



UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA

# **Diseño de un mecanismo de clustering en redes ad hoc para sistemas distribuidos en el contexto del sistema TLÖN**

**Angela Consuelo Checa Hurtado**

Universidad Nacional de Colombia  
Facultad de Ingeniería, Departamento de Sistemas e Industrial  
Bogotá, Colombia  
2019



# **Diseño de un mecanismo de clustering en redes ad hoc para sistemas distribuidos en el contexto del sistema TLÖN**

**Angela Consuelo Checa Hurtado**

Trabajo de grado presentado como requisito parcial para optar al título de:  
**Magister en Ingeniería Telecomunicaciones**

Director:  
Ph.D. Jorge Eduardo Ortiz Triviño

Línea de Investigación: Computación Aplicada

Grupo de Investigación en Lenguajes de Programación Distribuidos y Redes de Telecomunicaciones Dinámicas  
- TLÖN

Universidad Nacional de Colombia  
Facultad de Ingeniería, Departamento de Sistemas e Industrial  
Bogotá, Colombia  
2019



*Quien ve a todos los seres en su propio Ser, y su propio Ser  
en todos los seres, pierde el temor por completo.*

*Isa Upanishad*



## Agradecimientos

Abrazo fraternalmente a todas las personas que ayudaron en la realización de este proyecto y a aquellas que ofrecieron su apoyo incondicional en los momentos más difíciles. Todos los que caminaron conmigo en este tiempo aportaron de una u otra forma a mi crecimiento profesional. Agradezco especialmente a: Dilia Rosa Hurtado Mazorra, MD. Luis Arturo Checa, Helena Checa, MD. Luis Felipe Checa, Andres Checa, Sara Sofía Checa, Natalia Eraso, Amelia Checa, MD. Jimmy Eraso, MD. Monica Almeida, PhD. Jorge Eduardo Ortiz Triviño, PhD. Henry Zarate Ceballos, PhD. Lauren Raz, MSc. Henry Agudelo Zamora, MSc. Jorge Ernesto Parra, MSc. Juan Carlos Rivera y a todos los miembros del grupo de investigación TLÖN y del semillero de investigación en Seguridad Informática Uqbar. También extendiendo mi gratitud al Instituto de Ciencias Naturales y al programa de Informática de la Biodiversidad por brindarme un segundo hogar durante los últimos años.



## Resumen

*Clustering* es el proceso de agrupamiento de nodos o entidades en un sistema. Su implementación en las redes ad hoc inalámbricas permite agregar características de escalabilidad y auto-organización. Por otra parte, en sistemas altamente distribuidos y paradigmas como IoT y computación en la nube, juega un papel importante para solucionar problemas de asignación de carga, distribución de tareas y recursos. En este trabajo, se definió un modelo teórico de clúster físico y lógico mediante el uso de Bi-grafos y se planteó a un mecanismo de clustering para redes ad hoc en el contexto del sistema TLÖN, que está orientado al procesamiento distribuido. En éste, los nodos *gateway* que intercomunican diferentes clústeres tienen rol principal y el mayor nivel de jerarquía. Además, se describe la implementación de un escenario experimental de tres clústeres de redes ad hoc concatenados. Finalmente se presentan los resultados del análisis de la estabilidad de la conexión intra-cluster e inter-clúster mediante los parámetros de latencia y throughput.

**Palabras clave:** Clúster, Redes ad hoc, Redes inalámbricas, Sistemas distribuidos, Internet de las cosas, Computación en la nube, Nube móvil, escalabilidad, agrupamiento, territorio, sistema social-inspirado.

## Abstract

*Clustering* is the process of grouping nodes or entities in a system. Its implementation in wireless ad hoc networks allows adding features of scalability and self-organization. Additionally, in highly distributed systems and paradigms, such as IoT or Cloud Computing, it plays an important role in solving problems of load allocation, tasks assignment and resource distribution. In this research, a theoretical model of physical and logical cluster was defined making use of *Bigraphs* and a new approach of clustering mechanism is proposed for in ad hoc networks in the context of the TLÖN system, which is oriented to distributed processing. In this one the *gateway* nodes interconnect different clusters and have the main role and the highest level of hierarchy. In addition, an experimental scenario of three concatenated ad hoc network clusters was implemented. Finally, the results of the stability analysis of intra-cluster and inter-cluster connectivity were analyzed using the latency and throughput parameters.

**Keywords:** Cluster, Clustering, Ad hoc Networks, MANET, Wireless Networks, Distributed System, Cloud computing, Mobile Cloud, scalability, territory, Social-inspired system.

# Contenido

|  |            |
|--|------------|
| <b>Agradecimientos</b>   | <b>VII</b> |
| <b>Resumen</b>   | <b>IX</b>  |
| <b>Lista de símbolos</b>   | <b>XVI</b> |
| <b>1. Introducción</b>   | <b>2</b>   |
| 1.1. Justificación . . . . .   | 4          |
| 1.2. Identificación del Problema . . . . .                                       | 5          |
| 1.3. Objetivos . . . . .   | 6          |
| 1.3.1. Objetivo General . . . . .  | 6          |
| 1.3.2. Objetivos Específicos . . . . .   | 6          |
| 1.4. Sobre este trabajo . . . . .  | 6          |
| <b>2. Fundamentos de redes ad hoc</b>  | <b>8</b>   |
| 2.1. Redes ad hoc . . . . .  | 8          |
| 2.1.1. Tipos de redes ad hoc . . . . .   | 9          |
| 2.1.2. Aplicaciones de las redes ad hoc . . . . .                                | 12         |
| 2.1.3. Enrutamiento en redes ad hoc . . . . .                                    | 14         |
| 2.1.4. Clasificación de los algoritmos de enrutamiento en redes ad hoc . . . . . | 15         |
| 2.1.5. Algoritmos de enrutamiento más conocidos . . . . .                        | 23         |
| 2.2. Clustering en redes ad hoc . . . . .  | 26         |
| 2.2.1. Esquemas de clustering en redes ad hoc . . . . .                          | 26         |
| 2.3. Sistemas distribuidos . . . . .   | 28         |
| 2.3.1. Computación en la nube . . . . .  | 28         |
| 2.3.2. Internet de las Cosas . . . . .   | 31         |
| 2.3.3. Computación en la niebla (Fog Computing) . . . . .                        | 33         |
| 2.3.4. Computación de Frontera (Edge Computing) . . . . .                        | 35         |
| <b>3. Sistema TLÖN</b>   | <b>36</b>  |
| 3.1. TLÖN un sistema en construcción . . . . .                                   | 36         |
| 3.2. Conceptos generales . . . . .   | 36         |
| 3.3. Modelo social inspirado . . . . .   | 40         |
| 3.4. Concepto de clúster en el modelo social-inspirado . . . . .                 | 43         |

|  |            |
|--|------------|
| <b>4. Modelo teórico de clúster para el sistema TLÖN</b>                       | <b>53</b>  |
| 4.1. Definición de clúster físico en el sistema TLÖN . . . . .                 | 53         |
| 4.1.1. Proceso de conformación de clúster físico . . . . .                     | 54         |
| 4.1.2. Modelo de clúster físico con bigrafos . . . . .                         | 56         |
| 4.2. Definición de clúster lógico en el sistema TLÖN . . . . .                 | 68         |
| 4.2.1. Proceso de conformación del clúster lógico . . . . .                    | 70         |
| 4.2.2. Modelo de clúster lógico con bigrafos . . . . .                         | 71         |
| <b>5. Mecanismo de clustering para redes ad hoc basadas en el sistema TLÖN</b> | <b>76</b>  |
| 5.1. Implementación . . . . .  | 77         |
| 5.1.1. Clúster físico . . . . .  | 77         |
| 5.1.2. Clúster lógico . . . . .  | 82         |
| <b>6. Pruebas y resultados</b>   | <b>89</b>  |
| 6.1. Comunicación intra-clúster . . . . .                                      | 90         |
| 6.1.1. Análisis del tráfico en clúster 1 . . . . .                             | 90         |
| 6.1.2. Análisis del tráfico en clúster 2 . . . . .                             | 92         |
| 6.1.3. Análisis del tráfico en clúster 3 . . . . .                             | 94         |
| 6.2. Comunicación inter-clúster . . . . .                                      | 96         |
| <b>7. Conclusiones y recomendaciones</b>                                       | <b>100</b> |
| 7.1. Conclusiones . . . . .  | 100        |
| 7.2. Recomendaciones y trabajo futuro . . . . .                                | 102        |
| <b>A. Anexo: Protocolos de enrutamiento para redes ad hoc</b>                  | <b>104</b> |
| <b>B. Anexo: Modelamiento de sistemas distribuidos</b>                         | <b>107</b> |
| B.1. Cálculo II . . . . .  | 108        |
| B.1.1. Semántica operacional . . . . .   | 111        |
| B.2. Bigrafos . . . . .  | 112        |
| B.2.1. Definiciones . . . . .  | 113        |
| B.2.2. Operaciones Básicas . . . . .   | 115        |
| Referencias . . . . .  | 118        |
| <b>Bibliografía</b>  | <b>118</b> |

# Lista de Figuras

|       |   |    |
|-------|---|----|
| 2-1.  | Diferencia entre las redes ad hoc (a) y las redes dependientes de infraestructura (b)   | 8  |
| 2-2.  | Redes ad hoc Mesh: (a) jerárquica, multisaltos y heterogénea; (b) plana y homogénea   | 11 |
| 2-3.  | Ejemplo de MANET  | 12 |
| 2-4.  | Protocolos de enrutamiento en redes ad hoc (Alotaibi y Mukherjee, 2012)   | 18 |
| 2-5.  | Esquema de enrutamiento LAR direccionado (Raw, Lobiyal, Das, y Kumar, 2015)   | 20 |
| 2-6.  | Propuestas más relevantes de algoritmos de enrutamiento en MANETs. Adaptado de (Loo, Mauri, Ortiz, y Maltz, 2016)   | 24 |
| 2-7.  | Conformación de clústeres en una red ad hoc. Tomado de Zarate(2016)   | 27 |
| 2-8.  | Clasificación de los esquemas de clustering. Adaptado de Wei y Anthony Chan (2007)  | 27 |
| 2-9.  | Comunicación inter-clúster. Tomada de (Wei y Anthony Chan, 2007)  | 28 |
| 2-10. | Arquitectura básica distribuida-centralizada de una nube móvil (Fitzek, 2014).  | 30 |
| 2-11. | Nube móvil como un conjunto de recursos distribuidos (Fitzek, 2014)   | 31 |
| 2-12. | Arquitectura básica de 3 capas de IoT   | 32 |
| 2-13. | Visión de la computación en la nube (Openfog Consortium, 2017).   | 33 |
| 2-14. | Comunicación de nodos fog en dirección Este/Oeste (Openfog Consortium, 2017).   | 34 |
|       |   |    |
| 3-1.  | Modelo de capas del Sistema TLÖN  | 37 |
| 3-2.  | Taxonomía de la social Inspiración y la bio-inspiración. Modificado de Zárate (2016)  | 41 |
| 3-3.  | Modelo Sartreano  | 43 |
| 3-4.  | Entidades territoriales para el modelo TLÖN   | 50 |
|       |   |    |
| 4-1.  | Ejemplo de dos clústeres físicos independientes   | 54 |
| 4-2.  | Formación de un clúster. Fase 1: $t_1$ activación de modo ad hoc  | 57 |
| 4-3.  | Formación de un clúster. Fase 1: transición al tiempo $t_2$ , configuración de parámetros y activación de interfaz inalámbrica $w_0$ . Transición al tiempo $t_3$ y propagación de mensajes broadcast en la región 0. | 58 |
| 4-4.  | Formación de un clúster. Fase 2: $t_4$ llegada de un nuevo nodo a las cercanías de la región 0.   | 58 |
| 4-5.  | Fase 2: $t_5$ activación del modo ad hoc de $S_1$   | 59 |
| 4-6.  | Fase 2: $t_6$ activación de la interfaz inalámbrica de $w_1$  | 59 |
| 4-7.  | Fase 3: $t_7$ vinculación de dos nodos cercanos   | 60 |
| 4-8.  | Fase 3: $t_8$ representación formal de un clúster $z$ conformado por dos nodos vecinos $S_0$ y $S_1$ .  | 60 |
| 4-9.  | Representación formal de un clúster de nombre $z$ con $n$ nodos vecinos.  | 61 |

|  |    |
|--|----|
| 4-10. Fase 5: $t_1$ un nodo <i>gateway</i> con dos tarjetas de red se aproxima al territorio del clúster $z$ .   | 62 |
| 4-11. Fase 5: $t_2$ activación del modo ad hoc de $W_{21}$ y $W_{22}$ .  | 63 |
| 4-12. Fase 5: $t_3$ un nodo <i>gateway</i> se une al clúster existente $z$ y publica la existencia de una nueva red ad hoc con identificador $q$ .                             | 63 |
| 4-13. Fase 5: $t_4$ el nodo $S_2$ invita a los nodos cercanos a vincularse a la red $q$ .  | 64 |
| 4-14. Fase 5: $t_5$ $S_2$ invita a $S_3$ a vincularse al clúster $q$ .   | 64 |
| 4-15. Establecimiento de los clústeres $z$ y $t$ mediante las interfaces inalámbricas del nodo <i>gateway</i> $S_2$ .  | 65 |
| 4-16. (a) Grafo de lugar. (b) grafo de enlace. (c) grafo de lugar de los mensajes broadcast.   | 66 |
| 4-17. Diferentes formas de superposiciones territoriales del sistema TLÖN  | 69 |
| 4-18. Clústeres lógicos conformados por agrupación de agentes locales.   | 69 |
| 4-19. Clúster lógico dentro de un nodo: fase 0 y fase 1  | 72 |
| 4-20. Clúster lógico dentro de un nodo: transición de fase 2 a fase 3.   | 72 |
| 4-21. Clúster lógico dentro de un nodo: transición a fase 4, fase 5 y fase 6   | 73 |
| 4-22. Clúster lógico distribuido: fase 0.  | 73 |
| 4-23. Clúster lógico distribuido: fase 1. Intercambio de mensajes entre los agentes $v_0$ y $v_1$ de nodos diferentes.   | 74 |
| 4-24. Clúster lógico distribuido: fase 2, establecimiento de dos clústeres lógicos distribuidos.   | 74 |
| 4-25. Clúster lógico distribuido: fase 2, representación de los lugares lógicos que comparten recursos y superposición de los clústeres lógicos $y$ y $z$ en el agente $v_0$ . | 75 |
| 5-1. Implementación de tres clústeres físicos  | 78 |
| 5-2. Opción de clustering con una estación <i>gateway</i> $GW_1$   | 81 |
| 5-3. Opciones de clustering siguiendo regla de escalabilidad   | 81 |
| 5-4. Representación de un clúster lógico $X$ en un solo nodo.  | 82 |
| 5-5. Representación de clústeres lógicos distribuidos ( $z$ y $y$ )  | 84 |
| 5-6. Módulos del mecanismo de clustering para el sistema TLÖN  | 85 |
| 5-7. Comunicación de agentes locales   | 88 |
| 6-1. Escenario de prueba clúster 1   | 91 |
| 6-2. Latencia en clúster 1   | 91 |
| 6-3. Throughput clúster 1  | 92 |
| 6-4. Escenario de prueba clúster 2   | 93 |
| 6-5. Latencia en clúster 2   | 93 |
| 6-6. Throughput en clúster 2   | 94 |
| 6-7. Escenario de prueba clúster 3   | 94 |
| 6-8. Latencia en clúster 3   | 95 |
| 6-9. Throughput en clúster 3   | 95 |
| 6-10. Escenario de comunicación inter-clúster  | 96 |
| 6-11. Latencia en comunicación entre clúster 1 y clúster 2   | 97 |
| 6-12. Latencia en comunicación entre clúster 2 y clúster 3   | 97 |

|   |     |
|---|-----|
| <b>6-13.</b> Tráfico clúster 1 durante comunicación con clúster 2 . . . . .   | 98  |
| <b>6-14.</b> Tráfico clúster 2 durante comunicación con clúster 1 y 3 . . . . .   | 99  |
| <b>6-15.</b> Tráfico clúster 3 durante comunicación con clúster 2 . . . . .   | 99  |
| <b>B-1.</b> Diferencia entre tareas concurrentes y tareas paralelas . . . . .   | 108 |
| <b>B-2.</b> Servidor de impresión (Izq. antes de la interacción, Der. Después de la interacción)<br>$(\bar{b}a.S b(c).\bar{c}d.P)\tau \rightarrow (S \bar{a}d.P)$ . . . . . | 110 |
| <b>B-3.</b> Modelo cliente servidor Cálculo II (Varela, 2013) . . . . .   | 111 |
| <b>B-4.</b> Bigrafo adaptado de (Milner, 2009) . . . . .  | 113 |
| <b>B-5.</b> Bigrafo Abstracto (Milner, 2009) . . . . .  | 114 |
| <b>B-6.</b> Bigrafo con controles (Milner, 2009) . . . . .  | 114 |
| <b>B-7.</b> Bigrafos Elementales adaptado de (Milner, 2009) . . . . .   | 116 |
| <b>B-8.</b> Ion Discreto (Milner, 2009) . . . . .   | 116 |
| <b>B-9.</b> Anatomía de un bigrafo (Milner, 2009) . . . . .   | 117 |

# Lista de Tablas

|   |    |
|---|----|
| 2-1. Protocolos de enrutamiento de energéticamente eficientes en redes ad hoc, descritos en (Rashvand y Chao, 2013). . . . .        | 21 |
| 2-2. Protocolos de enrutamiento orientados a la seguridad . . . . .   | 22 |
| 3-1. Analogía de los principios rectores del ordenamiento territorial con las características de un clúster una red ad hoc. . . . . | 49 |
| 4-1. Elementos del bigrafo . . . . .  | 57 |
| 4-2. Reglas de reacción . . . . .   | 65 |
| 4-3. Elementos del bigrafo . . . . .  | 71 |
| 5-1. Dispositivos usados en la implementación . . . . .   | 79 |
| 6-1. Configuración clúster 1 . . . . .  | 90 |
| 6-2. Configuración clúster 2 . . . . .  | 90 |
| 6-3. Configuración clúster 3 . . . . .  | 90 |
| 6-4. Medía y desviación estándar de las mediciones en clúster 1 . . . . .   | 92 |
| 6-5. Medía y desviación estándar de las mediciones en clúster 2 . . . . .   | 94 |
| 6-6. Medía y desviación estándar de las mediciones en clúster 3 . . . . .   | 96 |
| 6-7. Medía y desviación estándar de las mediciones de latencia entre los tres clústeres . .   | 98 |

# Lista de símbolos

## Abreviaturas

| Abreviatura  | Término  |
|--------------|--|
| ACP          | Algebra of Communicating Systems                 |
| AE           | Algoritmos de Enrutamiento                       |
| AL           | Agente Local                                     |
| ALFRED       | Almighty Lightweight Fact Remote Exchange Daemon |
| AODV         | Ad-Hoc On-Demand Distance Vector routing         |
| AP           | Access Point Wireless                            |
| AUV          | Autonomous Underwater Vehicle                    |
| AWS          | Amazon Web Services                              |
| B.A.T.M.A.N. | Better Approach to Mobile Ad hoc networking      |
| BRS          | Bigraphical Reactive System                      |
| BS           | Base Station                                     |
| BSA          | Basic Service Area                               |
| BSS          | Basic Service Set                                |
| BSSID        | Basic Service Set Identifier                     |
| CCS          | Calculus of Concurrent Systems                   |
| CH           | Cluster Head                                     |
| CPU          | Central Process Unit                             |
| CR           | Cognitive Radio                                  |
| CSP          | Communicating Sequential Processes               |
| DARPA        | Defense Advanced Research Projects Agency        |
| DOS          | Denial Of Service                                |
| DSR          | Dynamic Souce Routing                            |
| ESS          | Extended Service Set                             |
| ESSID        | Extended Service Set Identifier                  |
| ET           | Entidad Territorial                              |
| FC           | Fog Computing                                    |
| FG           | Forwarding Group                                 |
| FGMP         | Forwarding Group Multi-cast Protocol             |
| GAF          | Geographical Adaptive Fidelity                   |

| <b>Abreviatura</b> | <b>Término</b>  |
|--------------------|---|
| GF                 | Greedy Forwarding                                       |
| GPS                | Global Positioning System                               |
| GRAA               | Geographic Routing Protocol for Aircraft Ad hoc Network |
| IBSS               | Independent Basic Service Set                           |
| ID                 | Identifier  |
| IEC                | International Electrotechnical Commission               |
| IEEE               | Institute of Electrical and Electronics Engineers       |
| IoT                | Internet of Things                                      |
| IP                 | Internet Protocol                                       |
| IPS                | Intrusion Prevention System                             |
| IPv4               | Internet Protocol Versión 4                             |
| ISO                | International Organization for Standardization          |
| ITU                | International Telecommunication Union                   |
| ITU-T              | ITU - Telecommunication Standardization Sector          |
| LAN                | Local Area Network                                      |
| LAR                | Location-Aided Routing                                  |
| LGPL               | Lesser General Public License Versión 2                 |
| LLC                | Logical Link Control                                    |
| LLDP               | Link Layer Discovery Protocol                           |
| LTE                | Long Term Evolution                                     |
| LoRa               | Long Range  |
| MA                 | Mobile Agent  |
| MAC                | Media Access Control                                    |
| MANET              | Mobile Ad hoc network                                   |
| MAODV              | Multi-cast Ad-Hoc On-Demand Distance Vector routing     |
| MAS                | Multi Agent System                                      |
| MCEDAR             | Multicast Core-Extraction Distributed Ad hoc Routing    |
| MPR                | Multipoint Relays                                       |
| MT                 | Máquina de Turing                                       |
| MTU                | Maximum Transmission Unit                               |
| NB-IoT             | Narrowband IoT by 3gpp                                  |
| OGM                | Originator Message B.A.T.M.A.N.                         |
| OLSR               | Optimized Link State Routing                            |
| ONU                | Organization for United Nations                         |
| OSI                | Open System Interconnection                             |
| P2P                | Point to Point  |
| PC                 | Personal computer                                       |
| PRBRS              | Priority Stochastic Bigraphical Systems with sharing    |
| RAM                | Random Access Memory                                    |

---

| <b>Abreviatura</b> | <b>Término</b>  |
|--------------------|---|
| RERR               | Request Error   |
| RFC                | Request For Comments                                    |
| RREP               | Route Request Response                                  |
| RREQ               | Route Request   |
| S.O.V.O.R.A.       | Sistema Operativo Virtualizado Orientado a Redes Ad hoc |
| SO                 | Sistema Operativo                                       |
| SOP                | Spectrum opportunity                                    |
| STA                | Wireless Station  |
| TCP                | Transmission Control Protocol                           |
| UAVs               | Unmanned Aerial Vehicles                                |
| UDP                | User Datagram Protocol                                  |
| UUID               | Universal Unique Identifier                             |
| V2I                | Vehicle to Infrastructure                               |
| V2V                | Vehicle to Vehicle                                      |
| VANET              | Vehicular Ad hoc Networks                               |
| VM                 | Virtual Machine   |
| WSN                | Wireless Sensor Networks                                |



# 1. Introducción

El termino clúster es un anglicismo utilizado en diversos ámbitos y áreas del conocimiento. Por esta razón puede ser definido de diferentes formas según el propósito y el entorno en el que se concibe (Xu y Wunsch, 2009), no obstante, en general la palabra clúster en inglés se refiere a un grupo de cosas o elementos similares.

En ciencias de la computación, por ejemplo, un clúster de computadores, se entiende como un conjunto de equipos o recursos de cómputo comunicados que trabajan juntos para cumplir una misma tarea. Este es ampliamente usado para propósitos específicos como en investigación intensiva, donde se necesitan grandes capacidades de procesamiento que no pueden ser brindadas por una sola máquina. Existen también los clústeres de bases de datos en los que se distribuye gran cantidad de información entre el grupo de máquinas que los componen, permitiendo, por ejemplo, el procesamiento de consultas de forma paralela (Jin y Buyya, 2001). Existen también los clústeres de servicios, que permiten administrar de forma distribuida las aplicaciones o servicios en un ámbito determinado.

Por otra parte, existen los clústeres de datos que están relacionados con el análisis de información (o minería de datos, derivado del inglés *data mining*) contenida en grandes bases de datos. En este caso, el termino clustering se refiere a la agrupación de datos de acuerdo con un criterio, comúnmente, este criterio se basa en la distancia (cercanía) o la similitud que bien puede ser una función de la distancia, por ejemplo, la distancia Euclidiana (Patel, 2003).

Debido a que existen múltiples entornos en los que se puede aplicar el termino clúster, es importante diferenciar que en el área de redes de telecomunicaciones, por lo general, este termino no se refiere a la agrupación, análisis y procesamiento de la información pura en el nivel de datos, sino a un agrupamiento de entidades que trabajan de forma paralela para ejecutar una o más tareas de forma distribuida en capas inferiores como la de red, transporte y físico. En este trabajo, se usará el término clúster para referirse al agrupamiento de un conjunto de recursos físicos o lógicos dentro de una red de comunicaciones.

La mecánica detrás de la elección de nodos o entidades para establecer un clúster es un algoritmo de clustering (Patel, 2003), también llamado *clustering análisis* (Xu y Wunsch, 2009) o simplemente clustering.

---

EL Concepto redes ad hoc fue desarrollado por la Agencia de Proyectos de Investigación Avanzados de Defensa (DARPA, por sus siglas en inglés) de Estados Unidos en 1970, para nombrar a las tecnologías de redes que operan sobre medios no cableados, establecidas de forma temporal para ofrecer servicios o aplicaciones específicas. Principalmente, se usan para propósitos de rescate después de un desastre o durante operaciones militares en un campo de batalla (Mohapatra y Krishnamurty, 2005), donde una infraestructura de red es imposible de establecer en un tiempo corto o es muy costosa. (Kumar, Dave, y Kuruksheeta, 2010)

Una red ad hoc es un sistema de comunicación auto-organizante, carece de infraestructura física predefinida, pues esta es creada y mantenida de forma dinámica. Esto quiere decir, que se puede implementar rápidamente sin necesidad de una planificación previa, por esa razón, las redes ad hoc también se conocen como redes de propósito específico.

En una red ad hoc, clustering es entendido como el proceso de división de la red en varias subredes, en donde se agrupan nodos y cada grupo es manejado por un nodo director o también llamado *Cluster Head* (Barbeau y Kranakis, 2007) que actúa como un controlador local. Los diferentes mecanismos o reglas para armar un clúster y elegir el *Cluster Head* en una red ad hoc son temas potenciales de investigación.

El sistema TLÖN es un modelo experimental concebido en el Grupo de Investigación en Lenguajes de Programación Distribuidos y Redes de Telecomunicaciones Dinámicas - TLÖN, para proponer soluciones a los desafíos que se presentan en la implementación de redes ad hoc en ambientes móviles, estocásticos y altamente distribuidos. Es un modelo social inspirado, esto significa que define el todo y sus partes basándose en la analogía de los modelos sociales. Así, las características de todos los elementos en el mundo artificial de TLÖN son abstracciones de conceptos asociados a la naturaleza de una sociedad. Siguiendo este precepto, se han elegido las nociones de Justicia de Jhon Rawls, Inmanencia de Baruch de Spinoza, Paradigma de Thomas Kuhn, así como también el concepto de Estado de Thomas Hobbes y las concepciones de existencia y esencia de Jean Paul Sartre (Zárate, 2016) para definir su esquema de computación.

TLÖN persigue el objetivo de encontrar una forma de implementar artificialmente los modelos sociales en un sistema de cómputo distribuido, de topología de red altamente dinámica, por ser cambiante y transitoria, que permita la prestación de múltiples servicios especializados de computación distribuida sin ser afectados por la característica ad hoc de sus nodos, tampoco por sus recursos limitados de procesamiento o almacenamiento ni por las características heterogéneas de los mismos. Para abordar este problema, se busca diseñar e implementar los módulos del sistema TLÖN para que cumplan tareas análogas a las de un Estado, es por eso que TLÖN define un pseudo-Estado TLÖN. Este trabajo propone el concepto formal de clúster dentro del modelo social-inspirado del sistema TLÖN.

Las dimensiones del sistema TLÖN se representan en un modelo de capas<sup>1</sup>, donde el territorio es equivalente a la red ad hoc o ambiente de red, que se ubica en la primera capa denominada Capa de Infraestructura. Las instituciones las conformaría el Módulo de Virtualización, que corresponde a la segunda capa, esta se encarga de los aspectos relacionados con la virtualización inalámbrica y el Sistema Operativo (SO). La tercera capa hace referencia al Sistema Multiagente, donde operarán operan las comunidades de agentes, de manera análoga a las personas o actores de la sociedad, que en el plano real se pueden agrupar en sociedades y conformar comunidades, estos agentes podrán trabajar en conjunto para proveer servicios en toda la red ad hoc. Finalmente, encontramos la Capa de Aplicaciones específicas del sistema de cómputo. De manera transversal a todas las capas están los componentes de Lenguaje del Sistema y su propia Ontología, que actúan como habilitadores de las interacciones en todas las capas.

## 1.1. Justificación

En una red ad hoc que carece de infraestructura predefinida, tiene topología cambiante, es auto-organizable y auto recuperable es importante estudiar los mecanismos para formar clústeres que permitan un buen comportamiento de la red ante los cambios causados por el movimiento de los nodos, por las fallas en los mismos o por la entrada o salida de nodos de la red. A su vez la implementación de mecanismos de clustering en una red ad hoc puede atacar objetivos importantes como la seguridad, la calidad del servicio, eficiencia energética y escalabilidad.

TLÖN implícitamente ha establecido la necesidad de desplegar un mecanismo de clustering de recursos físicos y lógicos al definir en su modelo de pseudo-Estado a las entidades de virtualización, dado que el concepto de virtualización esta fuertemente ligado al concepto de computación en la nube, que por definición implica un “agrupamiento” o “clustering” de recursos.

Muchos de los problemas que se presentan en la implementación de un sistema dinámico como el sistema TLÖN se derivan de una previsión en el aumento del número de usuarios de aplicaciones y servicios especializados. Para que este se adapte a las exigencias de la tecnología actual y a la vez sea escalable, se visualiza que es necesario el establecimiento de entidades tipo clúster dentro de una ad hoc, porque se considera que aportarían una mejora significativa para el objetivo de implementar y gestionar recursos y servicios de nube de una forma distribuida.

El estudio de este tipo de problemas concuerda con las necesidades emergentes de investigar mecanismos para abordar los desafíos de congestión, retardo, latencia y privacidad derivadas del auge de comunicaciones móviles y la computación en la nube. Esto esta unido a una elevada tasa de crecimiento en el número de dispositivos inteligentes IoT con capacidades de interconexión a redes que demandan esfuerzo en el diseño de nuevas estrategias para proveer mejores y más eficientes capaci-

---

<sup>1</sup>ver: <http://www.tlon.unal.edu.co/proyecto-tlon/acerca>

dades de computo en ambientes dinámicos.

Respecto al trabajo realizado previamente por los integrantes del grupo de investigación TLÖN, se observa, por ejemplo, que en Latorre (2015) los esfuerzos se centraron en la implementación de una sola red ad hoc y el establecimiento de comunicación dinámica entre los nodos de la misma. En Zárate (2016) se diseñó el Sistema Operativo Orientado a Redes Ad hoc (S.O.V.O.R.A) y se implementó uno de los componentes principales del modelo TLÖN: el orquestador, encargado de iniciar la red, gestionar recursos de cómputo y ser la base para la prestación de servicios dentro de la misma. Por otra parte Álvarez Rodríguez (2017) diseñó e implementó el modelo de agentes móviles TLÖN y su ambiente dentro de la red, con la capacidad de desplegarse y realizar tareas de acuerdo con roles definidos. Sin embargo, aún no se han realizado implementaciones de comunicación entre clústeres dentro de una red ad hoc, tampoco entre dos o más redes ad hoc diferentes y se identifica que aún falta plantear una definición formal de clúster en el contexto del sistema TLÖN, que proporcione lineamientos para el diseño de un mecanismo de clustering que permita la comunicación a nivel físico entre varias redes ad hoc. A nivel intangible también es importante la definición de clúster lógico o “nube” que permita agrupar los recursos distribuidos de los nodos con el fin de ejecutar procesos o tareas específicas y de esta forma aproximar el modelo experimental de TLÖN a los paradigmas de IoT y computación en la niebla (FC).

Este trabajo también es pertinente porque el sistema TLÖN aún se encuentra en construcción, y por eso ofrece temas de investigación potenciales que aún no se han explorado completamente en todas las capas del sistema. La definición de un mecanismo de clustering es uno de ellos.

## 1.2. Identificación del Problema

En los nuevos paradigmas de *Edge Computing* y *Fog Computing* se presentan desafíos abiertos a la investigación, que concuerdan con los que surgen en la implementación del sistema TLÖN. Generalmente, en este tipo de sistemas los equipos finales se conectan por redes inalámbricas donde la probabilidad de falla es alta, entre otras causas, por la naturaleza del medio de transmisión impredecible e inestable y en ocasiones no confiable. La conexión por vía inalámbrica implica que hay que tener en cuenta la movilidad de los nodos, ya que estos pueden cambiar de clúster de un momento a otro y esto no debe afectar la operación global del sistema. Por otra parte, pueden ingresar muchos nodos a conformar una red o clúster o pueden retirarse en cualquier momento, por ejemplo, en el caso de que su batería se agote. Estos son algunos de los factores que hacen que se preste especial atención a la compleja naturaleza de las fallas (Naha y cols., 2018) durante el diseño de este tipo de sistemas y, en concordancia con esto, la formación de clústeres y la formulación de leyes del pseudo-Estado TLÖN que definan el funcionamiento de los mismos, son herramientas para darle al sistema la característica de escalabilidad.

La importancia del estudio de clustering en sistemas altamente distribuidos se hace más evidente cuando se plantea el problema de establecer comunicación entre dos o más grupos de recursos de diferentes redes. En este sentido, surgen diferentes preguntas que motivan la investigación: ¿Cómo establecer comunicación entre grupos de nodos pertenecientes a diferentes redes dinámicas?, ¿Qué es exactamente clúster físico y clúster lógico en una red ad hoc basada en el sistema TLÖN?, ¿Qué reglas o criterios se deben tener en cuenta para agrupar los nodos? ¿Cómo implementar el comportamiento deseado del clúster?, ¿Cómo diseñar mecanismos de comunicación intra e inter-cluster? y ¿Qué comportamiento, característica o entidad de una sociedad se puede usar para abstraer el concepto de clúster dentro del sistema TLÖN?

## 1.3. Objetivos

### 1.3.1. Objetivo General

Diseñar un mecanismo de clustering en redes ad hoc para sistemas distribuidos en el contexto del sistema TLÖN.

### 1.3.2. Objetivos Específicos

- Definir el modelo teórico de clúster físico y lógico en una red ad hoc basada en sistema TLÖN.
- Definir el mecanismo de clustering físico y lógico para redes ad hoc basadas en sistema TLÖN.
- Implementar el mecanismo para el funcionamiento de clústeres físicos y lógicos en una red ad-hoc basada en sistema TLÖN.
- Validar el mecanismo de clustering en dos escenarios de prueba.

## 1.4. Sobre este trabajo

El presente documento se estructura de la siguiente manera, en el capítulo 1 se presenta la introducción a este trabajo, la justificación y objetivos. En el capítulo 2 se abordan los fundamentos de las redes ad hoc y algunos conceptos básicos como computación en la nube, IoT y computación en la niebla. En el capítulo 3 se describe el sistema TLÖN, el modelo social inspirado y se establece la definición de clúster usando la analogía del concepto de territorio. En el capítulo 4 se describe el modelo teórico de clúster para el sistema TLÖN usando el paradigma de Bigrafos del profesor Robert Milner (Milner, 2009) para modelar interacciones en sistemas distribuidos móviles. En el capítulo 5 se describe el mecanismo de clustering a nivel físico y lógico y se describe la implementación del experimento base para la realización de las pruebas. En el capítulo 6 se describen las pruebas realizadas para la

validación del mecanismo de clustering para redes ad hoc. Finalmente, en el capítulo 7 se exponen las conclusiones, recomendaciones y posibilidades de trabajo a futuro.

## 2. Fundamentos de redes ad hoc

Una red ad hoc difiere de las redes convencionales, dependientes de una infraestructura de red pre-establecida, como las redes cableadas o las redes de telefonía móvil, precisamente porque no requieren dispositivos intermedios ni redes de gran capacidad para establecer conexión entre los nodos participantes. Se configuran de forma rápida, su topología es dinámica y tienen un comportamiento estocástico. La Figura 2-1 ilustra el concepto de red ad hoc en contraste con las redes de modo infraestructura.

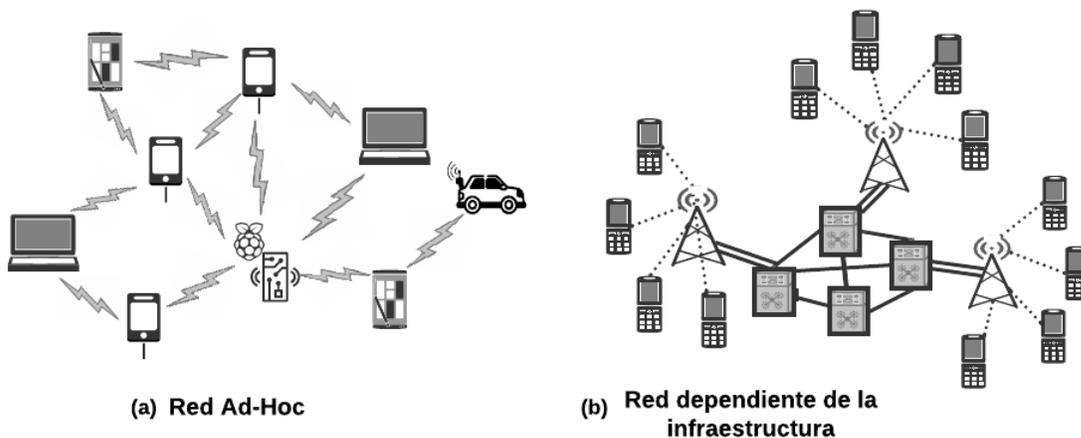


Figura 2-1.: Diferencia entre las redes ad hoc (a) y las redes dependientes de infraestructura (b)

### 2.1. Redes ad hoc

Ad hoc es una locución latina que se refiere a una solución específica elaborada para un problema particular y por lo tanto no es generalizable ni utilizable para otros propósitos. Es usada para referirse a algo que es adecuado para un fin determinado o para una situación particular (RAE, s.f.).

De modo que una red ad hoc es un tipo de red que se arma de manera improvisada y espontánea para un propósito específico, es un sistema de comunicación distribuido y auto-organizante, cuya infraestructura es creada y mantenida de forma dinámica. También se puede definir como un conjunto de dispositivos inalámbricos, que tienen libertad para entrar y salir de la red o capacidad moverse y auto-configurarse rápidamente para armar temporalmente una estructura no centralizada (Barbeau y Kranakis, 2007).

Esto significa que a diferencia de las redes convencionales de infraestructura, los nodos no tienen roles preestablecidos, pues cada nodo puede actuar como enrutador o como punto final y no existe un control o administración centralizado. Los nodos se conectan directamente unos con otros sin requerir estrictamente la intervención de alguna clase nodo central, bien sea router, *access point* (AP) o estación base. A pesar de esto, su ausencia en una red ad hoc no es imperativa pues este tipo de nodos podrían configurarse de cierta forma especial para hacer parte de una red ad hoc.

En una red ad hoc, los paquetes enviados desde un nodo origen deben pasar por cierto número de nodos vecinos, saltando de un nodo a otro hasta alcanzar el destino. Por esta razón, se dice que una red ad hoc tiene la característica de multisaltos o *multihop*. Esta es una ventaja que permite un fácil despliegue de la red, pero a su vez representa un reto para el diseño de protocolos de enrutamiento que permitan mantener la conectividad y la disponibilidad de la red aún cuando la topología cambie de forma rápida e impredecible.

### 2.1.1. Tipos de redes ad hoc

Las redes ad hoc pueden establecerse de diferentes formas y configurarse según su aplicación. Algunas de las formas más relevantes en las que se pueden clasificar son: según sus características de topología, rango de comunicación, configuración hardware de los nodos y según su dependencia de infraestructura.

**Según su topología, podemos encontrar redes ad hoc planas y jerárquicas.**

- **Planas**

En las redes planas o *flat* todos los nodos tienen las mismas funciones, de actuar como enrutadores y clientes. Ninguno tiene funciones especiales de comunicación o administración, todos tienen la misma jerarquía.

- **Jerárquicas**

Las redes ad hoc jerárquicas, como su nombre lo indica, establecen diferencias entre los nodos. Así, ciertos nodos cumplirán funciones especiales como ser el director o líder de un grupo de nodos, que serán identificados en la red como *Cluster Head*, y otros podrán desempeñar el rol de puerta de enlace hacia otros grupos de nodos, estos son los nodos Gateway. Conforme a esto, en la red se establecerán niveles de jerarquía.

### Según su rango de comunicación, encontramos redes ad hoc de un salto único (SingleHop) o multisaltos (Multihop)

- **Salto único**

En estas redes los nodos están al alcance directo uno de otro, a un solo salto (single-hop) y se mueven en el mismo rango de alcance, por lo tanto, para el intercambio de paquetes no requieren nodos intermedios.

- **Multisaltos**

En las redes ad hoc multisaltos o Multihop, los nodos no están en el rango de transmisión, no tienen alcance directo unos de otros, por lo tanto, existen nodos intermedios que colaboran en el reenvío de los mensajes.

### Según la configuración hardware de los nodos, las redes ad hoc pueden ser heterogéneas u homogéneas.

- **Heterogéneas**

Los nodos participantes tienen características hardware diferentes, también diferentes capacidades de memoria, procesamiento y espacio de disco e incluso sistemas operativos y aplicaciones diferentes.

- **Homogéneas**

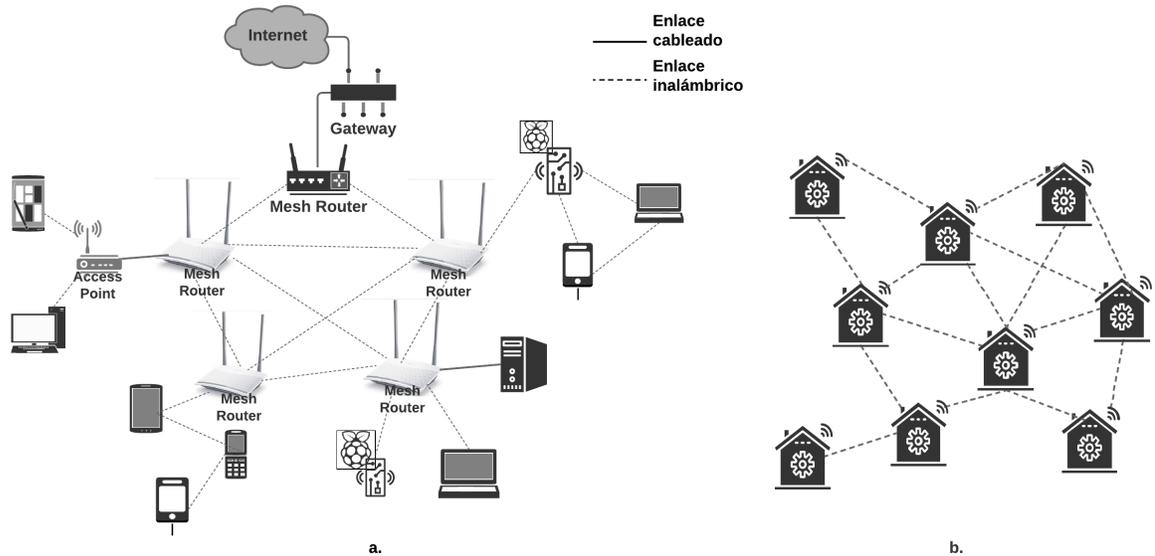
En estas redes, los nodos tienen las mismas características de hardware y software. En general presentan un mismo comportamiento y cada nodo tiene capacidad potencial para ofrecer los mismos servicios.

### Según su grado de dependencia de la infraestructura física, se clasifican en redes *Mesh* y redes ad hoc móviles.

- **Redes ad hoc Mesh**

Las redes ad hoc en malla se conocen comúnmente como redes *Mesh* y así se las denominará de ahora en adelante. Una red ad hoc de malla o *Mesh*, tiene una parte de la red que es fija, con un grado alto de permanencia en la red. Esa parte la conforman nodos que tienen una nula o poca movilidad. Muchas veces estos son equipos de mayor capacidad que se conocen como *Routers Mesh* o enrutadores de malla. Dos ejemplos de este tipo de redes se observan en la Figura 2-2.

La Figura 2-2 a. muestra un ejemplo de red ad hoc Mesh, jerárquica, multisaltos y heterogénea, donde los *Routers Mesh* se comunican armando una capa más alta de jerarquía para proveer comunicación a diferentes dispositivos móviles y no móviles. La figura 2-2 b. ilustra un ejemplo de una red de sensores ubicados en casas vecinas que se comunican de forma ad hoc para conformar una red ad hoc Mesh, plana y homogénea.



**Figura 2-2.:** Redes ad hoc Mesh: (a) jerárquica, multisaltos y heterogénea; (b) plana y homogénea

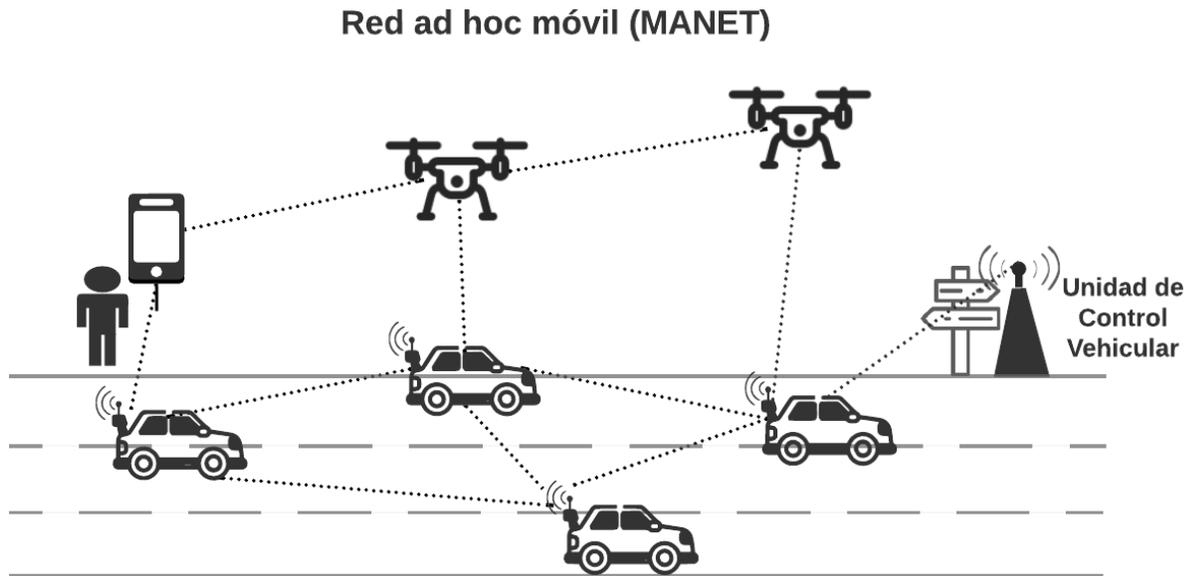
#### ■ Redes ad hoc móviles (MANET)

Las redes ad hoc móviles, o *Mobile Ad Hoc Networks* (MANETs). Por el contrario, están conformadas por dispositivos inalámbricos con un alto grado de movilidad y no hacen uso de nodos estáticos para interconectarse.

