# The HikerNet

## Principle, Applications and Simulation

Wolfgang Leister, Norsk Regnesentral

NUUG Møte

Oslo, 18. august 2005

Hardangervidda, Sep 2001, Arnt Flatmo
Hårteigen, 1690m, seen from Torehytten

# When telecommunication is out of reach ...

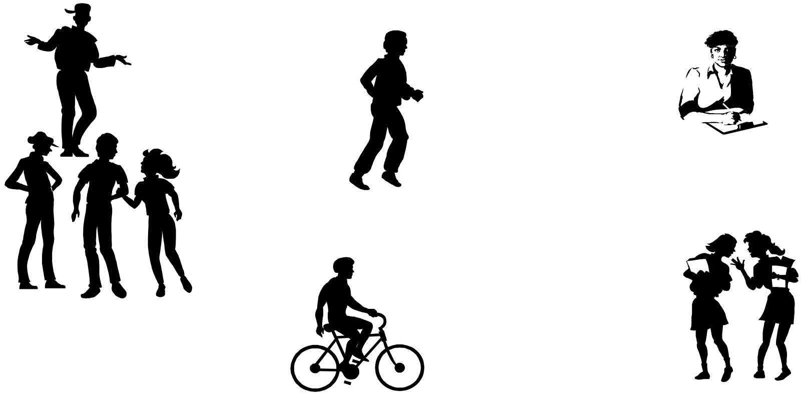
- Telecom infrastructure in remote areas not available
  - The telefonfjell phenomenon ...
- Use of satellite connections is too expensive
- Use of P2P ad-hoc messaging can build an alternative infrastructure
- all participants contribute

  and share task of message delivery
  - Mountain hiking
  - Developing countries
  - Sea, Jungle, ...
  - Cheaper messages
  - Games

Hardangervidda, Sep 2001, Arnt Flatmo
Hårteigen, 1690m, seen from Torehytten

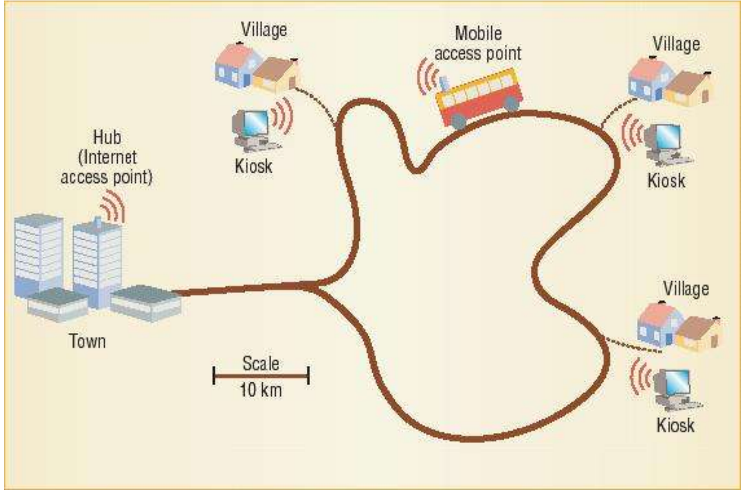
# Basic Idea for the HikerNet

- People move and meet!
- All participants carry a device
  - Integrated into cell phone or other items
  - Messages are carried with the device
- When participants meet messages are exchanged automatically using radio transmission
- Message replication
- Handy as user interface



# Related Technologies

- DakNet
  (MIT MediaLab)
- ZebraNet Wildlife Tracker
  (U Princeton)
- Mobile Ad-hoc Networks (manet)
  (IETF Working Group)
- FleetNet
- Cybiko Wireless Chat
- Email, SMS, MMS, ...
- Peer-to-Peer: Gnutella, Freenet, Eternity Services, ...

