Norsk Regnesentral
NORWEGIAN COMPUTING CENTER

www.nr.no

The HikerNet

can we show that it works?

Wolfgang Leister, Norsk Regnesentral

CREDO

Uppsala, November 2007

When telecommunication is out of reach ...

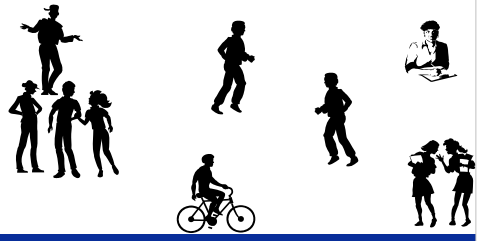
- No telecom infrastructure in remote areas ...
- Use of satellite connections is too expensive
- Build alternative messaging infrastructure
  - Based on P2P ad-hoc messaging
- all participants contribute

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

Norsk Regnesentral
NORWEGIAN COMPUTING CENTER

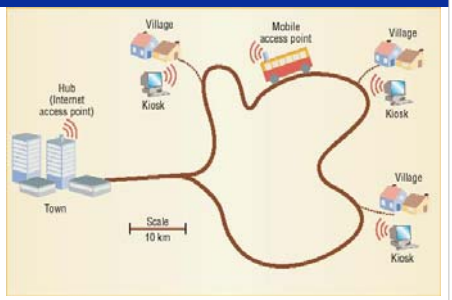
Basic Idea for the HikerNet

- People move and meet!
- All participants carry a device
  - e.g., cell phone or other items
  - Messages are carried with the device
- When participants meet:
  - Exchange messages automatically using radio transmission
- Use message replication
- Mobile phone as user interface



Related Technologies

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

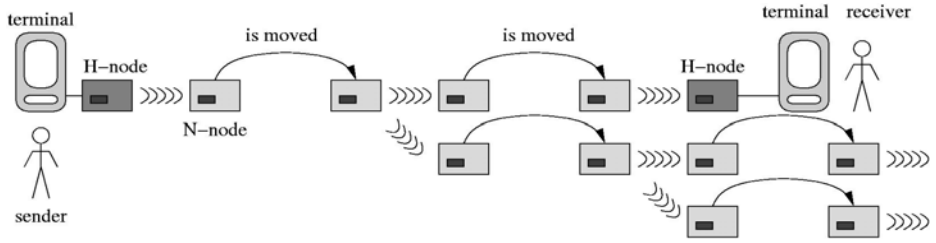


FleetNet

CYBIKO™ WIRELESS INTER TAINMENT COMPUTER

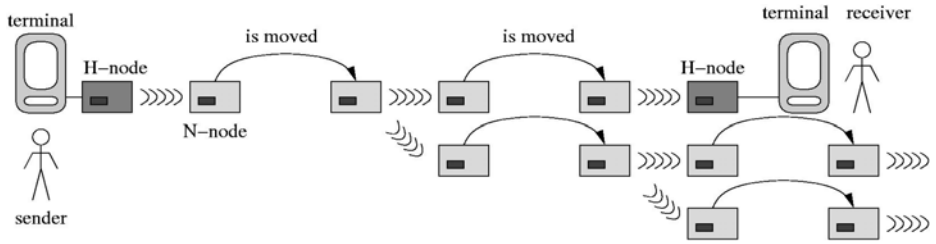
HikerNet principles (1)

- Store and forward of messages
- Use movements of participants
- Based on roles: Terminal, H-node, N-node
- H-node handles messages for one user
- N-nodes transport the messages



HikerNet principles (2)

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

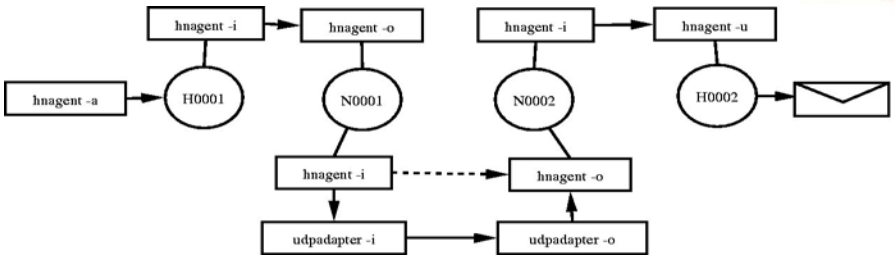


Extensions to the HikerNet

- 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

The Prototype Implementation

- HikerNet implementation written in C for Linux
- hnagent (uses pipes for input / output)
- can use “adapter” for underlying protocols
  - e.g., bluetooth, udp, tcp, ...
- can use pendrive for transporting messages

