Announcements

Midterm

- Evening of March 17th
- At 6 p.m. or 7:30 p.m. in TPL 203
- 75 minute closed book exam
- Review session 3/16; Time/room:TBD
- Sample midterm will be available online
Today’s Plan

- Ethernet slot time
- Overloading
- Recursive classes
ETHERNET TRANSMISSION ALGORITHM

Eager to send a packet

Packet becomes ready

Transmitting packet preamble

Receiver becomes idle

Waiting for backoff slots

Transmitting jamming signal

Jamming complete

Transmitting packet contents

Preamble complete

Receiver not idle

Receiver becomes busy

DONE

CARRIER SENSE MULTIPLE ACCESS WITH COLLISION DETECTION
Binary Exponential Backoff

After detecting a collision:

- Pick a random number between 0 and $2^{\text{attempts}} - 1$ slots
- Wait that many “time units” and then try again.
AMAZING ANIMATED DEMONSTRATION!!!
Overload
public class SkeletalBrowser extends GUIManager {
    ...

    // The complete list of web site addresses that have been entered
    private HistoryList history = new HistoryList();

    // Display the elements of the program's user interface
    public SkeletalBrowser() {
        // Create window to hold all the components
        this.createWindow( WINDOW_WIDTH, WINDOW_HEIGHT );
        ...
    }

    // When the user completes the entry of a new address, add it to the
    // history list if it isn't already there
    public void textEntered() {
        if ( ! history.contains( AddressEntry.getText() ) ) {
            history = new HistoryList( AddressEntry.getText(), history );
        }
        matches.setText( history.toString() );
        ...
    }
}
public class HistoryList {

    private boolean empty = false;     // true if nothing in list
    private String firstWebsite;       // The first web site in the list
    private HistoryList restOfWebSites;// The rest of the list of web sites

    // Create an empty list
    public HistoryList( ) {
        empty = true;
    }

    // Create a larger list from a new website and an existing list
    public HistoryList( String newSite, HistoryList existingList ) {
        firstWebsite = newSite;
        restOfWebSites = existingList;
    }

    // Produces a single String containing all the entries in the list
    // separated by new lines.
    public String toString() {
        if ( empty ) {
            return "";
        } else {
            return firstWebsite + "\n" + restOfWebSites.toString();
        }
    }
}
public class SkeletalBrowser extends GUIManager {
    ...

    // The complete list of web site addresses that have been entered
    private HistoryList history = new HistoryList();

    // Display the elements of the program's user interface
    public SkeletalBrowser() {
        // Create window to hold all the components
        this.createWindow( WINDOW_WIDTH, WINDOW_HEIGHT );
        ...
    }

    // When the user completes the entry of a new address, add it to the
    // history list if it isn't already there
    public void textEntered() {
        if (! history.contains( AddressEntry.getText() ) ) {
            history = new HistoryList( AddressEntry.getText(), history );
        }
        matches.setText( history.toString() );
        ...
    }
}
private boolean empty = false;     // true if nothing in list
private String firstWebsite;       // The first web site in the list
private HistoryList restOfWebSites; // The rest of the list of web sites

// Create an empty list
public HistoryList( ) {
    empty = true;
}

// Create a larger list from a new website and an existing list
public HistoryList( String newSite, HistoryList existingList ) {
    firstWebsite = newSite;
    restOfWebSites = existingList;
}

// determines whether the collection contains a given entry
public boolean contains( String site ) {
    if ( empty ) {
        return false;
    } else if ( firstWebsite.equals( site ) ) {
        return true;
    } else {
        return restOfWebSites.contains( site );
    }
}