# A Configuration Protocol for Embedded Devices on Secure Wireless Networks

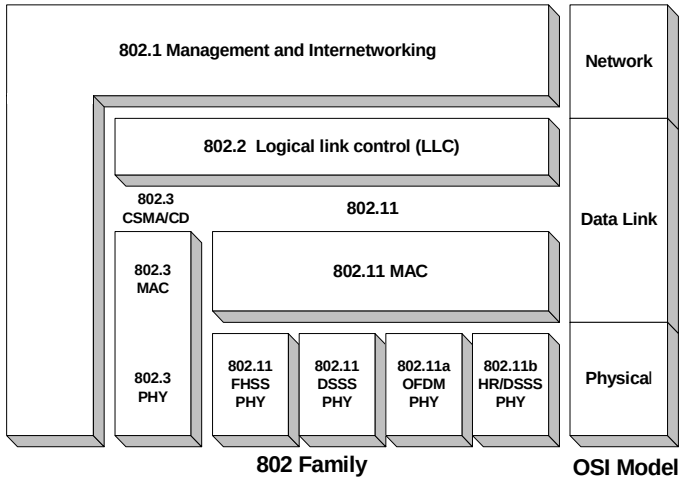
Larry Sanders
lsanders@ittc.ku.edu

16 May 2003

# Introduction

- Wi-Fi Alliance
  - Formally Wireless Ethernet Compatibility Alliance (WECA)
  - Formed to certify interoperability of Wireless LANs products based on IEEE 802.11 specification
  - Coined the term Wireless Fidelity (Wi-Fi)

- What is Wi-Fi
  - WLAN
  - Wireless Ethernet
  - 802.11a, 802.11b, 802.11g



# Motivation

- Stations on a Wi-Fi Network
  - Unable to utilize traditional configuration protocols such as DHCP until the host has link level connectivity
  - Requires user to enter extra parameters
    - Service Set Identifier (SSID or Network Name)
    - WEP encryption keys
  - Embedded wireless devices
    - Limited input capabilities
    - Additional interfaces costly

# Background

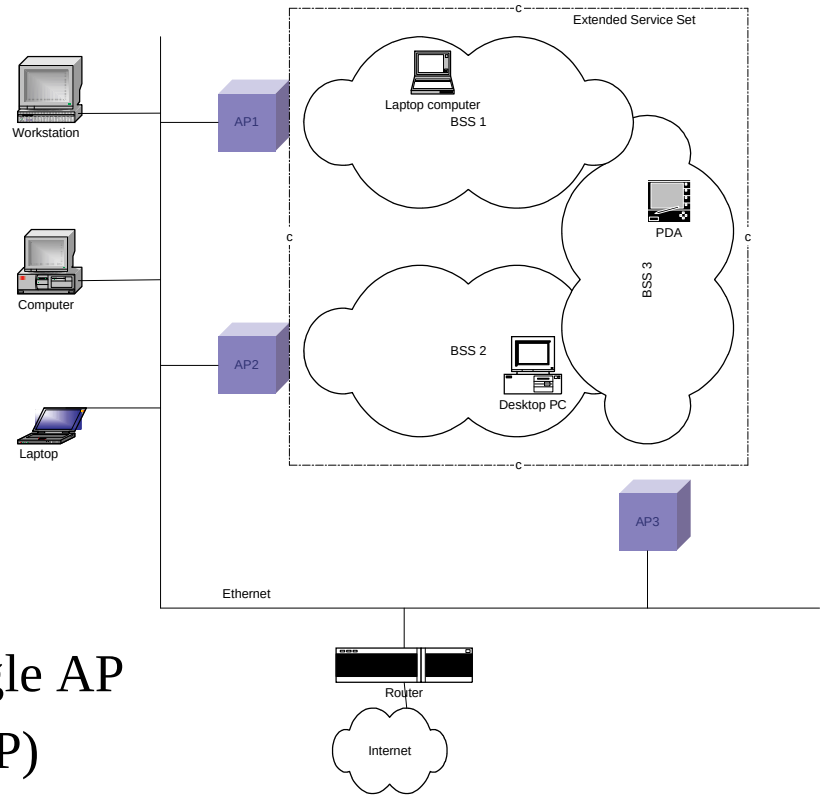
**Types of Wi-Fi Networks**

- Basic Service Set (BSS)
  - Group of Wi-Fi stations
- Independent networks (IBBS)
  - Sometimes called Ad-Hoc Networks
  - Stations communicate directly with each other
- Infrastructure networks (BSS)
  - Access Point (AP) used for all communications
  - Stations need only be within range of the AP
  - APs can assist stations with power management by buffering
  - Bridge to Ethernet network

