

# An Interactive Web GUI for Virtual Backbone in Wireless Networks

## Purpose of Project

- ❑ To design an interactive web interface for modeling virtual backbones in wireless ad hoc networks
  - Allows user to choose network parameters
  - Draw network topology

## Background

- ❑ **Wireless ad hoc Networks**
  - Decentralized wireless network
  - Determination of who serves who dynamically configured based on network connectivity
  - Depend on broadcast and multicast traffic for network management
- ❑ **Broadcast Storm Problem**
  - Extreme amounts of broadcast control message traffic is considered a broadcast storm
  - Consumes network resources sufficient to render the network incapable of transmitting important data traffic
- ❑ **Virtual Backbone**
  - Infrastructure to reduce broadcasting storm problem
  - Limits the control messages within the virtual backbone

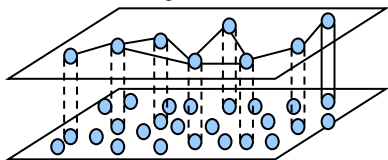
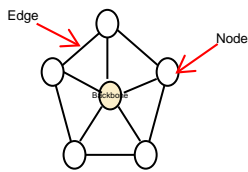


Illustration of Virtual Backbone

## Virtual Backbone Examples

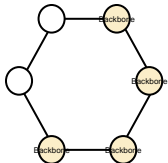
### Star Example

- ❑ Centralized nodes serving as backbone for many outer nodes



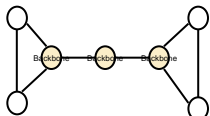
### Ring Example

- ❑ One or more consecutive backbone nodes in a network that forms a ring



### Bridge Example

- ❑ One or more consecutive backbone nodes connecting network clusters



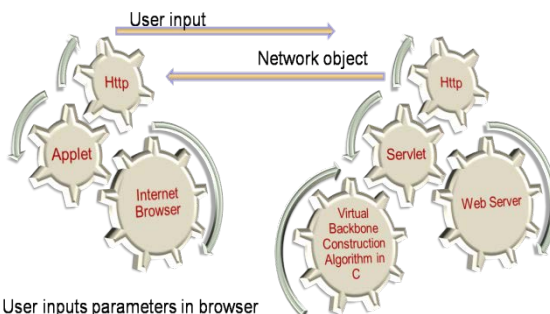
## Design

### Challenge

- ❑ Design an interactive web user interface to demonstrate a virtual backbone algorithm

### Solution

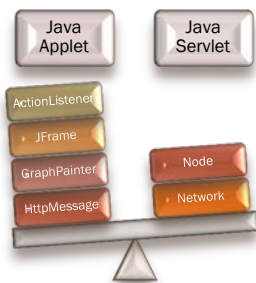
- ❑ Java Applet-Servlet technology to interact with algorithm written in C



1. User inputs parameters in browser
2. Applet creates an HttpMessage with user's input
3. HttpMessage sent to web server
4. Web server starts the Servlet
5. Servlet creates Network based on user input
6. Servlet writes Network object to a file and starts virtual backbone algorithm
7. Algorithm reads Network file, creates virtual backbone, writes Network to file
8. Servlet reads Network Object file and sends it back to Applet
9. Applet receives Network object and finds the edges
10. User defined network is displayed in browser

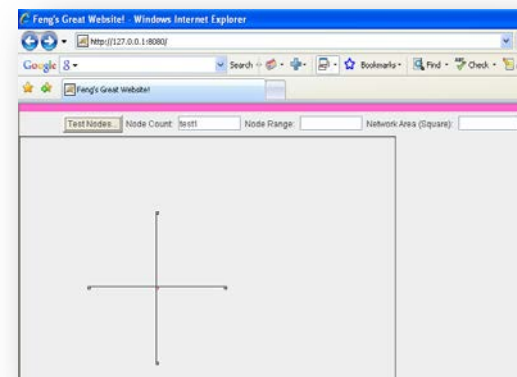
## Implementation

- ❑ Applet and Servlet have many other helper components:
  - Node and Network -describe objects that are displayed in the Applet
  - JFrame -handles the graphics that are displayed in the browser
  - ActionListener -listens for user requests such as button clicks
  - GraphPainter -draws the Node Network for display in the Applet
  - HttpMessage -formats the http communication to the Servlet



## Web Interface

- ❑ A visitor of the site will first be asked to enter three parameters:
  - Node Count – the number of Nodes to be displayed
  - Node Range – the wireless range, used to find edges
  - Network Area – length and width of the Network deployment area
- ❑ Upon entering the parameters and clicking the Test Nodes button with the mouse the Network will be built and viola! You have a Network
- ❑ All the work is transparent to the visitors:
  1. Passing the parameters to the Servlet
  2. Servlet creates the Network and sends it back to the Applet
  3. Applet finds the edges and sends Network to the GraphPainter
  4. GraphPainter displays the Network (before you can say "ad hoc!")



## Conclusions

- ❑ Java Applet-Servlet environment is suitable for demonstrating virtual backbone algorithm in ad hoc environment
- ❑ Applet-Servlet programming offers performance enhancements for users and web application developers
  - Part of code runs on server, part on user's computer
- ❑ Virtual Backbone algorithm was never fully integrated but Servlet is designed to mimic the result
- ❑ Future revisions to Servlet will integrate Virtual Backbone Algorithm
- ❑ Future projects may work on redesigning application in Visual C#, a more popular language with better support