



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Universidad Nacional de Colombia

Ingeniería de Sistemas y Computación

2025969 Modelos estocásticos y simulación en computación y comunicaciones (01)

Taller 1

Integrantes:

Javier Andrés Tarazona Jiménez	jtarazonaj@unal.edu.co
Yenifer Yulieth Mora Segura	ymoras@unal.edu.co
Juan Esteban Carranza Salazar	jcarranza@unal.edu.co
Grevy Joner Rincon Mejia	grrinconm@unal.edu.co
Jefferson Duvan Ramirez Castañeda	jeramirezca@unal.edu.co
Javier Andres Carrillo Carrasco	jacarrillo@unal.edu.co
Diego Nicolas Ramirez Maldonado	dieramirezma@unal.edu.co

Junio 16 de 2025

Índice

1. Introducción	3
2. Marco Teórico	3
2.1. Redes MANET	3
2.1.1. Características de las redes MANET	3
2.1.2. Aplicaciones de las redes MANET	4
2.2. Jerarquía de Redes MANET	4
2.2.1. Arquitectura jerárquica en MANET	4
2.2.2. Beneficios de la jerarquía en MANET	5
2.2.3. Principales problemas en MANETs jerárquicas de 2 niveles	5
2.2.4. Comunicación intra-clúster e inter-clúster	6
2.3. Protocolo de enrutamiento Cluster-Based	6
2.3.1. Principios del enrutamiento Cluster-Based	7
2.3.2. Liderazgo dinámico y algoritmo WCA	7
2.4. Simulación de MANET con ns-3	8
2.4.1. Características de ns-3.45 para la simulación de MANET	8
2.5. Modelos de movilidad en MANET	8
2.5.1. Modelo de movilidad Boids	9
2.6. Integración	9
2.6.1. Problemática en redes con movilidad tipo Boids para respuesta a emergencias	9
2.6.2. Propuesta de solución: Nodo CH-temporal	10
3. Descripción y Justificación del Problema a Resolver	10
3.1. Contexto del Problema	10
3.2. Problema Específico	11
3.3. Objetivo Principal	11
3.4. Hipótesis de Investigación	11
3.5. Justificación de la Solución Propuesta	11
3.6. Métricas de Evaluación	11
4. Diseño de la solución	12
4.1. Metodología	12
4.2. Estructura general del sistema	12
4.3. Reglas de comportamiento Boid	13
4.4. Espacio de simulación y condiciones de frontera	13
4.5. Actualización de posiciones	13
4.6. Visualización y exportación de datos	13
4.7. Comportamiento de líderes y seguidores	14
4.8. Configurabilidad del modelo	15
4.9. Diagrama de Flujo de la Simulación	16
4.10. Modelos Estocásticos y Distribuciones Empleadas en la Simulación de MANET para Extinción de Fuegos	17
5. Código Fuente	18
6. Manual Usuario	18
7. Manual Técnico	22
7.1. Requisitos para Ejecutar la Simulación	22
7.1.1. Configuraciones Adicionales	23
7.2. Estructura del Código	23
7.3. Implementación del Modelo Boids	23
7.3.1. Reglas Boids Implementadas	23

7.3.2. Jerarquía Líder-Seguidor	23
7.4. Ejecución de la Simulación	23
7.4.1. Compilación	23
7.4.2. Ejecución	23
7.5. Salida de Datos	24
7.5.1. Formato del Archivo CSV	24
8. Experimentación	24
8.1. Escenario 0:	25
8.2. Escenario 1:	26
8.3. Escenario 2:	28
9. Conclusiones	30
10.Recomendaciones	30
11.Referencias	31

1. Introducción

En un escenario de extinción de incendios forestales mediante drones cooperativos, la comunicación inalámbrica entre los dispositivos se ve constantemente afectada por obstáculos naturales (árboles, relieve), interferencias y la movilidad dinámica del enjambre. Este trabajo aborda un problema concreto: cuando un dron se desconecta momentáneamente del líder de su grupo (Cluster Head) debido a estas adversidades, pierde capacidad de coordinación crítica para la misión, retrasando la respuesta ante focos de incendio y aumentando el consumo energético por retransmisiones.

Para resolver este desafío operativo, proponemos una solución basada en redes MANET jerárquicas, donde los drones desconectados adoptan temporalmente el rol de líder (CH-temporal). Este enfoque práctico se sustenta en modelos teóricos de movilidad grupal y algoritmos de reclusterización dinámica, evaluando umbrales de desconexión basados en calidad de señal, energía remanente y presencia de obstáculos. La implementación en NS-3 demuestra que este método reduce el tiempo de recuperación de conexiones comparado con protocolos tradicionales, manteniendo la efectividad del enjambre incluso en condiciones adversas.

2. Marco Teórico

2.1. Redes MANET

Las redes móviles ad hoc, conocidas como **MANET (Mobile Ad Hoc Networks)**, son un tipo de red inalámbrica **descentralizada** y **auto-configurable** formada por dispositivos móviles que se comunican entre sí sin requerir una infraestructura fija, como estaciones base o routers cableados. Cada nodo en la red actúa simultáneamente como terminal —que genera y recibe información— y como enrutador, facilitando el envío de datos a través de múltiples saltos (*multihop*) hasta alcanzar su destino. Esta capacidad de autoorganización hace que las MANET sean especialmente útiles en contextos donde no es posible contar con infraestructura tradicional, como en zonas de desastre, operaciones militares, entornos rurales o redes vehiculares dinámicas.

El diseño y funcionamiento de una red MANET implica afrontar múltiples **desafíos técnicos** derivados principalmente de la **movilidad constante** de los nodos y la inestabilidad inherente a los enlaces inalámbricos. A diferencia de las redes convencionales, sustentadas en una **topología fija**, las MANET presentan una **topología altamente dinámica**, influenciada por el desplazamiento de los nodos, la interferencia del medio de transmisión y las restricciones energéticas. Esta naturaleza cambiante genera complejidades en aspectos críticos como el establecimiento y mantenimiento de rutas eficientes, la gestión del consumo energético, la calidad del servicio (QoS) y la seguridad de las comunicaciones.

2.1.1. Características de las redes MANET

Las redes **MANET** se caracterizan por la **movilidad continua** de sus nodos, como teléfonos, sensores, drones o vehículos, lo que provoca cambios constantes en la **topología de la red** durante su operación. Esta movilidad espacial, en dos o tres dimensiones, modifica dinámicamente los enlaces disponibles, ya que la **conectividad depende del rango de transmisión** entre nodos. Adicionalmente, estas redes se construyen de forma espontánea mediante procesos de **auto-configuración**, en los que basta con que los nodos compartan protocolos comunes de descubrimiento y enrutamiento para establecer enlaces funcionales. El **enrutamiento distribuido**, soportado por protocolos como **AODV**, **DSR**, **OLSR**, **B.A.T.M.A.N.** o **DYMO**, permite hallar rutas sin necesidad de un nodo central de control. Dado que los nodos pueden no estar siempre en contacto directo, la red opera mediante transmisión *multihop*, en la que los paquetes se reenvían a través de múltiples saltos para alcanzar su destino. Finalmente, las MANET deben enfrentar **restricciones inherentes de recursos**, como energía limitada, canal inalámbrico compartido y ancho de banda reducido, lo que condiciona el diseño de protocolos eficientes para el acceso al medio, calidad de servicio y seguridad.

A diferencia de las redes inalámbricas tradicionales como las redes celulares o Wi-Fi con infraestructura, las MANET operan bajo principios radicalmente distintos. Mientras que las primeras dependen de **puntos de acceso fijos** o torres de telecomunicaciones con **cobertura planificada** y **gestión centralizada**, las MANET prescinden de infraestructura fija y adoptan una **gestión distribuida**, en la que cada nodo

toma decisiones de enrutamiento de forma autónoma. Esta arquitectura les permite un **despliegue rápido y flexible**, ideal para entornos impredecibles o de emergencia, aunque con la contrapartida de **enlaces más inestables** y menor capacidad agregada. Así, mientras las redes tradicionales priorizan **estabilidad y rendimiento sostenido** a costa de altos costos de infraestructura, las MANET privilegian la **adaptabilidad** y la **autonomía** en escenarios dinámicos.

2.1.2. Aplicaciones de las redes MANET

Gracias a su **flexibilidad, autonomía y capacidad de autoorganización**, las redes MANET encuentran aplicación en diversos contextos donde las soluciones de comunicación tradicionales resultan inviables o insuficientes. En el ámbito **militar**, son fundamentales para establecer comunicaciones tácticas robustas y descentralizadas en entornos hostiles o cambiantes, permitiendo la conectividad entre vehículos, unidades en movimiento y sistemas no tripulados (UAVs). De forma similar, en situaciones de **emergencia** como terremotos, incendios forestales o inundaciones, estas redes facilitan la coordinación entre equipos de rescate, al permitir el despliegue rápido de enlaces de comunicación en ausencia de infraestructura.

Otras aplicaciones relevantes incluyen la creación de **redes comunitarias o rurales**, donde los proveedores de servicios de internet (ISP) no tienen presencia y se requiere una solución autosostenible para la conectividad local. En entornos urbanos, las **redes vehiculares ad hoc (VANET)**, derivadas de la lógica de las MANET, permiten la transmisión de alertas de tráfico y la coordinación entre automóviles para mejorar la seguridad vial. Por último, en escenarios de **Internet de las Cosas (IoT)**, las MANET sirven como soporte para el monitoreo distribuido de variables ambientales, cultivos o maquinaria en aplicaciones como la **agricultura de precisión** o la gestión remota de infraestructuras críticas.

2.2. Jerarquía de Redes MANET

A medida que las redes MANET crecen en tamaño y complejidad, la gestión descentralizada tradicional comienza a enfrentar serias limitaciones en términos de **eficiencia, escalabilidad y sobrecarga**. Para abordar estos desafíos, se recurre a la incorporación de **estructuras jerárquicas** que permiten organizar los nodos en niveles funcionales, como **clústeres, zonas o dominios**. En este modelo, ciertos nodos asumen roles especializados —como los **Cluster Heads (CH)** o **Gateways**— y conforman un **backbone lógico** encargado de facilitar el control distribuido de la red. Esta organización jerárquica permite **delegar tareas críticas** de enrutamiento, coordinación y asignación de recursos, lo que reduce la carga global y mejora el rendimiento en entornos altamente dinámicos. Además, al limitar la propagación de paquetes de control a contextos locales, se disminuye significativamente la congestión del canal y se incrementa la eficiencia operativa de la red.

2.2.1. Arquitectura jerárquica en MANET

La arquitectura jerárquica en redes MANET introduce un modelo organizativo por niveles que permite mejorar la escalabilidad y reducir la sobrecarga de señalización en entornos altamente dinámicos. A diferencia de las arquitecturas planas, en las cuales todos los nodos comparten responsabilidades similares en el proceso de enrutamiento, una red MANET jerárquica delega ciertas funciones de control y coordinación en nodos especializados, formando una estructura lógica de múltiples capas.

Este tipo de organización se implementa comúnmente a través de mecanismos de *clustering*, donde los nodos se agrupan en dominios o zonas locales. Cada clúster es gestionado por un nodo líder, conocido como **Cluster Head (CH)**, responsable de mantener la conectividad intra-clúster y de actuar como punto de enlace hacia otros clústeres. Sobre esta base, es posible establecer niveles adicionales de jerarquía mediante la inclusión de nodos de mayor rango, o *supernodos*, que se encargan de optimizar la comunicación entre grupos de CH, formando así un **backbone multihop** de control.