En la Figura 2-3 se plasma la idea de una MANET heterogénea y multisaltos, donde vehículos en la carretera intercambian información entre sí. También solicitan servicios cuando pasan por una unidad de control vehicular y en el aire drones que vigilan el tráfico reciben señales de los vehículos y a su vez pueden servir de puente para que una persona con un dispositivo móvil se comunique con otros vehículos.

#### Redes de radio cognitiva

Son otro tipo de redes ad hoc inalámbricas que implementan un acceso dinámico al espectro. Las redes ad hoc convencionales (no cognitivas), utilizan para todas sus transmisiones las mismas frecuencias del espectro en la banda de los 2.4Ghz o 5Ghz de acuerdo al estándar para redes inalámbricas IEEE802.11. La Radio Cognitiva o *Cognitive Radio* es una estrategia de transmisión, en la que los nodos tienen la capacidad de escanear el espectro de radiofrecuencias y adaptar su transmisión según las oportunidades de espectro disponible. Estas oportunidades son un conjunto de frecuencias que no están siendo ocupadas por usuarios primarios (con licencia) (Rashvand y Chao, 2013), en el periodo de tiempo en que están inactivos y, por lo tanto, dejan libre espacios en el espectro que los usuarios secundarios (sin licencia) aprovechan. Cuando el usuario primario vuelve a estar activo entonces inmediatamente el secundario abandonará los canales ocupados, porque la prioridad de ocupación del espectro la tienen los usuarios licenciados. Estas redes se conocen como redes ad hoc oportunistas.



**Figura 2-3.: Ejemplo de MANET**

### 2.1.2. Aplicaciones de las redes ad hoc

Gracias al creciente uso de dispositivos móviles y en general de equipos con capacidad de conectarse por una interfaz inalámbrica, las redes ad hoc pueden ser usadas en diferentes contextos, algunas de sus aplicaciones potenciales, son:

- **Aplicaciones militares:** Para proveer comunicación en ambientes hostiles donde no existe infraestructura de red y se requiere establecer comunicación de forma rápida entre personas en un campo de batalla o, por ejemplo, para coordinar vehículos militares no tripulados UAVs (Unmanned Aerial Vehicles), permitiendo que estos puedan enlazarse para formar grupos, como en el caso de los drones para habilitar la capacidad de volar en formación, realizar misiones conjuntas y reportar información a las unidades en tierra (Basagni, Conti, Giordano, y Stojmenović, 2013).
- **Recuperación en caso de desastre:** Para establecer redes después de un colapso en la infraestructura de red tradicional debido a desastres naturales o eventos catastróficos en donde se requiere establecer comunicación para atender emergencias y salvar vidas.
- **Redes de sensores:** también llamadas *Wireless Sensor Networks (WSN)*, consisten en el despliegue de dispositivos de bajo costo y baja potencia para capturar información en un ambiente, que también pueden ser transportarlos por seres humanos, animales u objetos para detectar las actividades que realizan o para rastrearlos. Mediante el uso de detectores de movimiento, temperatura o sensores de luz y actuadores, las WSN han abierto la posibilidad de implementar modernas aplicaciones como hogares inteligentes (*Smart Homes*) y salud Inteligente (*Smart*

*Health*). Las primeras permiten hacer uso eficiente de los dispositivos inteligentes del hogar, por ejemplo, para apagar equipos que no están siendo usados, como la calefacción. Por otra parte, los sistemas para *Smart Health* permiten aplicaciones que van más allá del monitoreo de signos vitales de una persona. Por ejemplo, en el área de la epidemiología se puede monitorear la propagación de una enfermedad o virus usando dispositivos móviles, para que sea más fácil obtener datos sobre la diseminación. Las aplicaciones que manejan información del estado de salud de un usuario también pueden detectar si una persona infectada está cerca de otra. De igual forma, por su característica inalámbrica son propicias para monitorear condiciones del medio ambiente, en la selva o bosques, donde no existe infraestructura de red cableada convencional (Basagni y cols., 2013).

- **Redes rurales:** en países en vías de desarrollo que carecen de infraestructura de redes, habilitan la interconectividad entre personas, comunidades y amplían a un bajo costo el área de servicio de internet en las instituciones educativas.
- **Redes para campus:** se entiende el campus, como un lugar donde se aglomeran personas para propósitos culturales, sociales y educativos. Allí las redes ad-hoc juegan un papel importante para activar el acceso e intercambio de información en todos los rincones de un campus, expandiendo las redes existentes para cubrir todas las áreas por donde se movilizan las personas, disminuyendo los huecos en los que los usuarios no tienen acceso a servicios como internet (Mohapatra y Krishnamurty, 2005). En un campus universitario, por ejemplo, las redes ad hoc proveerían a su comunidad capacidades de interconexión punto a punto en las salas de conferencia, en los salones o laboratorios, permitiendo transferencia de archivos e incluso jugar en red.
- **Comunicaciones vehiculares:** Denominadas redes ad hoc vehiculares o Vehicular Ad hoc Networks<sup>1</sup> (VANETs) proveen comunicaciones V2V entre vehículos cercanos equipados con transceptores inalámbricos o comunicación V2I con la infraestructura fija de control ubicada a un lado de la carretera (Basagni y cols., 2013), con el propósito de proveer conectividad en todos los lugares mientras los vehículos se mueven de un lugar a otro. Mediante esta tecnología es posible la implementación de sistemas de transporte inteligentes con variedad de aplicaciones como: control y monitoreo cooperativo de tráfico, control de servicios de transporte, prevención de choques, accidentes y cálculo de desvíos en las rutas (Kumar y cols., 2010). Comúnmente, se implementan en vehículos terrestres, pero su uso se ha extendido a aviones, helicópteros, barcos e incluso a vehículos autónomos subacuáticos (AUVs) (Salvá-Garau y Stojanovic, 2003).

<sup>1</sup>Para las especificaciones de las VANETs la IEEE ha definido el estándar 802.11p

### 2.1.3. Enrutamiento en redes ad hoc

Desde la concepción de las MANETs, un gran número de protocolos de enrutamiento han sido diseñados para proveer la característica de multisaltos en este tipo de redes (ver la lista de protocolos de enrutamiento para redes ad hoc en el Anexo A). Por lo tanto, existe una gran variedad de propuestas que van desde modificaciones del protocolo IP hasta esquemas complejos de enrutamiento jerárquico multinivel. Por lo general, estos protocolos se diferencian según el problema en el que se enfocan, de ahí la importancia de entender los retos que implica el diseño de soluciones de enrutamiento para redes ad hoc.

Los siguientes son los principales objetivos que se persiguen durante el diseño de protocolos para este tipo de redes (Belding-Royer, Conti, Giordano, y Stojmenovic, 2005):

- Minimizar la sobrecarga (*overhead*) de mensajes de control, debido a que el tráfico consume recursos de ancho de banda, procesamiento y batería. Este tipo de mensajes deben ser reducidos para no sobrecargar la red y disminuir el consumo de energía de los dispositivos finales.
- Los algoritmos deben ser livianos y no deben exigir ciclos exhaustivos de procesamiento. De esta forma, es posible extender el tiempo de batería de los nodos. La eficiencia en el uso de la batería de los nodos es un aspecto crítico por eso se han planteado algoritmos y mecanismos conscientes de la energía<sup>2</sup> que atacan este problema (Akhtar y Wang, 2016).
- Habilitar enrutamiento multisaltos, que permitan descubrir rutas entre el origen y destino teniendo en cuenta los rangos de transmisión inalámbrica. Considerando que por lo general los nodos no están conectados a una fuente de energía, entonces la potencia de transmisión está limitada a un rango máximo. De esta forma en las redes ad hoc es probable que los nodos no puedan conectarse directamente con otros nodos muy distantes en la misma red. De ahí surge la conveniencia de aplicar el principio de multisaltos (*multihopping*).
- Mantener la topología dinámica, ya que debido al alto grado de dinamismo de la red, los enlaces entre nodos pueden romperse en cualquier momento cuando los nodos se mueven. Pero ante estos cambios de topología, el protocolo debe ser capaz de reaccionar y mantener un camino para la conexión entre origen y destino
- Prevenir loops o bucles entre rutas, que pueden suceder cuando un nodo reenvía paquetes a otro nodo que ya ha participado en el enrutamiento de este mismo paquete, o que era el nodo fuente desde el que se estaban enviando los paquetes. En consecuencia se genera un desgaste de recursos.

---

<sup>2</sup>En las fuentes bibliográficas, se refieren a este tipo de algoritmos con las palabras *Energy aware* o *Energy efficient*

### 2.1.4. Clasificación de los algoritmos de enrutamiento en redes ad hoc

Los Algoritmos de Enrutamiento (AE) pueden clasificarse de varias formas. Dos de las principales son:

**Centralizados/Descentralizados:** cuando son centralizados un solo nodo central realiza el cálculo de las tablas de enrutamiento de cada uno de los nodos de la red y posteriormente envía la información de enrutamiento al nodo respectivo. Esta no es una buena opción para las redes ad hoc dado que son descentralizadas; es decir, cada nodo ejecuta tareas de enrutamiento y el cálculo de las rutas no se delega a un solo nodo.

**Proactivos/Reactivos:** en los protocolos proactivos los nodos siempre mantienen disponible la tabla de enrutamiento hacia los otros nodos y la recalculan durante periodos de tiempo usando diferentes mecanismos. Son óptimos para redes cableadas donde la topología casi no cambia y los recursos abundantes permiten mantener los grafos de conectividad. En los reactivos, los nodos no mantienen disponible en todo momento las rutas hacia todos los nodos vecinos. En cambio, el cálculo de rutas solo se realiza cuando se necesita, entonces se dice que el descubrimiento de rutas se realiza por demanda. Aunque los últimos presentan latencia en la adquisición de la ruta hacia un destino, gastan menos recursos computacionales en el cálculo constante de las tablas de enrutamiento, en consecuencia, son más adecuados para las redes ad hoc.

**AE Híbridos:** considerando que algunos AE presentan características similares, estos se pueden ubicar en dos o más categorías y se les denomina AE híbridos. En general, se denomina algoritmos híbridos a los que combinan las dos características: reactivo y proactivo.

Algunas formas de describir y clasificar los AE están determinadas por ciertos mecanismos o estrategias que usan. Por consiguiente, es importante familiarizarse con los conceptos especializados que definen aquellas categorías. Se describen los más conocidos a continuación.

#### Modos de difusión usados por los protocolos de enrutamiento

En redes de comunicaciones, el sufijo cast significa difusión. Existen tres clases principales de difusión de los paquetes:

**Unicast:** Difusión desde un origen a solo un único destino.

**Multicast:** Difusión desde un origen hacia un grupo definido de destinos.

**Broadcast:** Difusión amplia, desde un origen hacia toda la red. Cuando se produce una transmisión broadcast, todos los nodos presentes en la red “escuchan” el mensaje.

#### Tres estrategias comúnmente usadas por los AE

**Single path** o ruta única, donde una sola copia del mensaje viaja por la red desde origen a destino. Representa una relación uno a uno (1:1) entre ruta y destino, es decir, en la tabla de enrutamiento para cada destino se tiene una sola entrada con el siguiente salto.

**Multipath** o multi-trayectoria, donde un número pequeño de copias del mensaje original es transmitido por diferentes rutas desde la fuente al destino. Representa una relación muchos a uno (\*:1) entre ruta y destino. De esta forma, en la tabla de enrutamiento para un destino existen muchas entradas que indican el siguiente salto.

**Flooding**<sup>3</sup> con este mecanismo se crean muchas copias del mensaje original, que cada nodo de la red retransmite por difusión broadcast a cada uno de sus vecinos, sin importar si es al mismo nodo que originó la transmisión. El proceso se repite progresivamente en la red hasta que el paquete alcance su destino.

### Otras formas de clasificación de los algoritmos de enrutamiento en redes ad hoc

La clasificación mencionada anteriormente no está cerrada porque ubicar a los algoritmos en una sola categoría es una tarea compleja considerando que cada algoritmo cuenta con muchas características. Además, porque existen múltiples opciones y combinaciones de técnicas para realizar el enrutamiento, y en consecuencia, los enfoques de cada algoritmo dependen del caso de uso.

**AE Multicasting:** los algoritmos de multidifusión o AE multicast, como se referirá a ellos en adelante, hacen uso de la técnica de multicast para dispersar los paquetes en la red. Estos pueden ser de dos tipos: basados en árbol (*Tree-based*) y basados en malla (*mesh-based*).

*Tree-based:* consiste en establecer un único camino desde el emisor al receptor. Se construye una única ruta de reenvío de datos con forma de árbol, cuya raíz es el nodo origen de la sesión de multidifusión. Una sesión de multidifusión se refiere a todo el proceso de construcción de este árbol y de los eventos generados para la difusión de paquetes, administración de membresía de nodos a grupo y la identificación de los grupos de nodos que intervienen en ese proceso hasta su finalización (Medhi y Ramasamy, 2018).

*Mesh-based:* en estos, múltiples rutas son generadas, entonces pueden coexistir muchos caminos de difusión entre una dupla fuente y destino.

Durante los procesos de multicast se presentan variedad de situaciones, de modo que puede darse el caso en que múltiples sesiones multicast se ejecutan en paralelo en la red y por lo tanto pueden compartir partes de un árbol de reenvío de datos. Este proceso se llama *shared tree* o árbol compartido. Los algoritmos de multicast difieren entre sí en la forma en que se tramita el despacho de mensajes multicast o en el mecanismo para agrupar los nodos receptores durante la sesión multicast. Incluso pueden implementar jerarquías designando un líder del grupo multicast, *Multicast Group Líder*, que se encarga de mantener el número de secuencia de los reenvíos y enviar este número a su grupo (Ilyas, 2003).

---

<sup>3</sup>Su traducción literal es inundación. En redes de telecomunicaciones, se refiere a *inundar* la red con paquetes. Se usará el término en inglés debido su ambigüedad y descontextualización al ser usado en Español.

Otro aspecto en el que pueden diferenciarse los protocolos multicast es la manera en que implementan las dinámicas de grupo y las dinámicas de red. FGMP (*Forwarding Group Multi-cast Protocol*) es un ejemplo de algoritmo que no rastrea los enlaces sino los grupos de nodos que participan en la retransmisión multicast de paquetes. En este un grupo multicast de nodos es asociado a un grupo de re-transmisores llamado Forwarding Group (FG). Los nodos pertenecientes a este grupo son los únicos encargados de retransmitir un paquete multicast y aunque todos los demás nodos también lo reciben, solo los que están en el FG realizan un análisis y si el paquete recibido es un duplicado no lo retransmiten (Mohapatra y Krishnamurty, 2005) o al contrario, si detectan que no lo es.

Las redes ad hoc imponen serios problemas a los protocolos multicast, comúnmente pensados y diseñados para redes IP, o redes cableadas, que no tienen el mismo grado de dinamismo y donde los grupos de difusión no cambian en el tiempo. El problema en este caso es que la membresía en una red ad hoc varía con el tiempo. Por esa razón los AE multicast deben lidiar con esa característica. Por otra parte los AE multicast generan un problema porque inundan la red con mensajes de control. De ahí surgen diversidad de versiones que implementan mecanismos para contrarrestar esta desventaja.

Por ejemplo MAODV, que es un AE Multicast muy popular, es una extensión del protocolo AODV, que se detallará más adelante, pero en versión multicast, ya que el protocolo AODV originalmente es un protocolo unicast reactivo diseñado para nodos móviles en redes ad hoc. Otros protocolos como MCEDAR (*Multicast Core-Extraction Distributed Ad hoc Routing*) combinan los dos tipos de multicast basado en árbol y basado en malla.

Hay que mencionar también que los AE multicast, aunque son muy útiles en las redes, presentan una gran desventaja en cuanto a la seguridad, porque habilitan capacidades que pueden ser aprovechadas con intenciones maliciosas, por ejemplo facilitan ataques a la integridad de la red como *flooding attack*.

El *flooding attack* o ataque por difusión es un tipo de ataque de denegación del servicio, dirigido a un solo nodo o a una red completa (Loo y cols., 2016). Hace uso de técnicas de broadcast y tiene como objetivo acabar con los recursos de memoria de un sistema. Para esto, envía muchas peticiones de conexión a un destino. La víctima no podrá manejar esta situación por mucho tiempo, hasta que se satura y queda fuera de servicio. En una red ad hoc, el problema es aún más crítico, porque múltiples nodos pueden ser atacados y debido al procesamiento exhaustivo de grandes volúmenes de peticiones rápidamente agotarán su batería y quedarán fuera de servicio, hasta que toda la red colapse por completo.

**AE Multi-path:** Proveen ventajas significativas para una red a cambio de una alta complejidad. En estos, se usan múltiples rutas para enviar datos desde la fuente al destino. Aunque proporcionan más tolerancia a fallas y menos retardo en la búsqueda de rutas, gastan más recursos de red, como ancho de banda. Otra ventaja que presentan es que permiten el balanceo de carga mediante distribución del tráfico en situaciones de congestión, eso es porque permiten el almacenamiento de rutas disponibles o predeterminadas y redundancia de rutas al ofrecer alternativas para entregar los datos.

Alotaibi y Mukherjee (2012) realizaron una compilación de la bibliografía sobre estudios de los AE en redes ad hoc. De su artículo, se extrajo la Figura 2-4 para dar una idea de la cantidad de algoritmos de enrutamiento que han surgido en las últimas décadas. En el Anexo A se encuentra la lista con el nombre completo y las siglas de cada algoritmo.

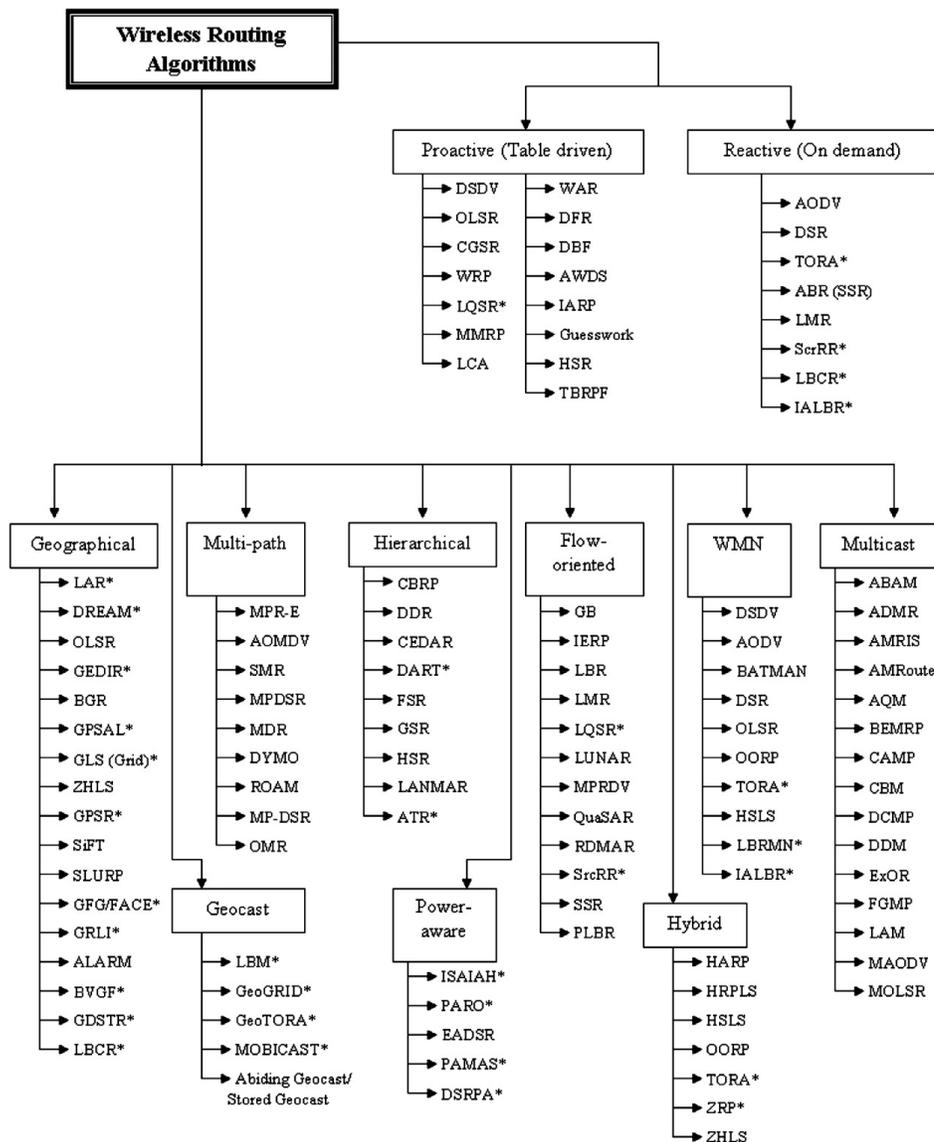


Figura 2-4.: Protocolos de enrutamiento en redes ad hoc (Alotaibi y Mukherjee, 2012)

Por otra parte, debido a la diversidad de mecanismos usados para abordar el problema del enrutamiento, existen otras categorías adicionales a las ya mencionadas. Algunas de ellas son:

**AE Geográfico:** para realizar el proceso de enrutamiento, en lugar de la dirección de red usa información

de coordenadas geográficas obtenidas mediante GPS o mediante puntos de referencia de un sistema de coordenadas fijo. De esta forma envía paquetes de datos o de control hacia las coordenadas del nodo destino si estas existen. Con esto evita búsquedas del destino a través de toda la red, reduciendo así el overhead de mensajes de control. En este tipo de algoritmos se usan una o más combinaciones de las estrategias *single path*, *multipath* y *flooding*. Un ejemplo de algoritmo geográfico es *Geographic Routing Protocol for Aircraft Ad hoc Network* (GRAA) descrito en Oubbati, Lakas, y Zhou (2017) que usa estratégicamente la información de ubicación geográfica para enrutar el tráfico entre aeronaves.

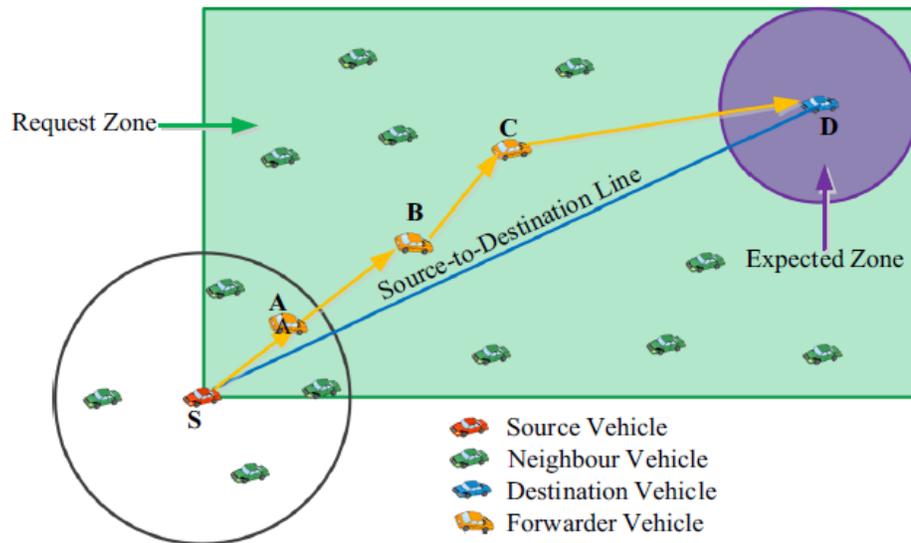
La mayoría de AE Geográficos de ruta única, se basan en una técnica llamada *Greedy Forwarding* (GF). Es una técnica en la que un nodo fuente, que requiere comunicación con un destino, reenvía los paquetes únicamente a su vecino más cercano geográficamente de entre todos sus vecinos. Esto se repite en cada nodo intermedio hasta alcanzar el destino. Al contrario de *flooding*, el reenvío solo se hace a un solo nodo vecino. Su principal problema, reconocido como *GF empty-neighbor-set problem* se presenta cuando el nodo no puede encontrar ningún vecino cercano a él, entonces el proceso de reenvío termina sin haber alcanzado el destino. Un ejemplo de AE geográfico comúnmente encontrado en los estudios de VANETs es el algoritmo de enrutamiento asistido por ubicación o posición LAR.

**AE Geográfico LAR:** el enrutamiento asistido por posición (*Location-Aided Routing*) es un algoritmo geográfico y a la vez es reactivo. Define dos áreas o zonas, área de expectativa (*expected zone*) y área de petición (*request zone*). La primera se refiere al área donde es más probable encontrar el nodo destino y la segunda el área por donde se deberá propagar la petición de ruta hacia el destino. La zona esperada se calcula muestreando la posición  $D$  del destino y su velocidad aproximada en un intervalo de tiempo, usando la fórmula  $v(t_1 - t_0)$  para obtener el radio de dicha área circular. La zona de petición será el área más pequeña que abarque tanto el área esperada como el área cercana al nodo origen (Figura 2-5).

**AE Geo-casting:** los mensajes se envían desde una fuente a múltiples destinos de forma similar al *multicasting*, pero en lugar de emplear la dirección de destino, usan únicamente la información de las coordenadas geográficas. En este tipo de AE los nodos de enrutamiento geográfico se llaman *Geo-Routers* y el área cercana al router y que es atendida por este, se denomina *área geográfica de servicio*.

**AE Power-aware:** para referirse a estos protocolos se usan frecuentemente las palabras en inglés *Energy-Efficient*, *Energy-Aware*, *Energy-conserving*, *Energy-Management*, entre otras. La movilidad y los recursos de procesamiento agotan la fuente de energía o batería de los nodos. Por lo tanto, en una red ad hoc cada nodo estará limitado por el factor del tiempo de vida de la batería. Por esto, es importante que los AE tengan en cuenta esta restricción. Sin embargo, los algoritmos tradicionales solo se enfocan en la conexión y el reenvío de los paquetes a través de la red sin tener en cuenta el gasto de energía requerido para su implementación.

Para incluir la eficiencia energética en algunos algoritmos como *Geographical Adaptive Fidelity* (GAF), que también se considera jerárquico y geográfico, los nodos pueden tener tres diferentes estados:



**Figura 2-5.:** Esquema de enrutamiento LAR direccionado (Raw y cols., 2015)

descubrimiento, activo o dormido, y a través de mecanismos que determinan cuáles nodos no están activos durante el enrutamiento, se envía una orden para que cambien a estado de reposo o *sleep state*, de esta manera se evita el desperdicio de batería.

Otra forma de ahorrar energía es mediante los esquemas de administración de la energía dependiendo de la actividad del nodo. De ahí que algunos algoritmos establezcan diferentes estados para un nodo, por ejemplo: libre, recibiendo, transmitiendo o dormido. Según la información obtenida de Barbeau y Kranakis (2007), se puede concluir que cuando un nodo se encuentra en modo dormido (*sleep*) presenta un ahorro significativo de energía, respecto al consumo durante los demás estados. En otros abordajes, el ahorro de energía se efectúa controlando la potencia transmisión. Cabe resaltar que la energía gastada en una transmisión es proporcional a la distancia entre emisor y receptor. Entonces, en dichos protocolos, se aumenta la potencia cuando el nodo destino está lejos y se disminuye cuando está cerca.

Algunos de los algoritmos que tienen como objetivo administrar eficientemente el consumo de energía en las redes ad hoc, se listan en la Tabla 2-1. Para una descripción detallada de cada uno, remitirse a Rashvand y Chao (2013). Muchos de estos, son el resultado de optimizaciones realizadas a otros algoritmos ya existentes; cuando se les aplica ciertas mejoras para atacar el problema de la energía, se convierten en nuevos protocolos y se agrega palabras al nombre del protocolo original para señalar que ahora cuentan con la característica de eficiencia energética.

| Sigla  | Nombre del algoritmo  |
|--------|---|
| ReLEE  | Reliable and energy-efficient routing   |
| EEOADR | Energy-Efficient On-Demand Routing Algorithm                                      |
| EECARP | Energy-Efficient Congestion-Aware Routing Protocol                                |
| PEER   | Progressive Energy-Efficient Routing  |
| PAR    | Power-aware Routing Protocol  |
| EEPAR  | Energy Efficient Power Aware Routing  |
| EARA   | Energy Aware Routing Algorithm  |
| E2IRP  | Energy Efficient Integrated Routing Protocol                                      |
| EELAR  | Energy Efficient Location Aided Routing   |
| LBPC   | Location-based Power Conservation Scheme for MANETs                               |
| CBRP   | Cluster-based routing protocol  |
| BAMPS  | Battery aware multiple input multiple output (MIMO) mesh network power scheduling |

**Tabla 2-1.:** Protocolos de enrutamiento de energéticamente eficientes en redes ad hoc, descritos en (Rashvand y Chao, 2013).

**Orientados a la seguridad:** las redes ad hoc son vulnerables a ataques porque comparten un canal de radio no protegido, no cuentan con un mecanismo de administración central de la red ni de asociación para los usuarios. Además, porque tienen recursos limitados y por la transitoriedad de su topología. Las vulnerabilidades de seguridad en una red ad hoc representan retos, al igual que para otro tipo de redes, en todas las capas del modelo OSI.

Las categorías de algoritmos descritas anteriormente solo se enfocaban en la resolución de problemas relacionados con las formas de direccionamiento, mecanismos de difusión o uso eficiente de energía. Sin embargo, no tienen en cuenta en su diseño los riesgos de seguridad. En ese aspecto algunas de las necesidades fundamentales son: evitar o detectar aplicaciones maliciosas como virus, asegurar la autenticación en la red con mecanismos de cifrado, proteger de ataques a los protocolos de enrutamiento y los mecanismos de reenvío de paquetes, proteger los protocolos de la capa de enlace e incluso proteger la capa física de ataques como *eavesdropping* y *jamming attack* que aprovechan la naturaleza inalámbrica de las transmisiones.

De forma general, los ataques pueden ser internos, si se originan por un nodo que está dentro de la red o externos si los origina una entidad foránea. Pueden ser activos, si modifican los paquetes o envían paquetes falsos o maliciosos, y pasivos, si no afectan por completo la operación normal de la red pero interceptan el tráfico o lo desvían.

El ataque *eavesdropping* es un tipo de ataque a la confidencialidad de los nodos de la red, también se conoce como *sniffing* o *snooping*. Es un ataque interno y pasivo, difícil de detectar porque parece que todo está bien en la red, cuando en realidad un nodo malicioso a logrado acceso ilegal a información

confidencial de uno o varios nodos, mediante mecanismos de interceptación tanto de los paquetes transmitidos como recibidos.

El término *jamming* se traduce en este contexto como interferencia o interceptación. El ataque *jamming* es un tipo de ataque de denegación de servicios sobre el medio de transmisión. En este se inyecta señales que interfieren en el canal impidiendo la transmisión y recepción de señal original. La pueden bloquear completamente y provocar indisponibilidad total o deteriorarla parcialmente provocando absorción y desaparición de paquetes y con ello una disminución en la calidad de los servicios.

Otros ataques internos y activos bien conocidos, que vulneran la disponibilidad de la red, son *Black Hole* y *Gray Hole*. En ambos, un nodo malicioso logra que los paquetes siempre lleguen a él. En el ataque *Black Hole* ese nodo elimina todos los paquetes que captura, por eso se dice que los paquetes se pierden en un agujero negro. Mientras que en el ataque *Gray Hole* el nodo se comporta de forma impredecible, a veces elimina paquetes o envía información de enrutamiento falsa pero otras veces reenvía normalmente los paquetes de solicitud y respuesta de descubrimiento de rutas. Al comportarse temporalmente como un nodo genuino hace más difícil su detección, en todo caso, su objetivo es sabotear el proceso de enrutamiento y deteriorar el desempeño de la red.

| <b>Sigla</b> | <b>Protocolo orientado a la seguridad</b>   |
|--------------|---|
| ACOR         | Admission Control Enabled On Demand Routing   |
| ARA          | Ant-colony based Routing Algorithm for MANETs   |
| ARAN         | Authenticated Routing for Ad-hoc Networks   |
| ARIADNE      | A Secure On-demand Routing Protocol for Ad-hoc Networks                               |
| CHAMP        | Caching and Multipath Routing   |
| ESAO DV      | Extra Secure Ad hoc On-demand Distance Vector   |
| MAODDP       | Mobile Ad-hoc On-demand Data Delivery Protocol  |
| MEPA         | Minimum Exposed Path to the Attack  |
| MuSeQoR      | Multipath Security-aware QoS Routing  |
| RAODV        | Reliable Ad hoc On-demand Distance Vector Routing Protocol                            |
| SAODV        | Secure AODV   |
| SDSR         | Secure DSR  |
| SEAD         | Secure Efficient Distance Vector Routing for Mobile Wireless Ad-hoc Networks          |
| SecMR        | Secure Multipath Routing  |
| SENCAS T     | Scalable Protocol for Unicasting and Multicasting in a large Ad-hoc Emergency Network |
| SRP          | Secure Routing Protocol   |

**Tabla 2-2.:** Protocolos de enrutamiento orientados a la seguridad

De manera semejante a lo que sucede con los protocolos de eficiencia energética, algunos algoritmos

orientados a la seguridad son versiones mejoradas de los algoritmos ya existentes, que incluyen estrategias para establecer la seguridad como objetivo principal. En este caso, se puede identificar un protocolo de este tipo porque su nombre tiene palabras como *Trusted*, *Secure*, *Reliable* o *Security-aware*. Algunos ejemplos de estos protocolos son descritos en Loo (2016), Basagni y cols (2013), Oubbati, Lakas y Zhou (2017) y se listan en la Tabla 2-2.

**AE Jerárquico:** en este tipo de algoritmos los nodos se organizan en grupos denominados zonas o clústers. Cada clúster tiene un nodo director o *Cluster Head* (CH) y al menos un nodo gateway. El primero mantiene la conectividad de todos los nodos demás dentro de su zona, el segundo tiene la capacidad de conectarse con CHs de otras zonas, por lo tanto, es el punto de acceso entre dos clústeres vecinos. Se producen así, niveles de jerarquía entre los nodos que cumplen estos roles y los que no, siendo estos últimos los de más baja jerarquía. Las diferencias entre los protocolos jerárquicos se basan en los mecanismos de asignación de identificadores (IDs) para cada nivel.

Las estrategias más usadas en AE jerárquicos combinan dos formas de enrutamiento: inter-cluster, para comunicaciones dentro del clúster, usando enfoque proactivo, e intra-clúster para comunicarse con otros clústeres vecinos usando el enfoque reactivo. Estos algoritmos permiten que la red sea más tolerante a cambios en la topología y su mayor ventaja es proporcionar escalabilidad, pues cuando el número de nodos de una red aumenta, su complejidad también se incrementa y es más difícil coordinar los nodos en una red sin jerarquía.

Algunos autores como Loo (2016) reestructuran la clasificación de los protocolos de enrutamiento para dar una visión más holística. Se tomaron los datos de esa referencia para realizar la Figura 2-6 donde se puede ver la distinción entre las categorías multicast y unicast en un primer nivel. A su vez divididas en protocolos proactivos, reactivos o híbridos y dentro de estos la subdivisión de aquellos que implementan jerarquía o no, de los que usan el enfoque el geográfico.

### 2.1.5. Algoritmos de enrutamiento más conocidos

Algunos de los algoritmos de enrutamiento más usados en las redes ad hoc son AODV, DSR, OLSR y BATMAN, se describen a continuación.

#### AODV

Su sigla significa *Ad Hoc On-Demand Distance Vector* y es un protocolo reactivo. En el descubrimiento de rutas combina las dos técnicas de unicast y multicast y usa números de secuencia para identificar las rutas más actualizadas. Los nodos mantienen una entrada de la tabla de enrutamiento si el parámetro de tiempo de vida se actualiza; en caso contrario se elimina. Esto quiere decir que se guarda mientras la ruta sea requerida por la fuente. Antes de enviar paquetes a un destino, primero se revisa si existe

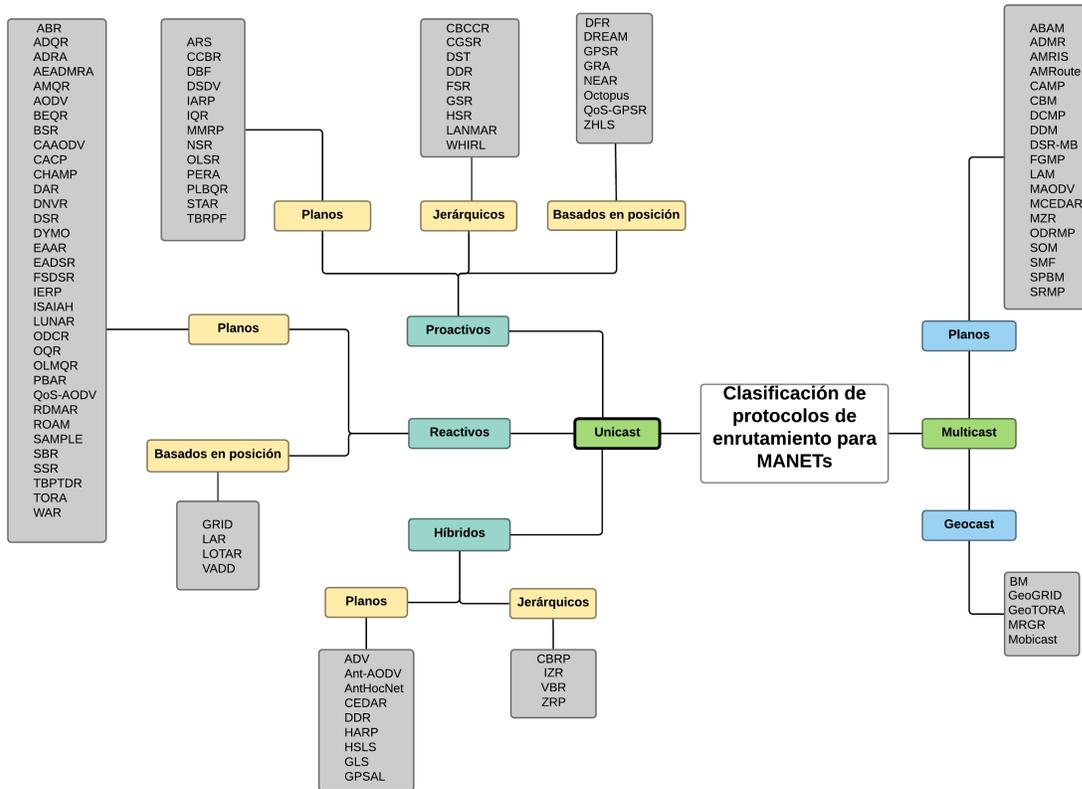


Figura 2-6.: Propuestas más relevantes de algoritmos de enrutamiento en MANETs. Adaptado de (Loo y cols., 2016)

una ruta. Si no la tiene, empieza el ciclo de descubrimiento mediante un paquete de solicitud de ruta o *Route Request* (RREQ). Cuando un nodo destino o nodo intermedio recibe ese mensaje, envía una respuesta de petición de ruta o *Route Request Response* (RREP) al origen, únicamente si el número de secuencia es igual o mayor que el que se ha especificado en el mensaje RREQ; de lo contrario, si es menor, será eliminado. Si se mueve un nodo o se pierde una conexión, se envía un mensaje RERR (*Request Error*) a los nodos involucrados. Solo entonces, la fuente puede reiniciar el proceso de descubrimiento de ruta.

En AODV, el retardo en el establecimiento de la conexión es bajo, aunque presenta mayor sobrecarga de mensajes de control porque se aumenta el tráfico al generar muchos paquetes de reenvío de ruta.

## DSR

El enrutamiento de fuente dinámica, o *Dynamic Source Routing* (DSR), es un protocolo en el que el proceso de descubrimiento de ruta se realiza mediante *flooding* de paquetes RREQ en la red y un solo un nodo realiza la retransmisión de un RREQ a todos los vecinos, que no han reenviado ese mismo paquete, excluyendo el nodo destino. Es decir, solo reenvía si el paquete no es un duplicado del RREQ original. En cada nodo por el que pasa el paquete RREQ, se agrega un encabezado. De esta forma,

se arma una lista ordenada de nodos para llegar a un determinado destino. Si se encuentra una ruta, se envía una respuesta RREP al origen usando los mismos saltos que se han guardado en la lista. La diferencia radica en que en este protocolo los nodos intermedios guardan en caché la información de las rutas y con esta estrategia pueden disminuir el tiempo en el proceso de descubrimiento de las mismas. Sin embargo, la adición progresiva de información a la cabecera del paquete RREQ genera mas carga de tráfico para la red y exige mas recursos de procesamiento.

## OLSR

*Optimized Link State Routing* (OLSR), es un protocolo proactivo, en el que los nodos actualizan la información de topología intercambiando periódicamente mensajes *Hello* con sus vecinos mas cercanos. Con ayuda de estos mensajes, cada nodo adquiere la información de la topología local. En OLSR, se elige un grupo de nodos entre los que se encuentren a un salto, y únicamente estos estarán habilitados para reenviar la información de la topología. Este grupo se denomina MPRs (Multipoint Relays, por sus siglas en inglés) y sus miembros son los encargados de generar la información de estado de enlace. Con esta estrategia se minimiza la cantidad de retransmisiones necesarias y se reduce el *overhead*, respecto a otros como AODV y DSR.

## B.A.T.M.A.N

Este protocolo de capa 2, conocido como BATMAN por sus siglas que significan *Better Approach To Mobile Ad hoc Networking*. Fue creado especialmente para redes ad hoc Mesh, con el propósito de superar algunas de las desventajas de OLSR.

OLSR presenta un problema de falta de sincronización entre la información de rutas guardada en cada nodo y la topología actual, debido al tiempo que tarda en propagarse a través de los mensajes de actualización. El protocolo BATMAN proporciona una solución a este problema eliminando los mensajes de diseminación de la topología. Cada nodo ejecuta tres tareas que son: 1) envío de mensajes originadores OGM (OriGinator Messages); 2) para cada destino en su tabla de enrutamiento realiza un procedimiento de *ranking* para verificar cual es el vecino que se encuentra a un salto y es el mejor siguiente salto para llegar a dicho destino, una forma de hacerlo es contando el número total de OGMs recibidos por cada vecino, y 3) reenvío broadcast de los OGM recibidos (Kulla, Hiyama, Ikeda, y Barolli, 2012), usando un mecanismo de *flooding* selectivo, para excluir a los vecinos que han reenviado previamente un mismo OGM que se acaba de recibir.

BATMAN es el protocolo y *batctl* es el comando interfaz que permite para acceder a todas las opciones configuración y depuración del módulo *Batman-adv*, que es un módulo del kernel de Linux que permite a las estaciones correr el protocolo (González, Ortiz, y Zárate Ceballos, 2016).

Los protocolos anteriormente descritos presentan ventajas y desventajas, aunque todo depende del escenario de aplicación de los mismos. De manera, se puede concluir que AODV es un algoritmo óptimo, pero no es el mejor. Se comporta mejor que DSR en ambientes de alta movilidad y en redes con un número elevado de nodos. Sin embargo, no proporciona una buena calidad del servicio. En contraste

con DSR, este presenta mucho más *overhead* que AODV aunque DSR es mejor para redes pequeñas, con tasas de tráfico y movilidad moderadas (Gulati y Kumar, 2014).

Por otra parte, el protocolo BATMAN tiene un mejor desempeño que OSLR en escenarios de alta movilidad, mostrando una mejor métrica de *throughput*, porque no elimina los paquetes cuando las rutas son inestables. A diferencia de OLSR, las mantiene en un buffer por cierta cantidad de tiempo, por eso es mejor en ambientes dinámicos. Por otra parte, BATMAN presenta mayor retardo que OSLR en redes de gran tamaño (Kulla y cols., 2012). No obstante lo anterior, de todos estos protocolos ninguno implementa mecanismos de seguridad.

## 2.2. Clustering en redes ad hoc

Clustering en una red ad hoc se entiende como el proceso de división de la red en varias subredes, donde se agrupan nodos y cada grupo es manejado por un nodo director o también llamado *Cluster Head* (CH) que actúa como un controlador local (Barbeau y Kranakis, 2007). Algunos autores, como Loo et al (2016), llaman Nodo Maestro al *Cluster Head* y nodo normal a cualquiera de los demás nodos. Los diferentes mecanismos o reglas para armar clúster y elegir el *Cluster Head* en una red ad hoc son temas potenciales de investigación.

Puede decirse que una red ad hoc esta constituida cuando exista por lo menos un clúster o grupo. Sin embargo, si la compone un número considerable de nodos, estos pueden sub-agruparse para cumplir tareas específicas y hacer que la red sea más escalable, es decir, podrá ampliar su cobertura valiéndose de mecanismos para optimizar las transmisiones. Esto es lo que se denomina una estructura jerárquica en redes ad hoc.

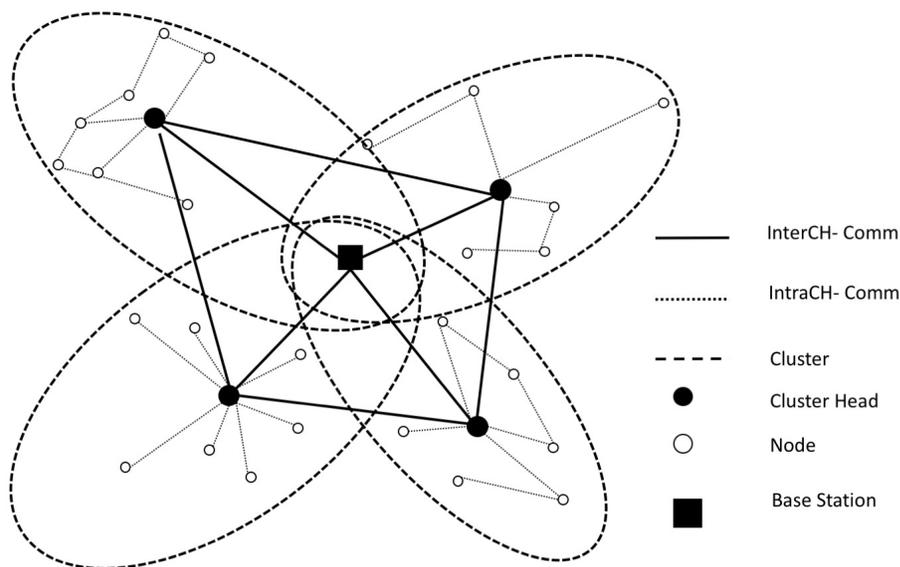
Entonces, un clúster en una red inalámbrica ad hoc, es un grupo de nodos asociados a través de un CH que define los límites de la operación del clúster y tiene capacidad de realizar tareas especiales que facilitan la comunicación intra-clúster. Además, puede administrar las rutas y el estado de los enlaces hacia otros clústeres, permitiendo la comunicación inter-clúster. En la Figura 2-7 se ilustra este concepto.