# Background

## Extended Service Set

- Created by linking BSSs
- Backbone network
  - Distribution System
  - Typically Ethernet
- Management
  - Wi-Fi stations associate with single AP
  - Inter-Access Point Protocol (IAPP)



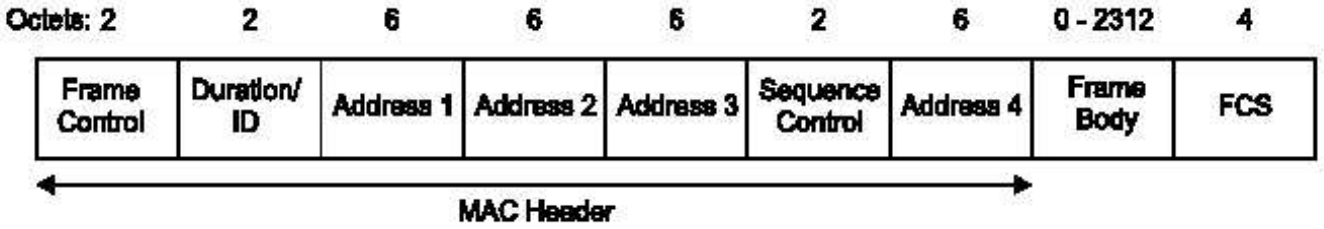
# Background

**Joining a Wi-Fi Network**

- Scanning
  - Find the network
  - Passive or Active
- Authentication
  - Open-System Authentication
    - Null Authentication – always successful
  - Shared-Key Authentication
    - Utilizes WEP
      - Challenge / response
- Association

# Background

## 802.11 MAC Frame




- Control
  - Type of frame, power management, WEP status, etc.
- Duration/ID
  - Use depends on type of frame
- Sequence Control
  - Fragmentation and discarding duplicate frames
- Frame Check Sequence
  - Error detection across entire frame (including 802.11 header)

# Background

## 802.11 MAC Frame (continued)

- Frame body
  - 802.11 Management frames
  - 802.11 Control frames
  - 802.11 Data frames
    - 802.2 Logical-Link Control (LLC) encapsulated data
- Address Fields (6 octets each)

| Function | Address 1 | Address 2 | Address 3 | Address 4 |
|---|---|---|---|---|
| | | | | |
| WDS | RA | TA | DA | SA |
| Independent BSS | DA | SA | BSSID | not used |
| To AP (infrastructure) | BSSID | SA | DA | not used |
| From AP (infrastructure) | DA | BSSID | SA | not used |

# Background

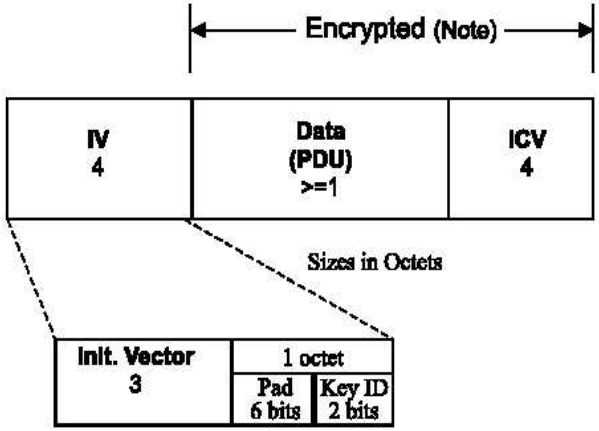
## 802.11 Wired Equivalent Privacy (WEP)

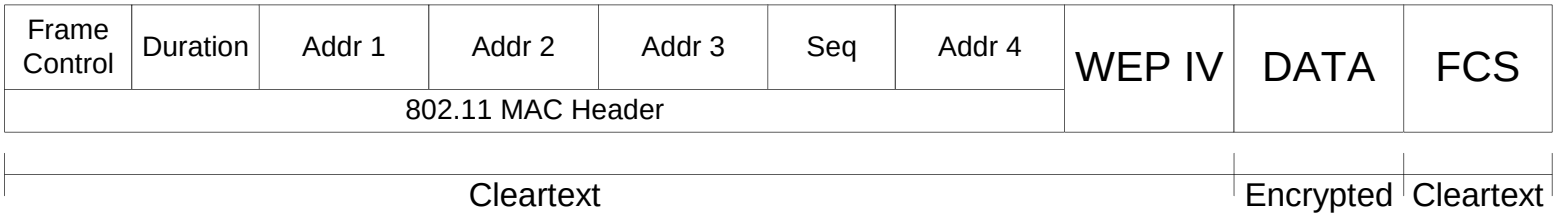
- Goal
  - Provide security similar to Ethernet
- Shared Keys
  - 40/64 and 104/128 bit standard key sizes
    - Concatenated with 24 bit Initialization Vector
- Utilizes RC4
  - Symmetric stream cipher
  - RSA Security, Inc.

