

SANS

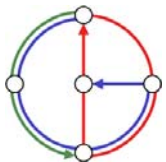
A Simple Ad hoc Network Simulator

Nicolas Burri

Roger Wattenhofer

Yves Weber

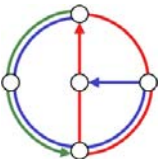
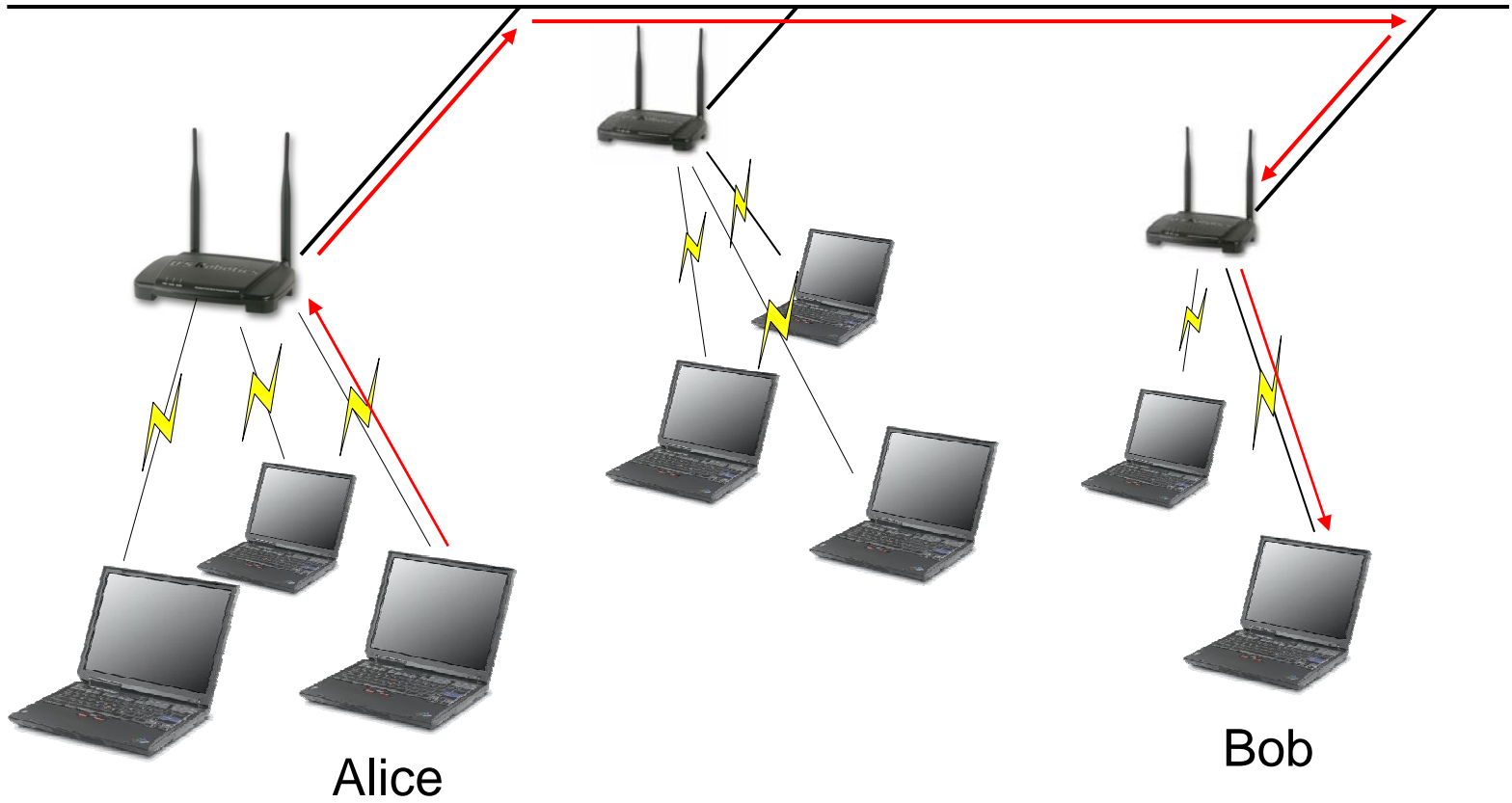
Aaron Zollinger



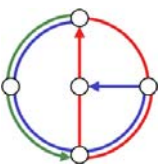
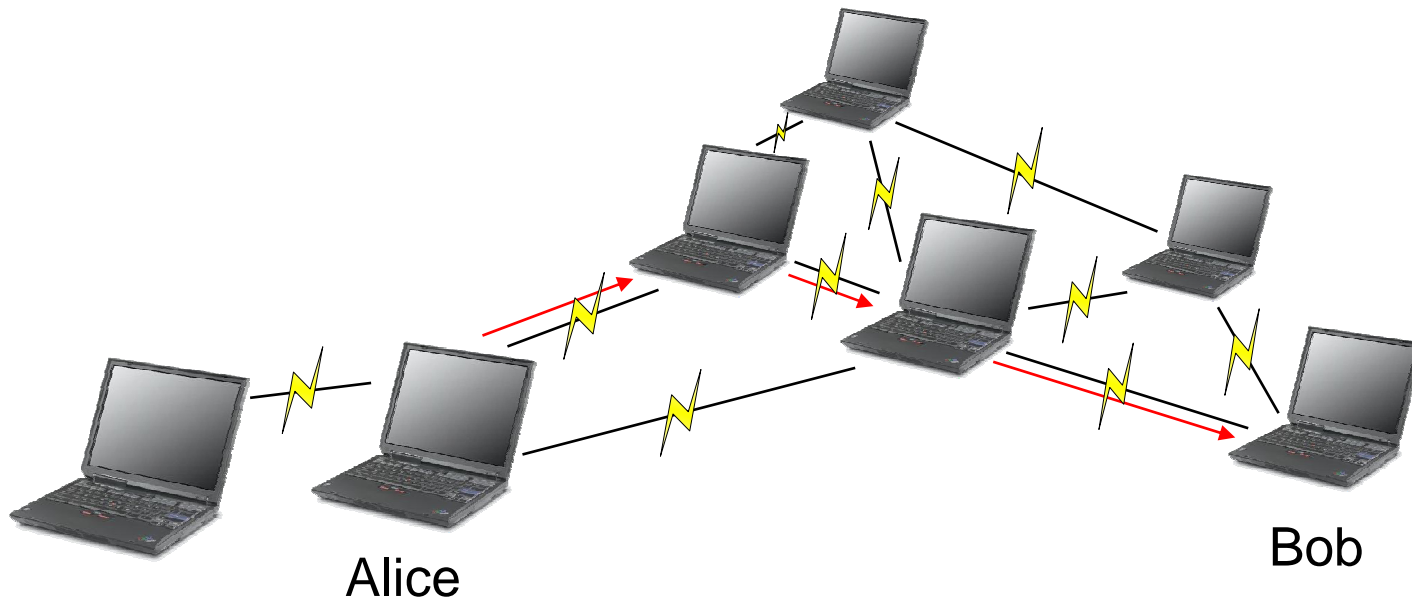
ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