Otro aspecto a tener en cuenta en el funcionamiento de un clúster es que si el nodo CH se retira o se interrumpe el enlace, la red debe ser lo suficientemente robusta para mantener la integridad de la comunicación. Por esa razón, debe existir un mecanismo para reemplazar el CH sin causar afectación, brindando así, la característica de auto-organización deseada en una red ad hoc.

### 2.2.1. Esquemas de clustering en redes ad hoc

Existen diversos mecanismos para realizar el agrupamiento en las redes ad hoc, denominados esquemas de clustering (*Clustering Schemes*). Wei y Anthony Chan(20017) proponen cuatro formas de clasificar estos esquemas (Figura 2-8) de acuerdo con los siguientes criterios:

1. Si el CH usa un solo salto en la red para alcanzar a sus nodos, el clustering es *Single Hop*; o si que requiere varios saltos, será Multi-saltos (*Multihop*)



**Figura 2-7.:** Conformación de clústeres en una red ad hoc. Tomado de Zarate(2016)

2. Si el mecanismo está basado o no en la información de localización de los nodos.
3. Si de forma global en la red todo está sincronizado (*Synchronous*) o si por el contrario es asíncrona (*Asynchronous*)
4. Si los nodos en la red son móviles o estacionarios, es decir, según si el clustering se realiza en una MANET o en una red ad hoc Mesh.



**Figura 2-8.:** Clasificación de los esquemas de clustering. Adaptado de Wei y Anthony Chan (2007)

Por otra parte Wei y Anthony Chan (2007) describen el clúster como una entidad que se compone de tres tipos de nodos: Cluster Head, Gateway y nodos normales. En cada clúster un nodo es seleccionado como CH que se encarga de dirigir el clúster. Pero es el nodo Gateway, que puede pertenecer

a más de un clúster, el que sirve como puente para establecer conexión con otros Cluster Heads y provee conectividad inter-clúster. Entonces, se crea una especie de *backbone* de la red que se compone de nodos CH y de nodos Gateway (Gavhale y Saraf, 2016). (Figura 2-9). Con esta estructura, se obtienen algunas ventajas: 1) reducción de la sobrecarga de mensajes (*overhead*), la sobrecarga por flooding, y evita colisiones de paquetes; 2) Disminuye el impacto ocasionado por cambios en los nodos permitiendo una topología más estable y 3) solo los nodos CH y Gateway necesitan mantener la información de enrutamiento.

Para el sistema TLÖN un clúster podría concebirse en dos dimensiones distintas: clúster físico y clúster lógico que actuaría como una "micronube". Así, una vez establecido un clúster físico en una red ad hoc, podría configurarse sobre este un clúster lógico o nube pequeña que se encargará de realizar tareas distribuidas específicas o de proveer un servicio determinado.

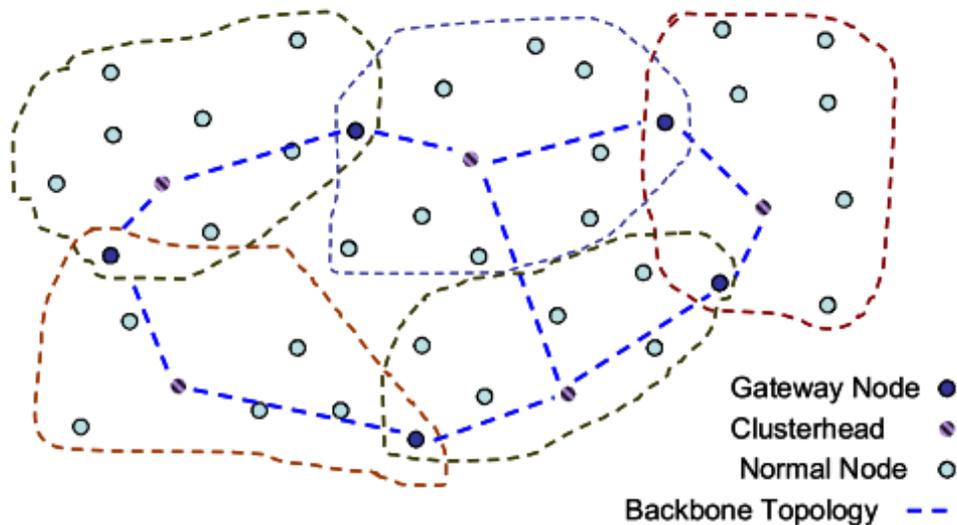


Figura 2-9.: Comunicación inter-clúster. Tomada de (Wei y Anthony Chan, 2007)

## 2.3. Sistemas distribuidos

### 2.3.1. Computación en la nube

La tecnología de nube convencional, reconocida con el término *Cloud*, que se encuentra en apogeo en los últimos años, permite acceder a servicios computacionales a través de una red, por ejemplo, para almacenar información en un espacio en internet, al que se puede acceder desde cualquier lugar. Entre los sistemas más conocidos se encuentran GoogleCloud, AmazonCloud y la iCloud de Apple (Antonopoulos y Gillam, 2017).

La palabra nube, como lo expresa Fitzek (2014), es una abstracción de un sistema que consiste en recursos distribuidos interconectados que son compartidos para un determinado propósito. Los ser-

vicios más comunes que provee una nube incluyen capacidades de procesamiento distribuido, acceso ubicuo a la información, almacenamiento distribuido, seguridad del software, plataformas de pruebas, entre otros. Los recursos de una nube que se ofrecen como servicios pueden ser tanto físicos (hardware) como intangibles (software o aplicaciones).

El concepto de virtualización está fuertemente asociado a la computación en la nube, y esto se debe a que la virtualización es el mecanismo que permite implementarla (Portnoy, 2012). Con la virtualización, por ejemplo, una sola máquina física o servidor físico puede multiplicar sus servicios desplegando múltiples servidores virtuales o máquinas virtuales cada una con sus propias características de sistema operativo, memoria y espacio de disco, en lugar de tener cada servidor en máquinas diferentes, lo que reduce el volumen de máquinas físicas necesarias. De esta forma, es posible aprovechar al máximo la capacidad de una sola máquina física. Sin embargo, para el usuario final, la división del trabajo entre servidores es imperceptible. Es decir, no percibirá el mecanismo de virtualización (Antonopoulos y Gillam, 2017). En consecuencia, los servicios se ofrecen de forma transparente.

La computación en la nube, conocida como *Cloud Computing*, ha solucionado una necesidad de disminuir los costos de operación y mantenimiento de los centros de cómputo propios de muchas empresas o entidades que no contaban con la infraestructura y logística suficiente para mantener actualizados sus servidores. Con la virtualización, estas tareas se han simplificado o incluso muchas organizaciones han trasladado la responsabilidad del mantenimiento de hardware y software a terceras empresas de alta capacidad tecnológica que se ocupan específicamente de esto, haciendo de la virtualización un negocio altamente rentable en la actualidad. La computación en la nube también ha impulsado el desarrollo de Internet de las cosas (IoT, por sus siglas en inglés).

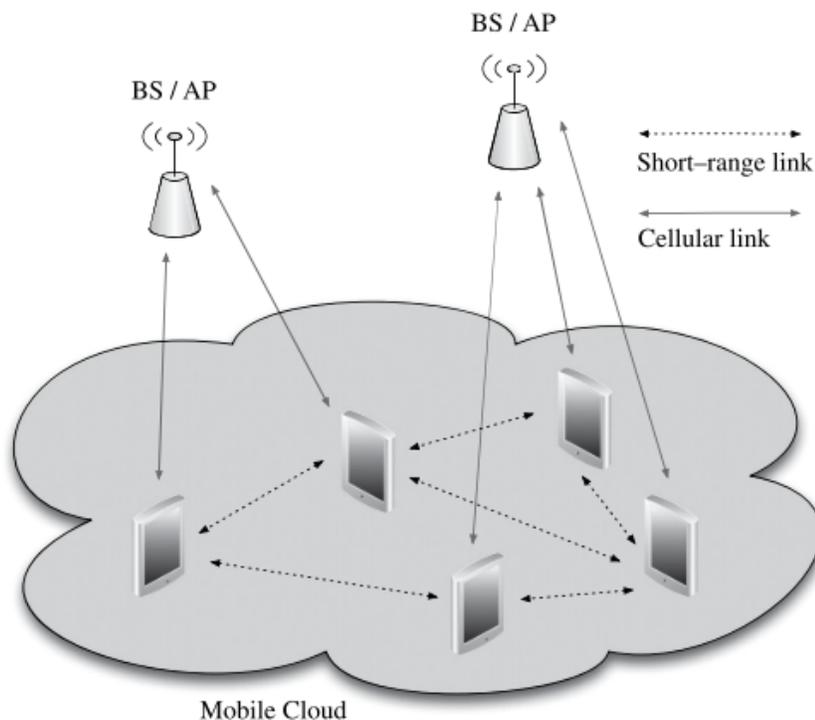
Los proveedores de servicios en la nube han implementado clústeres tanto en su infraestructura física como en su arquitectura de servicios y esto les permite multiplicar sus capacidades. Es por esto que se puede decir que las tecnologías propulsoras de la nube son aquellas relacionadas con mecanismos de clustering, virtualización y redes de telecomunicaciones.

### **Computación en una nube móvil**

Una nube móvil se define como un arreglo cooperativo de nodos conectados dinámicamente compartiendo recursos de manera oportunista (Fitzek, 2014). A partir de esto, se establecen entonces relaciones sociales entre los nodos, que definen la viabilidad para cooperar y determinan la forma en que la cooperatividad es establecida en una nube. Por otro lado poseen una característica de dinamismo porque los canales de comunicación inalámbrica son susceptibles a fluctuaciones temporales, así como a cambios en los nodos. Por ejemplo, cuando estos se mueven, se unen a la nube o la abandonan. De igual forma, se deben tener en cuenta las características de la conexión. Los nodos pueden estar conectados directamente entre sí, esto es llamado *peer-to peer* (P2P) o lógicamente mediante redes mediadoras. En estas, los nodos son concebidos como cualquier equipo de comunicación con

capacidades de establecer conexión con otros nodos y se consideran nodos oportunistas, porque toman ventaja de las oportunidades que surgen. Además, en este escenario, lo más importante son los recursos, que se refieren a cualquier entidad o medio que esté disponible en la red y pueda ser compartido a los recursos embebidos en los nodos.

Fitzek (2014), presenta una definición alternativa de nube móvil, donde se refiere a esta como un arreglo cooperativo de dispositivos inalámbricos cercanos. Cada uno de estos también puede ser conectado a otras redes a través de puntos de acceso (AP) o estaciones base (BS). De esta forma, una nube móvil tiene una arquitectura híbrida con características tanto de redes inalámbricas de área local como de redes celulares (centralizadas). En la Figura 2-10, se ilustra el concepto de nube móvil descrito anteriormente, donde los nodos móviles cooperan entre sí localmente y en una capa superpuesta están conectados por una red celular. Cada nodo cuenta con un conjunto de recursos que sirven de suministro para los servicios de la nube.



**Figura 2-10.:** Arquitectura básica distribuida-centralizada de una nube móvil (Fitzek, 2014).

Una ventaja de la nube móvil es que puede hacer uso eficiente de recursos que usualmente son limitados y costosos como la energía y el espectro. Abordando el concepto desde otra perspectiva, una nube móvil sirve como plataforma flexible para explotar grupos de recursos distribuidos que están conectados de forma inalámbrica y pueden agregarse o desagregarse de múltiples formas. En la Figura 2-11, se ilustra a cada dispositivo como un banco de recursos, entonces la nube móvil es equivalente a un grupo o clúster de recursos distribuidos y variables.

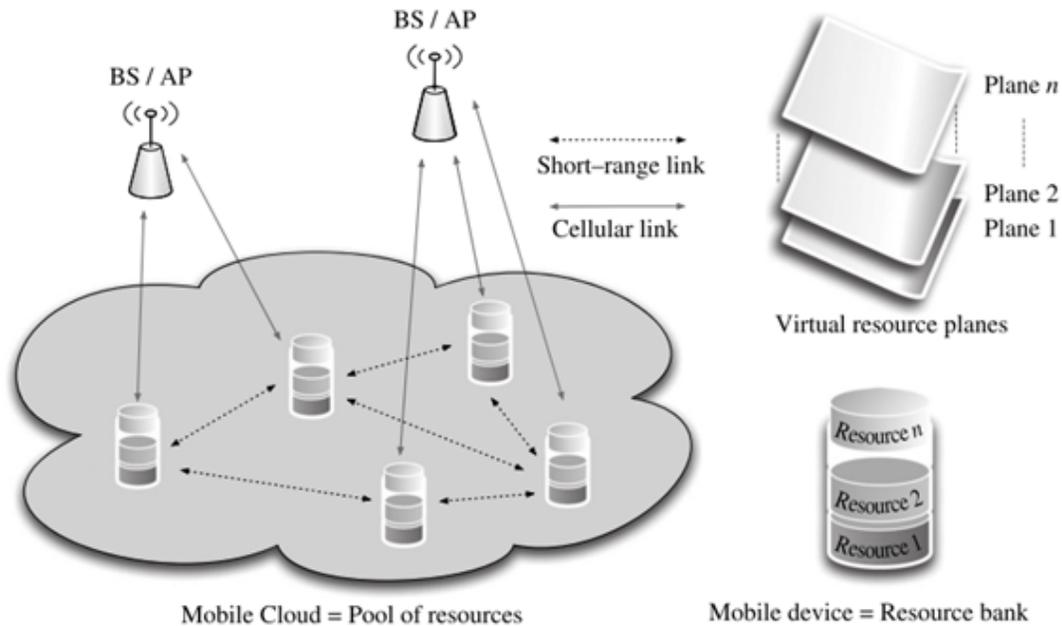


Figura 2-11.: Nube móvil como un conjunto de recursos distribuidos (Fitzek, 2014)

El concepto de nube móvil es crucial para abordar el problema de agrupamiento lógico en una red de procesamiento distribuido. Sin embargo, existen otros paradigmas emergentes que son importantes, dado que están evolucionando rápidamente en el mundo de la computación, estos son: Internet de las cosas (IoT), Computación en la niebla (Fog Computing) y Computación de frontera (Edge Computing).

### 2.3.2. Internet de las Cosas

Internet de las cosas fue definida en la recomendación de la ITU-T Y.2060 (ITU-T Study Group 20, 2012) como una infraestructura global que permite habilitar servicios avanzados de interconexión de cosas tanto físicas como virtuales, basados en tecnologías de la información y comunicaciones actuales y emergentes.

El término Internet de las cosas o IoT en su forma abreviada, por las siglas del inglés *Internet of Things*, denota una tendencia donde un gran número de dispositivos embebidos usan los servicios ofrecidos por los protocolos de Internet. Algunos de estos dispositivos cuando no son operados por intervención humana se denominan “objetos inteligentes” (*smart objects*). La mayoría de dispositivos IoT se conectan entre sí para formar redes o sistemas de propósitos específicos (Serpanos y Wolf, s.f.). En este sentido, el paradigma de IoT se articula con los conceptos de clúster y red ad hoc descritos anteriormente.

IoT permitirá que dispositivos electrónicos domésticos como refrigeradores o sensores se comuniquen entre sí a través de la nube. Sin embargo, el concepto también se extiende a aplicaciones industriales de donde se deriva el concepto de IoT Industrial (IIoT). Este consiste en insertar inteligencia a máquinas, sistemas y procesos industriales que, sumados a mecanismos de comunicación permiten

mejorar tareas de monitoreo y coordinación de una cadena productiva para lograr una alta calidad con una considerable reducción de los costos.

IoT también puede entenderse como un ambiente de interacción entre el mundo físico y el mundo digital, ya que existen diversas formas en las que se pueden establecer estas interacciones (ITU-T Study Group 20, 2012). Aunque es una tendencia en la actualidad y objeto de numerosos estudios, aún no existe una arquitectura estandarizada para la internet de las cosas. Sin embargo, uno de los enfoques más conocidos es la arquitectura de tres capas que se muestra en la Figura 2-12.



**Figura 2-12.:** Arquitectura básica de 3 capas de IoT

La primera es la capa de percepción o de las cosas, como su nombre lo indica, se relaciona con percibir o captar información del entorno. Esta es compuesta por dispositivos como sensores, actuadores y unidades de procesamiento que miden o detectan variables físicas o identifican otros objetos. La segunda es la capa de red, que permite conectar las cosas “inteligentes” y está compuesta por dispositivos de red y servidores. Sirve para transmitir y procesar la información captada por los sensores. La tercera es la capa de aplicación, la cual es responsable de entregar servicios al usuario. En esta capa se identifican aplicaciones IoT de propósito general como procesamiento o almacenamiento de datos y otras aplicaciones de propósito específico que se especializan en un conjunto de servicios particulares, por ejemplo, *Smart Homes* o *Smart Health* (Serpanos y Wolf, s.f.).

Las capacidades de la capa de percepción o capa de dispositivos se pueden clasificar en dos tipos: capacidades de dispositivo y capacidades de *gateway* (ITU-T Study Group 20, 2012). Algunas de sus características se describen a continuación:

- **Capacidades de dispositivo:** se refieren a la interacción directa con la red de comunicaciones. Aquí los dispositivos pueden recolectar y cargar información a la red o recibir información desde esta de forma directa, sin intermediarios. O también pueden interactuar indirectamente mediante *gateways* o dispositivos que ayudan en el envío o recepción de información hacia o desde

la red. Adicionalmente entre las capacidades de dispositivo dentro de la IoT, se resalta su capacidad de interconexión ad hoc, dado que se pueden dotar con la inteligencia necesaria para construir redes temporales de propósito específico, particularmente, en escenarios donde es crucial proveer escalabilidad inmediata y un rápido despliegue.

Otra habilidad deseada de los nodos es su capacidad de mantenerse en estados de reposo (*Sleeping*) o despiertos (*Waking up*), los cuales son esenciales para hacer uso inteligente de la batería o fuentes de alimentación eléctrica y en consecuencia, disminuir el consumo de energía.

- Capacidades de *gateway*: soporte de múltiples interfaces que permiten conectar los dispositivos a redes de diferentes tecnologías, ya sea inalámbricas o redes cableadas. De esta forma, será posible el uso de diferentes tipos de redes: de área local, redes telefónicas, redes celulares o incluso redes avanzadas como LTE.

También incluye la conversión de protocolos en la que las *gateway* o intermediarios permitirán que dentro de la capa de percepción coexistan equipos heterogéneos que usan diferentes protocolos de comunicación o actuarán como traductores para habilitar la interacción en la capa de percepción entre dispositivos que funcionan con tecnologías diferentes a las que usa la capa de red.

### 2.3.3. Computación en la niebla (Fog Computing)

La nube es vista como una capa alta en donde se encuentran agrupados muchos equipos de gran capacidad de procesamiento y almacenamiento que trabajan juntos. Esta capa se encuentra altamente diferenciada y separada de la capa de los dispositivos. Por lo general, los equipos que la componen que por general son servidores que se encuentran en centros de cómputo (*datacenters*) dentro de instalaciones alejadas del usuario final o de las entidades que ejecutan la percepción de las cosas. La Figura 2-13 muestra una representación de la computación en la nube sin restricciones (Openfog Consortium, 2017).

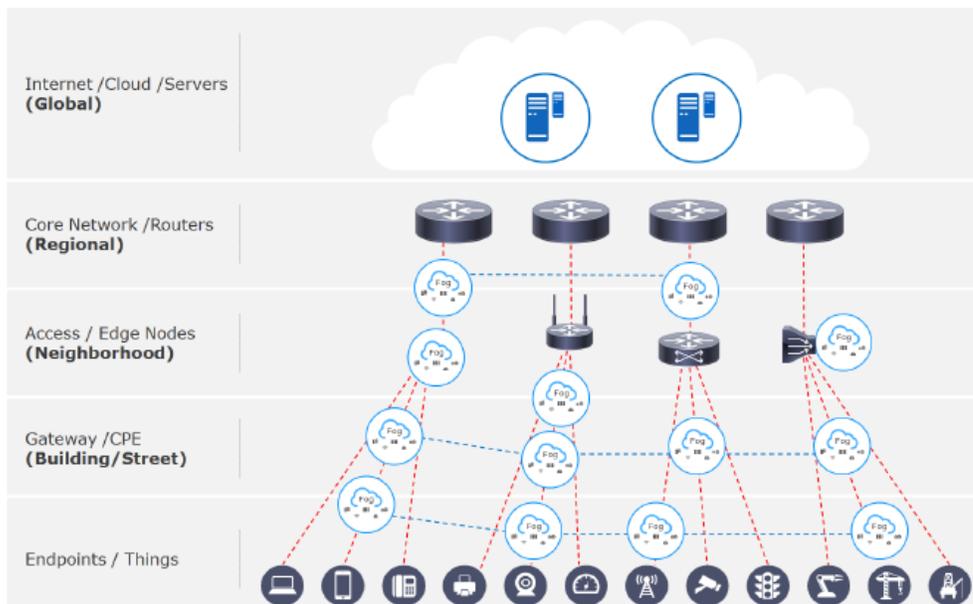


Figura 2-13.: Visión de la computación en la nube (Openfog Consortium, 2017).

Sin embargo, este paradigma es un modelo totalmente centralizado, ya que todo el procesamiento sucede en la nube, lo que significa que todos los pedidos y datos deben ser enviados a esta (Hu, Dhelim, Ning, y Qiu, 2017). Esto puede no parecer un problema, pero si se considera que el número de dispositivos conectados crecerá vertiginosamente para el año 2020 (Vaquero y Roderomero, 2020) es indiscutible que habrá un aumento potencial en el volumen de tráfico por dispositivo. Esto podría agotar los recursos de ancho de banda de las redes causando congestión y retardos en la comunicación.

Con el despliegue masivo de IoT, se prevé que millones de sensores y actuadores incrementarán las demandas de procesamiento en tiempo real y de aplicaciones sensibles al retardo. Se generarán grandes volúmenes de información, que requerirán un aumento exhaustivo en el esfuerzo de procesamiento y almacenamiento y en muchos casos no se justifica realizarlo de forma centralizada. Entonces, en ese escenario, la computación en la nube no es la solución más apropiada.

Una forma más inteligente de abordar el problema es mediante el paradigma de computación en la niebla o *Fog Computing*, que aparece para completar y optimizar los esfuerzos de la nube (Hu y cols., 2017). De forma general, este modelo de computación ubicua establece una capa de niebla o de micronebules cercanas a las cosas o dispositivos perceptores. Esta proximidad ayudará a la nube “grande” a hacer su trabajo de forma más eficiente, agilizando comunicación, reduciendo parámetros de latencia, evitando cuellos de botella y distribuyendo aún más los esfuerzos de procesamiento. En la Figura 2-14, se muestran nubes cercanas a la capa de las cosas, ubicadas en dispositivos de frontera de capa de red, que se pueden comunicar entre sí para formar uno o más clústeres de niebla en diferentes puntos del nivel de red, hasta conectarse con los servidores de la nube global.



**Figura 2-14.:** Comunicación de nodos fog en dirección Este/Oeste (Openfog Consortium, 2017).

### 2.3.4. Computación de Frontera (Edge Computing)

Tanto la computación en la niebla como la computación de frontera surgieron para enfrentar los desafíos se supone un exhaustivo y directo uso de la nube. Ambos tienen el objetivo de acercar los servicios de la nube a los dispositivos finales del usuario.

La computación de frontera o *Edge Computing* (EC), es un paradigma basado en la idea de ejecutar la computación y almacenamiento cerca de la fuente generadora de datos. En otras palabras, se trata de implementar tareas de cómputo en los dispositivos de frontera, que son inteligentes y tienen ciertas características potenciales. Estos se pueden intercomunicar para formar una red de frontera y, por otra parte, se enlazan con la nube, es decir, con un gran *data center* a través de una red como Internet (Dolui y Datta, 2017).

Aunque Fog y EC son paradigmas descentralizados, jerárquicos y distribuidos, la diferencia radica en la capacidad de cómputo y la cercanía al usuario final. En EC quienes ejecutan el procesamiento son los dispositivos de usuario o equipos finales, mientras que en Fog lo hacen los dispositivos de frontera de red, que en otras palabras, son los equipos conectados al usuario final un tanto más poderosos en cuanto a capacidades de cómputo (Hu y cols., 2017). En conclusión, en EC el procesamiento se ejecuta en el equipo final y en Fog está cerca del equipo final y no dentro de éste. De todas formas, ambos paradigmas están diseñados para proveer servicios de virtualización, permitir movilidad y escalabilidad.

## 3. Sistema TLÖN

### 3.1. TLÖN un sistema en construcción

El sistema TLÖN es una propuesta de sistema computacional distribuido, que se inspira en los modelos sociales establecidos en nuestro entorno humano, para abstraer conceptos y comportamientos que pretenden ser implementados de forma artificial en todos los componentes de sistema. Aunque es claro que no se podrá imitar con fidelidad la complejidad del comportamiento humano y social, se tiene la convicción de que un abordaje social inspirado permite definir las características deseadas del todo y cada una de sus partes, y así trazar los objetivos para hacer realidad la implementación de una “pseudo sociedad ideal” en un ambiente artificial altamente dinámico, como las redes ad hoc inalámbricas, que sea capaz de manejar efectivamente problemas como ubicuidad, movilidad, dinamismo, comportamiento estocástico, coordinación, virtualización y auto-configuración.

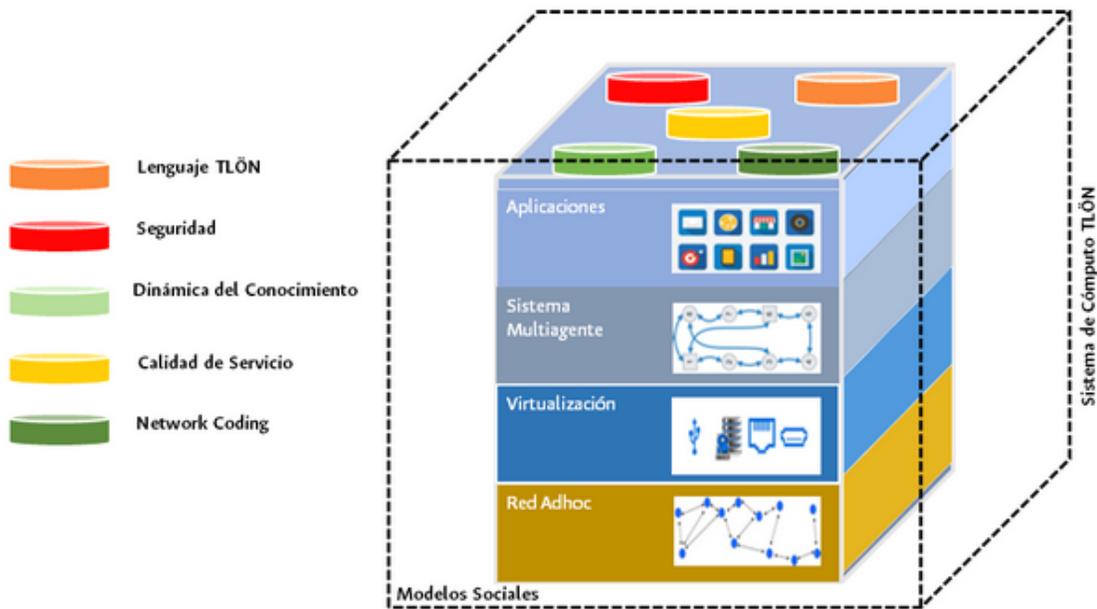
La construcción del sistema TLÖN es el proyecto principal del grupo de investigación en Redes de Telecomunicaciones Dinámicas y Lenguajes de Programación Distribuidos – TLÖN. La forma en que se explica este sistema es mediante un modelo de capas como el de la Figura 3-1. En la primera capa está la infraestructura, donde se encuentra la red ad hoc. La segunda capa contiene la virtualización inalámbrica y el Sistema Operativo. En la tercera capa, se cuenta con el sistema multiagente, donde operan comunidades de agentes que proveen servicios en toda la red. Finalmente, se tiene una capa de aplicaciones específicas del sistema de cómputo. De forma transversal, se incorporan componentes que generan interacciones en todas las capas, como son: lenguaje del sistema y su ontología, seguridad, calidad de servicio y codificación de red.

### 3.2. Conceptos generales

A continuación, se ampliarán algunos conceptos asociados al modelo, especialmente aquellos que se relacionan con la capa de red ad hoc. No se profundizará en las demás capas porque esa tarea esta fuera del alcance de este proyecto.

#### **Nodo**

La definición tomada de Millán (2003) nos dice que un nodo es un punto terminal de una red, o cualquiera de sus intersecciones. Su origen está en la raíz indoeuropea *ned*, que en latín produjo *nodus* y que dio origen al vocablo *net* en inglés y a los vocablos *nodo* y *nudo* en español.



**Figura 3-1.:** Modelo de capas del Sistema TLÖN

Por otra parte, en una definición relativa a la teoría de grafos, un nodo de red o vértice de una red, es uno de los objetos que están mutuamente conectados entre sí (Nykamp, s.f.). Las conexiones entre los nodos son llamadas aristas o enlaces.

Según el diccionario de términos de computadoras e Internet (Douglas, Covington, Covington, y Covington, 2009), se tienen dos definiciones:

1. Un computador individual, ocasionalmente un tipo de máquina, en una red.
2. Un punto de conexión en una estructura de datos como las listas enlazadas o árboles<sup>1</sup>.

El grupo TLÖN desarrolló una definición formal para nodo ad hoc, entendiendo el nodo como una representación de una variedad de dispositivos en los sistemas de telecomunicaciones. Un nodo puede representar un computador personal, un servidor, un teléfono celular, o un sensor. Un nodo es una máquina o equipo que permite el envío y recepción de información, que cuenta con un módulo de codificación de fuente, codificación de canal y modulación. Además, tiene la capacidad de guardar información que provenga de las capas superiores del modelo de referencia y puede transmitirla cuando sea necesario.

En este sentido, un nodo ad hoc  $N$  se define como una 5-tupla conformada por los siguientes elementos:

$$\mathcal{N} = \langle MT, P, M, \Theta, V \rangle \quad (3-1)$$

en donde,

<sup>1</sup>Esta definición es usada en el ámbito de las bases de datos.

- $MT$  es una Máquina de Turing definida como  $MT = \langle Q, q_0, F, \Sigma, \Upsilon, b, \delta \rangle$ , en donde:
  - $Q$  : es el conjunto finito de estados internos.
  - $q_0$  : es el estado inicial  $q_0 \in Q$
  - $F$ : es el conjunto de los estados de aceptación.  $\emptyset \neq F \subseteq Q$
  - $\Sigma$  : es el alfabeto de entrada.
  - $\Upsilon$  : es el alfabeto de cinta, que incluye a  $\Sigma$  es decir,  $\Sigma \subseteq \Upsilon$
  - $b \in \Upsilon$ : es el símbolo “blanco” (el símbolo  $b$ , no puede hacer parte del alfabeto de entrada).
  - $\delta$  : es la función de transición de la máquina.  $\delta : Q \times \Upsilon \rightarrow Q \times \Upsilon \times \{\leftarrow, \rightarrow\}$
- $P$  es la interfaz física que permite la interacción con otros nodos, de manera inalámbrica. Permite traspaso de información de control o información general. Maneja la potencia de la interfaz inalámbrica.
- $M$  representa el modelo de movilidad del nodo. Este modelo es una función que se pasa al nodo para que siga un patrón de movilidad. Algunos modelos que se pueden usar son: Random Walk, Random Waypoint, Random Direction Mobility Model, Gauss-Markov, Cite Section Mobility Model.
- $\Theta$  es la función que define el comportamiento del nodo. Esta función indica las distintas operaciones que va a tener a cargo el nodo.
- $V[.]$  es un vector con los posibles roles que puede tener el nodo.
  - $GOD$ : nodo que inicia la red Ad-hoc.
  - $REPLAY$ : nodo que sirve de puente entre dos o más nodos.
  - $SINK$ : es un nodo que solo recibe información.
  - $SOURCE$ : es un nodo que solo transmite información.
  - $Tx/Rx$ : es un nodo que trasmite y recibe información a la vez. Este rol permite entablar los servicios de la red en una comunicación *Full* o *Half dúplex*.

## Red ad hoc

Formalmente, las redes ad hoc son un grafo aleatorio (Newman, 2003) con un conjunto de vértices, comúnmente llamados nodos, en este caso móviles, unidos por un conjunto de enlaces denominados aristas, que cambian de forma dinámica en función del tiempo y las condiciones del ambiente, por ejemplo, las peticiones de los usuarios (TLÖN - Grupo de Investigación en Redes de Telecomunicaciones Dinámicas y Lenguajes de Programación Distribuidos, s.f.).

Una definición más robusta es la que los integrantes del grupo TLÖN establecieron, donde una red ad hoc  $\mathcal{R}$  es equivalente a la 6-tupla de la ecuación 3-2:

$$R = \langle \mathcal{N}, \mathcal{L}, G_{p(l)}, \mathcal{P}, H, \mathcal{I} \rangle \quad (3-2)$$

conformada por los siguientes elementos:

- $\mathcal{N}$  es un conjunto de nodos en la red, donde cada uno tiene las características de un nodo ad hoc.
- $\mathcal{L}$  es un multiconjunto que representa los enlaces existentes entre los nodos de la red. Puede expresarse como el conjunto formado por las duplas  $\{(i, j)_k / i, j \in \mathcal{N} \wedge k \in \mathbb{N}\}$ . Cada dupla  $(i, j)$  está formada por nodos de la red y valor de  $k$  representa la multiplicidad o número de ocurrencias de cada elemento en el conjunto. Los enlaces  $\mathcal{L}$  entre nodos funcionan a través de protocolos para comunicación inalámbrica como la familia 802.11x, LTE, Bluetooth, entre otros.
- $G_{p(l)}$  es un multigrafo aleatorio donde se reúnen todas las posibles configuraciones de los nodos en la red. El valor  $p_{i,j}(l)$  representa la probabilidad de tener un enlace de comunicación  $\mathcal{L}$  entre los nodos  $i, j$ .
- $\mathcal{P}$  es el conjunto de propósitos que puede tener el sistema. Puede entenderse como los diferentes servicios ofrecidos por la red y depende de las limitaciones de recursos, así como de las posibles configuraciones de la red.
- $H$  es un conjunto funciones de entropía para cada uno de los elementos en  $p(\mathcal{P})$ . Estas funciones permiten medir el nivel de organización del sistema en un estado particular. También modelan la probabilidad de la red para ofrecer uno o varios servicios en función de sus condiciones de operación.
- $\mathcal{I}$  es el conjunto de las posibles interacciones entre los nodos (comunicación, cooperación, altruismo, distribución de recursos, etc).

Sin embargo, la definición más cercana al ámbito del proyecto de sistema TLÖN es la siguiente:

Una red ad hoc es una nube móvil, siendo esta una plataforma computacional flexible, dinámica y estocástica que gestiona recursos computacionales distribuidos que son enlazados inalámbricamente; estos a su vez, pueden ser cambiados, movidos, aumentados y en general, combinados de formas novedosas (Fitzek, 2014).

El modelo TLÖN también adopta el concepto de Inmanencia Baruch de Spinoza<sup>2</sup>. Definida como toda actividad o comportamiento inherente al sistema de referencia cuya acción ocurre y perdura en su interior y tiene su razón de ser en ese mismo sistema y además, no es el resultado de una acción exterior a los elementos que la componen, aunque su estudio puede realizarse de forma aislada o

<sup>2</sup>Filósofo holandés, nació en Amsterdam (1632-1677) y es considerado uno de los grandes racionalistas de la filosofía del siglo XVII, junto con Descartes y Pascal.

independiente del sistema.

El grupo TLÖN ha establecido la definición formal de este concepto para el modelo, como la unión de todos los elementos que lo componen. De esta, forma la Inmanencia  $IN$  de TLÖN se define con la ecuación 3-3:

$$IN_{tlön} \equiv \bigcup_{i=1}^n C_i = \langle R, V, LG, S, A, MS, O, T \rangle \quad (3-3)$$

donde cada  $C$  es a su vez un subconjunto. Y los subconjuntos son:

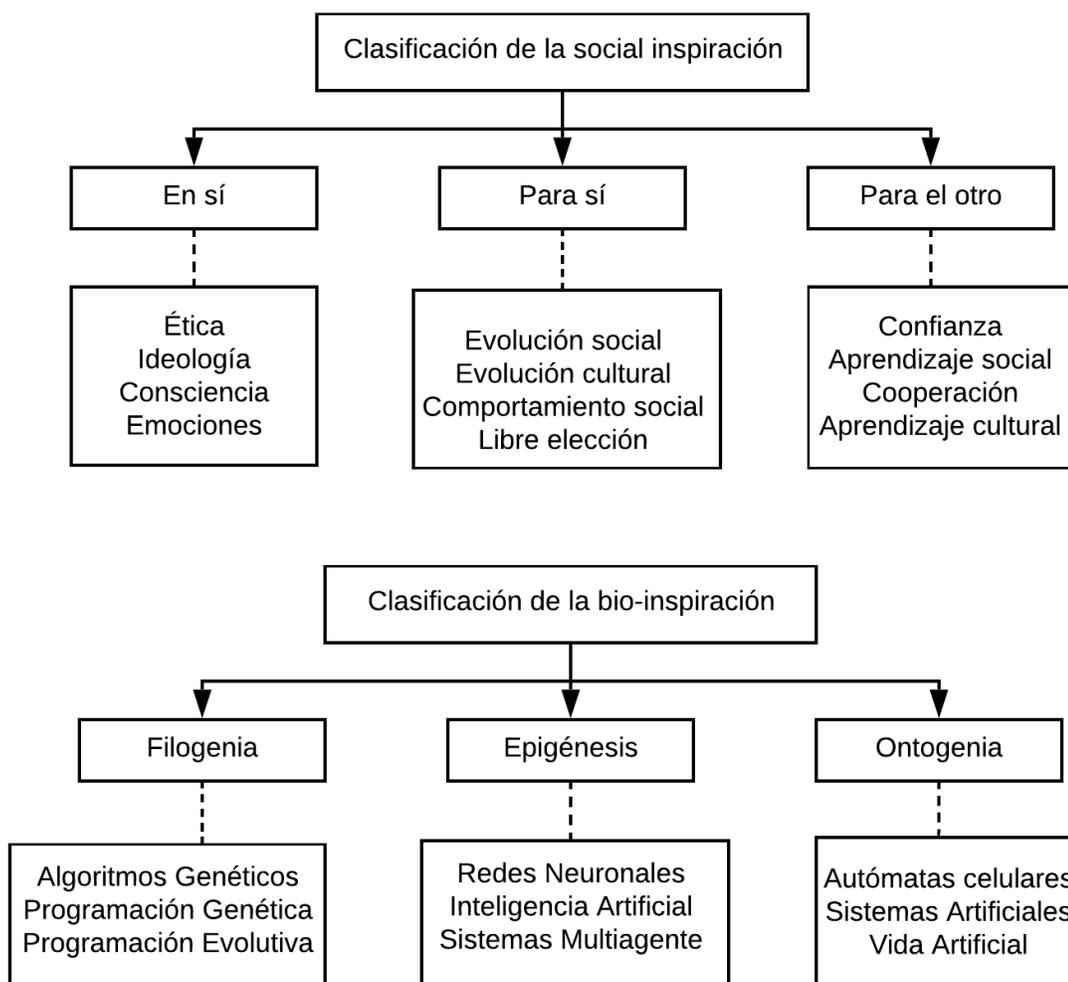
- $C_1 = R$ : Red ad hoc
- $C_2 = V$ : Virtualización
- $C_3 = LG$ : Es el lenguaje de Programación TLÖN.
- $C_4 = S$ : Sistema Multiagente.
- $C_5 = A$ : Aplicaciones.
- $C_6 = MS$ : Modelos sociales.
- $C_7 = O$ : Orquestación
- $C_8 = T$ : Temporalidad.

### 3.3. Modelo social inspirado

De manera semejante a los modelos bio-inspirados, estudiados y aplicados en muchas áreas de las ciencias de la computación, a partir de los cuales han surgido algoritmos que optimizan el funcionamiento de sistemas o aplicaciones, el modelo social inspirado pretende modelar características de la sociedad humana para concebir algoritmos o mecanismos que optimicen especialmente las funcionalidades de los sistemas de computación dinámicos, distribuidos y estocásticos.

El sistema TLÖN plantea una analogía de las tres dimensiones del modelo social inspirado a la vez inspirándose en los ejes taxonómicos que caracterizan a los modelos bio-inspirados (Figura 3-2), donde cada uno de los elementos puede ser superpuesto en las diferentes capas del sistema. Así como en los sistemas bio-inspirados se habla de una Filogenia, para el modelo social inspirado se estudiarán elementos del *En sí*, del ser o existencia pura. En ese caso se analizarán aspectos como la ética, la consciencia y las emociones. Por otra parte, en la bio-inspiración está la epigénesis entonces en la social-inspiración estará el *Para sí*, refiriéndose a todo aquello que se concibe para el propio sistema, para su propio desarrollo, ahí se ubican temas como la evolución social, cultural, el comportamiento social y la libre elección. Por último, si en bio-inspiración tenemos la ontogenia en la social-inspiración

tendremos el eje *Para el otro*. Donde se abarca la dimensión social, el sistema como un "ser" que no es aislado e interactúa con otros. En este último se investigarán temas como la confianza, el aprendizaje social, la cooperación y el aprendizaje cultural.



**Figura 3-2.:** Taxonomía de la social Inspiración y la bio-inspiración. Modificado de Zárate (2016)

En el modelo social cada capa es una abstracción de conceptos de una sociedad real. De esa forma en TLÖN como base está el territorio físico, que corresponde a la capa física. Es donde se establece la red ad hoc, y se encuentran todos los nodos y recursos físicos que conforman la red. Entonces, la red ad hoc es equivalente al "pseudo territorio". Al igual que los territorios de un país tienen diversos recursos naturales y características geográficas diferentes, en una red ad hoc inalámbrica los recursos pueden ser heterogéneos.

Más arriba, en una segunda capa, la virtualización es una abstracción de las instituciones en la sociedad. Estas ejercen funciones y ejecutan acciones que son para el beneficio de la población. En el plano social, implican alguna forma de organización humana, y no son necesariamente lugares físicos establecidos. De manera análoga, esto corresponde acertadamente con la capa de virtualización de TLÖN.

Esta capa permite un mejor manejo de todos los recursos del sistema. Al igual que las instituciones, que ejecutan acciones específicas de acuerdo al grupo de interés o que van dirigidas a ciertos sectores de la sociedad, la virtualización permite desplegar servicios diferentes que apoyen el funcionamiento de las otras capas de TLÖN. En este caso, un esquema de virtualización permite establecer reglas para que las comunidades de agentes puedan ejecutar acciones sobre las interfaces de red o sobre otro recurso y por otra parte, sirve de puente para que diversas aplicaciones puedan ser ofrecidas por la red ad hoc.

La tercera capa corresponde al Sistema Multiagente y está conformada por un grupo de agentes que trabajan cooperativamente. Un agente es una pieza de software que cumple con un servicio y se ejecuta de forma silenciosa y automática. Estos agentes son programas que actúan como una especie de mensajeros, pues además de llevar mensajes, pueden provocar cambios en un sistema, transportar valores de datos, argumentos, procedimientos o resultados que necesitan ser intercambiados a través de los nodos de una red. Por lo general, trabajan en nombre de un individuo, que puede ser un usuario o una organización. Por eso también se les conoce como Robots Software o Softbot (del inglés *Software Robots*) (Zwass, s.f.).

En TLÖN también existe el territorio lógico. Este abarca las acciones y relaciones o asociaciones entre los agentes y el orquestador, en consecuencia las capas de virtualización y de sistema multiagente están fuertemente relacionadas.

Es importante resaltar que un Sistema Multiagente es un conjunto de agentes que trabajan en conjunto, se comunican entre sí e interactúan para resolver problemas que están por encima de sus capacidades y conocimiento individual (Parra Amaris, Checa Hurtado, y Ortíz Triviño, 2015). Debido a su capacidad de autonomía y a la ausencia de control centralizado, los agentes pueden competir, cooperar o simplemente coexistir. Por eso, se puede pensar en la idea de construir mecanismos que permitan solucionar problemas a través de acciones colectivas de un grupo de agentes. Estas características los convierten en una alternativa viable para operar sobre sistemas como las redes ad hoc inalámbricas.

En TLÖN, el comportamiento de cada agente se define mediante el modelo sartreano del ser (Figura 3-3). En esta medida, para modelar sus funciones se tienen en cuenta tres dimensiones:

1. En sí: lo que soy (pasado y presente)
2. Para sí: lo que deseo ser en el futuro
3. Para el otro: lo que representa el ser para el prójimo.

En la sociedad existen personas con ciertos roles como los de los diputados o representantes que desarrollan proyectos en equipo para mejorar el bienestar de la población, así mismo se conforma en TLÖN una capa con comunidades de agentes basadas en la sociedad ideal de Rawls (2012).

En la última capa, las aplicaciones de TLÖN podrían entenderse como formas de un nivel más alto de abstracción en la sociedad. En ese sentido, se podrían modelar bajo los conceptos de cultura, hábitos y costumbres, entre otros. En este aspecto, aún falta investigación, pues esta capa todavía no ha sido desarrollada para el sistema TLÖN y es uno de los grandes retos por enfrentar en futuros trabajos.

