

Réseau Wi-Fi multi-sauts sur plateforme ESP

Arnaud Palgen

Département d'informatique
Service de réseaux et télécommunications
Université de Mons



1^{er} juillet 2020

- 1 Introduction
- 2 Protocoles de routage
- 3 ESP-MESH
- 4 Mise en oeuvre
- 5 Conclusion

Section 1

Introduction

Objectifs du projet

- Découvrir les protocoles de routages MESH existants
- Étudier le protocole AODV
- Étudier le protocole ESP-MESH
- Mettre en oeuvre le protocole ESP-MESH

Réseau Wi-Fi traditionnel

Définition

Réseau composé d'un noeud central appelé le point d'accès (AP) directement connecté à tous les autres noeuds (stations) du réseau. L'AP a alors pour rôle d'acheminer les paquets d'une station à une autre mais aussi des paquets vers des adresses IP externes.

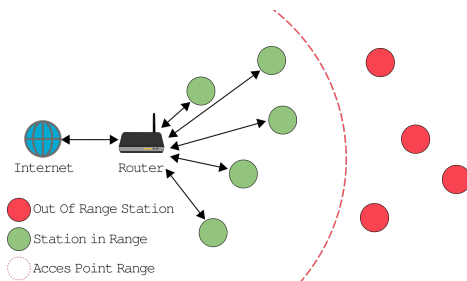


Figure – Réseau Wi-Fi traditionnel.

Réseau MESH multi-sauts

Définition

Réseau dans lequel les noeuds transmettent leurs paquets mais aussi ceux des noeuds à la portée de leur radio. Les routes utilisées pour acheminer les paquets sont obtenues à l'aide d'un protocole de routage.

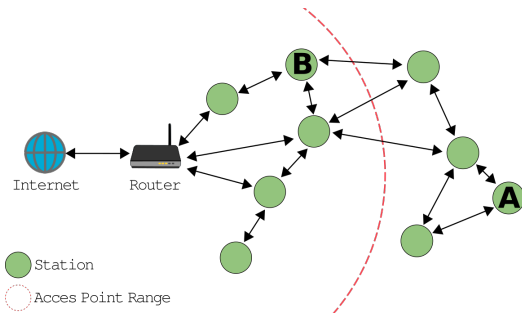


Figure – Réseau MESH.

Réseau MESH multi-sauts

Définition

Réseau dans lequel les noeuds transmettent leurs paquets mais aussi ceux des noeuds à la portée de leur radio. Les routes utilisées pour acheminer les paquets sont obtenues à l'aide d'un protocole de routage.

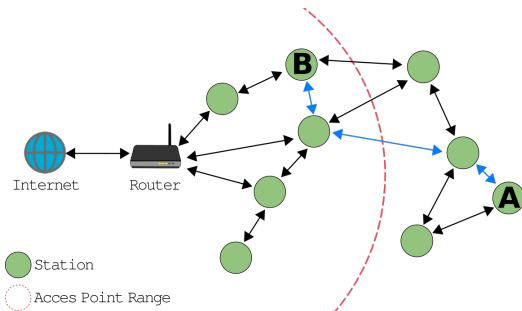


Figure – Réseau MESH.

Réseau MESH multi-sauts

Définition

Réseau dans lequel les noeuds transmettent leurs paquets mais aussi ceux des noeuds à la portée de leur radio. Les routes utilisées pour acheminer les paquets sont obtenues à l'aide d'un protocole de routage.

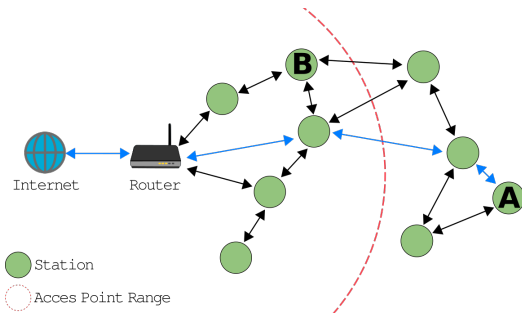


Figure – Réseau MESH.

