BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS DE LA ELECTRÓNICA
MAESTRÍA EN INGENIERÍA ELECTRÓNICA, OPCIÓN INSTRUMENTACIÓN ELECTRÓNICA

Tesis para obtener el grado de

MAESTRO EN INGENIERÍA ELECTRÓNICA

"Diseño e implementación de una red de comunicaciones para el control de formación de vehículos aéreos no tripulados"

Presenta:
Lic. Alejandro García Santiago*

Directores:
Dra. Josefina Castañeda Camacho
Dr. José Fermi Guerrero Castellanos

*Becario CONACYT

Puebla, Pue., Noviembre 2018

Agradecimientos

Agradezco al Consejo Nacional de Ciencia y Tecnología (CONACYT), por las becas de manutención y movilidad otorgada para la realización de este trabajo.

A la Benemérita Universidad Autónoma de Puebla, por mi formación académica y todos los beneficios otorgados como orgullo universitario.

A la Facultad de Ciencias de la Electrónica (FCE), por todas las facilidades brindadas (laboratorios, cursos, congresos y su beca de colegiatura) que hicieron posible la culminación de este trabajo.

A mi directora de tesis Dra. Josefina Castañeda Camacho, por sus incontables horas de asesoría, revisiones, compartir su experiencia e infinita paciencia, pero sobre todo por siempre creer en mi. Gracias.

A mi codirector de tesis Dr. José Fermi Guerrero Castellanos, por compartir su experiencia y su conocimiento.

A mi jurado de tesis Dr. Gerardo Mino Aguilar, M.C. Nicolás Quiroz Hernández y Dr. Richard Torrealba Meléndez, por sus valiosas observaciones para la mejora de mi trabajo.

Dedicado a mi familia

Pedro, Bety y Grecia
Sin ustedes nada de esto sería posible.

Índice general

Índice de figuras

Índice de tablas

Glosario y abreviaciones

AGPS	Assisted Global Positioning System
AODV	Ad-hoc On- Demand Distance Vector Routing protocol
AP	Access Point
Ad hoc	Palabra en latín cuyo significado es “adecuado o apropiado para esto”
BLE	Bluetooth Low Energy
DARPA	Defense Advanced Research Projects Agency
DGPS	Differential Global Positioning System
DSDV	Sequenced Distance Vector Routing protocol
DSR	Dynamic Source Routing protocol
DoF	Degrees of Freedom
FANET	Flying Ad hoc Networks
GPS	Global Positioning System
IMU	Inertial Measurement Unit
IP	Internet Protocol
IoT	Internet of Things
LAN	Local Area Networks
MAC	Medium Access Control
MANET	Mobile Ad hoc Network
OLSR	Optimized Link State Routing protocol
PAN	Personal Area Network protocol

PCB	Printed Circuit Board
PERL	Practical Extraction and Report Language
RERRs	Route Errors
ROS	Robot Operating System
RREPs	Route Replies
RREQs	Route Requests
VANET	Vehicular Ad hoc Network
VANT	Acrónimo en español del inglés UAVs “Unmanned Aerial Vehicles”
VRPN	Virtual-Reality Peripheral Network
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Networks

Introducción

Hoy en día, los VANTs (Vehículos Aéreos No Tripulados también conocidos como drones) representan un área de oportunidad en el sector público y civil, debido al importante papel que estos juegan en situaciones donde la vida del ser humano pudiera correr algún peligro. Actualmente, es muy común encontrar éstos dispositivos y adquirirlos con facilidad en el mercado debido a que el control individual de cada dron es un área que ya se encuentra dominada, facilitando su producción y comercialización.

La comercialización masiva de los drones ha llamado el interés del comercio internacional, haciendo que diferentes consultoras financieras realicen estudios de mercado en relación al área de los VANTs. Por ejemplo, los analistas de ABI Research predicen que el mercado comercial para los pequeños VANTs en México crecerá de un estimado de $652 millones de dólares en el 2014 a más de $5.1 mil millones de dólares para el 2019, llegando a ser el doble de grande que el mercado de defensa militar/civil [1]. Por otra parte, en un estudio similar realizado por analistas de Grand View Research, Inc., se muestra un crecimiento de $552 millones de dólares en el 2014 a $2.07 billones para el 2022 [2].

Éstas prometedoras cifras han hecho más frecuente que las empresas gubernamentales y privadas se encuentren integrando drones en sus procesos productivos. Tal es el caso de empresas como Amazon y HDL, quienes buscan realizar la entrega de sus productos a través de los VANTs para brindar un mejor servicio a sus consumidores [1, 2].

Además, debido a los múltiples usos que pueden tener los VANTs, nuevas líneas de investigación como: control colaborativo y comunicaciones inalámbricas sin infraestructura surgen, donde los protocolos de comunicación entre cada agente (dron) juega un papel fundamental para obtener un buen control del mismo y poder realizar tareas en conjunto. Los sistemas de múltiples vehículos aéreos no tripulados trabajando en forma colaborativa, pueden completar misiones de manera eficiente y en un lapso corto de tiempo comparado con el que se tomaría si solo se utilizara un sólo dron. Sin embargo, existe un problema cuando se quiere implementar el trabajo colaborativo en vista de que se requiere garantizar una buena comunicación entre cada dron que conforma la misión de tal forma que cada uno de ellos debe saber que tarea debe de realizar pero sin perder el área de trabajo ni el estado de sus compañeros.

Debido a que se trata de un sistema de múltiples VANTs volando simultáneamente, será necesario hacer uso de una red inalámbrica. El concepto red inalámbrica no es nuevo en

estos días debido a que han tenido una aceptación muy importante en nuestras vidas cotidianas, en virtud a la movilidad y libertad que nos ofrecen, en consecuencia nuevas tecnologías y protocolos de comunicaciones que permiten mejorar el desempeño de éstas tan demandadas redes han y siguen surgiendo.

En la actualidad, existe un tipo de redes llamadas Ad hoc o MANETs, las cuales ofrecen un red inalámbrica descentralizada, auto organizada y con capacidad de formar una red de comunicaciones que no depende de ninguna infraestructura fija [3].

Asimismo, éste tipo de red está formada por un cierto grupo de usuarios (nodos) distribuidos en cierta región, lo cuales delimitarán el tamaño y el área de cobertura de la red Ad hoc. Cada nodo dentro de la red cuenta con un transmisor y receptor que le permite comunicarse con los otros nodos. De esta manera, es posible realizar envíos y reenvíos de información a través de toda la red mediante el uso de protocolos de enrutamiento, obteniéndose una red con autonomía y flexibilidad en su topología, ideal para establecer una comunicación entre usuarios móviles (drones, computadoras, dispositivos) en lugares donde no se cuenta con infraestructura para montar una red [3].

Por ésta razón, éste trabajo de tesis busca estudiar las redes Ad hoc y los requerimientos de los sistemas VANTs para generar una propuesta que permita patrones de formación mediante una red inalámbrica descentralizada, brindando un aporte tecnológico a este gran nicho de mercado emergente y prometedor.

Objetivos

Objetivo general

Diseño e implementación de una red de comunicaciones para el control de formación de vehículos aéreos no tripulados.

Objetivos específicos

- Estudiar e investigar los principios de funcionamiento de las redes Ad Hoc.
- Analizar las normas y estándares que regulan los sistemas de transmisión de datos a través de redes Ad Hoc y seleccionar y proponer la opción más robusta para la comunicación entre VANTs.
- Simular el funcionamiento del sistema de transmisión de datos entre Vehículos Aéreos No Tripulados utilizando el protocolo seleccionado para tal fin.
- Implementar el sistema de transmisión de datos en una red de Vehículos Aéreos No Tripulados a través de una estrategia de control.
- Probar el sistema de comunicaciones en el control de una rutina de formación de Vehículos Aéreos No Tripulados y su capacidad de operación con respecto a lo reportado en la literatura.

Organización de la tesis

En el capítulo 1 se realiza la revisión del estado del arte, definiciones, conceptos y características de las redes Ad hoc y VANTs. También se incluye una revisión de las tecnologías de comunicaciones inalámbricas actuales herramientas de simulación y la forma en la que se estructura el diseño de la red. En el capítulo 2 se empieza con una breve introducción a lo que es el enrutamiento y su importancia en las redes Ad hoc, eventualmente se profundiza con las técnicas de enrutamiento existentes para este tipo de redes. En el capítulo 3, se plantean los requerimientos de la red a diseñar, se realiza una propuesta de la misma y se evalúan los posibles protocolos de enrutamiento que pudieran ser utilizados. Además, se describen las plataformas y herramientas que se utilizan para la simulación de la red propuesta. En el siguiente capítulo, se muestra el diseño del control colaborativo de múltiples VANTs desde el punto de vista de control (formalismo matemático que permite implementar dicho control), el cual será implementado en el algoritmo de control utilizados en los capítulos 5 y 6. En el capítulo 5, se realiza la simulación de la red propuesta en el simulador de redes ns2 y se diseña una plataforma de simulación que permite evaluar formaciones de VANTs tomando en consideración los sistemas de comunicaciones y el sistema de control. Esta plataforma es desarrollada en Matlab/TrueTime2.0. Finalmente, en el último capítulo se describe una plataforma experimental para pruebas de algoritmos de control de VANTs basado en ROS.

Capítulo 1

Marco teórico

En este capítulo se presentan definiciones y conceptos que se utilizarán a lo largo de la tesis. Se comienza con una breve descripción de los vehículos aéreos no tripulados, los sistemas colaborativos, su importancia y su crecimiento en los últimos años. Además se aborda el estado del arte y la clasificación de las redes Ad hoc.

1.1. Vehículos aéreos no tripulados

Los vehículos aéreos no tripulados también conocidos como drones, son aeronaves que pueden volar sin la necesidad de un operador humano. Este tipo de sistemas han sido muy utilizados hoy en día en áreas civiles y militares donde la vida del ser humano pudiera peligrar. Éstos pueden ser controlados a distancia mediante el uso de radiofrecuencia e incluso a través de un algoritmo computacional que le permita realizar una navegación autónoma. Un dron normalmente se encuentra conformado por los siguientes componentes principales:

- **Chasis o marco.** Se trata del esquelo del dron, el cual puede ser de cualquier material liviano. Dentro de los materiales más populares se encuentra el plástico, el aluminio, la fibra de carbono y el G10 (variación de la fibra de carbono).

- **Motores o actuadores.** Estos dispositivos son los encargados de mover las hélices de los drones permitiendoles volar. Éstos motores son de DC, los cuales deben ser cuidadosamente elegidos de acuerdo a sus hojas de especificaciones para no afectar el rendimiento del dron.

- **Tarjeta de Control.** Está tarjeta puede ser visualizada como una pequeña computadora integrada en una tarjeta de desarrollo. Ésta es la encargada de ejecutar los

algoritmos de control que rigen al dron y a su vez, enviar las señales de control necesarias a los actuadores.

- **Etapa de potencia.** Se trata de un pequeño PCB que permite acoplar la señal de control dada por la tarjeta de control hacia los motores debido a que ésta señal no cuenta con la suficiente potencia para activarlos.

- **Batería.** Éste dispositivo es la fuente de energía que alimenta a todo el sistema.

Si bien éstos son los elementos básicos que conforman un dron, el avance de la tecnología y la escalabilidad de los circuitos integrados con los que hoy en día se cuenta, ha hecho posible el desarrollo de diferentes aplicaciones que suman mas elementos al diseño de los VANTs, los cuales van desde añadir una cámara para realizar alguna exploración hasta estaciones móviles con múltiples sensores para la realización de algún tipo de monitoreo. La figura 1.1 muestra algunos tipos de drones comerciales de diferentes tamaños y formas.

Figura 1.1. Drones comerciales.

1.1.1. Panorama económico de los VANTS

El auge que han tenidos éstos sistemas en la población general trajo consigo diversas oportunidades y desafíos para los VANTs, teniendo como resultado despertar el interés de industrias, comercios y gobiernos para investigar y desarrollar nuevas aplicaciones para estos novedosos sistemas.
Por esta razón, diferentes analistas financieros han realizado estudios de mercado de los VANTs para estimar su crecimiento y buscar nuevos nichos de mercados. Por ejemplo, en un estudio realizado por los analistas de Grand View Research se ilustra el crecimiento significativo en las aplicaciones de los drones sólo en estados unidos. Ellos estiman un crecimiento de $552 millones de dólares en el 2014 a $2.07 billones para el 2022 [2].
La fig. 1.2, muestra el panorama comercial de los drones en USA y las principales áreas de aplicación [2].

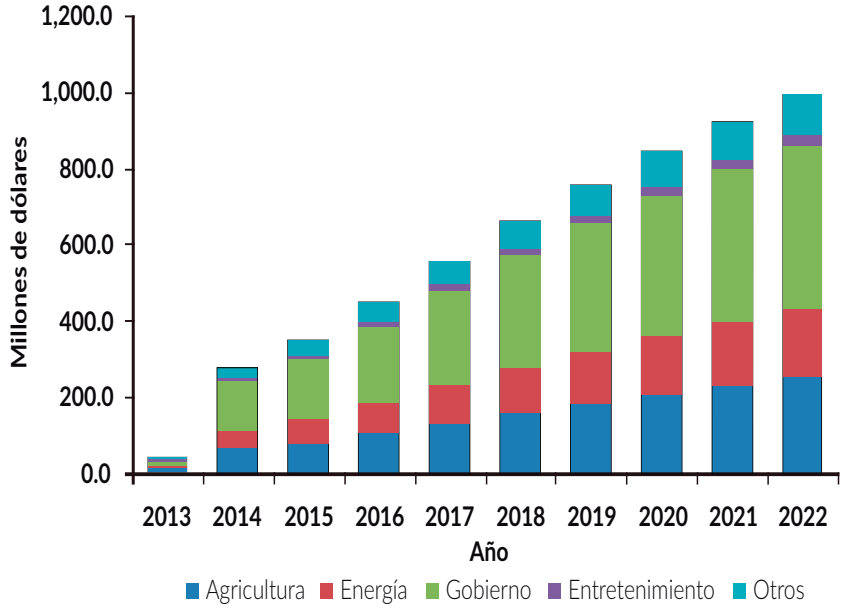


Figura 1.2. Mercado comercial de los drones en USA por área de aplicación (2013 - 2022).

Por otra parte, en un estudio realizado por los analistas de ABI Research se predice que el mercado comercial para los pequeños VANTs en México crecerá de un estimado de $652 millones de dólares en el 2014 a más de $5.1 mil millones de dólares para el 2019, llegando a ser el doble de grande que el mercado de defensa militar/civil [1].

1.1.2. Trabajo colaborativo

Debido a sus diferentes formas y tamaños, los drones pasan a ser un dispositivo versátil e ideal para desarrollar diferentes tipos de actividades como lo son: búsqueda y rescate, exploración, transporte, estación de monitoreo ambiental e incluso entretenimiento.

Sin embargo, en los últimos años gracias a la cantidad de investigación y desarrollo que se tiene sobre ésta área, es posible implementar el uso de múltiples drones en busca de proporcionar una mayor cobertura (véase figura 1.3) y reducir los tiempos en las ejecuciones de la misión que se le haya asignado. Además, otra de las ventajas que presentan este tipo de sistemas es que al tratarse de una red conformada por múltiples agentes (drones), es posible montar una red de comunicaciones entre ellos que les permitirá desempeñarse en lugares hostiles de vuelo, donde comúnmente no se cuenta con alguna infraestructura de comunicaciones, de tal modo que pueden implementarse incluso como estaciones móviles que proporcionen el servicio a cierta área.

Actualmente, este tipo de sistema es implementado mediante una infraestructura que esta integrada por dos partes. La primera se trata de una estación aérea, la cual se encuentra en la tarjeta de control del dron. La segunda es una estación en tierra, cuya función principal es mandar las tareas que realizará el enjambre de drones y recibir los datos generados por éstos al momento de realizar la tarea que le fue encomendada [4]. Por esta razón, la comunicación entre ellos es un factor fundamental. Unas de las redes inalámbricas que más se adapta a las necesidades de este tipo de sistemas son las redes Ad hoc debido a su robustez y adaptabilidad. Las redes VANTs pueden operar en condiciones que van desde las lentamente dinámicas hasta las dinámicas, precisando de enlaces intermitentes y topologías versátiles. Por éste motivo las redes multi-VANTs continúan siendo un tópico que no se ha estudiado con profundidad [5].

Figura 1.3. Enjambre de drones.

1.2. Redes Inalámbricas

Hoy en día, la evolución de la tecnología ha cambiando nuestro estilo de vida, trayendo consigo una dependencia al continuo intercambio de información entre nuestros dispositivos móviles (llámense computadoras, laptops, celulares, tabletas, entre otros); en virtud de la demanda que existe por millones de usuarios que cada día requieren de una red que les permita establecer una conexión para así poder compartir su información.
A pesar de que el concepto de red inalámbrica no es nuevo, éste tipo de redes son las más usadas hoy en día debido a la movilidad y libertad que nos ofrecen a la hora de utilizarlas, por lo tanto, actualmente existe una gran cantidad de investigación en relación a sus protocolos de comunicación de tal forma que cada vez es posible la conectividad entre un mayor número de usuarios (dispositivos móviles) sin que el desempeño de la red se vea afectada por el tráfico de datos.
Por otra parte, las redes inalámbricas son clasificadas de acuerdo a sus objetivos y características particulares [6]. A continuación se mencionarán algunos tipos de redes comunes y las redes Ad hoc, siendo estas últimas las de interés en éste trabajo.

1.2.1. Redes con infraestructura

Este tipo de red es la más popular hoy en día, un claro ejemplo de ellas es el proveedor de Internet o telefonía, el cual permite a sus usuarios establecer una conexión inalámbrica a través de un punto de acceso (AP) tal y como se muestra en la figura 1.4.

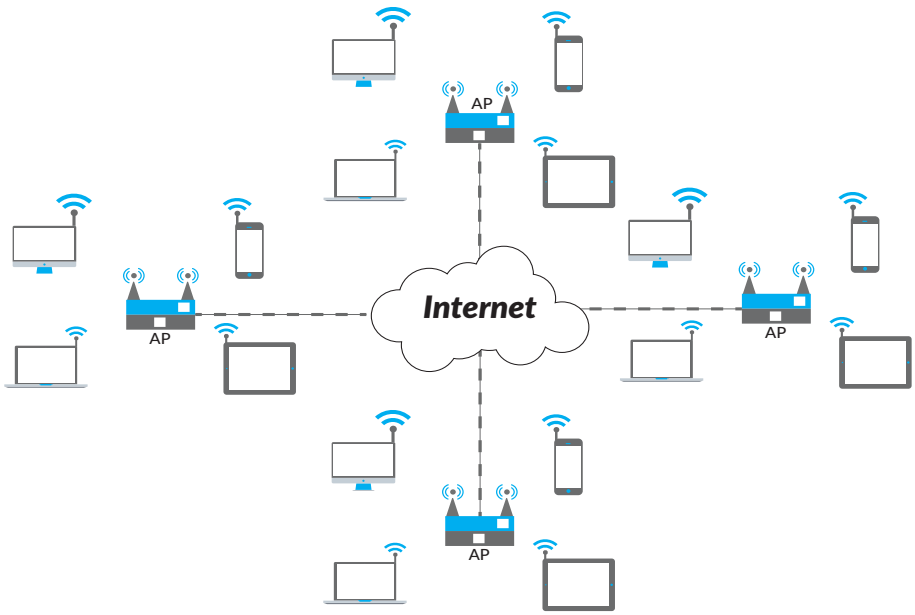


Figura 1.4. Ejemplo de una red con infraestructura.

1.2.2. Redes Ad hoc

Una red Ad hoc es un tipo de red inalámbrica descentralizada, auto organizada y con capacidad de formar una red de comunicaciones que no depende de ninguna infraestructura fija [3]. Éste tipo de red basa su comunicación en el intercambio de información con los usuarios móviles que conformen la red, adicionalmente cada nodo (usuario) dentro de la misma, tiene la capacidad de transmitir y recibir datos. La figura1.5 muestra un panorama general de éste tipo de redes.

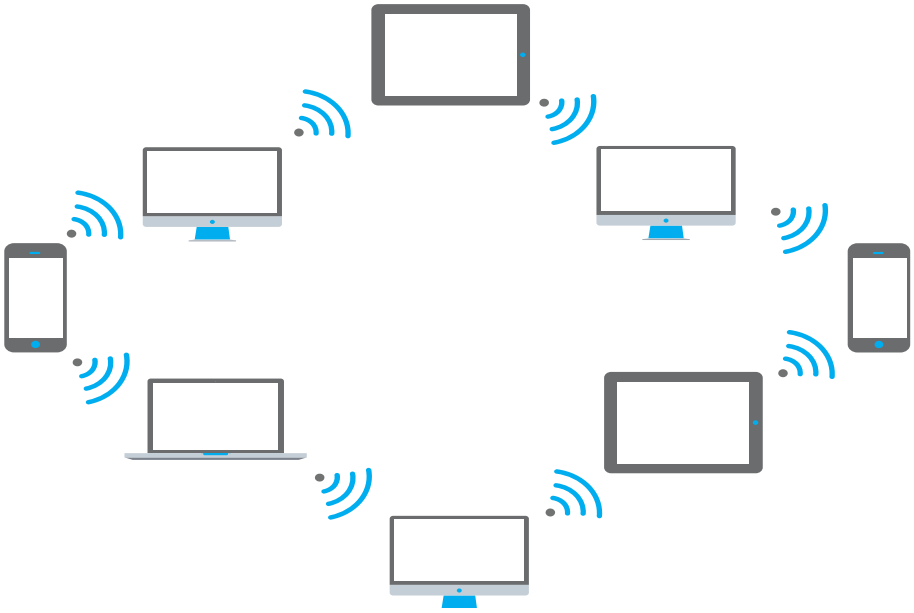


Figura 1.5. Ejemplo de una red Ad hoc.

El funcionamiento de éste tipo de red será profundizado en la sección 1.3.

1.2.3. Redes híbridas

Éstas redes combinan el modo de funcionamiento de las dos anteriores, lo que brinda una red más flexible debido a que pueden lograr una mayor cobertura. La figura 1.6 muestra un ejemplo de una red híbrida, donde es posible observar como el servicio de internet llega a través del AP a los dispositivos que se encuentren dentro de su cobertura, pero al mismo tiempo los dispositivos móviles pueden compartir información entre ellos mismos, lo cual resulta útil en caso de que alguno de estos no pudiese establecer una conexión directa con el AP.

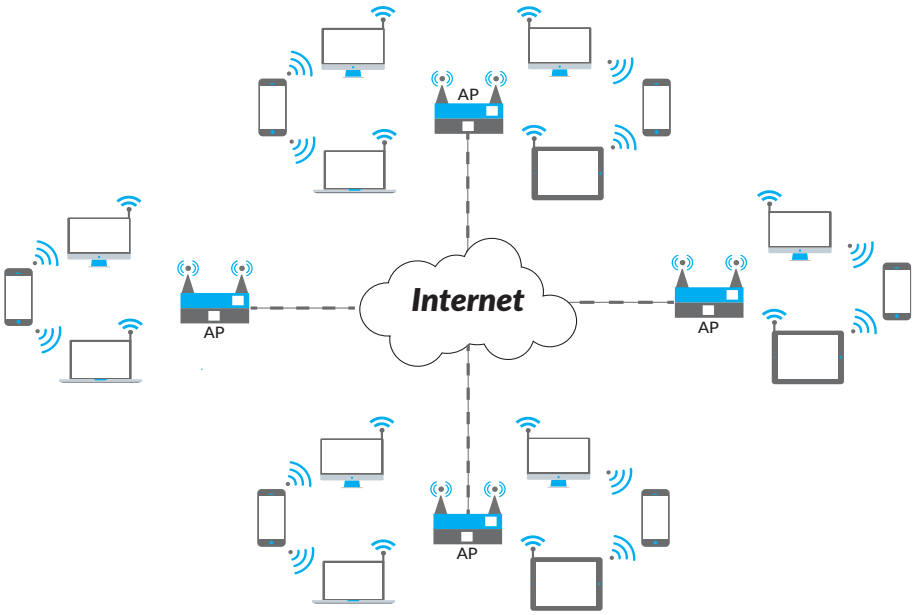


Figura 1.6. Ejemplo de una red híbrida.