En términos funcionales, esta jerarquía puede descomponerse en tres niveles principales:

- **Nivel de nodos finales:** compuesto por dispositivos móviles comunes como teléfonos, sensores, vehículos o UAVs, que generan y reciben datos, pero delegan el enrutamiento y la coordinación a nodos superiores.

- **Nivel de clústeres:** donde se agrupan nodos bajo la gestión de un CH, encargado del mantenimiento de rutas locales y de la estabilidad del clúster.
- **Nivel de supernodos o coordinadores interclúster:** responsables de enlazar múltiples clústeres, manejar tráfico agregado y reducir la latencia en trayectos de largo alcance.

Este enfoque jerárquico resulta especialmente útil en escenarios complejos como redes tácticas militares, operaciones de búsqueda y rescate, o redes vehiculares distribuidas, donde diferentes grupos de nodos pueden requerir coordinación diferenciada según su ubicación o función operativa.

2.2.2. Beneficios de la jerarquía en MANET

La incorporación de una **estructura jerárquica** en redes MANET aporta ventajas clave en entornos caracterizados por alta movilidad, recursos limitados y necesidad de escalabilidad. Entre los beneficios más relevantes se destacan:

- **Escalabilidad aumentada:** al centralizar parcialmente la toma de decisiones en nodos específicos, se reduce la complejidad del enrutamiento global, lo que permite gestionar un mayor número de nodos sin comprometer la estabilidad de la red.
- **Reducción de la sobrecarga de control:** el tráfico de señalización se limita principalmente al interior de cada clúster, evitando la propagación innecesaria de mensajes de control a través de toda la red y optimizando el uso del **ancho de banda**.
- **Mejora en la calidad de servicio (QoS):** la jerarquía facilita la aplicación de políticas diferenciadas de tráfico, priorizando flujos críticos como **voz**, **video en tiempo real** o datos de sensores, en función de su ubicación dentro de la estructura.
- **Tolerancia a fallos localizada:** la segmentación en clústeres permite aislar errores o desconexiones sin comprometer la totalidad de la red. La pérdida de un nodo o enlace puede ser gestionada de forma local, incrementando la **robustez** del sistema.
- **Eficiencia energética:** dado que solo los **Cluster Heads** y enlaces críticos participan activamente en la retransmisión de datos, los nodos ordinarios pueden operar en estados de bajo consumo, prolongando la vida útil de la red en escenarios energéticamente restringidos.

2.2.3. Principales problemas en MANETs jerárquicas de 2 niveles

La implementación de una arquitectura jerárquica de dos niveles presenta desafíos significativos, entre los cuales destacan los siguientes al modelar o simular redes MANET:

1. **Limitaciones de los modelos de movilidad grupal.** Modelos como *RPGM* asumen que todos los nodos siguen de forma continua al líder del grupo, sin considerar dinámicas más complejas como divisiones o fusiones entre clústeres. Esta simplificación conlleva una subestimación del tráfico de control necesario para mantener la cohesión del grupo y oculta pérdidas de paquetes que ocurren cuando los nodos se desconectan momentáneamente del clúster.
2. **Costos asociados al re-clustering.** Cuando un *Cluster Head* se desplaza rápidamente o pierde conectividad con sus nodos, debe transferir su rol a otro nodo, lo que implica procesos de reelección, resincronización de slots temporales (e.g. TDMA) y actualización de estados. Este comportamiento genera ráfagas de mensajes de control, elevando el consumo energético y el uso del canal en momentos críticos de inestabilidad topológica.
3. **Problemas de escalabilidad del escenario.** En áreas pequeñas, los clústeres tienden a rebotar en los límites del entorno simulado, generando patrones artificiales de movimiento. En cambio, en áreas demasiado grandes, pueden aparecer vacíos entre los CH y los nodos, provocando aislamiento temporal. Estas variaciones hacen que los resultados obtenidos sean altamente sensibles al tamaño del escenario, dificultando su extrapolación a redes reales.

4. **Ausencia de trazas jerárquicas reales.** La mayoría de conjuntos de datos disponibles públicamente para simulaciones están basados en redes planas, como movimientos individuales de vehículos o peatones. Esto obliga a parametrizar las simulaciones jerárquicas de forma arbitraria, reduciendo la reproducibilidad de los experimentos y dificultando las comparaciones justas entre diferentes protocolos de enrutamiento jerárquico.

2.2.4. Comunicación intra-clúster e inter-clúster

En el contexto de una **red MANET jerárquica**, el modelo de comunicación se organiza en dos planos complementarios que responden a la segmentación lógica de la red: la **comunicación intra-clúster** y la **comunicación inter-clúster**. Cada uno presenta características particulares en términos de alcance, complejidad y gestión de recursos.

- **Comunicación intra-clúster:** se refiere al intercambio de datos dentro de un mismo clúster, donde los nodos seguidores se conectan de forma directa o indirecta con su respectivo *Cluster Head* (**CH**). Este nodo central coordina el acceso al canal y puede aplicar esquemas de control eficientes, como **TDMA** o protocolos de **prioridad adaptativa**, con el fin de evitar colisiones y optimizar la transmisión local. Una herramienta clave para mantener esta organización es el uso de **beacons**, pequeños paquetes de control que el **CH** emite periódicamente para anunciar su presencia y permitir que los nodos seguidores verifiquen la conectividad. Estos **beacons** son esenciales para la **sincronización** del clúster, la **detección de eventos de desacoplamiento** y la **gestión dinámica de enlaces** en escenarios de alta movilidad. Esta centralización parcial reduce la **interferencia entre nodos** y mejora el rendimiento en aspectos como la **sincronización**, el **balanceo de carga** y el **mantenimiento de rutas estables y cortas**.
- **Comunicación inter-clúster:** comprende los flujos de información entre diferentes clústeres, y es gestionada principalmente por los propios **Cluster Heads** o por nodos designados como **gateways**. Estos nodos actúan como puentes entre dominios lógicos y emplean **protocolos de enrutamiento jerárquico** para mantener la conectividad global. La construcción de rutas entre clústeres puede apoyarse en mecanismos de agregación de información topológica parcial, lo cual permite reducir la **sobrecarga de señalización** y minimizar la **latencia**, especialmente en entornos con alta movilidad o distribución desigual de nodos.

La coexistencia de estos dos niveles de comunicación permite a la red adaptarse de forma más eficaz a entornos cambiantes, especialmente en escenarios donde se requiere coordinación entre múltiples grupos móviles, como en **operaciones de rescate**, **despliegues tácticos militares** o redes vehiculares distribuidas.

2.3. Protocolo de enrutamiento Cluster-Based

Los **protocolos de enrutamiento jerárquico basados en clústeres** (*Cluster-Based Routing*, CBR) surgen como solución a los problemas de **escalabilidad** en redes MANET de gran tamaño. Estos protocolos organizan la red en grupos lógicos autogestionados, en los que cada **clúster** es coordinado por un **Cluster Head (CH)**, elegido de manera dinámica, que centraliza el enrutamiento local y actúa como puente hacia otros clústeres, además los **nodos miembros** delegan las decisiones de enrutamiento al CH, reduciendo así la **sobrecarga computacional** y el **consumo energético**.

Entre las ventajas más destacadas frente a protocolos planos (como AODV) se incluyen:

- **Reducción del overhead de control:** se limita la propagación de mensajes de enrutamiento al ámbito del clúster, con reducciones reportadas de hasta un 58 % en tráfico de control.
- **Adaptabilidad topológica:** los eventos de movilidad afectan principalmente a nivel local, lo que reduce la necesidad de recomputaciones globales.

2.3.1. Principios del enrutamiento Cluster-Based

El funcionamiento de los protocolos CBR se estructura generalmente en tres fases:

1. Formación de clústeres:

La elección del **Cluster Head** se realiza con base en métricas como la **energía residual**, el **grado de conectividad** o la **estabilidad de movilidad**.

Ejemplo: en el protocolo HEED, el CH se selecciona ponderando la energía disponible y la densidad de nodos vecinos.

2. Mantenimiento de rutas:

Los CH intercambian información de accesibilidad a través de **gateways** (nodos frontera) y establecen rutas inter-clúster jerárquicas, evitando el uso de inundación global.

3. Reconfiguración dinámica:

Activada por la movilidad del CH o por cambios en la densidad nodal. Algoritmos como *Weighted Clustering* optimizan la reelegibilidad del CH para minimizar oscilaciones estructurales.

Si bien los protocolos de enrutamiento basados en clústeres ofrecen ventajas significativas en términos de **escalabilidad**, **eficiencia energética** y **reducción de sobrecarga**, también presentan ciertas limitaciones que deben ser consideradas según el contexto de aplicación. En escenarios con alta densidad nodal, la frecuencia de procesos de reelección de **Cluster Heads** tiende a aumentar. Asimismo, la naturaleza jerárquica de estos protocolos puede derivar en rutas menos óptimas, con trayectos más largos en comparación con esquemas planos, lo cual repercute en la **latencia** y el uso de **recursos de red**. Estas compensaciones operativas deben ser evaluadas cuidadosamente al diseñar soluciones de enrutamiento para entornos dinámicos y heterogéneos.

En el contexto de este trabajo, el **protocolo Cluster-Based** se extiende mediante la introducción del **nodo CH-temporal**, el cual actúa como **entidad de recuperación local** durante eventos de desacoplamiento, reduciendo la **latencia de re-clusterización** y manteniendo la **conectividad intra-clúster**.

2.3.2. Liderazgo dinámico y algoritmo WCA

Cada nodo puede desempeñar el rol de líder o seguidor. Al inicio de la simulación, se asignan aleatoriamente los roles. A los líderes se les asigna un objetivo dentro del área de simulación, el cual puede cambiar en respuesta a eventos externos.

Si un nodo seguidor se encuentra aislado, es decir, sin un líder en su radio de cohesión, evalúa su potencial de liderazgo utilizando el algoritmo *WCA* (Weighted Clustering Algorithm). Este algoritmo considera múltiples factores que se normalizan para obtener un puntaje de liderazgo (*WCA_Score*):

- *Energía residual* (E): porcentaje de energía restante del nodo, normalizada en el rango $[0, 1]$.
- *Grado de conectividad* (D): número de vecinos directos, normalizado respecto a un máximo teórico de 10.
- *Distancia a objetivos* (T): inversa de la distancia promedio del nodo a los objetivos activos, normalizada considerando un rango de hasta 200 metros.
- *Movilidad* (M): estabilidad del nodo, calculada como la inversa de su velocidad actual, con una velocidad máxima de 10 m/s.

El cálculo del puntaje se define como:

$$WCA_Score = w_1 \cdot E + w_2 \cdot D + w_3 \cdot T + w_4 \cdot M$$

donde los pesos utilizados son:

$$w_1 = 0,4, \quad w_2 = 0,3, \quad w_3 = 0,2, \quad w_4 = 0,1$$

El resultado se limita al rango $[0, 1]$ mediante:

$$WCA_Score = \max(0, \min(1, WCA_Score))$$

El nodo aislado con el mayor `WCA_Score` se autonombrará líder en ausencia de otros líderes cercanos.