WLAN at Home (Infrastructure Mode)



WLAN in the Woods (Ad Hoc Mode)

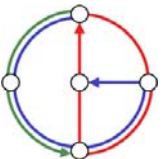
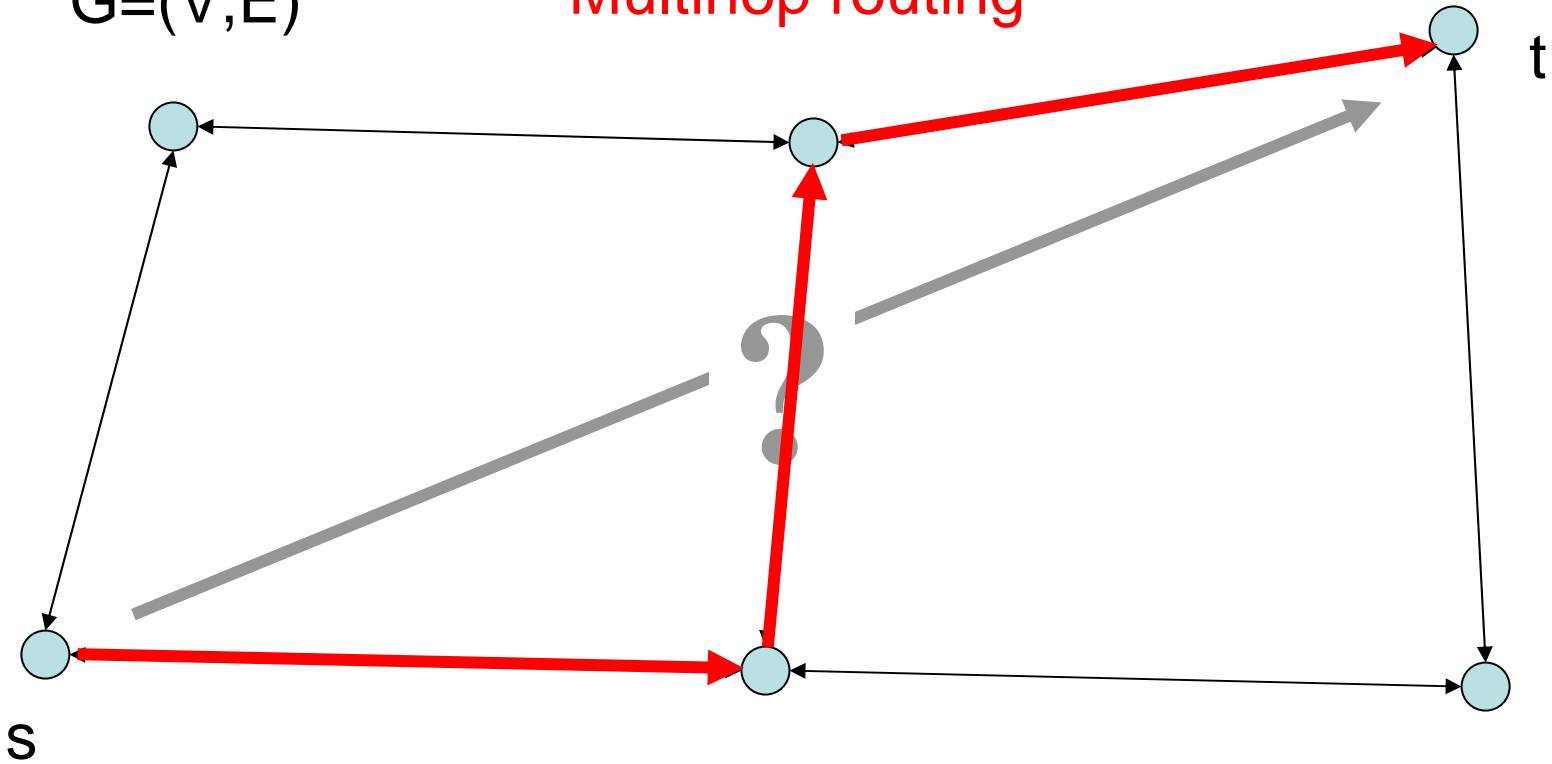


Routing



$G=(V,E)$

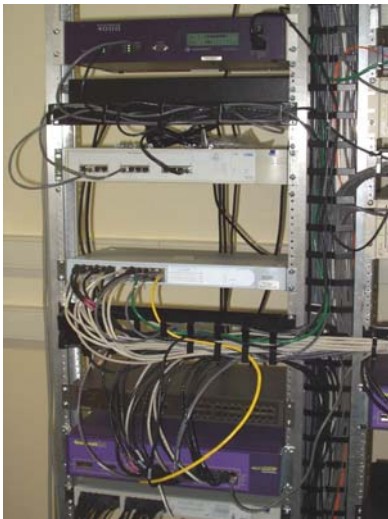
Multihop routing



Routing: Internet vs. Ad Hoc Networks

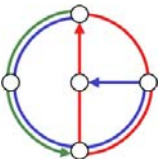
Internet

- Dedicated routers
- Mostly static structure
- Low packet loss
- Unlimited energy



Ad Hoc Networks

- No (or little) pre-deployed infrastructure
- Highly dynamic topology
- High packet loss
- Battery lifetime



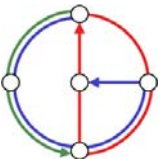
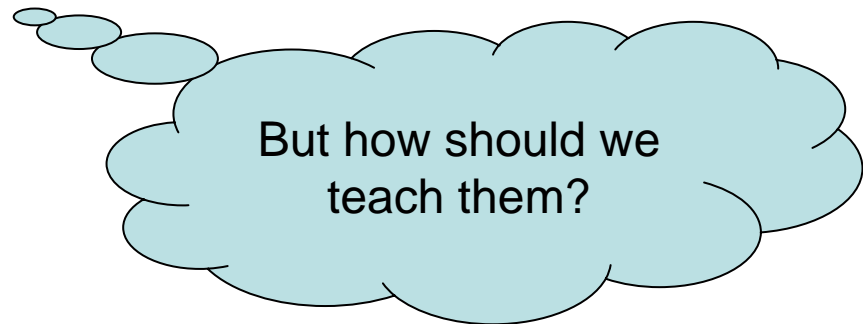
Consequences