1.3. Redes móviles Ad hoc

Las redes Ad hoc surgen alrededor de 1969. Originalmente se les conocía por “radio conmutada por paquetes ” la cual era ocupada con fines militares para interconectar nodos móviles en el campo de batalla, pero no fue hasta principios de los 70ś donde a través del proyecto DARPA se diera a conocer como red inalámbrica Ad hoc [7].
Éste tipo de red está formada por un cierto grupo de usuarios móviles distribuidos en cierta región, los cuales delimitan la cobertura de la red Ad hoc. Cada nodo de la red (usuario) cuenta con un transmisor y receptor que le permite comunicarse con los otros nodos. Además, al tratarse de una red auto organizada, ésta no requiere de una estación base, es decir, cada nodo perteneciente a la red es capaz de generar datos para cualquier otro nodo de la red. De esta forma, todos los nodos pueden funcionar (si es necesario) como estaciones retransmisoras para que los paquetes de datos sean encaminados a su destino final [8].

1.3.1. Elementos que conforman una red Ad hoc

La figura 1.7, muestra el diagrama general de una red Ad hoc, cuyos elementos principales son descritos a continuación.

- **Nodo.** Dispositivo con capacidad de procesamiento o cómputo que participan en la red. Idealmente se busca que cada uno de los nodos ocupe la misma tecnología de comunicación inalámbrica (IEEE 802.11 a/b/g/n wireless LAN and Mesh (Wi-Fi

certification), IEEE 802.15 wireless PAN, IEEE 802.16 acceso inalámbrica de banda ancha (WiMAX certification), IEEE 802.15.4 (ZigBee), etc. [7].

- **Vecino.** Se refiere a los nodos que se encuentran en comunicación directa entre ellos o que se encuentran en un rango de alcance. Normalmente para este tipo de red se estiman asignándoles un grado n, el cual hace referencia al número de saltos desde el nodo actual.

- **Enlace.** Surge cuando dos nodos se encuentran a una distancia específica (dependerá del protocolo de comunicación seleccionado), permitiéndoles intercomunicarse entre sí, a esta acción se le conoce como enlace entre nodos.

- **Distancia.** Hace referencia al número de enlaces involucrados para que la información o paquete llegue a su destino, en este tipo de red la distancia se mide en términos de saltos.

- **Mensaje.** Es la información o paquete que se desea transmitir de un nodo a otro, el tamaño del mismo estará dado por el protocolo de comunicación seleccionado.

- **Enrutamiento.** Es el camino o trayectoria que la información debe de seguir para llegar a su nodo destino, es posible que para llegar de un nodo 1 a un nodo N tenga que pasar por diferentes nodos y enlaces intermedios [9].

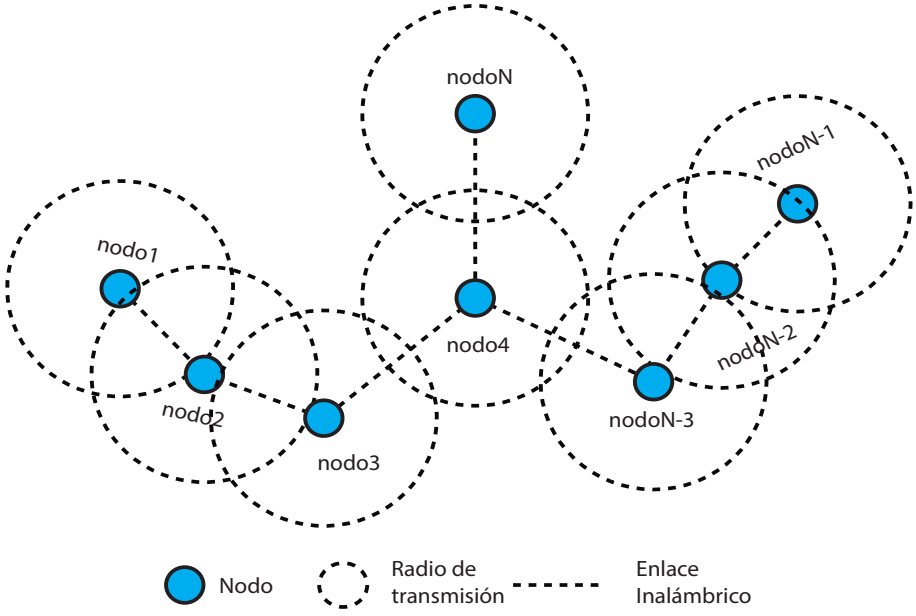


Figura 1.7. Representación de una red móvil ad hoc.

1.3.2. Características de las redes Ad hoc

Las características fundamentales de una red Ad hoc se describen enseguida:

- **Movilidad.** Sus nodos pueden ser rápidamente posicionados y desplegados en áreas sin estructura por lo que el usuario puede formar equipos de trabajo o enjambres para cierta tarea.
- **Multi-salto.** Esta estrategia implica múltiples saltos a través de distintos nodos desde la fuente hasta el destino final, además de brindar la posibilidad de rutas alternativas ante posibles obstáculos, reuso espectral y conservación de energía. Adicionalmente, éste esquema se caracteriza por proveer secuencias de saltos cortos para evitar la detección por parte de agentes externos [7].
- **Auto-organización.** La red determina por sí sola la configuración de parámetros como el direccionamiento, enrutamiento, agrupamiento, el identificador, el control de potencia, etc. Con la finalidad de proveer cobertura y servicio a todos los dispositivos que se encuentran en ella.
- **Conservación de energía.** Implementación de protocolos de ahorro de energía para garantizar el desempeño de la red en un tiempo relativamente grande.
- **Escalabilidad.** El diseño de la red soporta la adición de dispositivos, en función del diseño del manejador de IPs o su estructura jerárquica.
- **Seguridad.** Las redes Ad hoc son vulnerables a ataques de terceros, es decir, si alguna persona llega a introducir un paquete falso a un nodo, podrían desestabilizar el funcionamiento del mismo causando conflictos de operación.
- **Compatibilidad con VANTs.** Ésta es la aplicación más exitosa de estas configuraciones y un área de oportunidad para la investigación.
- **Acceso a internet.** La flexibilidad de las redes Ad hoc no solo las hace idóneas para aplicaciones militares y de investigación, si no que pueden ser ocupadas en áreas comerciales en la nueva era de las comunicaciones llamada IoT.

1.3.3. Tipos de redes Ad hoc

Las redes Ad hoc también conocidas como MANETs pueden ser clasificadas dependiendo de su aplicación, protocolos de comunicación, despliegue y por las tareas que se les haya asignado. Dentro de las clasificaciones principales de las redes Ad hoc se pueden encontrar la red MANET y la red FANET, siendo estas aplicadas a vehículos terrestres y vehículos aéreos respectivamente. A pesar de que estas dos redes tengan aplicaciones diferentes una FANET es un subconjunto de una VANET, la cual es un subgrupo de la MANET, tal y como se muestra en la figura 1.8 [10] .

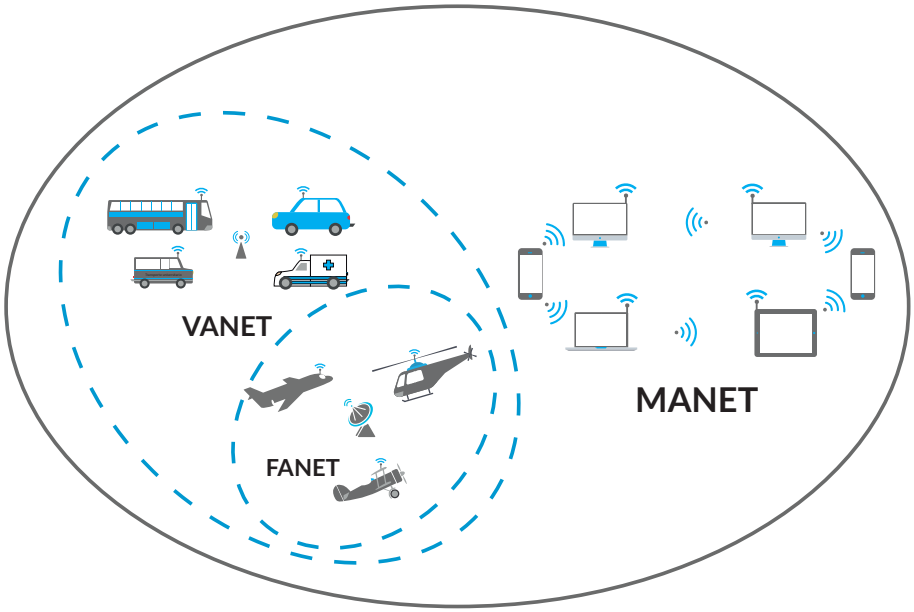


Figura 1.8. Relación de las redes FANET-VANET-MANET.

Si bien la figura 1.8 muestra la relación entre las redes FANET, VANET y MANET, es necesario mencionar que a pesar de que éstas comparten características comunes con las MANETs, cada una de ellas presentan varios desafíos de diseño único [5, 10]. Por esta razón, será necesario profundizar un poco más en las características de cada una las redes antes mencionadas.

En [5] y [10] se hacen comparaciones entre los tipos de redes Ad hoc. La tabla 1.1 muestra un resumen de dichas comparaciones. Siendo la FANET la que más se adecua a este trabajo.

Tabla 1.1. Comparación de las redes FANET,VANET y MANET.

Red / Criterio	MANET	VANET	FANET
Descripción	Nodos móviles inalámbricos conectados con otros nodos usando la red Ad hoc (No centralizada y sin infraestructura)	Los vehículos terrestres son los nodos móviles y la comunicación se realiza entre vehículos y con estaciones de control puestos en las carreteras	Las aeronaves son los nodos móviles y la comunicación se realiza entre VANTs y con estaciones de control terrestres
Movilidad	Baja. Sus velocidades típicas son de de 2m/s	Alta. Sus velocidades típicas van de 20m/s a 30m/s en carretera, mientras que en áreas urbanas de 6m/s a 10m/s	Sus velocidades típicas van de 0m/s a 100m/s esto dependerá de la misión que se encuentre realizando
Topología	Aleatoria	Estación de control en la carretera y una red Ad hoc entre los vehículos	Estación de control central y una red Ad hoc/mesh entre los VANTs
Cambios en la topología	Dinámica, los nodos se unen y dejan la red repentinamente. La red ésta propensa a partición	Más dinámicas que las MANETs. La partición es común	Estacionaria, lenta o rápida, puede desplegarse en enjambres de VANTs. La red ésta propensa a partición
Consumo energético	Protocolos eficientes de energía. Casi la mayoría de los nodos utiliza una batería, por lo cual la energía debe ser conservada	No necesita. Normalmente los dispositivos son alimentados por una batería de carro	Eficiencia energética para mini VANTs debido a que se necesitan baterías pequeñas para no afectar su rendimiento
Poder computacional	Limitado	Alto	Alto
Localización	GPS	GPS, AGPS, DGPS	GPS, AGPS, DGPS, IMU
Areas de Aplicación	Distribucion de información (emergencias, publicidad, compras, eventos,etc.)	Información de trafico y clima, mensajes de alerta, localización basada en servicios	Operaciones de rescate, agricultura, exploración, etc.

1.4. Conmutación de paquetes

Alrededor de los 70's y con la concepción TPC/IP surge una nueva forma de arquitectura llamada conmutación de paquetes, la cual permite transmitir datos digitales a largas distancias. Una red de conmutación de paquetes es un conjunto distribuido de nodos de conmutación de paquetes, los cuales, idealmente conocen el estado de la red completa. Sin embargo, dado que los nodos se encuentran distribuidos en toda la red existe un tiempo de retardo para que estos noten si han surgido algunos cambios dentro de la misma [11]. Las redes de telecomunicaciones basadas en conmutación de circuitos fueron originalmente diseñadas para tráfico de voz, una de sus características es que los recursos se dedican a una llamada en particular, sin embargo, las redes de conmutación de circuitos tienen dos inconvenientes [11] :

1. En una conexión típica de datos la línea se encuentra desocupada la mayor parte del tiempo, por lo que la conmutación de circuitos resulta ineficaz.

2. En una red de conmutación de circuitos la conexión ofrece una velocidad constante dado que emisor y receptor deben recibir los datos a la misma velocidad.

Por ende surge la necesidad de implementar un nuevo tipo de conmutación conocido como conmutación de paquetes. La conmutación de paquetes se realiza mediante el uso de dos técnicas:

- **Datagramas.** Cada paquete se transfiere independientemente sin que exista un proceso de establecimiento de conexión. Los paquetes pueden llegar fuera de orden con retardos grandes.

- **Circuito virtual.** Se fija una ruta previa al envío de algún paquete a través de una negociación. El paquete debe tener un identificador de la conexión a la que pertenece. La finalización de la conexión ocurre cuando se transmite un paquete de liberación (Call Request-Call Accept-Clear Request).

Una vez conocidas las técnicas de conmutación de paquetes, se listan sus ventajas y desventajas en la tabla 1.2.

1.4.1. Tamaño y formato de los paquetes

En general, los datos que se transmiten en una red son paquetes de pequeña longitud, siendo 1000 octetos una longitud típica de éstos.

Tabla 1.2. Ventajas y desventajas de las técnicas de conmutación de paquetes.

	Ventajas	Desventajas
Datagramas	Envío de poca información	Sin orden secuencial
	Robusto a errores en nodos	Lentitud para altos volúmenes
	Robusto a congestión	
	Aplicaciones en tiempo real	
Circuitos Virtuales:	Mucha información	Sensible a congestión
	Mayor Velocidad	Sensible a problemas de nodos volúmenes
	Orden Secuencial	
	Control de errores	

Cada paquete incluirá una parte de los datos para el usuario (o todos), más cierta información de control, la cual, es definida como la mínima cantidad de información que la red necesite para transmitir el paquete a través de ella y alcanzar el destino deseado. Ésta información de control es puesta idealmente, en el inicio de la trama del paquete, siendo este conocido como cabecera. De esta forma, en cada nodo que reciba el paquete, se almacenará temporalmente, se leerá su cabecera y se enviará al siguiente nodo que le permita llegar a su destino (algoritmo de enrutamiento). La figura 1.9 muestra el envío de diferentes tamaños, los cuales pueden ser fragmentos en pequeños paquetes dependiendo de la longitud del paquete que se desee transmitir [11].

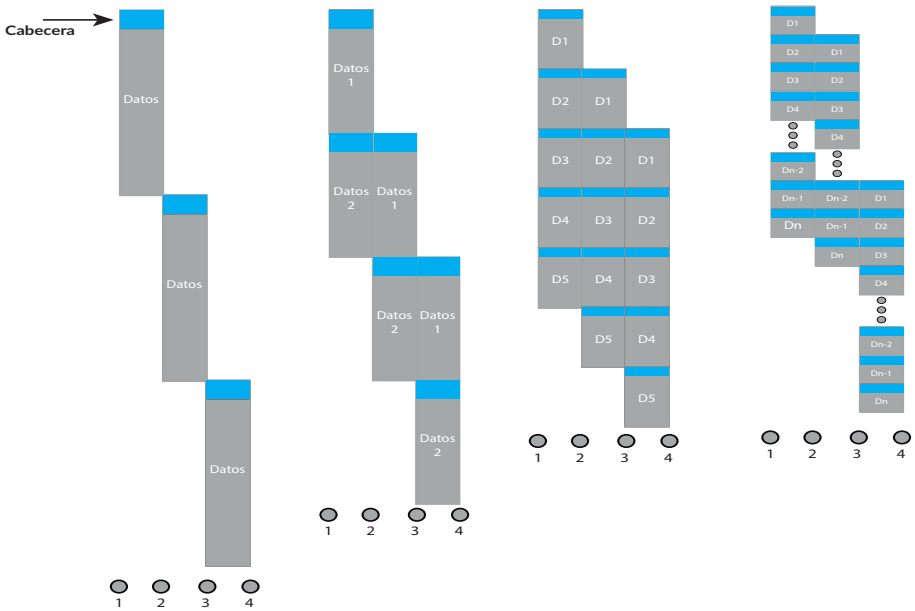


Figura 1.9. Envío de paquetes de diferentes tamaños a través de los nodo.

1.4.2. Comparación de las técnicas de conmutación de paquetes

Una vez conocidos los principios del funcionamiento de la conmutación de paquetes, se puede llevar al cabo la comparación de esta técnica con la de conmutación de circuitos, la figura 1.10 muestra un ejemplo, en el que se ilustra la transmisión de un mensaje a través de cuatro nodos, a los que se asocian tres tipos de retardo.

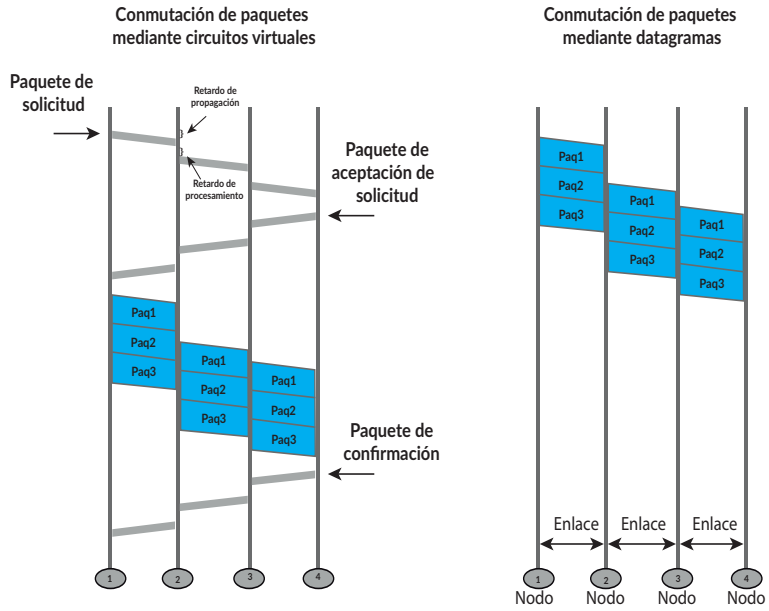


Figura 1.10. Comparación de las técnicas de conmutación de paquetes.

- **Retardo de propagación.** es el tiempo de propagación de una señal desde un nodo hasta el siguiente.

- **Tiempo de transmisión.** es el tiempo necesario para la transmisión de un paquete de datos.

- **Retardo de nodo.** Es el tiempo que necesita un nodo para realizar el procesamiento involucrado en la conmutación de datos.

1.4.3. Comunicación física y Control de acceso al medio MAC

Dado a que las redes Ad hoc son redes de naturaleza inalámbrica, la comunicación entre los nodos se realiza mediante la utilización de ondas electromagnéticas que se propagan en el medio. A causa de que no se utiliza un canal de transmisión dedicado, las ondas electromagnéticas presentan diferentes problemas como: atenuación, interferencias y bloqueo durante la transmisión de la información. Factores como: paredes, ventanas, puertas u otras señales de radiofrecuencia, son algunos que pudieran afectar la transmisión de los datos (al tratarse de propagación de ondas electromagnéticas, éstas son vulnerables a fenómenos como dispersión, reflexión y difracción) [9]. La atenuación que una señal sufre durante su transmisión es definida mediante la relación entre la potencia de la señal transmitida y la potencia de la señal recibida. Ésta relación puede ser afectada debido a la distancia recorrida por la señal e incluso factores a físicos dentro del medio.
Por otra parte, en el caso de la transmisión de información a través de redes inalámbricas, es posible encontrar interferencia que en el peor de los casos genere la pérdida total de la información, en medios compartidos, por lo que los usuarios no pueden transmitir siempre que ellos deseen. Para hacerle frente a este problema será necesaria la implementación de técnicas para controlar el acceso al canal compartido.
En virtud de esta problemática, surge el Control de Acceso al Medio (MAC), el cual define cuando un usuario de la red puede realizar una trasmisión en el medio compartido teniendo como resultado un mejoramiento en la eficiencia de la transmisión. Actualmente, existen varios estándares para la comunicación entre los dispositivos de la Ad hoc como: Bluetooth (IEEE 802.15.1), Ultra-WideBand (UWB, IEEE 802.15.3), Zigbee (IEEE 802.15.4), WiFi (familia IEEE 802.11), WiMAX (802.16) y redes celulares 3G [12]. La tabla 1.3 muestra las características principales de los estándares para las redes inalámbricas más utilizadas. Para la elección del estándar de transmisión de información será necesario tomar ciertos factores.

Capacidad. Para soportar el tráfico de datos deseado.

Fiabilidad. Para satisfacer los requisitos de disponibilidad.

Tipos de datos soportados. Relacionados con el tipo de datos que manejará la red.

El alcance del entorno. Capaz dar una cobertura total al área donde se desempeñará.

La familia IEEE 802.11 es de especial interés en esta tesis y sus características son mostradas en la primer fila de la tabla 1.4 [13, 14].

Tabla 1.3. Revisión de las tecnologías inalámbricas más utilizadas.

	Bluetooth 802.15.1	Wi-Fi 802.11	WiMAX 802.16	Red celular 3G
Ancho de banda típico	21 Mbps	54 Mbps	70 Mbps	384 + Kbps por conexión
Usos típicos	Conexiones entre dispositivos móviles	Conexión de dispositivos móviles a estaciones bases	Conexiones de edificios a torres de comunicaciones	384 + Conexiones entre celulares con torres de comunicaciones

Tabla 1.4. Estándar IEEE 802.11.

Protocolo	Frecuencia de operación	Tasa de datos máxima	Rango
802.11	2.4 a 2.5 GHz	2 Mbps	no especificada
802.11a	5.15 a 5.35, 5.47 a 5.725, 5.725 a 5.875 GHz	54 Mbps	75 metros
802.11b	2.4 a 2.5 GHz	11 Mbps	100 metros
802.11g	2.4 a 2.5 GHz	54 Mbps	75 metros
802.11n	2.4 GHz o 5 GHz bands	54 Mbps	125 metros

1.5. Simuladores de red

La naturaleza dinámica de las redes MANET complica su operación en entornos reales. Las pruebas deben realizarse con un número variable de dispositivos inalámbricos que deben moverse en un entorno con dimensiones ajustables siguiendo uno o varios patrones y con distintos rangos de velocidades. Es por esto que es muy común la utilización de simuladores de red que permiten la experimentación en un entorno controlado. Un simulador de red proporciona modelos del comportamiento de los diferentes elementos de una red de nodos móviles: transmisión en el medio físico, protocolos de encaminamiento, generación de tráfico, movilidad, dimensiones del escenario de prueba, etc [9]. Actualmente, existen varias plataformas que permiten simular el desempeño de una red, sin embargo dentro de los mas populares se encuentran: ns-2, OPNET y GloMoSim.

- **ns-2.** Es un simulador de eventos discretos dirigido a la investigación de redes. Ns proporciona soporte sustancial para la simulación de TCP, enrutamiento y protocolos de multidifusión sobre redes cableadas e inalámbricas (locales y de satélite) [15].