2.4. Simulación de MANET con ns-3

El **Network Simulator 3 (ns-3)** es una plataforma de simulación por eventos discretos ampliamente utilizada en el modelado y análisis de redes de comunicación. Su arquitectura modular y alto grado de personalización permiten simular una amplia variedad de escenarios, desde redes cableadas hasta entornos inalámbricos complejos como las **redes móviles ad hoc (MANET)**. Gracias a su soporte para múltiples capas de red, **ns-3** se ha consolidado como una herramienta de referencia en investigación académica y pruebas experimentales previas a la implementación real de sistemas.

En el contexto de MANET, **ns-3** ofrece capacidades específicas para modelar fenómenos como la **movilidad nodal**, la **variabilidad en la calidad del enlace** y el comportamiento de distintos **protocolos de enrutamiento**. Además, permite configurar dinámicamente patrones de tráfico, políticas de acceso al canal y esquemas de interacción entre nodos, lo cual resulta fundamental para evaluar el rendimiento de estas redes bajo condiciones controladas y reproducibles. Esta capacidad de abstracción y experimentación controlada lo convierte en una herramienta esencial para analizar la viabilidad y eficiencia de diferentes arquitecturas y estrategias en MANET.

2.4.1. Características de ns-3.45 para la simulación de MANET

La versión 3.45 de **ns-3** proporciona un marco de simulación robusto para redes MANET, con mejoras significativas en términos de **precisión, escalabilidad y realismo en la modelación de entornos inalámbricos**. Entre sus funcionalidades clave destacan:

1. Modelos avanzados de movilidad

Novedad en 3.45: implementación del modelo *Gauss-Markov 3D* para simular movilidad aérea, como en drones o UAVs.

Modelos soportados:

- `RandomWaypointMobilityModel` (movimiento aleatorio)
- `HierarchicalMobilityModel` (movilidad grupal jerárquica)
- `SteadyStateRandomWaypointMobilityModel` (elimina el sesgo hacia el centro)
- Integración con datos GPS reales mediante `TraceMobilityHelper`

2. Herramientas de análisis y visualización

Soporte para:

- `FlowMonitor` para métricas de rendimiento
- `NetAnim 3.45` para animación del comportamiento de red
- Exportación de datos en formatos `PCAP`, `SQLite` y visualización mediante `PyViz` para análisis en Python

2.5. Modelos de movilidad en MANET

La movilidad nodal es un factor crítico que determina el desempeño de redes MANET, afectando directamente la estabilidad de enlaces, overhead de control y eficiencia de enrutamiento. Para estudiar estos efectos, se han desarrollado tres categorías fundamentales de modelos:

- **Modelos no estructurados:** Movimiento aleatorio independiente (ej. Random Waypoint)
- **Modelos grupales:** Coordinación jerárquica entre nodos (ej. RPGM)
- **Modelos especializados:** Patrones específicos por aplicación (ej. táctico-militares)

2.5.1. Modelo de movilidad Boids

El modelo de movilidad **Boids**, propuesto por Craig Reynolds en 1987, simula el comportamiento colectivo de agentes autónomos a partir de reglas locales simples que generan dinámicas emergentes de grupo. Este modelo se basa en tres principios fundamentales:

- **Separación:** evitar colisiones con nodos cercanos manteniendo una distancia mínima segura.
- **Alineación:** ajustar la dirección del movimiento para coincidir con la orientación promedio de los vecinos.
- **Cohesión:** moverse hacia el centro de masa del grupo de nodos más cercanos.

A diferencia de enfoques jerárquicos como RPGM, el modelo Boids adopta una lógica completamente **descentralizada**, en la cual cada nodo toma decisiones en función de su **entorno local**, sin la necesidad de un líder designado.

Esta aproximación permite generar comportamientos emergentes realistas, como formaciones dinámicas, dispersión adaptativa o evasión de obstáculos de manera fluida. Su naturaleza descentralizada mejora la **tolerancia a fallos**, al eliminar puntos únicos de dependencia jerárquica, y facilita la **adaptación continua** a entornos cambiantes. Estas características lo hacen especialmente adecuado para simular **enjambres robóticos**, **equipos de rescate** o nodos móviles en misiones cooperativas no estructuradas.

Sin embargo, el modelo también presenta ciertas limitaciones. Requiere una **calibración precisa de parámetros** para mantener la cohesión grupal, puede exhibir **comportamientos impredecibles** en escenarios de gran escala, y demanda una mayor **carga computacional** que los modelos con control jerárquico.

2.6. Integración

La integración de los fundamentos teóricos expuestos – particularmente sobre **movilidad tipo Boids** (Sección 2.5.3), **liderazgo adaptativo mediante WCA** (Sección 2.3.2) y **simulaciones MANET** (Sección 2.4) – revela una brecha operativa crítica en escenarios de emergencia: la ausencia de mecanismos de respuesta inmediata cuando nodos aislados detectan incendios sin líderes cercanos. Este vacío motiva un marco de resiliencia dinámica que sintetiza:

1. **Autoproclamación de líderes temporales** para coordinación de emergencia en zonas sin cobertura de chNodes.
2. **Criterios de transferencia jerárquica** basados en mérito de nodo (WCA) y urgencia del siniestro.

La formalización de este marco sustenta la pregunta de indagación central, cuyas hipótesis y modelo operativo se derivan de principios de **sistemas auto-organizativos**, **teoría de respuesta a emergencias** y optimización dinámica de recursos.

2.6.1. Problemática en redes con movilidad tipo Boids para respuesta a emergencias

En redes MANET con **movilidad tipo Boids**, los nodos interactúan con base en principios locales de separación, alineación y cohesión, sin una jerarquía predefinida. No obstante, para tareas complejas de coordinación como la **extinción de incendios**, este comportamiento autoorganizado se complementa con la presencia de **líderes funcionales** o *Cluster Heads* (chNodes), seleccionados dinámicamente mediante el algoritmo WCA. Estos líderes asumen la responsabilidad de dirigir acciones colectivas, distribuir tareas y priorizar recursos en zonas críticas.

A pesar de la flexibilidad de este enfoque mixto, se identifican vulnerabilidades operativas en escenarios donde emergen focos de incendio en áreas sin cobertura inmediata de chNodes. Esto ocurre particularmente en zonas periféricas de la red o cuando los líderes activos se encuentran sobrecargados, desplazados o temporalmente desconectados. Como resultado, algunos nodos pueden quedar **aislados**, es decir, sin un líder dentro de su radio efectivo de influencia.

Ante esta situación, los nodos aislados que detectan incendios enfrentan una limitación estructural: no están autorizados para ejecutar acciones de respuesta, ya que dependen de la validación y coordinación del

chNode más cercano. Esto genera una **ventana de inacción**, en la cual el nodo permanece en estado pasivo mientras espera indicaciones externas, incluso si cuenta con los sensores, energía y proximidad necesarios para actuar.

Este vacío de coordinación tiene consecuencias directas en la eficiencia del sistema:

- Se retrasa la activación de mecanismos de contención en zonas críticas.
- Se pierde la oportunidad de responder tempranamente, cuando el incendio aún es manejable.
- Se desaprovechan recursos locales disponibles para la mitigación.

Esta problemática motiva la necesidad de incorporar un mecanismo de resiliencia adaptativa que permita a los nodos aislados asumir, de forma temporal, funciones de liderazgo ante la detección de eventos críticos, contribuyendo así a preservar la capacidad de respuesta del sistema frente a condiciones dinámicas e impredecibles.

2.6.2. Propuesta de solución: Nodo CH-temporal

Como respuesta a las limitaciones descritas previamente, se propone la incorporación de un **mecanismo de liderazgo temporal** que permita a los nodos aislados asumir un rol activo en situaciones de emergencia. En particular, cuando un nodo seguidor detecta un incendio (**s_fire**) en su proximidad y no tiene un *Cluster Head* (chNode) dentro de su radio efectivo de coordinación R , este puede transitar de forma autónoma a un **estado de líder temporal** (**tempCH**).

En este estado, el nodo realiza las siguientes acciones:

1. Se autoproclama **tempCH** para el manejo local del evento detectado.
2. Activa el protocolo de extinción del incendio en su área de influencia.
3. Coordina la colaboración de nodos cercanos, si los hay, para ejecutar acciones conjuntas de contención.
4. Permanece en estado de liderazgo hasta que detecta un chNode cercano con un valor de WCA superior, momento en el cual transfiere el control y retorna a su estado de nodo seguidor.

Esta estrategia ofrece múltiples beneficios operativos:

- **Reduce la latencia de respuesta** en eventos críticos donde la reacción temprana es esencial.
- **Mitiga la propagación** de incendios en zonas que, de otro modo, quedarían desatendidas.
- **Evita la necesidad de reconfiguración global**, actuando como una solución local, ágil y de bajo costo computacional.

La figura del **tempCH** funciona, por tanto, como una extensión adaptativa del esquema tradicional de liderazgo jerárquico, dotando al sistema de una **capacidad de resiliencia distribuida** que fortalece su desempeño ante condiciones altamente dinámicas y demandas imprevistas.

3. Descripción y Justificación del Problema a Resolver

3.1. Contexto del Problema

En una Red Ad Hoc Móvil (MANET) compuesta por nodos con **movilidad tipo Boids** y un esquema de **liderazgo adaptativo** que utiliza el algoritmo WCA (*Weighted Cluster Algorithm*) para la selección y mantenimiento de líderes, donde los **líderes (chNodes)** son los encargados de **extinguir fuegos (s_fires)** que aparecen aleatoriamente en el entorno de simulación, surge una pregunta fundamental:

¿No será que la **efectividad global en la extinción de fuegos** (medida por métricas como el tiempo promedio de extinción, el número total de fuegos extinguidos en un periodo dado) **se incrementa** si se permite que un nodo seguidor que previamente estaba **aislado** (sin un líder cercano dentro de un radio efectivo de influencia) **autoproclame temporalmente como líder hasta que se encuentre con uno con un mayor WCA**?

Esta problemática surge cuando los nodos experimentan:

- **Obstáculos naturales:** Árboles, relieve montañoso y estructuras que bloquean la comunicación
- **Interferencias electromagnéticas:** Degradación de la calidad de señal en entornos adversos
- **Movilidad dinámica:** Patrones de movimiento tipo Boids que pueden generar separaciones temporales

3.2. Problema Específico

El desafío operativo central radica en la **pérdida de efectividad operacional** cuando un nodo seguidor se aísla de su líder designado. En el contexto de extinción de incendios, esto se manifiesta como:

1. **Oportunidades perdidas de extinción:** Los nodos aislados no pueden responder a fuegos cercanos, esperando pasivamente la reconexión con un líder
2. **Ineficiencia en la cobertura espacial:** Las zonas donde se encuentran nodos aislados quedan desprotegidas hasta que se restablezca la conectividad
3. **Degradación del rendimiento global:** El tiempo promedio de extinción de fuegos se incrementa debido a la inactividad forzada de nodos aislados

3.3. Objetivo Principal

Este enfoque ofrece ventajas significativas en términos de **realismo**, puesto que captura comportamientos grupales observados en aplicaciones militares (unidades móviles) o VANETs (convoyes), y **eficiencia computacional**, al reducir el *overhead* de simulación comparado con modelos totalmente independientes. Sin embargo, presenta limitaciones importantes: requiere sintonización fina de d_{max} para evitar particiones prematuras de grupos, y asume cooperación plena entre nodos sin modelar deserciones aleatorias que puedan ocurrir en escenarios reales.

