB.

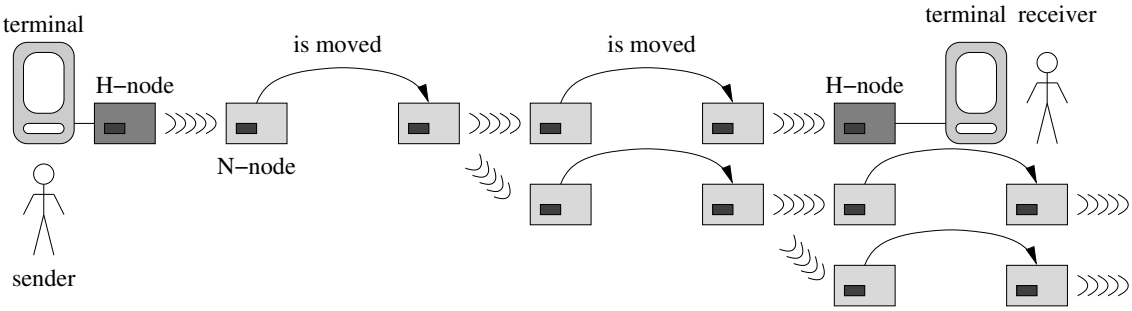

Figure 1: Illustration of the principles of message-propagation for the HikerNet.

ACK messages for broadcast messages are sent ordinarily by the HikerNet, in order to remove a message from the carousel.

- We introduce *publicly available terminals* for sending and receiving. The users could use a H-node implemented as a card or memory device. The user can access the messages stored in the H-node while using the terminal. A publicly available terminal should have an N-node integrated.
- N-nodes can be attached to other moving beings and objects, such as vehicles, animals near people that are moving within the area (sheep, elephants, cattle, dogs, birds[6] ...), without harming or disturbing them. The TN devices can be integrated with other objects, like watches, jewellery, etc. The integration into communication devices like modern mobile phones is useful.

3 Applications

The HikerNet is designed for communication outside areas with ordinary communication infrastructure. However, it might be also used as alternative transport network, e.g., for avoiding costs to providers of mobile phone networks. Note that the traffic pattern of the bearers of TN in cities might not be suitable for message propagation of the HikerNet, and cause overload situations.

The HikerNet will be suitable for messages that are not urgent, and that do not need a guarantee for coming through. A suitable message size would be from about some 10 Kbytes to the same size of todays MMS messages (up to 100 KBytes).

Besides the application of providing communication between people the HikerNet can be extended to operate as transport infrastructure for a sensor network. Some H-nodes could be integrated with sensors, e.g., a GPS receiver, attached, and send data to a receiver using the HikerNet. Possible applications could include biological field studies (see the ZebraNet [2]) or tracking of people on expeditions.

[6] See also the RFC-1149 [1].

3.1 Incentives for participating

The participants in the HikerNet have an interest, to get own messages transported. Therefore the user transports others messages also in order to get the system to work. The user can "buy" participation in the system by providing transport to others. The use of a reputation system can be considered.

The TN can be integrated into advanced mobile phones which provide both user interface and processing capabilities.

When a service for the HikerNet is introduced, and is used by some communities, the introduction of super-peers could give better connectivity, i.e., better access to messages. These super-peers can be installed at places where a business (e.g., restaurant, gas station) wants to attract customers.

3.2 Billing Models

Despite the HikerNet being intended as a system that can be used free of charge, billing models for message transport can be implemented, e.g., up-front payment, billing for messages at gateways, etc. The PKI solution (though with a rather low security level) and rather low amounts of money make a billing solution possible. In the most likely scenario messages between users are for free, while gateway services are billed for.

4 Message Propagation

The TN are transported passively by their bearers, and have no means of knowing of, or having an influence on the decisions of their bearers. In order to keep the TN devices cheap, the TN do not have knowledge of their positions, e.g., by the means of a GPS receiver[7].