- **OPNET.** Recién adquirido por Riverbed Modeler, es un simulador comercial que consta de un paquete de protocolos y tecnologías con un sofisticado entorno de desarrollo. Permite probar y demostrar diseños tecnológicos antes de la producción; aumentar la productividad de investigación y desarrollo de redes; desarrollar tecnologías y protocolos inalámbricos exclusivos; y evaluar mejoras en los protocolos basados en estándares [16].

- **GloMoSim.** Es otro simulador de red basado también en la utilización de eventos discretos y utiliza un lenguaje de programación paralela denominado Parsec, y desarrollado por el UCLA Parallel Computing Laboratory [9].

Finalmente, la tabla 1.5 muestra las características principales de los simuladores antes mencionados [9].

Tabla 1.5. Características de los simuladores de red.

	ns-2	GloMoSim	OPNET
Lenguaje	C++	C / Parsec	C / C++
Redes inalámbricas	Si	Si	Si
Escenarios	Tcl	Parsec	Interfaz gráfica
Licencia	GNU GPLv2	Uso educativo	Edición limitada
Otros	Más utilizado y bien documentado	Posibilidad de ejecución en paralelo	Simplicidad de uso

Para la parte experimental de ésta tesis, se eligió el simulador ns-2 debido a que es de uso libre y además es posible encontrar mucha información en relación a como emplearlo.

1.6. Descripción del diseño e implementación de la red

En la fig. 1.11 se presenta el diagrama a bloques de la red a diseñar, en donde cada uno de estos bloques será explicado detalladamente en los Capítulos 2, 3 y 5. Se desea diseñar una red de comunicaciones que permita realizar formaciones con VANTs mediante una estrategia de control colaborativo. Para lograrlo, el control necesita que todos los agentes (nodos) dentro de la red realicen un intercambio de información con otros agentes dentro de la red. En base a esto, se busca diseñar una red Ad Hoc con un protocolo de enrutamiento que permita al control establecer una conexión con cualquier agente perteneciente a la misma, aun cuando no se tuviera un enlace directo con él. De esta manera se podrá garantizar que todos los VANTs pueden compartir su información de forma confiable y así lograr su misión (formación deseada). Cabe mencionar que la posición de los agentes será leída a través de un observador externo, llámese GPS, OptiTrack, Marvelminds o cualquier tecnología que permita saber la ubicación de los agentes.

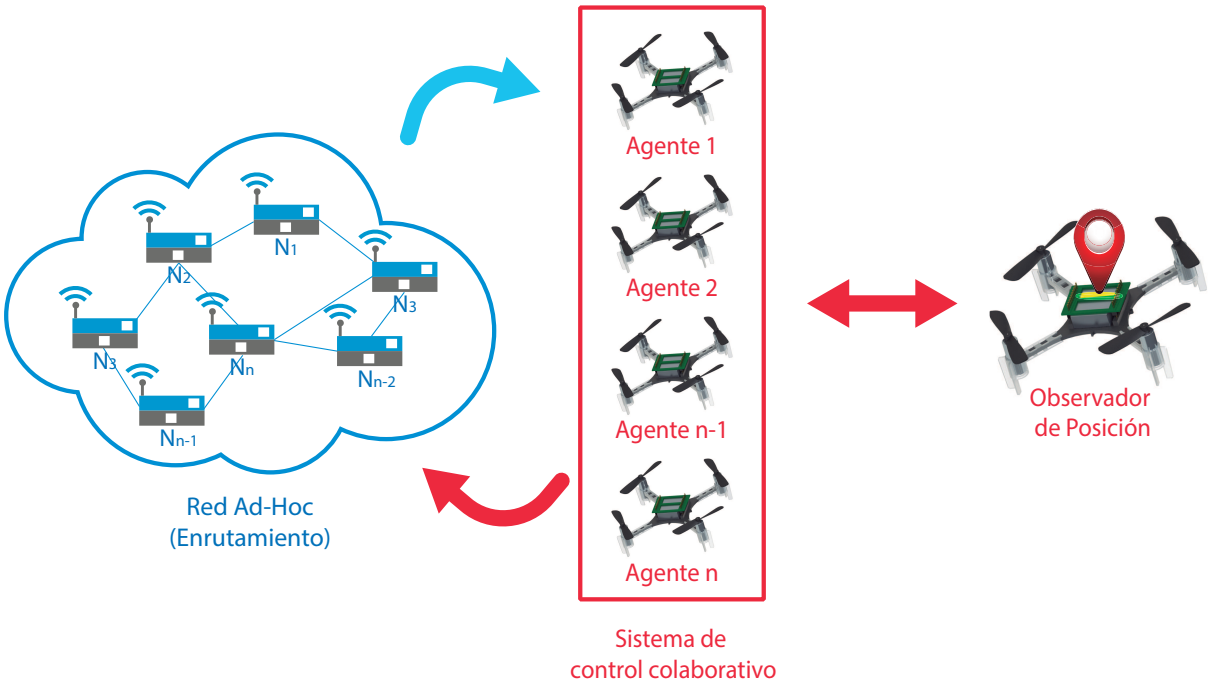


Figura 1.11. Diagrama a bloques de la red a diseñar.

Capítulo 2

Técnicas de enrutamiento en redes inalámbricas

En el presente capítulo se discuten los principales técnicas de enrutamiento utilizadas por las red Ad hoc, siendo estos uno de los puntos más complejos y primordiales en el diseño de redes debido al envío y recepción de información. Este capítulo empieza con una breve introducción a lo que es el enrutamiento y eventualmente da paso al análisis de las estrategias existentes para tal fin.

2.1. Enrutamiento

La función principal de una red de conmutación de paquetes es recibir paquetes de un nodo emisor y enviarlos hacia el nodo destino (Véase 1.4.2). Resulta imprescindible la selección de una ruta, aunque pudiera ocurrir que existan varias o más de una. En virtud de tener más de un camino que cumple con la entrega de los paquetes al nodo destino, surgen los algoritmos de enrutamiento de mínimo costo. Los requisitos del algoritmo de enrutamiento son:

- **Exactitud.** Ser capaz de transferir paquetes de un nodo A a un B.
- **Sencillez.** Más fácil de implementar.
- **Robustez.** Para sobreponerse a los problemas de tráfico y congestión de datos.
- **Estabilidad.** Para evitar variaciones en la definición de la ruta o camino.
- **Imparcialidad.** Que utilice todos los recursos de la red de forma equitativa.
- **Optimización.** Que mejore el desempeño de las conexiones.

- **Eficiencia.** Que implique no desperdiciar tantos recursos de la red sólo los necesarios.

Normalmente, la elección de la ruta o el camino se encuentra basado en algún criterio de funcionamiento. El más sencillo es elegir el camino más corto a través de la red, puesto que minimiza el consumo de recursos. No obstante, con el objetivo de generalizar dicha elección se han propuesto algunos criterios como:

a) El Número de saltos.

b) El Costo (Retardo, desempeño, etcétera).

2.1.1. Estrategias de enrutamiento

Hoy en día, se cuenta un gran numero de estrategias de búsqueda para abordar el enrutamiento en redes de conmutación de paquetes. Muchas de ellas son aplicables también a la interconexión de redes [8, 11]. A continuación se presentan algunas:

- **Enrutamiento estático.** No se requiere actualización de la información a menos que exista un cambio de topología. Se efectúa a través de tablas o matrices de enrutamiento fijas.

- **Método de inundaciones.** En ésta técnica cuando un nodo tiene un paquete que desea transmitir lo manda a todos los nodos vecinos. Los nodos vecinos lo retransmiten a todos los nodos vecinos menos al nodo del cual recibió el paquete de tal forma que el paquete en algún momento llega al nodo destino. Este método se ilustra en la figura 2.1.

- **Enrutamiento aleatorio.** La ruta se elige aleatoriamente o en función de alguna probabilidad.
  La probabilidad se puede basar en la velocidad de datos [11], en cuyo caso se tiene:

$$P_i = \frac{R_i}{\sum_j R_j} \tag{2.1}$$

  donde
  P_i =probabilidad de seleccionar el enlace i.
  R_i =velocidad del enlace i.

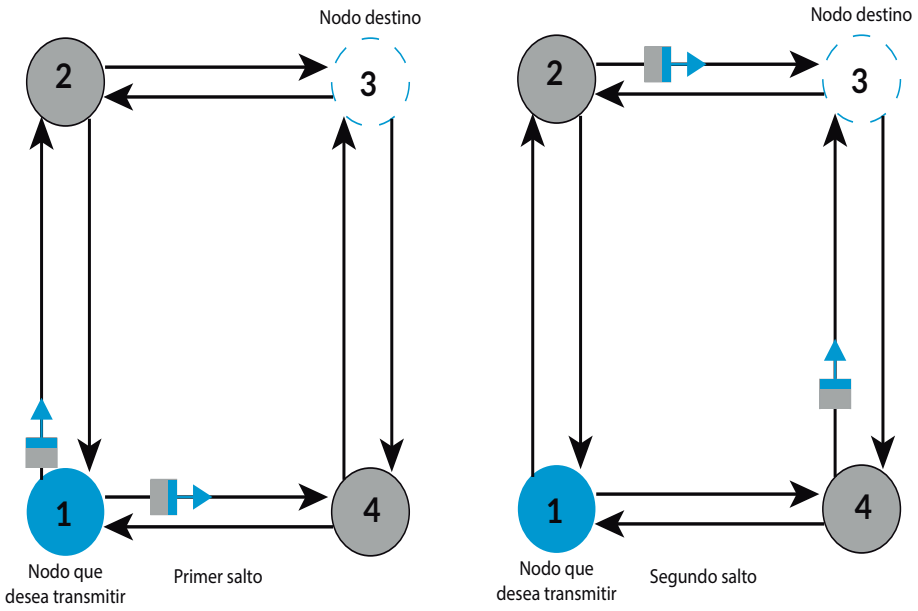


Figura 2.1. Ejemplo del método de inundación.

- **Enrutamiento adaptable.** La ruta se elige en función a fallos o congestión, es decir, la elección de la ruta cambia de manera en que la red cambia. Según [11], las principales condiciones que influyen en las decisiones de encaminamiento son

  Fallos. Cuando un nodo o una línea troncal fallan, no pueden volver a ser usados como parte de una ruta.

  Congestión. Cuando una parte de la red sufre saturación importante, es deseable encaminar los paquetes de forma que se rodee la zona congestionada, en lugar de realizar el encaminamiento a través de ella.

  Por esta razón, es necesario que los nodos intercambien información acerca del estado de la red lo que hace que esta técnica se encuentre presente en casi todas la redes de conmutación de paquetes.

Finalmente, la tabla 2.1 muestra las ventajas y desventajas de las técnicas antes mencionadas.

2.1.2. Algoritmos de mínimo costo

Hasta este punto se sabe que una red se encuentra compuesta por nodos conectados entre sí por enlaces bidireccionales, donde cada enlace tiene un costo asociado en cada sentido. Por lo que el costo total para poder transmitir un paquete desde un nodo origen hasta un nodo destino vendrá dado por la suma de todos los costos de los enlaces por los cuales pasará el paquete antes de llegar a su destino. Por ende, un algoritmo de mínimo

Tabla 2.1. Ventajas y desventajas de las técnicas enrutamiento.

Método	Ventajas	Desventajas
Enrutamiento estático	Simplicidad. Buen funcionamiento en redes fiables con carga estacionaria.	Falta de flexibilidad No reacciona ante fallos ni ante congestión en la red.
Método de inundaciones	Se prueban todos los caminos. Al menos una copia del paquete llega con el menor retardo.	Origina mucho tráfico.
Enrutamiento aleatorio	Menor tráfico, sencillez y robustez que la técnica de inundaciones.	
Enrutamiento adaptable	Optimización de la red. Retrasa la aparición de altos niveles de congestión.	Mayor costo de procesamiento. Congestionamiento.

costo obtiene para cada par de nodos, el camino de mínimo costo.
La mayor parte de los algoritmos de encaminamiento de mínimo costo utilizados en las redes de conmutación de paquetes y en las redes de internet son variantes de uno de los dos algoritmos más comunes: el de Dijkstra y el de Bellman-Ford [11].

Algoritmo de Dijkstra. Se basa en encontrar los caminos más corto o de costo mínimo entre un nodo fuente y los nodos que conforman la red, explorando los caminos en función de su longitud.

Algoritmo de Bellman-Ford. Se basa en encontrar los caminos más cortos o de costo mínimo entre un nodo fuente y los nodos que conforman la red, con la condición de que éstos contengan como mucho un enlace.

Los algoritmos de Dijkstra y Bellman-Ford son explicados en el apéndice A de manera detallada.

2.1.3. Clasificación de los protocolos de enrutamiento

Las estrategias de enrutamiento pueden ser clasificadas de acuerdo a la forma en que realizan su función de encaminamiento, en la literatura podemos encontrar tres tipos de clasificaciones para los protocolos de enrutamiento, lo cuales se describen a continuación:

Proactivos. Cada nodo crea una tabla de todas las posibles rutas para comunicarse con los otros nodos dentro de la red, estas tablas son periódicamente actualizadas mediante el envío de mensajes por la red, que permiten saber si ha habido algún

cambio dentro de ella. De esta manera cada nodo tendrá registradas en todo momento todas las rutas existentes, aunque no las necesite.

Reactivos. También conocidos como: bajo demanda. Crean una conexión entre dos nodos a partir de una petición de alguno de ellos. De esta manera encuentra una ruta del nodo origen hasta el nodo destino. Una vez encontrada esta ruta, ésta se mantendrá hasta que el paquete llegue a su destino o deje de existir. Este tipo de protocolo necesita un proceso de mantenimiento de la ruta [17].

Híbridos. Es la combinación de los dos protocolos antes mencionados.

Finalmente, la tabla 2.2 muestra las ventajas y desventajas de estos dos protocolos.

Tabla 2.2. Ventajas y desventajas de las de los protocolos de enrutamiento.

Protocolo	Ventajas	Desventajas
Proactivo	Acceso a todas las posibles rutas en cualquier momento	Produce más control de trafico Congestionamiento de la red
Reactivo	Eficiencia energética Mantenimiento eficaz de las rutas	Alto tiempo de procesamiento en descubrir las posibles rutas

2.2. Enrutamiento en las redes Ad hoc

Las Redes Móviles Ad Hoc (MANET's) fueron creadas para proporcionar comunicación entre un conjunto de dispositivos, implementando de una manera rápida y eficiente una red transitoria en lugares sin infraestructura de red. Sin embargo, para qué esto sea posible se hace necesaria la introducción en la red de protocolos de enrutamiento propio, debido a que los protocolos tradicionales de redes fijas no se adaptan a este tipo de ambientes móviles [18]. Por esta razón, en [19] podemos encontrar algunos de los protocolos más utilizados para las redes Ad hoc, dentro de ellos están:

Protocolos *Proactivos*.

DSDV. Es uno de los protocolos más recientes para las redes Ad hoc. Introduce secuencias de números que garantizan una ruta precisa para los datos, así como previene bucles dentro de la red.

OLSR. Utiliza algoritmos de caminos cortos, los cuales se basan en una técnica llamada: Multipoint Relays (MPRs).

Protocolos *Reactivos*.

AODV. Este protocolo es el más conocido, el cual utiliza una técnica llamada: Route Request Route Reply (RREQ/RREP), para el descubrimiento de las rutas y utiliza una secuencia de números mientras transmite la información, como lo hace el DSDV.

DSR. Este utiliza la misma técnica que el AODV para el descubrimiento de rutas, sin embargo el DSR almacena las rutas en una cache.

En las referencias [20–30], se realizan estudios de estos cuatro protocolos, mostrando que DSDV y AODV presentan buenos resultados en cuanto envíos y retardos en la entrega de datos. Por consiguiente, en este trabajo ambos serán considerados como posibles protocolos de enrutamiento y se describen a continuación.

2.2.1. DSDV

DSDV es un protocolo proactivo basando en el algoritmo de Bellmann-Ford (véase apéndice A). Este realiza un enrutamiento de nodo a nodo a través de un vector de distancia, donde cada nodo integrado a la red mantiene una tabla de enrutamiento con los siguientes cuatro campos principales: *Destino*, *Siguiente salto* , *Distancia (costo)* y *Numero de secuencia*. El llenado de esta tabla se realiza mediante un intercambio de información (mensaje de actualización) entre cada nodo y sus nodos vecinos [31, 32].

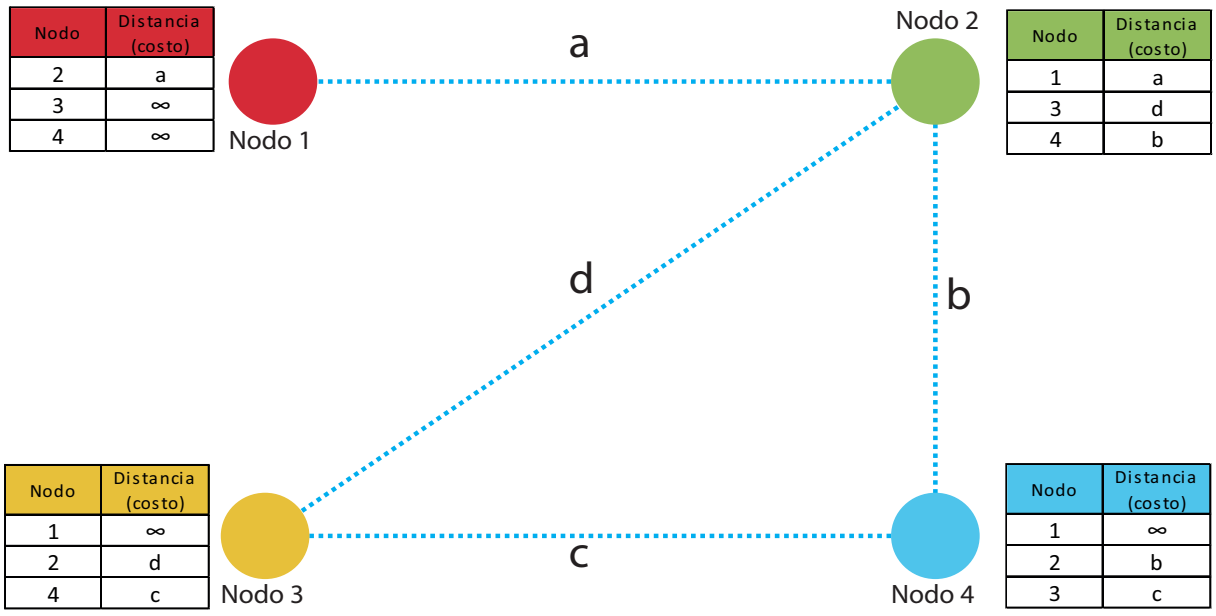


Figura 2.2. Mensajes de actualización del protocolo DSDV.

La fig. 2.2 muestra el mensaje de actualización que los nodos envían a sus vecinos, inicialmente se considera que todos los nodos conocen con que nodos se encuentran conectados,

el costo de la conexión y el número de nodos que conforman la red. Cuando no existe un enlace entre nodos, la distancia (costo) es tomada como ∞.
Una vez que los nodos transmiten su mensaje de actualización a todos sus nodos vecinos, cada uno de ellos genera su tabla de enrutamiento aplicando el algoritmo de mínimo costo para determinar la ruta optima (véase fig. 2.3). Debido a la movilidad de los nodos o una perdida en los enlaces, la tabla de enrutamiento puede cambiar conforme pasa el tiempo. Por esta razón, la actualización de sus datos debe realizarse frecuentemente para mantener la tabla de enrutamiento con los últimos cambios y garantizar un envío de los paquetes confiable.

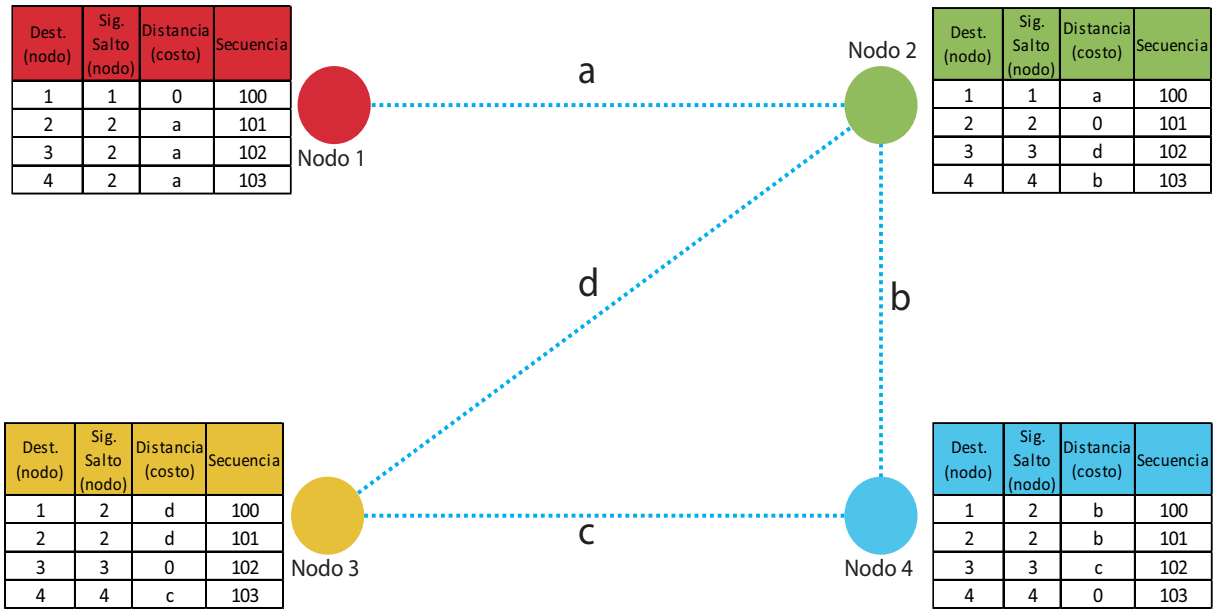


Figura 2.3. Tablas de enrutamiento generadas en cada nodo.

2.2.2. AODV

AODV es un protocolo reactivo basado en los protocolos DSDV y DSR. Este busca y establece una ruta temporal entre un nodo emisor y receptor solo cuando esta es requerida (bajo demanda), de esta forma logra reducir el tiempo de obtención de una ruta y disminuir el congestionamiento de los canales que se tienen en DSDV y DSR. El AODV opera bajo dos rutinas: rutina de descubrimiento de rutas y rutina de mantenimiento [32, 33].
La rutina de descubrimiento de la ruta ejecuta un algoritmo basando en mensajes de solicitudes y respuestas (*RREQs/RREPs*). En esta, el nodo emisor envía un mensaje *RREQs* indicando el nodo destino, el número de la secuencia y la cantidad de saltos del mensaje (inicialmente este valor es de 1). Por otra parte, los nodos que reciben dicho mensaje deciden reenviarlo (si no es el nodo destino) o quedárselo (si es el nodo destino). Dado que el mensaje *RREQs* puede llegar por múltiples rutas al nodo destino, este aplica un algoritmo de mínimo costo para establecer la ruta más corta. Una vez obtenida esta

ruta, el nodo destino envía un mensaje *RREPs* a través de esta para indicarle de su existencia a los nodos que la forman. La fig. 2.4 muestra un ejemplo de una rutina de descubrimiento para un envío de nodo 1 al nodo 3, asumiendo que no existe un enlace directo entre ambos.