3.4. Hipótesis de Investigación

Hipótesis Principal: La implementación de un mecanismo de **liderazgo temporal auto-proclamado** en nodos aislados mejora significativamente la efectividad global del sistema de extinción de fuegos en comparación con esquemas tradicionales de liderazgo fijo.

Hipótesis Específicas:

- H1: El tiempo promedio de extinción de fuegos disminuye cuando los nodos aislados pueden actuar como líderes temporales
- H2: El número total de fuegos extinguidos por unidad de tiempo se incrementa con el mecanismo propuesto
- H3: La cobertura espacial efectiva del sistema mejora al permitir liderazgo temporal en nodos aislados

3.5. Justificación de la Solución Propuesta

Nuestra propuesta introduce un **mecanismo de liderazgo temporal adaptativo** basado en los siguientes componentes:

3.6. Métricas de Evaluación

Para validar la efectividad de la solución propuesta, se utilizarán las siguientes métricas:

- **Tiempo promedio de extinción:** Tiempo transcurrido desde la aparición de un fuego hasta su extinción completa
- **Tasa de extinción:** Número de fuegos extinguidos por unidad de tiempo

Cuadro 1: Componentes del mecanismo de liderazgo temporal

Componente	Descripción
Detección de aislamiento	Algoritmo para identificar cuando un nodo pierde conectividad con líderes en su radio de influencia
Auto-proclamación temporal	Mecanismo que permite a nodos aislados asumir rol de líder con WCA temporal
Transición de liderazgo	Protocolo para transferir liderazgo cuando un nodo temporal encuentra un líder con mayor WCA
Gestión de conflictos	Resolución de situaciones donde múltiples nodos temporales coexisten

4. Diseño de la solución

La solución desarrollada está basada en el modelo de comportamiento de Boids para simular la movilidad de nodos en una red ad hoc. Este modelo adapta las reglas clásicas de *cohesión*, *alineación* y *separación*, incorporando elementos adicionales como nodos líderes, mecanismos de selección dinámica de líderes mediante el algoritmo de clustering ponderado (WCA), interacción con eventos externos (como incendios) y una infraestructura de comunicación entre nodos basada en enlaces bidireccionales.

Propósito General del Sistema

El propósito primordial de este sistema es simular la dinámica de una Red Ad Hoc Móvil (MANET), la cual integra un esquema de liderazgo adaptativo basado en el modelo de movilidad tipo Boids.

El objetivo central es evaluar el impacto del mecanismo de auto-proclamación de liderazgo en la **eficiencia de la supresión de fuegos** dentro de dicho entorno.

Alcance de la Simulación

La simulación se centra en responder la siguiente cuestión fundamental:

¿La efectividad global en la extinción de fuegos se incrementa al facultar a un nodo seguidor previamente aislado –definido como aquel que no detecta un líder cercano dentro de un radio de influencia extendido de 1,2 veces el radio normal de influencia del líder– para que se **autoprocamente temporalmente como líder**, siempre y cuando su `wcaScore` sea superior a 0.0?

Este rol de liderazgo **auto-proclamado** es transitorio y se mantiene **hasta que el nodo encuentra a otro líder con una puntuación WCA superior**, momento en el cual **cede su estatus** y retoma su rol de seguidor.

4.1. Metodología

El desarrollo siguió un enfoque incremental. Inicialmente se construyó la jerarquía básica de nodos (líderes y seguidores), luego se integró la lógica de comportamiento y movilidad basada en el modelo Boids. Posteriormente se añadieron componentes estocásticos y eventos del entorno, como focos de incendio. Finalmente, se realizaron múltiples simulaciones para evaluar el rendimiento del modelo frente a distintos escenarios y configuraciones.

4.2. Estructura general del sistema

El sistema está compuesto por los siguientes módulos principales:

- **BoidsMobilityModel**: clase que extiende el modelo de movilidad de NS-3 y contiene la lógica del comportamiento individual de cada nodo.

- **Gestor de líderes:** módulo encargado de la asignación inicial de líderes y del recálculo dinámico basado en WCA.
- **Módulo de eventos externos:** introduce elementos del entorno como incendios que influyen en el comportamiento de los nodos líderes.
- **Sistema de comunicación:** permite el intercambio de mensajes entre nodos vecinos, facilitando la detección de líderes, el seguimiento de objetivos y la formación de clusters.

4.3. Reglas de comportamiento Boid

Cada nodo se comporta como un Boid y actualiza su posición en función de su rol y de las siguientes reglas:

- **Separación:** evita la colisión entre nodos manteniendo una distancia mínima entre vecinos cercanos.
- **Alineación:** los seguidores adaptan su dirección en función de la velocidad promedio de sus vecinos más cercanos.
- **Cohesión:** los seguidores buscan mantenerse próximos a su líder si se encuentra dentro del radio de cohesión definido.
- **Atracción hacia objetivos:** los líderes se orientan hacia eventos externos (como focos de incendio) ubicados dentro de su radio de influencia.

4.4. Espacio de simulación y condiciones de frontera

La simulación se desarrolla en un área de 1000×1000 unidades. Se implementa una condición de frontera toroidal (*wrapping*), permitiendo que los nodos que salen por un borde reaparezcan por el lado opuesto. Esto evita acumulaciones artificiales y simula un entorno continuo sin bordes rígidos.

4.5. Actualización de posiciones

Cada nodo actualiza su posición en intervalos regulares de tiempo, considerando sus vectores de dirección y las fuerzas derivadas de las reglas del modelo Boid. Se aplican límites a la velocidad y se normalizan las direcciones para evitar aceleraciones bruscas o inmovilización de nodos.

4.6. Visualización y exportación de datos

Al finalizar cada simulación, se exportan datos relevantes a archivos de salida, incluyendo las posiciones de los nodos, los roles asignados (líder o seguidor), las trayectorias seguidas por cada nodo y la ubicación de eventos externos, como los incendios. Esta información permite un análisis post-simulación de métricas clave como cobertura, tiempo de respuesta o efectividad en la extinción de eventos.

4.7. Comportamiento de líderes y seguidores

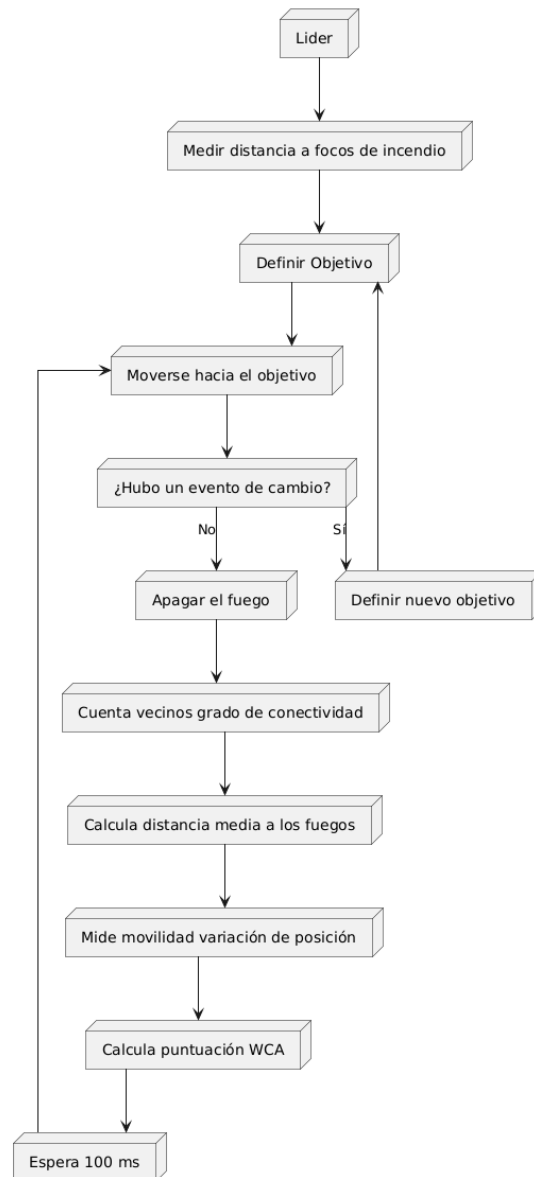


Figura 1: Diagrama de flujo del líder

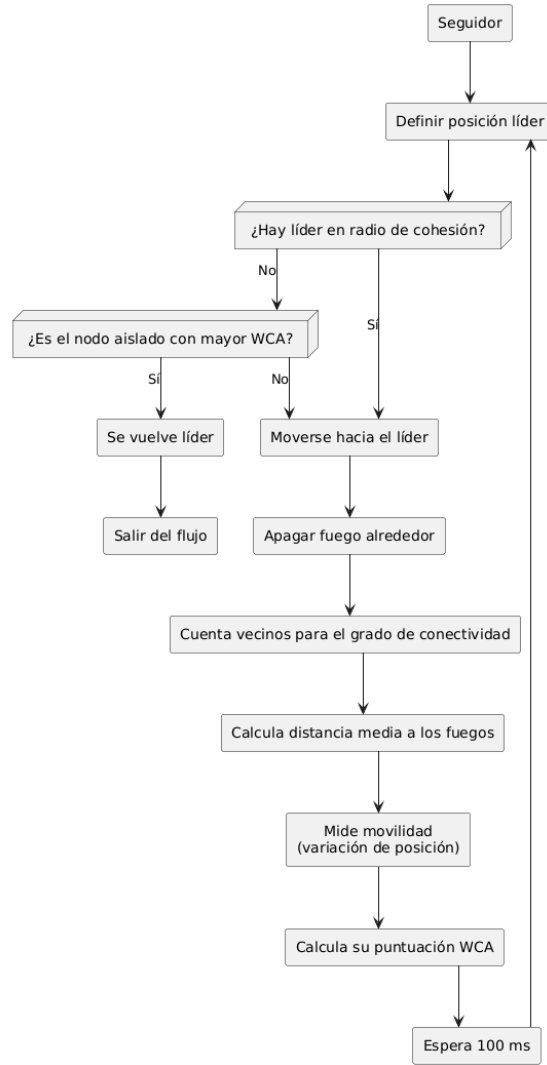


Figura 2: Diagrama de flujo del seguidor

4.8. Configurabilidad del modelo

La clase `BoidsMobilityModel` incluye métodos `Set` que permiten ajustar los parámetros más relevantes del comportamiento de los nodos, tales como el radio de cohesión de los seguidores, el radio de influencia de los líderes, la velocidad máxima de los nodos, la frecuencia de actualización de posición, la inicialización del rol de los nodos (porcentaje de líderes), y parámetros del algoritmo WCA como pesos y umbrales de energía o distancia.

En la Figura 3 se presenta el diagrama UML de la clase `BoidsMobilityModel`, donde se observan sus atributos principales, métodos de configuración, y lógica asociada al liderazgo y simulación de incendios.