Plateforme ESP32

Les noeuds du réseau MESH sont des cartes de développement équipées d'un ESP32-WROOM32.

- Wi-Fi 802.11 b/g/n 2.4 GHz
- 2 microprocesseurs Xtensa[®] 32-bit LX6
- Entre 3 et 4 euros
- Mécanismes d'économies d'énergie

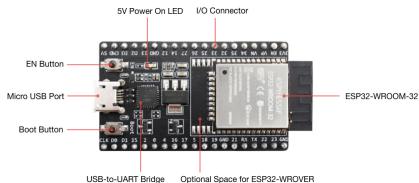


Figure – ESP32-DevKitC V4 with ESP32-WROOM-32 module [1].

Section 2

Protocoles de routage

Classification des potocoles de routage

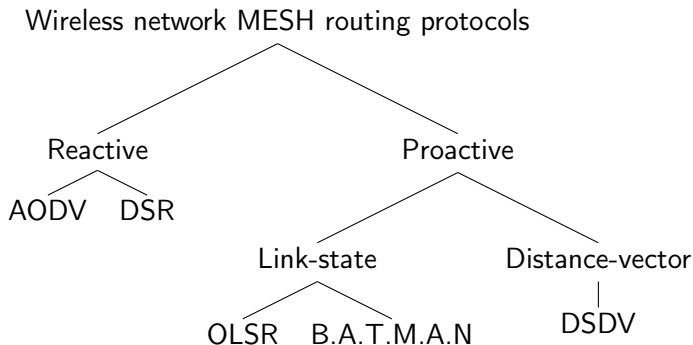


Figure – Classifications des protocoles de routages.

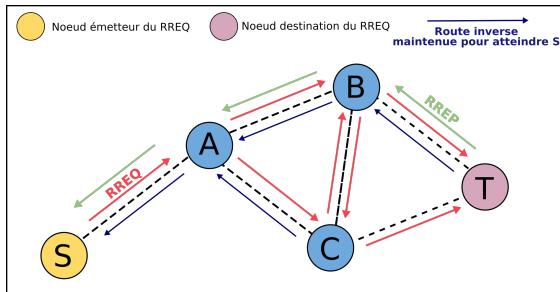
AODV :	Ad-hoc On-Demand Distance Vector
DSR :	Dynamic Source Routing
OLSR :	Optimized Link State Routing
B.A.T.M.A.N :	Better Approach to Mobile Adhoc Networking
DSDV :	Destination Sequence Distance Vector

AODV

Ad-hoc On-demand Distance Vector, RFC3561, est un protocole de routage réactif à vecteur de distance ne nécessitant pas une grande puissance de calcul et peu de mémoire.

Il définit 3 types de messages :

- **Route Request (RREQ)**
- **Route Reply (RREP)**
- **Route Error (RERR)**



Section 3

ESP-MESH

ESP-MESH

Protocole d'Espressif permettant d'établir un réseau MESH avec des ESP32.

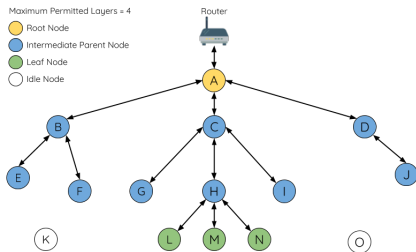


Figure – Réseau ESP-MESH

- minimise la hauteur de l'arbre
- protocole proactif
- construit sur 802.11
- permet la connectivité avec un réseau IP classique
- réagit automatiquement aux changements de topologie

Élection de la racine

- Si la racine est fixée par configuration, elle se connecte au routeur et le vote n'a pas lieu.
- Sinon, un vote a lieu pour élire la racine.