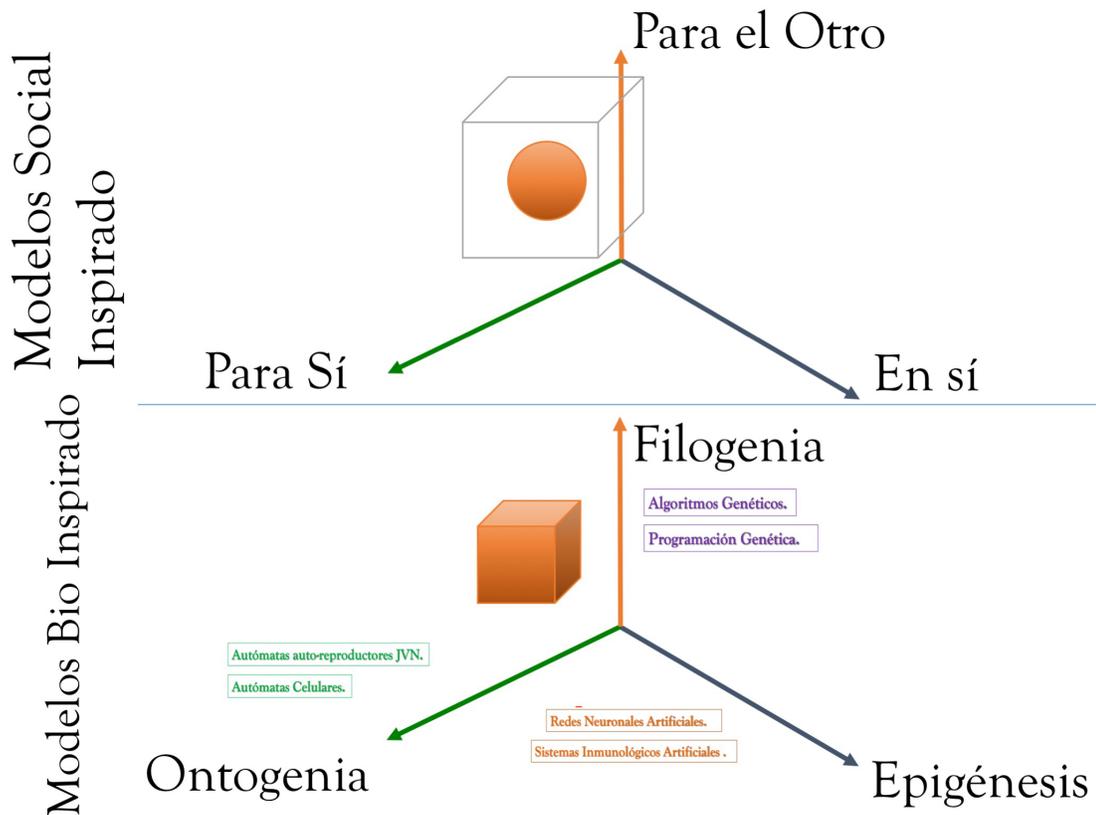


Figura 3-3.: Modelo Sartreano

Por otra parte, el modelo Sartreano también expone la idea de que “*la existencia precede a la esencia*”, entonces para el sistema TLÖN la “*existencia*”, podría ser equivalente a las capas de red ad hoc y de virtualización, que proporcionan el ambiente necesario para desplegar agentes y aplicaciones que serían la “*esencia*” del sistema.

Como este modelo se rige por los principios de justicia e inmanencia, abre las puertas para que emerjan otros principios derivados de estas, como equidad, solidaridad e incluso virtudes como la verdad, dentro de este Sistema Artificial.

### 3.4. Concepto de clúster en el modelo social-inspirado

Este TLÖN, como le llamamos al modelo, es como un planeta informático. Así como otros planetas, este necesita un territorio para existir, pero siendo un mundo virtual dinámico ¿cómo se define el territorio en TLÖN?

Como se menciona anteriormente, la red ad hoc constituye un “pseudo-territorio”. Entonces, en este sentido, es importante analizar el concepto de territorio en nuestra sociedad. El Territorio está asociado al poder sobre el espacio. Un país es un territorio, es decir, un espacio que está regido por normas,

leyes y estructuras de poder que lo delimitan y lo diferencian de otros espacios. Además, el territorio implícitamente asocia el concepto de límites. (Banco de la República, 2017).

El Estado es el que ejerce soberanía sobre su territorio. Aunque los territorios pueden ser países, también al interior de ellos o entre dominios de varios países pueden existir otro tipo de poderes que ejercen soberanía sobre el territorio. Por ejemplo, pueden existir lugares que las personas no tienen la libertad de visitar porque saben que son de dominio de un determinado grupo de personas o compañías multinacionales. También es el caso de las calles cerradas cercanas a entidades públicas, o los barrios peligrosos.

Para tener una visión de las diferentes formas en que se puede definir “territorio”, se estudió el artículo titulado: “*Territorio y territorialidad: Nueva categoría de análisis y desarrollo didáctico de la Geografía*” (Rodríguez Valbuena, 2010). El autor señala que el término territorio se ha usado para designar un espacio geográfico calificado por una pertenencia jurídica, esto significa un espacio político donde se ejerce la autoridad de un Estado, o de una entidad administrativa a menor escala (departamental, municipal o indígena). Otra forma en que se puede ver es como “un concepto relacional que insinúa un conjunto de vínculos de dominio, de poder, de pertenencia o de apropiación entre una porción o la totalidad del espacio geográfico y un determinado sujeto individual o colectivo”, de esta forma el espacio geográfico está revestido de dimensiones políticas, afectivas y de identidad o de su sumatoria.

El autor también muestra otro punto de vista a partir de lo que dijo Dimeo (1993), quien definió territorio desde una visión marxista clásica como un fragmento espacial donde se fusionan tres tipos de estructuras:

- Infraestructura = espacio físico que incluye el componente humano y sus actividades económicas
- Superestructura = los campos políticos ideológico y simbólico.
- Metaestructura = representada por la relación entre el individuo y el espacio.

Además de estas definiciones que implican relaciones de poder o soberanía y pertenencia, es propicio resaltar dos características que el autor expone en el artículo sobre el argumento de Montañez y Delgado, porque merecen especial atención dada la correspondencia con lo que plantea el sistema TLÖN, Estas son:

1. “Un territorio es desigual en el espacio, ahí ocurren y se sobreponen distintas territorialidades locales, regionales, nacionales y mundiales, con intereses distintos, con percepciones, valoraciones y actitudes territoriales diferentes, que generan relaciones de complementación, de cooperación y conflicto” (Montañez y Delgado citado en Rodríguez Valbuena, 2010, p.7).
2. “El territorio no es fijo, sino móvil, mutable, desequilibrado”, entonces, “la realidad geosocial es cambiante y requiere nuevas formas de organización territorial” (Rodríguez Valbuena, 2010, p.7).

La primera característica implica plantear que una red ad hoc puede tener territorios superpuestos que serían los clústeres. Cada clúster podría tener diferentes utilidades y tamaños según el alcance de su soberanía o capacidad. Entre estos se pueden desarrollar relaciones de cooperación, es decir, unos podrían complementar las acciones de otros, especializándose en tareas diferentes, pero es probable que esto también genere conflictos, que hay que prever en la red. Por ejemplo, un nodo podría pertenecer a diferentes clústeres tanto a nivel lógico como físico y el sistema debe tener la inteligencia para manejar este tipo de situaciones.

La segunda característica soporta la afirmación de que los clústeres serán móviles y cambiantes y que por eso requerirán que se ejecute de forma iterativa el mecanismo de reorganización y definición de los mismos. No obstante, las estructuras de control sobre un clúster deben relevarse constantemente.

Otros puntos importantes que se resaltan del artículo son:

- “La inteligencia territorial”, que suscita el entendimiento del territorio como una tarea colectiva que supone esfuerzo para abarcar la realidad desde el conocimiento y la acción.
- Que el espacio geográfico es esencialmente dinámico y heterogéneo sin importar cual sea su dimensión o su escala.
- El espacio es un producto social, de acuerdo con lo que dice Isnard(1978): “el espacio geográfico nace de la iniciativa humana y expresa el proyecto propio de cada sociedad.”
- El territorio lo integran un conjunto de elementos que lo definen. A partir la existencia de un espacio geográfico en el cual un sujeto o grupo ejerce algún tipo de dominio, estos elementos pueden ser objetivos, subjetivos, materiales o inmateriales y son construidos por los organismos sociales a partir de las proyecciones colectivas e individuales.

Por último, hay que tener en cuenta que la soberanía de un territorio no solo abarca las superficies, ya que el espacio también involucra otro tipo de límites, como los marítimos y aéreos. De manera análoga, en un clúster o territorio ad hoc, se ejerce soberanía sobre dos tipos de espacios, el lógico y el físico.

Todas las anteriores proposiciones concuerdan con la descripción del modelo TLÖN. En este sentido, la red ad hoc es un *pseudo-territorio* donde una estructura de poder es la virtualización y los orquestadores estos ejercen el dominio. Los nodos y los agentes serán los organismos sociales que lo construyen mediante sus interacciones y actividades dentro de ese espacio, pero el espacio de dominio a su vez está limitado por la conformación de uno o más clústeres.

### **Administración del territorio a nivel de país**

Ahora que se ha planteado el concepto de territorio, es importante examinar como se administra ese territorio en nuestro modelo de Estado. Por ejemplo, en Colombia, la definición de los límites territoriales del país a nivel geopolítico se estableció por la Constitución Política de 1991. La administración

de las estructuras y dinámicas dentro del territorio colombiano se establece de acuerdo con lo que dicta la Ley Orgánica de Ordenamiento Territorial (Ley 1454 de 2011).

El ex-Ministro del Interior y de Justicia Germán Vargas-Lleras, sobre esta ley expresa que: es el parámetro legal para la construcción de un modelo de gobierno local descentralizado, que facilite el cumplimiento de los fines esenciales del estado. Esta promueve la integración territorial, la competitividad, la participación ciudadana y la inversión estratégica de los recursos públicos, además defiende la unidad nacional y pretende mantener la integralidad y autonomía territorial (Ministerio del Interior y de Justicia, 2011).

### **Aspectos principales de Ley Orgánica de Ordenamiento Territorial**

Según esta ley, en el territorio se crearán formas flexibles de integración territorial, que son las Regiones de Planeación y Gestión y las Regiones Administrativas y de Planificación. Estas se ven como modelos de burocracia cero, definidas para promover el desarrollo y la competitividad. Además, se implementarán zonas de inversión para superación de pobreza y marginalidad, estas son instrumento de apoyo a la gestión de territorial permite mejor distribución (más proporcional) de recursos de los fondos de la nación.

Se podrán conformar provincias administrativas y de planificación que serán la base para la gestión municipal dentro de departamentos y para armonizar el disperso desarrollo municipal en torno a una planeación estratégica sobre intereses comunes. Se tendrá un modelo de integración regional que promueva alianzas estratégicas entre entidades territoriales, que eviten duplicidad de tareas y esfuerzos aislados de territorios que tengan condiciones más o menos homogéneas, para generar economías de escala y proyectos productivos, así como diversificación de competencias. Esto permite asignar a cada tipo de departamento o municipio competencias especiales, diferenciadas y que potencien sus ventajas comparativas (turismo, agroindustria, minería, etc.). De esta forma, esta ley general de principios, pretende regular la delegación competencias de la nación a las entidades territoriales y es la base para regulaciones posteriores.

Tomando como referencia lo anterior, se puede establecer una analogía para el sistema TLÖN, en donde los clústeres constituyan las entidades territoriales y provincias administrativas para la gestión de los recursos, que permitan crear zonas de diferenciación de tareas y permitan desplegar los servicios y aplicaciones de diferentes formas. Serán los clústeres físicos y lógicos los que permitirán la delegación de competencias en el sistema. Y los principios rectores que se expresan dentro de la Ley de Ordenamiento Territorial podrán apropiarse para establecer las características que deben cumplir las entidades o clústeres dentro del sistema TLÖN.

### Analogía de los principios rectores del ordenamiento territorial con las características de un territorio ad hoc o clúster

Se usará el concepto de Entidad Territorial (ET) o Región como el equivalente de un clúster en la red ad hoc. En la tabla 3-1 se exponen los principios de rectores de Ley de Ordenamiento Territorial con sus equivalentes en la red ad hoc.

| Principio                         | Objetivo/Característica en territorio Nacional  | Requerimiento/característica en Red Ad hoc de TLÖN   |
|-----------------------------------|---|--|
| Soberanía y unidad nacional       | Propiciar integridad territorial, seguridad y defensa. Garantizar una República unitaria descentralizada y la autonomía de cada ET. | Debe garantizar que la red sea segura, descentralizada y debe promover la autonomía en cada clúster. |
| Autonomía                         | Cada ET tiene autonomía de gestión de intereses   | Autonomía de administración de recursos intra-clúster  |
| Descentralización                 | Distribuir las competencias entre ET y esquemas asociativos. Escalar poder de decisión.   | Debe permitir escalar las responsabilidades en la red y agrupar recursos y dividir tareas.           |
| Integración                       | Ejecutar programas de cooperación entre regiones fronterizas.   | Permitir la comunicación inter-clúster.  |
| Regionalización                   | Establecer regiones de planeación y gestión, regiones administrativas y de planificación  | Permitir la subdivisión de la red en clústeres según los recursos a administrar.                     |
| Sostenibilidad                    | Conciliar el crecimiento económico, la sostenibilidad fiscal, la equidad social y la sostenibilidad ambiental.                      | Debe ser capaz de mantener el equilibrio cuando la red crece.  |
| Participación                     | Promover participación, concertación y cooperación.   | Debe permitir implementación de mecanismos de cooperación entre agentes.                             |
| Solidaridad y equidad territorial | Regiones de mayores capacidades apoyarán a regiones de menor desarrollo relativo. Procurar acceso equitativo a oportunidades.       | Clústeres de mayor capacidad deben poder ayudar a clústeres de menor capacidad.                      |

|                                 |   |  |
|---------------------------------|---|--|
| Diversidad                      | Reconocer diferencias geográficas, institucionales, económicas, sociales y culturales   | El clúster debe tener la capacidad de reconocer e incluir equipos de características heterogéneas.   |
| Gradualidad y flexibilidad      | Ajustar la Ley de Ordenamiento Territorial a las diferentes formas de división territorial. Adaptar progresivamente las entidades para permitir asignación flexible de recursos y competencias. | Un clúster podrá aumentar sus capacidades de administración de recursos progresivamente y sus competencias o responsabilidades pueden variar en el tiempo.   |
| Prospectiva                     | Tener visión compartida de país a largo plazo   | El mecanismo de clustering debe considerar una planeación a largo plazo.   |
| Paz y convivencia               | Promover esfuerzos de convivencia pacífica, impulsar políticas y programas de construcción de paz y fortalecimiento del tejido social.  | Debe promover el desarrollo de tareas distribuidas sin causar conflictos. Debe fortalecer la red   |
| Asociatividad                   | Promover la formación de asociaciones entre ET para generar sinergias, alianzas y objetivos comunes.  | El clúster debe promover la interacción con otros clústeres de la misma naturaleza. Dentro de un clúster se debe permitir la asociación de entidades para prestar servicios y funciones de administración comunes. |
| Responsabilidad y transparencia | Promover el control social de la gestión pública, rendición final de cuentas, responsabilidad política y administrativa de asuntos públicos.  | El clúster debe tener un mecanismo de reportes.  |
| Equilibrio social y territorial | Crear instrumentos para superar desequilibrios entre regiones   | El sistema debe contar con un mecanismo para balancear la carga en el sistema y repartir carga entre clústeres.  |

|                          |   |   |
|--------------------------|---|---|
| Economía y buen gobierno | Promover mecanismos asociativos que privilegien la optimización del gasto público y el buen gobierno en su confirmación y funcionamiento. | Los clústeres deben conformarse (planificarse) estratégicamente para asociar recursos que permitan una administración óptima. |
| Multietnicidad           | Permitir a las minorías ejercer su derecho de planeación y gestión dentro de la ET.   | Permitir a los nodos de escasos recursos la participación dentro de la red.   |

**Tabla 3-1.:** Analogía de los principios rectores del ordenamiento territorial con las características de un clúster una red ad hoc.

### Esquemas asociativos territoriales

El Estado promoverá esquemas asociativos de tipo administrativo y político que hagan posible la libre y voluntaria conformación de alianzas estratégicas entre dos o más entes territoriales, para organizar conjuntamente la prestación de servicios y funciones administrativas asignadas o ejecución de obras, que impulsen el desarrollo autónomo y autosostenible de las comunidades en aspectos sociales, económicos y culturales. Estos esquemas asociativos se efectúan mediante alianzas o bien por medio de convenios y contratos plan (Ministerio del Interior, 2011).

Según la ley de Ordenamiento Territorial (Ley 1454 de 2011), existen los siguientes tipos de asociaciones:

- Alianzas entre entidades territoriales nacionales y las de países vecinos y fronterizos;
- entre regiones administrativas y de planificación;
- entre regiones de planeación y gestión;
- asociaciones de departamentos;
- asociaciones de áreas metropolitanas donde se asocian dos o más áreas de este tipo, ya sean de un mismo departamento o de más departamentos, estas áreas metropolitanas actúan como instancias para el desarrollo municipal;
- asociaciones de distritos especiales que se dividen en localidades, tienen alcaldes locales y cada cuenta con presupuesto propio;
- provincias administrativas y de planificación, una provincia puede estar conformada por dos o más municipios geográficamente contiguos de un mismo departamento; y
- asociaciones de municipios.

Para complementar el modelo social-inspirado del proyecto TLÖN, puede ser conveniente abstraer a nivel informático la representación de los tipos de asociaciones mencionados para ser implementados en las capas de red, capa de virtualización y capa de sistema multiagente.

### Definición de territorio para el sistema TLÖN

El territorio ad hoc del sistema TLÖN es el conjunto de recursos hardware y software que se encuentran una determinada área de cobertura, este será un plano físico. Este puede subdividirse en entidades territoriales o clústeres que permiten una descentralización de la administración de la red, del control de los recursos y de la ejecución de tareas. Para TLÖN se definen dos esquemas principales de asociaciones territoriales: clústeres lógicos y clústeres físicos (Figura 3-4). Estas asociaciones deben cumplir los principios rectores del ordenamiento territorial de una red ad hoc de TLÖN.

El clúster físico define divisiones a nivel tangible entre entidades; en este caso, al agrupar nodos que pertenecen a diferentes clústeres físicos, conformando el plano de red.

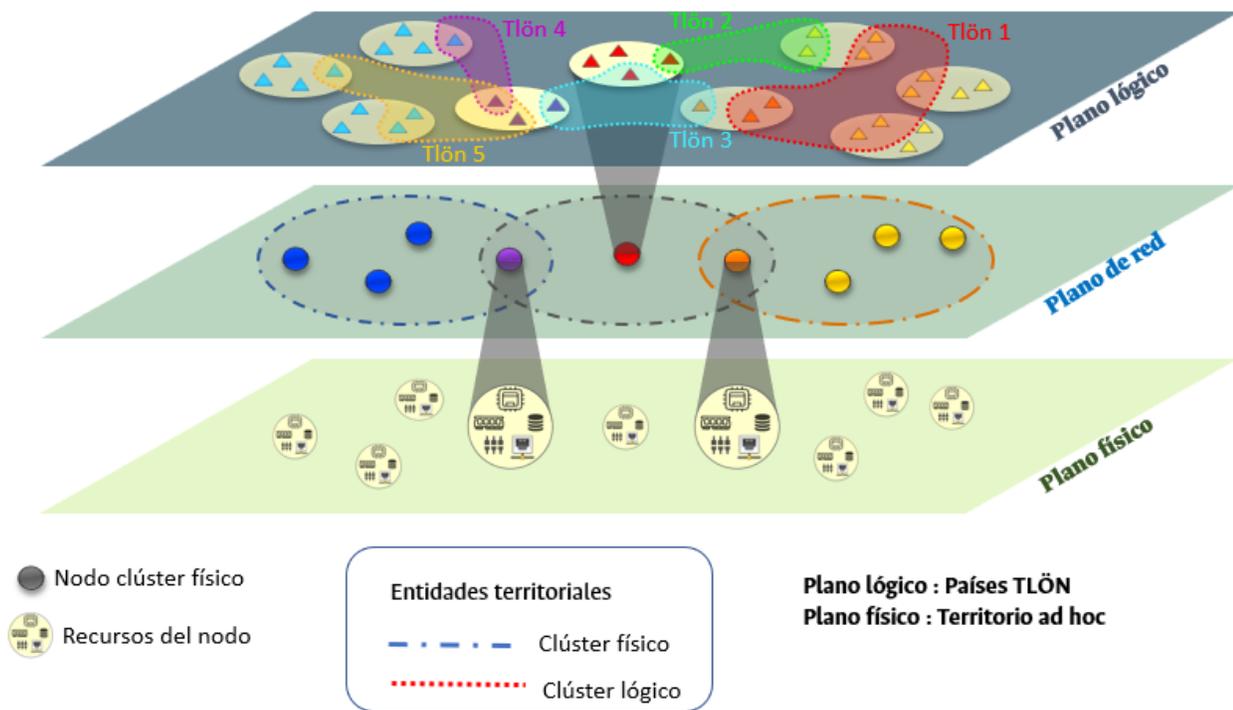


Figura 3-4.: Entidades territoriales para el modelo TLÖN

En el plano lógico tienen lugar los clústeres lógicos. Un clúster lógico define divisiones mediante agrupaciones de entidades lógicas; en este caso, para el procesamiento distribuido se asocian agentes dentro de diferentes nodos. A estos se les denominará Países TLÖN. En la Figura 3-4 son las agrupaciones demarcadas con línea punteada fina de colores rojo (TLön 1), verde (TLön 2), cian (TLön 3), púrpura (TLön 4) y amarillo (TLön 5).

Por lo tanto, a un clúster físico se pueden superponer distintas formas de territorialidades lógicas o clústeres lógicos, porque en el sistema TLÖN, los países en términos de territorio lógico no están restringidos a las fronteras de los clústeres físicos.

### **Movilidad entre territorios**

Los individuos pertenecientes a un mismo país se pueden mover con tranquilidad de una región a otra, de un departamento a otro, o dentro de los límites territoriales del mismo. Sin embargo, esto no se puede hacer tan fácilmente cuando se quiere viajar, moverse de un país a otro o establecerse en países de otros continentes. Para que un individuo pueda moverse entre naciones, necesita una visa o permiso. En este sentido, se requiere que entre los países de origen y destino existan alianzas definidas o algún tipo de convenio, acuerdo o contrato. El Ministerio de Relaciones Exteriores es entonces el encargado de establecer los parámetros y leyes de inmigración y de fijar los términos para adoptar acuerdos internacionales, acuerdos bilaterales o multilaterales como el Pacto Mundial para la Migración de la ONU o la Alianza del Pacífico.

El modelo social-inspirado del sistema TLÖN también apropia los postulados asociados al concepto de Estado definidos por el filósofo inglés Thomas Hobbes. No obstante, si analizamos que las leyes que recaen sobre un individuo fuera del territorio nacional son diferentes a las leyes de su país de origen. Esto concuerda con lo que afirma Hobbes en su libro *Leviatán* (Hobbes, 2001), que si un soberano envía a alguien como mensajero o es autorizado para realizar un viaje, sigue siendo súbdito, pero lo es por contrato entre soberanos. Además quien entra en los dominios de otro soberano queda sujeto a todas las leyes de ese territorio, a menos que tenga un privilegio por concesión del soberano o por licencia especial.

Entonces podríamos decir que si se considera al sistema TLÖN como un planeta, donde existen muchos países o territorios ad hoc, que llamaremos clústeres, donde cada un orquestador como gobernante, entonces, los agentes, súbditos de un orquestador A, que pertenecen a un clúster A, pueden moverse hacia otro territorio o clúster B, gobernado por el orquestador B, mediante un acuerdo de movilidad entre los orquestadores A y B. En este sentido, en el territorio B, el agente de A será un agente foráneo.

### **Ministerio de Relaciones Exteriores del sistema TLÖN**

Se puede afirmar que a nivel físico un clúster TLÖN podría ser equivalente a un territorio o espacio geográfico y un clúster lógico equivalente a un país virtual. Por consiguiente, para administrar la movilidad de nodos o agentes de un país hacia otros, se debe crear una entidad relacionada que cumpla funciones semejantes a las del Ministerio de Relaciones Exteriores. Este tiene entre sus funciones dirigir y coordinar la expedición de visas y ejecutar procesos que permitan un manejo efectivo de la dinámica migratoria. Teniendo en cuenta lo anterior, se considera que la definición detallada de las

funciones del Ministerio de Relaciones Exteriores del sistema TLÖN que habilite la movilidad y funcionamiento de entidades en clústeres foráneos podría ser un tema para un trabajo futuro.

## 4. Modelo teórico de clúster para el sistema TLÖN

Un clúster es la agrupación de un conjunto de recursos finitos, que pueden físicos o lógicos. En sistemas distribuidos como el sistema TLÖN estas agrupaciones suceden en la capa de red ad hoc y en la capa de sistema multiagente y son el resultado de interacciones a nivel espacial y temporal de los nodos y entidades. Durante el ejercicio investigativo se encontró que una alternativa para el modelamiento de estas interacciones es mediante el modelo de bigrafos diseñado por el profesor Robin Milner de la Universidad de Cambridge (Milner, 2009), porque establece un formalismo para modelar sistemas distribuidos y sistemas ubicuos que evolucionan en el tiempo, teniendo en cuenta el parámetro espacial y la conectividad. En consonancia con lo anterior, en este capítulo se presentará un modelo teórico de clúster físico y lógico para el sistema TLÖN y se explicará mediante bigrafos (definidos el Anexo B).

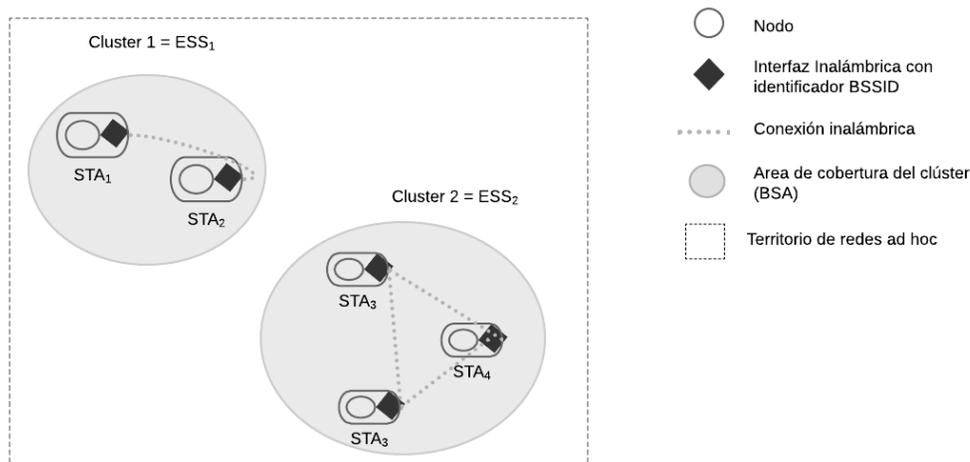
### 4.1. Definición de clúster físico en el sistema TLÖN

En el estándar IEEE 802.11, se define una estación inalámbrica (STA) o nodo inalámbrico como una unidad direccionable, es decir, un destino con dirección. Esto no significa que tenga una ubicación fija, sino que al contrario, en las redes inalámbricas las STA se mueven o pueden portarse de un lugar a otro. El estándar también establece que el conjunto básico de servicios (BSS) es el bloque de construcción principal de una red inalámbrica y se define como el conjunto de puntos de acceso (AP) y sus STAs asociadas, que tienen en común un identificador único del conjunto básico de servicios (BSSID). Los conjuntos de servicios básicos se pueden clasificar en modo infraestructura, cuando requieren de una estación central (o unidad mediadora) que comunica a las STAs, y en modo independiente (IBSS), conocidos coloquialmente como redes ad hoc. En ese caso, no existen APs y las estaciones pueden comunicarse entre sí directamente. De esta manera, puede decirse que en una red ad hoc o IBSS una STA es al mismo tiempo un AP. En consecuencia, el BSSID equivale al identificador o dirección MAC de la interfaz inalámbrica de un nodo miembro en una red ad hoc.

Por otra parte, el estándar define un conjunto de servicios extendido (ESS) como un tipo de red que permite la extensión del área de cobertura. Este agrupa uno o más BSS con el mismo identificador de servicio extendido (ESSID), que no es más que el nombre que identifica una red.

Con base en las anteriores definiciones, se establece que un ESS es una unidad de asociación en una red ad hoc y es equivalente a un clúster, dando lugar a la definición de clúster físico.

Un clúster físico en una red ad hoc es un grupo conformado por al menos dos estaciones inalámbricas (STAs) con sus respectivas direcciones físicas (BSSIDs), que pueden comunicarse entre sí por el medio inalámbrico y que están asociadas mediante un mismo nombre de red (ESSID).



**Figura 4-1.:** Ejemplo de dos clústeres físicos independientes

En la Figura 4-1, se muestra un ejemplo de dos clústeres independientes. El área en donde es posible desplegar redes ad hoc se denominará territorio ad hoc. En esta sucede la asociación de estaciones ad hoc inalámbricas en forma de clúster, donde cada uno tiene un área de cobertura determinada que limita su acción. Según el estándar 802.11, el área de cobertura de la transmisión de un STA hacia otro se conoce como transmisión direccional y el conjunto de todas las posibles transmisiones direccionales de un STA es el área de servicio básico (BSA) (IEEE Computer Society, 2012). Tanto el clúster 1 como el clúster 2 se componen de redes ad hoc que cuentan con dos y tres estaciones inalámbricas respectivamente. En este caso, cada estación está conformada por el nodo miembro del clúster o equipo físico que tiene un conjunto de recursos de hardware, además de una interfaz de comunicación inalámbrica.

#### 4.1.1. Proceso de conformación de clúster físico

Para comenzar la descripción del procedimiento de conformación de un clúster físico en una red ad hoc en el contexto del sistema TLÖN, se parte de la generación de un territorio ad hoc. Se asume que en este territorio es posible la creación de redes ad hoc porque los nodos o STAs dentro de este tienen el modo ad hoc activado en sus tarjetas de red inalámbricas y son potenciales participantes dentro del sistema.

Los nodos tendrán nombres diferentes según su participación dentro de la red. De esta forma encontraremos:

1. **Nodo desvinculado:** es una STA que está dentro del área de cobertura de una red ad hoc. Este percibe la señal de notificación de la existencia de una red ad hoc, pero no está vinculado a esta.

2. Nodo originador: también denominado "nodo Adán", es una STA que inicia una red ad hoc y para invitar a otros nodos a participar, publica el correspondiente ESSID mediante mensajes broadcast.
3. Nodo vecino: es una STA que esta vinculada a la red ad hoc y se encuentra a un salto de otro nodo miembro de la red.
4. Nodo gateway: en este modelo, se denomina nodo gateway en el territorio ad hoc a una STA con dos o mas interfaces inalámbricas que pertenece a dos o más clústeres. Es importante aclarar que el término gateway en otros ambientes de redes se usa para referirse al equipo que es puerta de acceso a internet, pero no representa lo mismo en este modelo. Aunque este nodo sí podría adoptar ese rol para brindar acceso a Internet a sus nodos asociados. En este modelo, la característica más importante de este nodo es permitir la interconexión entre varios clústeres.

En el procedimiento de conformación de un clúster, se deben presentar las siguientes fases:

- Fase 1. Un nodo originador activa su interfaz en modo ad hoc y establece los siguientes parámetros para definir el clúster físico: ESSID o nombre de la red, tamaño máximo de las tramas (MTU) y canal o frecuencia de operación. Este nodo publica en el territorio ad hoc el ESSID mediante broadcast de mensajes originadores a través de su interfaz inalámbrica.
- Fase 2. Un nodo cercano que se encuentra en el territorio ad hoc recibe los mensajes del nodo originador y acepta la invitación a vincularse a la red identificada con el ESSID correspondiente.
- Fase 3. El nodo originador y el nuevo nodo miembro ahora conforman un clúster identificado con ESSID en el territorio ad hoc. El nuevo nodo una vez esta vinculado al clúster se convierte en originador, esto sucederá con cada nuevo nodo miembro.
- Fase 4. Los dos nodos del clúster repiten el proceso de propagación de mensajes originadores para publicar la existencia del clúster al que pertenecen y el proceso vuelve a empezar. Este ciclo se puede repetir  $n$  veces hasta acabar los espacios de direccionamiento o durante el tiempo de vida de la red si no se especifica algún parámetro de calidad de servicio que establezca el número máximo de nodos por clúster.

Para la conformación de  $n$  clústeres en el territorio ad hoc, se deben presentar los siguientes eventos adicionales a los del procedimiento anterior:

- Fase 5. Un nodo con dos o más interfaces inalámbricas que llega al territorio ad hoc es un potencial nodo gateway. A través de una de sus interfaces, se vincula a un clúster existente con identificador ESSID.
- Fase 6. El nodo ahora es miembro del cluster  $ESSID_1$  a través de su interfaz  $w_1$ . Sin embargo, este tiene la capacidad de conformar otro clúster o  $n$  clústeres adicionales según la cantidad

de interfaces inalámbricas con las que cuenta. Debido a esto, un nodo de estas características tendrá un grado de jerarquía mayor que la de los nodos que cuentan con una sola interfaz inalámbrica. Entonces, el nodo gateway inicia el proceso de generación de un clúster  $ESSID_i$  por la interfaz  $w_i$ . El ciclo vuelve a repetirse por cada interfaz y según las necesidades del sistema.

Las definiciones y conceptos básicos asociados a bigrafos propuestos por Milner (2009) se encuentran en el Anexo B. A continuación, se detallarán estos procedimientos usando el formalismo de bigrafos propuesto por Milner(2009) para modelar la evolución espacial y temporal de los sistemas distribuidos y, de forma adicional, se usa la extensión del formalismo de sistemas reactivos de bigrafos estocásticos de prioridad con superposición (*Priority Stochastic Bigraphical Systems with sharing*, PSBRS) propuestos por Calder y Sevegnani (2013). En PRBRS, se introduce una operación de bigrafos adicional a las originalmente propuestas por Milner, para poder representar la situación en que dos nodos comparten el mismo espacio o cuando las redes inalámbricas se superponen en la misma área de cobertura. En el caso de los clústeres físicos y lógicos, se apropia este formalismo porque se reconoce como una alternativa para modelar el hecho de que dos clústeres se pueden superponer cuando un nodo pertenece a dos o mas clústeres al mismo tiempo. En el plano lógico, cuando uno o mas clústeres lógicos se superponen es porque tienen elementos en común en donde se comparten procesos sobre el mismo recurso. En este sentido, es importante recordar que en el sistema TLÖN los territorios lógicos no están supeditados a las fronteras de los clústeres físicos o a las fronteras de operación de un solo nodo. Son entidades territoriales dinámicas y sus asociaciones son a nivel de recursos virtualizados.

#### 4.1.2. Modelo de clúster físico con bigrafos

La región ad hoc o espacio donde se encuentra un nodo originador se representa con un rectángulo de líneas discontinuas. El óvalo representa el nodo físico y el círculo dentro del mismo corresponde a un sitio físico, que en este caso es la tarjeta de red para interconexión inalámbrica. En esta se pueden encontrar dos tipos de interfaces físicas o lógicas para habilitar la interconexión con otros componentes. El rombo dentro del círculo representa el controlador de la interfaz inalámbrica. El triángulo representa el control que efectúa una determinada acción sobre las interfaces de la tarjeta de red. Los puntos en los bordes del nodo representan los puertos que son puntos de posibles conexiones. Las conexiones o enlaces se representan con líneas de color verde. Los nombres se identifican con letras y representan enlaces o posibles enlaces hacia otros bigrafos. Son nombres externos que se publican en el entorno.

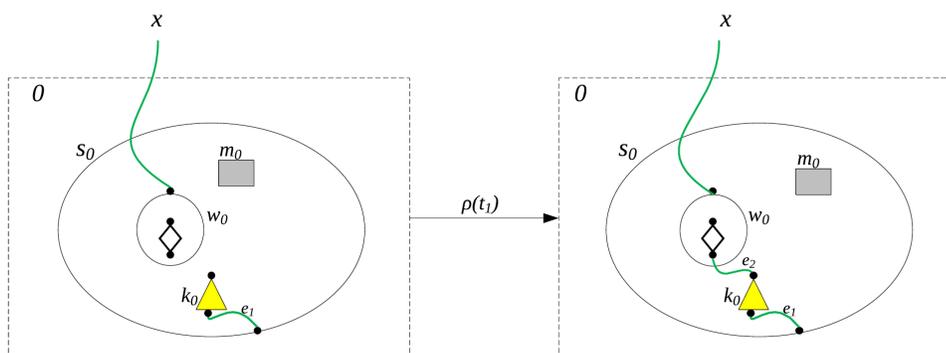
En la tabla 4-1 se listan las convenciones para la notación de los elementos de los bigrafos para este modelo.

**Tabla 4-1.:** Elementos del bigrafo

| Elemento | Descripción                                 |
|----------|---|
| $S_i$    | Nodo físico $i$                             |
| $w_i$    | Tarjeta de red inalámbrica $i$              |
| $k_i$    | Control del nodo $i$                        |
| $m_i$    | Recursos del nodo $i$                       |
| $b_i$    | Mensajes broadcast enviados por el nodo $i$ |

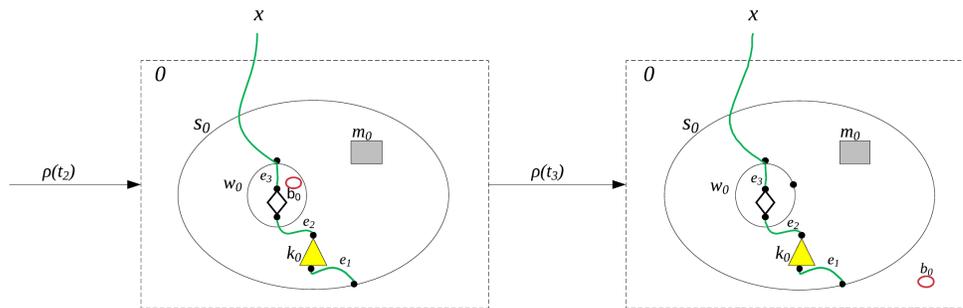
### Fase 1

En la Figura 4-2, se representa el primer estado para la formación de un clúster. En un tiempo inicial  $t_0$  en la región 0, el nodo  $S_0$  cuenta con una tarjeta inalámbrica  $w_0$  que está inactiva, pero tiene la capacidad de activarse en diferentes modos de operación, de manera que el nodo es un potencial nodo originador. El nodo también cuenta con el control  $k_0$  enlazado mediante una conexión  $e_1$ . La transición al segundo estado en un tiempo  $t_1$  sucede cuando el control  $k_0$  envía una señal que activa el modo ad hoc de la interfaz inalámbrica de  $w_0$ . Esta acción se representa con el enlace  $e_2$  entre la interfaz y el control.



**Figura 4-2.:** Formación de un clúster. Fase 1:  $t_1$  activación de modo ad hoc

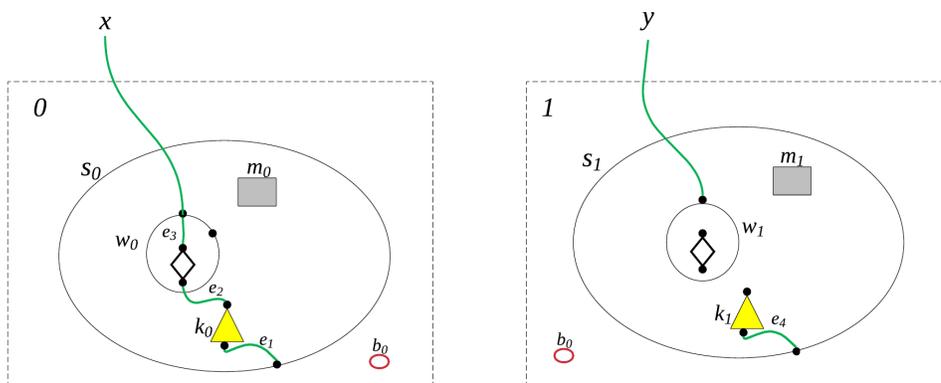
En la Figura 4-3, se representa la transición del tiempo  $t_1$  al tiempo  $t_2$ , en el que se llega a un tercer estado donde el nodo  $S_0$  realiza la configuración de los siguientes parámetros: MTU, el nombre del clúster (ESSID) representado por los nombres externos; la dirección física (MAC) y el canal de transmisión. Enseguida se activa la interfaz inalámbrica de  $w_0$  y esta interacción se simboliza con el enlace  $e_3$ . Con base en la configuración realizada se generan los mensajes broadcast originadores  $b_0$ . La Figura 4-3 también representa la transición a un cuarto estado en el tiempo  $t_3$  en donde los mensajes broadcast generados por la interfaz de  $S_0$  se difunden fuera del nodo en la región ad hoc 0 y se activa una interfaz virtual que está disponible para establecer futuras conexiones con otros nodos.



**Figura 4-3.:** Formación de un clúster. Fase 1: transición al tiempo  $t_2$ , configuración de parámetros y activación de interfaz inalámbrica  $w_0$ . Transición al tiempo  $t_3$  y propagación de mensajes broadcast en la región 0.

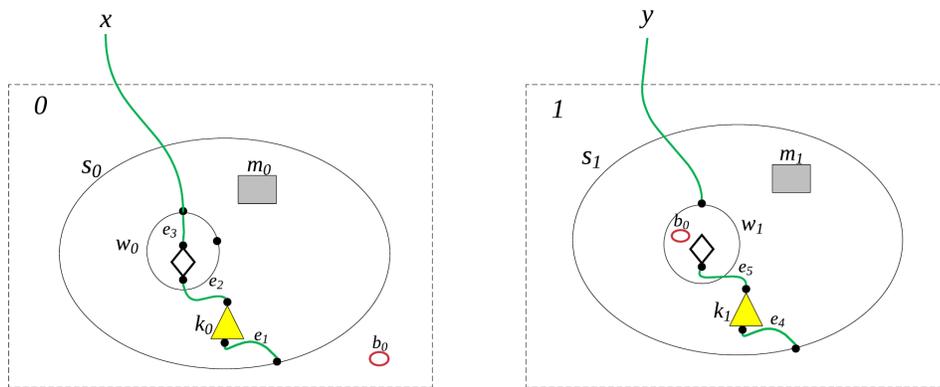
## Fase 2

La Figura 4-4 muestra el quinto estado en el tiempo  $t_4$  donde empieza la fase 2, con la llegada de un segundo nodo  $S_1$ , de características similares a  $S_0$ , a cercanías de la región 0. En este estado,  $S_1$  es un nodo desvinculado del clúster y ubicado en la región 1, con una interfaz inalámbrica en  $w_1$  y el control  $k_1$ . Los mensajes broadcast  $b_0$  enviados por  $S_0$  han llegado hasta la región donde se encuentra  $S_1$ .



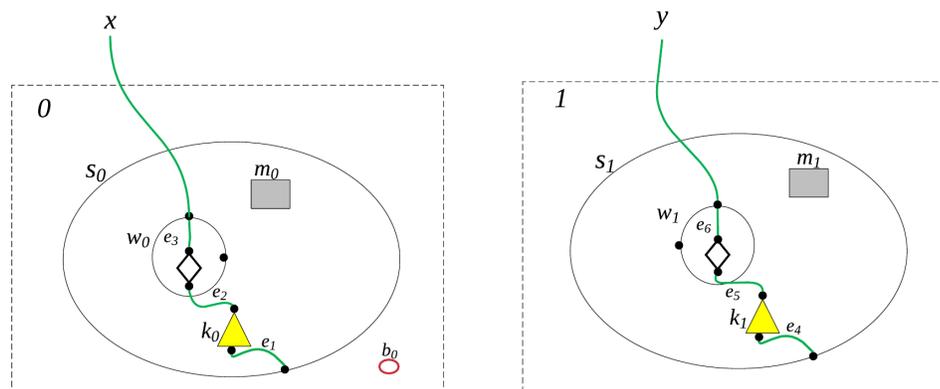
**Figura 4-4.:** Formación de un clúster. Fase 2:  $t_4$  llegada de un nuevo nodo a las cercanías de la región 0.

La Figura 4-5 representa la fase 2 en el cambio que se presenta en el tiempo  $t_5$ . Cuando el control  $k_1$  del nodo  $S_1$  activa el modo ad hoc ( $e_5$ ) y  $w_1$ , procesa el mensaje  $b_0$  y posteriormente acepta la invitación a unirse al clúster identificado con el *ESSID* publicado por  $S_0$ . En este bigrafo, las letras  $x$  y  $y$  son los nombres externos que representan los identificadores de clústeres que un nodo podría publicar en el medio.



**Figura 4-5.:** Fase 2:  $t_5$  activación del modo ad hoc de  $S_1$

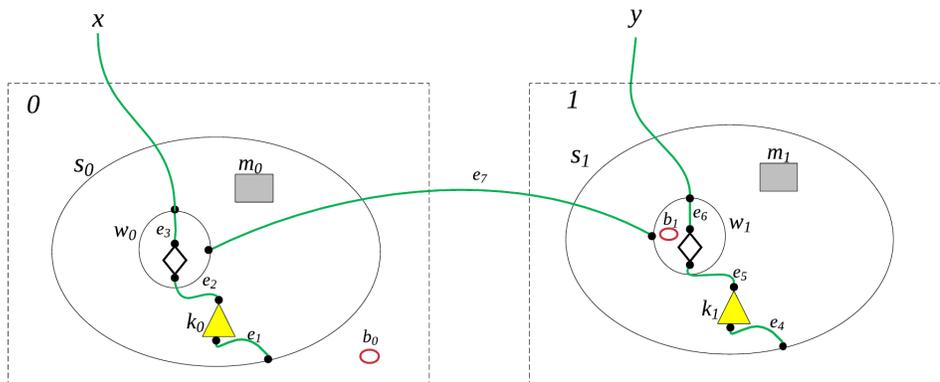
En el tiempo  $t_6$  (Figura 4-6), la interfaz inalámbrica de  $w_1$  se activa con ayuda de parámetros obtenidos a través de  $b_0$ . En este momento, los nodos están a punto de interconectarse. El enlace  $e_6$  simboliza que la interfaz inalámbrica ya está lista para efectuar comunicación en modo ad hoc en el territorio 1 hacia los nodos cercanos. Por lo cual, este es el estado inmediatamente anterior a la conformación un clúster.



**Figura 4-6.:** Fase 2:  $t_6$  activación de la interfaz inalámbrica de  $w_1$

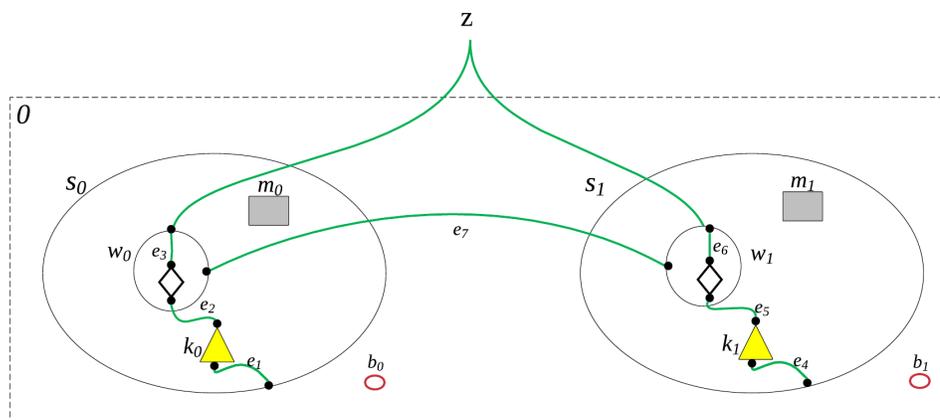
### Fase 3

La Figura 4-7 representa el momento  $t_7$  en el que los nodos  $S_0$  y  $S_1$  comienzan el proceso de intercambio de mensajes para enlazarse mediante sus interfaces lógicas. En  $w_1$  empieza el proceso de generación de mensajes broadcast  $b_1$ . En este momento, los nodos se ven entre sí; y por lo tanto, son nodos vecinos. El establecimiento de la conexión entre  $S_0$  y  $S_1$  se representa con  $e_7$ .



