

Segurança de redes em IoT Off Grid para coleta de dados via Drone

Resumo: IoT abrange uma ampla variedade de objetos, tecnologias, padrões de comunicação, sensores, atuadores em ambientes com energia e comunicação em rede. Em um ambiente desprovido de infraestrutura elétrica e de comunicação como prover tecnologia para monitoração e gerenciamento de utilidades diversas baseado em Internet das Coisas IoT? Como coletar estas informações, via drone, de um ambiente Off Grid com segurança? Um mapeamento sistemático da literatura será realizado para identificar o estado da arte. O resultado esperado é uma proposição de arquitetura de software que contemple a segurança da rede durante a coleta de dados do ambiente de IoT Off Grid.

Sumário

Cenário	1
Configuração dos Access Point.....	2
Configuração no micro acoplado ao drone.....	3
Configuração da rede wireless	3
Configuração da transferência dos dados.....	4
Configurar SSH.....	4
Script de transferência dos arquivos.....	4
Configuração no micro em cada estação	7

Cenário

Para validação dos procedimentos relatados no presente artigo foi usado o seguinte cenário:

Drone DJI AIR 2S tende em seu corpo acoplado um power bank marca KAI DI modelo KD-905 de 5000mAh alimentando um Raspberry Pi Zero 2 W e responsável pela coleta dos dados nas estações remotas.

Três diferentes estações off grid com os dados a serem coletados conforme descrito a seguir.

Estação 1 – composta por um hardware Raspberry Pi 3 model B, 1GB RAM, wireless 802.11 b/g/n configurado para ser um ponto de acesso com SSID offg01 e alimentada por um power bank PINENG modelo PN-999 de 20000mAh.

Estação 2 – composta por um hardware Raspberry Pi 4 model B, 8GB RAM, wireless 802.11 b/g/n/ac configurado para ser um ponto de acesso com SSID offg02 e alimentada por um power bank GEONAV modelo PB10KBK de 10000mAh.

Estação 3 – composta por um hardware Raspberry Pi 4 model B, 8GB RAM, wireless 802.11 b/g/n/ac configurado para ser um ponto de acesso com SSID offg03 e alimentada por um power bank INOVA modelo PB03 de 5000mAh.

Configuração dos Access Point

Cada uma das estações teve a sua configuração de access point conforme a documentação que se encontra na página de documentação da Raspberrypi conforme o link que segue:

Setting up a Routed Wireless Access Point

<https://www.raspberrypi.com/documentation/computers/configuration.html#setting-up-a-routed-wireless-access-point>

Configuração da estação 01

- SSID – offg01
- Password – 100senha
- Rede 192.168.4.0/24
- DHCP 192.168.4.100, 192.168.4.110
- Radio canal 7

Configuração da estação 02

- SSID – offg02
- Password – 100senha
- Rede 192.168.5.0/24
- DHCP 192.168.5.100, 192.168.5.110
- Radio canal 1

Configuração da estação 03

- SSID – offg03
- Password – 100senha
- Rede 192.168.6.0/24
- DHCP 192.168.6.100, 192.168.6.110
- Radio canal 3

Configuração no micro acoplado ao drone

Configuração da rede wireless

Este Raspberry Pi Zero 2W que está acoplado ao drone detecta automaticamente a rede wireless de cada um dos Acces Point das estações off grid. Para permitir que o micro do drone identifique o SSID de cada estação é necessário configurar WPA_SUPPLICANT conforme segue:

Editar o arquivo /etc/wpa_supplicant/wpa_supplicant.conf incluindo os SSIDs de cada estação.

```
network={
    ssid="offg01"
    psk="100senha"
}
network={
    ssid="offg02"
    psk="100senha"
}
network={
    ssid="offg03"
    psk="100senha"
}
```

Um link para mais detalhes desta configuração:

https://www.raspberrypi-spy.co.uk/2017/04/manually-setting-up-pi-wifi-using-wpa_supplicant-conf/

Conforme a proximidade do drone com relação a estação ao identificar o SSID a conexão é estabelecida. Assim o micro do drone fica associado a mesma rede da estação o que permite a transferência automática dos arquivos desde a estação para o micro do drone.