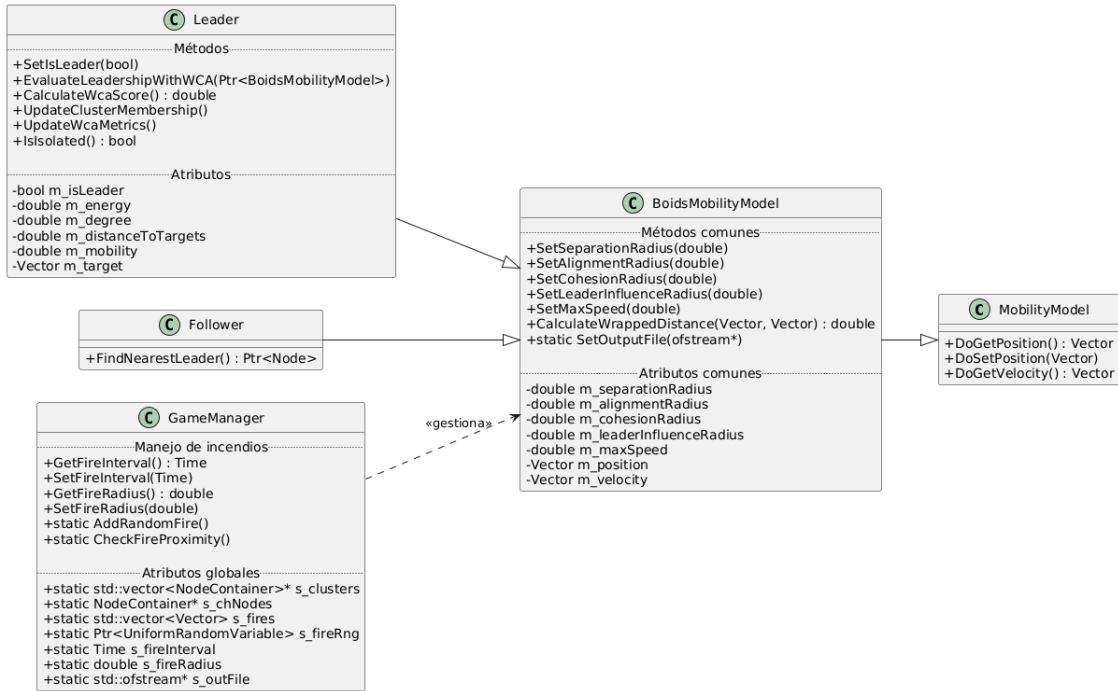


Figura 3: Diagrama UML de la clase BoidsMobilityModel.

4.9. Diagrama de Flujo de la Simulación

El diagrama de flujo de la simulación ilustra el proceso general de la simulación, desde la inicialización de los nodos y su asignación de roles, hasta la actualización periódica de sus posiciones y la gestión de incendios. En cada iteración, los nodos evalúan su estado, actualizan sus posiciones según las reglas del modelo Boids, y determinan si deben asumir un rol de líder temporal en caso de estar aislados.

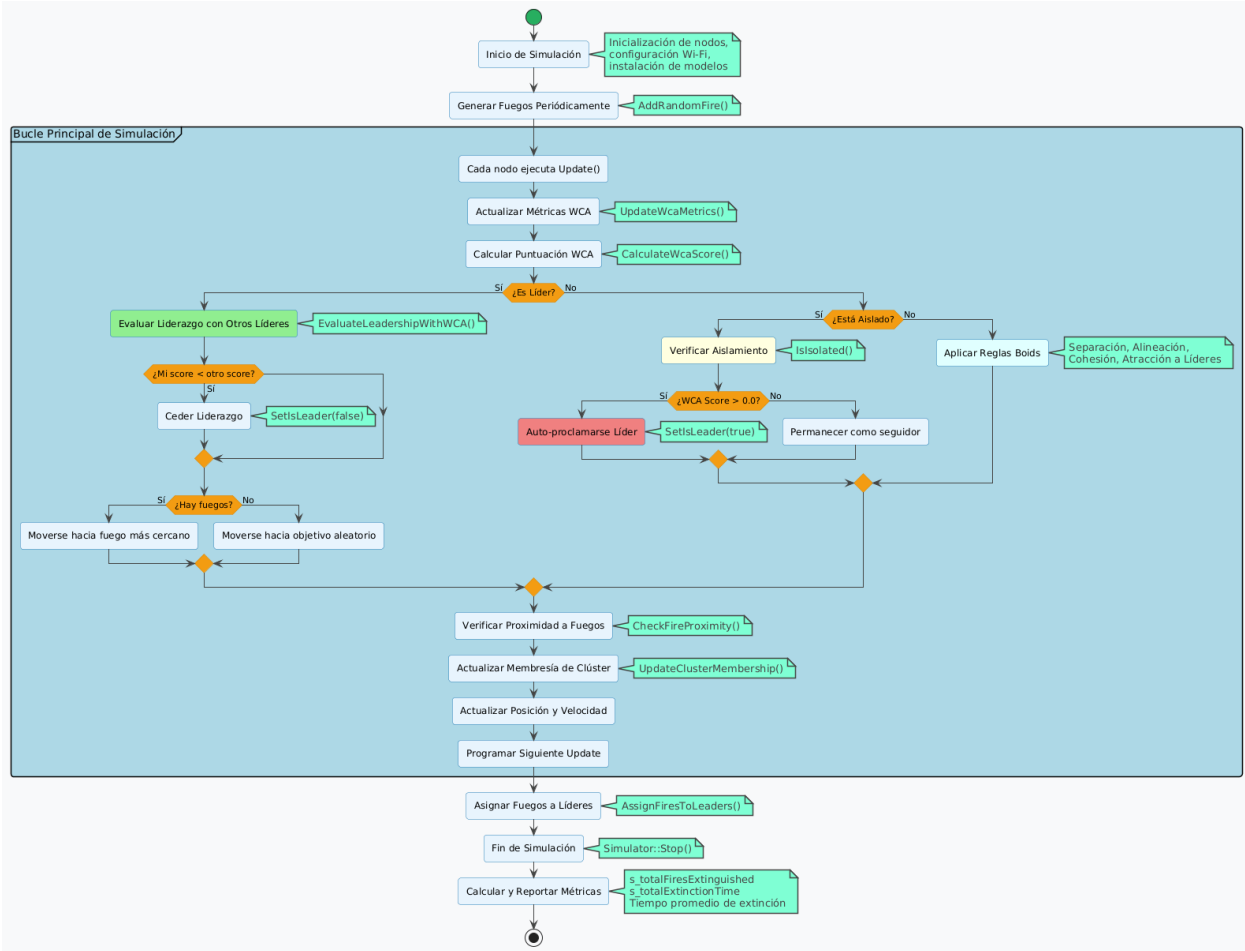


Figura 4: Diagrama de flujo de la simulación.

4.10. Modelos Estocásticos y Distribuciones Empleadas en la Simulación de MANET para Extinción de Fuegos

La simulación de Redes Ad Hoc Móviles (MANET) para la extinción de fuegos incorpora elementos de aleatoriedad para simular un comportamiento realista tanto en el entorno como en los nodos. Esto se logra mediante el empleo de diversos modelos estocásticos y distribuciones de probabilidad.

1. Generación de Fuegos: Proceso de Clúster de Thomas (Poisson Espacial)

Los fuegos no aparecen de manera completamente aleatoria, sino que tienden a agruparse, lo cual se simula mediante un **proceso de clúster de Thomas** [1-4]. Este proceso consta de dos etapas principales:

- **Generación de Centros de Clúster:** Se generan k (por defecto 3 [4]) centros de clúster de forma uniforme dentro del área de simulación (1000x1000 unidades) [3].

$$C_i.x \sim U(0, 1000) \quad \text{y} \quad C_i.y \sim U(0, 1000)$$

Donde C_i representa el i -ésimo centro de clúster, y $U(a, b)$ denota una **distribución uniforme continua** en el intervalo $[a, b]$ [3].

- **Generación de Focos de Fuego Individuales:** Alrededor de cada centro C_i , se generan los focos de fuego individuales F_j . Sus posiciones se determinan añadiendo un desplazamiento aleatorio que sigue

una **distribución normal (gaussiana)** [5-7].

$$\begin{aligned}F_{j..x} &= C_{i..x} + \delta_x \\ F_{j..y} &= C_{i..y} + \delta_y\end{aligned}$$

Donde δ_x y δ_y son variables aleatorias muestreadas de una **distribución normal** con media cero y una varianza determinada por la desviación estándar:

$$\delta_x \sim N(0, \text{desviacion}^2) \quad \text{y} \quad \delta_y \sim N(0, \text{desviacion}^2)$$

La **desviación estándar por defecto** para los fuegos es de **50.0 metros** [4]. Las coordenadas resultantes se recortan para asegurar que permanezcan dentro del área de simulación [6, 7].

- **Número de Fuegos por Evento:** Cada vez que se añade un nuevo evento de fuego, el número de fuegos generados es entre 1 y 3, determinado por una **variable aleatoria uniforme entera** [8].

$$N_{\text{fuegos}} \sim U_{\text{entero}}(1, 3)$$

- **Intervalo de Aparición de Fuegos:** Los nuevos fuegos aparecen en intervalos fijos.

$$T_{\text{intervalo_fuego}} = 8 \text{ segundos} [9, 10].$$

2. Consumo de Energía de los Nodos

La **energía** (m_energy) de los nodos, una métrica clave para el cálculo del puntaje WCA, disminuye con el tiempo. El consumo se modela de forma simplificada, restando un valor muestreado de una **distribución normal** en cada actualización.

$$\Delta \text{Energía} \sim N(0, 005, \sigma^2)$$

Donde la media de 0.005 representa el consumo promedio por unidad de tiempo. La varianza (σ^2) no se especifica directamente, pero es parte de la configuración del ‘NormalRandomVariable’ utilizado [20]. El valor de energía se asegura que no sea negativo ($\text{máx}(0, 0, \dots)$) [20].

5. Código Fuente

El código fuente completo se encuentra adjunto como Taller1.zip o en el siguiente repositorio de GitHub:

https://github.com/JavierTarazona06/ME01_Tareas/tree/main/Taller1/Code/boids

6. Manual Usuario

Este manual guiará al usuario a través de los pasos necesarios para instalar NS-3 en un entorno WSL (Windows Subsystem for Linux), configurar los archivos para la simulación de boids y ejecutar dicha simulación, así como visualizar sus resultados.

0. Prerequisitos

Para seguir este manual, es necesario tener conocimientos básicos de Linux y C++. Además, se recomienda familiarizarse con el entorno de NS-3 y sus conceptos fundamentales.

Antes de comenzar, asegúrate de tener lo siguiente:

- Conexión a Internet para descargar NS-3 y sus dependencias.
- Conocimientos básicos de terminal de Linux y C++.
- CMake y un compilador C++ (g++) instalados en tu sistema.