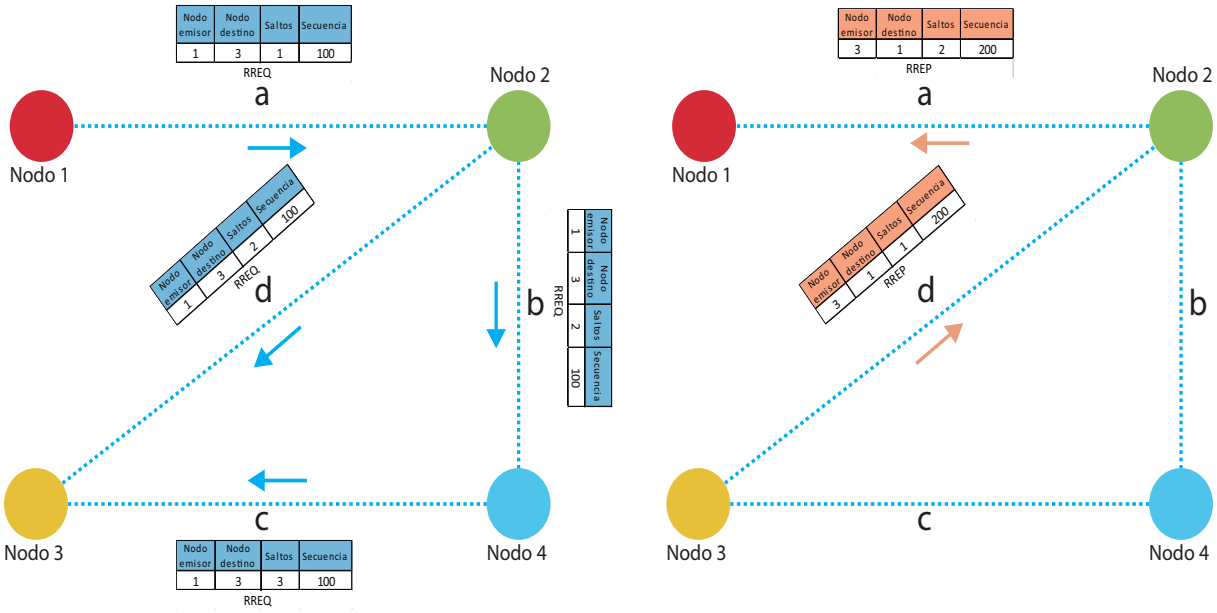


Figura 2.4. Rutina de descubrimiento de la ruta

Al termino de la rutina de descubrimiento cada nodo habrá generado su tabla de enrutamiento tal y como se muestra en la fig.2.5.

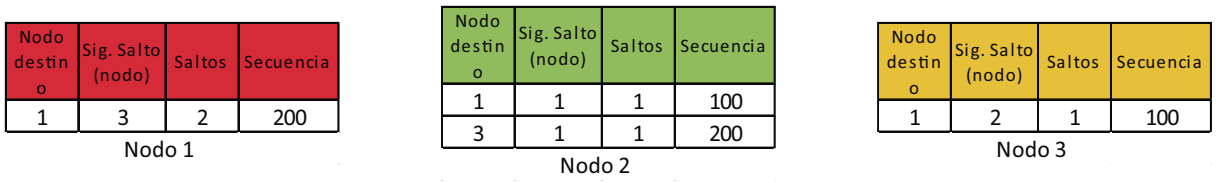


Figura 2.5. Tablas de enrutamiento generadas por cada nodo perteneciente a la ruta establecida.

Para el caso de la rutina de mantenimiento, cada nodo envía periódicamente el mensaje de *Hello* a sus vecinos. Esto con la finalidad de detectar si algún nodo ha sido dado de baja o se ha perdido la conexión entre ellos; en caso de ocurrir alguno de estos eventos el nodo envía un mensaje *RERR* para informar a los otros nodos de la red y deliberar si es necesario ejecutar la rutina de descubrimiento nuevamente.

Capítulo 3

Propuesta y métodos de evaluación de una red Ad hoc para VANTs

En éste capítulo se presentaran los requerimientos de los VANTs, el tipo de red Ad hoc a diseñar y el protocolo de enrutamiento que se utilizará. También, se muestran las diferentes métricas que existen para evaluar una red Ad hoc y las dos plataformas de simulación (ns-2 y Matlab/Simulink) donde se evaluará la red.

3.1. Definición del problema

La figura 3.2 muestra el panorama general de lo que se trabajará en este proyecto; el cual busca crear una FANET que permita la comunicación de cuatro VANTs cuya tarea es realizar formaciones.
La información que se trasmitirá a través de la red es la posición obtenida de un GPS (x, y, z) y las velocidades lineales (v_1, v_2, v_3), las cuales son dadas por el control del dron (véase figura 3.1).
Los datos deben de transmitirse a una frecuencia establecida por el operador (mínima de 50 Hz o 20 ms máximo) para cumplir con las condiciones del control disparado por eventos que utilizan los drones.

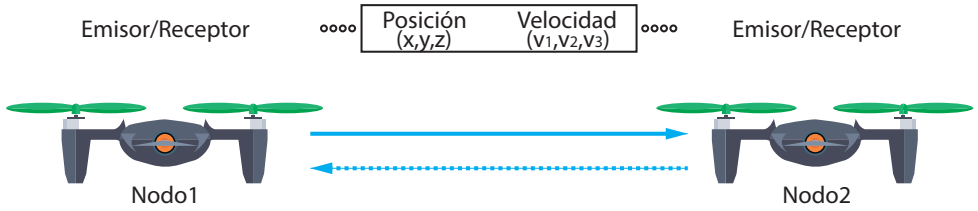


Figura 3.1. Comunicación entre dos VANTs.

La red mostrada en la fig. 3.2, será de tipo FANET donde los drones tendrán el papel de

nodos. Haciendo uso de las prestaciones de la red, estos podrán tener acceso a cada uno de los nodos que integren la red, sin importar que un dron (nodo) se encuentre lejos del nodo actual, es decir, que se encuentre fuera del radio de trasmisión del nodo emisor (el 802.11 brinda hasta 300 m teóricos de cobertura) y sea imposible establecer un canal de enlace directo.

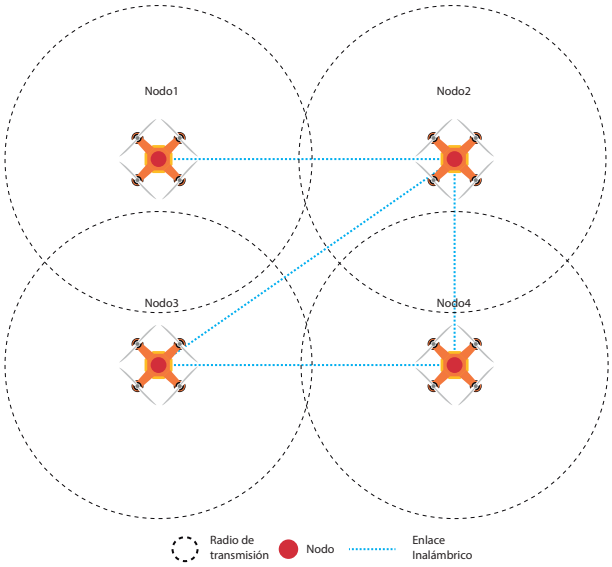


Figura 3.2. Implementación de una FANET para cuatro VANTs.

Por otra parte, la figura 3.3 muestra la arquitectura jerárquica básica de dos capas que describe la composición automática de servicios que caracterizan a la red FANET. Donde la tecnología que se ocupa para el control de acceso al medio es el estándar 802.11 discutido en 1.4.3.

Figura 3.3. Arquitectura de la composición automática de servicios de la FANET.

Para la selección del protocolo de enrutamiento se tomaron en cuenta las referencias [20–30], que muestran diferentes comparaciones de los protocolos de enrutamiento más utilizados para las redes Ad hoc. Por consiguiente, se ha optado por el uso de los protocolos: AODV y DSDV debido a que ambos presentan un buen rendimiento (estos dos protocolos son evaluados en el Capitulo 5 para la red propuesta).

3.2. Métricas de evaluación de una red Ad hoc

Actualmente en la literatura es posible encontrar numerosas comparativas entre los protocolos de enrutamiento. En las referencias [23–30] se muestran diferentes tipos de métricas utilizadas para valorar las redes Ad hoc. En esta sección se presentan las métricas más utilizadas para medir el desempeño de los diferentes protocolos.

- **Tasa de paquetes entregados (%).** Típicamente conocido como PRD (Packet Delivery Rate), muestra el éxito de un protocolo para entregar paquetes desde un nodo emisor hasta el nodo destino. Esta relación puede ser calculada como:

$$PDR = \frac{R_p}{S_p} * 100 \tag{3.1}$$

  donde R_p es el total de paquetes recibidos y S_p es el total de paquetes enviados.

- **Retardo extremo a extremo (s).** Comúnmente conocido como E2ED (End-to-End Delay), calcula los retardos ocurridos en la red y se define como el tiempo que demora un paquete de datos transmitidos en llegar a su destino a través de la red FANET. Esta métrica puede definirse como:

$$E2ED = T_R - T_S \tag{3.2}$$

  donde T_R es el tiempo en el que se recibió el paquete y T_S es el tiempo en el cual fue enviado.

- **Desempeño (bps).** También conocido como Thp (abreviación de Throughput), calcula el número de bits por segundo que llegaron al nodo destino. Esta métrica es utilizada como una medida de confiabilidad del protocolo bajo diferentes condiciones, por esta razón el throughput en la red debe ser lo mas alto posible. El Thp puede definirse como:

$$Thp = \frac{N_p * P_z * 8}{Seg} \tag{3.3}$$

donde N_p es el numero de paquetes recibidos exitosamente y P_z es el tamaño de los paquetes.

3.3. Plataformas de simulación

3.3.1. Simulador ns-2

Es un simulador de eventos discretos dirigido a la investigación en redes, el cual proporciona soporte sustancial para la simulación de TCP/IP, enrutamiento y protocolos de multidifusión a través de redes cableadas e inalámbricas (locales y de satélite) [15]. El simulador ns-2 se compone de dos partes: animación y análisis (véase figura 3.4). La animación permite observar el comportamiento de la red a través del tiempo de manera gráfica, mientras que el análisis guarda información especifica de los eventos realizados en la red durante su tiempo de simulación en un archivo llamado *trace file* (.tr). Este contiene una gran cantidad de datos como: tamaño de los paquetes que se envían, tiempo en los que envía, recibe e incluso cuando se pierden paquetes o se pide retransmisión.

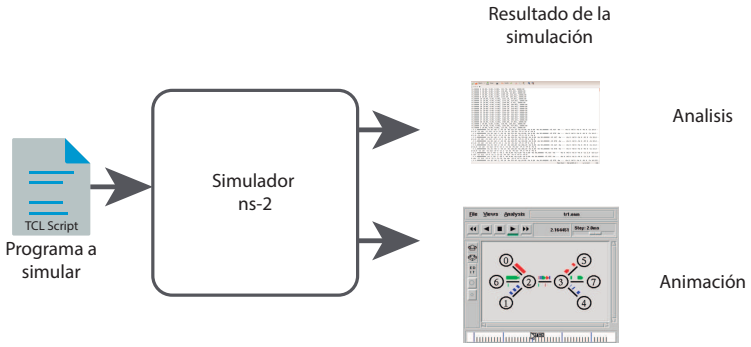


Figura 3.4. Simulador ns-2.

3.3.1.1. Archivo de rastreo (*trace file*)

Una vez ejecutado el programa a simular (TCL script), el simulador ns-2 crea un archivo llamado trace file (.tr) donde se almacena un análisis detallado de todos los eventos ocurridos en la red durante el tiempo de simulación. El archivo .tr debe ser claramente comprendido debido a que a partir de sus datos se realizan los análisis de la red. La tabla 3.1 muestra el formato del archivo *trace file* para una simulación de red inalámbrica [15, 34].

Tabla 3.1. Estructura del archivo *tracefile*

s	0.045678905	_1_	RTR	- - -	0	message	32	- - - - - - -	[1:255 -1:255 32 0]

donde los campos que conforman este formato son:

Acción: s – Envío, d – Removido, r – Recepción

Tiempo: El tiempo en el cual la acción es realizada

Nodo: Nodo donde pasa la acción

Capa: AGT – Aplicación
RTR – Enrutamiento
LL – Capa de enlace
IFQ – Cola de paquetes salientes (entre la capa de enlace y la capa mac)
MAC – MAC
PHY – Físico

Banderas: - - -

Secuencia: Indica el numero de secuencia del paquete

Tipo: Información sobre el tipo de paquete (mensaje, cbr, tcp, ack o udp)

Tamaño: Indica el tamaño del paquete

[- - - -]: Información de la capa MAC
1 – Duración del paquete en el encabezado de la capa MAC
2 – Dirección MAC del destino
3 – Dirección MAC de la fuente
4 – Tipo de MAC del cuerpo del paquete

Banderas: - - -

[- - - -]: Información de la capa MAC
1 – IP del nodo origen : n° de puerto
2 – IP del nodo destino (-1 = broadcast) : n° de puerto
3 – IP del encabezado ttl
4 – IP del próximo salto

TCP: Los últimos dos campos son información tcp: su número de secuencia y el número de acuse de recibo.

3.3.1.2. Interpretación de los *tracefile*

Los trace files proporcionan los datos detallados sobre los eventos que ocurren en la red, sin embargo, si se desea realizar un análisis será necesario extraer información relevante de la gran cantidad de datos que almacenan estos archivos. La extracción de estos datos

se realiza a través de cualquier lenguaje de programación que pueda manejar archivos de datos. Afortunadamente, existen varias herramientas utilizadas para estos fines y son de uso libre para varios sistemas operativos (Windows, Linux, etc.) [15, 34]. A continuación se mencionan algunas de las más utilizadas.

3.3.1.3. AWK

Es un lenguaje de programación diseñado para buscar lineas de archivos (u otras unidades de texto) que contengan ciertos patrones. Cuando una línea coincide con uno de los patrones, *awk* realiza acciones especificadas en esa línea. Esto permite realizar operaciones simples en archivos de datos, como: promediar los valores de una columna determinada, sumar o multiplicar el término por medio de varias columnas, todas las tareas de reformateo de datos, etc [15, 35].
Los programas *awk* se diferencian fácilmente de los lenguajes convencionales, debido a que estos están basados en datos (se describen los datos con los que se desea trabajar y qué hacer cuando los encuentra). Por otra parte, en los lenguajes convencionales es necesario describir cada paso que debe dar el programa (lenguajes de procedimiento). Cuando se trabaja con un lenguaje convencional generalmente es mucho más difícil describir claramente los datos que procesará el programa. Por esta razón, los programas de *awk* son a menudo más fáciles de leer y escribir [35].
Los scripts *awk* pueden ejecutarse de la siguiente manera:

awk -f script.awk sim.tr > outfile

donde, $sim.tr$ es el trace file generado de la red simulada y $outfile$ es el archivo que guarda los datos generados por el script en el *awk* programado.

3.3.1.4. grep

Grep es un comando de $UNIX$ que permite filtrar un archivo. Con este comando es posible crear un nuevo archivo que contenga solo las lineas que coincidan con una cierta secuencia de caracteres determinadas por el programador. Por ejemplo, si se desea conocer los paquetes del tipo *tcp* que pasaron del nodo a al b, la cadena de caracteres estaría dada por “a b tcp ”y se escribiría de la siguiente manera:

grep “a b tcp ” tr1.tr > tr2.tr

donde, $tr1.tr$ es el trace file original y $tr2.tr$ será el nuevo trace file, el cual solo contiene las lineas que coincidan con la cadena de caracteres establecida.

3.3.1.5. Extracción práctica y lenguaje de informe ("Practical Extraction and Report Language"(PERL))

Es un lenguaje de interpretación optimizado para escanear archivos de texto arbitrarios, extraer información de esos archivos de texto e imprimir informes basados en esa información [36]. Por otra parte, su fácil uso y su constante evolución lo han convertido en un lenguaje de propósito general y una de las herramienta más utilizadas para los datos web e internet [15]. El script *perl*(*.pl*) puede ejecutarse de la siguiente manera:

perl thp.pl tr1.tr a b >thp

donde, *thp.pl* es el script a ejecutar, *tr*1.*tr* es el trace file original y *a* y *b* son variables (nodo, tiempo, evento, etc.) que el script necesite saber para ejecutar su análisis.

3.3.2. Simulador TrueTime 2.0

TrueTime 2.0 es una herramienta de Matlab/Simulink para sistemas de control en tiempo real desarrollado por el departamento de control automático de la Universidad de Lund en Suecia. TrueTime permite la co-simulación de tareas de un controlador a través de kernels en tiempo real, transmisiones de red y dinámica continua de la planta. Las características del simulador incluyen [37]:

- Simulación de temporización compleja del controlador gracias a la ejecución de código, programación de tareas y comunicación de redes cableadas/inalámbricas.
- Posibilidad de escribir tareas mediante archivos M o funciones C ++. Asi mismo, es posible realizar llamados a los diagramas de bloques de Simulink desde dentro de las funciones del código
- Bloque de red (Ethernet, CAN, TDMA, FDMA, Round Robin, Ethernet conmutado, FlexRay y PROFINET)
- Bloque de red inalámbrica (802.11b WLAN y 802.15.4 ZigBee)
- Dispositivos con batería, escala dinámica de voltaje y relojes locales

La fig 3.5 muestra la librería con la que cuenta TrueTime 2.0. En este trabajo se utilizaran los bloques *TrueTime Kernel* y *TrueTime Wireless Network* los cuales se describen a continuación.

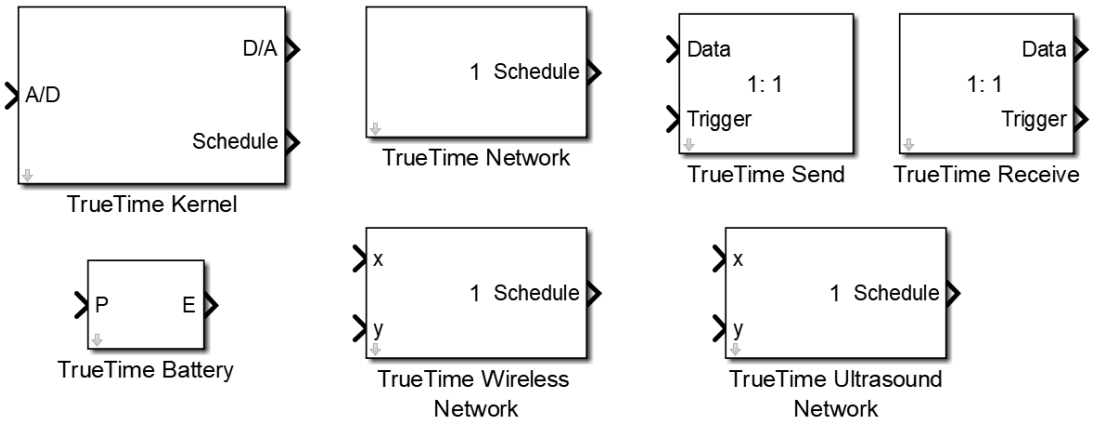


Figura 3.5. Librería de TrueTime 2.0.

3.3.2.1. TrueTime Kernel

El bloque *TrueTime Kernel* se configura a través del cuadro de diálogo mostrado en la fig 3.6 con los siguientes parámetros (algunos parámetros pueden ser configurados en el tiempo de ejecución con el comando ttSetKernelParameter) [37]:

Función de inicialización. Es el nombre del script (MEX o Matlab) de la inicialización.

Init function argument. Un argumento opcional para el script de inicialización. Este puede ser cualquier estructura de Matlab.

Clock offset. Tiempo constante de compensación del tiempo nominal (tiempo actual de la simulación).

Clock drift. Fluctuación en el tiempo. Por ejemplo, si se desea que el tiempo local se ejecute 1 % más rápido que el tiempo nominal solo bastara poner 0.01 en este campo.

Battery. Si se habilita este cuadro le indica al kernel que depende de una fuente de energía.

3.3.2.2. TrueTime Wireless Network

El bloque *TrueTime Wireless Network* simula acceso al medio y transmisión de paquetes en una red de área local. Por ejemplo, cuando un nodo dentro de la red trata de transmitir un mensaje (mediante el comando ttSendMsg), una señal de activación es enviada a este bloque en el canal de entrada correspondiente. Cuando la simulación de la transmisión del mensaje finaliza, el bloque *TrueTime Wireless Network* envía una nueva señal por la salida del canal correspondiente al nodo receptor y el mensaje transmitido

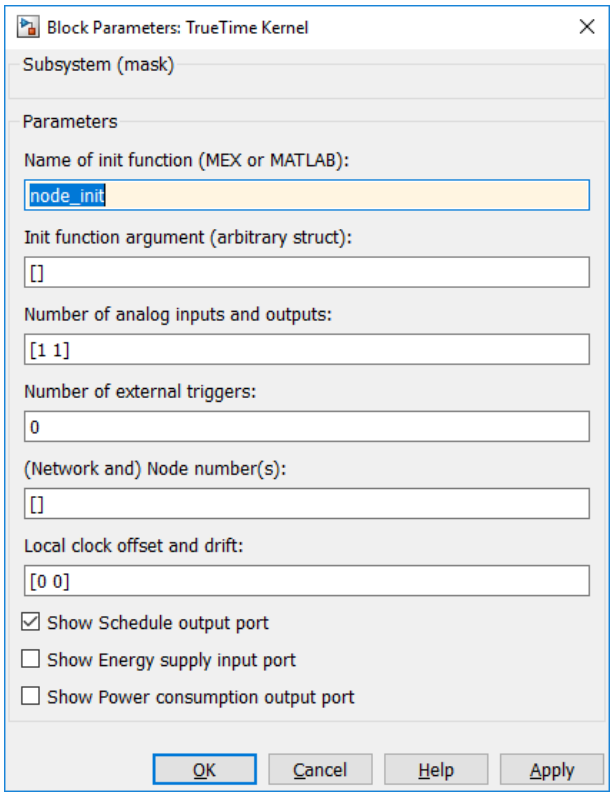


Figura 3.6. Cuadro de dialogo del bloque *TrueTime Kernel*.

se guarda en un buffer en la computadora del nodo de receptora. Por otra parte, un mensaje contiene la información acerca del envío, nodo receptor, datos arbitrarios del usuario (comúnmente son señales de medición y control), longitud del mensaje y algunos atributos opcionales como la prioridad y tiempo de vida [37].
Otra característica de este bloque es que considera la pérdida de ruta de las señales de radio, por lo cual cuenta con entradas x y y para especificar la ubicación verdadera de los nodos. En esta versión dos protocolos de redes son compatibles: IEEE 802.11b/g (WLAN) e IEEE802.15.4 (ZigBee) y el modelo de radio utilizado incluye soporte para:

- Redes inalámbricas Ad hoc.
- Antena isotrópica.
- Incapacidad para enviar y recibir mensajes al mismo tiempo.
- Atenuación de la señal de radio modelada como $\frac{1}{d^{\alpha}}$ donde d es la distancia en metros y α es el factor de perdidas por propagación.
- Interferencia de otras terminales.

