Freifunk-Firmware Testframework

BSc Praktikum WiSe 2015/16

TECHNISCHE UNIVERSITÄT DARMSTADT

darmstadt.freifunk.net

Chaos Darmstadt e.V.

Einleitung

TECHNISCHE UNIVERSITÄT DARMSTADT

Was ist Freifunk?

- nichtkommerzielle Initiative

Einleitung

TECHNISCHE UNIVERSITÄT DARMSTADT

Was ist Freifunk?

- nichtkommerzielle Initiative
- Ziel: Aufbau und Betrieb eines freien WLAN-Meshnetzes

Einleitung

TECHNISCHE UNIVERSITÄT DARMSTADT

Was ist Freifunk?

- nichtkommerzielle Initiative
- Ziel: Aufbau und Betrieb eines freien WLAN-Meshnetzes
- dezentrales Netz besteht aus den Routern vieler einzelner Betreiber

Einleitung

TECHNISCHE UNIVERSITÄT DARMSTADT

Was ist Freifunk?

- nichtkommerzielle Initiative
- Ziel: Aufbau und Betrieb eines freien WLAN-Meshnetzes
- dezentrales Netz besteht aus den Routern vieler einzelner Betreiber
- ca. 20.000 Freifunk-Router in Deutschland

Einleitung

TECHNISCHE UNIVERSITÄT DARMSTADT

Was ist Freifunk?

- nichtkommerzielle Initiative
- Ziel: Aufbau und Betrieb eines freien WLAN-Meshnetzes
- dezentrales Netz besteht aus den Routern vieler einzelner Betreiber
- ca. 20.000 Freifunk-Router in Deutschland
- OpenWrt basierte Firmware von Enthusiasten in den Communities vor Ort entwickelt

Einleitung

TECHNISCHE UNIVERSITÄT DARMSTADT

Was ist Freifunk?

- nichtkommerzielle Initiative
- Ziel: Aufbau und Betrieb eines freien WLAN-Meshnetzes
- dezentrales Netz besteht aus den Routern vieler einzelner Betreiber
- ca. 20.000 Freifunk-Router in Deutschland
- OpenWrt basierte Firmware von Enthusiasten in den Communities vor Ort entwickelt

Einleitung

TECHNISCHE UNIVERSITÄT DARMSTADT

Was ist Freifunk?

- nichtkommerzielle Initiative
- Ziel: Aufbau und Betrieb eines freien WLAN-Meshnetzes
- dezentrales Netz besteht aus den Routern vieler einzelner Betreiber
- ca. 20.000 Freifunk-Router in Deutschland
- OpenWrt basierte Firmware von Enthusiasten in den Communities vor Ort entwickelt

Um die Qualität der ausgelieferten Firmware über Releases hinweg sicherzustellen ist es notwendig diese vor dem Ausrollen ausführlich auf verschiedenen Modellen zu testen. Diese mühsame und zeitaufwendige Prozedur soll in diesem Projekt automatisiert werden.

Aufgabenstellung (1/2)

TECHNISCHE UNIVERSITÄT DARMSTADT

Konstruktion eines Testframeworks, das SOHO-Router mit einer Firmware beschreibt, konfiguriert und anschließend testet.

Aufgabenstellung (1/2)

TECHNISCHE UNIVERSITÄT DARMSTADT

Konstruktion eines Testframeworks, das SOHO-Router mit einer Firmware beschreibt, konfiguriert und anschließend testet.

Hierzu wird die folgende Hardware bereitgestellt:

- Managed Switch, 16 Port

Aufgabenstellung (1/2)

TECHNISCHE UNIVERSITÄT DARMSTADT

Konstruktion eines Testframeworks, das SOHO-Router mit einer Firmware beschreibt, konfiguriert und anschließend testet.

Hierzu wird die folgende Hardware bereitgestellt:

- Managed Switch, 16 Port
- regelbare Steckdosenleiste (z.B. Ubiquiti mPower)

Aufgabenstellung (1/2)

TECHNISCHE UNIVERSITÄT DARMSTADT

Konstruktion eines Testframeworks, das SOHO-Router mit einer Firmware beschreibt, konfiguriert und anschließend testet.

Hierzu wird die folgende Hardware bereitgestellt:

- Managed Switch, 16 Port
- regelbare Steckdosenleiste (z.B. Ubiquiti mPower)
- Embedded ARM-Board, z.B. RasPi2 (Controller)

Aufgabenstellung (1/2)

TECHNISCHE UNIVERSITÄT DARMSTADT

Konstruktion eines Testframeworks, das SOHO-Router mit einer Firmware beschreibt, konfiguriert und anschließend testet.