Élection de la racine

- Chaque noeud émet une *probe request*

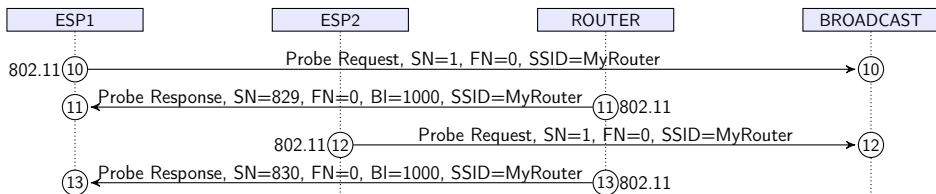


Figure – Diagramme de séquence d'une itération du vote.

Élection de la racine

- Chaque noeud vote en émettant un beacon contenant les informations du noeud ayant le meilleur RSSI avec le point d'accès Wi-Fi.

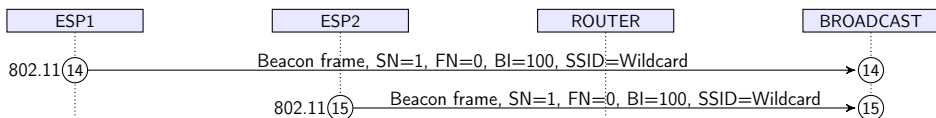


Figure – Diagramme de séquence d'une itération du vote.

Élection de la racine

A la fin du vote, chaque noeud calcule le ratio suivant :

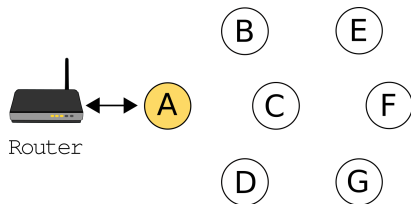
$$ratio = \frac{V(i)}{N}$$

$V(i)$ = nombre de votes pour le noeud i

N = nombre de noeuds participants

La racine du réseau sera le noeud ayant un ratio supérieur à un seuil fixé.

Formation des autres couches



Les noeuds à portée de la racine vont s'y connecter.

Figure – Construction du réseau.

Formation des autres couches

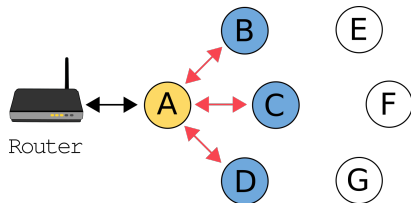


Figure – Construction du réseau.

Chaque noeud du réseau ESP-MESH émet des beacons contenant les informations suivantes :

- Type de noeud
- Couche sur laquelle se trouve le noeud
- Nombre de couches maximum autorisées dans le réseau
- Nombre de noeuds enfants
- Nombre maximum d'enfants

Formation des autres couches

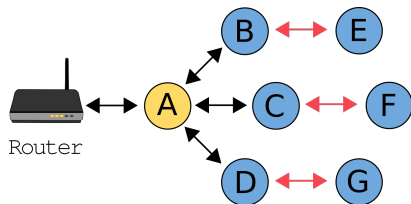


Figure – Construction du réseau.

Grâce à ces informations, les noeuds idle vont sélectionner un parent selon deux critères :

- 1 La couche sur laquelle se trouve le candidat parent
- 2 Le nombre d'enfants du candidat parent

Communications internes

- Les paquets ESP-MESH sont envoyés via `esp_mesh_send()`
- Ils sont reçus via la `esp_mesh_recv()`
- Une transmission multi-sauts utilise un paquet ESP-MESH transporté entre chaque noeud par une trame Wi-Fi différente.

Source	Destination	Protocol	Length
Espressi_0d:7e:1c	Espressi_0d:83:09	LLC	112

Frame 386: 112 bytes on wire (896 bits), 112 bytes captured (896 bits)

Radiotap Header v0, Length 26

802.11 radio information

IEEE 802.11 QoS Data, Flags:TC

Logical-Link Control

Organization Code: 18:fe:34 (Espressif Inc.)