# Background

## WEP Frame Format

- IV (4 octets)
  - Initialization Vector (3 octets)
    - Concatenated with WEP key
    - Typically implemented as counter
  - Key ID (1 octet)
    - 2 bit field that specifies which WEP key
- ICV (4 octets)
    - Integrity Check Vector (CRC-32)



# Background

## Problems with WEP

- Key Management
  - Must be distributed to all stations
- Packets can be spoofed and/or modified
  - No Integrity protection for 802.11 header
- 802.11 Authentication
  - No mutual authentication
  - Trivial to defeat station authentication
- Fluhrer, Mantin, Shamir (FMS attack)
  - Weak IVs
  - Assumes first byte of key stream can be recovered (0xAA SNAP header)
  - Only requires a few million packets to crack WEP

# Background

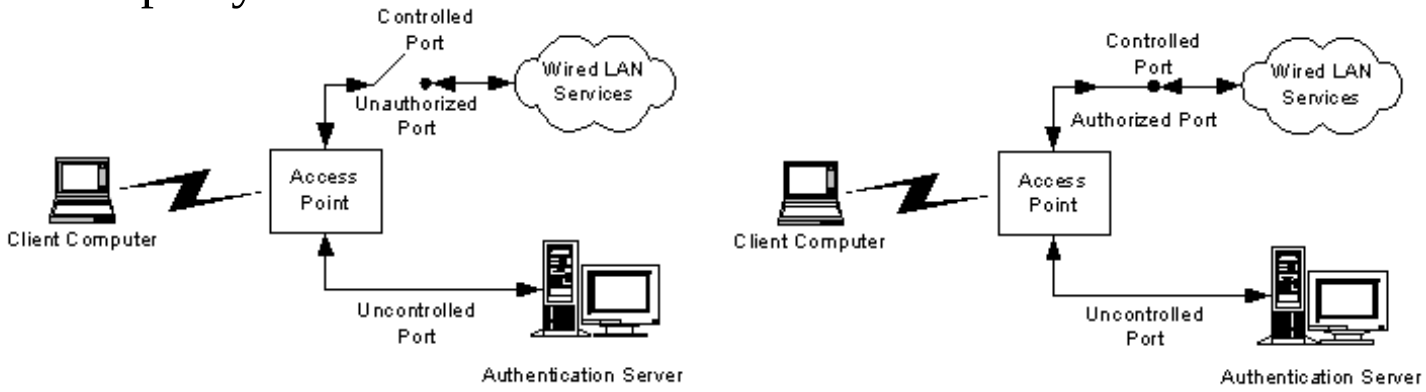
**802.11 Task Group i**

- Charged with increasing 802.11 security
- Draft due September 2003
- Two new protocols
  - Temporal Key Integrity Protocol (TKIP)
    - Short-term solution
    - Works with legacy hardware via software/firmware updates
  - Counter-Mode-CBC-MAC (CCMP)
    - Long-term solution for future hardware
    - Uses the Advanced Encryption System (AES)

# Background

**802.1x: Port based Network Access Control**

- Based on IETF's Extensible Authentication Protocol
- Facilitates mutual authentication
- Method to distribute encryption keys
  - Session keys
  - Group keys



# Background

**802.1x in the Home/Small Business**

- Generally, no Authentication Server on home networks

- Use of *pre-shared* keys
  - Wi-Fi Alliance endorsed vendor solution
  - Similar to WEP
    - Has to be distributed manually
    - Becomes base for session keys

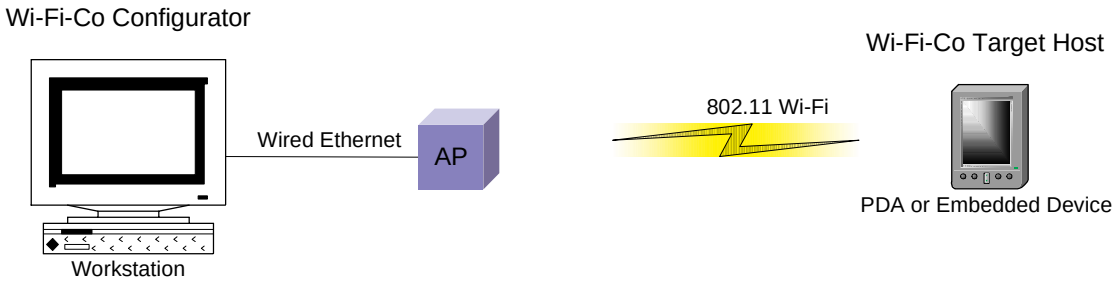
# Background

## Vendor Security Enhancements

- Non-broadcast SSID
  - SSID field in beacon packets zeroed
  - Not very effective
    - Attacks designed to force stations to re-associate
      - Exposes SSID
- MAC address filtering
  - Authorized list of MAC address
  - Somewhat effective
    - Most NICs allow users to set MAC