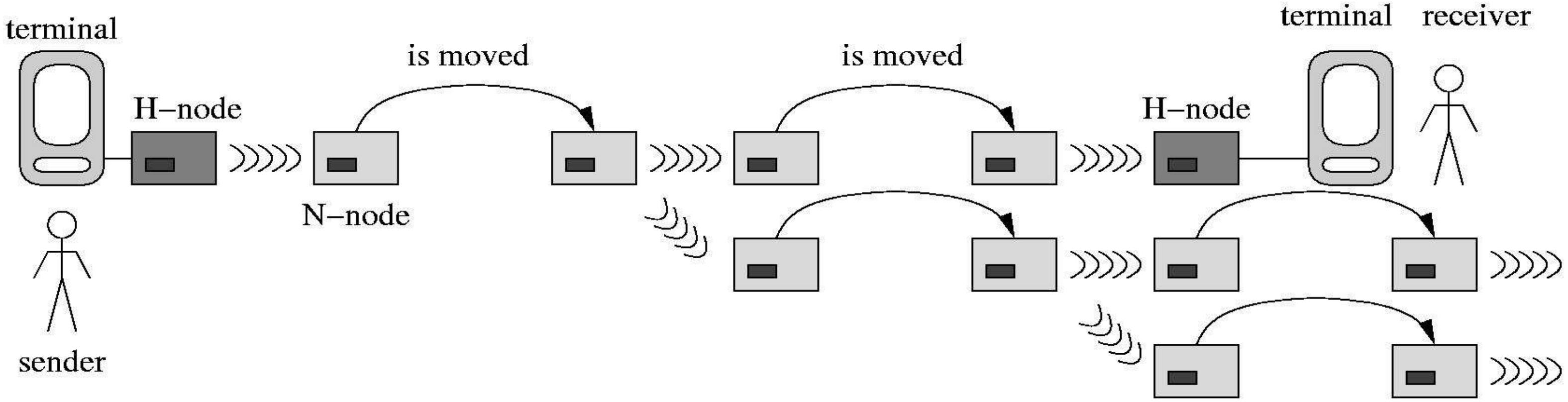


FleetNet

CYBIKO™ WIRELESS INTER TAINMENT COMPUTER

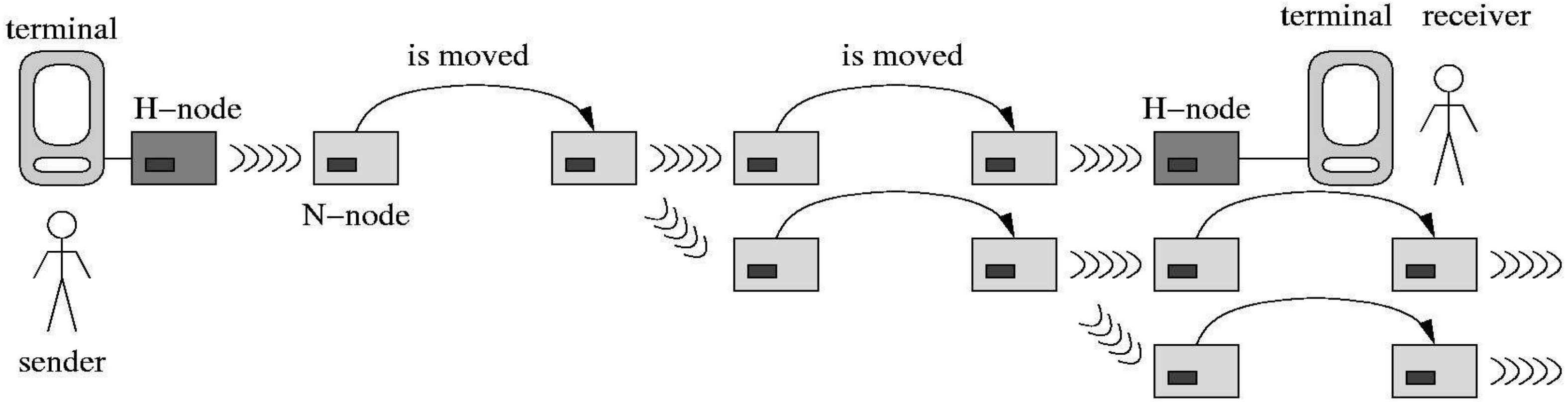
# Principles for the HikerNet

- ▶ Ad-hoc peer-to-peer
- ▶ Store and forward of messages
- ▶ Use movements of participants
- ▶ Non-time critical messages only



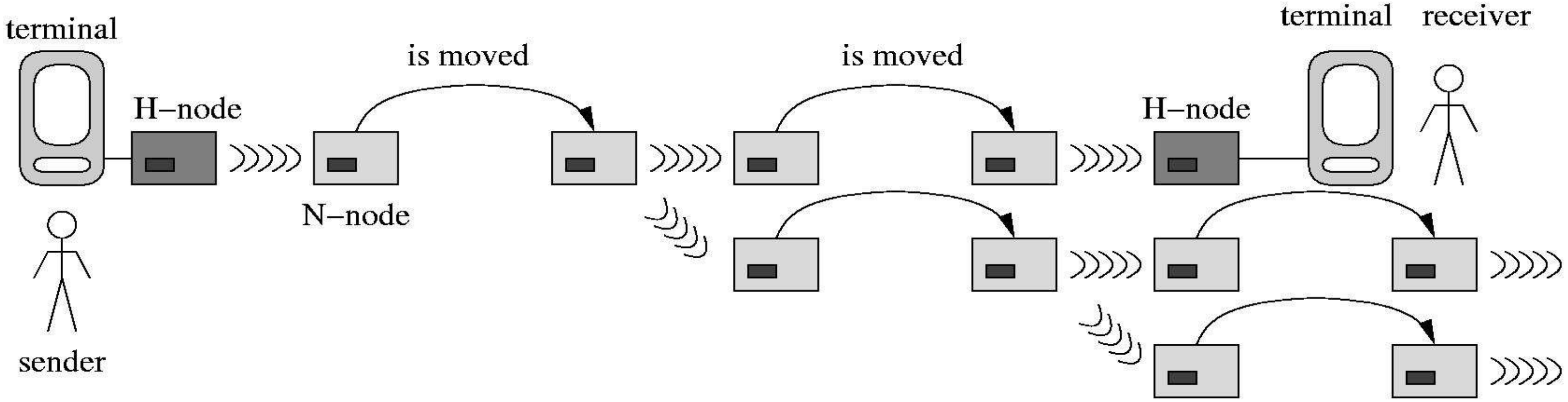
# HikerNet

- Based on roles: Terminal, H-node, N-node
- User writes message on terminal
- H-node handles messages for one user
- N-nodes transport the messages