Protocol ID: 0xeeee

Data (48 bytes)

0000	21 07 30 00 31 06 40 01	00 00 00 00 00 00 00 3c 71
0010	bf 0d 7e 1c 01 00 00 00 01 00 00 00	ee ee ee ee
0020	ee ee ee ee ee ee ee ee ee ee ee ee ee ee ee ee	

Figure – Capture Wireshark d'un paquet ESP-MESH.

Communications internes

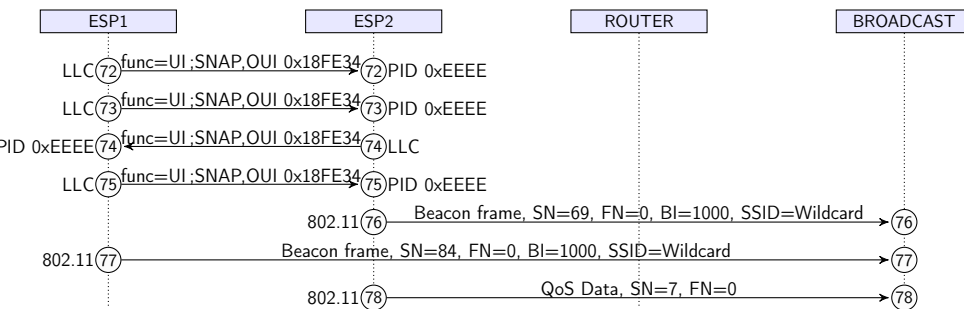


Figure – Diagramme de séquence d'échange de données.

Section 4

Mise en oeuvre

Proxy

- La racine est l'intermédiaire entre le réseau ESP-MESH et l'extérieur
- Établissement d'une connexion avec une IP externe via des sockets TCP
- Communications bidirectionnelles par la fonction `select()`
- Connexion initialisée par un noeud du réseau

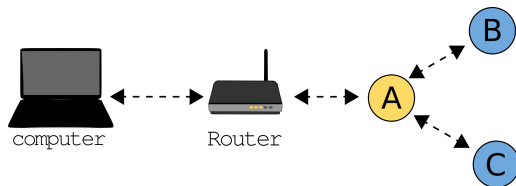
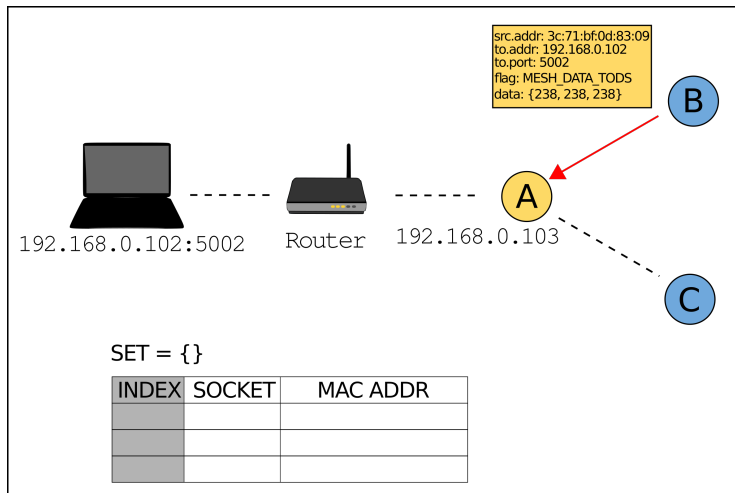


Figure – Proxy.