Hierzu wird die folgende Hardware bereitgestellt:

- Managed Switch, 16 Port
- regelbare Steckdosenleiste (z.B. Ubiquiti mPower)
- Embedded ARM-Board, z.B. RasPi2 (Controller)
- WLAN-Adapter 802.11ac

Aufgabenstellung (2/2)

TECHNISCHE UNIVERSITÄT DARMSTADT

Vorgaben zur Realisierung

- für Debian-basierte Linuxdistributionen (z.B. Raspbian)

[1] vgl. `http://chris.beams.io/posts/git-commit/`

Aufgabenstellung (2/2)

TECHNISCHE UNIVERSITÄT DARMSTADT

Vorgaben zur Realisierung

- für Debian-basierte Linuxdistributionen (z.B. Raspbian)
- mit Python 3 (>=3.3) nach gängigen Standards (z.B. PEP8)

[1] vgl. `http://chris.beams.io/posts/git-commit/`

Aufgabenstellung (2/2)

TECHNISCHE UNIVERSITÄT DARMSTADT

Vorgaben zur Realisierung

- für Debian-basierte Linuxdistributionen (z.B. Raspbian)
- mit Python 3 (>=3.3) nach gängigen Standards (z.B. PEP8)
- unter Revisionskontrolle mit Git und sprechenden Commit-Messages[1]

[1] vgl. `http://chris.beams.io/posts/git-commit/`

Aufgabenstellung (2/2)

TECHNISCHE UNIVERSITÄT DARMSTADT

Vorgaben zur Realisierung

- für Debian-basierte Linuxdistributionen (z.B. Raspbian)
- mit Python 3 (>=3.3) nach gängigen Standards (z.B. PEP8)
- unter Revisionskontrolle mit Git und sprechenden Commit-Messages[1]
- lizensiert unter MIT, LGPL oder vergleichbaren Lizenzen

[1] vgl. `http://chris.beams.io/posts/git-commit/`

Aufgabenstellung (2/2)

TECHNISCHE UNIVERSITÄT DARMSTADT

Vorgaben zur Realisierung

- für Debian-basierte Linuxdistributionen (z.B. Raspbian)
- mit Python 3 (>=3.3) nach gängigen Standards (z.B. PEP8)
- unter Revisionskontrolle mit Git und sprechenden Commit-Messages[1]
- lizensiert unter MIT, LGPL oder vergleichbaren Lizenzen

[1] vgl. `http://chris.beams.io/posts/git-commit/`

Aufgabenstellung (2/2)

TECHNISCHE UNIVERSITÄT DARMSTADT

Vorgaben zur Realisierung

- für Debian-basierte Linuxdistributionen (z.B. Raspbian)
- mit Python 3 (>=3.3) nach gängigen Standards (z.B. PEP8)
- unter Revisionskontrolle mit Git und sprechenden Commit-Messages[1]
- lizensiert unter MIT, LGPL oder vergleichbaren Lizenzen

Die Kommunikation mit den Routern erfolgt im Regelfall per Telnet/SSH bzw. in späteren Ausbaustufen auch per tftp (Recovery, Image flashen) bzw. HTTP (Konfiguration/Ersteinrichtung über Weboberfläche).

[1] vgl. `http://chris.beams.io/posts/git-commit/`

Setup

TECHNISCHE UNIVERSITÄT DARMSTADT

Für den Regelbetrieb ist ein Prüfstand mit folgendem Aufbau vorgesehen:

- Bis zu 6 SOHO Router

Setup

TECHNISCHE UNIVERSITÄT DARMSTADT

Für den Regelbetrieb ist ein Prüfstand mit folgendem Aufbau vorgesehen:

- Bis zu 6 SOHO Router
- LAN- und WAN-Port mit einem Managed Switch verbunden

Setup

TECHNISCHE UNIVERSITÄT DARMSTADT

Für den Regelbetrieb ist ein Prüfstand mit folgendem Aufbau vorgesehen:

- Bis zu 6 SOHO Router
- LAN- und WAN-Port mit einem Managed Switch verbunden
- LAN per Trunk, getagged an den Controller (RasPi2) angebunden

Setup

TECHNISCHE UNIVERSITÄT DARMSTADT

Für den Regelbetrieb ist ein Prüfstand mit folgendem Aufbau vorgesehen:

- Bis zu 6 SOHO Router
- LAN- und WAN-Port mit einem Managed Switch verbunden
- LAN per Trunk, getagged an den Controller (RasPi2) angebunden
- WAN mit Dual-Stack Internet-Anbindung

Setup

TECHNISCHE UNIVERSITÄT DARMSTADT

Für den Regelbetrieb ist ein Prüfstand mit folgendem Aufbau vorgesehen:

- Bis zu 6 SOHO Router
- LAN- und WAN-Port mit einem Managed Switch verbunden
- LAN per Trunk, getagged an den Controller (RasPi2) angebunden
- WAN mit Dual-Stack Internet-Anbindung
- VLAN und ggf. Network Namespaces auf Controller