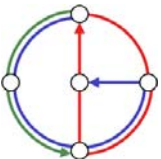
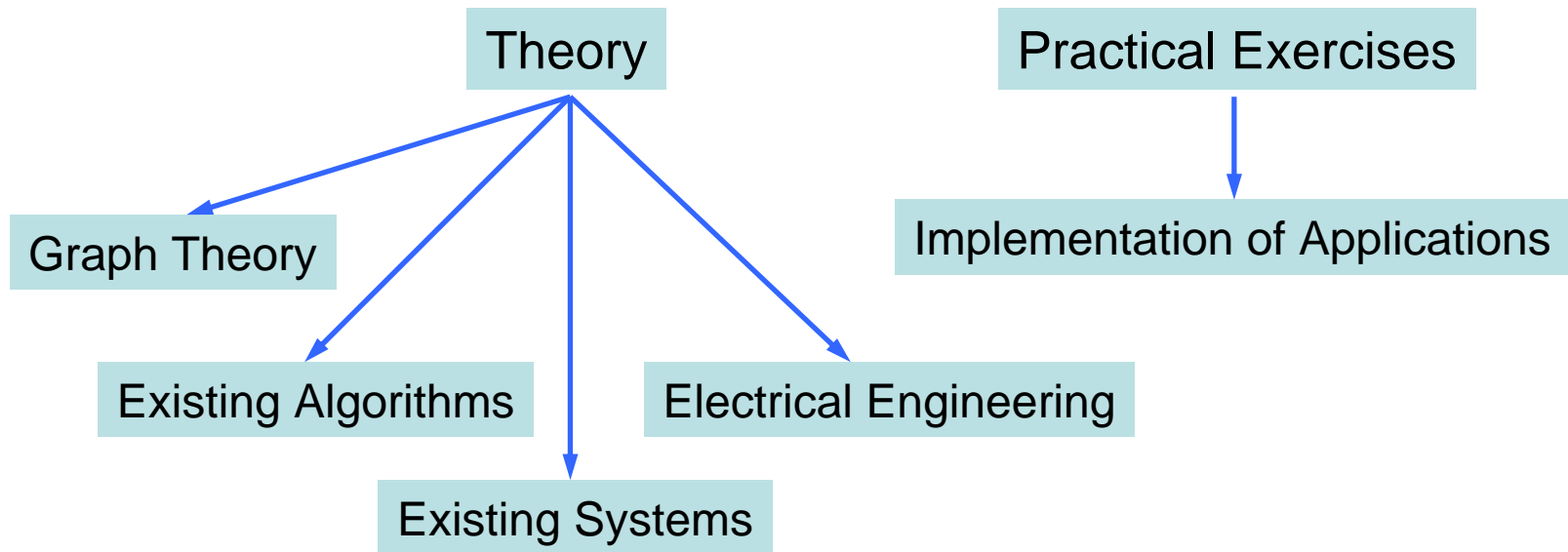
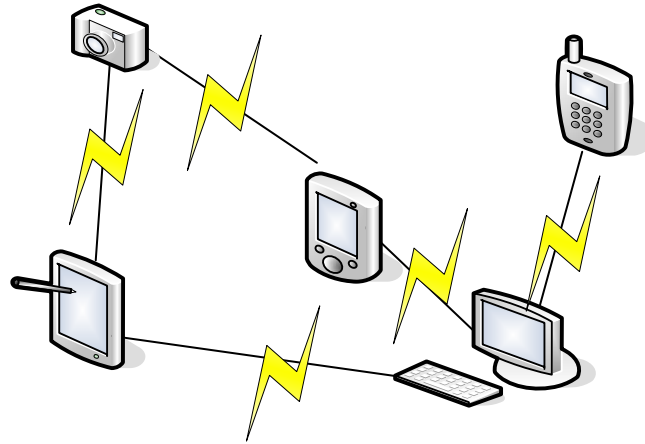
- Routing and Medium Access Control (MAC) are difficult tasks in ad hoc networks
 - Existing solutions for wired networks are not well suited for use in ad hoc networks
 - New paradigms and algorithms need to be developed



Lots of work for coming generations of computer scientists



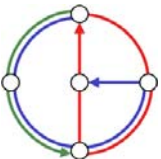
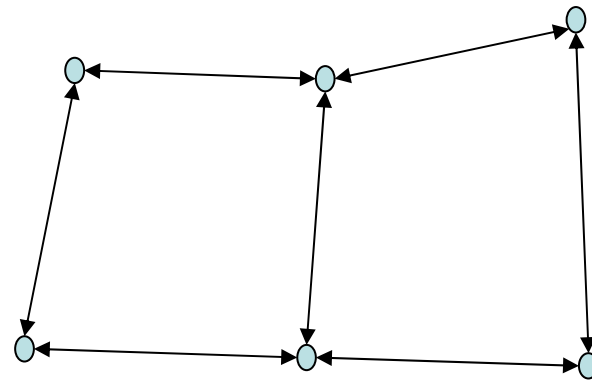
Teaching the Concepts of Ad Hoc Networks



Practical Exercises



- **Requirements**
 - One or more devices offering a radio network interface
 - Notebook
 - PDA
 - Test network consisting of several nodes



Students as Nodes

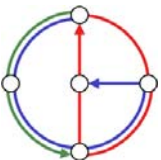
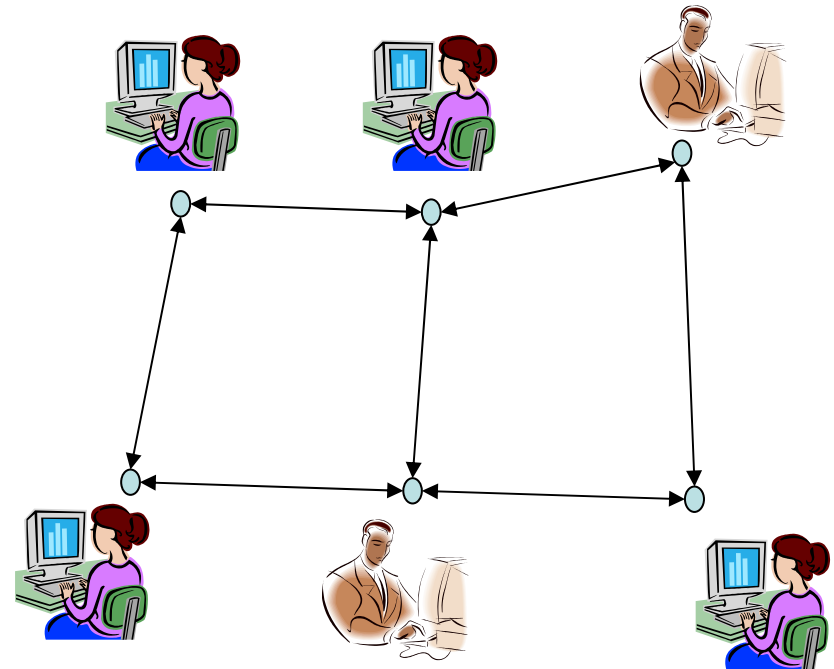