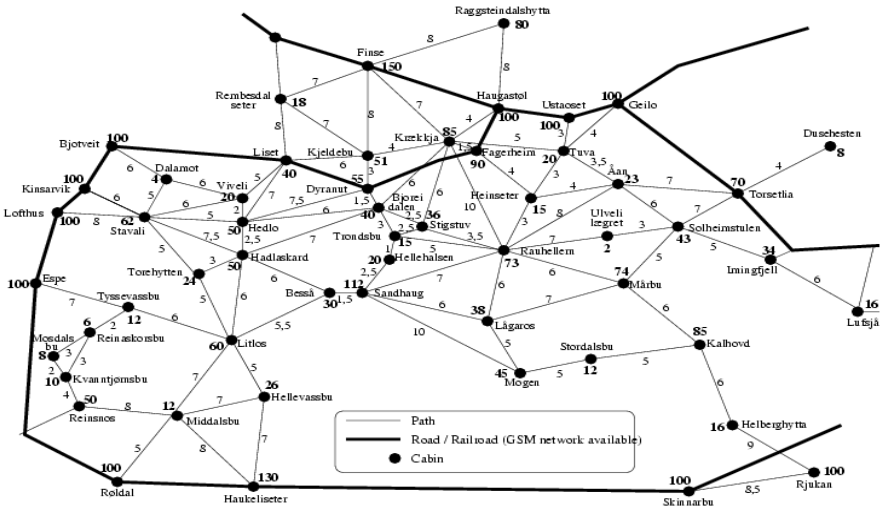


Can HikerNet work?

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

Norsk Regnesentral
NORWEGIAN COMPUTING CENTER

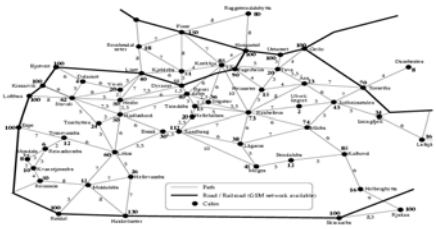
Topology of the simulated network



Norsk Regnesentral
NORWEGIAN COMPUTING CENTER

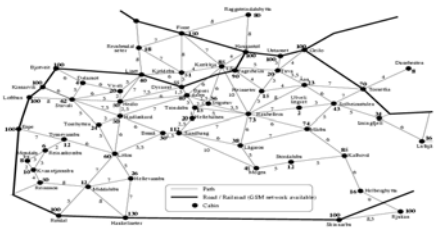
Simulation Design (1)

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



Simulation Design (2)

- Simulators for movements of hikers in mountain areas:
  - AlpSim (Gloor, Mauron, Nagel, 2003)
  - RBSim (Gimblett, Richards, Itami, 2001)
- Used for applications in tourism
- These take interest in area into account