# Architecture

## Wi-Fi-Co

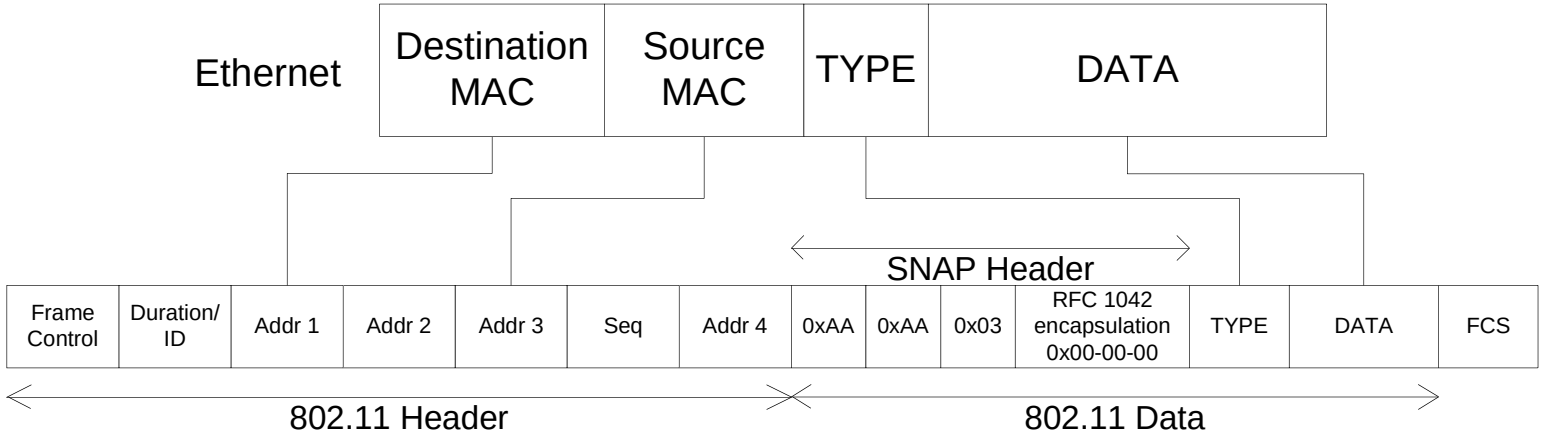



- *Configurator*
  - Host Sending Wi-Fi configuration parameters
  - Can be anywhere on the ESS network
- *Target*
  - Host receiving Wi-Fi configuration
  - Embedded wireless device, typically

# Architecture

## Wi-Fi-Co (continued)

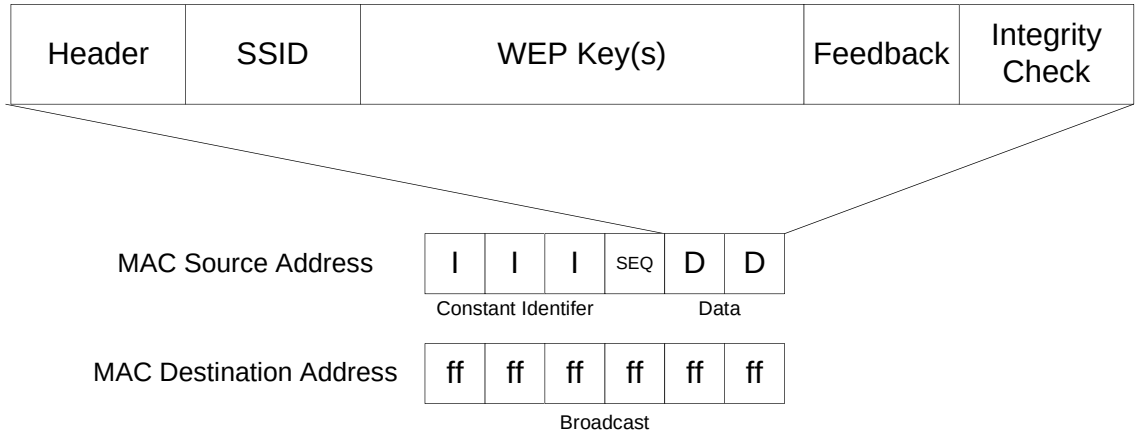
- General idea
  - 802.11 headers are unencrypted
  - Access Points copy MAC address during the bridging process
  - Data portion encrypted - no use to a station without keys
  - Source address - 6 octets of data
  - Broadcast



