SANS
A Simple Ad hoc Network Simulator

Nicolas Burri
Roger Wattenhofer
Yves Weber
Aaron Zollinger
WLAN at Home (Infrastructure Mode)
WLAN in the Woods (Ad Hoc Mode)
Routing

$G=(V,E)$

Multihop routing

$S \rightarrow t$
## Routing: Internet vs. Ad Hoc Networks

<table>
<thead>
<tr>
<th><strong>Internet</strong></th>
<th><strong>Ad Hoc Networks</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Dedicated routers</td>
<td>• No (or little) pre-deployed infrastructure</td>
</tr>
<tr>
<td>• Mostly static structure</td>
<td>• Highly dynamic topology</td>
</tr>
<tr>
<td>• Low packet loss</td>
<td>• High packet loss</td>
</tr>
<tr>
<td>• Unlimited energy</td>
<td>• Battery lifetime</td>
</tr>
</tbody>
</table>
Consequences

• Routing and Medium Access Control (MAC) are difficult tasks in ad hoc networks
  – Existing solutions for wired networks are not well suited for use in ad hoc networks
  – New paradigms and algorithms need to be developed

Lots of work for coming generations of computer scientists

But how should we teach them?
Teaching the Concepts of Ad Hoc Networks

Theory
- Graph Theory
- Existing Algorithms
- Electrical Engineering

Practical Exercises
- Implementation of Applications

Existing Systems
Practical Exercises

• **Requirements**
  – One or more devices offering a radio network interface
    • Notebook
    • PDA
  – Test network consisting of several nodes
Students as Nodes

• Each student represents a node of the graph
  – Nearly no control over the topology
  – Requires many participants and a lot of space
  – Results of experiments are not reproducible
  – Debugging is almost impossible

• Is there a better solution?
  Simulation is an alternative
Simulation

- Each instance of the program represents a **virtual node**
- The simulating PC **controls** the network **topology**
Existing Simulators

• Various simulators exist focusing
  – on functionality
  – not on usability for untrained users

• ns2
  ✓ General purpose simulator for all network layers
  ✓ Very powerful
  ❌ Requires special scripts
  ❌ Highly complex to use
SANS

- SANS has been designed for use in exercises

  ✓ Intuitive “Point and Click” user interface
  ✓ Support for generic Java programs
  ✓ Platform independent
  ✓ Real-time simulation execution
  ✓ Programs developed in SANS also run on real hardware
  ✓ Small size (70 kB)

  ❌ Limited underlying communication protocol
  ❌ Limited scalability
SANS: Interface
SANS: Interface
• Generic Java programs which also run on real hardware
  – Students may run their applications on notebooks or PDAs
• Communication must be UDP
  – For a good simulation of the properties of ad hoc networks, UDP Multicasts are well suited
SANS: Interface
SANS: Adding Edges
SANS: Link Properties

Transmission delay

Packet drop policy

Link direction
SANS: Flooding Example
SANS: Flooding Example
SANS: Flooding Example
SANS: Flooding Example
SANS: Flooding Example
SANS: Console Output

Receiver 10 started
Receiver 11 started
Receiver 12 started
Receiver 13 started
Receiver 14 started
Receiver 15 started
Receiver 16 started
Sending from 1 -> 16 TTL 5
2 --> 10 --> 12 --> 16
SANS in use

• “Mobile Computing” at ETH Zurich
• **Server-less instant messenger** as a long term homework
  – Find other users in the network
  – Send and receive messages to users not within direct communication range
  – Relay messages for other users

• The same task was assigned in three consecutive years
SANS in use

- The number of students was about 90 people and did not fluctuate much between 2002 and 2004
- 2-3 students/team
- Number of handed-in complete solutions by year:

[Graph showing the number of handed-in solutions by year with a note: Not only due to SANS]
The End

Download SANS at

http://dcg.ethz.ch/projects/SANS/Simulator.jar