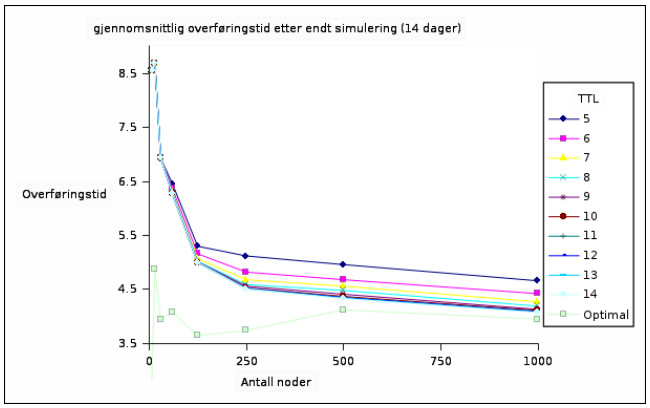


Architecture of the simulator

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

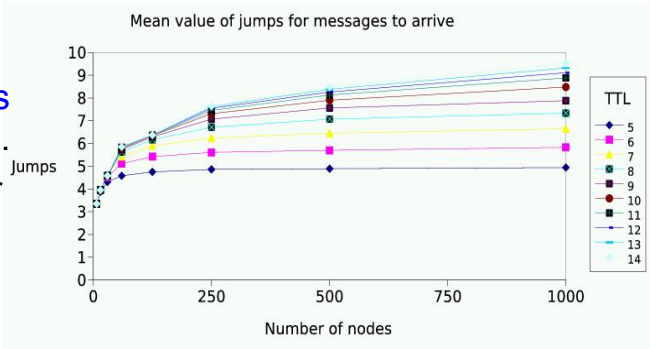
Results – Delivery time

- 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.



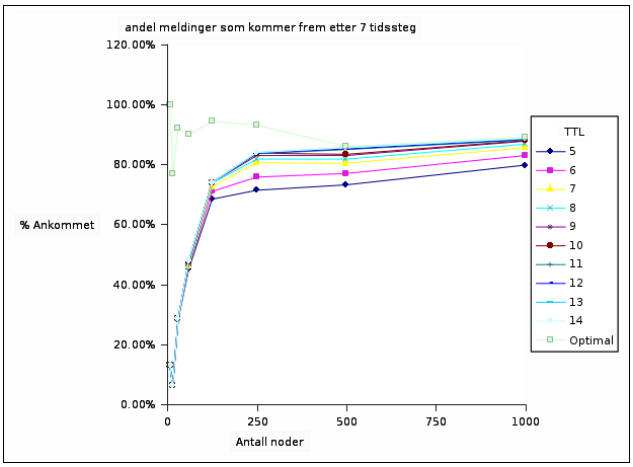
Results – Jumps

- 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.



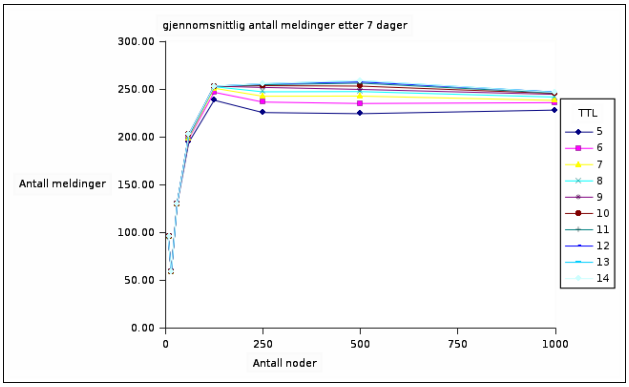
Results – Arrival rate

- 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.



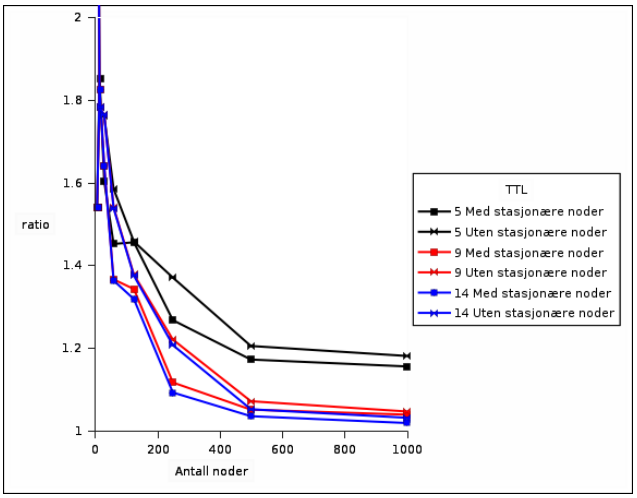
Results – Number of messages in network / Memory usage