# Architecture

## Source MAC address

- Protocol identifier (3 octets)
  - 10:00:00 – Private Ethernet MAC pool
- Sequence (1 octet)
- Fragmented *Configuration Data* (2 octets)



# Architecture

## Feedback

- Positive acknowledgement
  - Optional
  - Once target device is configured and has IP level connectivity
  - TCP connection back to Configurator
    - IP level address assigned
    - Statistics (for development)
  - Configurator must send its address
    - Configurator is modifying MAC addresses

# Architecture

**Protecting the WEP keys**

- Broadcast packets easily intercepted
  - On the wired Ethernet network portion
  - Any Wi-Fi station within range of an Access Point in the ESS
- Utilizes RC4
  - Shared key symmetric cipher
  - Embedded devices ship with pre-programmed key
    - Certificate with product code
  - Additional input required on the *Configuration* host
    - Much easier then input to embedded device

# Implementation

## Header

- Default key number
- SSID length
- Feedback
- WEP key lengths
- Version
- Encryption
  - Designates that the SSID and WEP fields are encrypted
- Mode/IBSS channel
  - 0 for Infrastructure (BSS or ESS)
  - Non-zero for IBSS
    - Specifies the channel the IBBS is on
- Buffer Length is 8 + 4 * KEY_LEN + SSID_LEN + (FB * 6)


# Implementation

## Wi-Fi Channel Hopping

- 802.11b
  - Direct Sequence Spread Spectrum
  - 14 Channels (11 in U.S.), 5 MHz wide
  - Energy leakage – 5 channels
- Access Point setting
  - Each BSS operates on a specific channel
  - Overlapping BSSs – 5 channels apart to avoid interference
  - Hopping all channels guarantees traffic has a chance to be received
  - Sequence generally 1,6,11,2,7,…

# Results

## Test Setup #2 (Small Network)

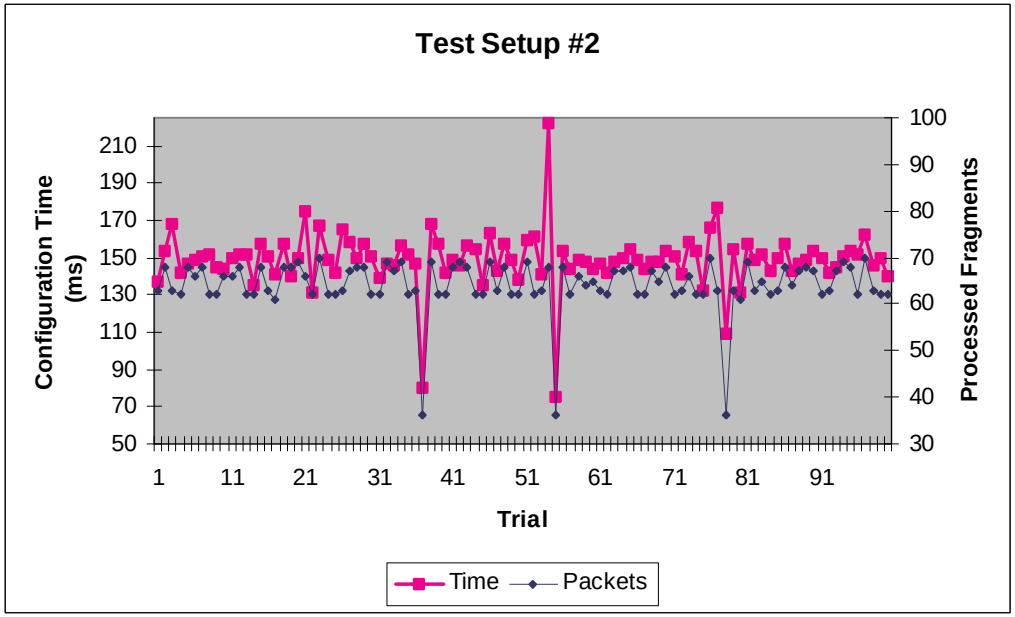
- Wired Ethernet *Configurator* → Wi-Fi *Target* station
- *Target* was Linux laptop
- Single, constant Wi-Fi channel