Configuração da transferência dos dados

Configurar SSH

A transferência de arquivos desde a estação até o micro acoplado ao drone será via SSH e assim é necessário liberar na estação o uso de SSH na porta 22 via configurações do raspberry.

Usando Preferencias / Raspberry configurações / interfaces é possível habilitar o SSH.

Para que a conexão SSH ocorra sem necessidade de informar usuário e senha é necessário habilitar um login sem password com o uso de chaves local/remota.

Na estação criar uma chave ssh com

```
ssh-keygen -t rsa -b 4096 -C your\_email@domain.com
```

Copiar esta chave para o micro que fica acoplado ao drone

```
ssh-copy-id remote_username@server_ip_address
```

uma documentação deste procedimento pode ser encontrada em:

<https://linuxize.com/post/how-to-setup-passwordless-ssh-login/>)

Script de transferência dos arquivos

O script em python sc06.py para identificar se o micro acoplado ao drone está visualizando a rede wireless de cada estação e assim iniciar a transferência dos arquivos é o que segue:

```
import os
import time
from datetime import datetime

hostname1 = "192.168.4.1" # primeiro no
hostname2 = "192.168.5.1" # Segundo no
hostname3 = "192.168.6.1" # Terceiro no

i = 1
n1 = 0
n2 = 0
n3 = 0

#and then check the response...

while i > 0 :
```

```

response = os.system("ping -c 1 " + hostname1)
if response == 0 and n1 == 0:
    print (hostname1, 'is up!')
    print ("vai chamar scp")
    data_hora_atual = datetime.now()
    dh_texto = data_hora_atual.strftime ('%d/%m/%Y %H:%M')
    print(dh_texto)
    resp= os.system("scp -p pi@192.168.4.1:/home/pi/arquivos/*.*
/home/pi/Doutorado/no_01/" )
    data_hora_atual = datetime.now()
    dh_texto = data_hora_atual.strftime ('%d/%m/%Y %H:%M')
    print(dh_texto)
    n1=1

response = os.system("ping -c 1 " + hostname2)
if response == 0 and n2 == 0:
    print (hostname2, 'is up!')
    print ("vai chamar scp")
    data_hora_atual = datetime.now()
    dh_texto = data_hora_atual.strftime ('%d/%m/%Y %H:%M')
    print(dh_texto)
    resp= os.system("scp -p pi@192.168.5.1:/home/pi/arqs_no_02/*.*
/home/pi/Doutorado/no_02/" )
    data_hora_atual = datetime.now()
    dh_texto = data_hora_atual.strftime ('%d/%m/%Y %H:%M')
    print(dh_texto)
    n2=1

response = os.system("ping -c 1 " + hostname3)
if response == 0 and n3 == 0:
    print (hostname3, 'is up!')
    print ("vai chamar scp")

```

```

data_hora_atual = datetime.now()
dh_texto = data_hora_atual.strftime('%d/%m/%Y %H:%M')
print(dh_texto)
resp= os.system("scp -p pi@192.168.6.1:/home/pi/arquivos/*.*
/home/pi/Doutorado/no_03/" )
data_hora_atual = datetime.now()
dh_texto = data_hora_atual.strftime('%d/%m/%Y %H:%M')
print(dh_texto)
n3=1
print ('intervalo de espera 20 segundos')
time.sleep(20)

```

Este script para ser executado automaticamente após o boot deste micro que fica acoplado ao drone, pode ser configurado via crontab com o comando

Crontab crontab -e

@reboot /usr/local/src/inicio.sh

Assim o script inicio.sh tem o seguinte conteúdo:

Criando script para limpar file*.* e chamar programa que copia em

/usr/local/src/inicio.sh (com chmod 777)

#!/usr/bin/bash

Cd /home/pi/Doutorado

Rm file*.*

Python sc06.py >> /home/pi/log

Uma documentação de como editar crontab:

https://www.raspberrypi.com/documentation/computers/using_linux.html#editing-the-crontab-file

Configuração no micro em cada estação

Para cada estação os arquivos com os dados coletados no ambiente de lot Off Grid devem ser colocados em um diretório o qual será integralmente copiado para o micro do drone.

Na configuração deste trabalho os arquivos estavam localizados em `/home/pi/arquivos`.