The HikerNet propagates messages toward the recipient, providing a reasonable level of reliability. Messages should arrive within reasonable time. Additionally means have to be taken to avoid overload of the devices, and unnecessary transfers. In the absence of positioning information the use of a controlled flooding-algorithm for message propagation is suitable.

4.1 Propagation Parameters

In the TN the HikerNet uses several parameters that control the propagation of messages:

- TTL (times to live): value is decreased each time a message is transferred. When the value reaches 0 the message is only transferred to H-nodes.
- TTR (times to replicate): After each transfer this value is set to a maximum value at the receiver. Each time the message is transferred, the value is decreased at the sender. When the value is 0, the message is only transferred to H-nodes.
- EXP (expiry date): This value is a timestamp after which the messages are no longer propagated. The messages are eventually removed from the N-nodes, when the maximum lifetime denoted by EXP has expired. Also ACK messages with the same ID can be removed some while[8] after the message has expired.

4.2 Moments of a protocol

The H-nodes and the N-nodes implement different roles in the HikerNet. Knowing the role and identity of the communication partners the protocol used can be optimised. Therefore the H-node addresses must be recognisable as such, e.g., by using a naming convention.[9] The name of an N-node is not relevant for the functionality of the HikerNet, except for debugging purposes.

The HikerNet is based on ad-hoc point-to-point connections (and broadcast messages). Therefore, the protocol should have the following characteristics:

- a connection can be interrupted at any time. This should not cause more damage than the interrupted transfer not taking place.
- connections should be short (in terms of length and time).
- There should be no unnecessary handshake.
- As a consequence the protocol should be as stateless as possible.
- All TN in the vicinity should be able to listen to a connection, and use the perceived information, instead of making connections of their own.
- The operation of the HikerNet should support fragmenting messages (at application level)[10].

The HikerNet protocol is based on separate messages containing protocol id, version, data length, an opcode, and the payload. Messages are transferred using the header, and the message as outlined in Figure 2. For the protocol to work, no particular message sequence is required; however, announcements instead of sending the messages directly can be used for optimisation purposes.

The most simple case is to send a message without prior negotiations between the partners (Opcode 1). Then the receiver decides whether to keep or discard the message.

Negotiation between partners is not required, but would optimise the protocol as follows:

- A sender announces a message with `IHAVE` *msg-id* `[type=`*type*`] [size=`*size*`] [to=`*to-address*`]`.
- When a receiver wants to receive the message it sends `GIVEME` *message-ID*.
- When a node receives a `GIVEME`-message, it sends `MSG` *message-ID*, continued by the message itself, including headers.[11].

All involved nodes must get the chance to exchange messages as equal partners, since the HikerNet is based on a P2P paradigm. Since the communication is based on broadcasting / multicasting, and connection duration is assumed to be rather short, there is no need for a protocol which uses handover betwen TN.

[7] At a later stage devices with this knowledge could be used. However, the operation of the HikerNet does not depend on this.

[8] The ACK message needs some time to arrive at the sender of the original message.

[9] In the implementation we use prefixes: H for the H-nodes, and N for the N-nodes.

[10] not implemented in the current prototype.

[11] The sender of the message is not necessarily the same that announced the message with the `IHAVE`-message

5 Addressing and Security

Each TN in the HikerNet can be identified by a unique name. For the H-nodes this unique name is used to address messages. Additionally, each H-node has a public and private key pair attached. which is used for identification and encryption. A central data base over H-nodes and their public keys is available in order to check, whether addresses are authentic.

Note that addresses for sender or receiver can be available several times in the HikerNet, without endangering the function of the system. In this case messages can arrive at one or more of the receivers with equal address nondeterministically.