**Avg. Conf Time**

**148 ms**

**Avg. CRC Failure**

**0.0**



# Results

## Test Setup #4 (Small Network)

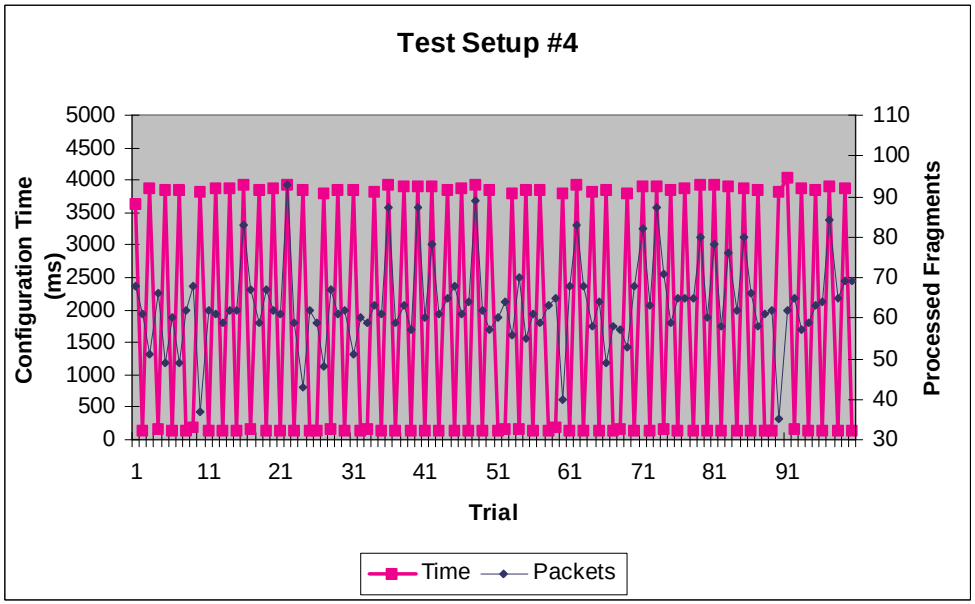
- Wired Ethernet ⟶ Wi-Fi *Target* station
- *Target* Linux laptop
- *Target* hopping three channels/second

**Avg. Conf. Time**

**1891 ms**

**Avg. CRC Failures**

**0.0**



# Results

## Test Setup #7 (ITTC's Network)

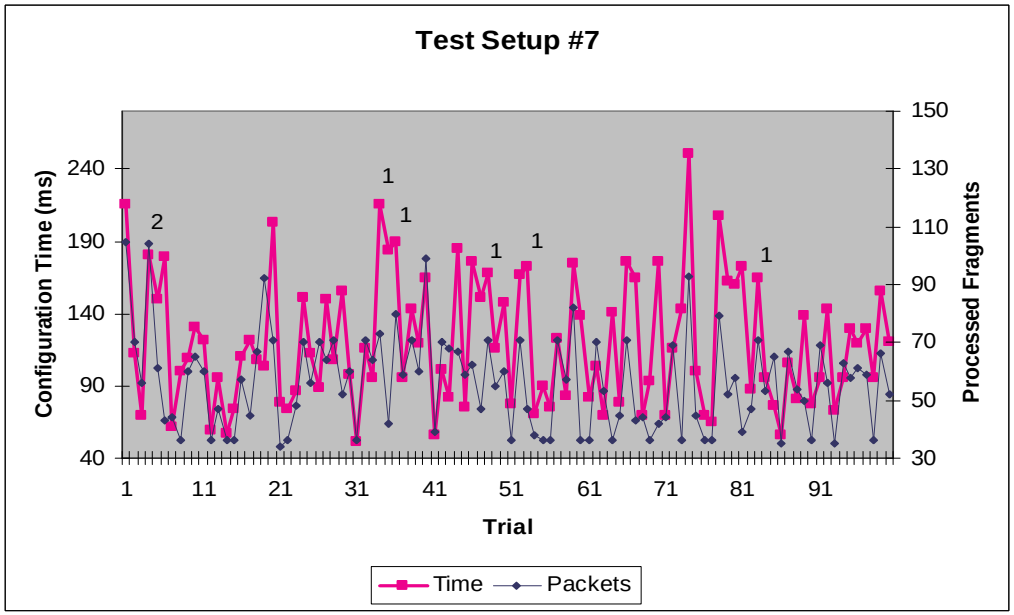
- Wired Ethernet → Wi-Fi *Target* station
- *Target* was Linux laptop
- Single, constant Wi-Fi channel