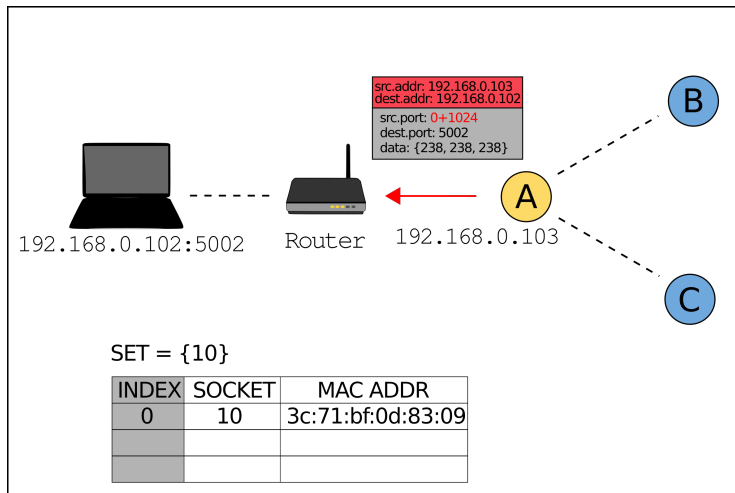
Fonctionnement du Proxy

1 Paquet ESP-MESH transmis à la racine



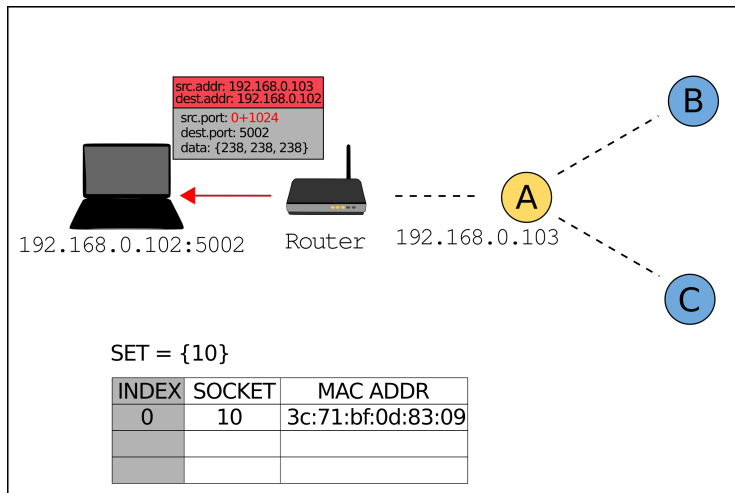
Fonctionnement du Proxy

2 Paquet TCP envoyé à la destination



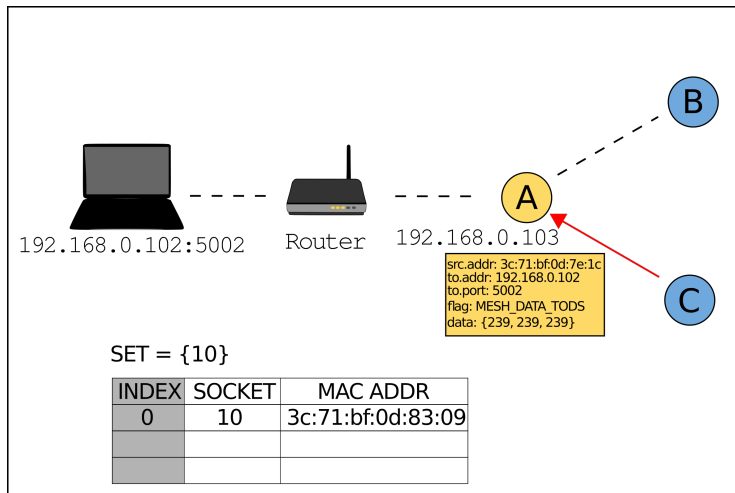
Fonctionnement du Proxy

2 Paquet TCP envoyé à la destination



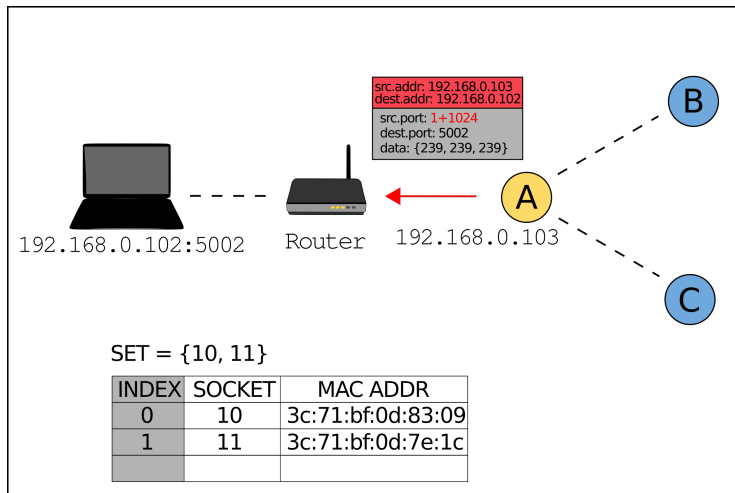
Fonctionnement du Proxy

3 Paquet ESP-MESH transmis à la racine



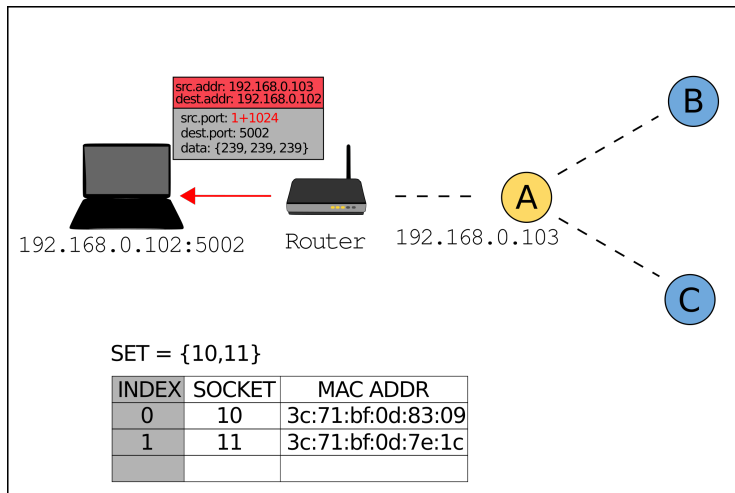
Fonctionnement du Proxy

4 Paquet TCP envoyé à la destination



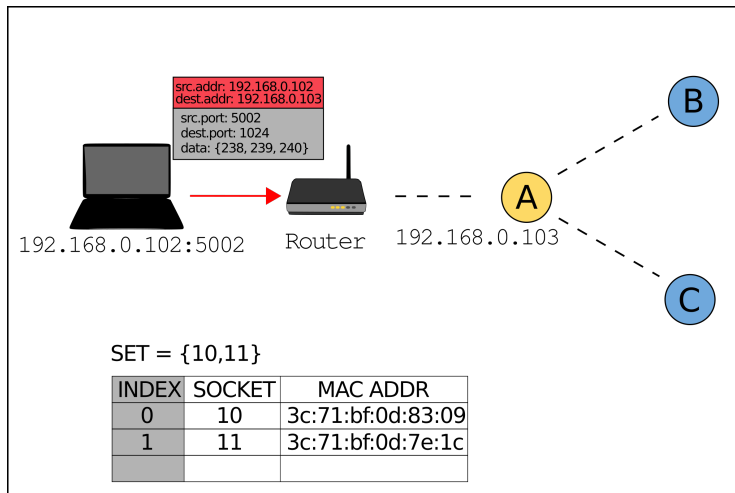