5.1 Security Issues

The key aspects for information security are to preserve the confidentiality, integrity and availability of the information in an organisation or system [3]. For the HikerNet the following security issues must be regarded:

- Traceability/Authenticity: Is the sender really the one he claims to be.
- Anti-Spam: Identification of the sender is necessary in order to prepare measures to avoid spam.
- Privacy: Messages should be encrypted in order to ensure privacy, since messages pass through many untrusted hosts.
- Data of the operation of the HikerNet could reveal information on the user, e.g., the position of a hiker, and who he communicates with. This leak-information must be kept at a minimum.[12]
- It is a design goal of the HikerNet that a node at maximum only can be as malicious as if this node would not exist at all.
- Authentication is important, since billing could be possible at e.g., gateways to other services.
- Messages can only be read when the correct keys are available. Therefore, messages cannot be unpacked in the N-nodes, while H-nodes can only unpack messages addressed to them.

[12] Most information can be encrypted in the inner envelopes. The To-address is currently not encrypted; however it is possible to encrypt this To-address, too. Since a flooding-algorithm is used the address can be encrypted using the private key of the receiver, and only the receiver can decide whether he is the correct recipient. See also the security issues of the Freenet [4, 5].

5.2 Security Operation of HikerNet

A public key system using private and public keys, and certificates is suitable for implementing a security infrastructure for the HikerNet. The following ideas support the security infrastructure:

- HikerNet IDs for sender and receiver have a public and private key pair attached, which can be used for encryption and authentication. This makes a PKI (public key infrastructure) necessary in order to build a chain of trust.
- A message, is encrypted using a session key, which is sent encrypted with the sender's private key, and the receiver's public key. To-address, message id, and parameters of the forwarding mechanism are sent openly.
- The HikerNet must establish reasonable level of trust; therefore the key length does not need to be very large.
- ACK messages contain the message id encrypted with the receivers private key. Using the public key of the receiver of the original message the ID can be checked to be authentic.
- ACK messages are checked for authenticity by using the public key of the receiver of the original message, which is available either in the original message, the stored keys in the node, or by requesting from the key server.

5.3 PKI

Since the HikerNet is based on an ad-hoc network, and since the transport-time is an issue, the HikerNet can only provide a reasonable level of security, which is below the security requirements of other systems. The challenge consists of the distribution of the public keys, which can be retrieved from a trusted host, with a known node name, and which public key is given (e.g., at install time). All nodes keep a list of public keys for nodes known to them.[13]

Public keys must be registered at the key server, after a node is generated. Note, that the key server can reject the node/key pair, when the node already exists.

When a hiker sends a message, the H-node checks whether the public key of the recipient is available. If so, the message is encoded and sent. If the public key of the recipient is not available, the message is held back[14], and a request for the key is sent to the

[13] Since public keys possibly can be renewed, it must be possible to store multiple keys for one node for a while.

[14] Alternatively, the message could be sent unencrypted.

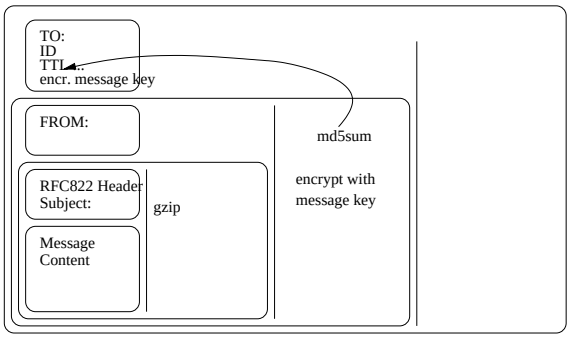

Figure 2: Schematic view of a HikerNet message.

key server. When the key arrives the key is stored in the H-node and used to encrypt the message.

The keys can be sent openly[15], but with a certificate; alternatively the key can be sent encrypted with the server's private key, and the user's public key.

When the key-pair changes, the key server is notified, using the the previous key for encryption and authentication.