**Avg. Conf. Time**

**119 ms**

**Avg. CRC Failures**

**.07**



# Results

## Test Setup #9 (ITTC's Network)

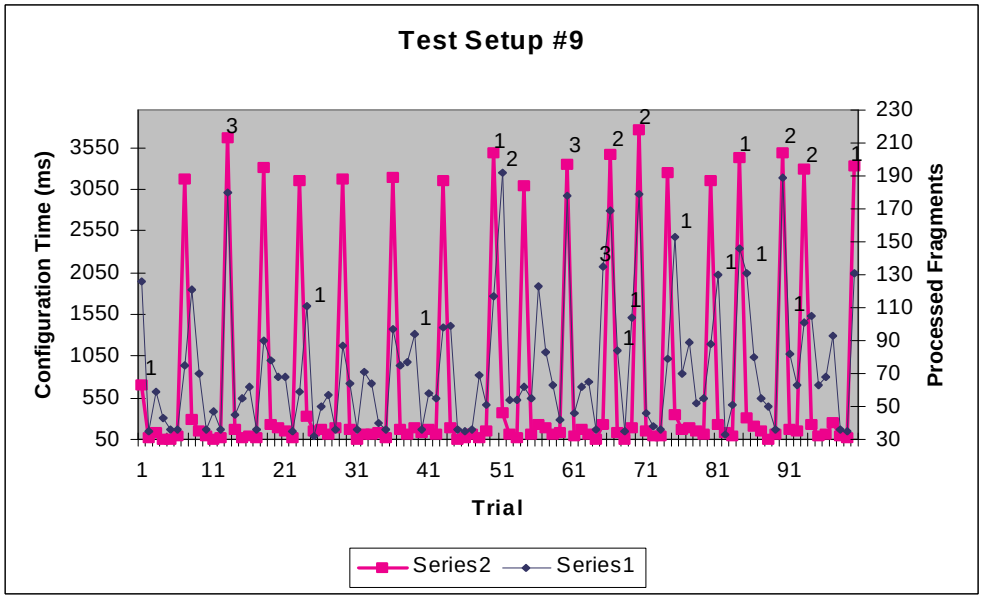
- Wired Ethernet → Wi-Fi *Target* station
- *Target* was Linux laptop
- *Target* hopping three channels/second

**Avg. Conf. Time**

**718 ms**

**Avg. CRC Failures**

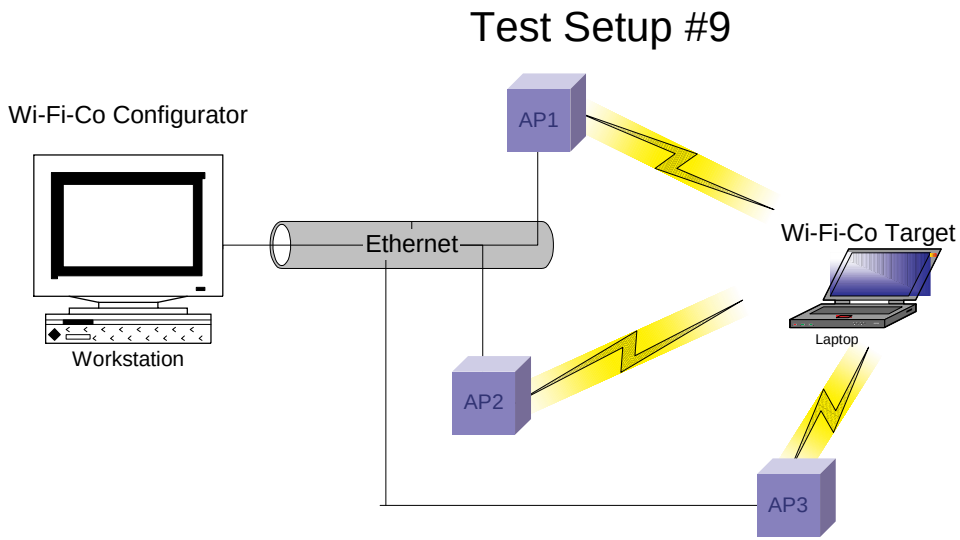
**.31**



# Results

## Test Setup #9 (Continued)

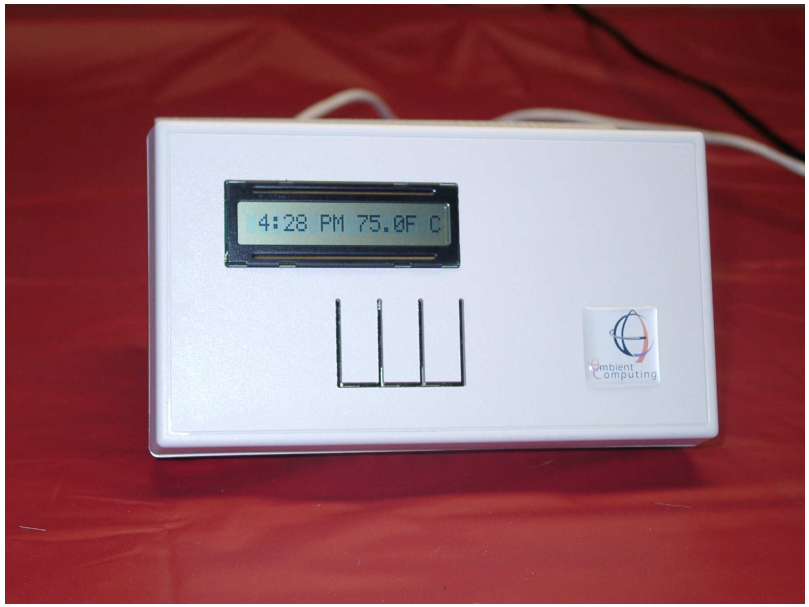
*Target Within Range of Three Wi-Fi Access Points*