Archivo	Función
<code>CMakeLists.txt</code>	<ul style="list-style-type: none"> - Instrucciones de construcción para compilar el proyecto. - Cómo integrar este módulo de movilidad (“mobility”) en la compilación de NS-3. - Define la librería <code>mobility</code>, lista los archivos fuente (incluyendo <code>boids-mobility-model.cc</code>), especifica dependencias (antenas, red) y registra los tests asociados.
<code>boids-mobility-model.h</code>	<ul style="list-style-type: none"> - Declaración de la clase <code>BoidsMobilityModel</code>, que hereda de <code>ns3::MobilityModel</code>. - Parámetros configurables: radios de separación, alineamiento, cohesión y liderazgo; velocidad máxima; bandera de “líder”; intervalos de “fuego” (evento interno). - Método estático <code>SetOutputFile(...)</code> para indicar dónde volcar la trayectoria de los nodos.
<code>boids-mobility-model.cc</code>	Implementación de la lógica Boids: <ul style="list-style-type: none"> - Cálculo de separación, alineamiento, cohesión y término de influencia de líderes. - Gestión de “fuegos” y escritura de posiciones en el archivo de salida.
<code>boids.cc</code>	Script de simulación “usuario” que configura nodos, asigna el modelo <code>BoidsMobilityModel</code> , fija parámetros y ejecuta la simulación.
<code>simulate/boids_positions.csv</code>	Archivo generado por la simulación NS-3 (desde <code>boids.cc</code>) con un registro tabulado de tiempo, nodo y coordenadas (x, y, z, ...).
<code>simulate/showNodes.py</code>	Script en Python que carga el CSV con <code>pandas</code> y usa <code>matplotlib</code> para animar los movimientos de los nodos.

Cuadro 2: Descripción de los archivos del proyecto Boids en NS-3

- WSL (Windows Subsystem for Linux) instalado en tu sistema Windows.
 - Preferiblemente con una distribución de Linux como Ubuntu.
- **Python 3** con los paquetes **pandas** y **matplotlib** instalados.
 - De no tenerlos instalados, puedes hacerlo ejecutando:


```
pip3 install pandas matplotlib
```

1. Instalación de NS-3 en WSL

Para instalar NS-3, se recomienda usar WSL, preferentemente con Ubuntu .

- **Iniciar WSL:** Abre tu terminal WSL.
- **Actualizar Paquetes:** Ejecuta el siguiente comando para actualizar la lista de paquetes:


```
sudo apt update
```
- **Instalar Dependencias Esenciales:** Instala las herramientas de compilación y descompresión necesarias:

```
sudo apt install g++
sudo apt install bzip2
sudo apt install make
sudo apt-get install cmake
```

- **Configurar Alias de Python:** NS-3 utiliza Python 3. Para asegurarte de que el comando `python` apunte a `python3`, añade un alias a tu archivo `.bashrc` y recarga la configuración:

```
echo alias python=python3 >> ~/.bashrc
source ~/.bashrc
```

2. Descarga y Preparación de NS-3

- **Navegar a la Carpeta de Instalación:** Dentro de WSL, tu sistema de archivos de Windows se encuentra en `/mnt/c` (si tu partición es C:) o `/mnt/d` (si es D:). Navega a la carpeta donde deseas instalar NS-3. Por ejemplo:

```
cd /mnt/c/Users/TuUsuario/Documentos/NS3_Projects
```

- **Descargar NS-3:** Descarga el archivo `ns-allinone-3.45.tar.bz2` desde la página oficial de NS-3 (<https://www.nsnam.org/releases/ns-allinone-3.45.tar.bz2>) y guárdalo en la carpeta deseada. Es preferible hacerlo desde el navegador web de Windows y luego acceder a la ruta desde WSL.

- **Extraer Archivos:** Una vez descargado, descomprime el archivo `tar.bz2`:

```
tar xjf ns-allinone-3.45.tar.bz2
```

Nota: Aunque las fuentes mencionan `ns-3.45.tar.bz2`, el paso de descarga sugiere `ns-allinone-3.45.tar.bz2`.

- **Acceder al Directorio de NS-3:** Navega al directorio principal de NS-3:

```
cd ns-3.45
```

3. Configuración y Compilación de NS-3

Desde el directorio `ns-3.45` (o la versión que estés usando), ejecuta los siguientes comandos:

- **Limpiar Compilación (Opcional pero recomendado):**

```
./ns3 clean
```

- **Configurar NS-3:** Este paso configura el entorno de compilación. Asegúrate de incluir las opciones para depuración, ejemplos, pruebas y NetAnim:

```
./ns3 configure \
—build-profile=debug \
—enable-examples \
—enable-tests \
—enable-build-version \
— —DNS3_NETANIM=ON
```

- **Compilar NS-3:** Este comando compila todo el proyecto NS-3. Puedes usar la opción `-j$(nproc)` para utilizar todos los procesadores de tu equipo y acelerar la compilación:

```
./ns3 build
```

Alternativamente, para usar todos los procesadores:

```
./ns3 build -j$(nproc)
```

■ **Verificar la Instalación (Opcional):**

- Ejecutar pruebas: `./test.py`
- Ejecutar un ejemplo básico: `./ns3 run first`
- Mostrar la versión instalada: `./ns3 show version`

4. Configuración y Ejecución de la Simulación de Boids

La simulación de boids implica modificar y añadir archivos al proyecto NS-3, de los ya presentes para una mayor simplicidad. Sin embargo también puedes crear links simbólicos a los archivos necesarios en el directorio `scratch` de NS-3. A continuación se detallan los pasos para configurar y ejecutar la simulación de boids:

■ **Crear el Archivo Principal del Proyecto (boids.cc):**

- Copia el archivo llamado `boids.cc` en el subdirectorio `scratch` de tu instalación NS-3. Por convención, se hace en `ns-allinone-3.45/ns-3.45/scratch`.
- El contenido de este archivo lo puedes encontrar en repositorio proporcionado anteriormente.

■ **Añadir Modelos de Movilidad:**

- Copia los archivos `boids-mobility-model.cc` y `boids-mobility-model.h` a la ruta `ns3.xx/src/mobility/model`.
- El contenido de `boids-mobility-model.cc` se encuentra en los extractos de "Modelo de Movilidad Boids con Liderazgo Adaptativo".
- El contenido de `boids-mobility-model.h` se encuentra en los extractos de "Modelo de Movilidad Boids para Redes Inalámbricas".

■ **Modificar CMakeLists.txt:**

- Debes modificar el archivo `ns3.xx/src/mobility/CMakeLists.txt`.
- Añade la línea `model/boids-mobility-model.cc` para que se compile.

■ **Recompilar NS-3:** Después de hacer estos cambios, necesitas volver a compilar NS-3 para que los nuevos archivos se integren:

```
./ns3 build
```

O, para usar todos los procesadores:

```
./ns3 build -j$(nproc)
```

■ **Ejecutar la Simulación de Boids:**

- Desde el directorio principal de NS-3 (`ns-3.43`), ejecuta tu proyecto de boids:

```
./ns3 run scratch/boids
```

(Asumiendo que tu archivo se llama `boids.cc` en la carpeta `scratch`).

5. Parámetros y Salida de la Simulación de Boids

■ Modificar Parámetros de la Simulación:

- Puedes ajustar el número de líderes (Cluster-Heads) y miembros directamente en el archivo `boids.cc`.
- Busca las siguientes líneas y modifica los valores según necesites:

```
static const uint32_t N_CH = 2; // Cantidad de lideres (Cluster-Heads)
static const uint32_t N_MEM = 25; // Cantidad de miembros (seguidores)
```

Recuerda recompilar después de modificar `boids.cc` para que los cambios surtan efecto.

■ Generación de Datos:

- El programa `boids.cc` genera un archivo `boids_positions.csv`.
- Este archivo contiene datos de tiempo, ID del nodo, coordenadas X e Y, si el nodo es líder (`IsLeader`), y si la entrada representa un fuego (`IsFire`).

6. Visualización de la Simulación

■ Script de Visualización:

- El archivo `boids_positions.csv` se utiliza con el script de Python `showNodes.py` para visualizar la simulación.
- El script `showNodes.py` lee los datos del CSV y utiliza `matplotlib` para animar el movimiento de los boids y la aparición de fuegos.

■ Características de `showNodes.py`:

- **Boids:** Se dibujan como puntos, con líderes en rojo y seguidores en azul.
- **Fuegos:** Se dibujan como triángulos amarillos [72].
- **Zonas de Influencia de Líderes:** Se representan como círculos rojos con transparencia alrededor de los líderes, con un radio de 100.
- **Controles de Visibilidad:** Puedes activar o desactivar la visualización de fuegos y zonas de líderes mediante botones interactivos.

■ Ejecutar la Visualización:

- Asegúrate de que `showNodes.py` esté en la misma carpeta que `boids_positions.csv`.
- Ejecuta el script de Python: `python showNodes.py`

7. Manual Técnico

7.1. Requisitos para Ejecutar la Simulación

A continuación se detallan los requisitos necesarios para ejecutar la simulación en NS-3 bajo WSL:

■ Entorno linux (Para este ejemplo se uso WSL con Ubuntu 24.04):

- Paquetes esenciales instalados:

```
sudo apt update && sudo apt upgrade -y
sudo apt install build-essential git python3 python3-pip
```

- NS-3 (versión 3.37+): Descargado desde <https://www.nsnam.org/>.

- **Sistema operativo:** Windows 10/11 con **WSL** configurado.
- **Almacenamiento:** 6 GB libres.
- **Python 3.8+** con librerías:

```
pip install matplotlib numpy pandas
```

7.1.1. Configuraciones Adicionales

- Permisos de `sudo` en WSL.
- Variables de entorno (`PATH`) configuradas.

7.2. Estructura del Código

El proyecto consta de los siguientes componentes principales:

- `boids-mobility-model.h`: Definición de la clase `BoidsMobilityModel`
- `boids-mobility-model.cc`: Implementación del modelo de movilidad
- `main.cc`: Configuración y ejecución de la simulación

7.3. Implementación del Modelo Boids

7.3.1. Reglas Boids Implementadas

El modelo implementa las tres reglas clásicas de Boids:

1. **Separación:** Evitar colisiones con vecinos cercanos
2. **Alineación:** Alinear velocidad con vecinos cercanos
3. **Cohesión:** Moverse hacia el centro de masa del grupo

7.3.2. Jerarquía Líder-Seguidor

- Los nodos líderes tienen comportamiento diferente a los seguidores
- Los líderes buscan fuegos en el entorno
- Los seguidores siguen a los líderes y aplican las reglas Boids

7.4. Ejecución de la Simulación

7.4.1. Compilación

```
$ ./ns3 clean
$ ./ns3 configure --build-profile=debug --enable-examples --enable-tests
$ ./ns3 build -j$(nproc)
```

7.4.2. Ejecución

```
$ ./ns3 run scratch/manet_boids --nClusterHeads=2 --nFollowers=10
```

Las variables son opcionales, en caso de no especificarse `nClusterHeads=2` `nFollowers=10` se toman por defecto

7.5. Salida de Datos

7.5.1. Formato del Archivo CSV

La simulación genera un archivo `boids_positions.csv` con el formato:

```
Time,NodeId,X,Y,IsLeader,IsFire  
0.1,1,125.3,87.2,1,0  
0.1,2,120.5,90.1,0,0  
0.1,-1,300.0,400.0,0,1
```

- Time: Tiempo del evento en milisegundos
- NodeId: Id del nodo relacionado al evento, se usa -1 para elementos que no sean nodos
- X: posición en x
- Y: Coordenadas en y
- isFire: Para especificar que es fuego el nodo

8. Experimentación

Para probar la hipótesis, se introduce una **funcionalidad clave** en la Condición Experimental que modifica el comportamiento de los nodos seguidores. Un nodo seguidor (`!m.isLeader`) ahora tiene la capacidad de **autoproclamarse temporalmente como líder** si cumple las siguientes condiciones, evaluadas en cada ciclo de `Update()`:

1. No es un líder actual (`!m.isLeader`).
2. Se encuentra **aislado** (`IsIsolated()`), es decir, sin líderes dentro de un **radio de influencia efectivo ampliado** (`m.leaderInfluenceRadius * 1.2`).
3. Posee un **puntaje WCA** (`wcaScore`) superior a 0.0.