Fonctionnement du Proxy

4 Paquet TCP envoyé à la destination



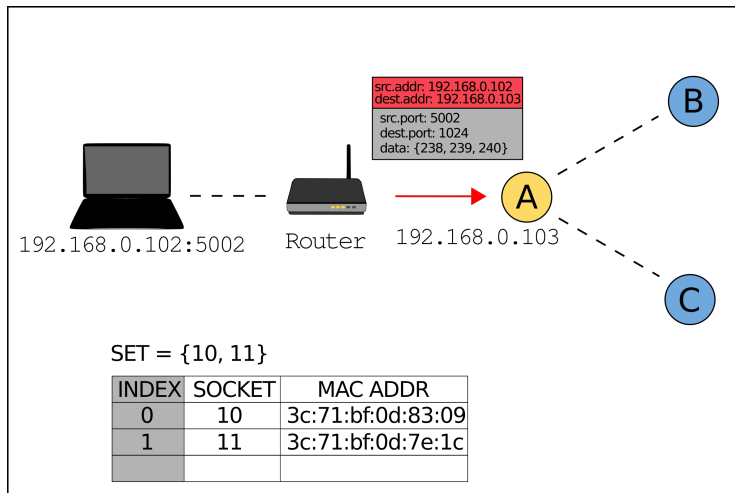
Fonctionnement du Proxy

5 Paquet TCP transmis à la racine



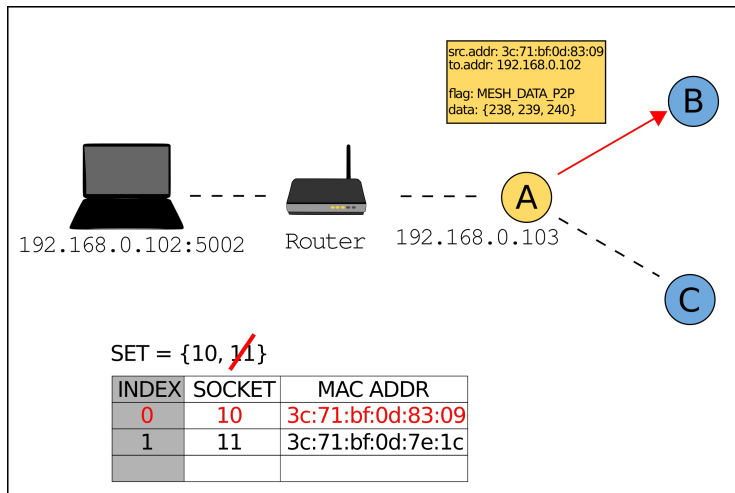
Fonctionnement du Proxy

5 Paquet TCP transmis à la racine



Fonctionnement du Proxy

6 Paquet ESP-MESH transmis au noeud de destination



Limitation

Le driver Wi-Fi d'IDF, l'environnement de développement choisi, ne permet pas d'avoir plusieurs connexions simultanées avec des noeuds voisins

- Cause de la topologie d'ESP-MESH