# Results

## Embedded Device Tests

- Wi-Fi enabled Smart Wireless Thermostat
  - Designed by Ambient Computing, Inc.
  - Rabbit 2000 8-bit microcontroller
  - 30 MHz clock
  - USB Prism 2.5 Wi-Fi card
  - Rapid prototype
  - Experimental Wi-Fi drivers
  - Poor network performance



# Results

## Test Setup #12 (ITTC's Network)

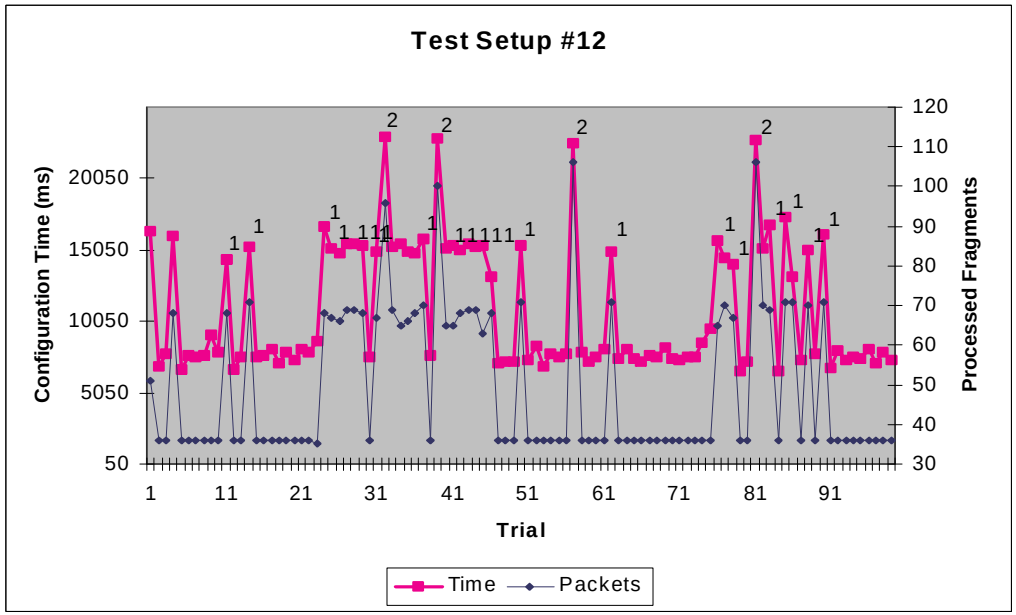
- Wired Ethernet → Wi-Fi enabled embedded device
- *Target* was Smart Wireless Thermostat
- *Target* Hopping 1 channel every 5 seconds

**Avg. Conf. Time**

**10.8 *Seconds***

**Avg. CRC Failures**

***.29***



# Conclusions

- Successfully tested on several networks
  - Small home network
  - Enterprise network at ITTC
- *Target* software ported to
  - Linux (tested on Debian, Gentoo, and RedHat)
  - Embedix Embedded Linux (Sharp Zaurus)
  - Dynamic C for the Rabbit Microprocessor (Smart Wireless Thermostat)

# Conclusions

- Challenges overcome
  - Ease of porting to many different platforms, operating systems, and distributions
- Lessons learned
  - Network “safe” implementation
    - No limit on initial rate *Configurator* sent packets
    - Fine for 100 Mbps network
    - Nearly saturated low speed, long haul Wi-Fi link to remote lab

# Demo

**Wi-Fi Toaster 9000**

Congratulations, you've have purchased the most advanced toaster on the planet! In moments you'll be logged on to best tasting toast you've ever pinged. Be sure to check out our other fine products such as the ***Wi-Fi Can Opener 1000.***

***Please Keep this certificate in a safe place***

**Serial Number: 0129831321332**

**Product Code: 90FA-C387-8712-7AC9**

**Limited Warranty**

IN NO EVENT SHALL THE DIRECT VENDOR'S LIABILITY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM THE USE OF THE PRODUCT OR ITS DOCUMENTATION EXCEED THE PRICE PAID FOR THE PRODUCT.

The manufacture is **not** responsible of loss of toast and/or bagels due to network stability problems due to, but not limited to, fiber cut, worms, DDoS attacks, hackers, viruses, or user stupidity.

# Questions?

Larry Sanders
lsanders@ittc.ku.edu

16 May 2003