Si cumple estas condiciones, se le asigna el rol de líder (`SetIsLeader(true)`), se le designa un **objetivo aleatorio** (`m.target`) dentro del área de simulación (0–1000 en X e Y), y su **velocidad** (`m.velocity`) se **incrementa ligeramente (x1.05)** para facilitar su búsqueda de fuegos.

Al igual que los líderes iniciales, los nodos autopromovidos siguen participando en el mecanismo de evaluación de liderazgo del WCA. Si se encuentran con un líder con mayor puntaje, pueden **ceder su rol** y volver a ser seguidores.

Escenarios de Experimentación

Para evaluar el impacto de esta propuesta, se comparan los resultados de la **Condición de Control** y la **Condición Experimental (Hipótesis Propuesta)** bajo diferentes configuraciones del número de nodos seguidores. La **efectividad global en la extinción de fuegos** se mide mediante el **número total de fuegos extinguidos** y el **tiempo promedio de extinción**.

Parámetros Comunes para Todos los Escenarios:

- Número de Líderes Iniciales (N Clusters): 2
- Área de Simulación: 1000x1000 unidades
- Radio de Detección de Fuego (`s.fireRadius`): 30.0 metros
- Intervalo de Aparición de Fuegos (`s.fireInterval`): Cada 8 segundos (1 a 3 fuegos por evento, distribución tipo clúster de Thomas)

- **Velocidad Máxima de Líderes:** 6.5 m/s
- **Velocidad Máxima de Seguidores:** 6.0 m/s
- **Duración de la Simulación:** 100 segundos

8.1. Escenario 0:

- **Objetivo:** Evaluar el impacto del número de seguidores en la efectividad de la extinción de fuegos, comparando una condición de control (sin autoproclamación) y una condición experimental (con autoproclamación).
- **Configuración experimental:**
 - **Número de Seguidores (N Members):** 10, 15, 25
 - **Condiciones de Simulación:** Control y Experimental
- **Distribución inicial:** Los nodos se distribuyen aleatoriamente en el área de simulación, con líderes y seguidores asignados según la configuración.
- **Reglas activas:** Los seguidores siguen las reglas de separación, alineación y cohesión del modelo Boids, además de la atracción hacia los líderes. En la condición experimental, los seguidores pueden autoproclamarse líderes si cumplen las condiciones mencionadas anteriormente.

La variación del **número de seguidores (N Members)** es esencial para analizar la robustez y escalabilidad del sistema bajo distintas densidades.

Cuadro 3: Resultados de la Simulación para el Escenario 0

Pruebas	Escenario 0	N. de Seguidores	Condición	Fuegos Apagados	Tiempo Promedio (s)
1		10	Control	15	8.67
			Experimental	16	11.38
2		15	Control	12	17.33
			Experimental	17	17.50
3		25	Control	11	11.88
			Experimental	16	7.81

A continuación, en la figura 5 se presenta una gráfica que permite comparar el comportamiento de los nodos en los diferentes escenarios y condiciones. En la gráfica, se observa cómo el número de nodos seguidores afecta la efectividad de la extinción de fuegos, tanto en la condición de control como en la experimental (que permite evaluar la indagación). Es notorio que la condición experimental mejora el número de fuegos extinguidos en todos los escenarios, especialmente en el escenario 3, donde la diferencia es notable.

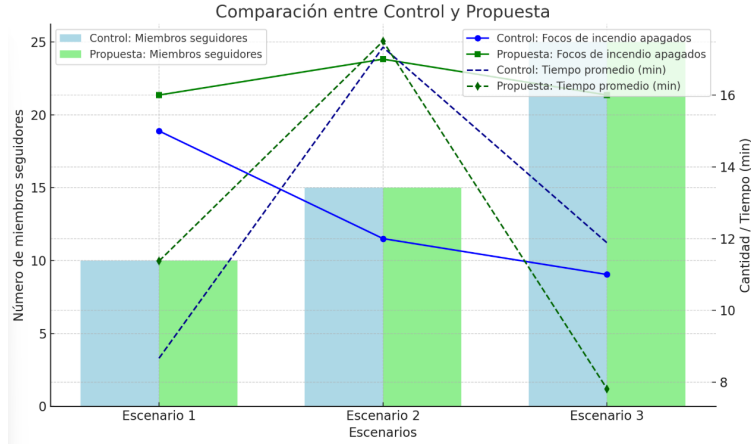


Figura 5: Comparación de la efectividad de la extinción de fuegos entre las condiciones de control y experimental en diferentes pruebas para el escenario 0.

8.2. Escenario 1:

- **Objetivo:** Evaluar el comportamiento de un sistema de movilidad jerárquica basado en el modelo Boids, en un escenario con aparición dinámica de focos de incendio. El sistema integra liderazgo dinámico mediante el algoritmo WCA y comunicación local entre nodos para la formación y mantenimiento de clústeres.
- **Configuración experimental:**
 - **Cantidad de nodos:** 27 (2 líderes iniciales, 25 seguidores).
 - **Área de simulación:** 1000×1000 unidades con frontera toroidal.
 - **Duración:** 200 segundos.
 - **Modelo de movilidad:** BoidsMobilityModel, configurado con cohesión, alineación, separación y atracción hacia objetivos.
 - **Velocidades máximas:** líderes a 10 m/s, seguidores a 7 m/s.
 - **Eventos externos:** focos de incendio generados cada 8 segundos, con radio de aparición de 30 metros y ubicación distribuida según un proceso de Poisson agrupado.
- **Distribución inicial:** Los nodos se agrupan en dos clústeres iniciales, con una desviación normal en torno a los líderes asignados. La asignación de roles y posiciones es aleatoria, controlada mediante semilla de simulación para repetibilidad.
- **Reglas activas:** separación, alineación, cohesión (para seguidores), y atracción a eventos (para líderes). Los seguidores también ejecutan WCA si pierden contacto con un líder, evaluando su potencial de liderazgo según energía, conectividad, movilidad y proximidad a objetivos.
- **Resultados observados:**
 - Los líderes reaccionan rápidamente ante la aparición de focos de incendio, desplazándose de forma directa hacia ellos.
 - Sin embargo, se detectó una dispersión significativa de los seguidores. Muchos seguidores perdieron conexión con su líder tras los primeros movimientos hacia eventos, lo que llevó a una fragmentación de los clústeres.
 - Solo uno de los líderes logró mantener un grupo cohesionado de seguidores hasta el final de la simulación.

- Los seguidores dispersos no lograron reorganizarse de forma efectiva, a pesar de tener habilitado el mecanismo de liderazgo por WCA.

■ **Análisis:**

- La dispersión observada sugiere que el peso de la cohesión en la dinámica Boid puede ser insuficiente frente a trayectorias rápidas hacia eventos.
- El intervalo de actualización de posición y la intensidad de la fuerza de atracción al líder podrían requerir ajuste para favorecer una mayor adherencia de los seguidores.
- Aunque el algoritmo WCA está habilitado, su activación no resultó en una reorganización efectiva, lo que podría indicar que los parámetros de umbral o visibilidad de vecinos son demasiado restrictivos.

- **Visualización:** Se utilizaron trazas de movimiento exportadas a NetAnim para analizar las trayectorias de los nodos. La visualización confirmó la tendencia a la disgregación de los clústeres y la falta de reagrupamiento posterior.

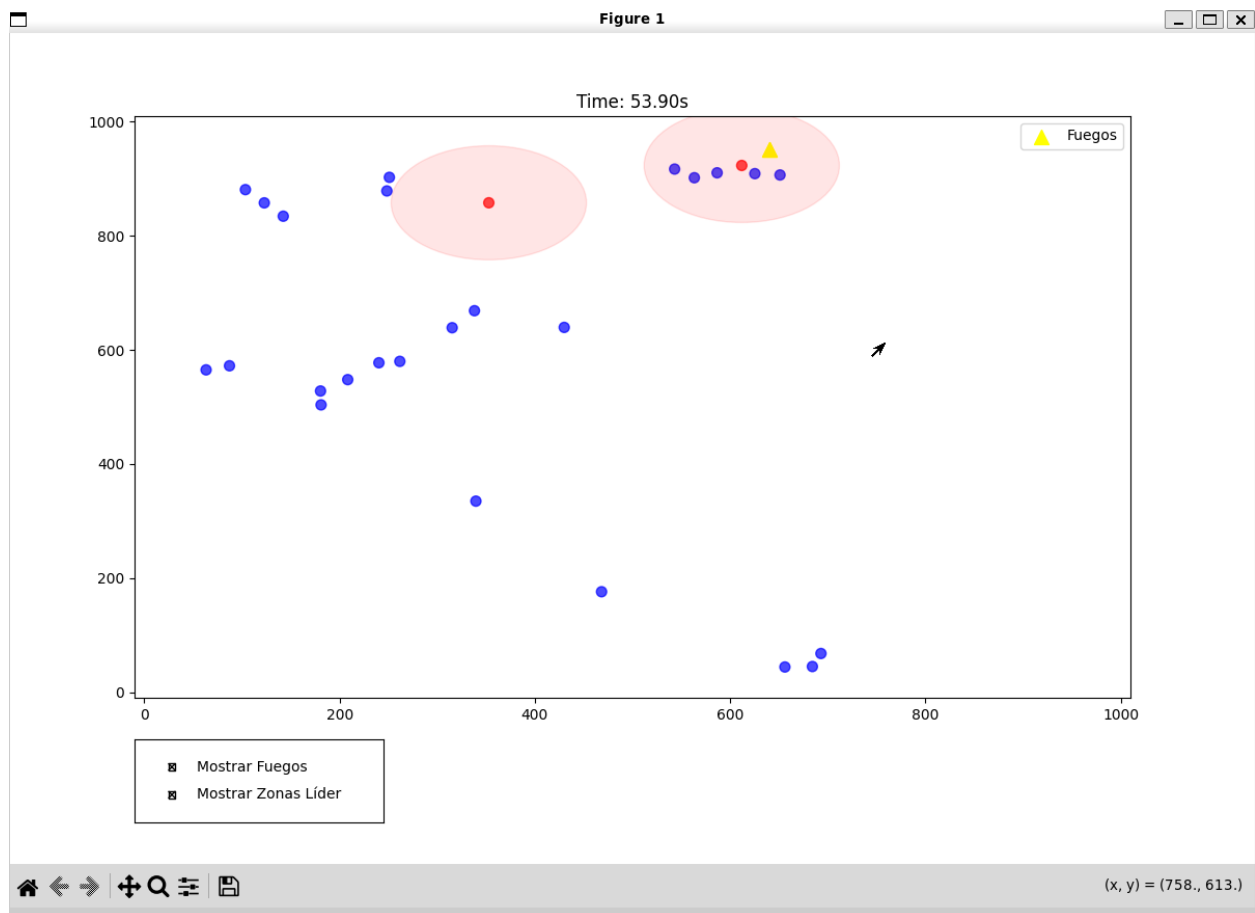


Figura 6: Escenario de simulación 1

Este caso de prueba destaca la complejidad de mantener la cohesión grupal en presencia de eventos dinámicos. A su vez, motiva ajustes en la ponderación de reglas de comportamiento y condiciones de activación del liderazgo distribuido para lograr una movilidad adaptativa más robusta.