El bloque *TrueTime Wireless Network* se configura a través del cuadro de diálogo mostrado en la fig 3.7 con los siguientes parámetros (algunos parámetros pueden ser configurados en el tiempo de ejecución con el comando ttSetNetworkParameter):

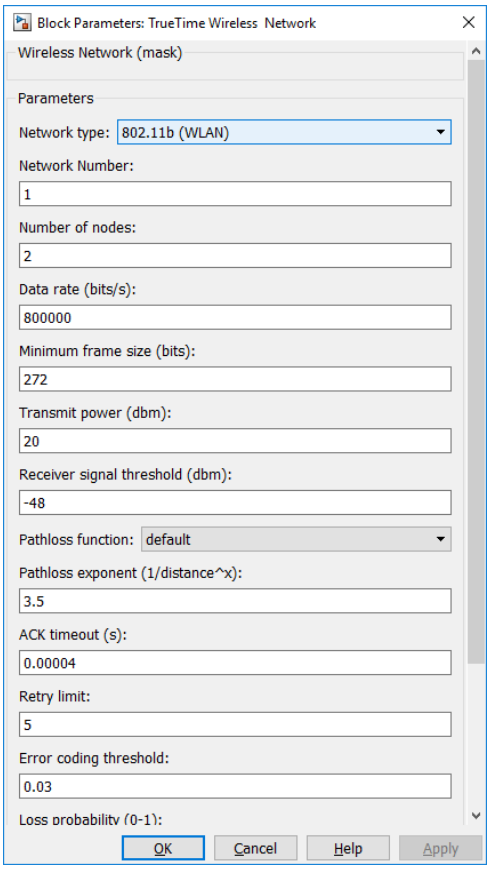


Figura 3.7. Cuadro de dialogo del bloque *TrueTime Wireless Network*.

Network number. Es el identificador de la red, el cual debe ser enumerado a partir de 1. Las redes cableadas e inalámbricas no permiten usar el mismo número.

Number of nodes. Indica el numero de nodos que formarán parte de la red. Este parámetro determina el tamaño de las entradas y salidas del bloque (Snd, Rcv and Schedule).

Data rate (bps). Indica la velocidad de la red.

Minimum frame size (bits). Mensaje o marco más corto, el cual es rellenado para cumplir con la longitud mínima. Denota el tamaño mínimo de la trama.

Transmit power. Indica cuán fuerte es la señal de radio, lo que permite calcular su alcance.

Receiver signal threshold. Indica el umbral con el que compara la energía de la señal recibida y así determinar si el medio esta ocupado.

Path-loss exponent. Indica la atenuación de la señal de radio, la cual es modelada como $\frac{1}{d^{\alpha}}$ donde d es la distancia en metros y α es el factor de pérdidas por propagación. Este ultimo típicamente oscila en un rango de 2-4.

ACK timeout. Indica el tiempo de espera para el mensaje de confirmación (ACK), antes de concluir que el mensaje fue perdido y solicite una retransmisión.

Retry limit. Es el número de veces que el nodo tratará de retransmitir el mensaje antes de darse por vencido.

Error coding threshold. Indica el porcentaje de error del bloque en un mensaje que el codificador puede manejar. Por ejemplo, ciertos esquemas de codificación pueden reconstruir completamente un mensaje si el bloque tiene menos del 3 % de error. Este porcentaje puede ser calculado mediante la relación señal a ruido, donde el ruido es el resto de las transmisiones en curso.

Capítulo 4

Diseño del control colaborativo de múltiples VANTs

En este capítulo se muestra el diseño de una estrategia de control colaborativo para la formación de un conjunto de VANTs tipo VTOL el cual fue tomado de la referencia [38] y será utilizado en los Capítulos 5 y 6.

4.1. Fundamentos matemáticos

4.2. Teoría de grafos

Un grafo está definido por un par $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, donde $\mathcal{V}$ es un conjunto finito no vacío cuyos elementos se llaman **vértices** o **nodos** y $\mathcal{E}$ es un conjunto cuyos elementos se llaman **aristas** o **ejes** [39]. Utilizando esta definición y enfocándose en las topologías de comunicación podemos considerar que los nodos o vértices son los vehículos aéreos no tripulados mencionados en capítulos anteriores y los ejes o aristas son considerados como la red de comunicación entre ellos. Existen dos tipos de grafos:

- Dirigidos: Si las aristas son *pares ordenados* de vértices $\mathcal{V}$, entonces se dice que el grafo $\mathcal{G}$ es dirigido
- No dirigidos: Si las aristas son *pares no ordenados* de vértices$\mathcal{V}$, entonces diremos que el grafo $\mathcal{G}$ es no dirigido.

Existen diversas formas de representar un grafo dirigido o no dirigido, entre la más destacada se encuentra la matriz de adyacencias que se describe de la siguiente forma:
Sea $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ un grafo de $\mathcal{V}$ nodos. La matriz de adyacencias A para $\mathcal{G}$ es una matriz A_{nxn} de valores booleanos, donde A(i,j) es verdad si y solo si existe una arista del nodo i al nodo j.

$$A(i,j) = \begin{cases} 1, \text{si existe la arista (i,j)} \\ 0, \text{en caso contrario} \end{cases}$$

El grafo de la fig. 4.1 será utilizado para implementar la configuración de comunicación entre los agentes donde A_1, A_2 y A_3 representa los VANTs-VTOL. Por lo tanto, la matriz de adyacencias se describe de la siguiente forma:

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \tag{4.1}$$

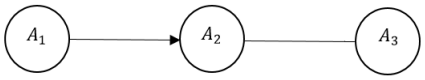


Figura 4.1. Grafo dirigido con tres nodos y tres vértices.

La matriz diagonal de grados, es una matriz diagonal denotada por D, donde el elemento (i,i) es igual al grado de vértices. Por lo tanto matriz de grado de nuestro grafo en la fig. 4.1 está descrito por:

$$D = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{4.2}$$

La matriz Laplaciana $\mathcal{L}$ de $\mathcal{G}$ está definida como $\mathcal{L} = D - A$ y está descrita de la siguiente forma:

$$\mathcal{L} = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{pmatrix} \tag{4.3}$$

4.3. Planteamiento del problema de consenso lider-seguidor

Se propone una estrategia de control colaborativo que permita realizar la formación de un conjunto multi VANT utilizando el consenso líder-seguidor que permita el transporte de carga, donde cada agente debe conocer el estado de sus vecinos. En la fig. 4.2 se muestra la configuración utilizada para proponer el control colaborativo, el cual esta compuesto por dos agentes y un agente virtual que será el líder de los otros agentes:

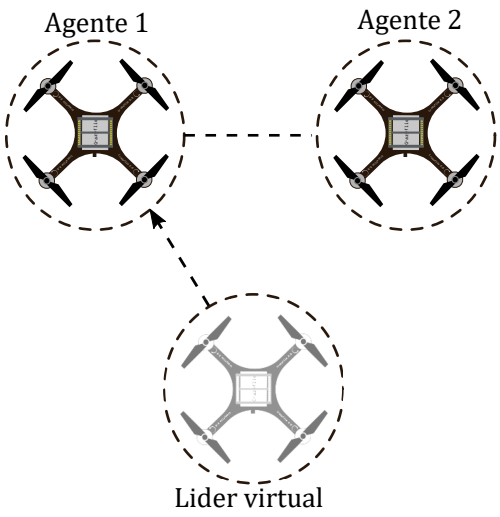


Figura 4.2. Grafo de tres agentes.

4.3.1. Propuesta de control colaborativo

Para proponer el control colaborativo líder-seguidor, considere un conjunto de VANTs-VTOL (véase fig. 4.3)

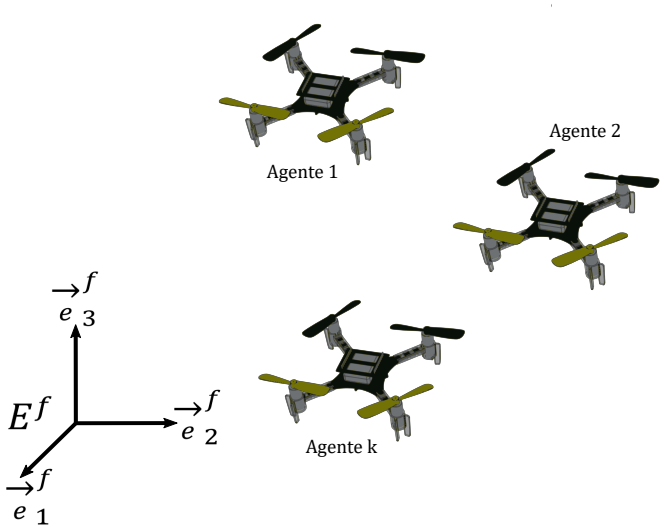


Figura 4.3. Grupo de N-agentes.

donde su sistema de ecuaciones del movimiento traslacional está descrito por la ecuación 4.4 y el movimiento rotacional en la ecuación 4.5 para cada uno de los agente, donde $k \in \mathcal{N} = k, ..., N$ representa el número del VANT y N el número de agentes.

$$\Sigma_{T_k} := \begin{cases} \dot{P}_k = V_k \\ \dot{V} = -g\vec{e}_{3_k}^{\,b} + \frac{1}{m_k}\mathcal{R}_{fb_k}(q_k)\,\vec{e}_{3_k}^{\,b} \end{cases} \tag{4.4}$$

$$\Sigma_{R_k} := \begin{cases} \dot{q}_k = \frac{1}{2}\Xi(q_k)\omega_k \\ \dot{\omega}_k = J_k^{-1}(-\omega_k \times J\omega_k + \Gamma_k) \end{cases} \tag{4.5}$$

Dado que el subsistema de traslación depende a su vez del subsistema de rotación de un VANT-VTOL, emplearemos la proposición del control de orientación (1) para cada agente que se encuentra descrito de forma extensa en el Capítulo 3 de la referencia [38].

Proposición 1. *Considere la dinámica rotacional del cuerpo rígido descrito por (4.5) con las siguientes entradas de control acotadas* $\Gamma = (\Gamma_1\ \Gamma_2\ \Gamma_3)^T$ *tal que*

$$\Gamma_{i_k} = -\sigma_{M_{i2_k}}\left(\hat{\xi}_{i_k} + \sigma_{M_{i1_k}}(\bar{\lambda}_{i_k}[\omega_{ei_k} + \rho_{i_k}\tilde{q}_{i_k}])\right) \tag{4.6}$$

con $i_k \in \{1, 2, 3\}$ *y donde* $\sigma_{M_{i1_k}}$ *y* $\sigma_{M_{i2_k}}$ *son funciones de saturación. Asuma que* $K_0 < M_{i2_k} - M_{i1_k}$ *y* $M_{1i_k} \geq 3\bar{\lambda}_{i_k}\rho_{i_k}$. $\bar{\lambda}_{i_k}$ *y* ρ_{i_k} *son parámetros positivos.* $\hat{\xi} = J\eta_1$ *con* η_1 *la estimación de perturbaciones desconocidas* $J^{-1}\xi$. *Por lo tanto, las entradas estabilizan robustamente el cuerpo rígido al origen* $(1\ 0^T\ 0^T)^T$ (*i.e.* $q_0 = 1, q_v = 0$ y $\omega = 0$) *con un dominio de atracción igual a* $\mathbb{S}^3 \times \mathbb{R}^3 \setminus (-1\ 0^T\ 0^T)^T$.

El objetivo es diseñar una ley de control colaborativo para la estabilización de un sistema multi VANT-VTOL a una determinada posición en el espacio. Por lo tanto, utilizando el sistema de ecuaciones del movimiento de traslación (ecuación 4.4).
Asumiendo que utilizando el control de orientación en la proposición 1 se puede estabilizar la dinámica del *Yaw* del VANT-VTOL, esto es, $\psi = 0$. Posteriormente, después de un tiempo suficientemente largo, el sistema se convierte en:

$$\begin{aligned} \dot{p_1} &= v_1 \\ \dot{p_2} &= v_2 \\ \dot{p_3} &= v_3 \end{aligned} \tag{4.7}$$

$$\begin{aligned} \dot{v_1} &= T\theta \\ \dot{v_2} &= -T\phi \\ \dot{v_3} &= T \end{aligned} \tag{4.8}$$

θ y ϕ pueden ser vistos como un intermediario de entrada para controlar 4.7 y 4.8. Para esto se definen los *controles virtuales* que permiten obtener la posición deseada.

$$\theta_d = mk_{p_1}(p_1^d - p_1) - mk_d v_1 \tag{4.9}$$

$$\phi_d = -mk_p(p_2^d - p_2) + mk_d v_2 \tag{4.10}$$

Posteriormente, se elije como empuje positivo el control de entrada llamado *fuerza de empuje* que se describe como:

$$T = mk_{p_3}(p_3^d - p_3) - mk_{d_3}v_3 + mk_{i_3}\int(p_3^d - p_3)dt \tag{4.11}$$

Utilizando el control de orientación en la proposición 1 y el control de posición en la ecuación (4.9, 4.10 y 4.11) de cada agente. Se propone un algoritmo de consenso líder-seguidor, el cual solo se realiza en la posición x y y. Esto se debe a que el control colaborativo está diseñado para transportar una carga utilizando un conjunto de VANT-VTOL a una misma altura y con un peso de carga proporcional en todos los agentes.

$$A_1 = \begin{cases} \phi_{d_1} = u_{1_1} = k^1_{p_x}(P_{x_2} - P_{x_1} + \Delta_{xij}) + k^1_{d_x}(v_{x_2} - v_{x_1}) + k^1_{p_x}(P^d_x - P_{x_1}) + k^1_{d_x}(v^d_x - v_{x_1}) \\ \theta_{d_1} = u_{2_1} = -[k^1_{p_y}(P_{y_2} - P_{y_1} + \Delta_{yij}) + k^1_{d_y}(v_{y_2} - v_{y_1}) + k^1_{p_y}(P^d_y - P_{y_1}) + k^1_{d_y}(v^d_y - v_{y_1})] \\ T_1 = k^1_{p_z}(P^d_z - P_{z_1}) - k^1_{d_z}v_{z_1} + mg \end{cases} \tag{4.12}$$

$$A_2 = \begin{cases} \phi_{d_2} = u_{1_2} = k^2_{p_x}(P_{x_1} - P_{x_2} + \Delta_{xij}) + k^2_{d_x}(v_{x_1} - v_{x_2}) \\ \theta_{d_2} = u_{2_2} = -[k^2_{p_y}(P_{y_1} - P_{y_2} + \Delta_{yij}) + k^2_{d_y}(v_{y_1} - v_{y_2})] \\ T_2 = k^2_{p_z}(P^d_z - P_{z_2}) - k^2_{d_z}v_{z_2} + mg \end{cases} \tag{4.13}$$

Donde $P = (P^d_x \ P^d_y \ P^d_z)^T$ y $v = (v^d_x \ v^d_y \ v^d_z)^T$ representan la posición y velocidad lineal del líder-virtual, respectivamente. $P_1 = (P_{x_1} \ P_{y_1} \ P_{z_1})^T$ y $P_2 = (P_{x_2} \ P_{y_2} \ P_{z_2})^T$ representan la posición del agente 1 y 2, respectivamente. El nodo líder es un exosistema que funge como señal de referencia deseada. k_p, k_d representa las ganancias de sintonización del control colaborativo. Δ_{xij}, Δ_{yij} representa la distancia de cada agente en el eje x y y. El agente 2 no recibe directamente la referencia deseada del líder, sin embargo en T, recibe la posición P^d_z y P^d_z del líder virtual.

Capítulo 5

Evaluación de una red Ad hoc para VANTs a nivel simulación

En éste capítulo se presenta la evaluación a nivel simulación de la red Ad hoc propuesta en el Capítulo 3 mediante el uso de las plataformas de simulación: ns-2 y Matlab/Simulink/TrueTime2.0.

5.1. Simulación de una red FANET de 4 nodos móviles en ns-2 (Macro-ambientes)

En esta sección se muestra la simulación de una FANET de 4 nodos móviles en ns-2, la cual realiza cuatro formaciones. Para esta simulación, una cobertura de 100 m es considerada para las antenas de los nodo y una distancia inicial de 100 m entre nodos continuos y de 141.5 m entre las diagonales. Los rangos de 100 m fueron establecidos variando la potencia del receptor, misma que puede ser calculada a partir del modelo de propagación utilizado. Se considero un modelo de propagación en el espacio libre (véase fig. 5.1) para distancias no mayores a 50 m. La potencia del receptor esta dada por:

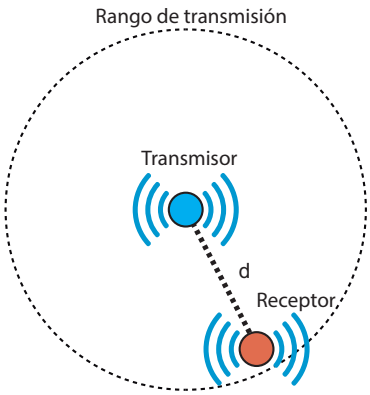


Figura 5.1. Modelo de propagación libre.

$$P_r = \frac{P_t * G_t * G_r * \lambda^2}{(4 * pi * d)^2 * L} \tag{5.1}$$

donde P_t es la potencia de la señal transmitida. G_t y G_r son la ganancia de la antena en el emisor y receptor respectivamente. $L(L \geq 1)$ es la pérdida del sistema. λ es la longitud de la onda y d es la distancia en metros entre el transmisor y el receptor.
Por otra parte, un modelo de propagación de dos rayos tierra (véase fig. 5.2) es considerado para distancias por arriba de 50 m.

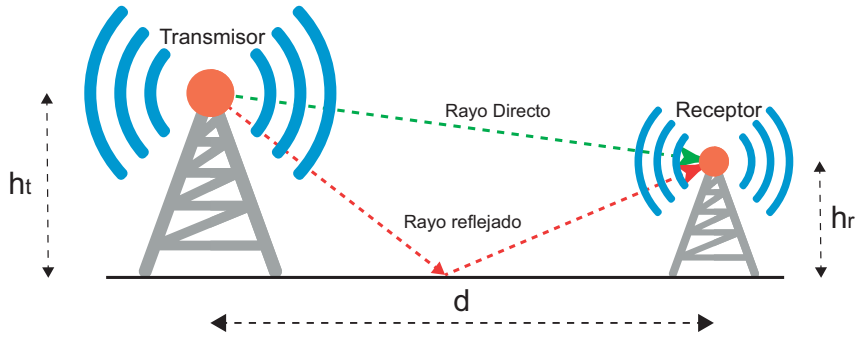


Figura 5.2. Modelo de propagación dos rayos tierra.

La potencia recibida esta dada por:

$$P_r = \frac{P_t * G_t * G_r * (h_t^2 * h_r^2)}{d^4 * L} \tag{5.2}$$

donde h_t y h_r son la altura de las antenas del emisor y receptor respectivamente.
Para esta simulación se tomó: $L = 1$, $G_t = 1$, $G_r = 1$ y $P_t = 0.2818$ (W). Por otro lado, la tabla 5.1 muestra las posiciones en metros de las formaciones realizadas por la red, elegidas de modo que existiera por lo menos un caso en donde dos nodos necesiten establecer una conexión y ambos se encuentren fuera de su radio de cobertura, haciendo que la red ejecute su sistema de enrutamiento para establecer un enlace alterno a través de los otros nodos que la forman.

Tabla 5.1. Posición en metros de las formaciones realizadas en la simulación.

	Nodo 0		Nodo 1		Nodo 2		Nodo 3	
	x0	y0	x1	y1	x2	y2	x3	y3
Formación Inicial	25	150	125	150	125	50	25	50
Formación # 1	60	80	20	40	100	120	140	160
Formación # 2	20	100	60	20	120	20	160	100
Formación # 3	100	50	1	50	200	50	300	50
Formación Final	25	150	125	150	125	50	25	50

A continuación, se realiza la simulación de la FANET haciendo uso de los protocolos

de enrutamiento: AODV y DSDV en el simulador ns-2 2.35 corriendo bajo el sistema operativo Ubuntu 14.4. También, se evalúan las métricas discutidas en 3.2, para elegir el mejor protocolo que se adecue a la demanda de la red. Esta simulación se ejecutaa considerando un área de 500 x 500 m.

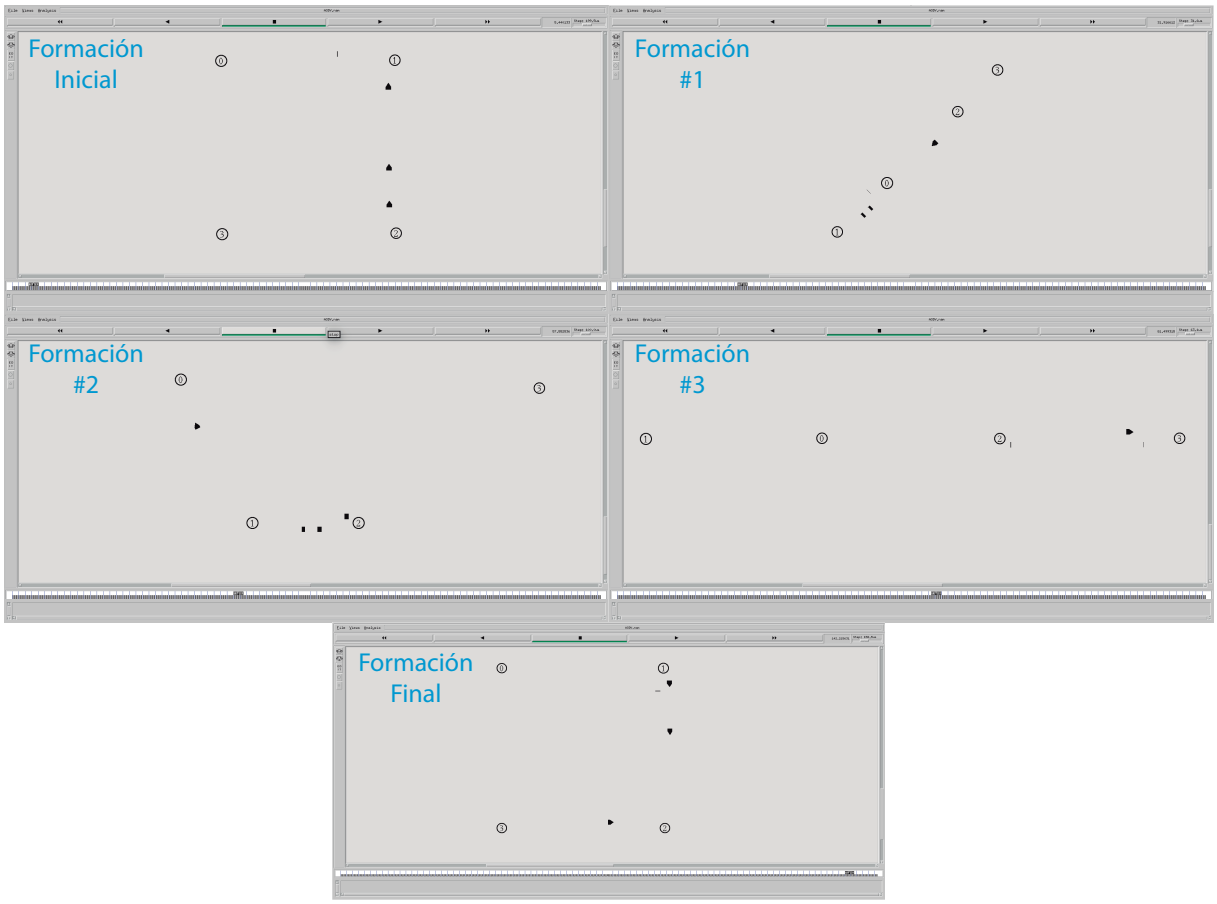


Figura 5.3. Formaciones propuestas para la simulación de la FANET en ns-2.

La tabla 5.2 muestra con detalle los parámetros que fueron considerados para la simulación. Cabe mencionar que durante esta la velocidad y la altura son las mismas para cada nodo.