**Figura 4-7.:** Fase 3:  $t_7$  vinculación de dos nodos cercanos

La Figura 4-8 muestra la unión de los nombres  $x$  y  $y$  en un nombre  $z$  para indicar que ahora los nodos están unidos a una red ad hoc con el mismo *ESSID*. Esta figura es la representación formal de un clúster identificado con el nombre  $z$  y conformado por los nodos  $S_0$  y  $S_1$  que ahora hacen parte de la región 0 ampliada, es decir, la suma de las áreas de cobertura de los dos nodos. El nodo  $S_1$  se convierte en otro nodo originador y propaga mensajes broadcast  $b_1$  dentro de su área de cobertura para anunciar la existencia del clúster  $z$  e invitar a otros nodos cercanos a hacer parte del mismo. Se concluye que en este modelo un clúster es el resultado de la operación de unión de bigrafos =  $S_{0xz} \mid S_{1zy}$



**Figura 4-8.:** Fase 3:  $t_8$  representación formal de un clúster  $z$  conformado por dos nodos vecinos  $S_0$  y  $S_1$ .

#### Fase 4

Esta fase se representa en la Figura 4-9. En esta, se repiten las fases 2 y 3 cada vez que un nodo  $s_i$  se acerca a la región 0 y acepta hacer parte del clúster  $z$ . Con la vinculación de  $n$  nodos se irá conformando una estructura de malla que conecta a todos los nodos y se identifica con el nombre del clúster. Cada

nuevo nodo del clúster efectuará el procedimiento hasta establecer un enlace como  $e_{11}$  con su vecino mas cercano. El resultado de la unión de  $n$  bigrafos en la misma región  $0$  es el clúster  $z$ .

$$\text{Clúster } z = S_{0xz} | S_{1yz} | \dots | S_{ijz}$$

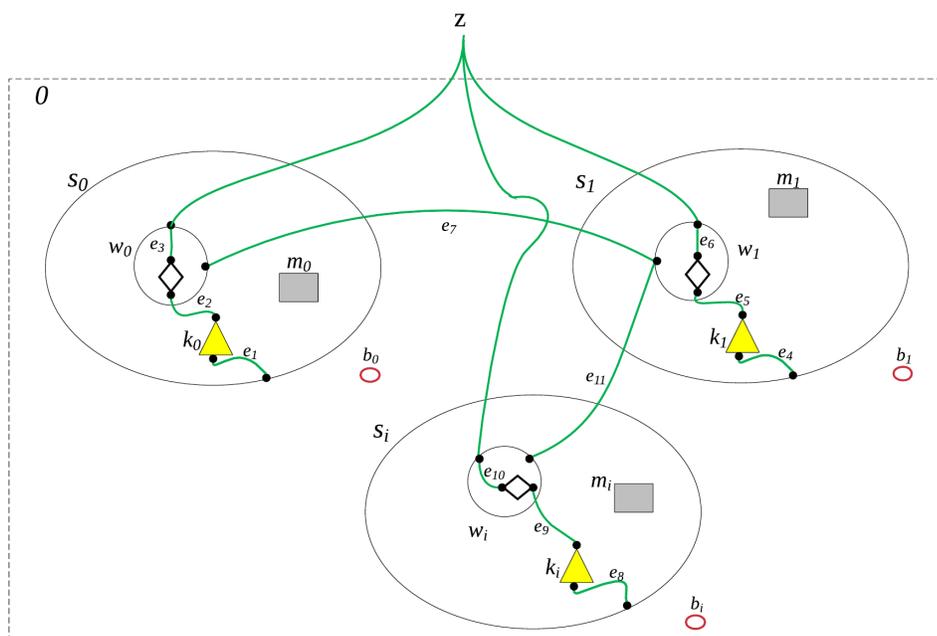


Figura 4-9.: Representación formal de un clúster de nombre  $z$  con  $n$  nodos vecinos.

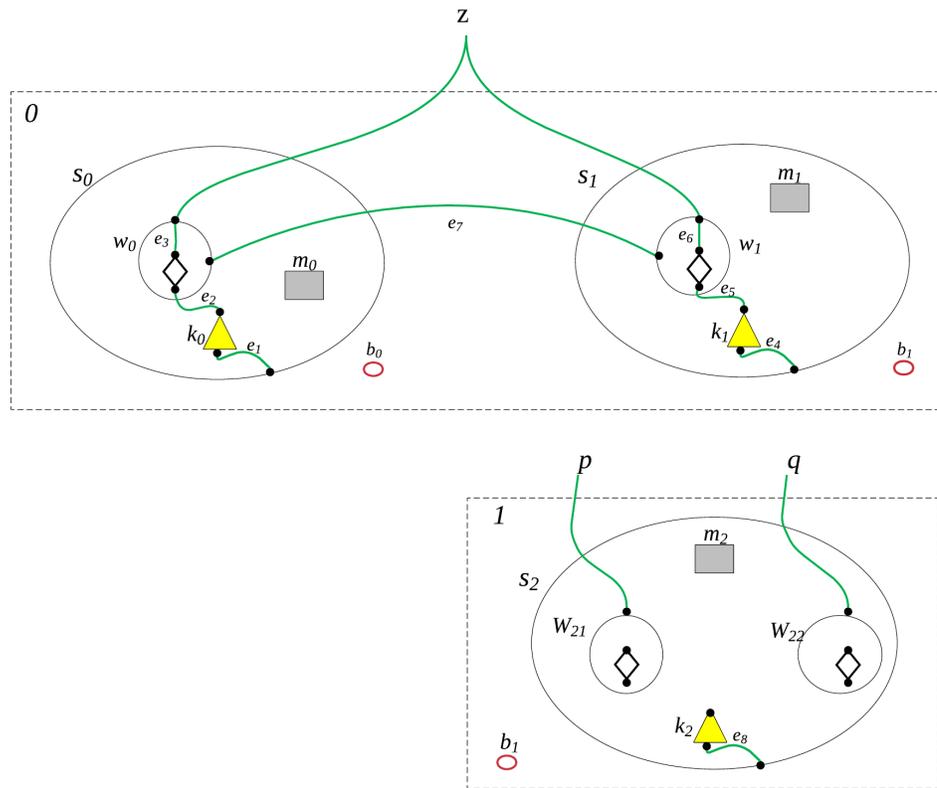
## Fase 5

En las fases anteriores se describió el proceso de conformación de un solo clúster. De manera semejante, en esta fase se extiende el proceso, esta vez para generalizar la formación de cierto número de clústeres ( $m$ ), cada uno con  $n$  nodos, en un territorio ad hoc. Para conformar más de un clúster, se requiere que nodos *gateway* lleguen al territorio ad hoc. Por lo tanto, en este modelo establece que el nivel de jerarquía en la red está dado por el número de interfaces inalámbricas que tenga un nodo. De esta manera, son los nodos *gateway* los que habilitan la división de la red en clústeres diferentes.

Un nodo con 2 interfaces inalámbricas podrá conformar dos clústeres diferentes. Teóricamente un nodo con  $n$  interfaces, podrá generar  $n$  clústeres y hacer parte de ellos. Entonces, en este escenario sucede que un mismo nodo puede pertenecer a  $n$  clústeres, sin embargo, esto no es así con sus interfaces. Cada interfaz de un nodo *gateway* pertenece un clúster diferente. De esta forma, se presenta la operación de *sharing* en los bigrafos, porque exceptuando sus interfaces inalámbricas, los demás recursos del nodo son el elemento común a todos los clústeres que conforman.

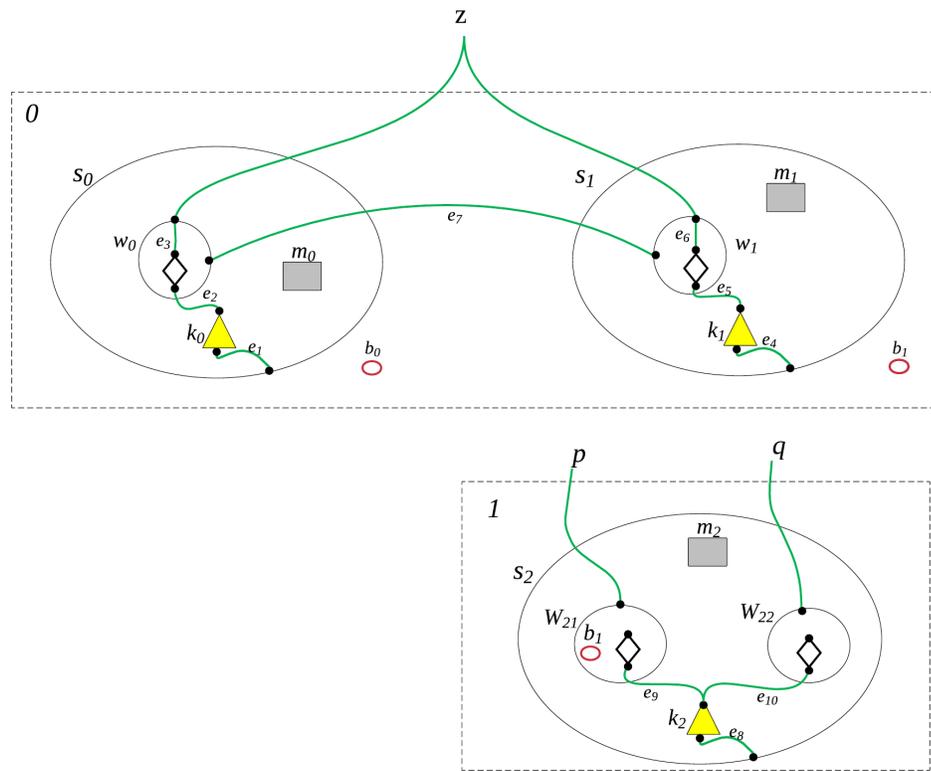
Para explicar el proceso, se asumirá la creación de dos clústeres ( $m = 2$ ). En el tiempo  $t_1$  de la fase 5, al territorio ad hoc de un clúster  $z$  ya conformado por dos nodos se acerca un nodo  $S_2$  con dos interfaces inalámbricas: una en  $W_{21}$  y otra en  $W_{22}$ . El nodo  $S_2$  se encuentra en la vecindad del nodo

$S_1$  y por lo tanto recibe los mensajes broadcast  $b_1$  del nodo  $S_1$ . Además, cuenta con un controlador  $k_2$  que está en espera de órdenes para realizar acciones sobre las interfaces inalámbricas (Figura 4-10).

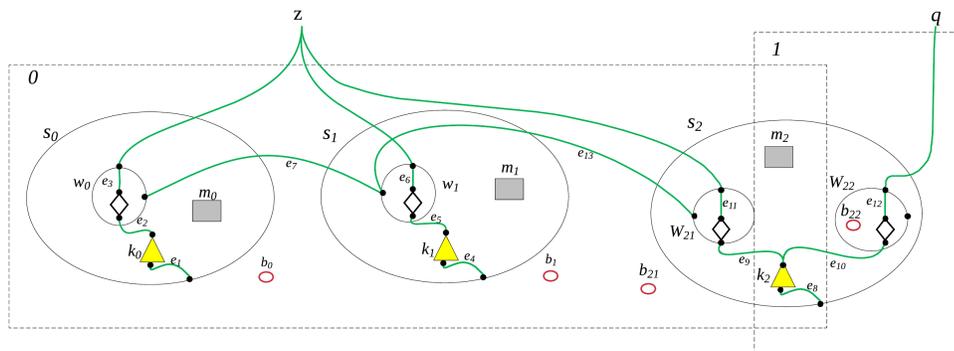


**Figura 4-10.:** Fase 5:  $t_1$  un nodo gateway con dos tarjetas de red se aproxima al territorio del clúster  $z$ .

En el tiempo  $t_2$  de la fase 5, el controlador  $k_2$  activa el modo ad hoc en las dos tarjetas inalámbricas de  $S_2$ . Esto se representa mediante  $e_9$  y  $e_{10}$ . La interfaz inalámbrica  $w_{21}$  recibe el mensaje  $b_1$  y extrae de este la información necesaria para establecer sus propios parámetros de configuración (Figura 4-11). En el tiempo  $t_3$  (Figura 4-12), el nodo  $S_2$  configura su interfaz  $w_{21}$  con los parámetros requeridos para hacer parte del clúster  $z$ . Una vez se activa la interfaz inalámbrica ( $e_{11}$ ), se establece el enlace  $e_{13}$  para iniciar la comunicación con su nodo vecino  $S_1$  y empezar a ser miembro del clúster  $z$  a través de la interfaz  $w_{21}$ . En consecuencia,  $S_2$  empieza a hacer parte de la misma región ad hoc 0 en la que se puede comunicar con los nodos miembros del clúster  $z$ . Sin embargo, una parte del nodo que corresponde a la interfaz  $w_{22}$  pertenece a la región ad hoc 1. En este paso, sucedió de forma concurrente que la interfaz  $w_{22}$  se configuró con parámetros determinados para crear un nuevo clúster. Inmediatamente después de la activación de la misma ( $e_{12}$ ), este nodo se convierte en nodo originador de un nuevo clúster identificado con  $q$ . Entonces genera mensajes broadcast  $b_{22}$  a través de  $w_{22}$  para anunciar la existencia del mismo. En este momento,  $S_2$  tiene la capacidad de invitar a a ser parte del clúster recién conformado a los nodos cercanos que reciben mensajes  $b_{22}$ .

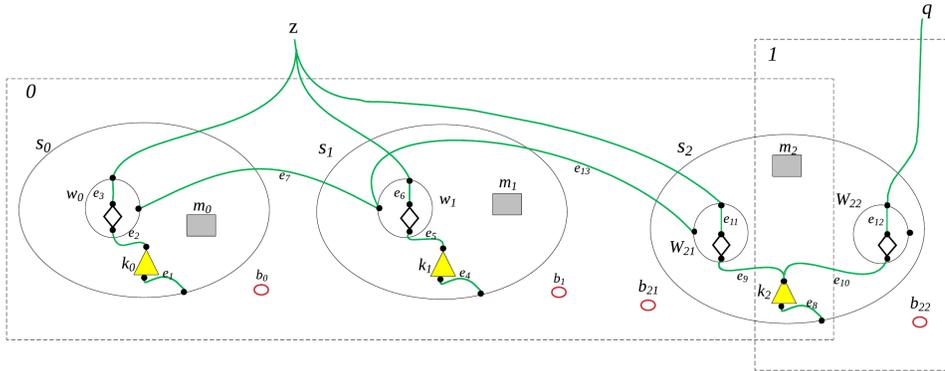


**Figura 4-11.:** Fase 5:  $t_2$  activación del modo ad hoc de  $W_{21}$  y  $W_{22}$ .



**Figura 4-12.:** Fase 5:  $t_3$  un nodo gateway se une al clúster existente  $z$  y publica la existencia de una nueva red ad hoc con identificador  $q$

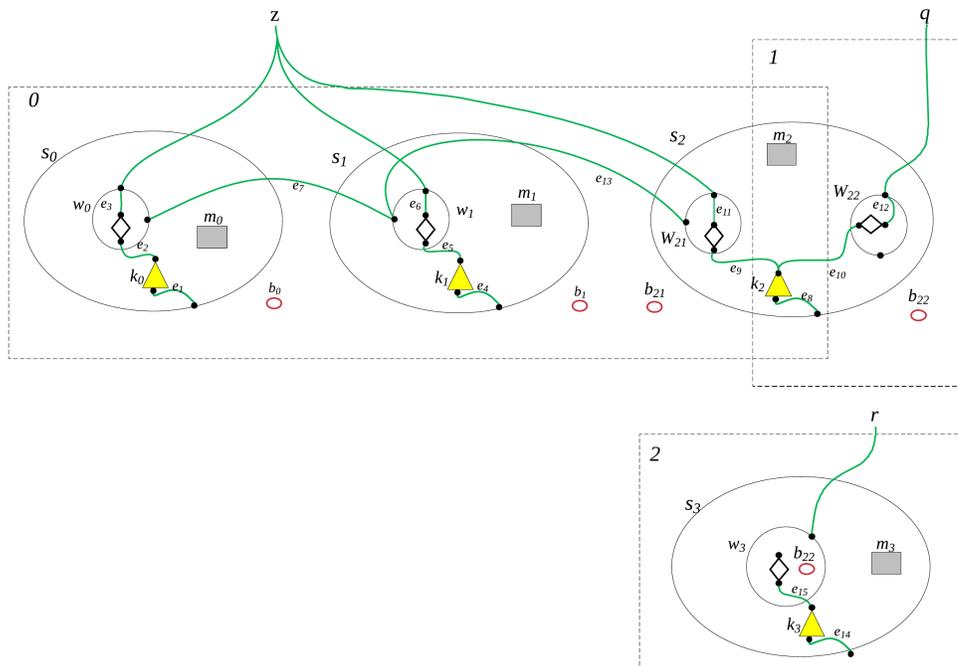
En la Figura 4-13, se ilustra la fase en el tiempo  $t_4$  cuando el nodo gateway  $S_2$  pertenece al clúster  $z$  a través de  $W_{21}$  en la región 0 y ha publicado la creación de otra red ad hoc en la región 1 a través de  $W_{22}$ . Este nodo propaga mensajes originadores  $b_{22}$  en espera de que otros nodos cercanos se unan a la red  $q$  para conformar un nuevo clúster.



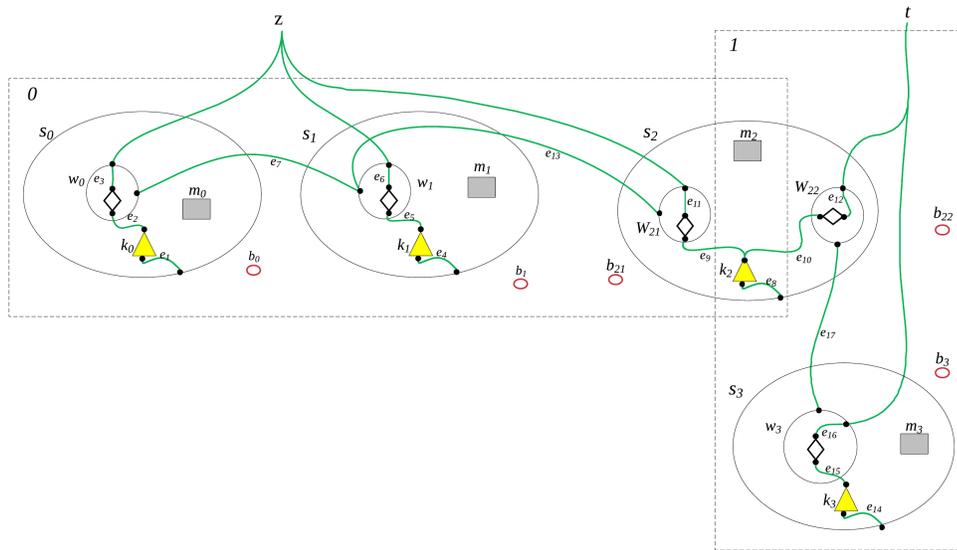
**Figura 4-13.:** Fase 5:  $t_4$  el nodo  $S_2$  invita a los nodos cercanos a vincularse a la red  $q$ .

En el tiempo  $t_5$  ilustrado en la Figura 4-14, un nodo cercano  $S_3$  en modo ad hoc escucha los mensajes  $b_{22}$ , los procesa y establece los parámetros para configurar  $w_3$ .

La Figura 4-15 representa conformación del clúster  $t$ , que se denota con otro nombre diferente a  $q$  para simbolizar el establecimiento del vínculo entre dos nodos asociados  $S_2$  y  $S_3$ . Solo en el momento en que se activa la interfaz  $w_3$  ( $e_{16}$ ) y se interconectan los nodos mediante  $e_{17}$ , se puede decir que existe un grupo de nodos asociados en una red diferente. Ahora  $S_3$  se convierte en nodo originador y propaga la existencia del clúster a través de  $b_3$ .



**Figura 4-14.:** Fase 5:  $t_5$   $S_2$  invita a  $S_3$  a vincularse al clúster  $q$



**Figura 4-15.:** Establecimiento de los clústeres  $z$  y  $t$  mediante las interfaces inalámbricas del nodo gateway  $S_2$ .

Las reglas de reacción que se generan mediante las fases de conformación de clústeres físicos se sintetizan en la Tabla 4-2.

| Regla de Reacción | Descripción  | Figura   |      |
|-------------------|--------------|--|------|
| 1                 | $R_{ahon}$   | Los nodos cercanos activan sus interfaces de red en modo ad hoc                                      | 4-4  |
| 2                 | $R_{origin}$ | Un nodo crea un clúster y publica su existencia  | 4-3  |
| 3                 | $R_{cast}$   | Un nodo que originó un clúster envía mensajes broadcast en la región ad hoc                          | 4-4  |
| 4                 | $R_{conf}$   | Un nodo configura los parámetros necesarios para unirse al clúster de un nodo vecino                 | 4-5  |
| 5                 | $R_{join}$   | Los nodos se enlazan y conforman el clúster $z$  | 4-8  |
| 6                 | $R_{gway}$   | Un nodo gateway llega al territorio ad hoc, activa sus interfaces en modo ad hoc                     | 4-11 |
| 7                 | $R_{sel}$    | Un nodo gateway selecciona las acciones a realizar, unirse a un clúster $z$ y crear un clúster nuevo | 4-12 |
| 8                 | $R_{gwcast}$ | Un nodo gateway participa en el clúster $z$ y envía mensajes broadcast en el clúster $q$             | 4-14 |

**Tabla 4-2.:** Reglas de reacción

La definición formal del bigrafo finalmente construido de la Figura 4-15 se encuentra en las Ecuaciones 4-1, 4-2, 4-3, 4-4, 4-5, 4-6 y 4-7.

De acuerdo con teoría de bigrafos, las interacciones en el proceso de conformación de  $m$  clústeres se pueden modelar por un hipergrafo (grafo de enlace) y grafo de lugar. La Figura 4-16 (a) muestra el ejemplo de un grafo de lugar de los dos clústeres ( $m = 2$ ) conformados en el territorio ad hoc. La Figura 4-16 (b) corresponde a su hipergrafo y (c) al grafo de lugar de los mensajes broadcast que se difunden en las regiones 0 y 1.

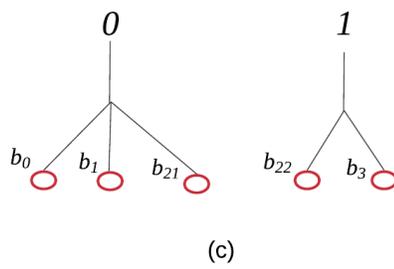
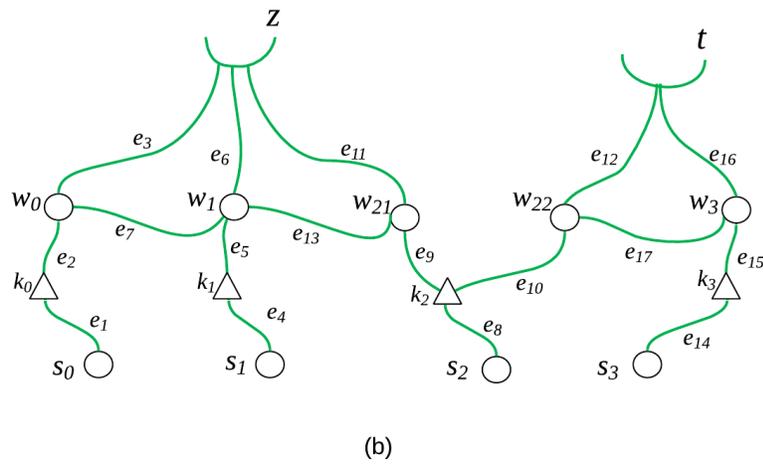
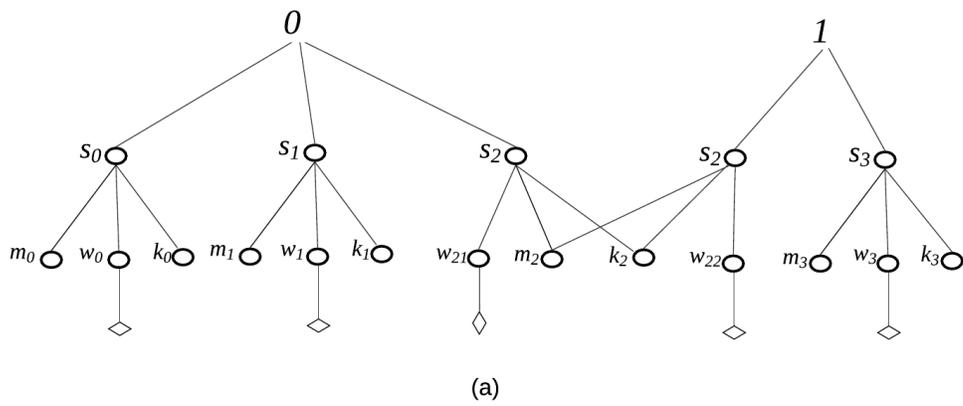


Figura 4-16.: (a) Grafo de lugar. (b) grafo de enlace. (c) grafo de lugar de los mensajes broadcast.

## Definición del bigrafo de clúster físico

$$\mathcal{K} = \{\mathcal{N} : \{1,2\}, \mathcal{W} : 3, \mathcal{I} : 2\} \quad (4-1)$$

$$V = \{s_0, s_1, s_2, s_3, w_0, w_1, w_{21}, w_{22}, w_3, k_0, k_1, k_2, k_3\} \quad (4-2)$$

$$E = \{e_0, e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}, e_{14}, e_{15}\} \quad (4-3)$$

$$crtl(v) = \begin{cases} N : 1, & \text{if } v \in \{s_0, s_1, s_3\} \\ N : 2, & \text{if } v \in \{s_2\} \\ W : 3, & \text{if } v \in \{w_0, w_1, w_{21}, w_{22}, w_3\} \\ I : 2, & \text{if } v \in \{k_0, k_1, k_2, k_3\} \end{cases} \quad (4-4)$$

$$prnt(v) = \begin{cases} S_0, & \text{if } v \in \{w_0, k_0, m_0\} \\ S_1, & \text{if } v \in \{w_1, k_1, m_1\} \\ S_2, & \text{if } v \in \{w_{21}, w_{22}, k_2, m_2\} \\ S_3, & \text{if } v \in \{w_3, m_3\} \end{cases} \quad (4-5)$$

$$link(l) = \begin{cases} e_7, & \text{if } l \in \{w_0, w_1\} \\ e_{13}, & \text{if } l \in \{w_1, w_{21}\} \\ e_{17}, & \text{if } l \in \{w_{22}, w_3\} \\ e_1, & \text{if } l \in \{s_0, k_0\} \\ e_4, & \text{if } l \in \{s_1, k_1\} \\ e_8, & \text{if } l \in \{s_2, k_2\} \\ e_{14}, & \text{if } l \in \{s_3, k_3\} \\ e_9, & \text{if } l \in \{k_2, w_{21}\} \\ e_{10}, & \text{if } l \in \{k_2, w_{22}\} \\ z, & \text{if } l \in \{w_0, w_1, w_{21}\} \\ t, & \text{if } l \in \{w_{22}, w_3\} \end{cases} \quad (4-6)$$

$$P = \{(s_0, 0), (s_1, 0), (s_2, 0), (s_2, 1), (s_3, 1), (w_0, 0), (w_1, 0), (w_{21}, 0), (w_{22}, 1), (w_3, 1), \\ (m_0, 0), (m_1, 0), (m_2, 0), (m_2, 1), (m_3, 1), (k_0, 0), (k_1, 0), (k_2, 0), (k_2, 1), (k_3, 1)\} \quad (4-7)$$

## 4.2. Definición de clúster lógico en el sistema TLÖN

El sistema TLÖN se considera una nube móvil conformada por estaciones inalámbricas capaces de comunicarse en modo ad hoc. Conforme a esto, se articula el concepto de nube móvil de Fitzek(2014) para definir un territorio lógico o clúster lógico en TLÖN.

Un clúster lógico es un conjunto de recursos distribuidos conectados de forma inalámbrica que pueden ser intercambiados, desplazados, incrementados y combinados en diferentes formas. En términos del modelo social inspirado es una entidad territorial abstracta con la capacidad de establecer distintas asociaciones territoriales realizando un mapeo dinámico de los recursos lógicos de la red. Estos recursos pueden ser agentes, servicios, software, o virtualización de recursos físicos disponibles en cada nodo como procesador, memoria RAM, almacenamiento, dispositivos de entrada/salida, sensores, actuadores e interfaces de red (Fitzek, 2014).

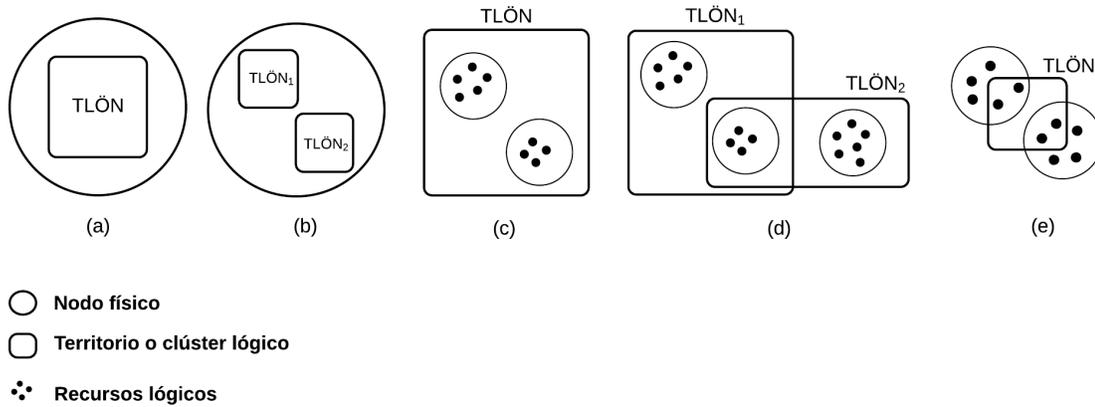
En la sección anterior el clúster físico se limitaba a la agrupación de estaciones inalámbricas, es decir, el establecimiento de los límites de un territorio físico en la capa red ad hoc, en esa dimensión la forma en que se intercomunican clústeres físicos es mediante nodos *gateway*. En esta sección se aborda el clúster lógico, considerado como un país abstracto TLÖN, que en otras palabras es un espacio donde el sistema TLÖN puede ser desplegado y ejercer algún tipo de control. Como se afirmó anteriormente en la asociación de componentes lógicos, la acción del clúster lógico no esta limitada por los clústeres físicos o al espacio físico de un solo nodo, sino que puede establecerse entre los recursos de diferentes nodos o de diferentes clústeres físicos. De acuerdo con lo anterior, se plantea diferentes formas de asociaciones territoriales lógicas:

**Países de TLÖN:** agrupaciones de agentes que se encuentran o habitan en diferentes nodos físicos o en diferentes clústeres físicos. También se denominará clúster físico distribuido.

**Departamentos de TLÖN:** agrupaciones de agentes que se encuentran en un solo nodo físico o agrupaciones de recursos lógicos como procesamiento dentro de un mismo nodo.

**Provincias administrativas de TLÖN:** clústeres de aplicaciones que se despliegan de forma distribuida entre diferentes nodos y tienen la función de coordinar tareas o prestar servicios específicos.

Al contrario que en la geografía real del planeta tierra donde los territorios son contiguos y delimitados, los territorios lógicos para el sistema TLÖN pueden superponerse. En este sentido, el clúster físico se asemejaría más la realidad territorial pero el clúster lógico es una forma abstracta de agrupación que no esta limitada a los espacios físicos. En consecuencia, el sistema TLÖN puede desplegarse en un mismo nodo una o n veces, en uno o más nodos, en uno o más clústeres físicos y en uno o más clústeres lógicos y de diferentes formas, permitiendo superposiciones territoriales. La Figura 4-17 ilustra una idea de las diferentes formas de superposiciones territoriales del sistema TLÖN.



**Figura 4-17.:** Diferentes formas de superposiciones territoriales del sistema TLÖN

De forma que un sistema TLÖN puede existir en un solo nodo. Figura 4-17 (a).

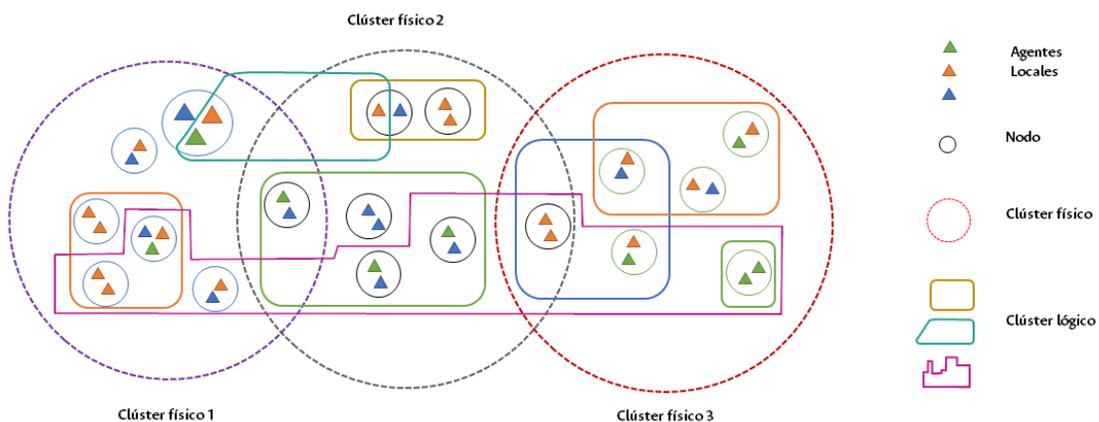
En un solo nodo pueden existir 2 o más clústeres lógicos cada uno con un sistema TLÖN diferente. Figura 4-17 (b).

Un clúster lógico puede estar conformado por dos o mas nodos y un sistema TLÖN puede desplegarse sobre éstos. Figura 4-17 (c).

Pueden existir dos clústeres diferentes donde se despliegan TLÖNes diferentes, que comparten los recursos de un nodo común. Figura 4-17 (d).

Y, un clúster lógico puede conformarse con parte de los recursos de uno o más nodos. Figura 4-17 (e).

En este trabajo se define el clúster lógico como una agrupación de agentes locales con el objetivo de cumplir tareas sobre un grupo de recursos específico. El clúster lógico puede conformarse con agentes locales que se encuentran en nodos diferentes, de manera que pueden superponerse. Es decir, un agente local puede pertenecer a uno o más clústeres lógicos. La Figura 4-18 presenta la idea anterior en un escenario más amplio, con agentes participando en diferentes clústeres lógicos sobre tres clústeres físicos.



**Figura 4-18.:** Clústeres lógicos conformados por agrupación de agentes locales.

La agrupación de los agentes es posible a través de un mecanismo de comunicación entre los mismos, de modo que las operaciones que se analizan para la conformación del clúster lógico se refieren a la comunicación entre los procesos que ejecutan los agentes.

### 4.2.1. Proceso de conformación del clúster lógico

Para describir el proceso de conformación de un clúster lógico se asume que los nodos involucrados se encuentran interconectados por sus interfaces inalámbricas. Por lo tanto, hacen parte de un mismo clúster o de diferentes clústeres físicos y a través de un mecanismo de comunicación inter-clúster es posible crear asociaciones lógicas.

Se presentarán dos casos de conformación de clústeres lógicos. En el primero se genera un clúster lógico dentro de un mismo nodo (departamento de TLÖN). Y en el segundo se genera un clúster lógico distribuido. En el cual se agrupan agentes distribuidos en diferentes nodos para conformar dos clústeres lógicos distintos (países de TLÖN).

#### *Clúster lógico en un nodo*

Estado 0: En el nodo se activa una entidad de registro de agentes, que actúa como controlador de los mismos. Una de las funciones del registro es escanear las características del estado del nodo e identificar el número de agentes que puede desplegar. Es importante recordar que, un agente es una pieza de código contenida dentro un dispositivo. Teóricamente el número máximo de agentes locales desplegados en un nodo es  $n$ . Sin embargo, este valor está limitado por las capacidades de procesamiento del sistema. Para el ejemplo se asumirá el valor de  $n=3$ .

Fase 1: Se inicia cuando el registro genera instancias de agentes locales independientes, según las necesidades de la aplicación.

Fase 2: El registro envía información de configuración del clúster a dos agentes y éstos habilitan los puertos necesarios para establecer la comunicación.

Fase 3: Se establece comunicación bidireccional entre los agentes locales. Se conforma un clúster interno con dos agentes locales.

Fase 4: El registro identifica que es necesario desplegar otro agente y que tiene la capacidad de generarlo. Por lo tanto, activa la instancia de un tercer agente local y le envía los parámetros del clúster lógico para que se pueda vincular.

Fase 5: El tercer agente local se vincula al clúster lógico estableciendo comunicación con uno de los agentes locales miembro. El clúster lógico aumenta de tamaño y los agentes miembros están preparados para que se les asigne una tarea.

Fase 6: El registro asigna una tarea de manipulación de recursos y en consecuencia cada agente efectúa acciones sobre una porción de un recurso. Ahora virtualmente particionado. El proceso puede repetirse  $n$  veces dependiendo de las necesidades del sistema y de los parámetros que determinan el número máximo de agentes locales.

### Clúster lógico distribuido

Fase 0: se inicia con  $n$  nodos vecinos que cuentan con agentes locales, registrados y controlados por la unidad de registro respectiva. Se asume que están interconectados en un territorio ad hoc. El clustering físico es transparente para los agentes, ya que estos pueden pertenecer al mismo clúster o a diferentes clústeres, lo importante es que los nodos que los contienen puedan alcanzar otros nodos y conocer la existencia de otros agentes. Es en este punto donde el clúster físico juega un papel muy importante, pues este define quienes podrán o no participar del clúster lógico y habilitará a nivel de red las rutas para enlazar los nodos participantes. Si las rutas inter-clúster no existen entonces no será posible la comunicación entre nodos de diferentes clústeres físicos y los agentes no podrán interactuar.

Fase 1: el mecanismo de clustering lógico de los nodos busca entre sus vecinos aquellos que tienen agentes con los que puede agruparse. Uno de los agentes locales inicia la petición de enlace hacia el agente de otro nodo. Cuando el nodo vecino acepta la petición se abren puertos de comunicación en los nodos de ambos agentes y a través de estos se establecen los enlaces de transmisión y recepción de mensajes entre agentes. Se conforma un clúster cuyo nombre es solo conocido internamente por los agentes que están enlazados.

Fase 2: cuando otros nodos vecinos aceptan unirse al clúster lógico, estos abren puertos adicionales para enviar peticiones a otros agentes y establecer la comunicación con otros agentes en el territorio ad hoc. Por ejemplo, es posible que un nodo con dos o más agentes, divida parte de sus agentes para que unos pertenezcan a determinado clúster lógico y que otros hagan parte de un clúster completamente diferente. El proceso se repite con cada nodo vecino que desea participar.

#### 4.2.2. Modelo de clúster lógico con bigrafos

Las interacciones y reglas para la conformación de clústeres lógicos se describen mediante un modelo de bigrafos.

La Tabla 4-3 se listan las convenciones para la notación de los elementos de los bigrafos para el modelo de clúster lógico.

**Tabla 4-3.:** Elementos del bigrafo

| Elemento | Descripción                            |
|----------|--|
| $S_i$    | Nodo físico $i$                        |
| $v_i$    | Agente local $i$                       |
| $k_i$    | Entidad de registro $i$                |
| $m_i$    | Recursos del nodo $i$                  |
| $e_i$    | enlace $i$ de comunicación de procesos |

### Clúster lógico en un nodo

La Figura 4-19 muestra la transición de la fase 0 a la fase 1 en la conformación de un clúster lógico dentro de un nodo. Los agentes  $v_0$  y  $v_1$  disponen de puertos definidos para establecer una posible comunicación. El registro  $k_0$  decide que en esta ocasión estos agentes solo establecerán comunicación con agentes del mismo nodo.

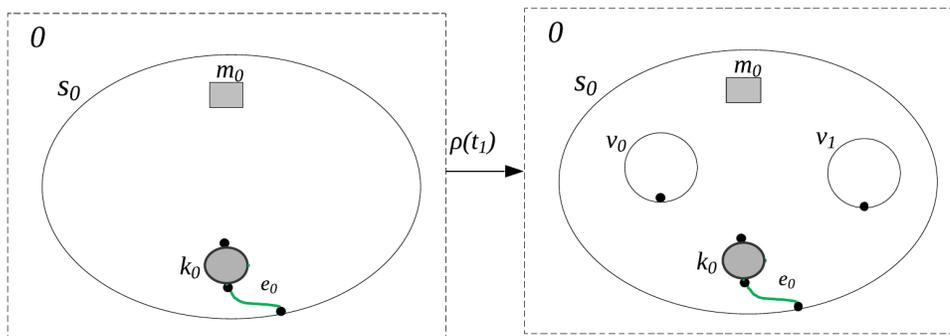


Figura 4-19.: Clúster lógico dentro de un nodo: fase 0 y fase 1

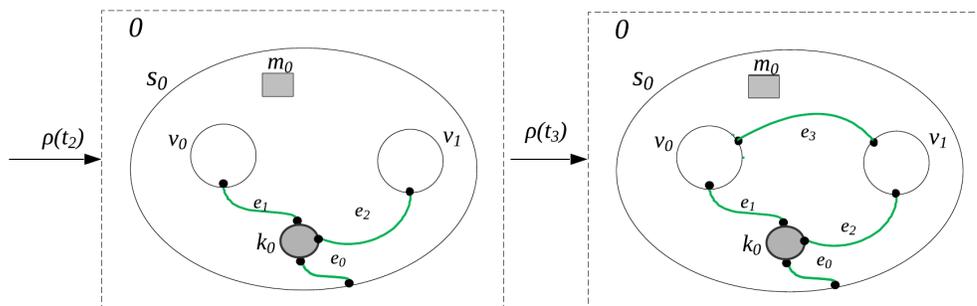


Figura 4-20.: Clúster lógico dentro de un nodo: transición de fase 2 a fase 3.

En la fase 2, la entidad de registro configura los parámetros de funcionamiento del clúster. Los agentes activan sus puertos de comunicación. En la fase 3, los agentes  $v_0$  y  $v_1$  realizan un intercambio de mensajes para comprobar la conexión y se establece el clúster. (Figura 4-20).

En la Figura 4-21 se representa el momento de la fase 4 cuando el registro despliega un tercer agente  $v_2$  y este recibe los parámetros de configuración que le permiten unirse al clúster existente. En este punto el registro podría decidir otra acción diferente para este nuevo agente  $v_2$ , y configurarlo para hacer parte de un clúster distribuido. En la fase 5, el agente  $v_2$  se une al clúster local, ahora identificado con el nombre  $x$ . Por último, en la fase 6, la entidad de registro u otra entidad del sistema puede acceder al clúster lógico y asignar tareas sobre un recurso. En consecuencia, cada agente puede acceder a una porción de un recurso, de esa forma el sistema podrá ejecutar tareas de forma concurrente sobre el mismo.

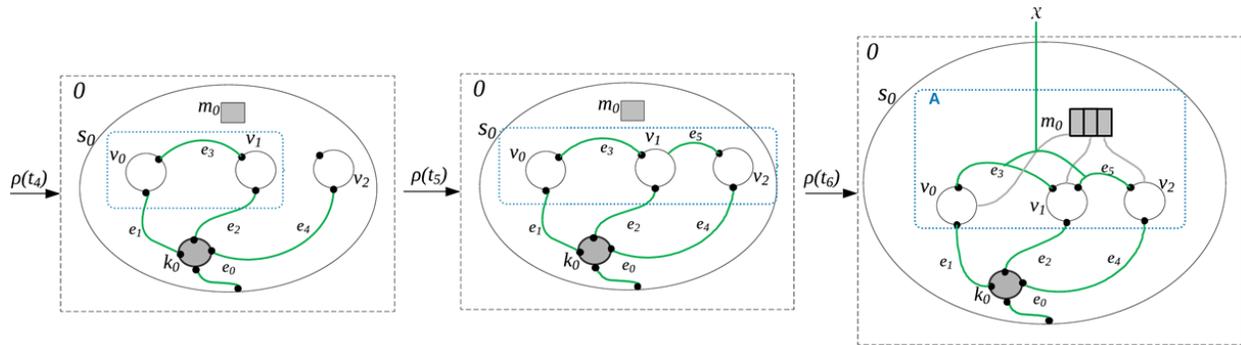


Figura 4-21.: Clúster lógico dentro de un nodo: transición a fase 4, fase 5 y fase 6

### Clúster lógico distribuido

La Figura 4-22, muestra la fase 0 en la conformación de un clúster lógico distribuido. Se asume que los nodos están interconectados por un enlace  $e_4$  y es posible que los agentes conozcan la dirección del nodo con el que se van a vincular. La unidad de registro busca en la red los nodos con los que puede asociarse y solicita información sobre los agentes remotos disponibles.