6 Implementation

In order to study the HikerNet we implemented some aspects of the HikerNet in the program `hnagent`, which controls the contents and operations of a given node. The application is implemented in C as a command-line tool. While the actions are given in the parameter list, the data in or out is through stdin or stdout, to be used in a pipe. This implements only one-way communication, from one node to another, sending messages without prior negotiation. The use of `hnagent` is illustrated in Figure 3.

The implementation of `hnagent` is done in C for Linux, using the zlib and openssl libraries.

The `hnagent` program can be used to implement the HikerNet by using pen-drives (USB mass storage devices) together with docking stations: Some TN can be stored on pen drives, while other TN are stored on stationary PCs. After inserting a pen-drive containing a HikerNet node, and mounting it, messages can be forwarded between a pen drive TN and the local TN using a pipe `hnagent -i ...| hnagent -o ...`. Using the USB hotplug facility this can be automated, in order to make it easy for users just to insert the pen-drive and wait until the communication is finished. Gateways to Internet mail can be included in the docking stations.

For using wireless protocols we introduce adaptors that communicate with `hnagent` by Unix-pipes using the protocol described above. The adaptors send and/or receive messages using e.g., Bluetooth. For a proof of concept we implemented the application `udpadapter`, which uses the UDP protocol of the Internet.

7 Simulation of HikerNet

HikerNet has been simulated using the net of cabins of the Hardangervidda, and the implementation with `hnagent`. The detailed results can be found in [6].

Simulations done for a theoretical situation at the Hardangervidda in Norway revealed that the HikerNet works when more than 125 nodes are available, and a TTL value of 10 is used. In the simulation 500 messages were sent out. According to our simulations for more than 250 nodes after four days 45% of the messages have arrived, after seven days 80%, and after ten days nearly all messages have reached their destination.

We also got indications that the HikerNet works better when stationary nodes are introduced. This effect could be used to introduce super-peers at the cabins.

8 Related systems

An application for building infrastructure in rural areas based on similar thoughts has been described recently [7]. Sensor networks, e.g., the ZebraNet [2], are based on similar paradigms. The idea of using the postal system as a generic digital communication mechanism is described in a recent publication

[15] This would however, possibly reveal whom the sender is talking to. However, this helps the other nodes to know the public key without the need to request it again, if possible.

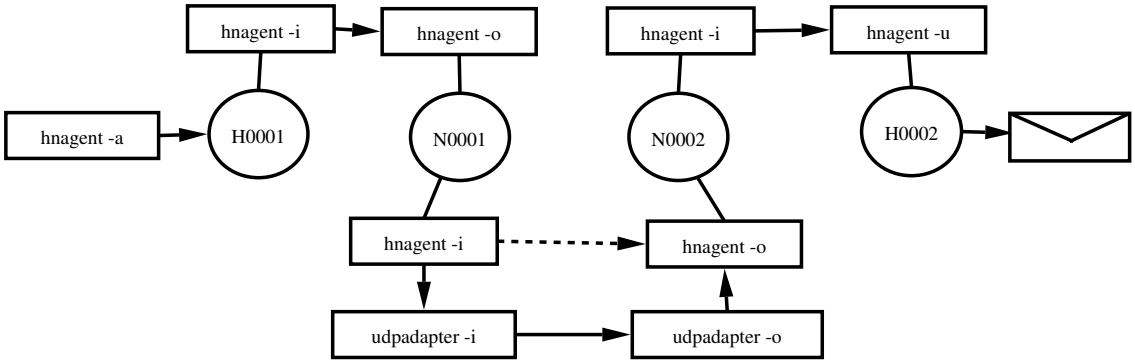

Figure 3: Illustration of the use of hnagent.

[8].

The HikerNet is designed as a peer-to-peer (P2P) system (see e.g., [9, 10]) applied to message exchange. The major property of a P2P system suggests that mostly partners with equal properties are involved in solving a problem. For message passing this paradigm suggests the use of many smaller devices rather than a client-server system. The client-server paradigm is used in today's Internet email, even though former implementations of email services, e.g., UUCP email, Fidonet, the USENET, were indeed implemented as P2P systems.

