6.111 Project Checklist
Kiarash Adl
Nandi Bugg
Sachin Shinde

General System Function: Provide standard telephony features to users on a telecommunications network.

UI Module
Function: Managing inputs and outputs from the User (through buttons, switches, and the LCD character display) and the AC '97 Audio Codec, and submitting service requests and receiving service responses from the Application Module.

Checklist:
- Scrolling ASCII text on the LCD character display.
- Context-Dependent Menus that allow the following actions:
    - Dial a call
    - End a call
    - Accept a call
    - Reject a call
    - Set headphone volume
    - Playback voicemail messages
    - Erase voicemail messages
    - Set numbers for Call Blocking
    - Set number and mode for Call Forwarding
    - Transfer a Call
    - Place a Call on Hold
    - Initiate Call Waiting
    - Initiate Conference Call
    - Join Conference Call
- If time permits, these menus will also allow for the following actions:
    - Set a personalized voicemail greeting.
    - Assign names for numbers in an address book.
- Context-Dependent Shortcut Buttons for common actions, including but not limited to the following:
    - Dial a call
    - End a call
    - Accept a call
    - Reject a call
    - Set headphone volume
- If time permits, we shall also allocate shortcut buttons and/or switches for speed dial.

Method of Demonstration:
- Successful real-time operation for the entire system

Application Module
Function: Provides common network services for the User Interface module above it.

Checklist:
- Properly handle Service Requests including but not limited to the following:
    - Initialize the Network
    - Dial a call
    - End a call
    - Accept a call
    - Reject a call
    - Place a call on hold
    - Forward an incoming call
    - Transfer a call
    - Switch to an incoming call while busy (Call Waiting)
    - Call another party while busy (Conference Call)

Method of Demonstration:
- Successful real-time operation for the entire system.
- Testbench for the individual module.

Presentation Module
Function: This module converts data sent from the Application module into a format suitable for the network and sends it to the Session module (and vice-versa). In particular, if time permits, it implements audio compression and decompression through the Internet Low Bitrate Codec (iLBC).

Checklist:
- As a baseline, the Presentation module simply passes audio data unaltered from the Application Layer to the Session Layer and vice-versa.
- If time permits, we will implement the iLBC for compression and decompression
- Compression:
    - Properly convert 160 samples of 8 kHz, 16-bit audio (20 ms) to a 304-bit frame of compressed data, with bitrate of 15.2 kbps.
- Decompression
    - Properly convert a 304-bit frame to 160 samples of 8kHz, 16-bit audio (20 ms). If the packet did not arrive on time, the decompressor uses Packet Loss Concealment (PLC) to generate the 160 samples.
    - We may achieve high throughput on the decompressor by pipelining and utilizing many multipliers. This will help for conference calls, where many 304-bit frames must be decompressed per 20 ms.

Method of Demonstration:
- Successful real-time operation for the entire system.

- Testbench for the individual module. If time permits the iLBC to be implemented, compression testing will use a sample audio waveform and compare results to precalculated 304-bit frames. Decompression testing is vice-versa.

Session Module

Function: This module provides virtual communication sessions for the Presentation module above it. A session may be thought of as a correspondence of messages between two nodes in the network.

Checklist:
- Start a session.
- End a session.
- Synchronize messages in a session.
- If needed, provide synchronization checkpoints.
- Manage disorderly termination (aborts).
- Use multiple unicast sessions for call conferencing.
- If time permits, use multicast sessions instead of unicast sessions.

Method of Demonstration:
- Successful real-time operation for the entire system.
- Testbench for the individual module.

Transport Module

Function: This module provides a virtual link for end-to-end transmission of error-free messages for the Session module above it.

Checklist:
- Packetize messages from the Session layer into packets.
- Assemble packets from the Network layer into messages.
- Guarantee reliability in the event the network becomes disconnected.
- If time permits, provide end-to-end flow control

Method of Demonstration:
- Successful real-time operation for the entire system.
- Testbench for the individual module.

Network Module

Function: This module provides an end-to-end virtual link for reliable transmission of error-free packets (in order) for the Transport module above it.

Checklist:
- Perform virtual circuit routing using a shortest-path algorithm.
- If time permits, use virtual circuit tree routing for multicast sessions.

- If time permits, use optimal routing instead of shortest-path routing.
- If time permits, manage point-to-point flow control.
- Manage network initialization.
- If needed, perform error-detection (in addition to that performed in the DLC module).

Method of Demonstration:
- Successful real-time operation for the entire system.
- Testbench for the individual module.

Data Link Control (DLC) Module
Function: This module provides a point-to-point virtual link for reliable transmission of error-free packets (in order) for the Network module above it.

Checklist:
- Media Access Control (MAC)
  - Control access to the multipoint link to avoid collisions with CSMA/CD.
- Logical Link Control (LLC)
  - Provide reliability with:
    - error-detection (CRC), and
    - error-control (Selective Repeat ARQ)
  - Manage multipoint link initialization.

Method of Demonstration:
- Successful real-time operation for the entire system.
- Testbench for the individual module.

Physical Interface (PHY) Module
Function: This module provides a virtual bit pipe to the DLC module above it, not immune to errors or collisions.

Checklist:
- Perform 8b/10b encoding/decoding.
- Perform clock recovery on incoming packets.
- When not sending, determine if the bus is idle.
- If the bus is not idle, attempt to recover incoming packet.
- When sending, determine if a collision occurs.

Method of Demonstration:
- Successful real-time operation for the entire system.
- Successful real-time operation for the individual module.
- Testbench for the individual module.