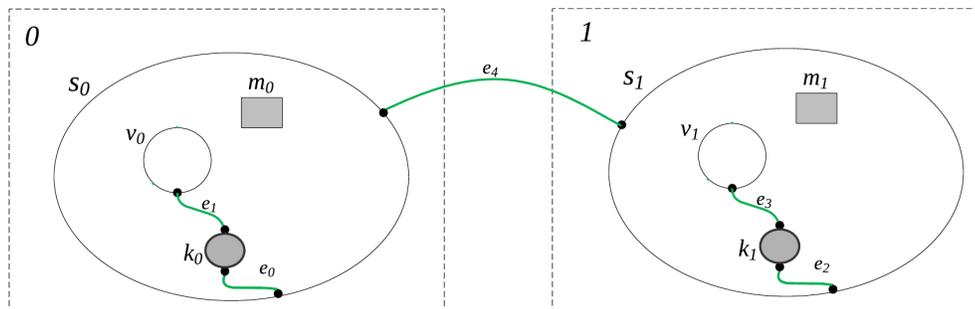
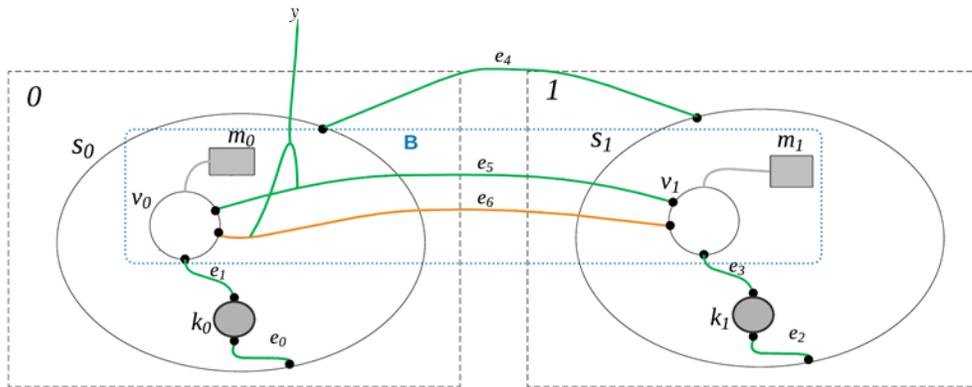


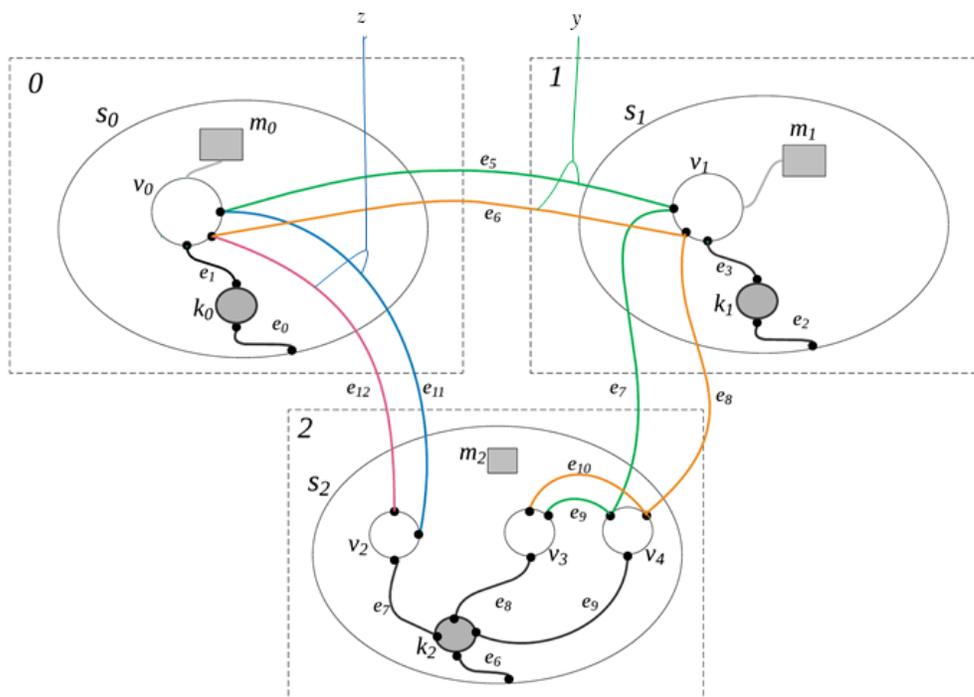
Figura 4-22.: Clúster lógico distribuido: fase 0.

En la fase 1 (Figura 4-23), los agentes distribuidos en nodos diferentes establecen comunicación mediante dos puertos diferentes, uno sirve para recepción y otro para transmisión. Se conforma el clúster de nombre  $y$ . Los recursos  $m_0$  y  $m_1$  se unen de forma virtual para formar un banco de recursos que pueden ser manipulados por agentes de otros nodos. De esa manera, el sistema tendrá más posibilidades de asignación de recursos para los agentes participantes.

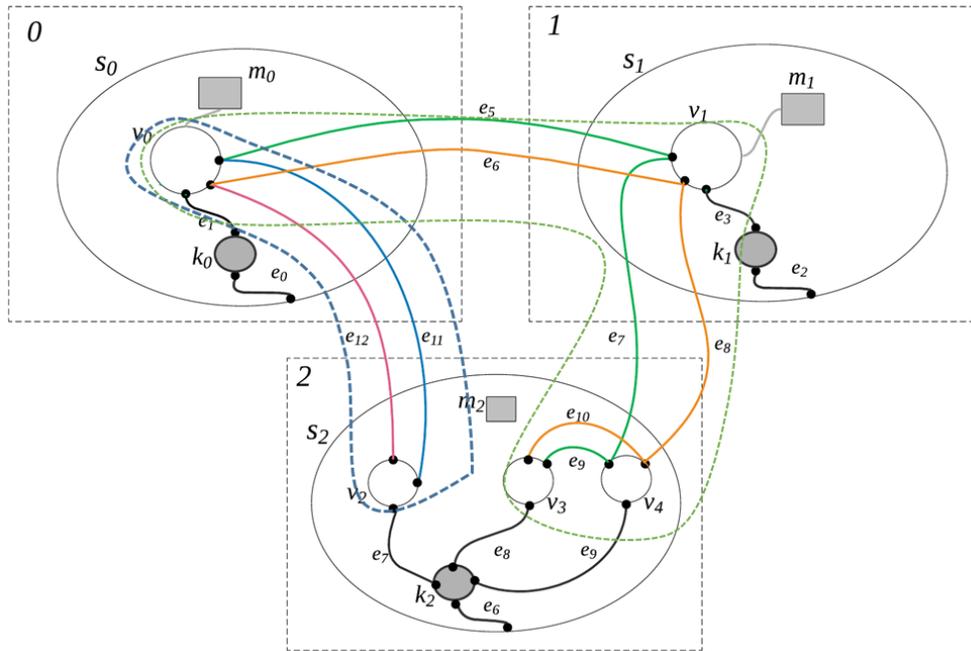


**Figura 4-23.:** Clúster lógico distribuido: fase 1. Intercambio de mensajes entre los agentes  $v_0$  y  $v_1$  de nodos diferentes.

La Figura 4-24 muestra un posible caso de la fase 3. El clúster se amplía con la llegada de un tercer nodo que distribuye dos de sus agentes ( $v_3$  y  $v_4$ ) para unirse al clúster  $y$  y un tercer agente  $v_2$  para armar un nuevo clúster  $z$  con el agente  $v_0$ . En la Figura 4-25 se representa la superposición de dos territorios lógicos  $y$  y  $z$ , donde el agente  $v_0$  se considera un recurso lógico compartido. Ahora los recursos  $m_0$ ,  $m_1$  y  $m_2$  pueden ser distribuidos en los clústeres conformados. En consecuencia, se descubre una nueva necesidad para el sistema TLÖN, relacionada con el desarrollo de una política de asignación de recursos entre clústeres.



**Figura 4-24.:** Clúster lógico distribuido: fase 2, establecimiento de dos clústeres lógicos distribuidos.



**Figura 4-25.:** Clúster lógico distribuido: fase 2, representación de los lugares lógicos que comparten recursos y superposición de los clústeres lógicos  $y$  y  $z$  en el agente  $v_0$ .

## 5. Mecanismo de clustering para redes ad hoc basadas en el sistema TLÖN

En este capítulo se describirá el experimento base y los procedimientos que se llevaron a cabo en la implementación del mecanismo de clustering. El experimento diseñado se basa en la conformación de tres clústeres físicos sobre los cuales se conformaron dos clústeres lógicos. Este fue el insumo que sirvió para las pruebas realizadas, además es importante resaltar que fue resultado de numerosas experimentaciones previas y deducciones obtenidas mediante ensayo y error. Del experimento base se extrajeron elementos para plantear el modelo descrito en el capítulo 4 y las características del mecanismo que se describirán a continuación.

El mecanismo de clustering para el sistema TLÖN consta de dos partes: la primera es el establecimiento de clústeres físicos y la segunda es la conformación de clústeres lógicos.

En la implementación se utilizó el estándar 802.11 para la comunicación inalámbrica. Se escogió la banda de frecuencias 2.4 Mhz y en esta banda se usaron los canales 1 (2412 MHz), 4 (2427 MHz) y 7 (2442 MHz) para desplegar tres clústeres. Se escogieron estos canales porque están separados y presentan menos interferencia co-canal. Se uso el protocolo de redes ad hoc BATMAN, ya que facilita la comunicación punto a punto entre estaciones inalámbricas y permite una manipulación de la red gracias a su servicio de mensajería Alfred, además de las múltiples opciones del comando batctl, que actúa como interfaz de comunicación con el módulo del kernel batman-adv.

Se denotará el clúster físico con la letra  $F$  y las variables necesarias para armarlo son: identificador del clúster  $essid$ , tamaño máximo de la trama MTU  $m$ , canal de transmisión  $ch$ , conjunto de interfaces inalámbricas  $W$ , y conjunto  $B$  de interfaces virtuales BATMAN que participan en el clúster para la capa 2. De esta forma, un clúster físico  $F$  en el territorio ad hoc, se define mediante una función que procesa esas variables.  $F = f(essid, m, ch, W, B)$ .

### Procedimientos ejecutados en un nodo

El **nodo originador** o primer nodo que inicia la red ad hoc debe ejecutar los siguientes pasos:

1. Detección del número de interfaces inalámbricas disponibles
2. Activación del modo ad hoc en sus interfaces
3. Establecimiento de las variables MTU, ESSID y canal para cada clúster que desea establecer.

4. Activación y configuración de interfaces inalámbricas
5. Creación de interfaces de capa 2 por cada una de las interfaces inalámbricas.
6. Activación del protocolo de enrutamiento para redes ad hoc y publicación de las redes creadas mediante mensajes broadcast.

En los **nodos cercanos**, se debe realizar un proceso similar, pero se agregan pasos adicionales porque los nodos que llegan a la zona de una red existente primero deben procesar la información de clústeres existentes. En este caso, los pasos son:

1. Detección del número de interfaces inalámbricas disponibles
2. Activación del modo ad hoc en sus interfaces
3. Recepción de notificaciones de clústeres disponibles en el territorio ad hoc
4. Selección de un clúster, que implica la obtención de parámetros de configuración MTU, ESSID y canal, de acuerdo con la información publicada por los nodos miembro del clúster seleccionado en el territorio ad hoc.
5. Configuración de cada una de sus interfaces inalámbricas para asociarse al clúster o clústeres elegidos.
6. Creación de interfaces de capa 2 por cada interfaz inalámbrica que se asoció a un clúster en el paso anterior.
7. Activación del protocolo de enrutamiento para redes ad hoc en la interfaz elegida en el paso 5 y publicación de las redes creadas mediante mensajes broadcast.
8. Si el nodo tiene más de una interfaz inalámbrica, se debe efectuar verificación de la disponibilidad de canales para la creación de nuevos clústeres por sus demás interfaces.
9. Por cada interfaz sin uso, realizar pasos 3 a 6 del nodo originador.

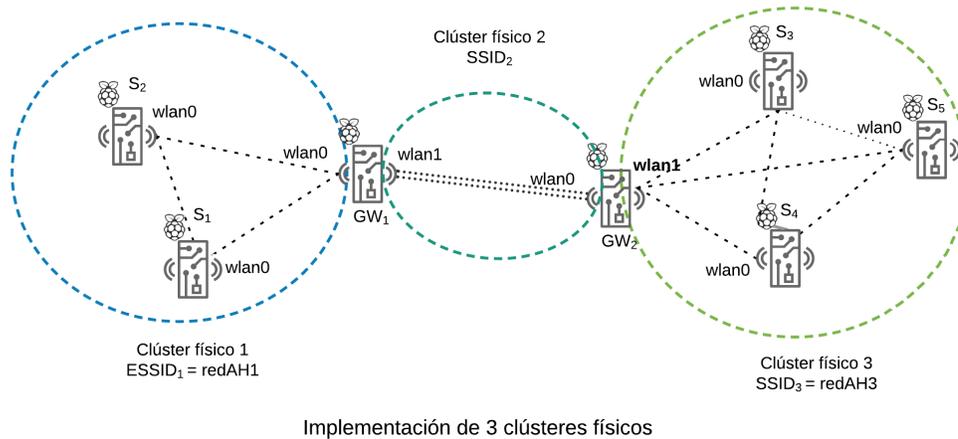
## 5.1. Implementación

### 5.1.1. Clúster físico

En el modelo teórico descrito en el Capítulo 4, en un territorio ad hoc se pueden armar  $n$  clústeres. Sin embargo, debido a que el estándar 802.11 define un total de 14 canales disponibles en la banda de frecuencias de 2.4 Ghz, el número máximo de clústeres que se pueden armar en un territorio ad hoc con un margen de interferencia relativo es 14. Por otra parte, en el estándar 802.11n también se definen 24 canales en la banda de 5 Ghz. En consecuencia, en esa banda el número teórico máximo de clústeres sería 24.

Para las estaciones inalámbricas, se eligieron equipos RaspberryPi, con las características que se presentan en la Tabla 5-1.

En la Figura 5-1 se muestra la disposición de los nodos para lograr el objetivo final del experimento base, que es la conformación de tres clústeres encadenados. Para implementar la comunicación interclúster se dotó a dos nodos con dos interfaces inalámbricas cada uno (nodos Gateway), los cuales a su vez conforman el clúster 2.



**Figura 5-1.:** Implementación de tres clústeres físicos

En el territorio ad hoc  $\mathcal{T}$ , los clústeres físicos se representarán con  $F_1$ ,  $F_2$  y  $F_3$ . Los nodos  $S_1$ ,  $S_2$ ,  $S_3$ ,  $S_4$ ,  $S_5$  tienen una sola interfaz inalámbrica, por lo tanto, tienen la menor jerarquía en el territorio ad hoc. Los nodos de mayor jerarquía son  $GW_1$  y  $GW_2$  porque tienen dos interfaces inalámbricas. Entonces, son los nodos gateway los que hacen posible la comunicación interclúster. Para proveer esa funcionalidad, es necesario configurar las rutas puente en los nodos gateway, ya que sin estas rutas, los clústeres quedarán aislados unos de otros y no será posible el paso del tráfico.

Con la implementación del mecanismo de clustering, se agruparon los nodos de la siguiente forma:

$$redAH_1 = \{ S_1, S_2, GW_1 \}$$

$$redAH_2 = \{ GW_1, GW_2 \}$$

$$redAH_3 = \{ S_3, S_4, S_5, GW_2 \}$$

Los conjuntos  $w_i$  representan las interfaces inalámbricas que componen cada uno de los nodos. En otras palabras, es el número de tarjetas de red 802.11:

$$w_{S_1} = \{ wlan0_{s_1} \}$$

$$w_{S_2} = \{ wlan0_{s_2} \}$$

$$w_{GW_1} = \{ wlan0_{gw1}, wlan1_{gw1} \}$$

$$w_{S_3} = \{ wlan0_{s_3} \}$$

$$w_{S_4} = \{ wlan0_{s_4} \}$$

$$w_{S5} = \{ wlan0_{s5} \}$$

$$w_{GW2} = \{ wlan0_{gw2}, wlan1_{gw2} \}$$

El conjunto de  $W_i$  es el conjunto de interfaces inalámbricas *wlan* que participan del clúster  $F_i$ :

$$W_1 = \{ wlan0_{s1}, wlan0_{s2}, wlan0_{gw1} \}$$

$$W_2 = \{ wlan1_{gw1}, wlan0_{gw2} \}$$

$$W_3 = \{ wlan0_{s3}, wlan0_{s4}, wlan0_{s5}, wlan1_{gw2} \}$$

**Tabla 5-1.:** Dispositivos usados en la implementación

| Modelo                              | Características  | # núcleos |
|-------------------------------------|--|-----------|
| Raspberry Pi Zero W Rev 1.1         | Procesador: ARMv6-compatible processor rev 7 (v6l)<br>32 bits<br>Disco Duro: 2 Gb<br>Memoria RAM: 512 M<br>Interfaz WiFi: 1 (Interna)<br>Version de Kernel: Linux 4.19.42+<br>S.O. Raspbian GNU/Linux 9 stretch                  | 1         |
| Raspberry Pi 1 Model B Plus Rev 1.2 | Procesador: ARMv6-compatible processor rev 7 (v6l)<br>Disco Duro: 8 Gb<br>Memoria RAM: 512 M<br>Interfaz WiFi: 1 Externa TP-link TL-WN721N<br>Versión de kernel: Linux raspberrypi 4.14.98+<br>S.O. Raspbian GNU/Linux 9 stretch | 2         |
| Raspberry Pi 3 Model B Rev 1.2      | Procesador: ARMv7 Processor rev 4 (v7l)<br>Disco Duro: 30 Gb<br>Memoria RAM: 1 Gb<br>Interfaz WiFi: 1 + 1 Externa TP-link TL-WN721N<br>Versión de kernel: Linux raspberrypi 4.9.59-v7+<br>S.O. Raspbian GNU/Linux 9.1 stretch    | 4         |

El conjunto de  $B_i$  es el conjunto de interfaces virtuales BATMAN *bat* de la capa de enlace, que participan del clúster  $F_i$ :

$$B_1 = \{ bat0_{s1}, bat0_{s2}, bat0_{gw1} \}$$

$$B_2 = \{ bat1_{gw1}, bat0_{gw2} \}$$

$$B_3 = \{ bat0_{s3}, bat0_{s4}, bat0_{s5}, bat1_{gw2} \}$$

El conjunto de tuplas con interfaces de un nodo gateway, el correspondiente clúster y canal al que están asociadas, se representa con la letra  $G$ :  $G_1 = \{ \langle wlan0_{gw1}, redAH1, 1 \rangle, \langle wlan1_{gw1}, redAH2,$

4 >}

$$G_2 = \{ \langle wlan0_{gw2}, redAH2, 4 \rangle, \langle wlan1_{gw2}, redAH3, 7 \rangle \}$$

El conjunto de los nodos gateway  $\gamma$  existentes en  $\mathcal{T}$  es igual a la unión de todos los conjuntos  $G_i$ .

$$\gamma \equiv \bigcup_{i=1}^n G_i$$

$$\gamma = \{ G_1, G_2 \}$$

La definición de los clústeres físicos es la siguiente:

$$F_1 = f(\text{redAH1}, 1532, 1, W_1, B_1)$$

$$F_2 = f(\text{redAH2}, 1532, 4, W_2, B_2)$$

$$F_3 = f(\text{redAH3}, 1532, 7, W_3, B_3)$$

Para establecer la conexión interclúster, en los nodos gateway se configuran rutas que habilitan el paso del tráfico entre los clústeres. También se pueden establecer reglas de tráfico si se desea mayor control sobre la interacción entre miembros de los clústeres.

Las reglas de enrutamiento se representan con  $E$  y las rutas con  $r$ . A partir esto, se establece que la regla para enlazar  $F_1$  y  $F_3$  de forma bidireccional está dada por las rutas configuradas en  $GW_1$  y  $GW_2$ , de forma que  $E_{1 \times 3} = \{ r(gw_1 \rightarrow gw_2), r(gw_2 \rightarrow gw_1) \}$

### Regla para la escalabilidad

Para garantizar la escalabilidad en el territorio ad hoc y provocar la división de la red y expansión del área de cobertura, es necesario establecer una regla que evite que los nodos gateway generen loops de clústeres innecesarios. Por lo tanto, una estación gateway con  $m$  interfaces puede generar  $m$  clústeres, si previamente no existen clústeres en el territorio ad hoc. De esta manera, dicho nodo puede originar clústeres para los cuales este será el punto de interconexión. Sin embargo, si ya existen clústeres ocupando ciertos canales, el nodo gateway con  $m$  interfaces que llega al territorio debe generar al menos un clúster en un canal libre y en consecuencia tendrá la opción de unirse como máximo a  $m - 1$  clústeres existentes. De esta forma, se facilita el encadenamiento de clústeres para ampliar el área de cobertura del territorio ad hoc. Si por ejemplo, en la estructura de la Figura 5-1, el nodo  $GW_1$  llega al territorio cuando ya se han creado los clústeres 1 y 3 de forma independiente, si no existe  $GW_2$ , entonces puede unirse por cada una de sus interfaces los clústeres 1 y 3 respectivamente (Figura 5-2). De otra manera, si ya existe  $GW_2$ , entonces  $GW_1$  deberá intentar enlazarse con el nodo gateway detectado. Para esto, debe unirse a uno de los clústeres existentes. Si se une al mismo clúster al que pertenece  $GW_2$ , debe crear un clúster diferente para generar la posibilidad de que otro nodo gateway pueda enlazarse con  $GW_1$  por un clúster diferente a los existentes. También puede unirse por la interfaz libre al clúster 1 y de esa manera concatenar los clústeres 1 y 3. Entonces, por la regla de escalabilidad será  $GW_2$  el que deba generar un nuevo clúster. En otro caso  $GW_1$ , puede unirse al clúster 1 y por la regla de escalabilidad debe desplegar un nuevo clúster para vincularse con  $GW_2$ , posibilitando la conformación de un clúster de gateways. (Figura 5-3).

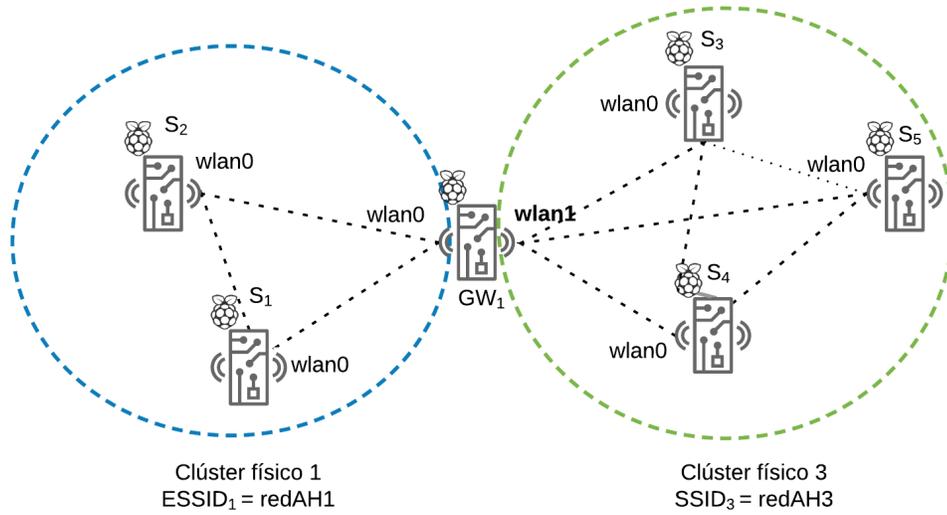


Figura 5-2.: Opción de clustering con una estación gateway  $GW_1$

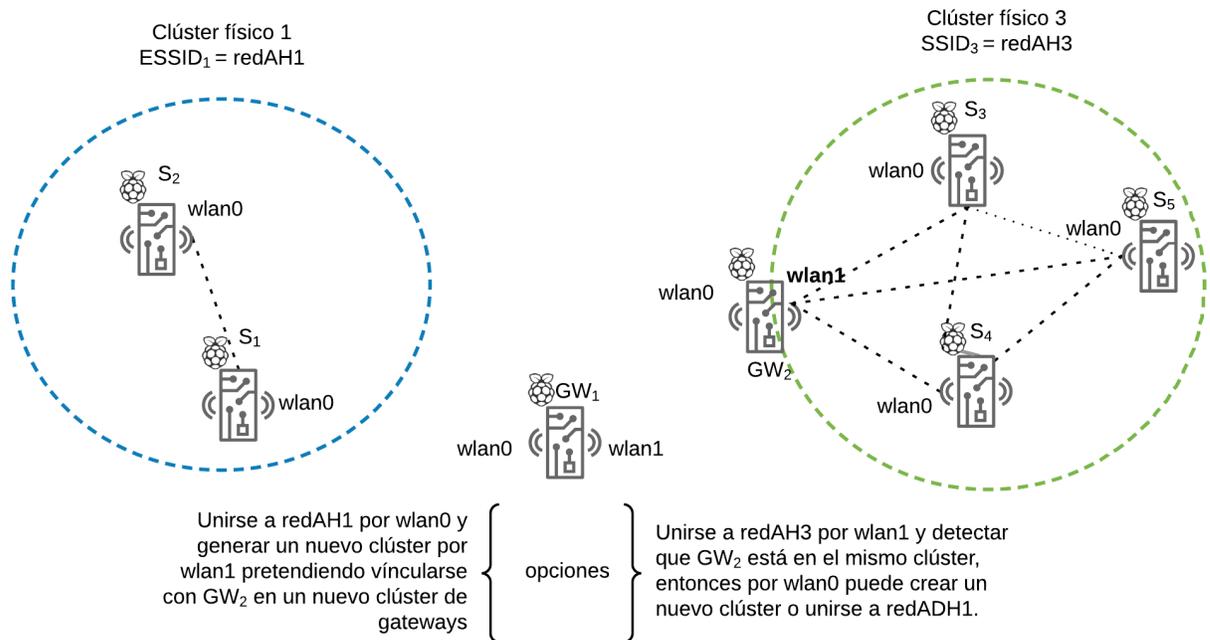


Figura 5-3.: Opciones de clustering siguiendo regla de escalabilidad

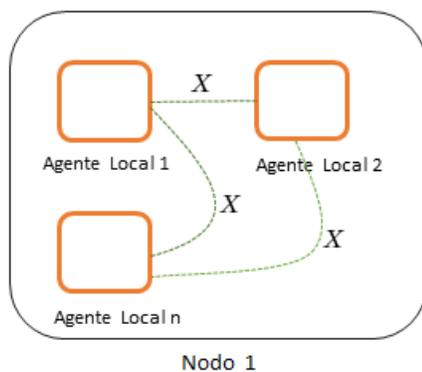
## Algoritmo

El pseudocódigo propuesto para la implementación de clustering se muestra en el cuadro del algoritmo 1. El procedimiento es el siguiente: en un arreglo se almacenan todos los canales de banda WiFi que se pueden usar. El nodo en cuestión realiza el proceso de escaneo de la red. Si en el territorio ad hoc hay redes IBSS, el nodo almacena los datos de los clústeres existentes. Si existe al menos un clúster ad hoc, entonces el nodo analiza cuantas interfaces WiFi tiene disponibles para usar. Si solo tiene una, se une a un clúster existente. Si tiene más de una, busca si en el territorio hay algún equipo gateway, es decir, algún nodo con jerarquía mayor a 1 y posteriormente selecciona una gateway entre las existentes y se vincula a uno de los clústeres que esta ha publicado. Con el resto de las interfaces que aún están libres, adquiere información de los canales usados por los clústeres y escoge los canales del arreglo que estén libres y si considera necesario crea nuevos clústeres. Si no hay nodos gateway en el territorio ad hoc, entonces puede unirse a clústeres diferentes por cada interfaz. Por el contrario, si no hay clústeres (redes IBSS) en el territorio ad hoc, por cada interfaz del nodo, puede crear un nuevo clúster, publicar su existencia y esperar a que otros nodos se unan.

### 5.1.2. Clúster lógico

Para la creación del clúster lógico se describen los dos escenarios: 1) Creación de clúster lógico dentro de un mismo nodo (Figura 5-4) y 2) creación de clúster lógico distribuido (entre nodos diferentes) (Figura 5-5).

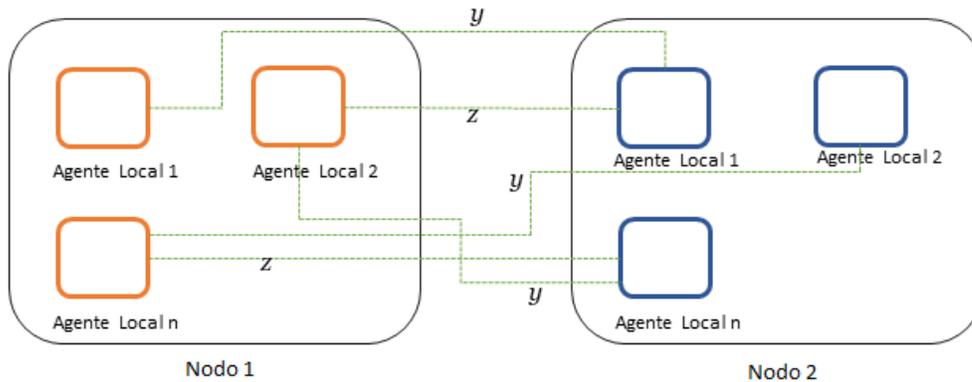
Para la creación de clúster lógico se parte de una clase Agente Local que se instancia múltiples veces dentro de un mismo nodo e implementa un método que permite la comunicación entre agentes locales. Para la creación y configuración de múltiples agentes locales, se usará una clase registro que permite controlarlos. Se crea en el sistema TLÖN un módulo de servicio de clustering o agrupamiento de agentes.



**Figura 5-4.:** Representación de un clúster lógico  $X$  en un solo nodo.

El agente local diseñado por el grupo de investigación TLÖN es un componente fundamental del sistema TLÖN. Este agente es una pieza de código escrita en lenguaje Python, que se encarga de adminis-

**Algorithm 1** Clustering**Ensure:** Wireless interfaces ON**Require:** Ad hoc Mode (B.A.T.M.A.N.) enable**Require:**  $count(w_i) \geq 1$  $w[] \leftarrow getInterfaces()$  $Channels \leftarrow [1, 2, 3, \dots, 14]$  $StartScan(T)$  $F[] \leftarrow getClustersAdHoc()$  $m \leftarrow len(w)$ **if**  $F > 0$  **then**    **if**  $len(w) > 1$  **then**        **for**  $f$  in  $F$  **do**             $getGW()$              $\gamma \leftarrow add\_G_i$         **end for**         $c \leftarrow m - 1$         **if**  $len(\gamma) \geq 1$  **then**            **for**  $1 : c$  **do**                 $selectCluster()$  in  $\gamma$                  $joinCluster()$             **end for**            **for**  $1 : (m - c)$  **do**                 $createCluster()$             **end for**        **else**            **for**  $1 : m$  **do**                 $selectCluster()$  in  $F$                  $joinCluster()$             **end for**        **end if**    **else**         $selectCluster()$  in  $F$     **end if****else**    **for**  $1 : m$  **do**         $createCluster()$     **end for****end if=0**



**Figura 5-5.:** Representación de clústeres lógicos distribuidos ( $z$  y  $y$ )

trar, monitorear y gestionar los recursos usados dentro del sistema. Las capacidades de comunicación del agente son posibles gracias al servicio de mensajería *Alfred* que se instala en los nodos y funciona sobre el protocolo BATMAN. Este servicio permite al agente la comunicación con otros módulos del sistema. Además de las funcionalidades de monitoreo, el agente es capaz de generar un log (archivo de registro) de las interacciones que éste tiene con el ambiente, es decir, con el territorio ad hoc. Otra de las funcionalidades importantes del agente es reservar recursos para la ejecución de aplicaciones distribuidas en la red, compartir la información entre los nodos vecinos y ejecutar detección de fallas (Zárate, 2016).

Sin embargo, hasta ahora el sistema TLÖN no tenía un mecanismo de clustering de agentes en un mismo nodo (departamentos TLÖN) y tampoco entre agentes de diferentes nodos (países TLÖN). Esta capacidad de agrupación se considera conveniente para distribuir tareas específicas en el sistema. Con el establecimiento de clústeres físicos la comunicación entre nodos en el nivel de red esta garantizada. De esa forma, es posible la conformación de clústeres lógicos en el territorio ad hoc mediante relaciones entre los agentes de diferentes nodos. Cada asociación entre agentes se etiquetará con un nombre y este será el identificador del clúster lógico. En la Figura 5-5, las conexiones de los agentes locales etiquetadas con el nombre  $x$  pertenecen a un clúster y las que tienen el nombre  $y$  pertenecen a otro clúster. Entonces, se puede decir que el clúster lógico se conforma a medida que se implementa el etiquetado de las comunicaciones entre agentes.

El propósito de la existencia de un clúster lógico es la ejecución de tareas de forma distribuida sobre los recursos del territorio ad hoc. Para compartir los recursos, el sistema TLÖN cuenta con un sistema operativo virtualizado orientado a redes ad hoc (S.O.V.O.R.A), descrito en Zárate (2016), que permite mapear y administrar los recursos físicos de la red. Gracias a este, los agentes pueden acceder o realizar operaciones sobre los los recursos virtualizados del sistema. Por lo tanto, la creación de clústeres de agentes locales agrega otro nivel de distribución más profundo de las funciones y servicios del sistema TLÖN.

En términos de software, los diferentes agentes locales son objetos instancias de la clase LocalAgent.py. Mediante una clase Registro.py, se generan múltiples instancias de estos agentes y se generan hilos de comunicación para enlazar los agentes. Por otra parte, la clase Adhoc.py permite crear los clústeres físicos; una vez establecidos se obtienen los datos necesarios para encontrar otros agentes que están disponibles en la red y son potenciales miembros del clúster lógico. El modelo propuesto agrega un elemento nuevo que es el Registro y crea nuevas funcionalidades para implementar el servicio clustering. Los tres componentes del modelo de clustering para el sistema TLÖN se muestran en la Figura 5-6. El componente nuevo es Registro y se modificó el código fuente original para aplicar mejoras a los módulos LocalAgent y Adhoc.



**Figura 5-6.:** Módulos del mecanismo de clustering para el sistema TLÖN

Los parámetros necesarios para crear el clúster lógico son: la dirección IP del nodo donde habita el agente ( $ip$ ), el identificador del agente ( $id$ ) y el puerto de comunicaciones del agente ( $port$ ). En este caso, el nombre del enlace entre agentes ( $k$ ) es el identificador del clúster lógico.

Para el modelo el agente lógico  $A$  se representa por la n-tupla que contiene los parámetros anteriores más el identificador del recurso que maneja. Conforme a esto, se define de la siguiente forma:

$$A = \langle id, ip, port, k, R_j \rangle$$

Donde  $R_j$  son los recursos de un nodo y aunque pueden ser de diferentes tipos, se agrupan en un mismo conjunto de recursos básicos definidos mediante el siguiente conjunto de tuplas:

$\mathcal{R} = \{ C, M, S, W, IO \}$ , donde:

→  $C$  = CPU

→  $M$  = Memoria RAM

→  $S$  = Almacenamiento (Storage)

→  $W$  = Interfaces de comunicación inalámbrica

→  $IO$  = Dispositivos de entradas y salida (Inputs/Outputs)

El clúster lógico, que se denotará con  $\ddot{O}$ , se define como el conjunto de todos los agentes locales asociados con un mismo nombre de clúster  $k$ . Por otra parte, el conjunto de nodos que contienen los agentes que participan en el clúster lógico, se denota con el conjunto:

$$S(k) = \{ S_1, S_2, \dots, S_n \}$$

Cada nodo aporta un conjunto de agentes al clúster, por esta razón, el clúster lógico se puede analizar como un conjunto de subconjuntos.

Aunque teóricamente el clúster lógico parece no tener límites para agrupar agentes locales, en la implementación real, este límite está dado por las capacidades del procesador del dispositivo. Una de las características más importantes de los computadores actuales es que cuentan con unidades de procesamiento central (CPUs), también llamados procesadores, con múltiples núcleos (cores). Cada núcleo es un componente que maneja procesos y ejecuta acciones sobre el sistema. A mayor cantidad de núcleos, mayor será la capacidad de ejecutar en menor tiempo múltiples tareas de forma concurrente y/o paralela. De esta forma, el número de núcleos y tecnología de la CPU determina la forma en que se pueden virtualizar y manipular los recursos de todo el sistema.

En la actualidad, algunos equipos de alta gama, con altas capacidades de procesamiento, tienen una tecnología que permite la partición de los recursos por cuotas. De esa manera, a un programa o aplicación específico se le podría asignar una fracción del procesador, por ejemplo, el 25 % de CPU. En consecuencia, el control sobre los recursos que maneja puede subdividirse para aprovechar al máximo las capacidades de procesamiento. Sin embargo, algunos equipos como los modelos de RaspberryPi Zero, B1, 2 y 3 no permiten una asignación de cuotas de recursos, por lo tanto esta partición porcentual no es posible en el contexto del experimento realizado en este trabajo.

A partir de esto, surge la limitación del modelo solución planteado respecto al tamaño máximo de un clúster lógico, porque en este caso, el número máximo de agentes es igual al número de núcleos del equipo RaspberryPi. Sin embargo, el modelo se puede extender para ser implementado en arquitecturas que permitan el manejo de cuotas de recursos de CPU. En ese caso, el número máximo de agentes locales que se podría desplegar sería mayor.

Con la tecnología de las RaspberryPi solo se puede asignar una entidad o agente local por cada núcleo para manejar los recursos del sistema. Por lo tanto, en un nodo el número máximo de agentes que se puede desplegar es igual al número de núcleos ( $CoreN$ ) de su procesador o unidad de procesamiento central.

Se presenta la definición por comprensión del clúster lógico unitario o departamento TLÖN, de la siguiente forma:

$$\ddot{O} = \{ A_i : i \in \mathbb{N}, i \leq CoreN \wedge \forall i, k \in A_i \}$$

El clúster lógico distribuido o país TLÖN es el conjunto de Agentes  $A_{js}$ , donde  $j$  es el identificador del agente y  $s$  es el identificador del nodo que contiene el agente cuyas conexiones tienen el mismo nombre  $k$ . Se representa de la siguiente forma:

$$\ddot{O}_k = \{ A_{js} : j, s \in \mathbb{N}, j \leq CoreN(s) \wedge \forall j k \in A_{js} \}$$

El número máximo de elementos (agentes) que puede contener un clúster lógico es la suma del número de núcleos  $CoreN$  de cada nodo  $s$ . Como se observa en la siguiente fórmula:

$$| \ddot{O} | = \sum_{s=1}^n CoreN_s$$

Por ejemplo, un país TLÖN  $\ddot{O}_z$  está conformado por 3 agentes, cada uno ubicado en un nodo diferente ( $|S|=3$ ), cuyas capacidades de CPU son  $CoreN_1 = 3$  núcleos,  $CoreN_2 = 2$  núcleos y  $CoreN_3 = 2$  núcleos. Entonces el número máximo de agentes asociados que puede tener  $\ddot{O}_z$  es:

$$| \ddot{O}_z | = 3 + 2 + 2 = 7$$

Dicho de otra forma, el clúster lógico de agentes con conexiones marcadas con el nombre  $z$  contiene los subconjuntos de agentes  $A_j$ , donde  $j$  es función del número de núcleos de  $s$ , denotado como  $CoreN(s)$  y donde  $s$  es el identificador del nodo que pertenece al conjunto  $S$  de nodos que participan en el clúster lógico.

La definición por extensión del conjunto que representa el clúster lógico identificado por  $z$  cuando está conformado con su máxima capacidad, es la siguiente:

$$\ddot{O}_z = \{ A_{11}, A_{21}, A_{31}, A_{12}, A_{22}, A_{13}, A_{23} \}$$

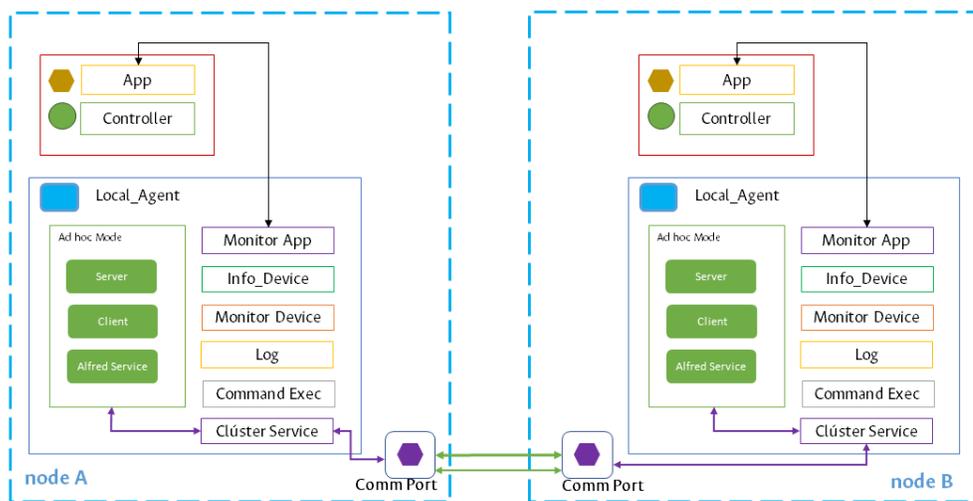
### Comunicaciones entre agentes locales

Con el objetivo de implementar la comunicación entre agentes locales, en primera instancia una entidad de registro presente en el nodo invoca al módulo ad hoc para ejecutar la creación de clústeres físicos en el territorio ad hoc; mediante el proceso descrito anteriormente. Una vez establecidas las conexiones físicas, se dispone de un ambiente para desplegar los agentes. Posteriormente, el registro ejecuta acciones para establecer el número máximo de agentes locales que se puede desplegar y aplica la configuración del clúster lógico que se va a crear.

Una vez el registro despliega los agentes en el nodo, el agente local habilita los puertos por los cuales puede comunicarse con otros agentes. Para encontrar otros agentes, realiza consultas en el territorio ad hoc y cuando encuentra un nodo vecino realiza una solicitud de conexión y en seguida establece un hilo de comunicaciones etiquetado a través del cual se comunicará con el agente destino. El mecanismo de comunicación de agentes se basa en la manipulación de hilos de la clase *Thread* de Python.

En concreto, los agentes locales construirán el clúster a medida que van conformando una malla de procesos comunicados y etiquetados. Implementando así una comunicación de agentes distribuidos en el sistema TLÖN.

La Figura 5-7 muestra los módulos del agente local del sistema TLÖN, ahora a través de las interacciones entre el módulo ad hoc y el servicio de clustering que manipula el puerto de comunicaciones (Comm Port), a través del cual los agentes se pueden comunicar.



**Figura 5-7.:** Comunicación de agentes locales

Las conexiones a las que se hace referencia en el ámbito del clúster lógico, no son conexiones en el medio de transmisión como las del clúster físico. Las líneas verdes que enlazan los módulos *CommPort* de cada agente en la Figura 5-7, simbolizan hilos que comunican los procesos de ambos agentes.

## 6. Pruebas y resultados

A continuación se describen las pruebas efectuadas sobre los clústeres implementados. Es importante resaltar que debido a que este trabajo es empírico, la validación consistió en iteraciones de experimentos para analizar el comportamiento de los clústeres. Durante experimentos se realizó un ejercicio de análisis y observación del fenómeno estudiado a través de la medición de los parámetros de latencia y throughput de cada clúster. No se realizaron comparaciones con otros estudios debido a que en la revisión de la literatura se identificó que la mayoría de trabajos sobre el tema de clustering en redes inalámbricas corresponden a simulaciones. Algunos ejemplos de esas investigaciones se pueden encontrar en Elhdhili, Ben Azzouz, y Kamoun (2008), Neethu y Singh (2015), Papakostas y Katsaros (2014), Tiwana y Gupta (2009), Jiankang, Yingying, y Shifeng (2016), entre otros.

En contraste con las investigaciones teóricas, un trabajo experimental tiene la ventaja de que los resultados corresponden a escenarios reales. Aunque, por otra parte, presenta dificultades por la cantidad de tiempo invertido para lograr las condiciones de ejecución del experimento. Para este trabajo, la implementación del clustering involucró tareas previas en cada nodo y preparación del ambiente de trabajo. Algunas de esas tareas fueron: selección de hardware e instalación de controladores de interfaces de red, instalación y configuración de sistema operativo, paquetes de software, protocolos, herramientas auxiliares y configuración de la red de gestión de los nodos.

Para observar el funcionamiento de los clústeres y su comportamiento en la banda de frecuencias 2.4 Ghz del estándar 802.11 se utilizó la herramienta *Wireshark*. Esta es una herramienta de software libre, que sirve para analizar protocolos de redes de comunicaciones. Con esta herramienta es posible capturar el tráfico de las redes y detectar problemas en las mismas.

Entre los protocolos que soporta *Wireshark* se encuentra el protocolo BATMAN. Adicionalmente, para capturar tráfico de protocolos inalámbricos fue necesario el uso de un adaptador *AirPcap* que es un dispositivo hardware USB que se acopla con *Wireshark* para ofrecer una completa visibilidad de los paquetes en múltiples canales de 802.11 a/b/g/n y también con la ventaja de que genera reportes.

Para las pruebas se implementó el mecanismo de clustering conformando tres clústeres de redes ad hoc inalámbricas. Los parámetros de configuración de cada clúster se muestran en las tablas **6-1**, **6-2** y **6-3**.

Las pruebas efectuadas incluyen un análisis del tráfico inter e intra-clúster. Para esto se llevaron a cabo medidas basadas en envío de paquetes de forma continua entre los nodos y de estos se extrajo

| CLUSTER 1  |  |                        |                 |             |
|--|--|------------------------|-----------------|-------------|
| Nodos Jerarquía 1<br>2 Rpi ZeroW (1 Interfaz WiFi c/u) | Nodos Gateway<br>1 Rpi 3 Modelo B(2 interfaces WiFi c/u) | Canal<br>1 (2.412 GHz) | ESSID<br>redAH1 | MTU<br>1500 |

**Tabla 6-1.:** Configuración clúster 1

| CLUSTER 2              |  |                        |                 |             |
|------------------------|--|------------------------|-----------------|-------------|
| Nodos Jerarquía 1<br>0 | Nodos Gateway<br>2 Rpi 3 Modelo B(2 interfaces WiFi c/u) B | Canal<br>4 (2,427 GHz) | ESSID<br>redAH2 | MTU<br>1500 |

**Tabla 6-2.:** Configuración clúster 2

| CLUSTER 3  |   |                            |                 |             |
|--|---|----------------------------|-----------------|-------------|
| Nodos Jerarquía 1<br>1 Rpi Model 1<br>1 Rpi 3 Model B<br>(1 Interfaz WiFi c/u) | Nodos Gateway<br>1 Rpi 3 Modelo B (2 interfaces WiFi c/u) | Canal<br>7 (2,442 GHz GHz) | ESSID<br>redAH3 | MTU<br>1500 |

**Tabla 6-3.:** Configuración clúster 3

los datos de retardo de envío de paquetes (latencia) y el tasa de transferencia efectiva. De cada una de estas medias se calculó el promedio y la desviación estándar. La herramienta usada para medir el *throughput* fue el siguiente comando *batctl*:

```
# batctl -m [interfazbatman] tp [ direccionMACdestino ]
```

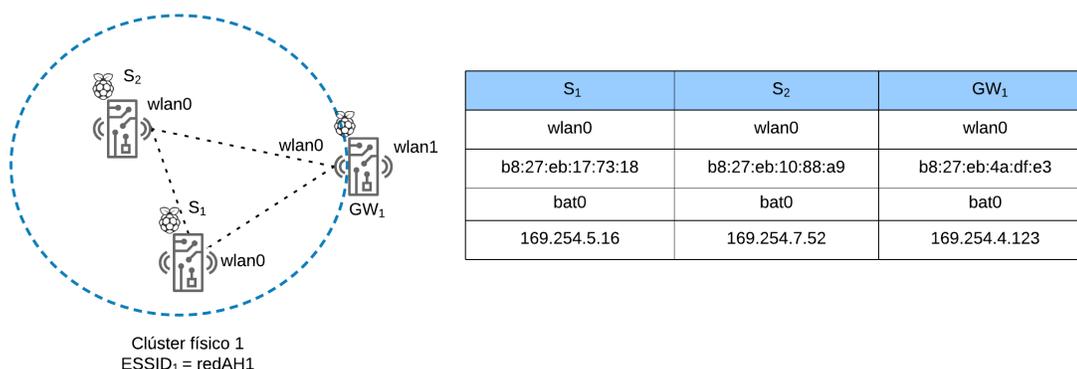
Los resultados se presentan de la siguiente manera: primero, se muestran los resultados para cada clúster por separado (tráfico intra-clúster), y en segundo lugar, se muestran los resultados del tráfico global de los tres clústeres en funcionamiento, sobre los cuales se ejecutaron pruebas de envío de paquetes entre los diferentes nodos.