The **Freenet** [4, 5, 11] is used for content distribution (vs. message distribution in the HikerNet), with an emphasis of protecting the free speech and privacy of author and recipient. In the Freenet data are replicated and stored on several nodes. However, since only a part of the available data are stored on one node, data can disappear when requested less often, which can make less often used documents disappear from Freenet.

The so called **Eternity Services** [12] are an application of P2P to storage. These store data for long time periods. Because of their distributed nature, data replication and dispersion the storage method is suited for long terms, and beyond the access of someone wanting to remove the data.

Gnutella (see e.g., Chapter 8 of [9]) is a peer-to-peer service for file-sharing. The Gnutella-protocol has a search method which is based on viral propagation of the file query, while the content distribution as such follows a client-server paradigm using the HTTP-protocol. In order to prevent overload situations the Gnutella protocol uses a TTL parameter and a message ID in order to avoid that the same message is propagated several times. Interesting thoughts about scalability in the Gnutella Network are outlined in [13].

9 Conclusions and Future Work

Parts of this proposal for the HikerNet have been implemented in a prototype, and some aspects have been simulated. Other situations that cover situations beyond the cases covered in [6] must be performed, e.g., situations in densely populated areas like cities.

The current implementation lacks the security infrastructure, which must undergo a proof of concept. The implementation in real devices, e.g., mobile phones will be necessary prior to a field test.

On the theoretical side optimisations to the protocol must be designed, Especially issues like power saving must be looked at, since the battery capacity of devices is limited.

References

[1] D. Waitzman. A Standard for the Transmission of IP Datagrams on Avian Carriers. *IETF*, RFC 1149, 1990.

[2] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet, 2002.

[3] NS-ISO/IED 17799. *Informasjonsteknologi: Administrasjon av informasjonssikkerhet (ISO/IEC 17799:2000)*. Norsk Standard, 2000.

[4] I. Clarke. *A Distributed Decentralised Information Storage and Retrieval System*. Master Thesis, Division of Informatics, University of Edinburgh, 1999.

[5] I. Clarke, O. Sandberg, B. Wiley, and T. Hong. Freenet: A Distributed Anonymous Information Storage and Retrieval System. In H. Federrath, editor, *Designing Privacy Enhancing Technologies, Proc. International Workshop on Design Issues in Anonymity and Unobservability*. Springer, 2001.

[6] W. Leister and E. Garberg. The simulation of the hikernet. In *Proc. Simulation und Visualisierung 2005*. Society for Computer Simulation International, 2005. to appear.

[7] A. Pentland, R. Fletcher, and A. Hasson. DakNet: Rethinking Connectivity in Developing Nations. *IEEE Computer*, January, 2004.

[8] R. Wang, S. Sobti, N. Garg, E. Ziskind, J. Lai, and A. Krishnamurthy. Turning the postal system into a generic digital communication mechanism. In *Proc. SIGCOMM'04*. ACM, 2004. `http://an.kaist.ac.kr/courses/cs644/papers/wang.pdf`.

[9] Andy Oram, editor. *Peer-To-Peer: Harnessing the Power of Disruptive Technologies*. O'Reilly, 2001.

[10] M. O'Reilly. O'Reillys Peer-to-Peer Summit. Web pages, `http://www.oreillynet.com/pub/a/linux/2000/09/22/p2summit.html`, 2000.

[11] I Clarke, S. Miller, T. Hong, O. Sandberg, and B. Wiley. Protecting Free Expression Online with Freenet. *IEEE Internet Computing*, pages 40–49, January/February 2002.

[12] J. Anderson. The Eternity System. Web pages, `http://www.cl.cam.ac.uk/users/rja14/eternity/eternity.html`, 2000.

[13] J. Ritter. Why Gnutella Can't Scale. No, Really. Web pages, `http://www.darkridge.com/~jpr5/doc/gnutella.html`, 2001.