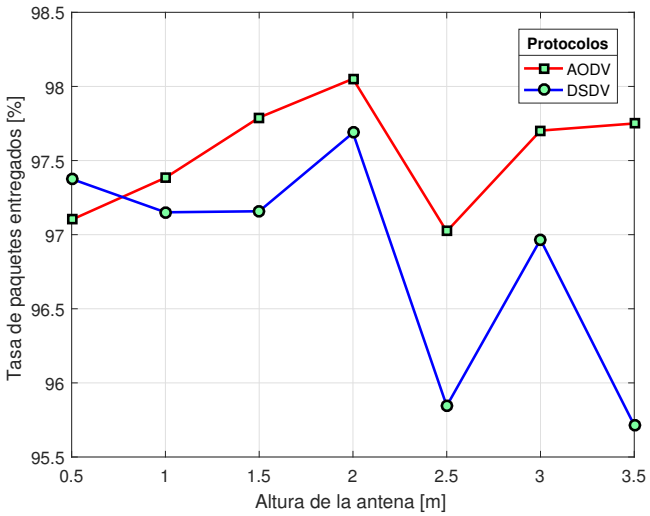
Por otro lado, la evaluación de las métricas se realizaron de forma global (para toda la red) y fueron programadas en *awk y PERL*. Los resultados son mostrados en gráficas utilizando *MATLAB*.

Tabla 5.2. Parámetros para la simulación.

Parámetros	Valores
Tipo del canal	Channel/Wireless channel
Modelo de radio propagación	Propagation/ Two ray ground/FreeSpace
Tipo de interfaz de la red	Phy /wirelessphy
Altura de la red (m)	0.5 - 3.5
Tipo de MAC	Mac/802.11
Tipo de interfaz de cola	Queue/Drop Tail
Tipo de capa de enlace	LL
Antena	Antenna/omni antenna
Área (m x m)	500x500
Número de nodos móviles	4
Velocidad de los nodos (m/s)	1,5,10,15,20,25
Tipo de fuente	TCP/NewReno
Protocolo de enrutamiento	AODV, DSDV
Sistema operativo	Ubuntu 14.04
Versión de NS2-2	2.35

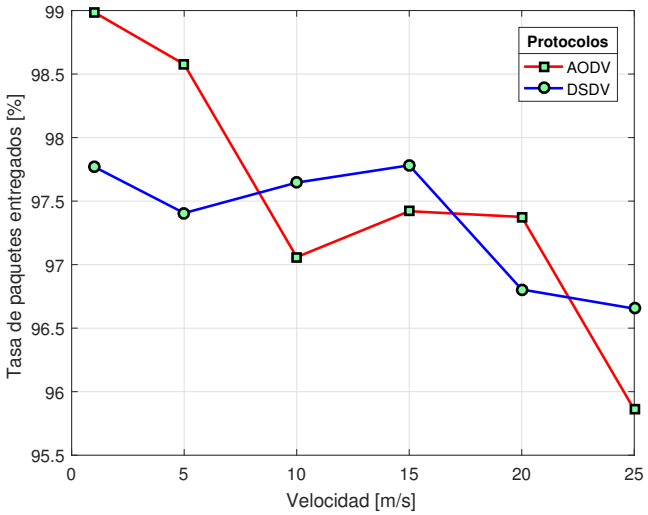
5.1.1. Métrica PDR

Las figuras 5.4 y 5.5 muestran la tasa de paquetes entregados para los protocolos AODV y DSDV cambiando la altura de las antenas y la velocidad de movilidad de los nodos respectivamente. El protocolo AODV presenta su mejor tasa del 98.0514 % para una altura de 2 m y del 98.9863 % para una velocidad de 1 m/s. En contraste, el protocolo DSDV que presenta su mejor tasa del 97.6867 % para una altura de 2 m y del 98.9863 % para una velocidad de 15 m/s.



	0.5	1	1.5	2	2.5	3	3.5
AODV	97.1008	97.384	97.7903	98.0514	97.0222	97.7019	97.7501
DSDV	97.3758	97.1509	97.1579	97.6867	95.8446	96.9669	95.7143

Figura 5.4. Tasa de paquetes entregados variando la altura de la antena.

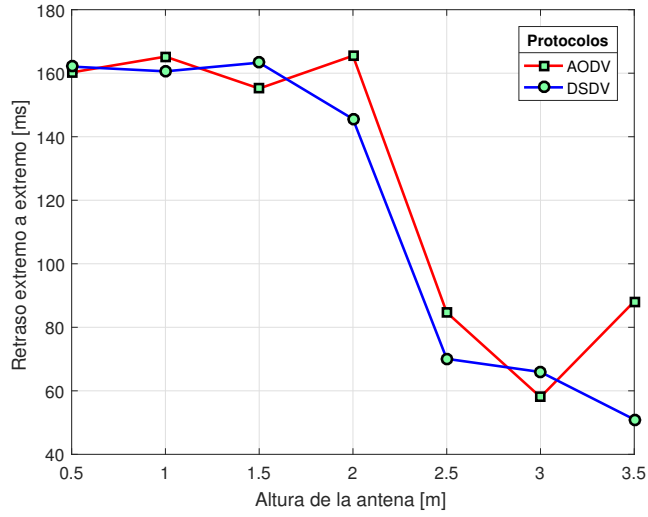


	1	5	10	15	20	25
AODV	98.9863	98.5772	97.0593	97.4205	97.3754	95.8586
DSDV	97.7703	97.4057	97.6457	97.7817	96.8044	96.6538

Figura 5.5. Tasa de paquetes entregados variando la velocidad de los nodos.

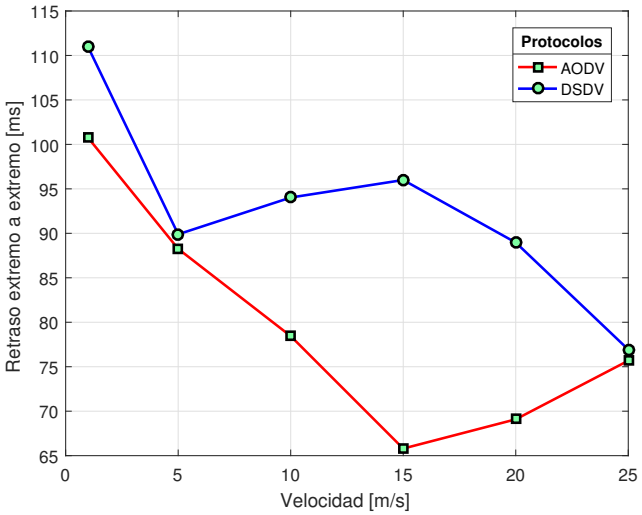
5.1.2. Métrica E2ED

Las figuras 5.6 y 5.7 muestran el retraso extremo a extremo para los protocolos AODV y DSDV moviendo la altura de las antenas y la velocidad de los nodos respectivamente. El protocolo AODV presenta su menor retraso con 58.0377 ms para una altura de 3 m y 65.7977 ms para una velocidad de 15 m/s. En contraste, el protocolo DSDV presenta su menor retraso con 50.9746 ms para una altura de 3.5 m y 76.9342 ms para una velocidad de 25 m/s.



	0.5	1	1.5	2	2.5	3	3.5
AODV	160.246	165.215	155.178	165.488	84.6672	58.0377	88.1378
DSDV	162.075	160.587	163.318	145.578	70.0709	65.9257	50.9746

Figura 5.6. Retraso extremo a extremo variando la altura de la antena.

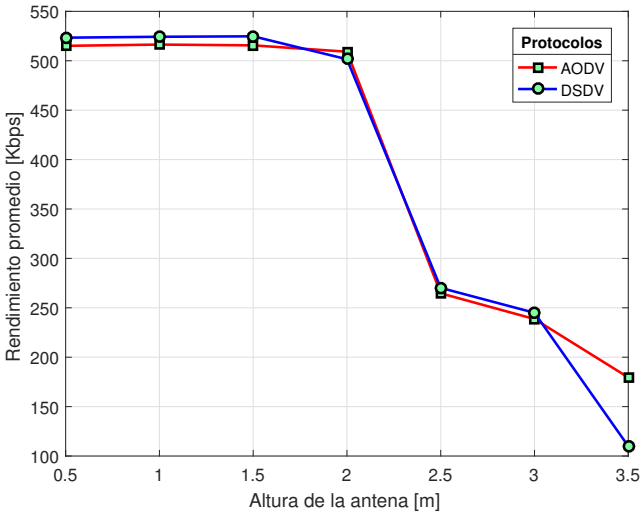


	1	5	10	15	20	25
AODV	100.735	88.2519	78.4442	65.7977	69.0858	75.678
DSDV	110.938	89.8901	94.0248	95.9949	88.9625	76.9342

Figura 5.7. Retraso extremo a extremo variando la velocidad de los nodos.

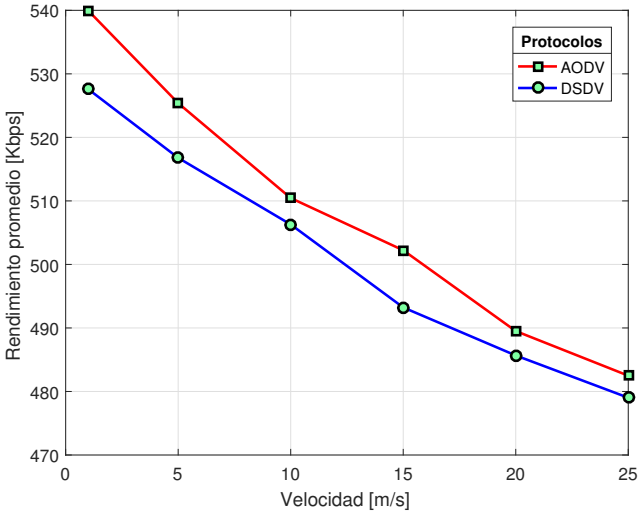
5.1.3. Métrica Thp

Las figuras 5.8 y 5.9 muestran el rendimiento promedio para los protocolos AODV y DSDV moviendo la altura de antena y la velocidad de los nodos respectivamente. El protocolo AODV presenta su mejor rendimiento con 516.39 $Kbps$ para una altura de 1 m y 65.7977 $Kbps$ para una velocidad de 15 m/s. En contraste, el protocolo DSDV presenta su mejor rendimiento con 524.74 $Kbps$ para una altura de 1.5 m y 76.9342 $Kbps$ para una velocidad de 25 m/s.



	0.5	1	1.5	2	2.5	3	3.5
AODV	515.11	516.39	515.59	509.16	264.64	238.62	179.57
DSDV	523.37	524.28	524.74	501.53	270.15	244.95	109.95

Figura 5.8. Rendimiento promedio variando la altura de la antena.



	1	5	10	15	20	25
AODV	539.97	525.39	510.45	502.23	489.51	482.49
DSDV	527.6	516.78	506.29	493.23	485.67	479.02

Figura 5.9. Rendimiento promedio variando la velocidad de los nodos.

5.1.4. Consumo energético

La fig. 5.10 muestra el consumo energético en cada nodo para los protocolos AODV y DSDV. Esta simulación fue realizada añadiendo los parámetros mostrados en la tabla 5.3. Una vez realizada la simulación, se calculó el promedio del consumo energético de los cuatro nodos. Por consiguiente, el protocolo AODV tiene un consumo de 17.53812025 J mientras que el DSDV de 19.551985 J.

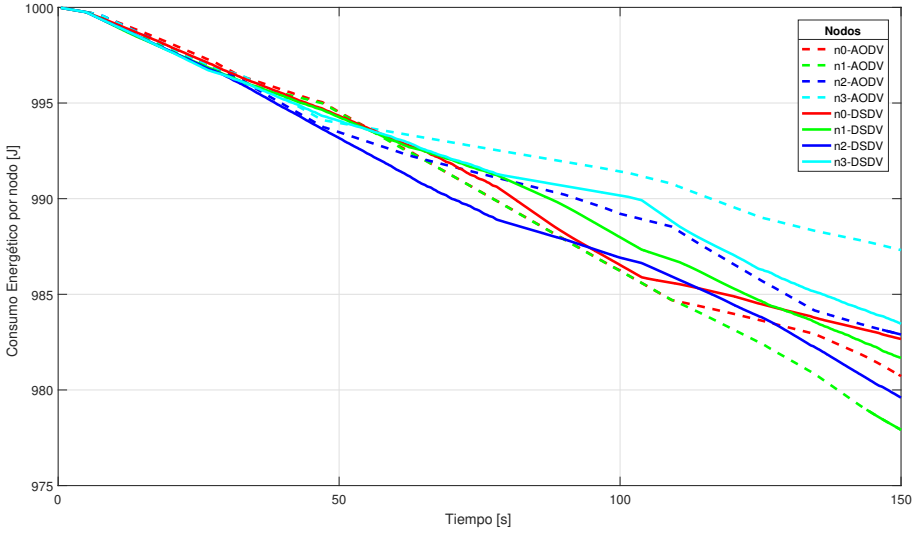


Figura 5.10. Consumo energético de los protocolos con una velocidad de movilidad de 5 m/s.

Tabla 5.3. Parámetros para la simulación de la energía

Parámetros	Valor
Modelo de energía	EnergyModel
Potencia del transmisor (W)	0.10
Potencia del receptor (W)	0.25
Potencia en estado dormido (J)	0.05
Energía inicial (J)	1000

De acuerdo con los resultados obtenidos en la simulación donde la altura de la antena era variable, en términos generales tenemos que el protocolo AODV tiene una tasa de paquetes entregados del 97.54 % siendo este 0.71 % mejor que el DSDV. La altura de la antena que favorece a esta métrica es de 2 m. El rendimiento promedio (Thp) para el AODV es de 391.29$Kbps$, el cual resulta 1.46 % mejor que el otro protocolo. Esta métrica tiene su mejor desempeño para alturas de la antena en un rango de 1 m a 1.5 m. Por otra parte, el AODV presenta un consumo energético de 13.79 J, mientras que el DSDV 14.19 J. Sin embargo, el protocolo DSDV resulta tener un retraso extremo a extremo de 116.93 ms siendo este 6.66 % más rápido que el AODV. El rango de la antena favorable para esta métrica es de 3 m a 3.5 m. De acuerdo con estos resultados obtenidos, el AODV resulta ser el mejor protocolo en un rango de 1.5 m a 2.5 m.

En contraste, para la simulación donde se varia la velocidad de los drones se obtuvo lo siguiente: el protocolo AODV tiene una tasa de paquetes entregados del 97.54 % siendo este 0.20 % mejor que el DSDV. La velocidad del nodo que favorece a esta métrica es de 1 m a 5 m. El rendimiento promedio (Thp) para el AODV es de 508.34$Kbps$, el cual resulta 1.35 % mejor que el otro protocolo. Esta métrica tiene su mejor desempeño para alturas de la antena en un rango de 1 m/s a 5 m/s. Por otra parte, el AODV presenta un consumo energético de 17.53 J, mientras que el DSDV 19.55 J. Sin embargo, el protocolo AODV resulta tener un retraso extremo a extremo de 79.66 ms siendo este 14.14 % más rápido que el DSDV. El rango de la antena favorable para esta métrica es de 15 m a 25 m. Para esta simulación, nuevamente, el AODV resulta ser el mejor protocolo para implementar en un rango de velocidad de 1 m/s a 5 m/s.

De las simulaciones anteriores, se puede notar que existen pequeñas diferencias entre los protocolos AODV y DSDV. Esto se debe a que ambos protocolos son considerados altamente idóneos para las redes Ad hoc [23–30]. Además, estos fueron probados en una red pequeña. Por lo que, si se deseará percibir una notable diferencia entre ambos protocolos, una simulación de mas de 200 nodos seria una buena opción. Por otra parte, la simulación de la antena variable fue realizada utilizando el modelo de propagación de dos rayos, sin embargo, es posible ver que a medida que la altura de la antena aumenta la red presenta peores resultados, debido a que para distancias cortas (< 100 m), la potencia del receptor empieza a aumentar, provocando afectaciones en la oscilación de los rayos emitidos para

este modelo. Debido a que para distancias cortas no es recomendado este tipo de modelo, el simulador ns-2 propone como valores predeterminados una altura de 1.5 m a 550 metros.

5.2. Simulación de una red FANET de 4 nodos móviles utilizando TrueTime 2.0 (Macro-ambiente - Micro-ambiente)

En esta sección se muestra la simulación de una FANET de 4 nodos móviles operando con el protocolo de enrutamiento AODV. La simulación se realizó haciendo uso del toolbox TrueTime 2.0 en la plataforma de MATLAB/Simulink. Nuevamente, el rango de transmisión para cada nodo fue de 100 m, el cual se fijó variando la potencia del receptor, misma que es calculada utilizando la fórmula del modelo de propagación utilizado por TrueTime 2.0, como se muestra a continuación:

Modelo de propagación utilizado por TrueTime2.0

$$P_r = \frac{P_t}{(d+1)^{3.5}} \tag{5.3}$$

donde P_t es la potencia de la señal transmitida, d es la distancia en metros y 3.5 es el exponente de pérdidas normalizado a un ambiente urbano.
La figura 5.11 muestra el diagrama a bloques en Simulink de la FANET con 4 nodos móviles, la cual utiliza el AODV como protocolo de enrutamiento.

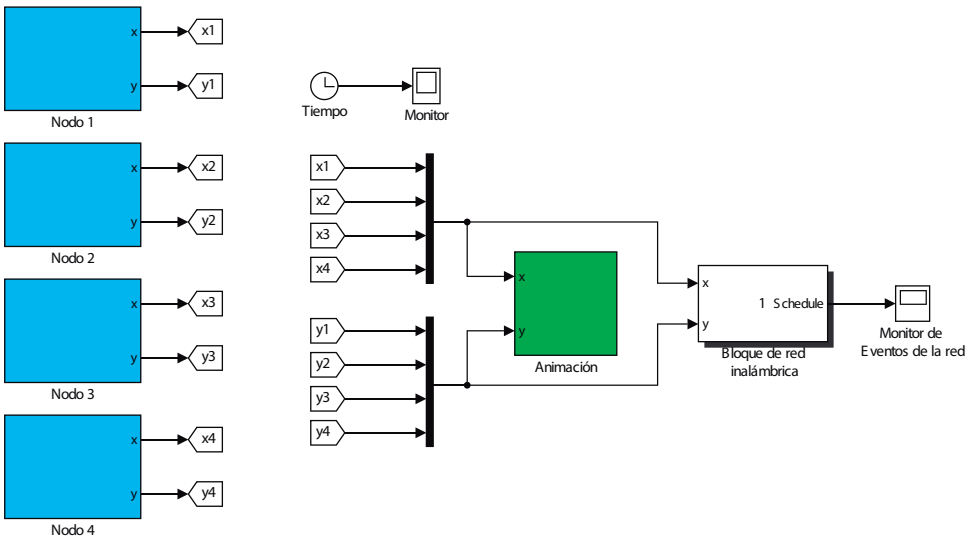


Figura 5.11. Protocolo AODV para una Fanet de 4 nodos.

La fig. 5.12 muestra el diagrama a bloques que representa a un nodo. En este diagrama se puede observa el uso del bloque *TrueTime Kernel* discutido en 3.3.2. El cual es confi-

gurado con una función *node_int.m* y esta conformado por dos entradas y salidas (x,y) (véase 3.6).

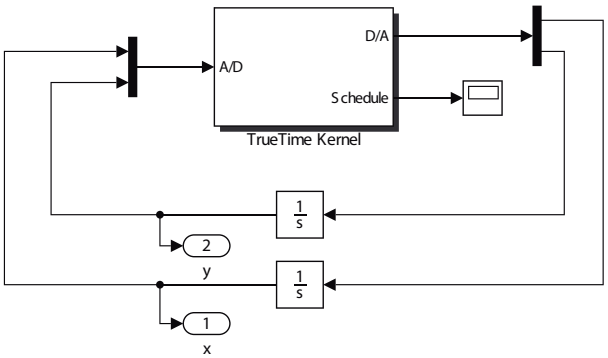


Figura 5.12. Diagrama a bloques de un nodo

Por otra parte, el bloque *Animación* se encarga de graficar la posición y radio de cobertura de los nodos a lo largo de la simulación. Este bloque se ejecuta cada 100 ms y su área es de 400 x 400 m (Macro-ambiente).
Finalmente, el *Bloque de red inalámbrica* (*TrueTime Wireless Network*), es el encargado de simular la comunicación inalámbrica entre los cuatro nodos que forman la red. Adicionalmente, se toman en cuenta diferentes parámetros para ser considerados en la simulación, los cuales son configurados a través de su ventana de diálogo (véase la fig. 3.7). Para esta simulación se consideraron los siguientes parámetros:

Tabla 5.4. Parámetros para la configuración del bloque TrueTime Wireless Network.

Parámetro	Valor
Tipo de red	802.11b
Número de la red	1
Número de nodos	4
Tasa de envío (bits/s):	1
Tamaño minino de la trama (bits):	512
Potencia del transmisor (dBm):	24.4994 (0.2818 W)
Umbral de la señal del receptor (dBm)	-45.68
Exponente de pérdida de la ruta	3.5
Tiempo de espera para el mensaje de confirmación (s):	0.00004
Limite de reintentos	5
Umbral de error del codificador	0.03

En esta simulación se realiza solo una formación, en la cual los nodos se encuentran a una distancia promedio de 100 m de sus nodos vecinos y de 142 m de su nodo más lejano (véase fig. 5.13) a una velocidad de 5 m/s. Las posiciones de los nodos se muestran con detalle en la tabla 5.5. El apéndice B muestra el pseudocódigo de las tareas de envío y recepción de datos utilizado por el TrueTime 2.0 para el protocolo de enrutamiento AODV.

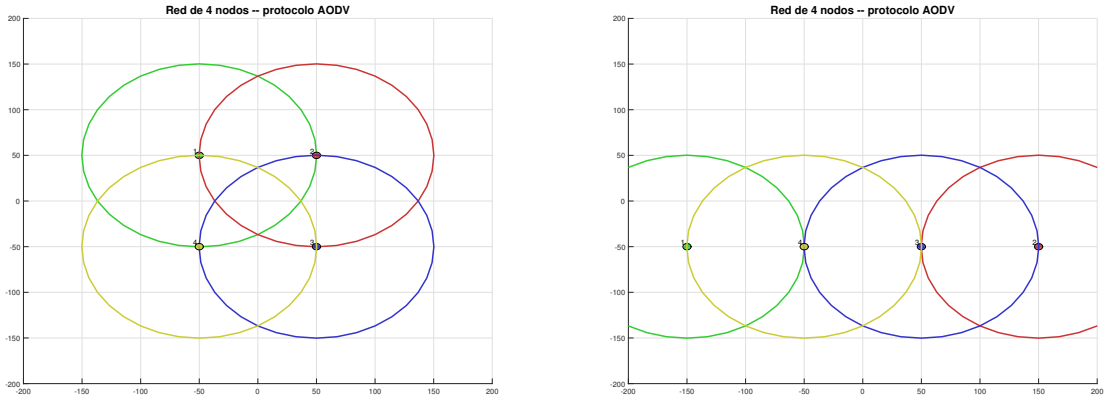


Figura 5.13. Formaciones realizadas en la simulación.

Tabla 5.5. Posiciones en metros de las formaciones realizadas en la simulación.

	Nodo 1		Nodo 2		Nodo 3		Nodo 4	
	x1	y1	x2	y2	x3	y3	x4	y4
Formación Inicial	-50	50	50	50	50	-50	-50	-50
Formación # 1	-150	-50	150	-50	50	-50	-50	-50

La fig. 5.14 muestra la parte inicial del reporte de eventos generados en la red, donde es posible observar que el primer evento se realiza en la capa de aplicación e indica que se desea establecer una transmisión del *nodo* 1 al 4 . Dado que el sistema acaba de comenzar, no tiene información sobre una ruta existente que logre dicha comunicación. Sin embargo, notifica al *nodo* 1 que no existe una ruta válida, almacena el paquete de datos que se desea enviar y comienza su rutina de detección de rutas.

```
Tiempo: 0.02| Aplicación | El Nodo#1 desea transmitir al Nodo#4 los datos: 90  76 con un tamaño de: 20
No existe una ruta valida
Almacenando mensaje 1
Tiempo: 0.02| Aplicación | El Nodo#2 desea transmitir al Nodo#4 los datos: 46   1 con un tamaño de: 20
No existe una ruta valida
Almacenando mensaje 1
Time: 0.021374| Una nueva ruta ha sido establecida entre el Nodo#1 y el Nodo#4
---> 1  4 <---
1 mensajes de datos en el buffer:
Enviando mensaje almacenado 1 al  Nodo#4
Aplicación | El Nodo#4 recibió los datos: 90  76 en el tiempo: 0.022286
Buffer vaciado
```

Figura 5.14. Reporte de eventos generados durante la simulación (inicio).