## 6.1. Comunicación intra-clúster

### 6.1.1. Análisis del tráfico en clúster 1

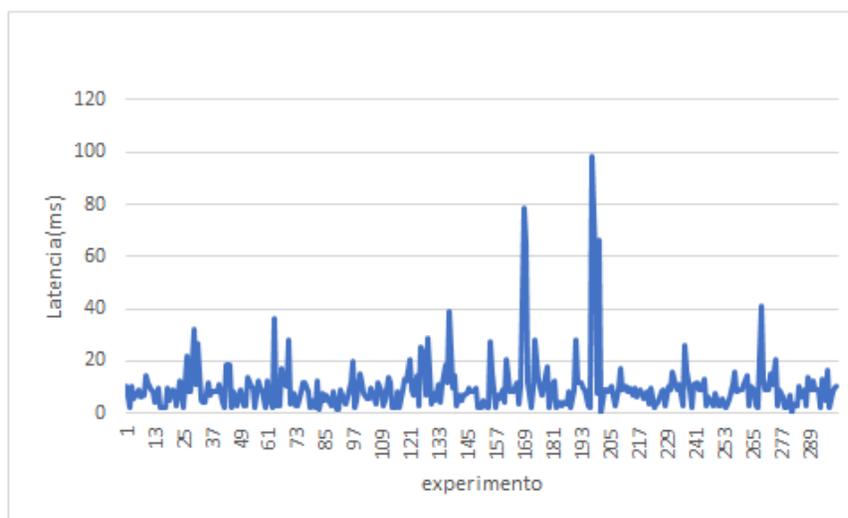
Para las mediciones del tráfico se generó un clúster ad hoc con la estructura que se observa en la Figura 6-1 . Se realizaron pruebas de envío de paquetes entre  $S_1$  y  $S_2$ , de  $S_2$  a  $GW_1$ ,  $S_1$  a  $GW_1$  y de  $GW_1$  a  $S_1$  y  $S_2$ . Por cada interfaz inalámbrica de un nodo, el protocolo BATMAN crea una interfaz virtual identificada con el nombre *bat*. Con esa interfaz virtual se enlazan todos los equipos en una malla. A cada interfaz *bat* el protocolo asigna una dirección IP que sirve como identificador dentro de la malla.

Otro parámetro que usa el protocolo BATMAN, es la dirección MAC de cada interfaz inalámbrica, con esta es posible realizar las pruebas de *ping* en la capa 2 de la red.



**Figura 6-1.:** Escenario de prueba clúster 1

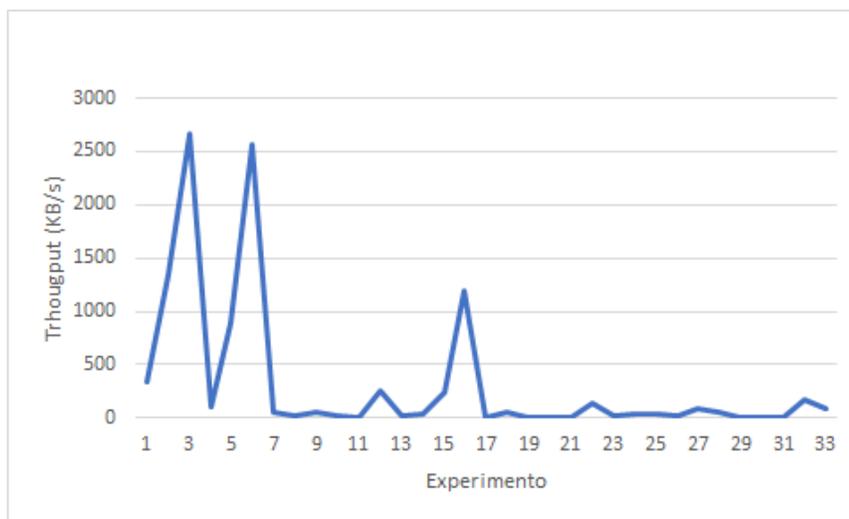
Se observó que el clúster 1, configurado en el canal 1 presentó un buen comportamiento en el momento de establecer la red. Los nodos se conectaban entre sí rápidamente y en las pruebas de retardo, no se observó pérdida de paquetes. En general, este clúster fue el que presentó mejor desempeño. La Figura 6-2 muestra el tiempo que tardan en ser recibidos los paquetes, enviados mediante el comando *ping*, entre los nodos del clúster 1. El eje horizontal muestra el número de veces (número de experimentos) que se realizó el envío del comando *ping* entre los nodos del clúster y el eje vertical muestra el tiempo en milisegundos que tardó en ser transmitido y obtener una respuesta.



**Figura 6-2.:** Latencia en clúster 1

En la Figura 6-3 se observa el envío sucesivo de ráfagas de paquetes y su correspondiente tasa de éxito en la entrega. Se observa un pico de rendimiento en los primeros instantes, cuando se enviaron trazas

de datos de 14 MB. Dado que los paquetes llegaron a su destino rápidamente se observa un disminución a partir del experimento número 7. De todas formas, ante cualquier evento de envío de paquetes de gran tamaño, esta medida de flujo se incrementaba. Esto nos demostró un buen desempeño del clúster 1.



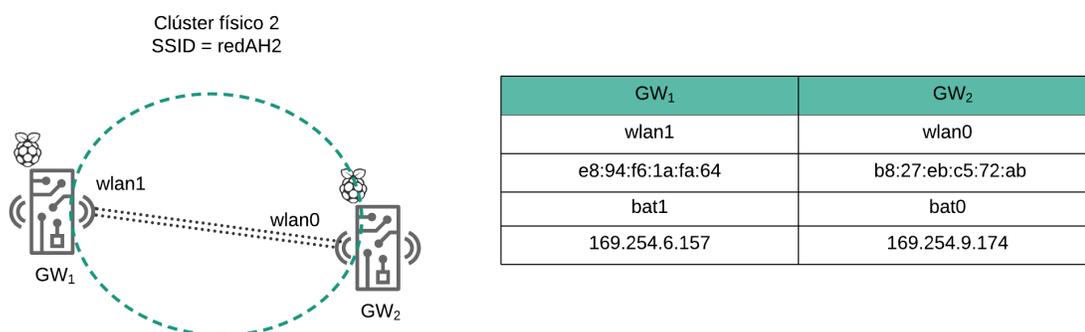
**Figura 6-3.:** Throughput clúster 1

|                      | Latencia [ms] | Throughput [Kb/s] |
|----------------------|---------------|-------------------|
| Promedio:            | 9,815         | 319,95            |
| Desviación estándar: | 10,545        | 676,90            |

**Tabla 6-4.:** Medía y desviación estándar de las mediciones en clúster 1

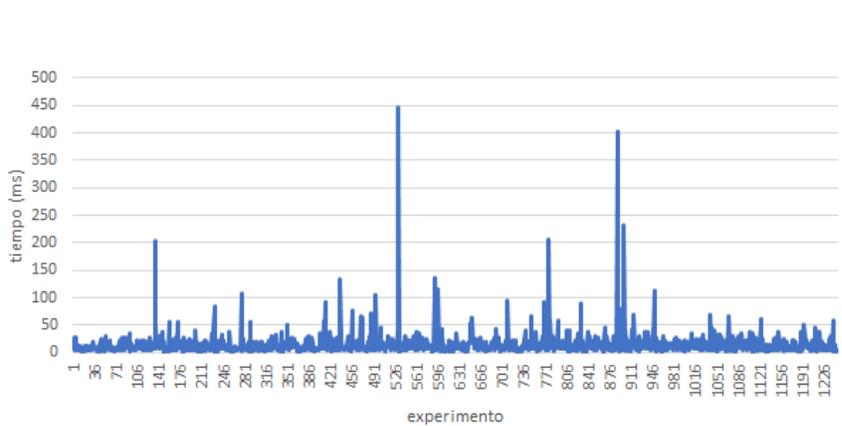
### 6.1.2. Análisis del tráfico en clúster 2

La Figura 6-4 muestra la estructura del clúster 2, así como las direcciones MAC y BATMAN usadas para las pruebas. Este clúster, conformado con dos equipos gateway, presentó más problemas que los otros dos en el momento de establecerse el vínculo entre los nodos. Cuando se configuró en el canal 4 se observó que los nodos no se enlazaban y el clúster tardaba algunos minutos en conformarse. Al analizar la ocupación del espectro en ese canal, se observó que existían otras redes operando sobre el mismo canal y en canales cercanos, en consecuencia, la interferencia no permitía establecer el clúster. Sin embargo, cuando este clúster se configuraba en el canal 11, menos usado en ese momento, el clúster 2 se establecía mucho más rápido. Esto nos demostró que la interferencia co-canal ejerce un efecto negativo en el establecimiento del clúster.



**Figura 6-4.:** Escenario de prueba clúster 2

De manera semejante a lo realizado en el clúster 1, se enviaron paquetes entre los nodos y se observó su comportamiento. Durante las pruebas, en este clúster se presentaron pérdidas de paquetes hasta en un 7 %. La figura 6-5 muestra que se realizó el experimento un mayor número de veces respecto a los otros clústeres, esperando que el comportamiento se normalizara con el paso del tiempo. Sin embargo, se evidenció un mayor retardo en este clúster. Al contrario de lo que sucedía con los otros clústeres, el tiempo de descubrimiento de vecinos era notoriamente elevado y en muchas ocasiones era necesario reiniciar las interfaces inalámbricas para que los nodos se enlazaran.



**Figura 6-5.:** Latencia en clúster 2

También se observó que durante algunos experimentos el rendimiento en la red (Figura 6-6) no era el más óptimo y que se perdía por completo la conexión con los nodos vecinos, por lo tanto, la tasa de entrega de paquetes se reducía a causa de la inestabilidad de la conexión.

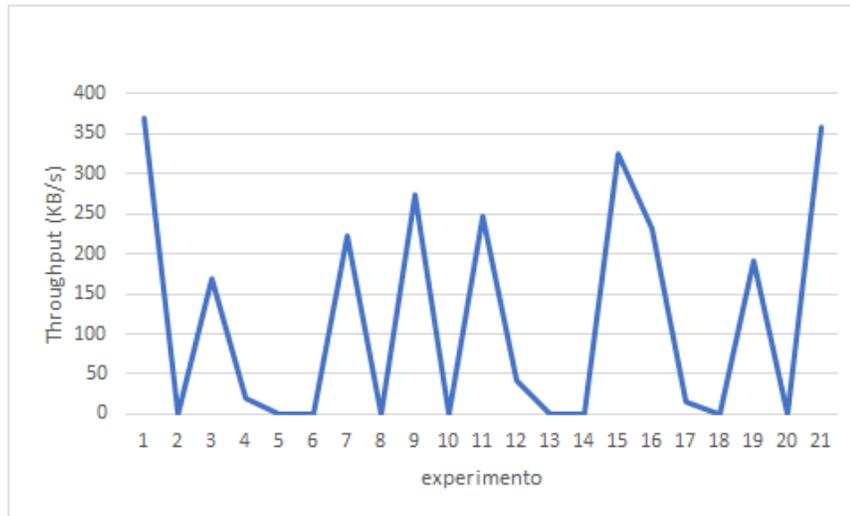
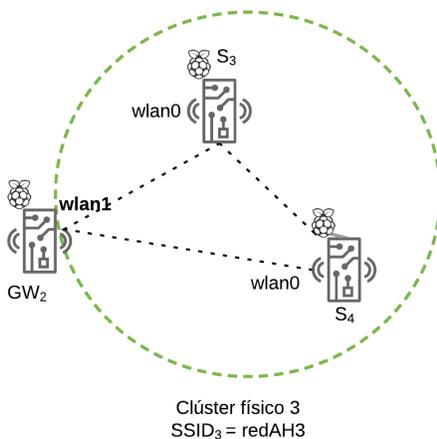


Figura 6-6.: Throughput en clúster 2

|                      | Latencia [ms] | Throughput [Kb/s] |
|----------------------|---------------|-------------------|
| Promedio:            | 15,04         | 220,79            |
| Desviación estándar: | 24,07         | 172,64            |

Tabla 6-5.: Medía y desviación estándar de las mediciones en clúster 2

### 6.1.3. Análisis del tráfico en clúster 3

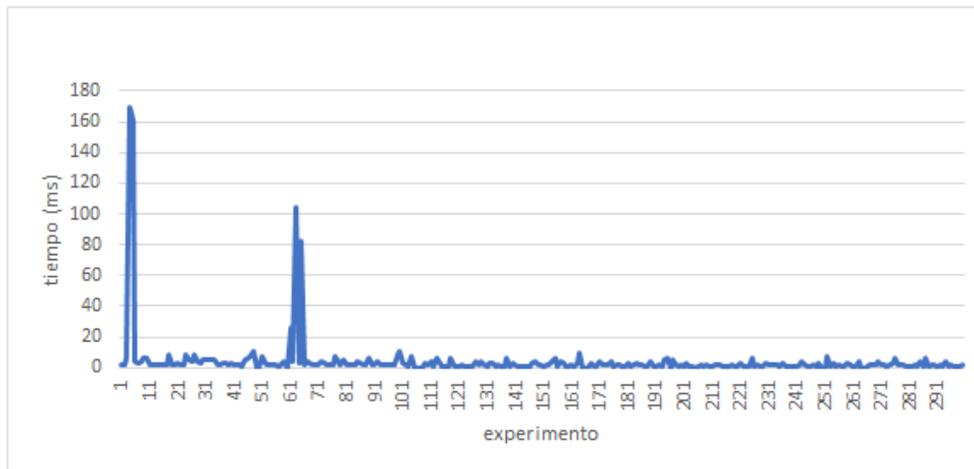


| S <sub>3</sub>    | S <sub>4</sub>    | GW <sub>2</sub>   |
|-------------------|-------------------|-------------------|
| wlan0             | wlan0             | wlan1             |
| e8:94:f6:1a:f9:ce | e8:de:27:af:fc:b6 | e8:94:f6:1a:69:01 |
| bat0              | bat0              | bat1              |
| 169.254.11.145    | 169.254.8.111     | 169.254.6.185     |

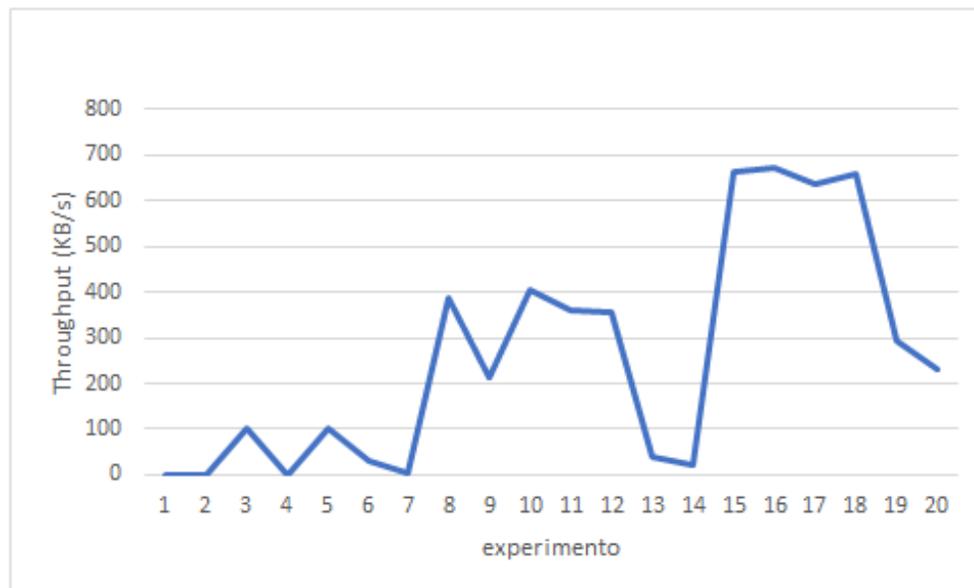
Figura 6-7.: Escenario de prueba clúster 3

Durante las pruebas de retardo en el clúster 3, se observó un promedio de pérdida de paquetes de 4%. La Figura 6-8 presenta la latencia en milisegundos durante experimentos sucesivos. Las princi-

pales pérdidas se presentan en los primeros experimentos. Se evidencia que a medida que los nodos van intercambiando más paquetes esta pérdida se disminuye hasta un 0 %. Se concluye que entre más tiempo permanezcan comunicándose, más estable será el enlace entre los nodos, entre menos paquetes se intercambien, el registro interno de vecinos se ira actualizando detectando que no se ha presentado actividad con algún nodo y será más fácil que se pierda la conexión por lo tanto será mas probable la perdida de paquetes.



**Figura 6-8.:** Latencia en clúster 3



**Figura 6-9.:** Throughput en clúster 3

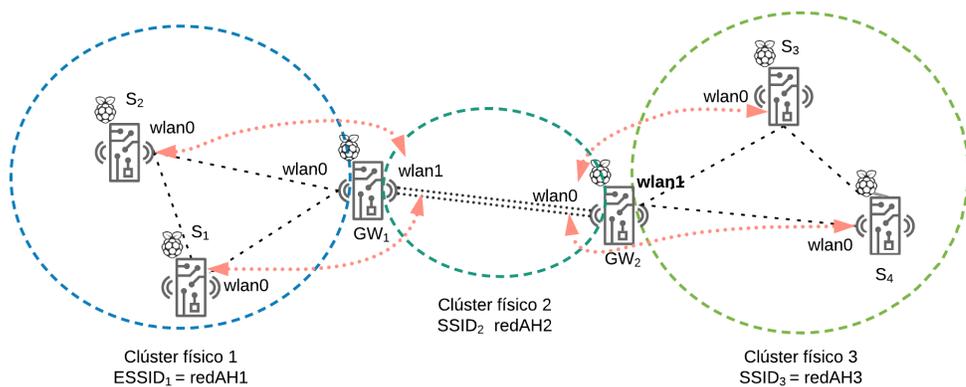
|                      | Latencia [ms] | Throughput [Kb/s] |
|----------------------|---------------|-------------------|
| Promedio:            | 10,89         | 259,01            |
| Desviación estándar: | 15,36         | 248,08            |

**Tabla 6-6.:** Medía y desviación estándar de las mediciones en clúster 3

## 6.2. Comunicación inter-clúster

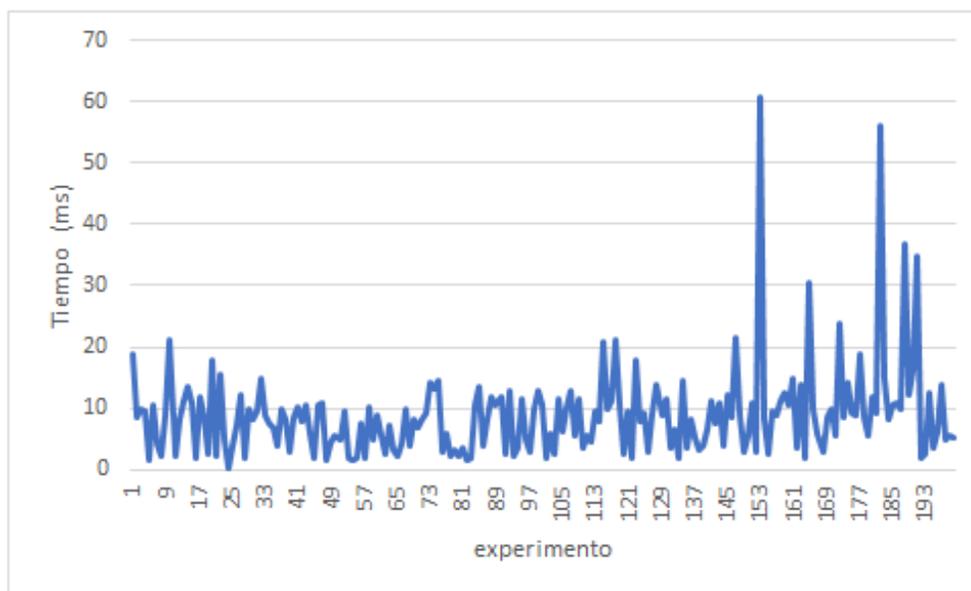
La comunicación inter-clúster es posible gracias a dos factores: la configuración de rutas en los nodos gateway y la activación de la opción *bonding* del protocolo BATMAN. Gracias a al mecanismo de *bonding*, que permite la activación alterna de las interfaces de red, se puede habilitar el paso de tramas por las rutas en nodos duales, es decir, aquellos con dos interfaces de red inalámbrica. Esto provee una capacidad de optimización multi-interfaz que hace que los nodos puedan actuar como punto de enrutamiento entre dos redes ad hoc en malla. Cuando se activa en los nodos gateway dicha opción, las dos interfaces de un nodo pueden ser enlazadas para que trabajen de forma colaborativa en la tarea de distribución del tráfico.

La Figura 6-10 muestra el escenario de comunicación entre los tres clústeres. Las flechas de color indican las rutas de los paquetes en las pruebas comunicación inter-clúster.



**Figura 6-10.:** Escenario de comunicación inter-clúster

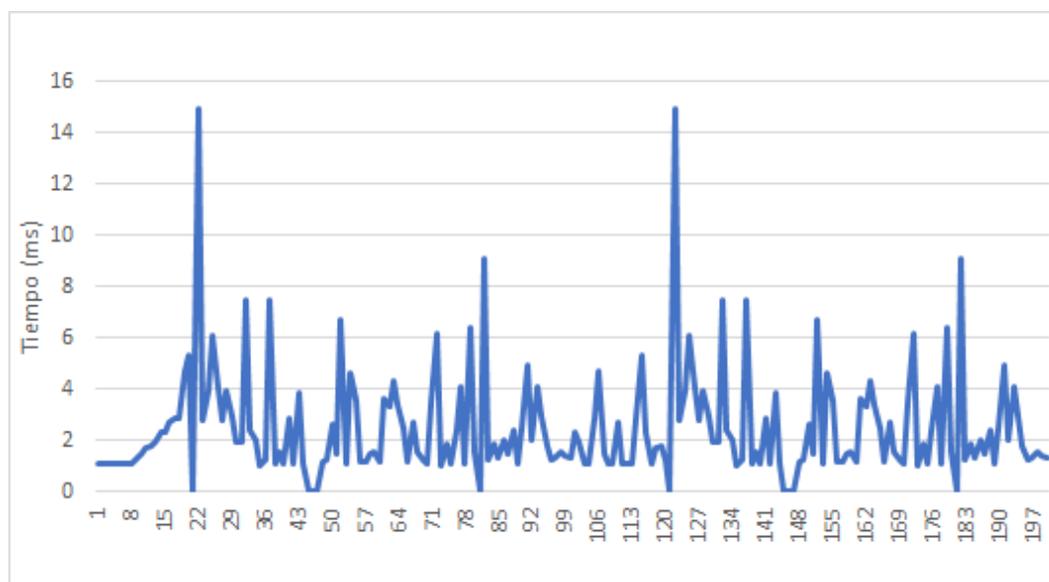
Para probar la conectividad inter-clúster se realizaron pruebas de envío de paquetes entre las diferentes interfaces de los nodos gateway hacia el clúster vecino. De la siguiente manera: De la interfaz  $bat_1$  de  $GW_1$  hacia la interfaz  $bat_0$  de  $S_0$  y  $S_1$  y en sentido contrario. De esta forma se observó conectividad entre los clústeres 1 y 2. Por otra parte, en el clúster 3 se realizó el envío desde los nodos  $S_3$  y  $S_4$  hacia la interfaz  $bat_0$  de  $GW_2$  y posteriormente desde el clúster 2 a los nodos del clúster 3. Lo más importante para estas pruebas, es lograr establecer las rutas correctas, pues el protocolo BATMAN, asigna de forma aleatoria las direcciones IP de sus interfaces virtuales  $bat_i$  cada vez que estas se activan. Si las están rutas mal configuradas en los nodos gateway se impedirá el paso del tráfico.



**Figura 6-11.:** Latencia en comunicación entre clúster 1 y clúster 2

La Figura 6-11 se observa la latencia cuando se realizaron 200 envíos del comando *ping* entre los clústeres 1 y 2.

En la Figura 6-12 se observa el tiempo de envío de los paquetes entre nodos del clúster 3 y el clúster 2. En los experimentos no se observó una pérdida de paquetes significativa.

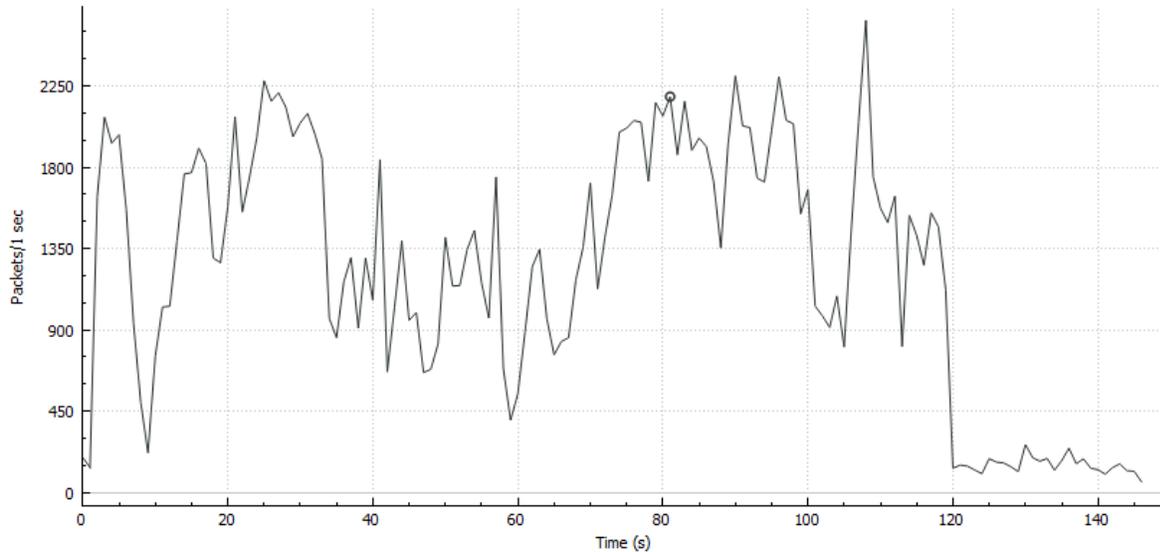


**Figura 6-12.:** Latencia en comunicación entre clúster 2 y clúster 3

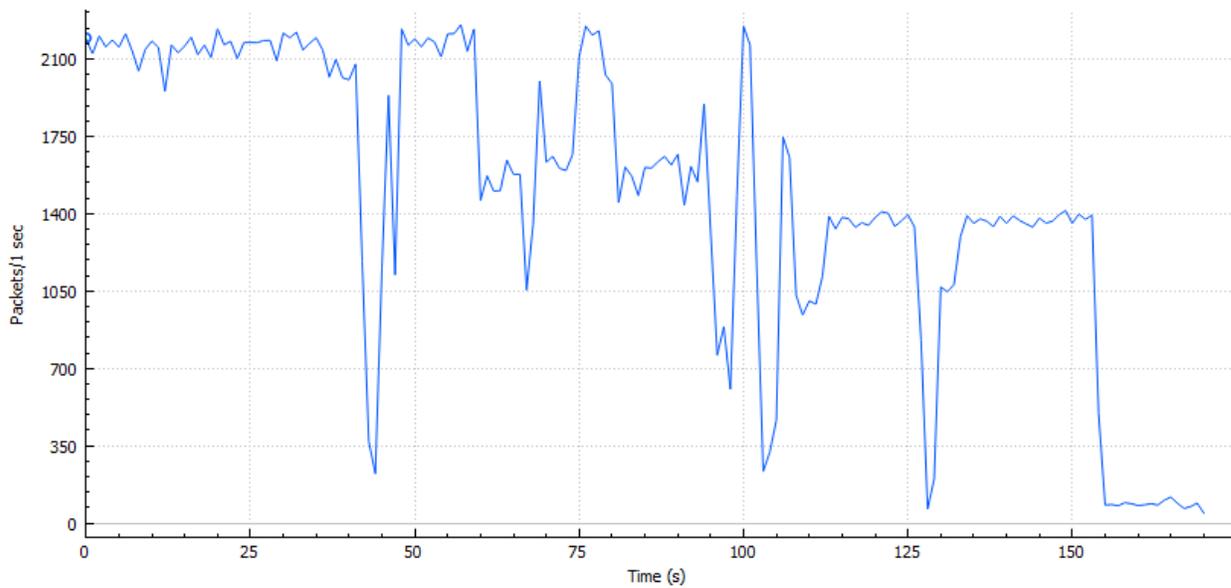
|                      | Latencia clústeres 1-2 | Latencia clústeres 2-3 |
|----------------------|------------------------|------------------------|
| Promedio:            | 8,908                  | 2,487                  |
| Desviación estándar: | 7,524                  | 2,145                  |

**Tabla 6-7.:** Medía y desviación estándar de las mediciones de latencia entre los tres clústeres

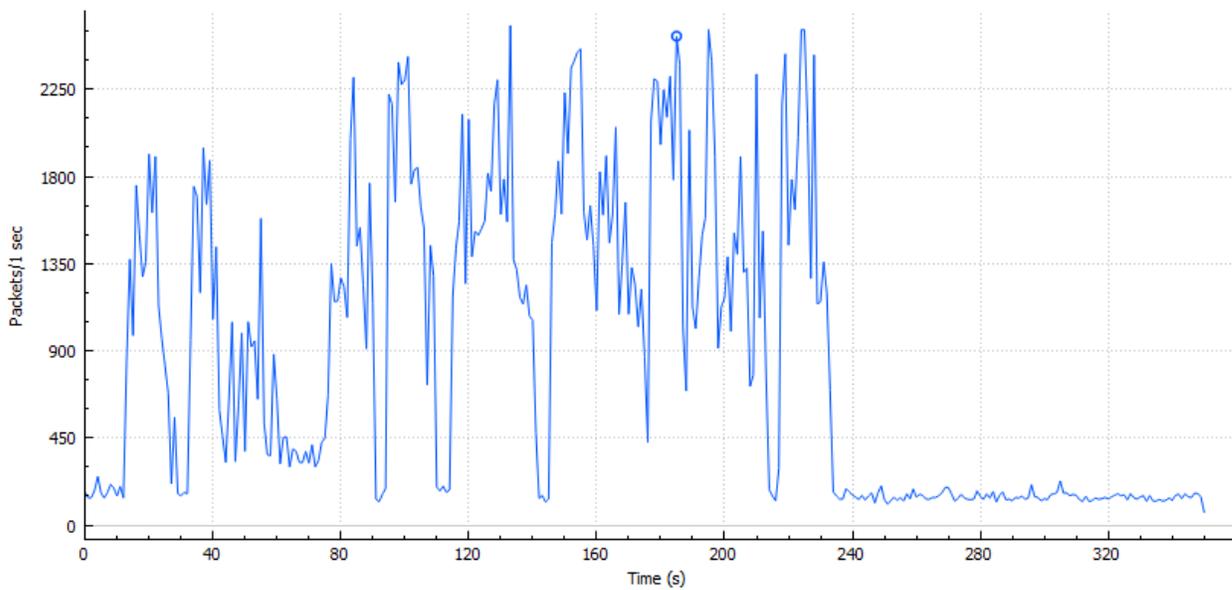
Respecto al análisis del tráfico global inter-clúster en un principio se planeo hacer uso de las herramientas de captura *AirPcap* y *Wireshark*. Se pretendía observar simultáneamente todo el tráfico de los tres clústeres. Sin embargo, se identificó un inconveniente a causa de las limitaciones de modelo del adaptador *AirPcap*. Dado que este permitía capturas de tráfico en un solo canal de la banda 2.4Ghz. De forma contraria a lo que se creía, no fue posible visualizar el tráfico de todos los clústeres de forma simultánea, considerando que para este trabajo no disponia con herramientas que tuvieran tales capacidades. A pesar de esto, durante el despliegue de los tres clústeres se realizaron las capturas de forma independiente, una por canal de operación y de estas se obtuvo las Figuras 6-13, 6-14 y 6-15, que muestran el tráfico de cada clúster durante los experimentos de comunicación inter-clúster. Los picos que se observan en las gráficas corresponden a los momentos en que se efectuó el envío de tramas de paquetes de 14 MB entre nodos.



**Figura 6-13.:** Tráfico clúster 1 durante comunicación con clúster 2



**Figura 6-14.:** Tráfico clúster 2 durante comunicación con clúster 1 y 3



**Figura 6-15.:** Tráfico clúster 3 durante comunicación con clúster 2

## 7. Conclusiones y recomendaciones

Como resultado del ejercicio de investigación, exploración, análisis y experimentación se adquirieron nuevos conocimientos en áreas relacionadas al fenómeno investigado. Se logró establecer una propuesta diferente en comparación con los enfoques comúnmente encontrados en la revisión de la literatura, respecto a la jerarquía en un clúster sobre redes ad hoc. De forma gradual se revelaron los postulados que permiten ahora presentar las conclusiones de este trabajo. Estas pueden dar una nueva perspectiva sobre cómo implementar la escalabilidad y la organización en las redes ad hoc para procesamiento distribuido. Además, pueden servir como referente para futuros trabajos de investigación.

### 7.1. Conclusiones

- Un mecanismo de clustering para redes ad hoc es apropiado para complementar y mejorar la escalabilidad del sistema TLÖN. Los clústeres físicos y lógicos pueden abstraer conceptos del mundo real, como la Ley Orgánica de Ordenamiento Territorial, para modelar su comportamiento y establecer las características que deben tener para articularse con los otros módulos social-inspirados previamente desarrollados para el sistema TLÖN.
- Abordar el problema de agrupamiento de entidades en el sistema TLÖN ofrece ventajas en dos dimensiones. Por una parte, los clústeres lógicos permiten crear relaciones asociativas entre agentes y este es el primer paso para la definición de las comunidades de agentes de la capa 3. Por otra parte, los clústeres físicos son una forma de ordenamiento territorial que permite establecer límites de operación en un sistema distribuido como el sistema TLÖN y hacen posible el establecimiento del entorno para las posteriores aplicaciones que se desarrollen en el grupo de investigación.
- Es posible implementar el agrupamiento de nodos mediante clústeres físicos para extender el rango de operación una red ad hoc en el contexto del sistema TLÖN. Considerando que una red con el estándar 802.11 puede manejar un número aproximado de 30 nodos (Gast, 2005) en un mismo canal de operación, sin que exista un alto grado de interferencia y garantizando un rendimiento aceptable, dicho número puede ser incrementado, mediante la implementación de clustering, ya que conjuntos de clústeres que usan canales diferentes dentro de la banda de trabajo se pueden enlazar mediante equipos *gateway*. En consecuencia, el número de nodos interconectados puede ser mayor, permitiendo el crecimiento e la red.

- Al contrario de lo que se pensaba antes de desarrollar el proyecto, en un clúster físico conformado sobre un territorio ad hoc, no solo un nodo debe tener la capacidad de tomar el rol de Cluster Head, sino que todos los nodos participantes deben tener la capacidad de tener dicho rol. En una red ad hoc altamente dinámica y descentralizada, donde todos los nodos tienen la capacidad de ser enrutadores y estaciones cliente, no es apropiado que se concentre todo el tráfico del clúster en un solo nodo. Esto en parte va en contra del paradigma de las redes ad hoc y se orienta más a las arquitecturas de redes cableadas. En ese sentido, se concluye, que el rol de Cluster Head no es tan relevante para el funcionamiento de un clúster. Una alternativa más coherente con una red ad hoc es que todos los nodos con una sola interfaz inalámbrica tengan el mismo nivel de jerarquía. Ahora bien, si se trata de la conformación de varios clústeres en un territorio ad hoc y la comunicación entre los mismos, es el nodo *gateway* el que tiene el protagonismo, no porque atienda todas las peticiones de los nodos y maneje el tráfico dentro de un clúster, sino porque cuenta con más de una interfaz de red inalámbrica y permite el paso del tráfico para hacer posible la comunicación con otros clústeres. Entonces, se plantea que la jerarquía en un clúster físico se debe determinar por el número de interfaces inalámbricas que tenga un nodo. De hecho, el modelo se podría extender para incluir niveles mayores de jerarquía, por ejemplo, si un nodo *gateway* cuenta con interfaces para comunicarse con otras redes diferentes a la red inalámbrica 802.11.
- Se comprobó que usando el protocolo BATMAN permite implementar redes ad hoc jerárquicas, heterogéneas y multisalto, aboliendo la tarea de diferenciación de nodos de una forma diferente, donde no es necesario un solo nodo director Cluster Head. Además, se observó que el tráfico dentro de los clústeres físicos puede aislarse estableciendo un ESSID para cada clúster y que el nodo *gateway* actúa como cualquier otro nodo ad hoc vecino participando en la red. Sin embargo, cuando es requerido el paso del tráfico inter-clúster, se pueden habilitar las rutas necesarias para dar paso a la comunicación y esta sucederá de forma transparente para los nodos participantes.
- Un clúster lógico permite la implementación de un nivel de distribución más profundo de las funciones y servicios del sistema TLÖN. Sirve para mejorar las capacidades de control sobre los recursos distribuidos y para ejecutar tareas de forma concurrente. También habilita la prestación de servicios encapsulados.
- La comunicación entre agentes y su asociación en clústeres lógicos es posible mediante la implementación de comunicación con hilos y mediante el etiquetado de dicha comunicación.
- El establecimiento de clústeres lógicos en el sistema TLÖN permite compartir los recursos de los nodos participantes del sistema y permiten un mayor aprovechamiento de las capacidades de procesamiento de cada nodo.

## 7.2. Recomendaciones y trabajo futuro

Como resultado de las actividades de investigación, surgieron nuevos planteamientos de problemas que pueden ser abordados en futuros trabajos de investigación en línea de sistemas distribuidos, ya sea sobre redes inalámbricas o en el proyecto del sistema TLÖN.

- El experimento realizado con tres clústeres físicos demostró que los nodos *gateway* con dos interfaces inalámbricas tienen buen desempeño cuando pertenecen a dos clústeres distintos. Sin embargo, sería conveniente analizar cómo es el rendimiento de un nodo *gateway* cuando pertenece a tres o más clústeres físicos al mismo tiempo.
- Para complementar el modelo social-inspirado del sistema TLÖN, se considera importante estudiar la forma de desarrollar una institución equivalente al Ministerio de Relaciones Exteriores de TLÖN, que tenga la capacidad de establecer los contratos o acuerdos entre países de TLÖN, para permitir a los agentes móviles pasar de un clúster a otro y operar como agentes foráneos.
- Los mensajes que se envían dentro de un clúster físico implementado en el contexto del sistema TLÖN no están cifrados, por lo tanto, pueden ser visualizados por intrusos o mediante un dispositivo o software de escaneo de redes. Por esta razón, es de vital importancia el desarrollo de un sistema de cifrado de mensajes para el sistema.
- Se observó que la interferencia co-canal es un factor que retrasa el proceso de conformación de un clúster porque dificulta la detección mutua y la vinculación a un mismo clúster. Por esta razón, es necesario un estudio del espectro antes de realizar los experimentos y se recomienda más investigación sobre las formas de mitigar esta interferencia en las redes ad hoc.
- Cuando grupos de nodos conforman un clúster físico, estos pueden vincularse sin restricciones. Aunque esta característica de apertura es necesaria, se requiere de un mecanismo de autenticación para proveer seguridad al sistema y evitar que nodos maliciosos participen en la red y efectúen ataques.
- Una vez establecidos los clústeres lógicos en el sistema TLÖN, se puede acceder a los recursos de forma distribuida, pero se detectó que es importante desarrollar una política de asignación equilibrada y controlada de recursos entre clústeres.
- Se comprobó que es posible la creación de clústeres de redes ad hoc inalámbricas con nodos *gateway*, usando el protocolo BATMAN y el estándar 802.11 sobre equipos RaspberryPi. Sin embargo, aún falta probar si el modelo de clústering planteado también sirve para crear clústeres operando con otros estándares, por ejemplo con dispositivos IoT, usando tecnologías como LoRa<sup>1</sup> o Narrowband IoT<sup>2</sup> (NB-IoT).

---

<sup>1</sup><https://lora-alliance.org/>

<sup>2</sup><https://www.3gpp.org/news-events/1733-niot>

- 
- Para futuros experimentos e implementaciones se recomienda usar en todos los nodos las mismas versiones del protocolo BATMAN y de sus servicios de mensajería ALFRED, así como también de la herramienta batctl, dado que en algunas ocasiones se presentan conflictos de compatibilidad, sobre todo cuando se quiere explorar herramientas avanzadas y mejoradas liberadas en las últimas versiones de BATMAN.

## A. Anexo: Protocolos de enrutamiento para redes ad hoc

| Sigla   | Nombre del protocolo   |
|---------|--|
| ABAM    | On-Demand Associativity-Based Multi-cast                           |
| ABR     | Associativity-Based Routing  |
| ACOR    | Admission Control Enabled On Demand Routing                        |
| ADMR    | Adaptive Demand-Driven Multi-cast Routing                          |
| AGC/SGC | Abiding Geo-cast/Stored Geo-cast, Time-Stable Geo-casting          |
| ALARM   | Adaptive Location Aided Routing Protocol - Mines                   |
| AMRIS   | Ad-Hoc Multi-cast Routing protocol utilizing Increasing id-numbers |
| AMRoute | Ad-Hoc Multi-cast Routing Protocol                                 |
| AODV    | Ad-Hoc On-demand Distance Vector Routing Protocol                  |
| AOMDV   | Ad-Hoc On-demand Multi-path Distance Vector Routing                |
| AQM     | Ad-Hoc QoS Multi-cast  |
| ARA     | Ant-colony based Routing Algorithm for MANETs                      |
| ARAN    | Authenticated Routing for Ad-hoc Networks                          |
| ARIADNE | A Secure On-demand Routing Protocol for Ad-hoc Networks            |
| ATR     | Augmented Tree-based Routing                                       |
| AWDS    | Ad-Hoc Wireless Distribution Service                               |
| BATMAN  | Better Approach To Mobile Ad hoc Networking                        |
| BEMRP   | Bandwidth-Efficient Multi-cast Routing Protocol                    |
| BGR     | Blind Geographic Routing   |
| BVGF    | Bounded Voronoi Greedy Forwarding                                  |
| CAMP    | Core-Assisted Mesh Protocol  |
| CAR     | Connectivity-Aware Routing   |
| CBM     | Content Based Multi-cast   |
| CBRP    | Cluster Based Routing Protocol                                     |
| CEDAR   | Core Extraction Distributed Ad-Hoc Routing                         |
| CGSR    | Cluster-head Gateway Switch Routing protocol                       |
| CHAMP   | Caching and Multipath Routing                                      |
| DAER    | Distance-Aware Epidemic Routing                                    |
| DART    | Dynamic Address Routing  |
| DBF     | Distributed Bellman-Ford Routing Protocol                          |
| DCMP    | Dynamic Core Based Multi-cast Routing Protocol                     |
| DDM     | Differential Destination Multi-cast                                |
| DDR     | Distributed Dynamic Routing Algorithm                              |
| DFR     | Direction Forward Routing  |
| DREAM   | Distance Routing Effect Algorithm for Mobility                     |
| DSDV    | Dynamic Destination-Sequenced Distance Vector routing protocol     |
| DSR     | Dynamic Source Routing   |
| DSRPA   | Dynamic Source Routing Power-Aware                                 |
| DYMO    | Dynamic MANET On-demand Routing Protocol                           |
| EADSR   | Energy Aware Dynamic Source Routing Protocol                       |
| ESAODV  | Extra Secure Ad hoc On-demand Distance Vector                      |
| FGMP    | Forwarding Group Multi-cast Protocol                               |

|           |  |
|-----------|--|
| FSR       | Fisheye State Routing protocol   |
| GB        | Gafni-Bertsekas  |
| GDSTR     | Greedy Distributed Spanning Tree Routing   |
| GEDIR     | Geographic Distance Routing  |
| GeoGRID   | Geographical GRID  |
| GeoTORA   | Geographical TORA  |
| GFG/FACE  | Greedy FACE Greedy   |
| GLS       | Geographic Location Service  |
| GPSAL     | GPS Ant-Like Routing Algorithm   |
| GPSR      | Greedy Perimeter Stateless Routing   |
| GRAA      | Geographic Routing protocol for Aircraft Ad hoc Network                              |
| GRLI      | Geographic Routing without Location Information                                      |
| GSR       | Global State Routing protocol  |
| Guesswork | Guess Work   |
| HARP      | Hybrid Ad-Hoc Routing Protocol   |
| HRPLS     | Hybrid Routing Protocol for Large Scale Mobile Ad-Hoc Networks with Mobile Backbones |
| HSL       | Hazy Sighted Link State routing protocol   |
| HSR       | Hierarchical State Routing protocol  |
| IALBR     | Interference-Aware Load-Balancing Routing  |
| IARP      | Intra-zone Routing Protocol/pro-active part of ZRP                                   |
| IERP      | Inter-zone Routing Protocol/reactive part of ZRP                                     |
| ISAIAH    | Infra-Structure Aodv for Infrastructured Ad-Hoc networks                             |
| LAM       | Lightweight Adaptive Multi-cast  |
| LANMAR    | Landmark Routing Protocol for Large Scale Networks                                   |
| LAR       | Location-Aided Routing protocol  |
| LBCR      | Load-Balancing Curveball Routing   |
| LBM       | Location Based Multi-cast  |
| LBR       | Link life Based routing  |
| LBRMN     | Load-Balancing Routing for Mesh Networks   |
| LCA       | Linked Cluster Architecture  |
| LGM       | Location Guided Multicast  |
| LMR       | Lightweight Mobile Routing protocol  |
| LQSR      | Link Quality Source Routing  |
| LUNAR     | Lightweight Underlay Network Ad-Hoc Routing  |
| MAODDP    | Mobile Ad-hoc On-demand Data Delivery Protocol                                       |
| MAODV     | Multi-cast Ad-Hoc On-Demand Distance Vector routing                                  |
| MCEDAR    | Multicast Core-Extraction Distributed Ad hoc Routing                                 |
| MDR       | Multi-path on-Demand Routing   |
| MEPA      | Minimum Exposed Path to the Attack   |
| M-LANMAR  | Multicast Landmark Routing Protocol for Large Scale Networks                         |
| MMRP      | Mobile Mesh Routing Protocol   |
| MOBICAST  | Mobile Just-in-time Multicasting   |
| MOLSR     | Multi-cast Optimized Link State Routing  |
| MP-DSR    | Multi-Path Dynamic Source Routing  |
| MPRDV     | Multi-point Relay Distance Vector protocol   |
| MPR-E     | Multi-Path Routing with EAPAR  |
| MuSeQoS   | Multipath Security-aware QoS Routing   |
| MZR       | multicast routing protocol based on zone routing                                     |
| ODMRP     | On-Demand Multicast Routing Protocol   |
| OLAM      | On-demand location aware multicast protocol  |
| OLSR      | Optimized Link State Routing Protocol  |
| OMR       | On-demand Multi-path Routing   |
| OORP      | Order-One Routing Protocol   |
| PAMAS     | Power-Aware Multi-Access Protocol with Signaling                                     |
| PARO      | Power-Aware Routing Optimization Protocol  |

|        |   |
|--------|---|
| PCCB   | Power Cost Calculate Balance  |
| PLBR   | Preferred link based routing  |
| QuaSAR | QoS aware source initiated Ad-Hoc routing   |
| RAODV  | Reliable Ad hoc On-demand Distance Vector Routing Rrotocol                        |
| RDMAR  | Relative-Distance Micro-discovery Ad hoc Routing protocol                         |
| ROAM   | Routing On-demand Acyclic Multi-path  |
| SAODV  | Secure AODV   |
| SDSR   | Secure DSR  |
| SEAD   | Secure Efficient Distance Vector Routing for Mobile WirelessAd-hoc Networks       |
| SecMR  | Secure Multipath Routing  |
| SENCAS | Scalable Protocol for Unicast and Multicasting in a large Ad-hocEmergency Network |
| SiFT   | Simple Forwarding over Trajectory   |
| SLURP  | Scalable Location Update-Based Routing Protocol                                   |
| SMR    | Split Multi-path Routing  |
| SrcRR  | Source Routing for Roofnet  |
| SRP    | Secure Routing Protocol   |
| SSR    | Signal Stability Routing protocol   |
| TBRPF  | Topology Dissemination based on Reverse-Path Forwarding routing protocol          |
| TORA   | Temporally-Ordered Routing Algorithm routing protocol                             |
| WAR    | Witness Aided Routing   |
| WARM   | Wireless Ad Hoc Real-Time Multicast   |
| WRP    | Wireless Routing Protocol   |
| ZHLS   | Zone-based Hierarchical Link State routing  |
| ZRP    | Zone Routing Protocol   |

# B. Anexo: Modelamiento de sistemas distribuidos

## Proceso

Un proceso es una serie de tareas o instrucciones ejecutadas por un computador en un sistema operativo multitarea. De forma más general, un proceso es una serie de eventos o acciones. En un sistema, puede representar un comportamiento que se hace evidente durante un tiempo determinado.

