Implementing Automatic Handover Solutions for Linux-Based Mobile Devices

Nickolay Amelichev
Open Source & Linux Lab

http://osll.fruct.org
Problem Statement

- **Flexible client-side handover solution**
- **Automatic** handover – no manual user intervention
- Must work on **Linux-based devices** with minimal adaptation
- Always choose **best WLAN** (by some criteria)
Core Components

Monitor Script

Network Evaluator

Switchers
- WLAN $x \rightarrow 3G$
- $3G \rightarrow WLAN \ z$
- ...

Network Information Storage
Evaluator Control Flow

- **Evaluator**
  - Regularly calls **Monitor**
  - **Read** Previous Measurements
  - **Store** Current Measurements

- **Monitor**
  - **Scan** for Neighboring Networks
  - **Measure** Network Parameters
  - **Choose** Best Network(s)
  - **Output** Best Network(s)

- **Network Information Storage**
  - Reports best network(s)
Framework Components

Fixed Components

- Persistence (using Google Protocol Buffers)
- Vote for Best Network
- Measurement Series

Customizable Components (default implementations provided)

- Scanners, which detect networks
- Network Parameters & Measurement Units
- Measurers, which measure network parameters
- Evaluators, which rank networks

Evaluators, which rank networks
**Example: Signal Strength**

**Goal:** rank WLANs by received signal strength (more is better)

**4 Components** Implemented in ~400 LOC:

- **WLAN Scanner** (using *iwlist scanning*)
- **WLAN Measurer** (using results from the scanner)
- **Parameter** “Signal Strength” and Measurement **Units** “dBm” and “%”
- **Signal Strength Evaluator.** Network rank = its signal strength in %.
Example: Signal Strength

- Monitor
  - A regular time interval passes...
  - Calls Evaluator
  - Calls all WLAN Scanners...
  - Returns Scan Results

WLAN Scanner

- 00:1E:58:B8:AB:A3 shivu wlan
- 00:18:E7:8C:B6:D2 SJCE_STUDENT wlan

...
Example: Signal Strength

WLAN Scanner returns Scan Results

Unique Network ID

Info from all scanners is consolidated

Network Information Object

Network Type

Human-Readable Network Name

Information about non-existent networks is removed...
Example: Signal Strength

All measurers are consulted about network parameter values for respective networks.

"What is the value of Signal Strength in %?"

Values for each network are returned... (if they can be measured)

00:1E:58:B8:AB:A3 (shivu): 37%
00:18:E7:8C:B6:D2 (SJCE STUDENT): 92%

Measured values are added to the series and stored...

Average and st. dev for the series is calculated.
Example: Signal Strength

All evaluators are called to rank networks...

“What are the ranks for all networks?”

Ranks for each network are returned...

Network with maximal rank (within some relative threshold) is reported...

This was the only one evaluator, so no voting. SJCE_STUDENT wins.

shivu: rank 37
SJCE_STUDENT: rank 92

SJCE_STUDENT: rank 92

SJCE_STUDENT wins.
Subtle Details

• Last network state is always kept
  – And automatically discarded if too old

• Automatic conversion of measurement units

• Careful calculation of mean and st.dev.

• Stability:
  – Network can safely disappear from scanning results for a brief period of time. It would not participate in ranking, but is not deleted
  – Network ranks are compared within a configurable threshold
  – Network existence time is tracked; too early appearing networks are not ranked
Routing part of the traffic through 3G, and other part through WLAN

- **When both WLAN and 3G are available**, traffic is **divided**
  - Division by protocol type, port #, etc.
  - Switching scripts **switch between WLANs** to get best signal, throughput etc.
- **Else**, everything goes through 3G
Better Switching Scripts

Modern Linux has a powerful firewall (iptables) and supports multiple routing tables (iproute2). We use both to mark traffic and route marked traffic through desired interfaces when 3G and WLAN is available:
Target Platforms

Required

- Linux Kernel 2.6.28+
- Qt 4.7+, QtMobility 1.2+
- libprotobuf-lite

Recommended

- iproute2
- iptables
- openvpn & tun.ko
- wireless-tools
Results & Future Work

Current Results

- Created a flexible framework for creation of custom handover solutions on the client side
- Successfully implemented and tested a signal-strength based handover solution
- Improved Network Switchers offload some traffic through WLAN (e.g., heavy downloads, web surfing)

Future Work

- Transparent switching between WLANs while offloading traffic from 3G
- Offloading based on environmental parameters (e.g., more battery-consuming network gets less traffic)
- Take multiple network parameters into account when deciding which WLAN is best
Questions?

Nickolay Amelichev
namelichev@acm.org

Open Source & Linux Lab,
http://osll.fruct.org, osll@fruct.org