- Each student represents a node of the graph

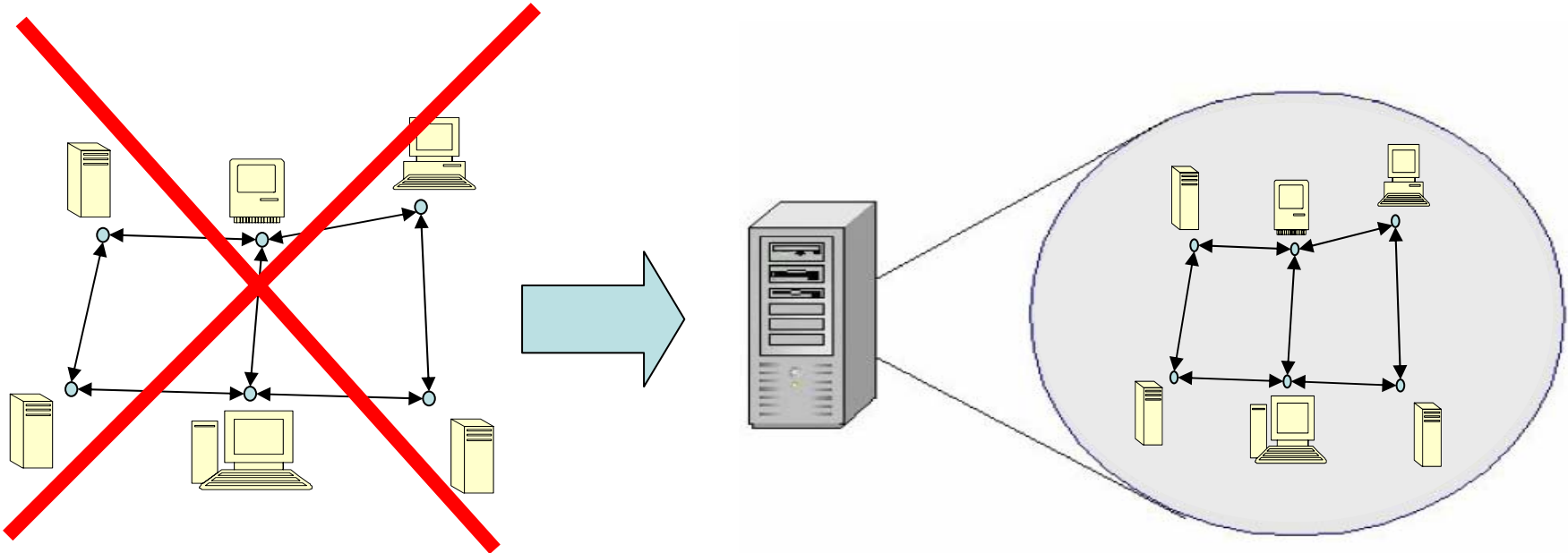
- Nearly no **control** over the **topology**
- Requires **many participants** and a lot of space
- Results of experiments are **not reproducible**
- **Debugging** is almost impossible

- Is there a better solution?

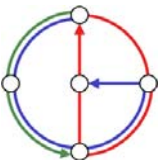
➔ Simulation is an alternative



Simulation



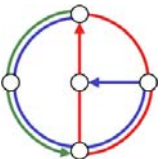
- Each instance of the program represents a **virtual node**
- The simulating PC **controls** the network **topology**



Existing Simulators



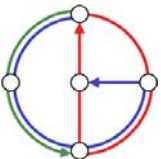
- Various simulators exist focusing
 - on **functionality**
 - **not** on **usability** for untrained users
- ns2
 - ✓ General purpose simulator for all network layers
 - ✓ Very powerful
 - ✗ Requires special scripts
 - ✗ Highly complex to use



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 - Simple Ad Hoc Network Simulator

Statistics

- # of clients: 0
- # of links: 0
- # of packets sent: 0

Edit Graph manually

- Add Client
- Remove Client
- Edit Links
- Start Clients

Load/Save Graph

- New Graph
- Load Graph
- Save Graph
- Save Graph As

Miscellaneous

- Settings
- Hide Output Wnd
- Exit

Edit client

Class name (incl. package): TestSender

Parameters for main():

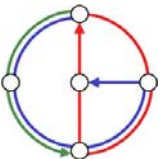
Classpath: C:\JavaSamples

Internal name: Sender

Position: x: 269 y: 295

This client is not yet running

OK Cancel



SANS: Interface

SANS - Simple Ad Hoc Network Simulator

Statistics

# of clients:	8
# of links:	0
# of packets sent:	0

Edit Graph manually

Add Client

Remove Client

Edit Links

Start Clients

Load/Save Graph

New Graph

Load Graph

Save Graph

Save Graph As

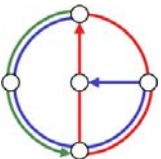
Miscellaneous

Settings

Hide Output Wnd

Exit

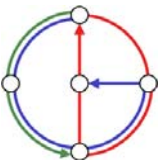
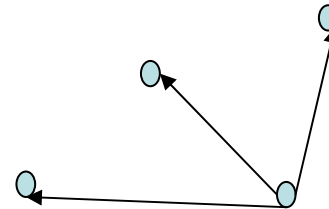
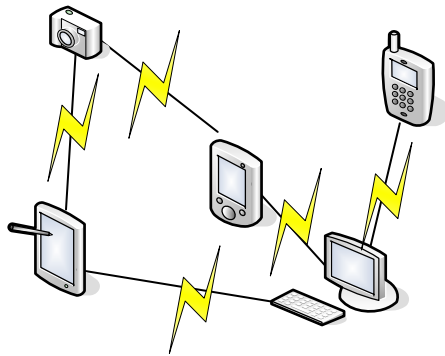
Sender



Programs running on nodes



- 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 - Simple Ad Hoc Network Simulator

Statistics

# of clients:	8
# of links:	0
# of packets sent:	0

Edit Graph manually

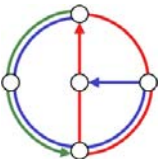
- Add Client
- Remove Client
- Edit Links
- Start Clients