Funktionsumfang (1/2)

TECHNISCHE UNIVERSITÄT DARMSTADT

Das Testframework soll unter anderem folgende Funktionen bieten:

- Hardwareerkennung (Model, Revision, Primäre MAC-Addresse)

Funktionsumfang (1/2)

TECHNISCHE UNIVERSITÄT DARMSTADT

Das Testframework soll unter anderem folgende Funktionen bieten:

- Hardwareerkennung (Model, Revision, Primäre MAC-Addresse)
- Kaltstart der Geräte

Funktionsumfang (1/2)

TECHNISCHE UNIVERSITÄT DARMSTADT

Das Testframework soll unter anderem folgende Funktionen bieten:

- Hardwareerkennung (Model, Revision, Primäre MAC-Addresse)
- Kaltstart der Geräte
- Tests für einzelne Geräte starten/stoppen

Funktionsumfang (1/2)

TECHNISCHE UNIVERSITÄT DARMSTADT

Das Testframework soll unter anderem folgende Funktionen bieten:

- Hardwareerkennung (Model, Revision, Primäre MAC-Addresse)
- Kaltstart der Geräte
- Tests für einzelne Geräte starten/stoppen
- Auswahl und Flashen der zu testenden Firmware

Funktionsumfang (1/2)

TECHNISCHE UNIVERSITÄT DARMSTADT

Das Testframework soll unter anderem folgende Funktionen bieten:

- Hardwareerkennung (Model, Revision, Primäre MAC-Addresse)
- Kaltstart der Geräte
- Tests für einzelne Geräte starten/stoppen
- Auswahl und Flashen der zu testenden Firmware
- Auszuführende Tests auswählen

Funktionsumfang (1/2)

TECHNISCHE UNIVERSITÄT DARMSTADT

Das Testframework soll unter anderem folgende Funktionen bieten:

- Hardwareerkennung (Model, Revision, Primäre MAC-Addresse)
- Kaltstart der Geräte
- Tests für einzelne Geräte starten/stoppen
- Auswahl und Flashen der zu testenden Firmware
- Auszuführende Tests auswählen
- Abbruch einzelner Tests

Funktionsumfang (1/2)

TECHNISCHE UNIVERSITÄT DARMSTADT

Das Testframework soll unter anderem folgende Funktionen bieten:

- Hardwareerkennung (Model, Revision, Primäre MAC-Addresse)
- Kaltstart der Geräte
- Tests für einzelne Geräte starten/stoppen
- Auswahl und Flashen der zu testenden Firmware
- Auszuführende Tests auswählen
- Abbruch einzelner Tests
- Statusübersicht über laufende Tests

Funktionsumfang (1/2)

TECHNISCHE UNIVERSITÄT DARMSTADT

Das Testframework soll unter anderem folgende Funktionen bieten:

- Hardwareerkennung (Model, Revision, Primäre MAC-Addresse)
- Kaltstart der Geräte
- Tests für einzelne Geräte starten/stoppen
- Auswahl und Flashen der zu testenden Firmware
- Auszuführende Tests auswählen
- Abbruch einzelner Tests
- Statusübersicht über laufende Tests
- Ergebnisreport

Funktionsumfang (2/2)

TECHNISCHE UNIVERSITÄT DARMSTADT

Das Testframework soll unter anderem folgende Funktionen bieten:

- Steuerung der Tests per Weboberfläche und SSH

Funktionsumfang (2/2)

TECHNISCHE UNIVERSITÄT DARMSTADT

Das Testframework soll unter anderem folgende Funktionen bieten:

- Steuerung der Tests per Weboberfläche und SSH
- Erkennung von Software-Defekten und Firmware-Recovery

Funktionsumfang (2/2)

TECHNISCHE UNIVERSITÄT DARMSTADT

Das Testframework soll unter anderem folgende Funktionen bieten:

- Steuerung der Tests per Weboberfläche und SSH
- Erkennung von Software-Defekten und Firmware-Recovery
- Automatischer Abruf neuer Firmware-Images vom Buildserver

Funktionsumfang (2/2)

TECHNISCHE UNIVERSITÄT DARMSTADT

Das Testframework soll unter anderem folgende Funktionen bieten:

- Steuerung der Tests per Weboberfläche und SSH
- Erkennung von Software-Defekten und Firmware-Recovery
- Automatischer Abruf neuer Firmware-Images vom Buildserver
- Unattended Betriebsmodus

Funktionsumfang (2/2)

TECHNISCHE UNIVERSITÄT DARMSTADT

Das Testframework soll unter anderem folgende Funktionen bieten:

- Steuerung der Tests per Weboberfläche und SSH
- Erkennung von Software-Defekten und Firmware-Recovery
- Automatischer Abruf neuer Firmware-Images vom Buildserver
- Unattended Betriebsmodus
- Ergebnisreport per Email

Funktionsumfang (2/2)

TECHNISCHE UNIVERSITÄT DARMSTADT

Das Testframework soll unter anderem folgende Funktionen bieten:

- Steuerung der Tests per Weboberfläche und SSH
- Erkennung von Software-Defekten und Firmware-Recovery
- Automatischer Abruf neuer Firmware-Images vom Buildserver
- Unattended Betriebsmodus
- Ergebnisreport per Email
- Archivierung von Testberichten und Logdateien (syslog, dmesg)

Beispielhafte Tests

TECHNISCHE UNIVERSITÄT DARMSTADT

- Durchlaufen des Konfigurationsassistenten

Beispielhafte Tests

TECHNISCHE UNIVERSITÄT DARMSTADT

- Durchlaufen des Konfigurationsassistenten
- Log-Prüfung, z.B. auf Kernel Panics

Beispielhafte Tests

TECHNISCHE UNIVERSITÄT DARMSTADT

- Durchlaufen des Konfigurationsassistenten
- Log-Prüfung, z.B. auf Kernel Panics
- Verbindung als WLAN-Client

Beispielhafte Tests

TECHNISCHE UNIVERSITÄT DARMSTADT

- Durchlaufen des Konfigurationsassistenten
- Log-Prüfung, z.B. auf Kernel Panics
- Verbindung als WLAN-Client
- Erkennung von Mesh-Verbindungen

Beispielhafte Tests

TECHNISCHE UNIVERSITÄT DARMSTADT

- Durchlaufen des Konfigurationsassistenten
- Log-Prüfung, z.B. auf Kernel Panics
- Verbindung als WLAN-Client
- Erkennung von Mesh-Verbindungen
- Prüfung auf IPv6-Konnektivität

Beispielhafte Tests

TECHNISCHE UNIVERSITÄT DARMSTADT

- Durchlaufen des Konfigurationsassistenten
- Log-Prüfung, z.B. auf Kernel Panics
- Verbindung als WLAN-Client
- Erkennung von Mesh-Verbindungen
- Prüfung auf IPv6-Konnektivität
- Erreichbarkeit der Gateways

Beispielhafte Tests

TECHNISCHE UNIVERSITÄT DARMSTADT

- Durchlaufen des Konfigurationsassistenten
- Log-Prüfung, z.B. auf Kernel Panics
- Verbindung als WLAN-Client
- Erkennung von Mesh-Verbindungen
- Prüfung auf IPv6-Konnektivität
- Erreichbarkeit der Gateways
- Verbindung zu Update-Server

Beispielhafte Tests

TECHNISCHE UNIVERSITÄT DARMSTADT

- Durchlaufen des Konfigurationsassistenten
- Log-Prüfung, z.B. auf Kernel Panics
- Verbindung als WLAN-Client
- Erkennung von Mesh-Verbindungen
- Prüfung auf IPv6-Konnektivität
- Erreichbarkeit der Gateways
- Verbindung zu Update-Server
- Funktion des Autoupdaters

Beispielhafte Tests

TECHNISCHE UNIVERSITÄT DARMSTADT

- Durchlaufen des Konfigurationsassistenten
- Log-Prüfung, z.B. auf Kernel Panics
- Verbindung als WLAN-Client
- Erkennung von Mesh-Verbindungen
- Prüfung auf IPv6-Konnektivität
- Erreichbarkeit der Gateways
- Verbindung zu Update-Server
- Funktion des Autoupdaters
- Zeitsynchronisation mit NTP

Beispielhafte Tests

TECHNISCHE UNIVERSITÄT DARMSTADT

- Durchlaufen des Konfigurationsassistenten
- Log-Prüfung, z.B. auf Kernel Panics
- Verbindung als WLAN-Client
- Erkennung von Mesh-Verbindungen
- Prüfung auf IPv6-Konnektivität
- Erreichbarkeit der Gateways
- Verbindung zu Update-Server
- Funktion des Autoupdaters
- Zeitsynchronisation mit NTP
- Internet-Verbindung für Clients

Beispielhafte Tests

TECHNISCHE UNIVERSITÄT DARMSTADT

- Durchlaufen des Konfigurationsassistenten
- Log-Prüfung, z.B. auf Kernel Panics
- Verbindung als WLAN-Client
- Erkennung von Mesh-Verbindungen
- Prüfung auf IPv6-Konnektivität
- Erreichbarkeit der Gateways
- Verbindung zu Update-Server
- Funktion des Autoupdaters
- Zeitsynchronisation mit NTP
- Internet-Verbindung für Clients
- Überprüfung der laufenden Dienste