# HikerNet (2)

- To types of messages: MSG, ACK
- Messages identified by unique ID
- Protocol parameters
  - TTL (times to live)
  - TTR (times to replicate)
  - Expiry date



# Extensions to the HikerNet

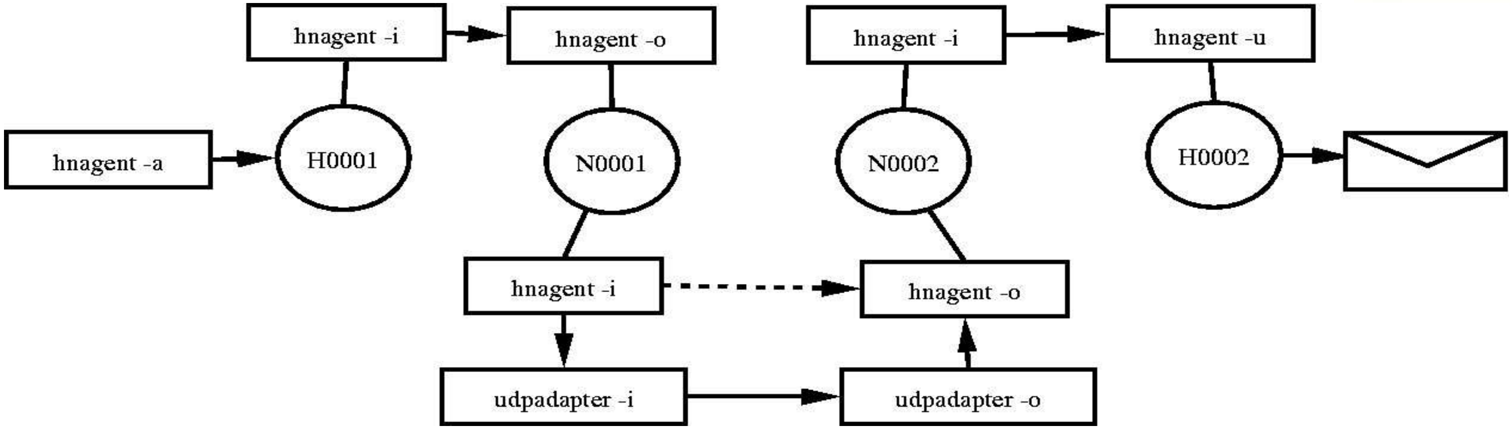
- Stationary N-nodes (message hubs)
- Stationary relays (N-nodes with several manifestations)
- Bridges (stationary relays that connect larger areas)
- Gateways (to other services, e.g., Internet email)
- Broadcasting (radio) of messages with carousel
- Publicly available terminals
- Attach N-nodes to moving objects / animals

# Service examples

- messaging (text, images)
- Voice, message service
- Automated messages (traffic, public transportation, …)
- News messages
- Collective collecting of data (traffic info, movies)
- Tracking (GPS/timestamps messages)
- Anonymous chat
- Games and communities (collecting music?)

# The Prototype Implementation

- HikerNet implementation written in C for Linux
- hnagent (uses pipes for input / output)
- can use “adapter” for protocols
- can use pendrive for transporting messages