Load/Save Graph

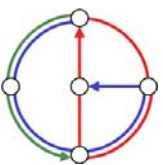
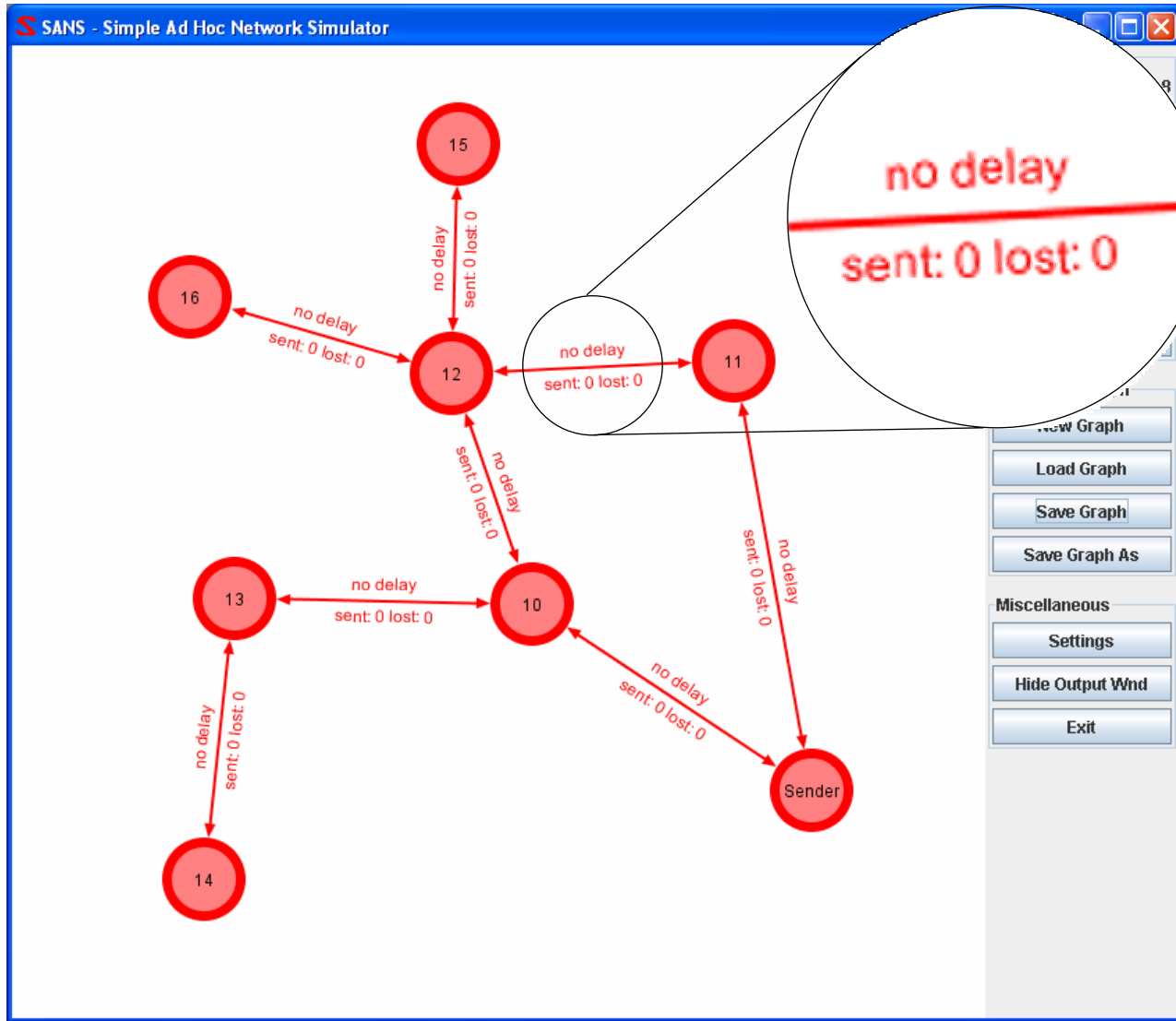
- New Graph
- Load Graph
- Save Graph
- Save Graph As

Miscellaneous

- Settings
- Hide Output Wnd
- Exit



SANS: Adding Edges



SANS: Link Properties



Edit link properties

Delay settings

- no delay
- constant delay ms
- delay according to distance between nodes. Use factor

Apply delay settings to all links

Error settings

- no error
- constant error probability %
- deterministic error pattern (use + for success and - for error)

Apply error settings to all links

Direction settings

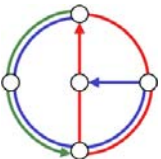
- 12 ==> 16
- 12 <== 16
- 12 <=> 16

OK **Cancel**

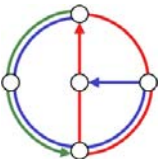
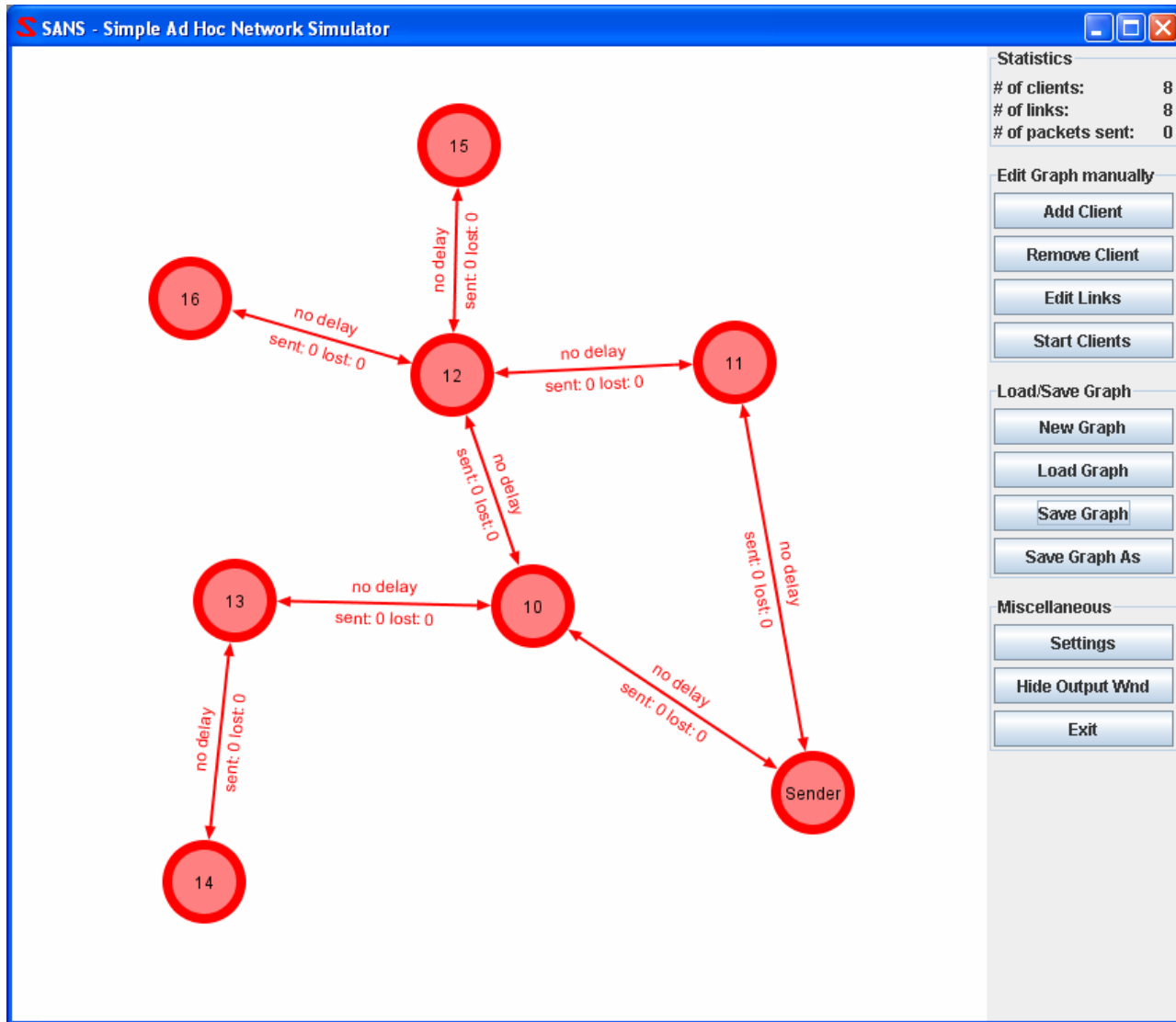
Transmission delay