- ESP-NOW permet de communiquer simultanément en Wi-Fi avec plusieurs voisins
- ESP-NOW pourrait permettre le développement d'un protocole tel qu'AODV

Performances

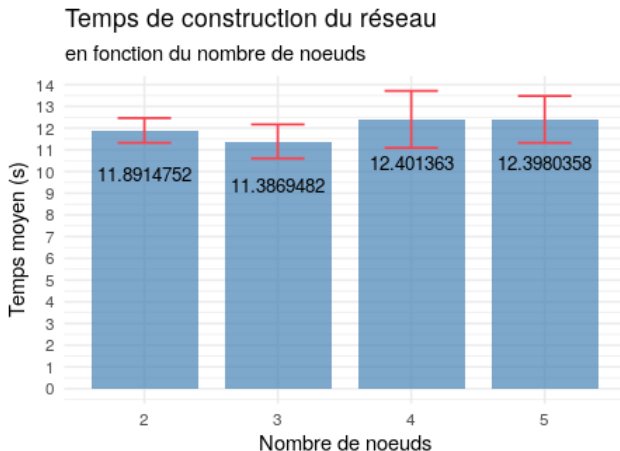
Espressif fournit une estimation des performances d'ESP-MESH pour un réseau de maximum 100 noeuds, 6 couches et un nombre d'enfants par noeud de 6.

Temps de construction	< 60 secondes
Latence par saut	10 à 30 millisecondes
Temps de réparation	Si la racine tombe : < 10 secondes Si un noeud enfant tombe : < 5 secondes

Table – Performances d'ESP-MESH [6].