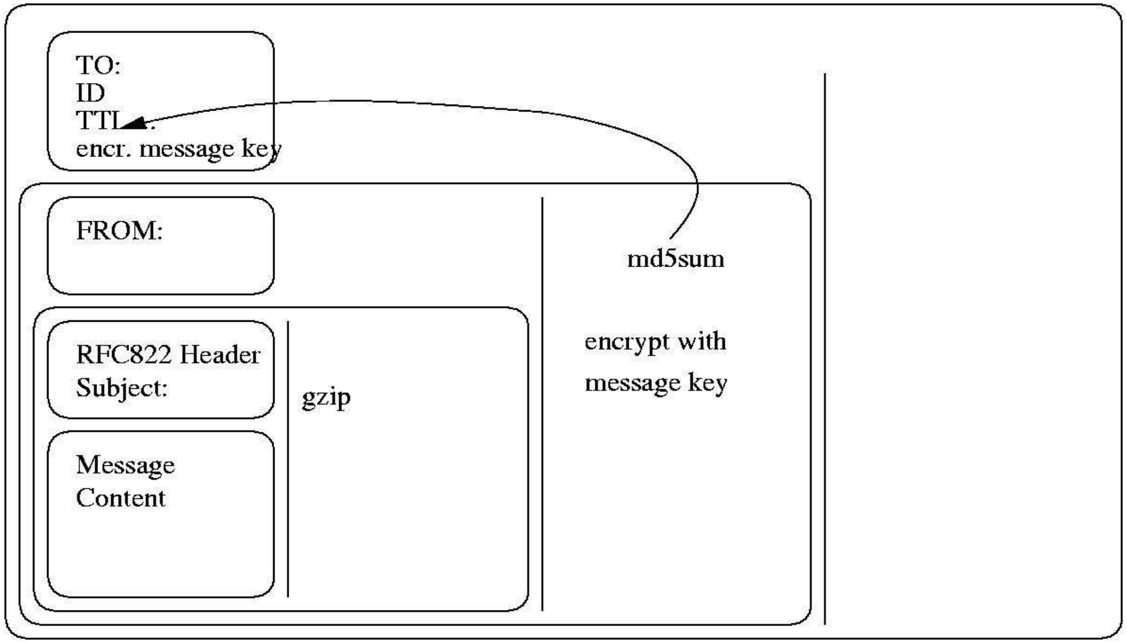


# Security in the HikerNet

- Security =
  - Confidentiality + Integrity + Availability
- Important for the HikerNet:
  - Tracability / Authenticity
  - Anti-Spam
  - Privacy (HikerNet can unwantedly leak information)
- Encrypted messages
- National / international legislation

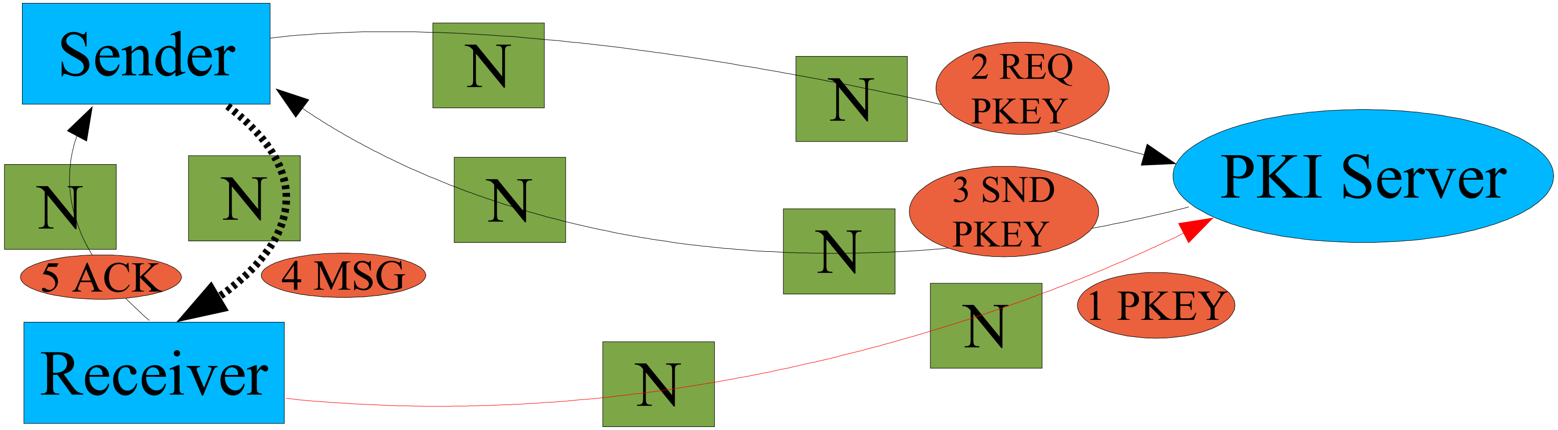
# Message Format

- Messages are encrypted with message key
- Only receiver address and necessary information in visible header