Packet drop policy

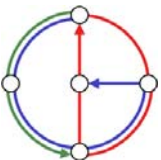
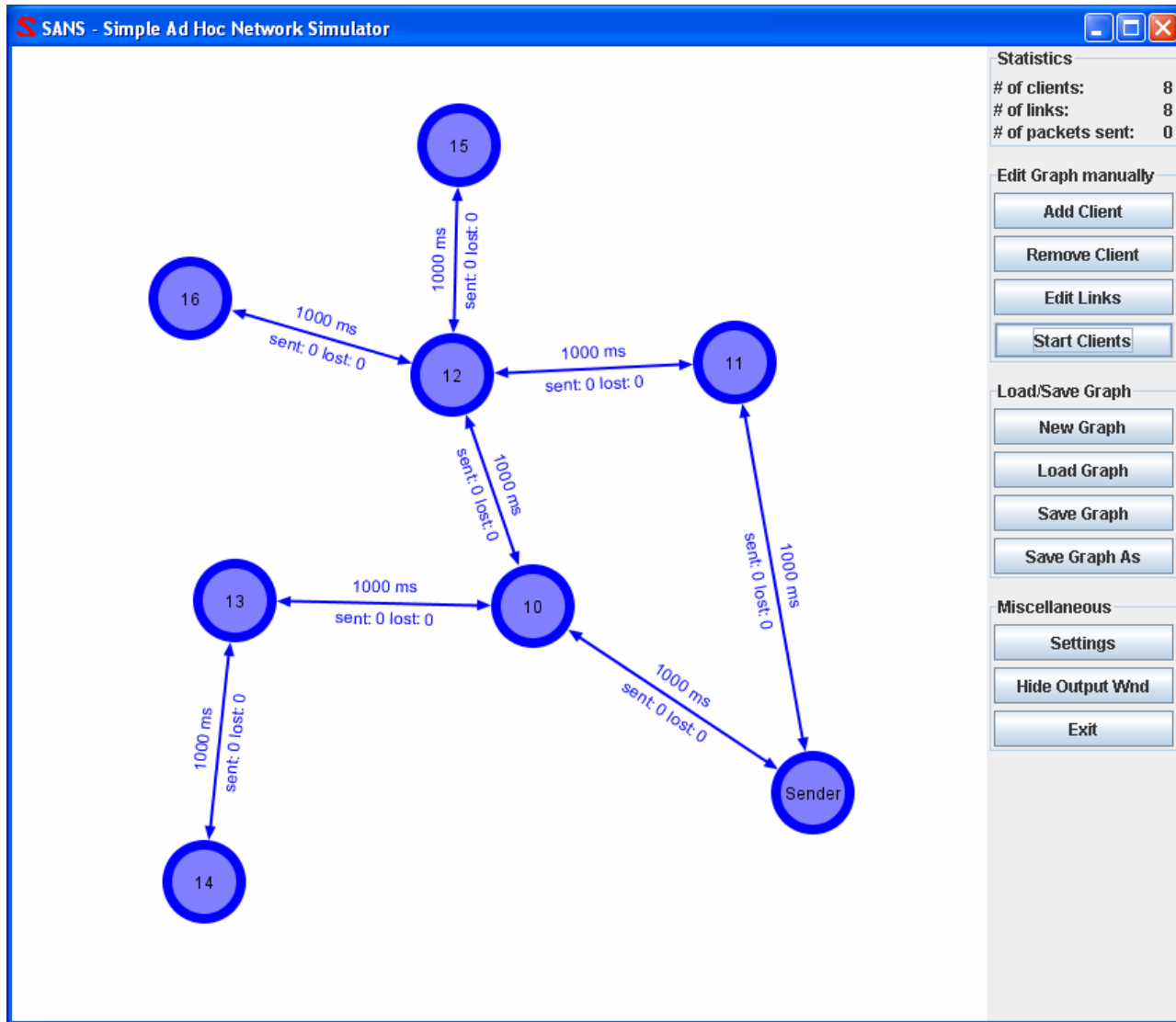
Link direction