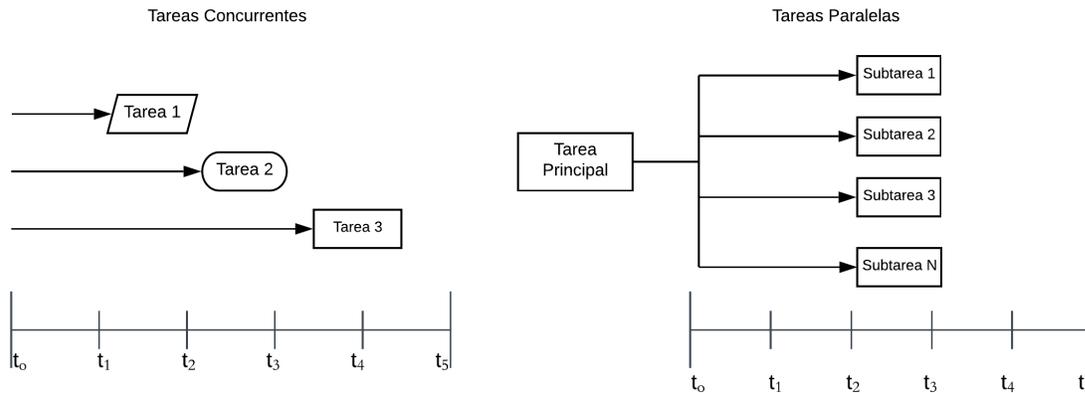
Los procesos pueden ejecutarse de forma paralela o concurrente. No debe confundirse estos términos que aunque están relacionados son diferentes.

La concurrencia, está relacionada con el hecho de existir o suceder o ser terminado al mismo tiempo, el termino esta relacionado con la expresión “correr junto” o “encontrarse”. En matemáticas se refiere a la unión de tres o mas líneas, o la tendencia de tres o más líneas de ir hacia un mismo punto (Oxford University Press, s.f.). Debe distinguirse de paralelismo, que se refiere la ocurrencia de algo al mismo tiempo o de manera similar, o a la ejecución de operaciones simultáneamente. Para entender mejor esta diferencia ver la Figura B-1.

En programación con el lenguaje Python, una aplicación puede ser concurrente pero paralela porque efectúa mas de una tarea al mismo tiempo, pero las tareas no están divididas en subtareas. Puede ser paralela pero no concurrente, porque solo hace una tarea a la vez y las tareas pueden ser divididas en subtareas que pueden procesarse en paralelo. Ni paralela ni concurrente, porque solo trabaja en una tarea en un determinado tiempo y esa tarea nunca se divide en subtareas. O puede ser ambas, paralela y concurrente cuando trabaja en múltiples tareas al mismo tiempo y las tareas están divididas en subtareas que se ejecutan al mismo tiempo.

## Clasificación de los modelos de cómputo

Los modelos de cómputo están compuestos en su mayoría por tres características comunes de los lenguajes de programación, **la sintaxis, los tipos y la semántica**, con base en estos elementos se pueden clasificar los modelos de cómputo en cuatro grupos fundamentales (Bruni y Montanari, 2017), para describir las interacciones, su estructura, los tipos y los niveles de representación necesarios para describir un modelo de cómputo, los cuatro grandes grupos son:



**Figura B-1.:** Diferencia entre tareas concurrentes y tareas paralelas

- **Modelos Imperativos (IMP):** , en estos modelos se introducen la notación  $\lambda$ , la recursividad , la equivalencia programada, la composición, la integridad y la completitud.
- **Modelos funcionales de Alto nivel (HOFL):** en estos se estudian los roles de los tipos de sistemas, los conceptos principales de este tipo de modelos provienen del dominio teórico y las distinciones de evaluación perezosa y ansiosa.
- **Modelos Concurrente no deterministas (CCS):**Estos modelos están basados en operaciones semánticas etiquetadas, se introduce la notación de bisimulación, equivalencias y congruencias observacionales. Modelos de movilidad de nombres y especificaciones de sistemas con lógica modal y temporal.
- **Modelos Probabilísticos - Estocásticos (PEPA):** son modelos donde se explotan las teorías de las cadenas de Markov, para realizar análisis cuantitativos de sistemas posiblemente concurrentes.

Estos tipos de modelos de cómputo, tienen su fundamento matemático. Permiten entender y razonar sobre el comportamiento de los programas, y a su vez permiten interpretar de forma completa las construcciones de los programas, otorgando herramientas para analizarlos y verificar con precisión su comportamiento.

En los modelos concurrentes no deterministas. Comúnmente aparece el concepto de cálculo  $\pi$  como modelo de representación para sistemas distribuidos. También el concepto de bigrafo como una forma de modelar las interacciones ese tipo de sistemas. A continuación se dará una breve descripción de estos conceptos.

## B.1. Cálculo $\Pi$

El cálculo  $\pi$  es un modelo formal para la especificación, análisis y verificación de sistemas compuestos por procesos concurrentes que se comunican (Varela, 2013). El cálculo  $\pi$  ofrece un marco conceptual

para entender la movilidad, al igual que un conjunto de herramientas matemáticas para representar sistemas móviles y razonar sobre su comportamiento. (Jensen, 2006)

El cálculo  $\pi$  pertenece a la familia del álgebra de procesos. Familia compuesta por diversos tipos como los son CCS (Calculus of Communication Systems), CPS (communicating sequential processes), ACP (Algebra of Communicating Processes), entre otros. Algunas variaciones del cálculo  $\pi$  son : *Distributed Join Calculus*, *Distributed pi-calculus*, *Ambient calculus* y *Oz* y son lenguajes influenciados por el cálculo  $\pi$ .

El cálculo  $\pi$  tiene dos aspectos, el primero de ellos es ser teoría de sistemas móviles y el segundo ser un modelo general de computación. Como teoría de sistemas móviles, ofrece una rica mezcla de técnicas para razonar sobre el comportamiento de los sistemas. Como modelo de computación toma la **interacción como primitiva**.

En el cálculo  $\lambda$  la primitiva es la **aplicación** de la función. De forma similar como el cálculo  $\lambda$  es la base de los lenguajes funcionales, el cálculo  $\pi$  y las variantes- son la base de varios lenguajes de programación concurrentes como *Pict*, *Join* y *TyCO*, su unidad funcional es el proceso. Un **proceso** es una serie de acciones o eventos, nuestro caso un **proceso** es el comportamiento de un sistema.

El cálculo  $\pi$  ofrece un marco conceptual para entender la movilidad junto con las herramientas matemáticas para representar sistemas móviles y razonar sobre su comportamiento. Surgen preguntas como: ¿Qué es movilidad? Cuando se habla de *sistemas móviles*, ¿Cuáles son las entidades que se mueven dichas entidades? ¿En qué espacio se mueven. Una respuesta a estas preguntas se puede dar considerando dos tipos de movilidad.

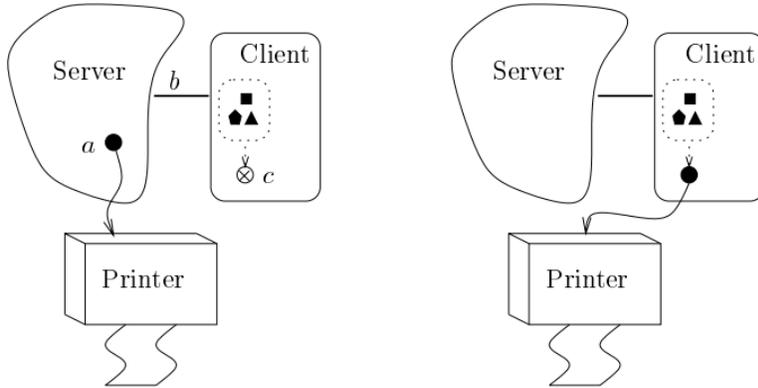
Existen dos tipos de movilidad:

- de **Enlaces** que se mueven en un espacio abstracto de procesos enlazados, y
- de **Procesos** que se mueven en un espacio abstracto de nombres enlazados.

El cálculo  $\pi$  trata la primera clase de movilidad: expresa de forma directa el movimiento de enlaces en un espacio de procesos enlazados, en este sentido hay dos clases de entidades básicas en el cálculo  $\pi$ : nombres y procesos. La computación concurrente es modelada en el cálculo  $\pi$  como procesos que se comunican sobre canales compartidos, para dar un ejemplo sencillo con este tipo de representación se tienen en cuenta las siguientes consideraciones:

- a. Las variables de Procesos son representadas con letras Mayúsculas  $P, Q, R$ .
- b. Los canales y nombres son denotados con letras minúsculas  $a, b, c$ .
- c. Las interacciones sobre los canales (Lectura y Escritura), se representan como el prefijo  $\alpha$

Para tener una idea más precisa de la sintaxis y modelado de procesos usando Cálculo  $\pi$ . En la figura B-2 se muestra un ejemplo sencillo de un modelo cliente servidor, donde se representa el flujo para imprimir el contenido de un archivo. (Milner, 1999), por otra parte se puede observar su representación sobre cálculo  $\pi$  en la figura B-3.



**Figura B-2.:** Servidor de impresión (Izq. antes de la interacción, Der. Después de la interacción)

$$(\bar{b}a.S|b(c).\bar{c}d.P)\tau \rightarrow (S|\bar{a}d.P)$$

Los siguientes son los elementos que componen el ejemplo y su respectiva notación en cálculo  $\pi$ .

- $P$  es el proceso cliente.
- $S$  es el proceso Servidor .
- $\tau$  es el operador de transformación .
- $a$  es el acceso a la impresora .
- $b$  es el acceso al servidor.
- $c$  es un marcador de posición.
- $d$  es el acceso a algún dato.
- $(\bar{b}a.S|b(c).\bar{c}d.P)\tau \rightarrow (S|\bar{a}d.P)$  Representación de las interacciones de los procesos cliente y servidor .
- $(\nu.a)(\bar{b}a.S|R)|b(c).\bar{c}d.P)\tau \rightarrow (\nu.a)(S|R|\bar{a}d.P)$  Representación de las interacciones denotando los puertos  $\nu$ .

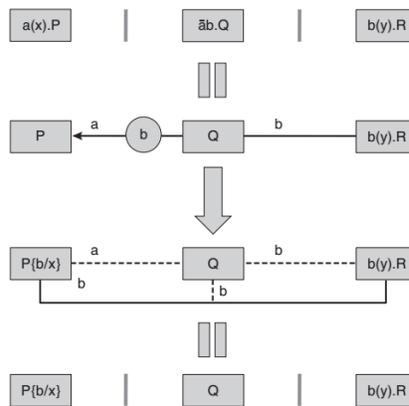


Figura B-3.: Modelo cliente servidor Cálculo II (Varela, 2013)

### B.1.1. Semántica operacional

El elemento básico de operación e cálculo  $\pi$  es el *nombre*  $\mathcal{N}$  denotado como un conjunto infinito con un rango  $a, b, c, \dots, z$  elementos los cuales podrían funcionar como puertos de comunicación, variables y datos de las variables, un conjunto de (agentes) identificadores sobre  $\mathcal{A}$ . Los elementos mas relevantes para realizar operaciones son:

- El agente vacío  $\mathbf{0}$  el cual no desempeña ninguna acción.
- El *prefijo de Salida*  $\bar{a}x.P$ . el nombre  $x$  es enviado a lo largo del nombre  $a$ , y a partir de entonces el agente continua como  $P$ .
- El *prefijo de Entrada*  $a(x).P$ . indica que un nombre fue recibido a lo largo de  $a$  y  $x$  es un lugar para recibirlo, Después el agente de entrada continua como  $P$ , pero con el nombre recién obtenido reemplazando a  $x$ .
- El *prefijo de Silencio- no interacción*  $\tau.P$ . el cual representa a un agente que puede evolucionar a  $P$  sin interacción con el ambiente.
- Una *Suma*  $P + Q$  indica un agente que puede representar cualquier  $P$  o  $Q$
- Una *Composición Paralela*  $P|Q$  representa el comportamiento combinado de  $P$  y  $Q$  ejecutandose en paralelo.
- Un *Match* if  $x = y$  then  $P$ . se espera que este agente tenga el mismo comportamiento de  $P$  si  $x$  y  $y$  tienen el mismo nombre, de lo contrario no hace nada.
- Un *Mismatch* if  $x \neq y$  then  $P$ . este agente podría comportarse como  $P$  si  $x$  y  $y$  no tienen el mismo nombre, de lo contrario no hace nada.

- Una *Restricción*  $(\nu x)P$ . este agente se comporta como  $P$  pero este agente es local, significa que este inmediatamente no puede ser usado como un puerto para comunicarse entre  $P$  y su ambiente.
- Un *Identificador*  $A(y_1, \dots, y_n)$  donde  $n$  es la aridad de  $A$ . Cada identificador tiene una *Definición*  $A(x_1, \dots, x_n) \text{ def } = P$  donde  $x_i$  puede ser una pareja diferente y la intuición es que  $A(y_1, \dots, y_n)$  se comporta como  $P$  con  $y_i$  reemplazando a  $x_i$  por cada  $i$ . Puede verse como la declaración de un proceso  $x_1, \dots, x_n$  como parámetros formales y el identificador  $A(x_1, \dots, x_n)$  como una invocación con los parámetros actuales de  $y_1, \dots, y_n$ .

Aunque la representación del cálculo  $\pi$  permite validar la concurrencia dentro de un sistema y modelar las interacciones entre procesos. La movilidad de los mismos presenta restricciones para modelar interacciones y cambios de estados en sistemas ubicuos donde se requiere de una representación espacial y lógica, por esa razón la ruta de estudio continua con el modelo de Bigrafos B.2, dado que es una herramienta que permite modelar el mecanismo propuesto en este trabajo.

## B.2. Bigrafos

Los sistemas Bigraficos Reactivos (BRSs por sus siglas en inglés) son modelos de computación en los cuales la *ubicación* y la *conectividad* son esenciales (Milner, 2006). En el contexto de IoT e incluso de Internet la ubicación geográfica es un desafío para el control de sistemas distribuidos. Los bigrafos son desarrollados a partir del cálculo de acciones, e involucra ideas de distintas fuentes como la Máquina Abstracta Química (Cham), de Berry y Boudol, el cálculo  $\pi$  de Milner, Parrow y Walker, presentado en la sección B.1, las redes de interacciones, los ambientes móviles, las fusiones explícitas de Gardner y Wischik, el Nomadic Pict de Wojciechowski y Sewell y la teoría de sistemas reactivos de Leifer y Milner, al igual que una contribución de las redes de Petri.

Este modelo ha sido desarrollados con dos objetivos claros *i)* ser capaz de modelar directamente aspectos de sistemas ubicuos enfocados en la conectividad móvil (grafo de enlace) y la movilidad local (grafo de lugar) simultáneamente y *ii)* proveer una unificación de las teorías existentes para el desarrollar una teoría general en la cual varios modelos de cálculos de sistemas concurrentes y móviles puedan ser representados.

### Bigrafo:

En esencia un bigrafo consiste de un **un grafo de lugar**; un bosque cuyos nodos representan una diversidad de artefactos computacionales, y un nodo **un grafo de enlace**, el cual es un hypergrafo que conecta los puertos de los nodos. Los bigrafos pueden ser reconfigurados por el significado de las **reglas de reacción**. El bigrafo  $\check{G}$  tiene nodos  $V = \{v_0 + \dots + v_5\}$  y vértices  $E = \{e_0, e_1, e_2\}$

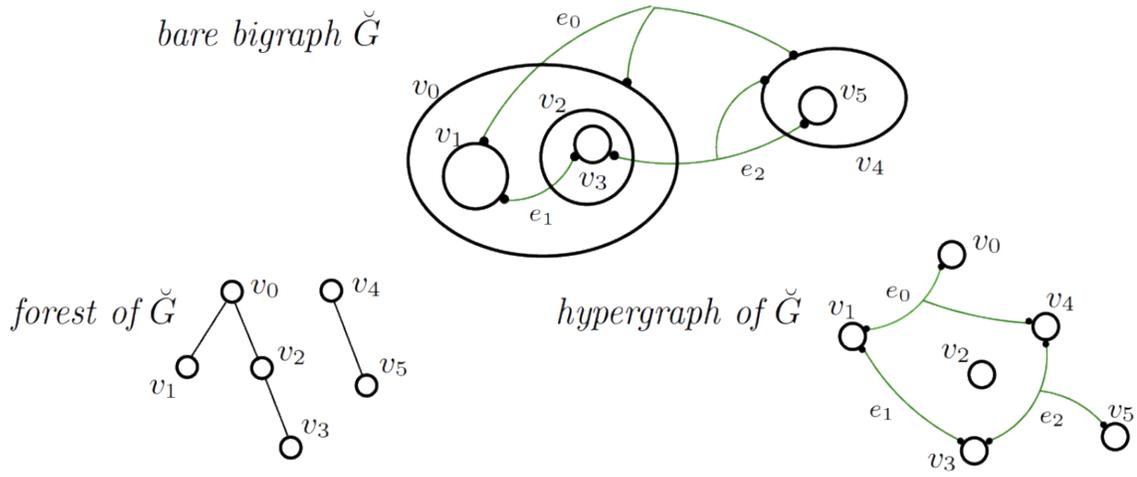


Figura B-4.: Bigrafo adaptado de (Milner, 2009)

**Definición:** Una regla de reacción consiste de un **redex** (el patrón a ser cambiado) y un **reactum** (el patrón de cambio), los cuales reconfiguran el bigrafo y describen las interacciones de conectividad y lugar de los nodos. Los sistemas ubicuos son modelados en espacios discretos compartidos involucrando tres conceptos: agente, ubicación y conectividad, cuando se reconfigura el espacio hay dos conceptos adicionales: el movimiento y la interacción, del mismo modo el movimiento es representado por reglas de reacción las cuales modifican el espacio y la conectividad, es decir hay un cambio de estados en el bigrafo.

#### Componentes del Bigrafo:

Un bigrafo con nodos  $V$  y vértices  $E$  tiene un bosque cuyos nodos son  $V$ ; este a su vez tiene un hipergrafo con nodos  $V$  y vértices  $E$ .

### B.2.1. Definiciones

Un bigrafo es un par  $B = \langle B^P, B^L \rangle$  un grafo de lugar y un grafo de enlace. Estos son sus constituyentes, su cara externa está denotada por el par  $\langle n, Y \rangle$  donde  $n$  y  $Y$  son la cara externa de  $B^P$  y  $B^L$  respectivamente, del mismo modo para la cara interna  $\langle m, X \rangle$ .

Cada Bigrafo  $G$  tiene dos interfaces  $I$  y  $J$ , definidas como  $G : I \rightarrow J$ ,  $I$  es la interface interna y  $J$  la interface externa. Una interfaz es una tripla  $\langle m, \vec{X}, X \rangle$ , donde  $m$  es la amplitud (número de sitios o raíces),  $X$  el conjunto completo de nombres locales y globales, y  $\vec{X}$  indican las ubicaciones de cada nombre local.

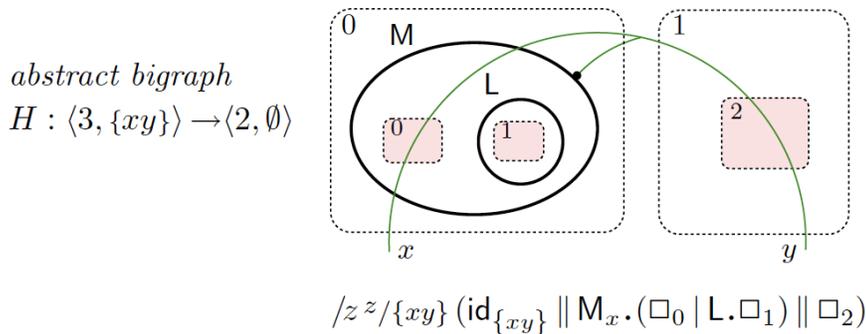


Figura B-5.: Bigrafo Abstracto (Milner, 2009)

Un bigrafo abstracto  $H$  como el mostrado en la figura B-5, es aquel que no tiene denotados los nombres de los nodos, ni de los enlaces. Los nodos de un bigrafo puede ser de diferentes tipos; esto refleja que pueden contribuir de forma diferenciada a las dinámicas. Cada bigrafo tiene asignado un tipo llamado control  $K$  el cual determina la aridad libre ligada por  $v : K$ , este control indica la forma de acceder al nodo, aplicación u otro elemento del bigrafo. Se especifican mediante una *firma* como la de la ecuación B-1.

$$K = \{K : 2, L : 0, M : 1\} \tag{B-1}$$

La firma del bigrafo  $G$  de la Figura B-6 esta dada en la ecuación B-1, la cual representa la aridad de cada uno de los nodos.

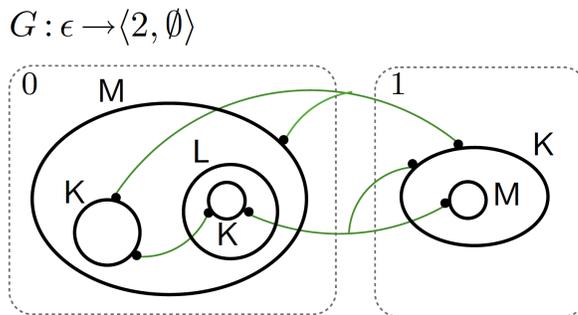


Figura B-6.: Bigrafo con controles (Milner, 2009)

**Construcción** Se pueden hacer bigrafos más grandes a partir de otros más pequeños a través de sus interfaces; esta construcción es definida en términos de los grafos constituyentes de lugar y enlace

**Definición B.2.1** (bigrafo concreto puro). Un bigrafo puro (concreto) sobre una firma  $\mathcal{K}$ , toma la forma  $G = (V, E, ctrl, G^P, G^L) : I \rightarrow J$  donde  $I = \langle m, X \rangle$  y  $J = \langle n, Y \rangle$  son sus caras internas y externas, cada una combinando una amplitud con un conjunto finito de nombres globales plasmados sobre  $\mathcal{X}$ , Esto dos primeros componentes  $V$  y  $E$  son conjuntos de nodos finitos y vértices respectivamente. El tercer componente

$ctrl : V \rightarrow \mathcal{K}$ , un mapa de control, asignado a cada nodo. Los restantes dos son:

$G_P = (V, ctrl, prnt) : m \rightarrow n$  un grafo de lugar

$G_P = (V, E, ctrl, link) : X \rightarrow Y$  un grafo de lugar

**Definición B.2.2** (grafo de lugar). Un grafo de lugar  $A = (V, ctrl, prnt) : m \rightarrow n$  tiene una amplitud interna  $n$  y una externa  $m$ , ambos ordinales finitos, un conjunto de nodos  $V$  con un mapa de control  $ctrl : V \rightarrow \mathcal{K}$ ; y un mapa principal  $prnt : m \uplus V \rightarrow V \uplus n$ . El mapa principal es aciclico, que es  $prnt^k(v) \neq v$  para todo  $k > 0$  y  $v \in V$ .

**Definición B.2.3** (grafo de enlace). Un grafo de enlace  $A = (V, E, ctrl, link) : X \rightarrow Y$  tiene un conjunto  $X$  de nombres internos,  $Y$  de nombres externos,  $V$  de nodos y  $E$  de enlaces. También tiene una función  $ctrl : V \rightarrow \mathcal{K}$  llamada mapa de control, y una función  $link : X \uplus P \rightarrow E \uplus Y$  llamado mapa de enlace. donde  $P = \sum_{v \in V} ar(ctrl(v))$  es el conjunto de puertos de  $A$ .

## B.2.2. Operaciones Básicas

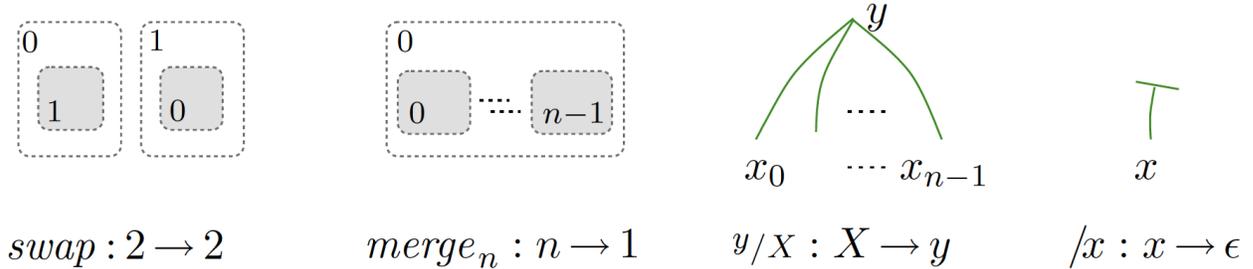
Es de resaltar que todos los bigrafos pueden ser contruidos desde otros elementales a partir de la composición ( $\circ$ ) y el producto tensor  $\otimes$ , existen otras dos operaciones derivadas que son el *producto paralelo*  $G_1 || G_2$  y el *producto principal*  $G_1 | G_2$ , estos productos son definidos en las interfaces del bigrafo. (La función identidad es usada en esta operaciones es descrita como  $Id_s$ ).

Para los bigrafos  $G_1$  y  $G_2$  que no comparten nombres o nombre internos, es posible realizar un *producto tensor*  $G_1 \otimes G_2$  por la yuxtaposición de sus grafos de lugar, construyendo la unión de sus grafos de enlace, e incrementando los índices de los sitios en  $G_2$  por el número de  $G_1$ .

En cambio el *producto paralelo*  $G_1 || G_2$  es como el producto tensor, excepto que los nombres globales pueden ser compartidos: if  $y$  es compartido, todos los puntos de  $y$  en  $G_1$  y  $G_2$  se convierten en puntos de  $y$  en  $G_1 || G_2$ .

Se pueden *componer* bigrafos  $G_2 : I \rightarrow I'$  y  $G_1 : I' \rightarrow J$ , manteniendo el bigrafo  $G_1 \circ G_2 : I \rightarrow J$  al conectar los sitios de  $G_1$  con las raíces de  $G_2$ , eliminando ambos y conectando nombres de  $G_2$  con nombre internos de  $G_1$ .

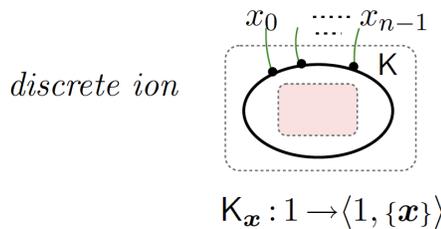
*elementary placings and linkings*



**Figura B-7.:** Bigrafos Elementales adaptado de (Milner, 2009)

Continuando con este acercamiento a la teoría de bigrafos se enunciarán los bigrafos elementales, usados para la construcción de reglas de interacción y las operaciones entre bigrafos, básicamente existen tres tipos de bigrafos elementales, las primeras dos clases son libres de nodos. Si un bigrafo libre de nodos tampoco tiene enlaces se llama *colocación*; si no tiene lugares se llama un enlace. Hay lugares y enlaces elementales, desde los cuales todos los otros pueden ser formados usando composición y producto, estos se ven en la figura B-7, *swap* es la operación de intercambio entre regiones, *merge* es la unión de los lugares, estos dos bigrafos se forman a partir de los grafos de lugar, por parte de los grafos de enlace están el bigrafo de *substitución* donde es reemplazado y escrito el nombre en el enlace e interfaz indicados, la *closure* es el cierre de un  $x$ -enlace.

La tercera clase de bigrafo elemental es el *ion discreto* que es un simple nodo de control  $K$  de aridad  $n$  con sus puertos enlazados en  $n$  diferentes nombres de salida  $x$  y tiene simplemente un sitio, como se ve en la figura B-8.



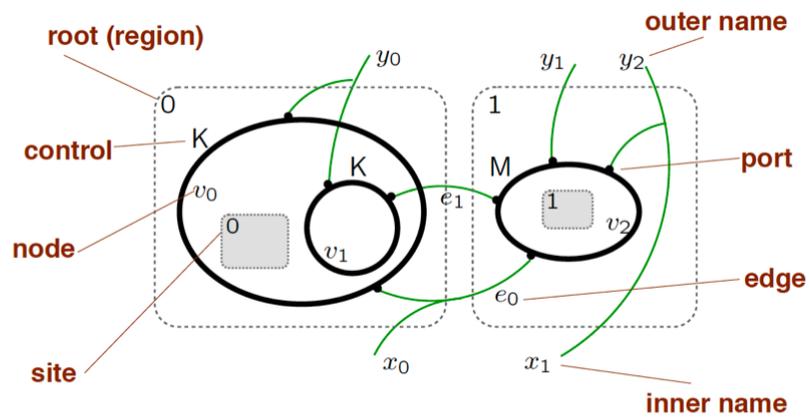
**Figura B-8.:** Ion Discreto (Milner, 2009)

Varios sistemas han sido modelados con bigrafos como el modelo de comunicación IEEE 802.11 (Calder y Sevegnani, 2014), sistemas multiagente (Mansutti, Miculan, y Peressotti, 2014), sistemas de archivos (Birkedal, Damgaard, Glenstrup, y Milner, 2007) y con los trabajos realizados por el grupo de investigación en Lenguajes de Programación Distribuidos y Redes de Telecomunicaciones Dinámicas, se busca modelar el sistema distribuido TLÖN con este tipo de formalismos.

Para terminar esta breve introducción a la teoría de bigrafos la figura B-9 resume y representa los componentes de un bigrafo, estos elementos permiten identificar las reacciones, y en general las

abstracciones físicas y lógicas posibles, con el fin de definir las reglas de reacción e identificar las interacciones entre nodos, lugares, raíces, enlaces, nombres internos, nombre externos e interfaces.

### The anatomy of bigraphs



**place** = **root** or **node** or **site**

**link** = **edge** or **outer name**

**point** = **port** or **inner name**

**Figura B-9.:** Anatomía de un bigrafo (Milner, 2009)

## Referencias

- Akhtar, A. M., y Wang, X. (2016). Cross-Layer Designs for Energy-Efficient Wireless Ad-hoc Networks. En M. Z. Shakir, M. A. Imran, K. A. Qaraqe, M.-S. Alouini, y A. V. Vasilakos (Eds.), *Energy management in wireless cellular and ad-hoc networks* (pp. 147–168). Cham: Springer International Publishing. Descargado de [https://doi.org/10.1007/978-3-319-27568-0\\_{\\_}7](https://doi.org/10.1007/978-3-319-27568-0_{_}7) doi: 10.1007/978-3-319-27568-0\_7
- Alotaibi, E., y Mukherjee, B. (2012, feb). A survey on routing algorithms for wireless Ad-Hoc and mesh networks. *Computer Networks*, 56(2), 940–965. Descargado de <http://linkinghub.elsevier.com/retrieve/pii/S138912861100377X><https://linkinghub.elsevier.com/retrieve/pii/S138912861100377X> doi: 10.1016/j.comnet.2011.10.011
- Álvarez Rodríguez, Germán, D. (2017). *Construcción un sub-sistema de movilidad de agentes artificiales de software que permita generar desempeñan sobre el Sistema TLÖN acuerdo con el rol que ellos cuyos recursos se comparten a través de una red Ad hoc* (MSc Thesis, Universidad Nacional de Colombia). doi: 63595
- Antonopoulos, N., y Gillam, L. (Eds.). (2017). *Cloud Computing* (Second Edi ed., Vol. 15). Cham: Springer International Publishing. Descargado de <http://link.springer.com/10.1007/978-3-319-54645-2> doi: 10.1007/978-3-319-54645-2
- Banco de la República. (2017). *Territorio - Enciclopedia*. Descargado de <http://enciclopedia.banrepcultural.org>
- Barbeau, M., y Kranakis, E. (2007). *Principles of Ad Hoc Networking*. Chichester, UK: John Wiley & Sons, Ltd. doi: 10.1002/9780470512494
- Basagni, S., Conti, M., Giordano, S., y Stojmenović, I. (2013). *Mobile Ad Hoc Networking* (S. Basagni, M. Conti, S. Giordano, y I. Stojmenovic, Eds.). Hoboken, NJ, USA: John Wiley & Sons, Inc. Descargado de <http://doi.wiley.com/10.1002/9781118511305> doi: 10.1002/9781118511305
- Belding-Royer, E. M., Conti, M., Giordano, S., y Stojmenovic, I. (2005). Routing Approaches in Mobile AD HOC Networks. En *Mobile ad hoc networking* (pp. 275–300). Hoboken, NJ, USA: John Wiley & Sons, Inc. Descargado de <http://doi.wiley.com/10.1002/0471656895.ch10> doi: 10.1002/0471656895.ch10
- Birkedal, L., Damgaard, T. C., Glenstrup, A. J., y Milner, R. (2007). Matching of bigraphs. *Electronic Notes in Theoretical Computer Science*, 175(4), 3–19.
- Bruni, R., y Montanari, U. (2017). *Models of computation*. Springer.
- Calder, M., y Sevegnani, M. (2014). Modelling IEEE 802.11 CSMA/CA RTS/CTS with stochastic bigraphs with sharing. *Formal Aspects of Computing*, 26(3), 537–561. doi: 10.1007/s00165-012-0270-3
- Dolui, K., y Datta, S. K. (2017). Comparison of edge computing implementations: Fog computing, cloudlet and mobile edge computing. *GloTS 2017 - Global Internet of Things Summit, Proceedings*. doi: 10.1109/GIOTS.2017.8016213

- Douglas, D. A., Covington, M. A., Covington, M. M., y Covington, C. A. (2009). *Dictionary of Computer and Internet Terms, Tenth Edition*.
- Elhdhili, M. E., Ben Azzouz, L., y Kamoun, F. (2008). CASAN: Clustering algorithm for security in ad hoc networks. *Computer Communications*, 31(13), 2972–2980. doi: 10.1016/j.comcom.2008.04.001
- Fitzek, F. H. P. (2014). *Mobile Clouds*. John Wiley and Sons Ltd.
- Gast, M. (2005). 802.11 Wireless Networks: The Definitive Guide (n.º April). Descargado de <http://books.google.com/books?id=9rHnRzzMHLIC{&}pgis=1>
- Gavhale, M., y Saraf, P. D. (2016). Survey on Algorithms for Efficient Cluster Formation and Cluster Head Selection in MANET. *Physics Procedia*, 78(December 2015), 477–482. Descargado de <http://dx.doi.org/10.1016/j.procs.2016.02.091> doi: 10.1016/j.procs.2016.02.091
- González, J. E., Ortiz, J. E., y Zárate Ceballos, H. (2016). Towards the design of a social-inspired module for the decision making of nodes in an ad hoc network. *Sistemas y Telemática*, 14(36), 9–25. doi: 10.18046/syt.v14i36.2212
- Gulati, M. K., y Kumar, K. (2014). Performance Comparison of Mobile Ad Hoc Network Routing Protocols. *International journal of Computer Networks & Communications*, 6(2), 127–142. doi: 10.5121/ijcnc.2014.6211
- Hobbes, T. (2001). *Leviatan*. Fondo de Cultura Económica.
- Hu, P., Dhelim, S., Ning, H., y Qiu, T. (2017). Survey on fog computing: architecture, key technologies, applications and open issues. *Journal of Network and Computer Applications*, 98, 27–42. Descargado de <http://dx.doi.org/10.1016/j.jnca.2017.09.002> doi: 10.1016/j.jnca.2017.09.002
- IEEE Computer Society. (2012). *IEEE Std 802.11-2016 Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications* (Vol. 2016).
- Ilyas, M. (2003). *The Handbook of Ad Hoc Wireless Networks*. Boca Ratón, Florida: CRC Press.
- ITU-T Study Group 20. (2012). Recommendation ITU-T Y.206. Descargado de <https://www.itu.int/rec/T-REC-Y.2060-201206-I>
- Jensen, O. H. (2006). Mobile Processes in Bigraphs Ole Høgh Jensen October 2006. (October).
- Jiankang, Y., Yingying, C., y Shifeng, S. (2016). Research of Clustering and Group Leader Election Algorithm on Hierarchy Ad Hoc Network Academy of Armored Forces Engineering. (3), 15–17.
- Jin, H., y Buyya, R. (2001). Cluster Computing Tools, Applications, and Australian Initiatives for Low Cost Supercomputing. *of High Performance Computing*. Descargado de <http://ww2.cs.mu.oz.au/{~}raj/papers/IEAustMonitor.pdf>
- Kulla, E., Hiyama, M., Ikeda, M., y Barolli, L. (2012). Performance comparison of OLSR and BATMAN routing protocols by a MANET testbed in stairs environment. *Computers and Mathematics with Applications*, 63(2), 339–349. Descargado de <http://dx.doi.org/10.1016/j.camwa.2011.07.035> doi: 10.1016/j.camwa.2011.07.035
- Kumar, R., Dave, M., y Kurukshetra, N. I. T. (2010). Mobile Agent as an Approach to Improve QoS in Vehicular Ad Hoc Network. *Mobile Ad Hoc Networks*, 67–72.

- Latorre, J. F., Sanchez Cifuentes, J. F., y Zárate Ceballos, H. (2015). Model in an ad-hoc network for data acquisition in sensor networks. En *Colcom*. Popayan.
- Loo, J., Mauri, J. L., Ortiz, J. H., y Maltz, D. A. (2016). *Mobile Ad Hoc Networks: Current Status and Future Trends* (Vol. 2; J. Loo, L. M. Jaime, y J. Ortiz, Eds.) (n.º May). AUERBACH PUBLICATIONS. Descargado de <https://books.google.com/books?id=Z1p8A1681tEC&pgis=1>
- Mansutti, A., Miculan, M., y Peressotti, M. (2014). Multi-agent systems design and prototyping with bigraphical reactive systems. En *Ifip international conference on distributed applications and interoperable systems* (pp. 201–208).
- Medhi, D., y Ramasamy, K. (2018). Multicast Routing. En *Network routing* (pp. 260–285). Elsevier. Descargado de <https://linkinghub.elsevier.com/retrieve/pii/B9780128007372000107> doi: 10.1016/B978-0-12-800737-2.00010-7
- Millán, J. (2003). *Nodo en Vocabulario de ordenadores e Internet*. Descargado 2019-06-06, de [http://jamillan.com/v{}\\_nodo.htm](http://jamillan.com/v{}_nodo.htm)
- Milner, R. (1999). *Communicating and mobile systems: the pi calculus*. Cambridge university press.
- Milner, R. (2006). Pure bigraphs: Structure and dynamics. *Information and Computation*, 204(1), 60–122. doi: 10.1016/j.ic.2005.07.003
- Milner, R. (2009). *The Space and Motion of Communication Agents*. Cambridge University Press.
- Ministerio del Interior y de Justicia. (2011). LOOT. [Ley 1454 de 2011]. Bogotá, D. C..
- Mohapatra, P., y Krishnamurty, S. V. (2005). *AD HOC NETWORKS Technologies and Protocols*. Boston, MA: Springer US.
- Naha, R. K., Garg, S., Georgakopoulos, D., Jayaraman, P. P., Gao, L., Xiang, Y., y Ranjan, R. (2018, jul). Fog Computing: Survey of Trends, Architectures, Requirements, and Research Directions. Descargado de <http://arxiv.org/abs/1807.00976>
- Neethu, V. V., y Singh, A. K. (2015). Mobility Aware Loose Clustering for Mobile Ad hoc Network. *Procedia Computer Science*, 54, 57–64. Descargado de <http://dx.doi.org/10.1016/j.procs.2015.06.007> doi: 10.1016/j.procs.2015.06.007
- Nykamp, D. (s.f.). *Node definition*. Descargado 2019-06-06, de [https://mathinsight.org/definition/network{}\\_node](https://mathinsight.org/definition/network{}_node)
- Openfog Consortium. (2017). OpenFog Reference Architecture for Fog Computing. *Reference Architecture*(February), 1–162. doi: 10.1049/iet-net.2015.0034
- Oubbati, O. S., Lakas, A., Zhou, F., GüneÅ, M., y Yagoubi, M. B. (2017). A survey on position-based routing protocols for Flying Ad hoc Networks (FANETs). *Vehicular Communications*, 10, 29–56. doi: 10.1016/j.vehcom.2017.10.003
- Oxford University Press. (s.f.). *concurrent | Definition of concurrent in English by Oxford Dictionaries*. Descargado 2019-06-07, de <https://en.oxforddictionaries.com/definition/concurrent>
- Papakostas, D., y Katsaros, D. (2014). A simulation-based performance evaluation of a randomized MIS-based clustering algorithm for ad hoc networks. *Simulation Modelling Practice and Theory*, 48, 1–23. Descargado de <http://dx.doi.org/10.1016/j.simpat.2014.06.012> doi: 10.1016/j.simpat.2014.06.012

- Parra Amaris, J. E., Checa Hurtado, A. C., y Ortíz Triviño, J. E. (2015). Bacteria Agent Colony inside an Ad-Hoc Network Comunidad de Agentes Bacterianos en una Red. , 347–350.
- Patel, N. (2003). Cluster analysis. MIT Open courseware Sloan School of Management Lecture 11 Data Mining. Descargado de <https://ocw.mit.edu/courses/sloan-school-of-management/15-062-data-mining-spring-2003/lecture-notes/lec11.pdf>
- Portnoy, M. (2012). *Virtualization Essentials* (Vol. 6) (n.º 2). Sybex.
- RAE. (s.f.). *ad hoc*. Descargado 2018-05-17, de <http://lema.rae.es/dpd/?key=ad+hoc>
- Rashvand, H. F., y Chao, H.-c. (2013). *Dynamic Ad Hoc Networks*.
- Raw, R. S., Lobiyal, D. K., Das, S., y Kumar, S. (2015). Analytical Evaluation of Directional-Location Aided Routing Protocol for VANETs. *Wireless Personal Communications*, 82(3), 1877–1891. Descargado de <http://dx.doi.org/10.1007/s11277-015-2320-7> doi: 10.1007/s11277-015-2320-7
- Rodríguez Valbuena, D. (2010). Territorio y territorialidad Nueva categoría de análisis y desarrollo didáctico de la Geografía. *Uni-pluri/versidad*, 10(3).
- Salvá-Garau, F., y Stojanovic, M. (2003). Multi-cluster protocol for Ad Hoc mobile underwater acoustic networks. *Oceans 2003: Celebrating the Past... Teaming Toward the Future*, 1, 91–98. doi: 10.1109/OCEANS.2003.178527
- Serpanos, D., y Wolf, M. (s.f.). *Internet-of-Things (IoT) Systems*. Springer.
- Tiwana, B. S., y Gupta, A. (2009). A distributed algorithm for traffic aware clustering in ad hoc networks. En *2009 IEEE Asia-Pacific Services Computing Conference, APSCC 2009*. doi: 10.1109/APSCC.2009.5394138
- TLÖN - Grupo de Investigación en Redes de Telecomunicaciones Dinámicas y Lenguajes de Programación Distribuidos. (s.f.). *Página web*. Descargado 2019-05-20, de <http://www.tlon.unal.edu.co/proyecto-tlon/acerca>
- Vaquero, L. M., y Rodero-merino, L. (2020). Finding your Way in the Fog : Towards a Comprehensive Definition of Fog Computing. , 44(5), 27–32.
- Varela, C. (2013). *Programming Distributed Computing Systems*.
- Wei, D., y Anthony Chan, H. (2007). Clustering ad hoc networks: Schemes and classifications. *2006 3rd Annual IEEE Communications Society on Sensor and Adhoc Communications and Networks, Secon 2006*, 3(C), 920–926. doi: 10.1109/SAHCN.2006.288583
- Xu, R., y Wunsch, D. C. (2009). Cluster Analysis. En *Clustering* (pp. 1–13). Hoboken, NJ, USA: John Wiley & Sons, Inc. Descargado de <http://doi.wiley.com/10.1002/9780470382776.ch1> doi: 10.1002/9780470382776.ch1
- Zárate, H. (2016). *Diseño de un Sub-Sistema de Cómputo Distribuido que permita implementar virtualización inalámbrica para gestionar recursos (Procesamiento, memoria, almacenamiento y dispositivos E/S) distribuidos en una Red Ad Hoc, mediante el modelo de pseudo Estado* (Tesis Doctoral no publicada). Universidad Nacional de Colombia - Sede Bogotá.
- Zwass, V. (s.f.). *Agent | computer science*. Descargado 2019-06-06, de <https://www.britannica.com/technology/agent>