# PKI for HikerNet

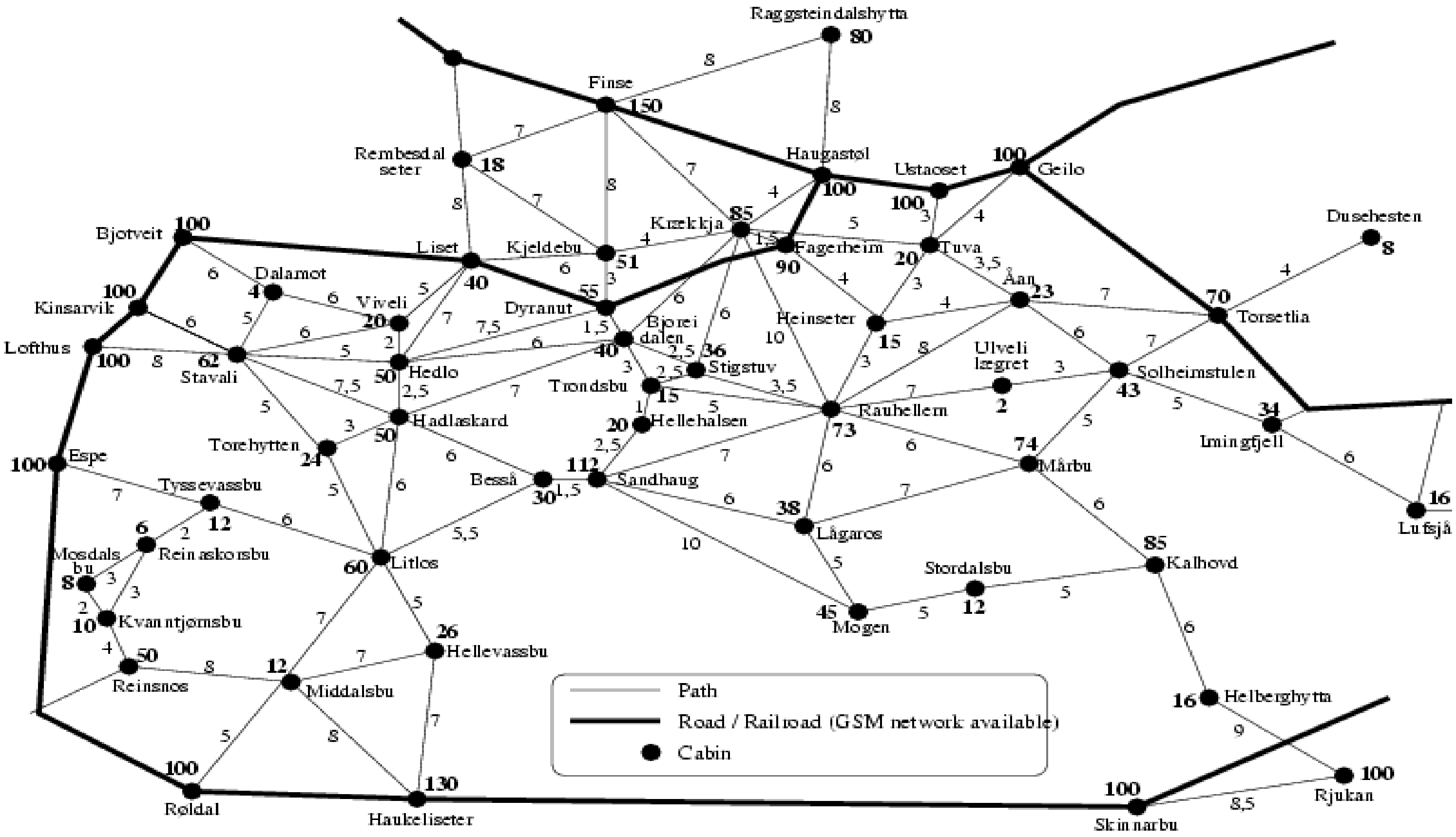
- Each H-node has private/public key pair
  - Encryption / authentication
- Central server keeps data base of public keys
  - Request public keys from server
  - Mechanisms for changes of public keys



# Can HikerNet work?

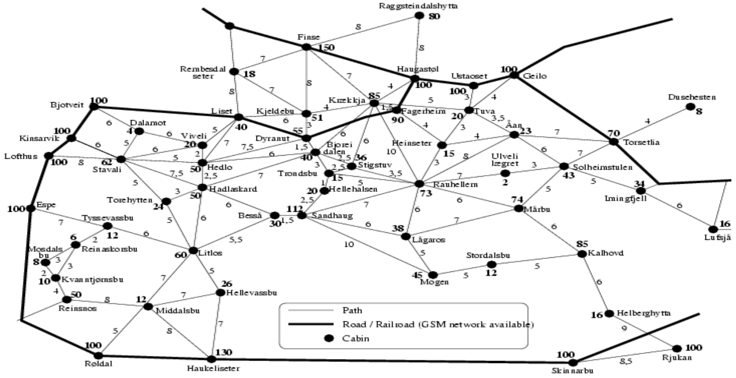
- Simulation of the HikerNet
  - before deployment
- Parameters
  - system parameters (TTL, TTR, Expiry date)
  - #users / #nodes
  - Which hardware (memory, processor, ...)?
  - Delivery time
  - How many messages do arrive?

# Topology of the simulated network



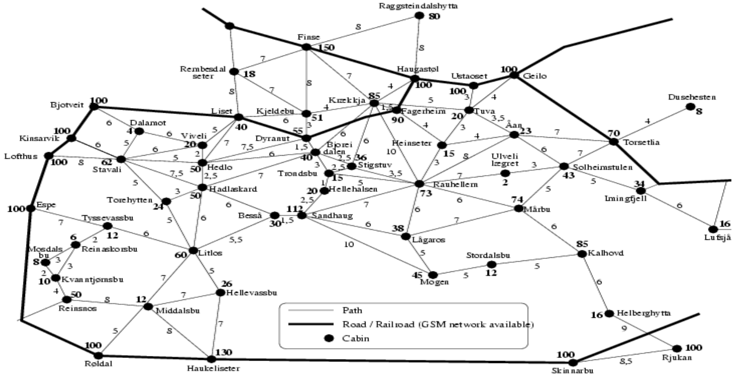
# Simulation Design (1)

- Nodes communicate once a day, at the cabins
- All nodes move to a neighboring cabin once a day
- Choice of next cabin:
  - Random neighboring cabin
  - Weighted neighboring cabin (dependent on #beds)
- Stationary nodes