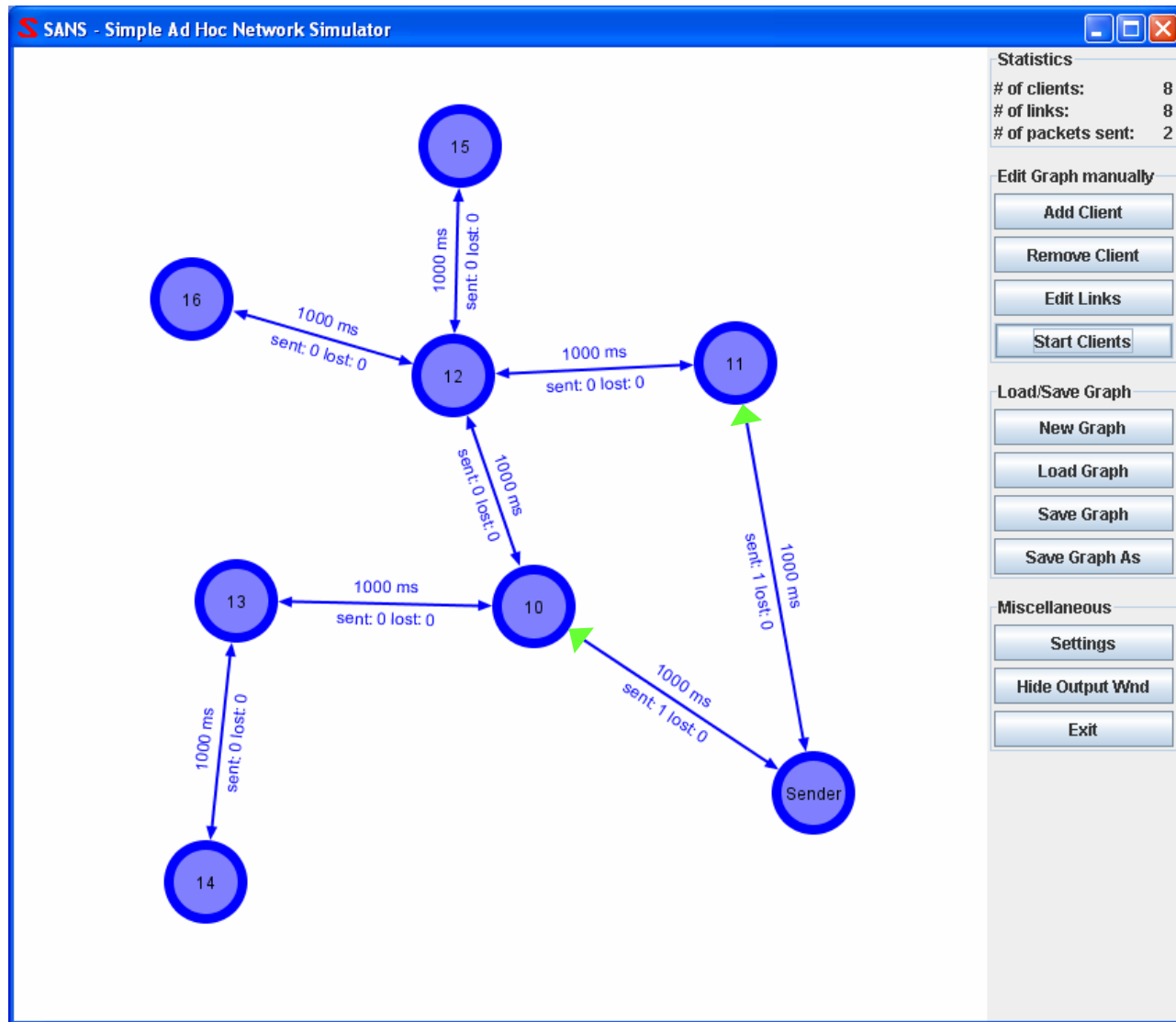
SANS: Flooding Example



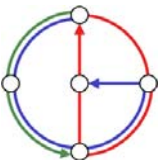
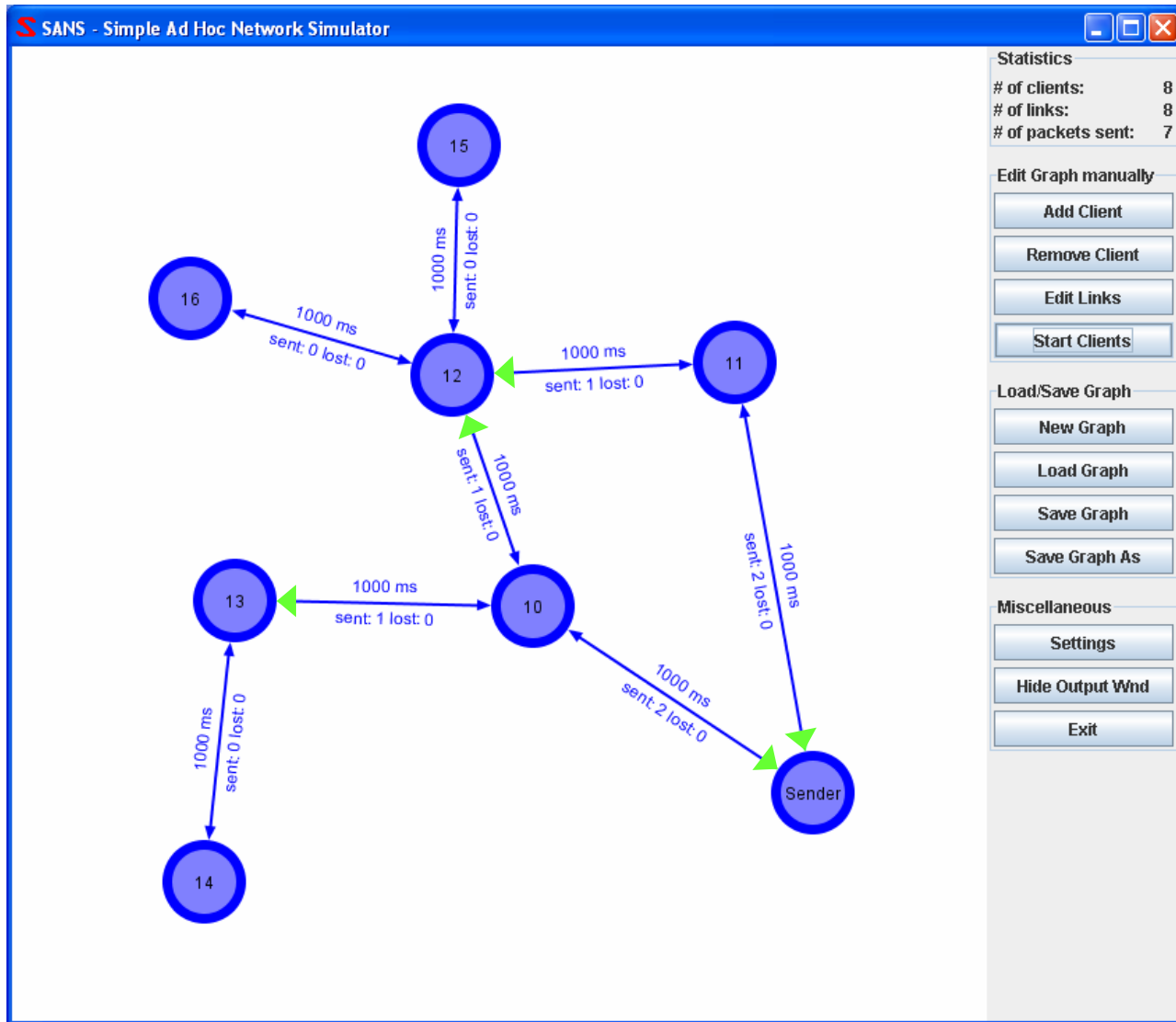
SANS: Flooding Example