8.3. Escenario 2:

- **Objetivo:** Evaluar la efectividad del modelo de movilidad jerárquica basado en Boids en presencia de múltiples líderes iniciales (cuatro), en un entorno con aparición de focos de incendio. El objetivo es observar la formación de clústeres, la estabilidad del liderazgo y la capacidad de los seguidores para mantener la cohesión y reorganizarse dinámicamente.
- **Configuración experimental:**
 - **Cantidad de nodos:** 29 (4 líderes configurados inicialmente, 25 seguidores).
 - **Área de simulación:** 1000×1000 unidades con frontera toroidal.
 - **Duración:** 200 segundos.
 - **Modelo de movilidad:** `BoidsMobilityModel`, con reglas activas de cohesión, alineación, separación y atracción a eventos.
 - **Velocidades máximas:** líderes a 10 m/s, seguidores a 7 m/s.
 - **Eventos externos:** focos de incendio generados cada 8 segundos, con radio de aparición de 30 metros y ubicación definida por distribución de Poisson agrupada.
- **Distribución inicial:** Los nodos se agrupan en cuatro clústeres en torno a los líderes iniciales, distribuidos mediante desplazamientos normales aleatorios. La configuración aleatoria es controlada por semilla para permitir repetibilidad del experimento.
- **Reglas activas:** Separación, alineación, cohesión (para seguidores) y atracción a eventos (para líderes). Además, los seguidores pueden activar el algoritmo WCA si pierden contacto con un líder.
- **Resultados observados:**
 - A pesar de haber configurado cuatro líderes, solo dos se manifestaron efectivamente durante la simulación; los otros no ejercieron liderazgo activo o no lograron mantener seguidores.
 - Se observaron inicialmente múltiples clústeres pequeños, pero rápidamente dos líderes se agruparon, formando un único clúster principal.
 - La mayoría de los seguidores no logró mantener una cohesión prolongada; muchos se dispersaron progresivamente durante la simulación.
 - No se observó una reorganización efectiva de los seguidores dispersos, ni la activación visible de nuevos líderes por WCA.
- **Análisis:**
 - La unificación de líderes sugiere una posible falta de mecanismos de repulsión o diferenciación entre clústeres líderes, lo cual puede conducir a una convergencia no deseada.
 - La poca manifestación de los cuatro líderes podría deberse a conflictos de visibilidad, sobreposición de zonas de influencia o ineficiencia en el mecanismo de activación del liderazgo.
 - La dispersión de los seguidores y la falta de reclustering dinámico indican que los umbrales del algoritmo WCA podrían estar mal calibrados, o que la frecuencia de reevaluación de liderazgo es insuficiente.
- **Visualización:** Se utilizaron trazas en NetAnim para observar la evolución espacial de los nodos. Se confirmó visualmente la fusión temprana de los líderes activos, la dispersión creciente de los seguidores, y la ausencia de nuevos clústeres tras la disgregación.

Este caso de prueba revela la necesidad de mecanismos de estabilización del liderazgo, así como la importancia de sintonizar adecuadamente los parámetros de cohesión y activación de WCA. También sugiere que, en presencia de múltiples líderes, se deben considerar estrategias para evitar colapsos en un solo clúster, preservando la descentralización y cobertura espacial del sistema.

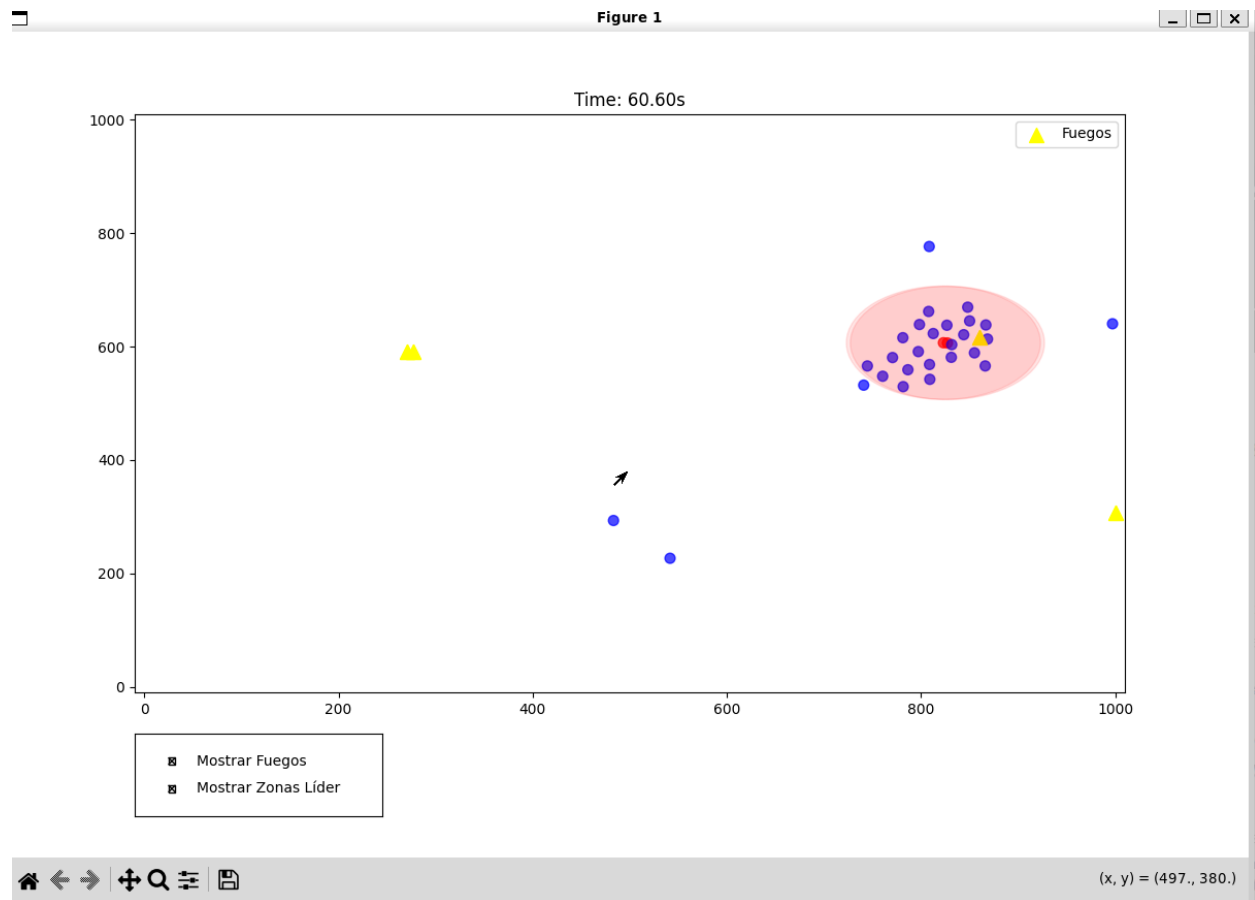


Figura 7: Escenario de simulación 2

9. Conclusiones

- **Liderazgo distribuido adaptativo:** La propuesta desarrollada demuestra que las redes MANET pueden beneficiarse significativamente al incorporar mecanismos que combinen movilidad colectiva con toma de decisiones distribuidas. Al permitir que nodos seguidores asuman roles activos de forma temporal en ausencia de líderes cercanos, se potencia la capacidad de respuesta local sin comprometer la estructura general del sistema. Este enfoque no solo mejora la cobertura en escenarios dinámicos, sino que también reduce la dependencia de la conectividad continua con nodos jerárquicos, promoviendo una mayor autonomía operativa. La interacción entre comportamiento grupal, percepción del entorno y lógica de liderazgo demuestra ser una vía efectiva para enfrentar entornos altamente cambiantes y exigentes como los de emergencia.
- **Realismo en la movilidad gracias al modelo Boids:** La integración del modelo Boids permite simular comportamientos realistas en redes móviles. El uso de este modelo reproduce dinámicas naturales de agrupamiento, separación y alineación, lo cual resulta útil para representar redes MANET en situaciones como búsqueda, exploración o evacuación.
- **Limitaciones observadas en cohesión y liderazgo:** La fuerza de cohesión y la frecuencia de evaluación del liderazgo podrían estar subestimadas. También puede ser necesario un sistema de búsqueda activa de líderes o una mejor gestión del rango de cohesión. Esto se evidencia en los escenarios de prueba, donde muchos seguidores se separaban de su líder a medida que avanzaba la simulación.

El uso del modelo Boids reproduce dinámicas naturales de agrupamiento, separación y alineación, lo cual resulta útil para representar redes MANET (redes móviles ad hoc) en situaciones como búsqueda, exploración o evacuación.

La fuerza de cohesión y la frecuencia de evaluación del liderazgo podrían estar subestimadas. También puede ser necesario un sistema de búsqueda activa de líderes o una mejor gestión del rango de cohesión. Esto se evidencia en los escenarios de prueba, donde se evidenció que muchos seguidores se separaban de su líder a medida que avanzaba la simulación. El modelo de movilidad jerárquica basado en Boids, con liderazgo adaptativo, muestra potencial para mejorar la eficiencia en la extinción de incendios en redes móviles. Sin embargo, se requiere una mayor sintonización de parámetros y una mejor gestión de la cohesión y el liderazgo para optimizar el rendimiento. En este caso, la implementación del modelo Boids con liderazgo adaptativo ha demostrado ser efectiva para simular el comportamiento de nodos móviles en un entorno dinámico. Los resultados obtenidos en los escenarios de prueba sugieren que, aunque el modelo es prometedor, aún existen áreas de mejora, especialmente en la gestión de la cohesión y el liderazgo.

Finalmente, con la indagación de la hipótesis propuesta, se ha observado que la autoproclamación de líderes por parte de los seguidores mejora la efectividad en la extinción de incendios, lo que sugiere que el modelo puede ser adaptado para situaciones más complejas y dinámicas.

10. Recomendaciones

- Aunque el movimiento fue simulado correctamente, se recomienda analizar también cómo este afecta la **calidad de las comunicaciones**, especialmente en entornos con interrupciones externas o desplazamientos abruptos. Esto incluye evaluar métricas como la pérdida de paquetes, la latencia o la variación en el rendimiento de transmisión.
- Se sugiere incorporar un **modelo más avanzado de propagación de señal** en futuras simulaciones, que considere interferencias, obstáculos y degradación dinámica del canal, para obtener resultados más precisos y representativos de entornos reales.
- Es recomendable realizar pruebas con diferentes **tamaños de red** y escenarios más extensos (mayor número de nodos y áreas más grandes) para validar la escalabilidad del sistema propuesto y observar cómo se comporta el liderazgo adaptativo bajo condiciones de mayor carga.

11. Referencias

- [1] S. K. Sarkar, T. G. Basavaraju y C. Puttamadappa, Ad Hoc Mobile Wireless Networks: Principles, Protocols, and Applications, 2^a ed. Boca Raton, FL: CRC Press, 2013.
- [2] M. Barbeau y E. Kranakis, Principles of Ad Hoc Networking, 1^a ed. Chichester, England; Hoboken, NJ: Wiley, May 2007.