# Simulation Design (2)

- There are simulators for movements of hikers in mountain areas!
  - AlpSim (Gloor, Mauron, Nagel, 2003)
  - RBSim (Gimblett, Richards, Itami, 2001)
- Used for applications in tourism
- Take interest in area into account



# Architecture of the simulator

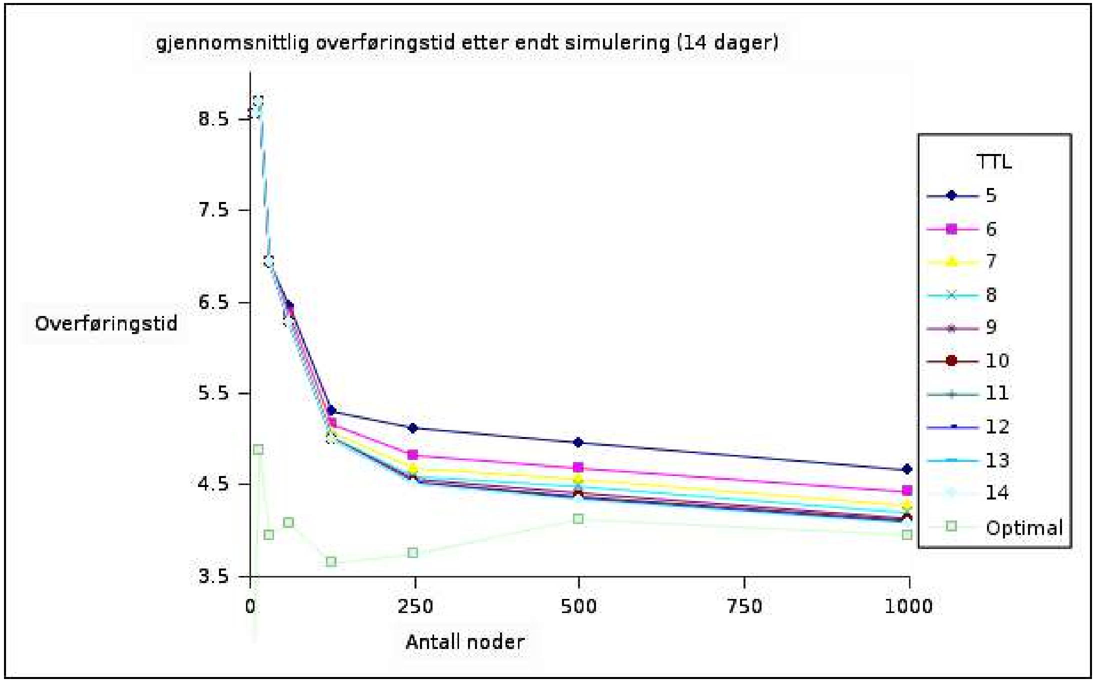
- Simulation designed by Erlend Garberg @ Ifi
- Two components
  - Hiker-movement component
    - Simulation of hiker movements, meetings
  - Communication simulation (CS)
    - Simulates communication between nodes
    - Message generation
    - Calls existing HikerNet prototype
- HikerNet implementation written in C for Linux
- Simulation written in python

# Measurements

- Delivery time
- Percentage of arrived messages
- Memory usage
- Number of messages in network
- Do stationary nodes have an influence?

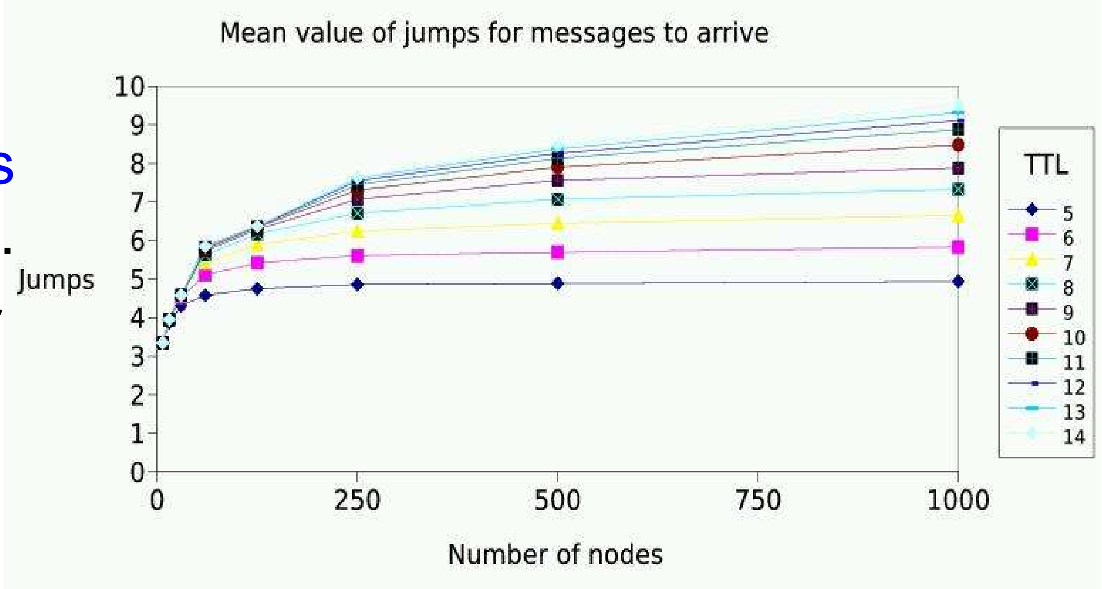
# Results – Delivery time

- Delivery time is reduced when number of nodes increases.
- Delivery time is reduced when TTL is larger (significantly for TTL < 10)
- Average delivery time graph stabilizes towards 4 days, and for TTL=9 and 250 nodes.