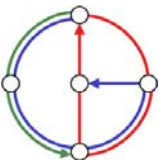
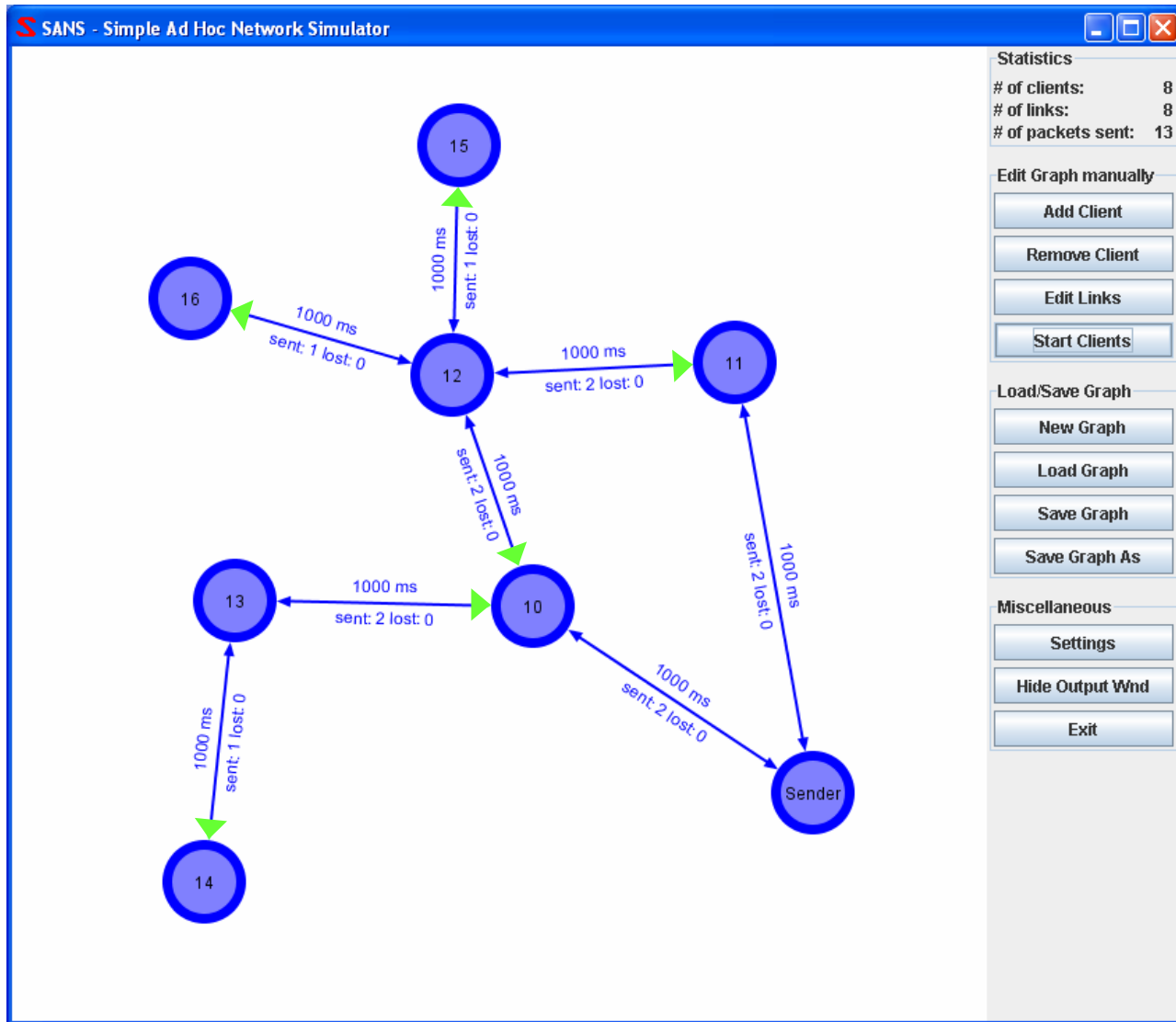
SANS: Flooding Example



SANS: Flooding Example



SANS: Flooding Example

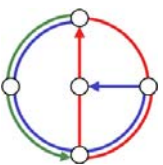


SANS: Console Output



```
Output Window
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
```

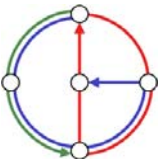
All Emulator 10 11 12 13 14 15 16 Sender



SANS in use



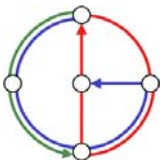
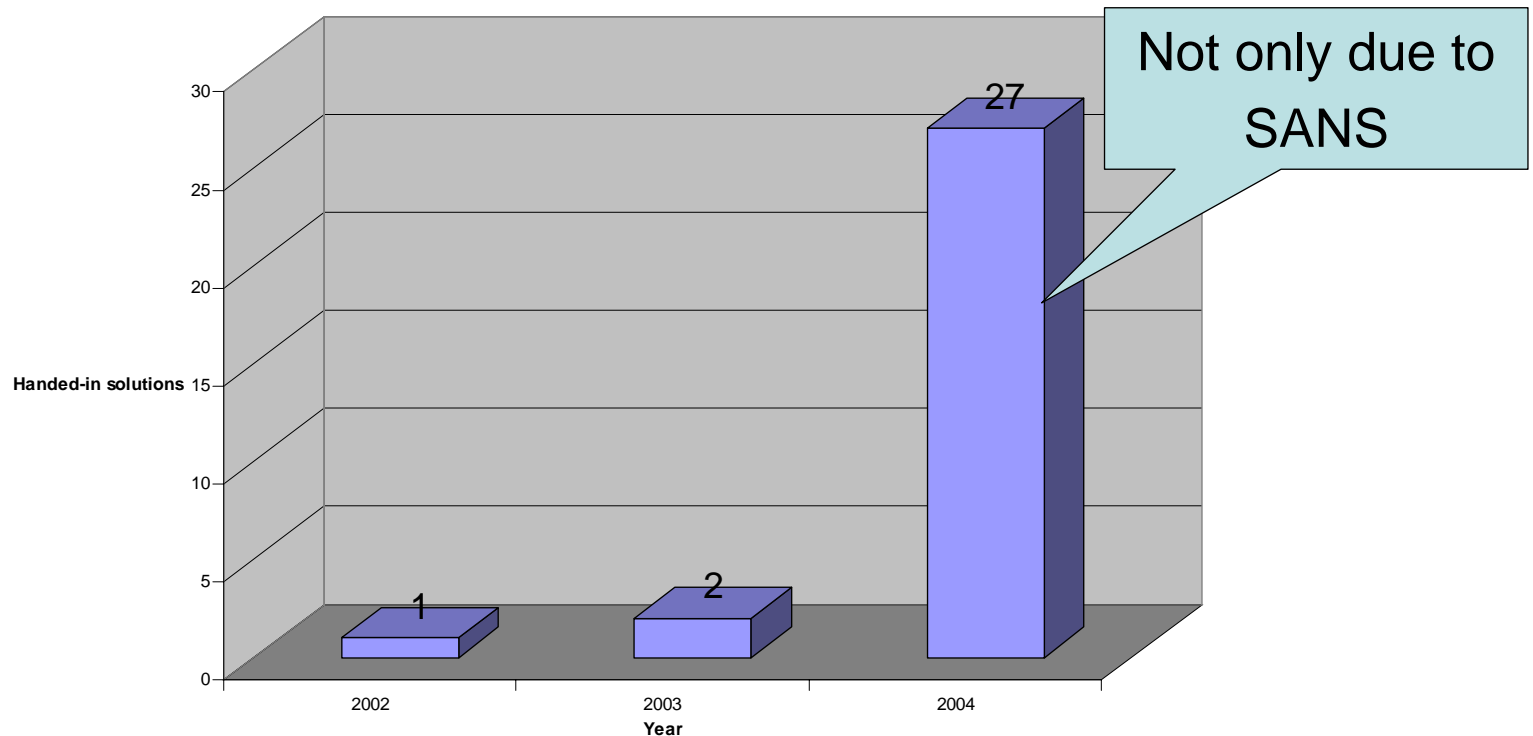
- “Mobile Computing” at ETH Zurich
- **Server-less instant messenger** as a long term homework
 - **Find** other **users** in the network
 - **Send** and **receive messages** to users not within direct communication range
 - **Relay messages** for other users
- The same task was assigned in three consecutive 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:



The End



Download SANS at

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