El algoritmo AODV en esta simulación tarda de 20 *ms* a 50 *ms* en encontrar una ruta si es que ésta existiera. Una vez que la red establece una ruta válida del *nodo* 1 al 4, la red opta por enviar los paquetes almacenados en el buffer. Esto se debe a que dicha tarea esta establecida como de alta prioridad. De esta manera, se logra que el buffer siempre se encuentre disponible para evitar pérdidas de paquetes. Una vez enviado el mensaje de datos, el nodo receptor retorna un acuse indicando que se recibieron exitosamente los

datos y el tiempo en el que llegaron (véase fig. 5.14).
Los datos que se transmiten en el mensaje son aleatorios y buscan simular los datos que se enviarán cuando sea implementado el control. Por el momento, se envían las posiciones de x y y con un tamaño de trama de 20 bits; una resolución de 10 bits para cada campo de datos que se desee enviar, sin embargo, este puede crecer tanto como se deseé.
La fig. 5.15 muestra que conforme transcurre el tiempo surge una nueva solicitud de envío (del *nodo* 2 al 4). La red vuelve a almacenar los datos en el buffer y busca nuevas rutas. En este caso, una nueva ruta es encontrada en $t = 13.52\ s$ y vuelve a repetir las tareas anteriormente mencionadas. En este momento, la red cuenta con dos rutas para el envío de la información, sin embargo, al ser un protocolo reactivo este solo mantiene las rutas por un cierto tiempo si esta deja de usarse. Por esta razón un algoritmo de mantenimiento de ruta es ejecutado cada 10 ms.

```
Tiempo: 13.52| Aplicación | El Nodo#2 desea transmitir al Nodo#4 los datos: 46  57 con un tamaño de: 20
Existe un ruta establecida en la tabla de enrutamiento para esta solicitud ---> 2  3  4 <---
Aplicación | El Nodo#4 recibió los datos: 79  68 en el tiempo: 13.5209
Aplicación | El Nodo#4 recibió los datos: 46  57 en el tiempo: 13.5215
Tiempo: 14.02| Aplicación | El Nodo#1 desea transmitir al Nodo#4 los datos: 80   5 con un tamaño de: 20
Existe un ruta establecida en la tabla de enrutamiento para esta solicitud ---> 1  4 <---
Tiempo: 14.02| Aplicación | El Nodo#2 desea transmitir al Nodo#4 los datos: 60   5 con un tamaño de: 20
Existe un ruta establecida en la tabla de enrutamiento para esta solicitud ---> 2  3  4 <---
Aplicación | El Nodo#4 recibió los datos: 80   5 en el tiempo: 14.0209
Aplicación | El Nodo#4 recibió los datos: 60   5 en el tiempo: 14.0215
Tiempo: 14.52| Aplicación | El Nodo#1 desea transmitir al Nodo#4 los datos: 41  30 con un tamaño de: 20
Existe un ruta establecida en la tabla de enrutamiento para esta solicitud ---> 1  4 <---
Tiempo: 14.52| Aplicación | El Nodo#2 desea transmitir al Nodo#4 los datos: 88   1 con un tamaño de: 20
Existe un ruta establecida en la tabla de enrutamiento para esta solicitud ---> 2  3  4 <---
Aplicación | El Nodo#4 recibió los datos: 41  30 en el tiempo: 14.5209
Aplicación | El Nodo#4 recibió los datos: 88   1 en el tiempo: 14.5215
Tiempo: 15.02| Aplicación | El Nodo#1 desea transmitir al Nodo#4 los datos: 77  98 con un tamaño de: 20
Existe un ruta establecida en la tabla de enrutamiento para esta solicitud ---> 1  4 <---
Tiempo: 15.02| Aplicación | El Nodo#2 desea transmitir al Nodo#4 los datos: 99  79 con un tamaño de: 20
Existe un ruta establecida en la tabla de enrutamiento para esta solicitud ---> 2  3  4 <---
Aplicación | El Nodo#4 recibió los datos: 77  98 en el tiempo: 15.0209
Aplicación | El Nodo#4 recibió los datos: 99  79 en el tiempo: 15.0215
```

Figura 5.15. Reporte de eventos generados durante la simulación (descubrimiento de rutas).

```
El Nodo#3 perdio la conexión con el Nodo#2
El Nodo#3 perdio la conexión con el Nodo#4
Tiempo: 15.52| Aplicación | El Nodo#2 desea transmitir al Nodo#4 los datos: 34  29 con un tamaño de: 20
Existe un ruta establecida en la tabla de enrutamiento para esta solicitud ---> 2  3  4 <---
Tiempo: 16.02| Aplicación | El Nodo#2 desea transmitir al Nodo#4 los datos: 34  53 con un tamaño de: 20
Existe un ruta establecida en la tabla de enrutamiento para esta solicitud ---> 2  3  4 <---
El Nodo#2 perdio la conexión con el Nodo#3
El Nodo#4 perdio la conexión con el Nodo#3
Tiempo: 16.52| Aplicación | El Nodo#2 desea transmitir al Nodo#4 los datos: 73  31 con un tamaño de: 20
No existe una ruta valida
Almacenando mensaje 1
```

Figura 5.16. Reporte de eventos generados durante la simulación (pérdida de la ruta).

A lo largo de la simulación podemos observar como las rutas de transmisión son alternadas de acuerdo con la solicitud de envío por parte de los nodos. Sin embargo, en $t = 15\ s$ existe una pérdida del enlace entre los *nodos* $3 - 4$ y $3 - 2$. Por otra parte, el *nodo* 2 le

indica a la red que necesita realizar una transmisión con el *nodo* 4, por lo que nuevamente se toma la solicitud, se almacenan los datos que necesiten ser enviados en un buffer y se busca una nueva ruta. Estos eventos se muestran en la fig, 5.16.
En la simulación una nueva ruta se encuentra a través del *nodo* 1 en $t = 27.52\ s$. Una vez establecida la conexión se envían todos los datos almacenados en el buffer y se continua con la simulación hasta terminar su formación la cual dura 50 s (véase fig. 5.17).

```
Time: 27.0233| Una nueva ruta ha sido establecida entre el Nodo#2 y el Nodo#4
 ---> 2  1  4 <---
22 mensajes de datos en el buffer:
Enviando mensaje almacenado 1 al  Nodo#4
Enviando mensaje almacenado 2 al  Nodo#4
Enviando mensaje almacenado 3 al  Nodo#4
Buffer vaciado
Aplicación | El Nodo#4 recibió los datos: 6  99 en el tiempo: 27.046
Aplicación | El Nodo#4 recibió los datos: 58  42 en el tiempo: 27.0472
Aplicación | El Nodo#4 recibió los datos: 52  33 en el tiempo: 27.0489
                                          :
                                          :
                                          :
Tiempo: 27.52| Aplicación | El Nodo#2 desea transmitir al Nodo#4 los datos: 46  57 con un tamaño de: 20
Existe un ruta establecida en la tabla de enrutamiento para esta solicitud ---> 2  1  4 <---
Aplicación | El Nodo#4 recibió los datos: 46  57 en el tiempo: 27.5215
Tiempo: 28.02| Aplicación | El Nodo#2 desea transmitir al Nodo#4 los datos: 80   5 con un tamaño de: 20
Existe un ruta establecida en la tabla de enrutamiento para esta solicitud ---> 2  1  4 <---
Aplicación | El Nodo#4 recibió los datos: 80   5 en el tiempo: 28.0215
Tiempo: 28.52| Aplicación | El Nodo#2 desea transmitir al Nodo#4 los datos: 60   5 con un tamaño de: 20
Existe un ruta establecida en la tabla de enrutamiento para esta solicitud ---> 2  1  4 <---
Aplicación | El Nodo#4 recibió los datos: 60   5 en el tiempo: 28.5215
Tiempo: 29.02| Aplicación | El Nodo#2 desea transmitir al Nodo#4 los datos: 41  30 con un tamaño de: 20
Existe un ruta establecida en la tabla de enrutamiento para esta solicitud ---> 2  1  4 <---
Aplicación | El Nodo#4 recibió los datos: 41  30 en el tiempo: 29.0215
```

Figura 5.17. Reporte de eventos generados durante la simulación (búsqueda de una nueva ruta).

Algo importante que se debe tener en cuenta a la hora de transmitir datos sobre un canal, es que no debe de haber transmisiones simultaneas debido a las colisiones y pérdidas de información que se pudieran tener. Una de la características importantes del bloque *TrueTime Wireless Network* es que se le ha incluido soporte para la incapacidad de transmitir simultáneamente, por esta razón este módulo cuenta con prioridades de tareas y un monitor de eventos. La fig. 5.18 muestra los eventos en los nodos a lo largo de la simulación.
El monitor de eventos de transmisión presenta tres posibles estados para los nodos a la hora de transmitir a través de tres niveles, los cuales son: Alto=ejecutándose, medio=esperando, bajo=inactivo. Estos tres estados garantizan que a la hora de transmitir simultáneamente exista un tiempo de espera si es que algún otro nodo cuenta con una prioridad más alta, esto puede verse en la fig.5.19 , donde se realiza un acercamiento al reporte eventos generados durante la simulación.

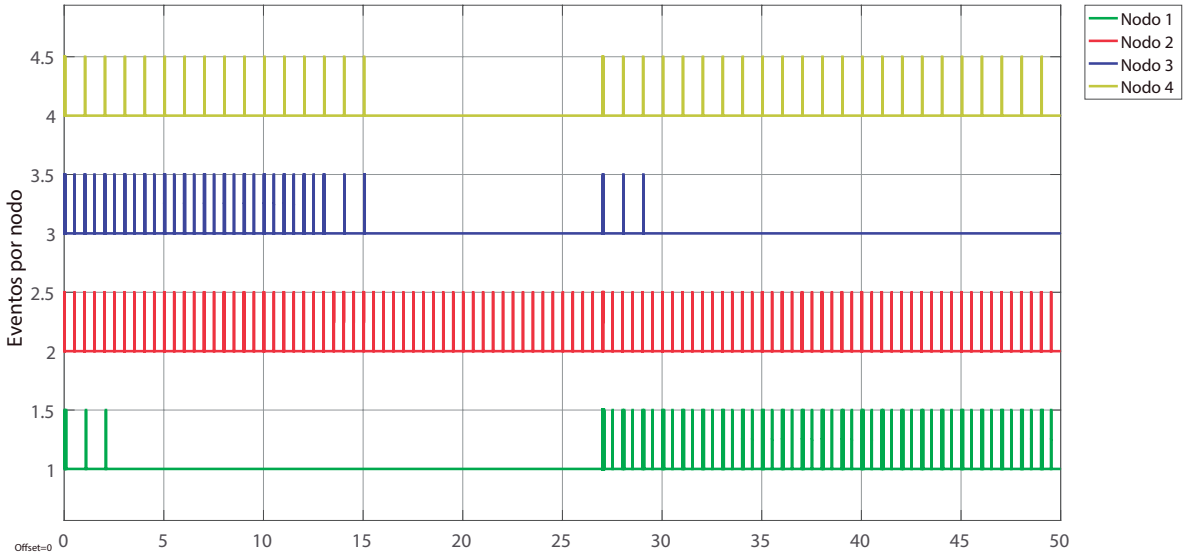


Figura 5.18. Monitor de eventos de la red.

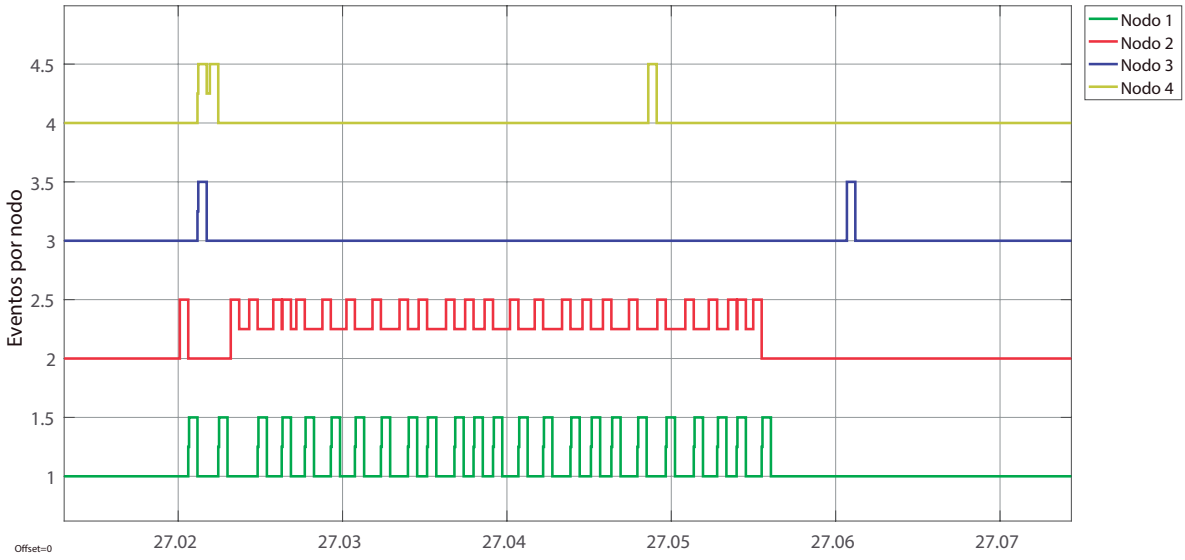


Figura 5.19. Zoom del monitor de eventos de la red.

5.2.1. Simulación implementando el control

En esta sección, se acopla el sistema de comunicaciones a un sistema de control colaborativo para 4 agentes. Donde el bloque *control* es el encargado de indicarle al *sistema de comunicaciones* el momento exacto donde se requiera una transmisión y el tipo de datos que requiere enviar (posiciones, velocidad, etc.). La fig. 5.20 muestra un diagrama general del sistema propuesto, donde es posible observar la interacción entre el sistema de control y el de comunicaciones. El bloque control implementa un control colaborativo distribuido para la formación de robots móviles, mientras que, el sistema de comunicaciones implementa una red tipo FANET utilizando un protocolo de enrutamiento AODV y comunicación inalámbrica 802.11b (WLAN).

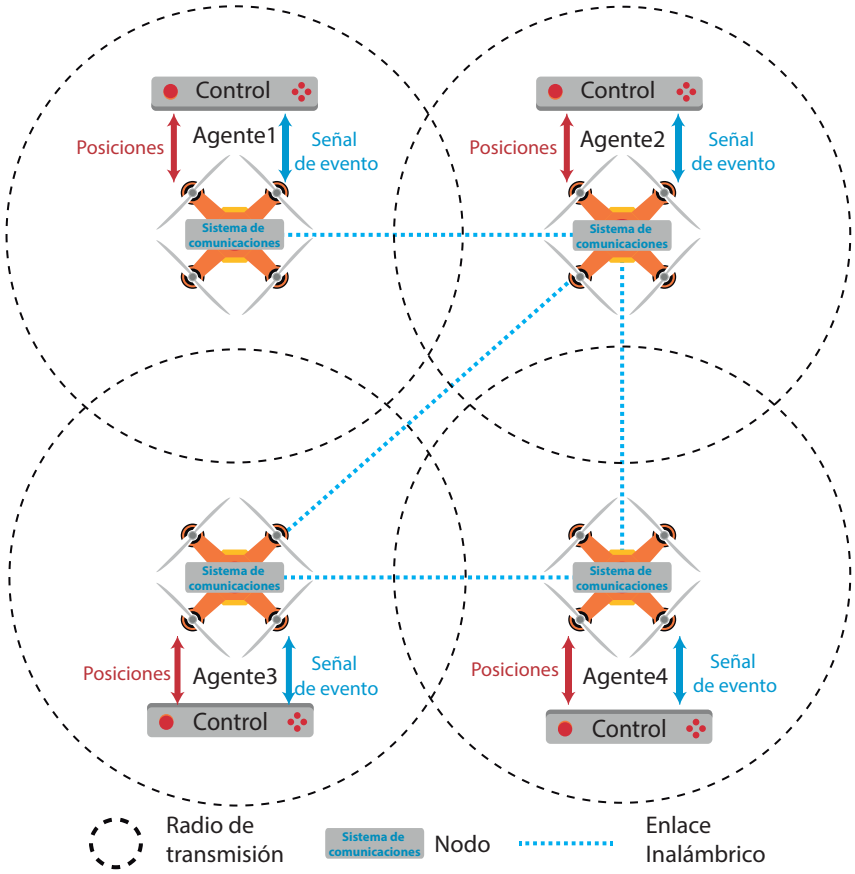


Figura 5.20. Diagrama general de la plataforma de simulación en Simulink.

El sistema de control se muestra en el apéndice B. Este se basa en un control cooperativo entre los 4 agentes que forman la red para cumplir un objetivo en común[40]. En este caso, los agentes parten de una posición inicial y mediante un consenso convergen a un punto deseado. Para realizar esta tarea, los nodos (agentes) necesitan compartir información con sus nodos vecinos, la cual es enviada a través de la red mediante una señal de evento (o señal de envío) generada por el control. La fig. 5.21 muestra un ejemplo de las señales de

envío generadas por el control.

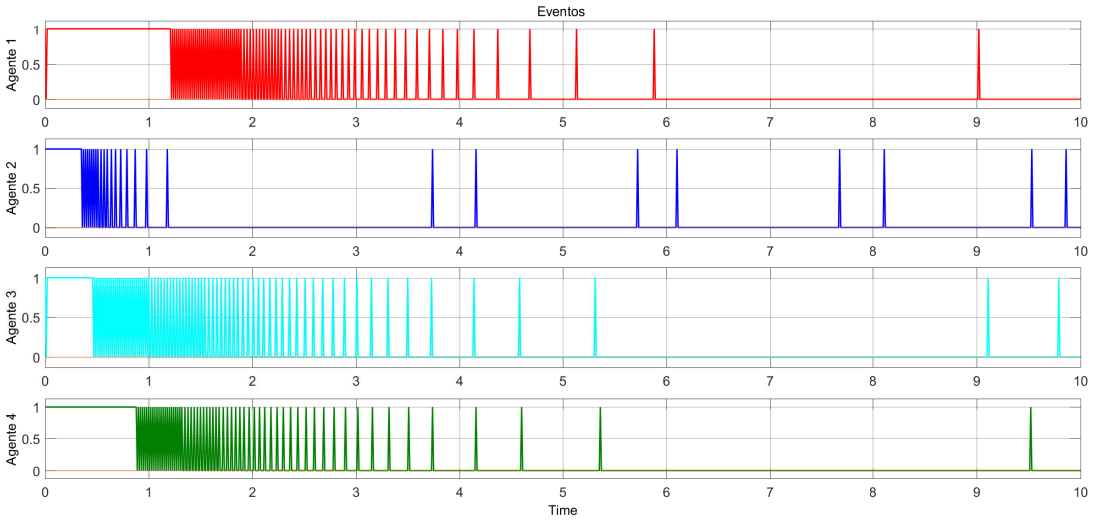


Figura 5.21. Solicitud de envío generado por cada agente.

La fig. 5.22 muestra el diagrama a bloques de la plataforma de simulación propuesta para una red tipo FANET bajo una estrategia de control cooperativo basada en TrueTime2.0 y Simulink (el diagrama original en simulink ha sido colocado en el apéndice B). Esta se encuentra compuesta por tres subsistemas: sistema de control, sistema de comunicación y un sistema de visualización de resultados con animación para el usuario.

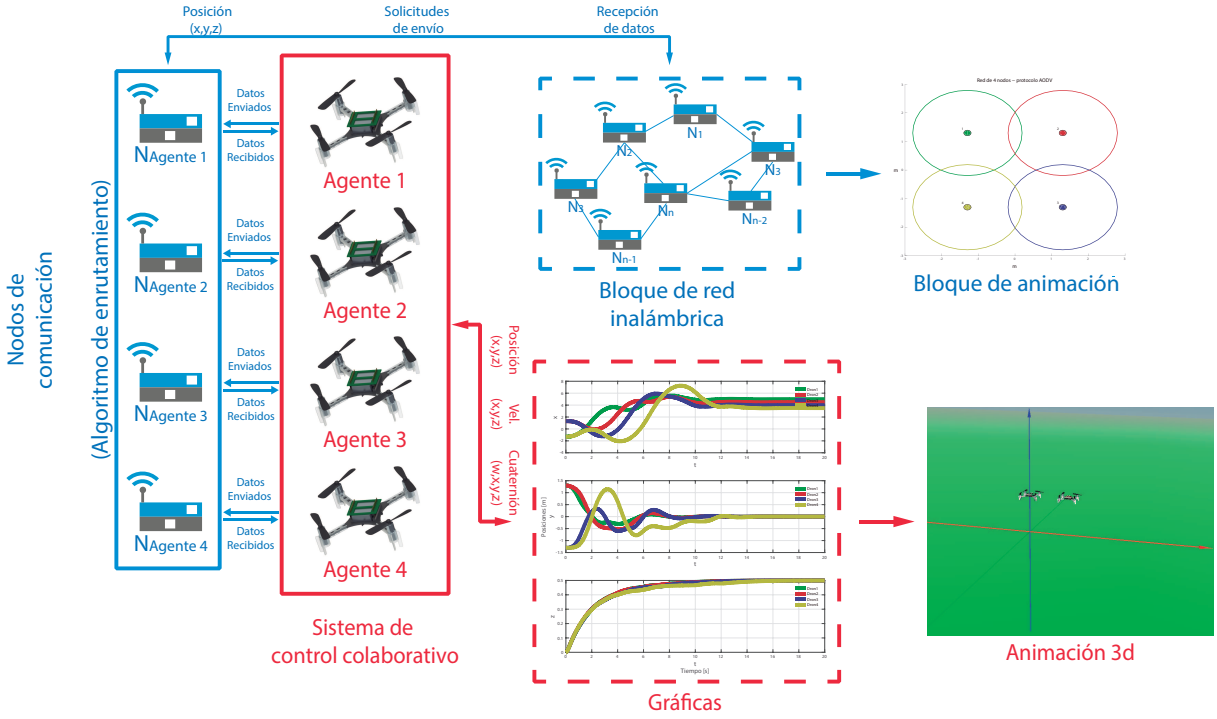


Figura 5.22. Diagrama a bloques de la plataforma de simulación en Simulink/Matlab.

Sistema de comunicaciones. Se encuentra indicado dentro de la linea azul, cuenta con: cuatro nodos móviles, un bloque de animación y el bloque de red inalámbrica. La fig. 5.23 muestra un diagrama general del sistema bajo una estrategia de control, mismo que funciona de la siguiente manera: cada nodo cuenta con su posición, identificador, datos que se requieren enviar (*Posicion* y *Velocidad*) y banderas de envío *Nodox*, tal y como se muestra en la fig. 5.24.