- 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 from simulation

- 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

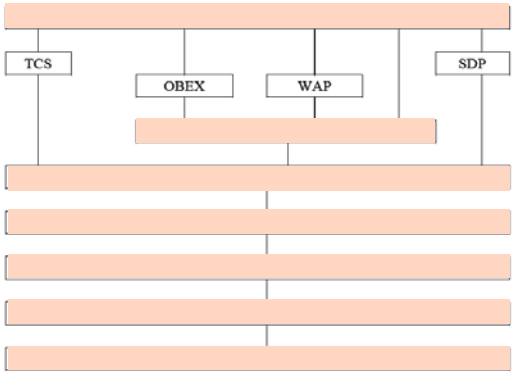
Java / Bluetooth Implementation

- Implementation by Markus Voss @
- Implementation on mobile phones
  - J2ME / JABWT
  - Only one application at a time
- Bluetooth stack
  - RFCOMM service of Bluetooth
- Node Rendez-vous / SPAN
  - to save battery

Bluetooth

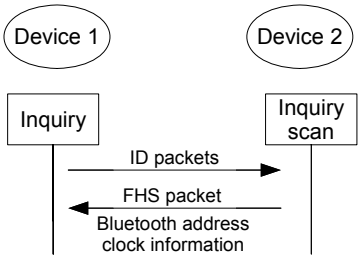
- Wireless communication technology
- Protocol stack
  - Java: JABWT
  - Linux: bluez
- Frequency hopping
- ISM frequency band:
  - 2.4 GHz
- Reaches ca. 10 m

- Protocol stack
- HikerNet uses RFCOMM

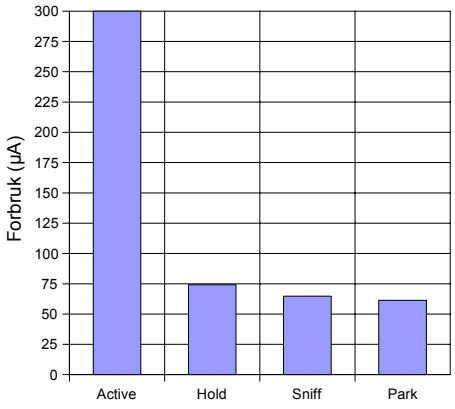


Bluetooth-Modes

- Device detection
  - Inquiry – Inquiry scan
  - >10.24 sec
- Master-Slave
  - Page – Page scan

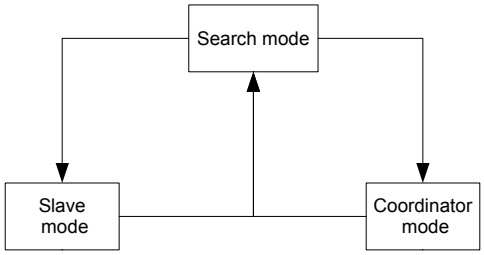


- Connected state:
  - active, hold, sniff, park
  - Energy consumption:



HikerNet – Bluetooth Operation

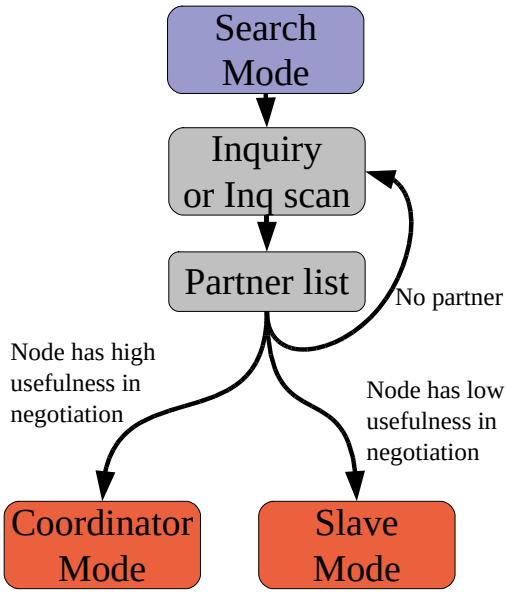
- Uses Node Rendez-vous
  - Modified SPAN
  - Wakeup when needed to save battery
- Three modes



- Search mode
  - Detect other devices
  - Negotiate Coordinator/Slave
- Coordinator mode
  - Detectable
  - Communication with slaves
- Slave mode
  - Not detectable
  - Communication with coordinator only