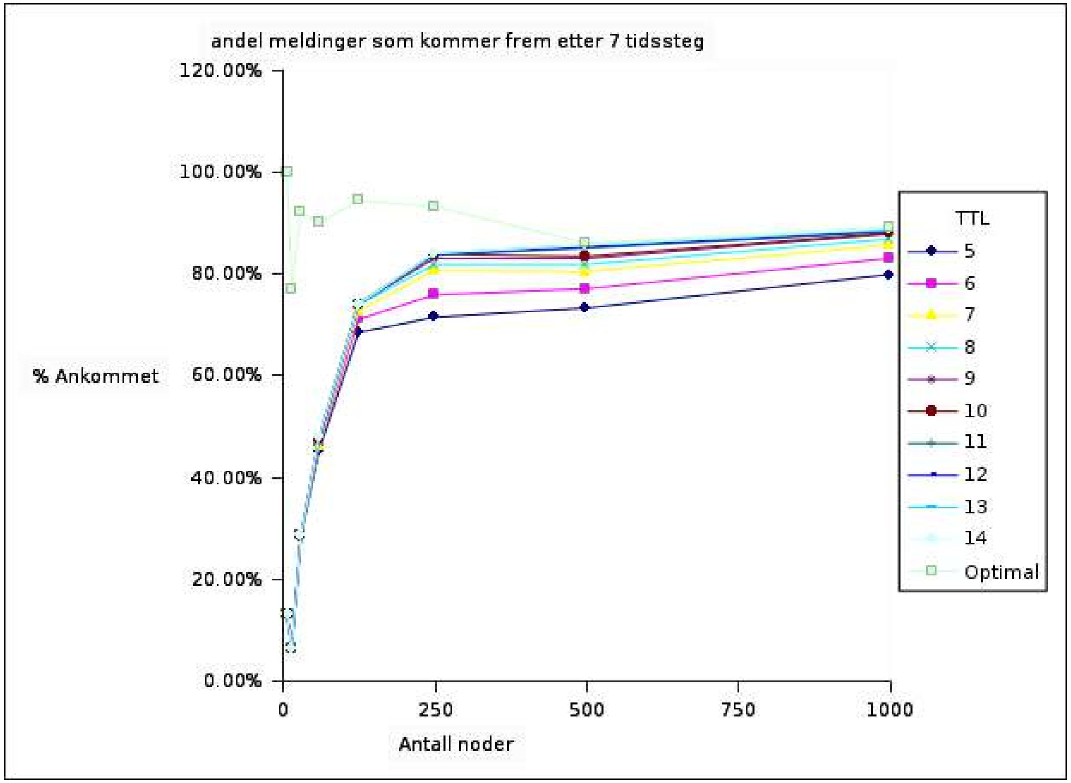
# Results – Jumps

- While delivery time is reduced when number of nodes or TTL increases,
- The mean number of jumps increases at the same time.
- Reason: TTL limits number of jumps; however: pathes with additional jumps are faster in time.