Performances

Nous avons effectué des mesures du temps de construction du réseau avec 2, 3, 4 et 5 noeuds (après la remise du projet).



Extension à Wireshark

esp-mesh					
No.	Time	Source	Destination	Protocol	Length
378	12.306832933	Espressi_0d:7e:1c	Espressi_0d:83:09	esp-mesh	252
380	12.309240035	Espressi_0d:7e:1c	Espressi_0d:83:09	esp-mesh	252
382	12.310708667	Espressi_0d:7e:1c	Espressi_0d:83:09	esp-mesh	92
384	12.313291055	Espressi_0d:83:09	Espressi_0d:7e:1c	esp-mesh	92
386	12.316775314	Espressi_0d:7e:1c	Espressi_0d:83:09	esp-mesh	112
388	12.341319660	Espressi_0d:7e:1c	Espressi_0d:83:09	esp-mesh	112
390	12.343526265	Espressi_0d:7e:1c	Espressi_0d:83:09	esp-mesh	112

Frame 386: 112 bytes on wire (896 bits), 112 bytes captured (896 bits) on interface 0

- Radiotap Header v0, Length 26
- 802.11 radio information
- IEEE 802.11 QoS Data, Flags:TC
- Logical-Link Control
 - DSAP: SNAP (0xaa)
 - SSAP: SNAP (0xaa)
 - Control field: U, func=UI (0x03)
Organization Code: 18:fe:34 (Espressif Inc.)
Protocol ID: 0xeeee
- Data (48 bytes)
 - Data: 21073000310640010000000000003c71bf0d7e1c01000000...
 - [Length: 48]
- ESP-MESH Protocol
 - Destination address: 00:00:00_00:00:00 (00:00:00:00:00:00)
 - Source address: Espressi_0d:7e:1c (3c:71:bf:0d:7e:1c)
 - Flags: 0100000001000000
 - Data: ee

Figure – Aperçu de Wireshark utilisant le dissecteur.

Section 5

Conclusion

Conclusion

- L'étude d'ESP-MESH a permis d'en apprendre un maximum sur ce protocole malgré l'absence du code source.
- La mise en oeuvre d'ESP-MESH a permis d'acquérir une maîtrise avancée des sockets en C.
- Le développement du dissecteur Wireshark pourrait être amélioré.
- L'implémentation d'un protocole de routage tel qu'AODV pourrait être réalisée à titre de comparaison avec ESP-MESH.
- Une étude du fonctionnement d'ESP-NOW pourrait être réalisée.

Section 6

Références

Références I

 [ESP32-DevKitC V4 Getting Started Guide.](https://docs.espressif.com/projects/esp-idf/en/latest/hw-reference/get-started-devkitc.html)

<https://docs.espressif.com/projects/esp-idf/en/latest/hw-reference/get-started-devkitc.html>.

Accessed : 04-06-2020.

 [ESP-IDF Programming Guide.](https://docs.espressif.com/projects/esp-idf/en/v3.3.1/)

<https://docs.espressif.com/projects/esp-idf/en/v3.3.1/>.

Accessed : 04-06-2020.

 [Espressif Systems.](#)

ESP32 Series Datasheet, 2020.

Rev : 3.4.

 [E. Belding-Royer C. Perkins and S. Das.](#)

Ad hoc on-demand distance vector (aodv) routing.

RFC 3561, RFC Editor, July 2003.

Références II



802.11 lua dissector.

[https:](https://ask.wireshark.org/question/16067/80211-lua-dissector/)

[//ask.wireshark.org/question/16067/80211-lua-dissector/](https://ask.wireshark.org/question/16067/80211-lua-dissector/).

Accessed : 09-06-2020.



ESP-MESH api guide.

[https://docs.espressif.com/projects/esp-idf/en/v3.3.1/
api-guides/mesh.html](https://docs.espressif.com/projects/esp-idf/en/v3.3.1/api-guides/mesh.html).

Accessed : 04-06-2020.