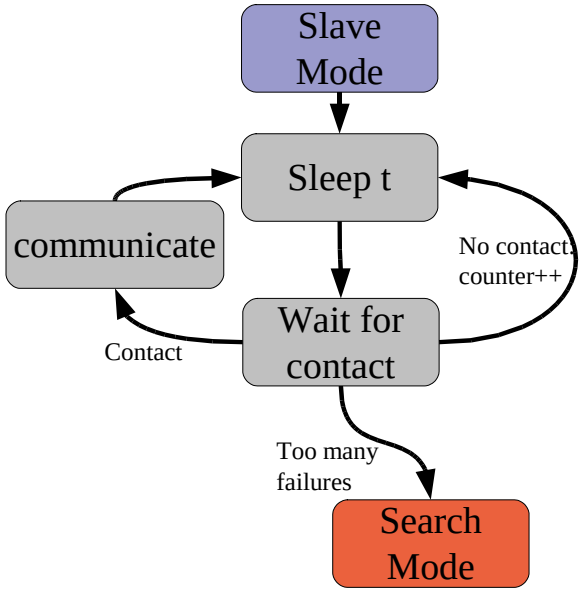
Search Mode

- Usefulness-factor
  - Decides whether coordinator or slave
  - Depends on memory, battery, #simultaneous connections, history, ...

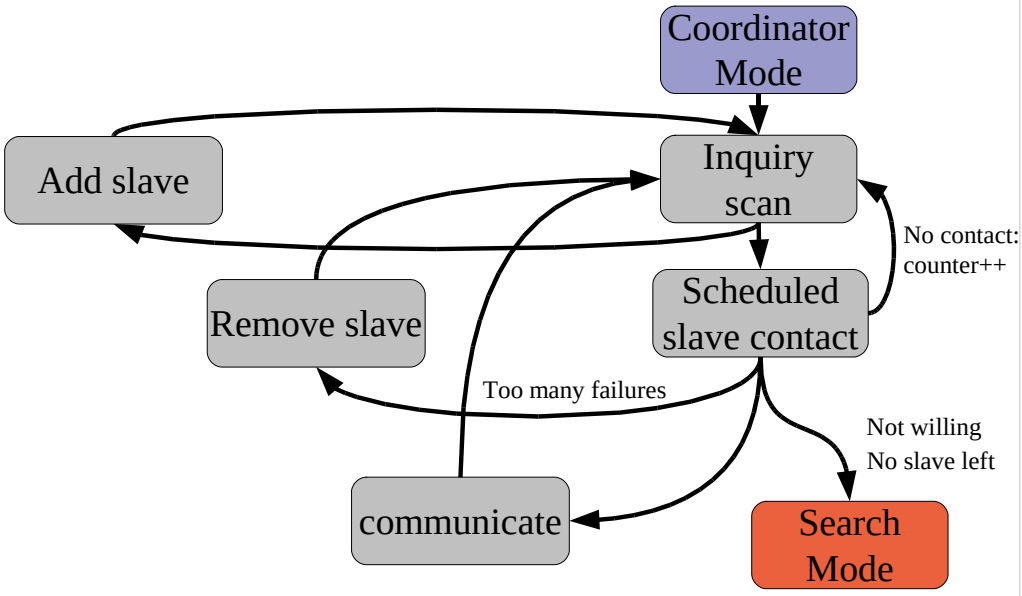


Slave Mode

- Slave
  - In energy saving mode most of the time
  - Java cannot switch off bluetooth entirely
  - Communicates with one coordinator only



Coordinator Mode



Implementation

- Implementation on simulator works
- Both test phones had limitations / bugs
  - Stack overflow /Phone crash when many bluetooth connections
- Measurements could not be performed
- Theoretical evaluations suggest energy saving

Current work and considerations

- Implementing message encryption and security-infrastructure based on
  - Message keys
  - private/public key pairs
- Can CREOL give answers to
  - Can HikerNet work?
  - Can we say more about delivery rates, delivery time, #hops, ... ?
  - What about energy saving?
  - Is the SPAN-variant always working?


CREDO