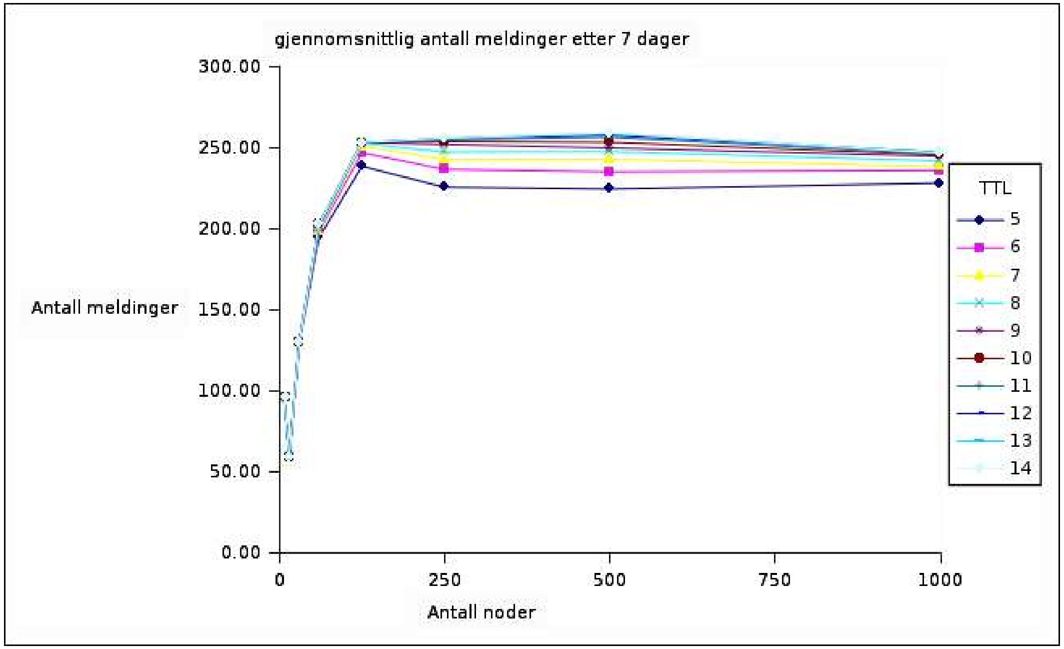
# Results – Arrival rate

- Arrival rate of messages rises when number of nodes increases
- Arrival rate of messages rises when TTL (up to TTL<10)
- After one week over 80% of the messages have arrived.



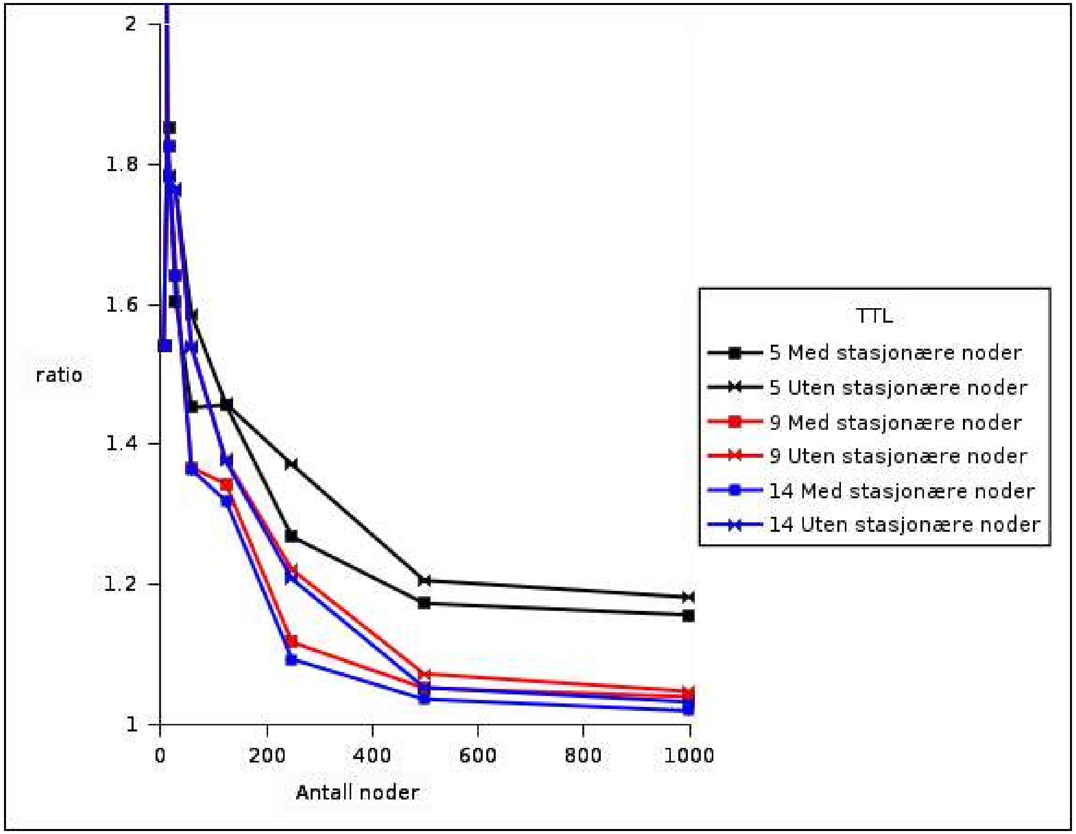
# Results – Number of messages in network / Memory usage

- The number of messages in the network rises when number of nodes increases.
- The number of messages in the network rises for larger TTL-values.
- Memory usage and number of messages are proportional.



# Results – Stationary nodes

- Stationary nodes reduce the number of nodes necessary for the same performance.
- For small numbers of nodes stationary nodes give better performance.



# Conclusions

- For sufficient number of users (>100) the average delivery time is close to optimal delivery time.
  - It takes >10 days until all messages have arrived.
  - The users must accept that messages do not arrive.
  - The users must accept that delivery time varies.
- Performance is dependent of topology.
- Hardware requirements are modest.
- TTL=9

# Future work and considerations

- Implement security-infrastructure
- Implement HikerNet in Java for mobile phones
- Adjustments of the HikerNet to other applications and scenarios
- Games / Communities
  - Distribution of music, like collector cards
  - Communication hotspots attract other business
  - Is communication speed high enough for today's user in mass market?