

Web Based Interfaces for Digital 2- Way Radio

NW Digital Radio Corporation

Our Market

- Amateur (Ham) Radio Operators
 - Emergency Communications Support
 - Digital Data (Messaging), Digital Voice
 - Ad-Hoc Networking
 - Situational Awareness
 - Geolocation
 - Experimentation
 - New Modulation Techniques
 - New Protocols
 -

Radio Design

- 30 Watts UHF (430-450 Mhz.)
- Embedded ARM Based Computing Platform
- I/Q 2-point Modulation and Detection
- Ethernet
- 4 Host USB Ports
- Software Defined Modems (Socket Interface)
- Software Defined Protocols (Socket Interface)
- Embedded Applications on Linux OS (Debian)

Configuration and Control

- No Physical Controls
- All Control is Via Network (Ethernet, WiFi, etc.)
 - Frequency
 - Power
 - Protocol
 - Modulation

Monitor and Communicate

- Monitor
 - Environment
 - Performance (RSSI and BER)
- Voice and Data I/O
 - Streamed
 - Packet Switched

The Software Challenge

- Sockets to Hardware for Control and Monitoring
- Sockets to Multi-Stage Pipeline
 - Voice and/or Data
 - Protocol
 - Modem
 - RF
 - Configuration and Transport
- No / Low Jitter Streaming for Digital Voice

The Software Challenge

- Target Rich Clients
 - Multiple Operating Systems / Environments
 - Smart Phones, Tablets, Desktops, Purpose Built
 - I18n / Support for Disabilities
- Inter-Application Data Streams
 - On Processor
 - Over Internet
 - No UI Endpoint

Websockets

- Started by learning about Websockets
 - Full interaction between modern browsers and web application
 - Useable with non-web GUI elements
 - Implemented some crude applications
 - Required intimate knowledge
 - Gave some of the functionality needed

node.js to the Rescue

- Discovered node.js as a framework Fall 2012
- Provided a container and the use of JavaScript
 - Server
 - Client
- Introduction to programmers
- Much quicker implementations with reusable design patterns

“Packet Capture”

- Existing network based programs use well known or configurable ports
- Use “pcap” module
 - Leverage pcap filters from tcpdump
- Capture packets (UDP or TCP)
 - Protocol handlers (D-STAR / APRS)
 - Events update model in app
 - Push model state as JSON to “subscribers”

Endpoint Socket Communications

- Open sockets to applications and modules
 - Use “dgram” module
 - Communication
 - Exchange streams
 - Data and Configuration
- Authentication
 - Use “crypto” module
- Monitor and Control Radio

Web Interfaces

- Use socket.io to abstract transport
 - Pass JSON objects bidirectionally
 - Replicate state model on server and web
 - Update via JSON objects
- Use http or express to manage presentation path
 - Light HTML with JavaScript DOM modification
 - Do not use JADE or other layout languages
- Manage sessions for multiple users

Shared Model

- Multiple clients in application(s)
 - Endpoint sockets
 - Packet capture
- Server for Web/socket.io clients
- Data is passed between clients and server using shared objects within the application.
 - Queues
 - Events

Questions