

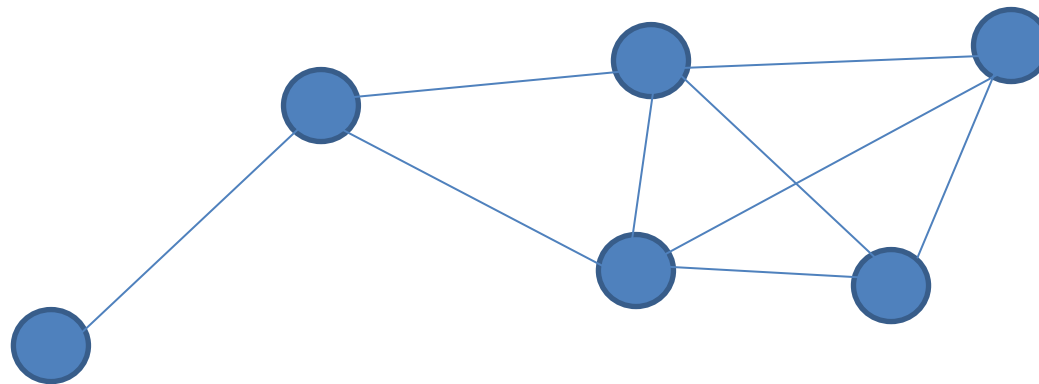


Ad - hoc network for Maemo Testbed

Malakhov Alexander
SUAI

Ad-hoc network

- Ad-hoc network is a decentralized network
 - No central server
- Dynamic topology
 - Node could join & leave randomly
 - Node could change position randomly



Usage scenario

- Advantages
 - Fast deployment
 - Free of cost
 - Geographically closer people - closer socially (in some cases)
 - Conference
- Usage scenario
 - File sharing
 - Http server with business card
 - Finding friends
 - Communication (phone, chat)

Our project for maemo

- Routing protocol
- Member management protocol
- User application (Ad-hoc client)
 - Chat
 - Phone
 - File search & exchange protocol

Protocols

- Routing
 - Mad-hoc implementation of AODV
 - Based on libpcap
 - Doesn't require new kernel or any kernel patching
- Member management protocols
 - Name $\leftarrow \rightarrow$ IP
 - Every user has unique symbolic name
 - Protocol resolve name to IP
 - Contact list (e.g. as in ICQ)
 - User has list of names (e.g. friends)
 - Availability of each friends should be shown (red, green)

WORK IN PROGRESS

Ad-hoc client

- GTK interface (contact list, name resolving)
- Chat
- Phone (Linphone port)
- File sharing protocol (Gnutella port)

How to test all this stuff?

- Real life (10 people with N810 walking in a yard)
 - Difficult
 - Long
 - Experiments are unrepeatabe
- Simulation
 - Approximate results
 - Requires development of simulation model for software components (e.g. routing details)

Our approach: testbed

- Put several N810 on table
- Run real software (routing, member management, ad-hoc client)
- Emulate topology changes
- Emulate user activity

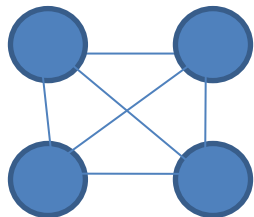
How it looks...



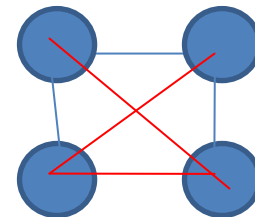
How it works: mobility emulation

- Of course all nodes hear each other (they are on the same table)
- But our add-on to mad-hoc dynamically edit neighbors table according to some scenario
 - Add/Remove neighbors

“Real” connectivity



after work our add-on



How it works: in general

- Scenario generator creates scenario for each node using traffic & mobility models as input
- Scenario is uploaded to all nodes
- Scenario player on each node replays scenario and save the logs
- Main disadvantage: MAC layer works as in worst case (all nodes conflict with each other)

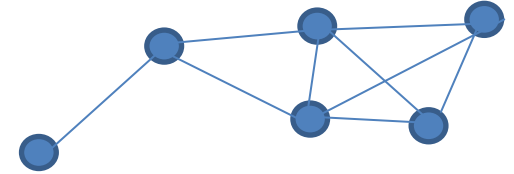
How it works

Traffic model

Mobility model



Scenario



Scenario player



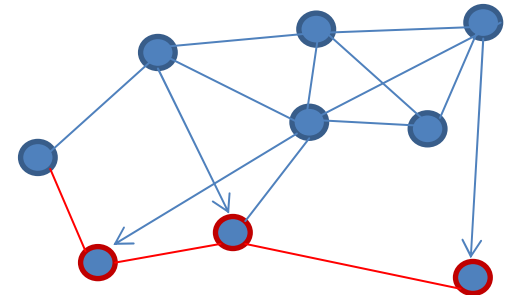
Avg. delay

Throughput

Battery consumption



Logs



Scenario generator

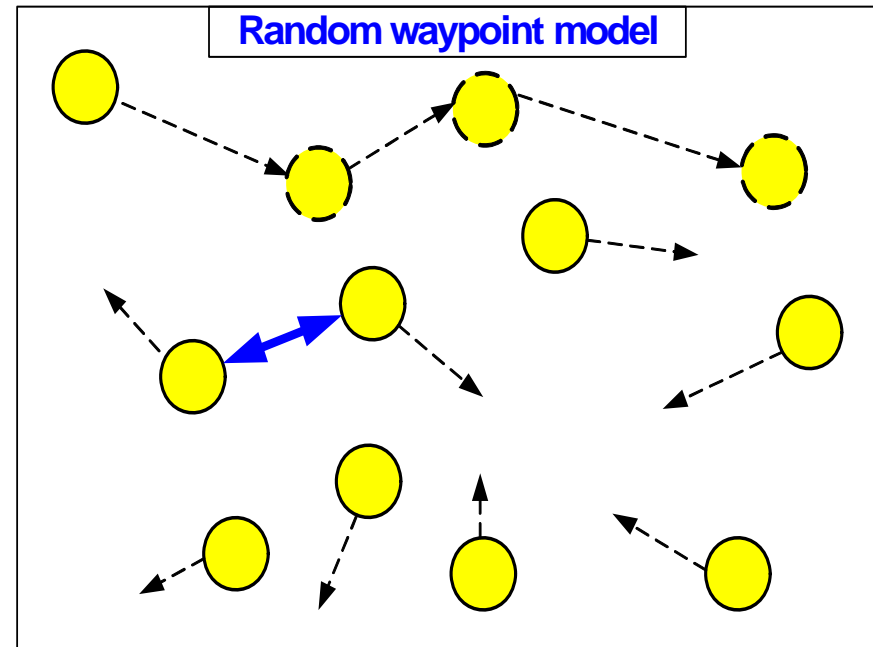
- Random waypoint model

- Characteristic

- Network size
 - Field size
 - Nodes speed
 - Waiting time
 - Experiment time

- Scenario Example:

- 10.51.3.1
 - 1000 2000
 - Means node with IP=10.51.3.1 is invisible for current node from 1000ms to 2000ms



Status

- Done:
 - Scenario generator for topology
 - Scenario player for topology
- In work:
 - Scenario generator and player for traffic