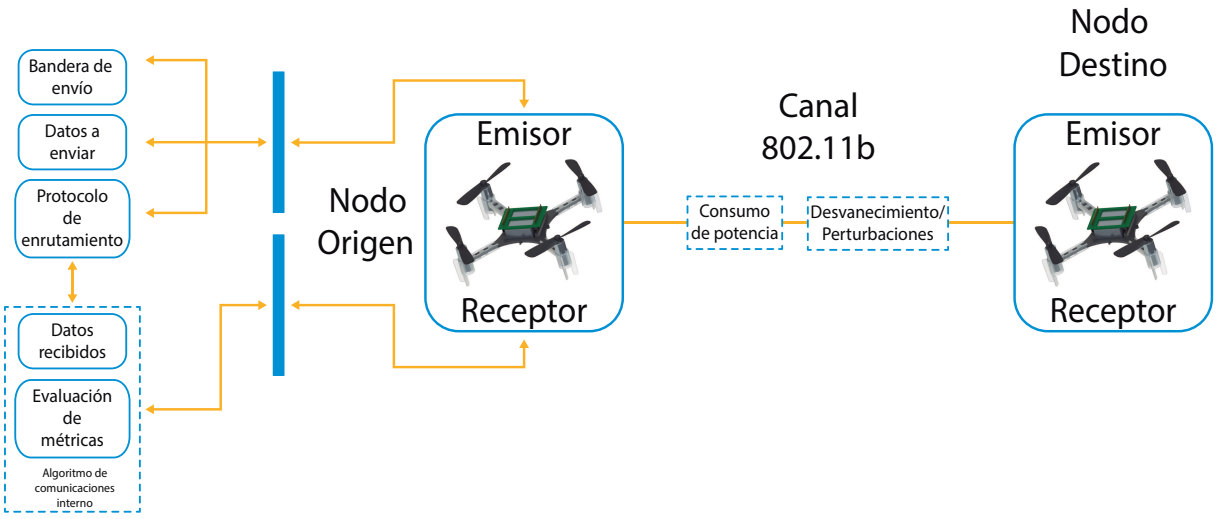


Figura 5.23. Diagrama a bloques del sistema de comunicaciones.

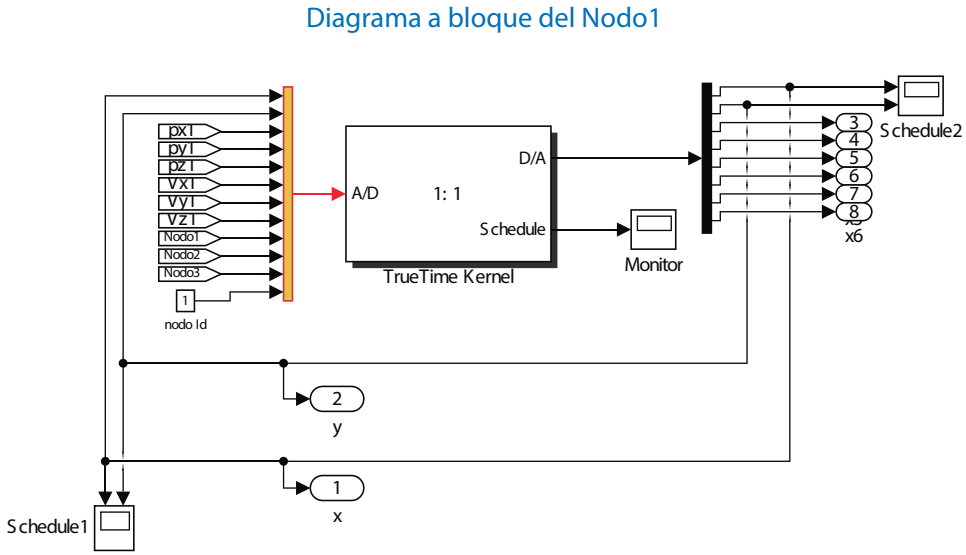


Figura 5.24. Diagrama de un nodo.

Una vez que se genera una solicitud de envío, el sistema de comunicaciones realiza el envío de los datos *Posicion* y *Velocidad* de acuerdo a lo solicitado. Para ello se ejecuta una rutina de descubrimiento de ruta mediante el algoritmo de enrutamiento AODV. Cabe mencionar que los envíos de datos más sencillos y rápidos se presentan cuando un solo nodo transmite en un intervalo de tiempo. Sin embargo; esto no

sucede en la mayoría de los casos. De la fig. 5.21 podemos observar que hay instantes de tiempo donde múltiples agentes desean transmitir simultáneamente. Por esta razón, es necesario crear ventanas que permitan a cada nodo transmitir en cierto tiempo y para evitar colisiones.

Sistema de control. Se encuentra indicado dentro de la linea roja, este cuenta con dos sistemas de control (cada uno contiene su control colaborativo, su control de rotación y su control de traslación) con sus entradas y salidas correspondientes. El sistema funciona de la siguiente manera: este recibe los datos de sus vecinos de acuerdo a la topología mostrada en la fig. 5.20, aplica un control colaborativo del tipo PID y decide si es necesario enviar su estatus o solicitar nueva información de sus vecinos.

Sistema visual. Se encuentra indicado dentro de la linea verde, este sistema es el encargado de mostrar el comportamiento del sistema mediante gráficas y una animación diseñada en el software ***Vrealm*** de Matlab, el cual permite visualizar el comportamiento del dron en un ambiente 3D (rotación y traslación). De esta manera se busca tener una simulación con una afición cercana a la realidad.

5.2.2. Simulación de una FANET para el consenso de 4 agentes (Micro-ambiente)

Por cuestiones infraestructura, la implementación de la red se lleva a cabo en un ambiente controlado con distancias pequeñas (nivel laboratorio *indoor*). Por esta razón, las siguientes simulaciones son realizadas considerando las distancias con las que se realizarán dichas pruebas.
La primer simulación consta de 4 nodos móviles. Estos parten de posiciones iniciales y realizan una formación sobre el eje x (véase Tabla.5.6 y Fig. 5.30). Por otra parte, la tabla 5.7 muestra los parámetros considerados por el sistema de comunicaciones para ésta simulación.

Tabla 5.6. Posiciones en metros de las formaciones realizadas para el consenso de cuatro agentes

	Nodo 1		Nodo 2		Nodo 3		Nodo 4	
	x1	y1	x2	y2	x3	y3	x4	y4
Formación Inicial	-1.3	1.3	1.3	1.3	1.3	-1.3	-1.3	-1.3
Formación Final	5	0	4.5	0	4	0	3.5	0

Tabla 5.7. Parámetros considerados para la simulación de 4 agentes.

Parámetro	Valor
Tipo de red	802.11b
Número de la red	1
Número de nodos	7
Tasa de envío (bits/s):	1
Tamaño minino de la trama (bits):	512
Potencia del transmisor (dBm):	24.4994 (0.2818 W)
Umbral del la señal del receptor (dBm)	10.6312
Radio de alcance (m)	1.5
Exponente de pérdida de la ruta $\left(\frac{1}{distancia^x}\right)$	3.5
Tiempo de espera para el mensaje de confirmación (s):	0.00004
Limite de reintentos	5
Umbral de error del codificador	0.03

Una vez iniciada la simulación, el sistema comienza a enviar los datos cada vez que el control los requiera. El sistema genera un reporte de sus eventos, mismos que son mostrados en la terminal de Matlab (fig. 5.25). Comparando este reporte con los mostrados con anterioridad, es posible observar que se cuenta con un mensaje de datos más largo y de punto flotante. Esto se debe a que ya se está trabajando con los datos reales y no de prueba, es decir, los valores necesarios para que el control realice su función.

```
Existe un ruta establecida en la tabla de enrutamiento para esta solicitud ---> 3  5  2  7  6  4 <---
Aplicación | El Nodo#2 recibió los datos: 25           1           2     -1.12998     1.12664     0.108853     0.668614    -0.677155     0.184703          50 en el tiempo: 0.60384
Aplicación | El Nodo#4 recibió los datos: 26           3           4     1.21042     -1.21382      0.1111    -0.464901       0.4563     0.187943          50 en el tiempo: 0.60757
Tiempo: 0.65| Aplicación | El Nodo#2 desea transmitir al Nodo#1 los datos: 27           2           1      1.19038      1.19055     0.121706     -0.52036    -0.316545     0.188517          50 con un tamaño de: 50
Existe un ruta establecida en la tabla de enrutamiento para esta solicitud ---> 2  7  1 <---
Tiempo: 0.6501| Aplicación | El Nodo#2 desea transmitir al Nodo#3 los datos: 28           2           3      1.19038      1.19055     0.121706     -0.52036    -0.316545     0.188517          50 con un tamaño de: 50
Existe un ruta establecida en la tabla de enrutamiento para esta solicitud ---> 2  5  3 <---
Aplicación | El Nodo#1 recibió los datos: 27           2           1      1.19038      1.19055     0.121706     -0.52036    -0.316545     0.188517          50 en el tiempo: 0.65317
Aplicación | El Nodo#3 recibió los datos: 28           2           3      1.19038      1.19055     0.121706     -0.52036    -0.316545     0.188517          50 en el tiempo: 0.65458
>>
```

Figura 5.25. Reporte de eventos generados durante la simulación de control.

La simulación se evalúa en un lapso de 20 s con un costo computacional de alrededor de 20 min. La fig. 5.26 muestra la trayectoria de los nodos desde el punto de vista de comunicaciones, misma que, puede ser corroborada con las gráficas de posición generadas durante la simulación y que se muestran en la fig. 5.27.

La posición deseada para esta simulación fue de x=5, y=0 y z=0.5. Esta puede ser visualizada como la misión del sistema. Ahora, si bien existe solo un punto de consenso para todos los drones, el algoritmo de control considera una distancia de 0.5 m entre ellos para evitar colisiones. En base a la fig.5.27, es posible observa un consenso exitoso en cada uno de los ejes. Por otra parte, la fig.5.28 muestra como la velocidad en cada uno de sus ejes tiende a cero, lo que indica que los drones mantienen su posición.

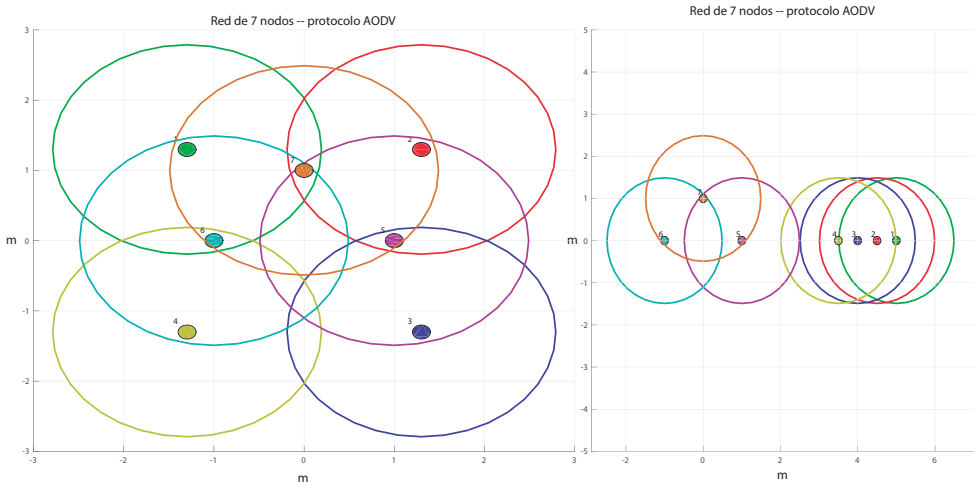


Figura 5.26. Animación del sistema de comunicaciones.

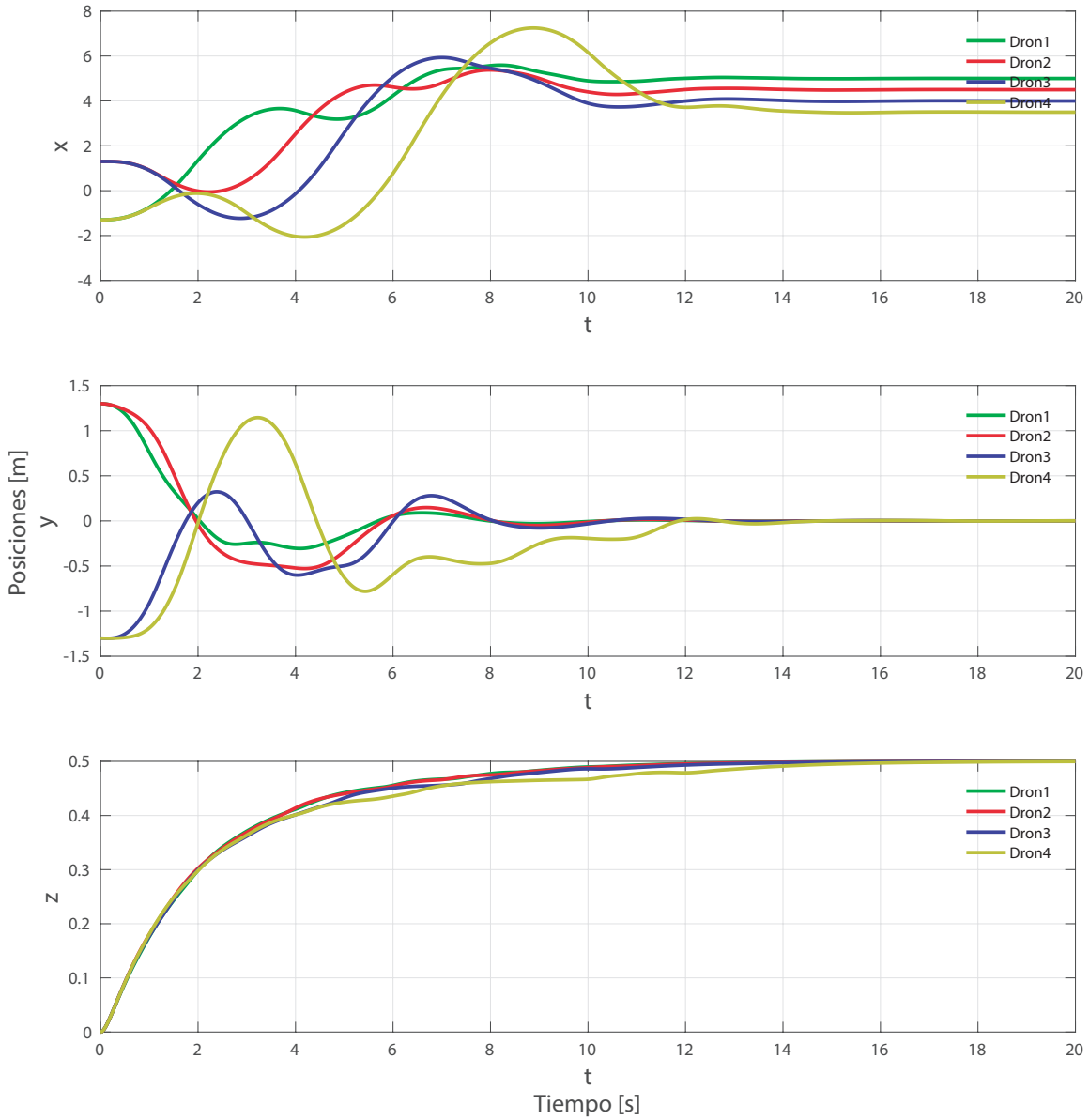


Figura 5.27. Señales de Posición en metros (ejes x,y,z).

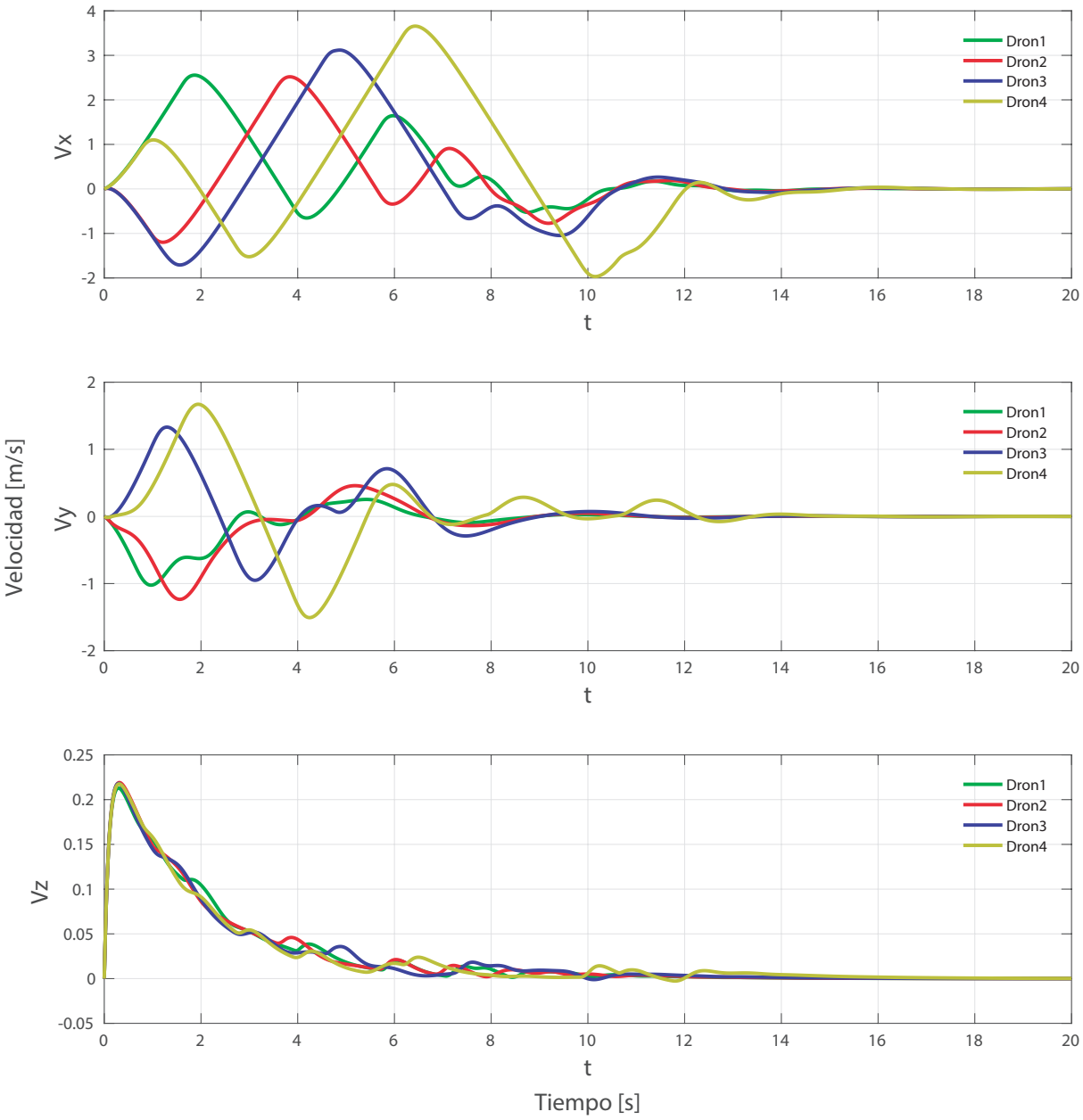


Figura 5.28. Señales de Velocidad en m/s (ejes x,y,z).

Por otra parte, la fig. 5.29 muestra el resultado de la misma evaluando solo el control (sin tomar en cuenta la parte de las comunicaciones), es decir, el sistema da por hecho que los datos siempre se encuentran disponibles al momento de realizar la evaluación. Comparando esta imagen con la obtenida en la plataforma desarrollada (fig. 5.27), es posible notar que la diferencia entre ellas es casi nula y ambas logran el consenso de manera exitosa.

La idea de los nodos de apoyo surge con la limitante de 1.5 m (este puede ser variado a través del umbral de la señal del receptor) del radio de cobertura establecido a cada nodo dentro de la red, ya que si se toman las posiciones iniciales establecidas sin nodos intermedios, los nodos principales (1,2,3 y 4) no sabrían de la existencia de sus nodos

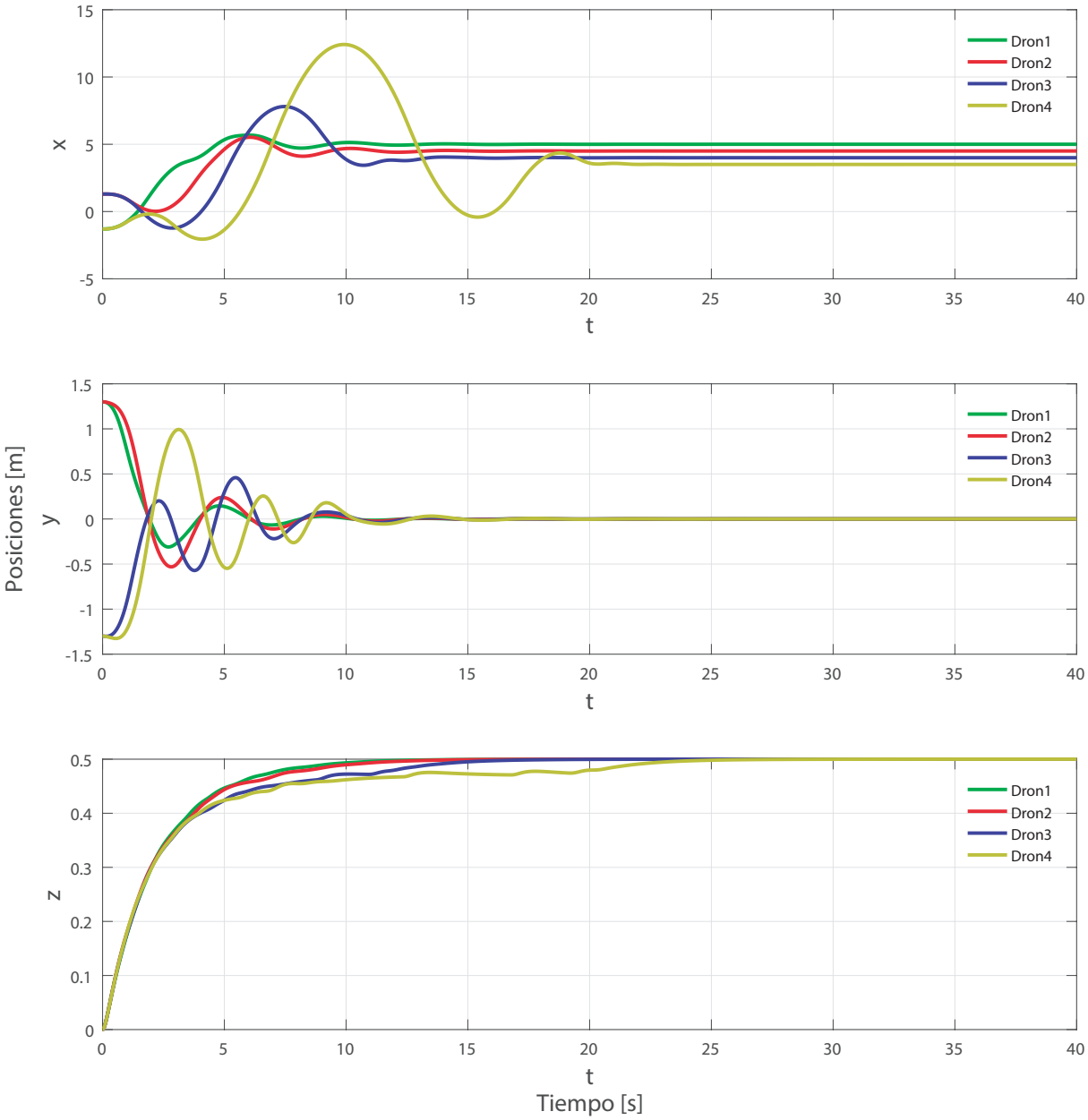


Figura 5.29. Señales de posición en metros generadas por el control sin comunicaciones.

vecinos. En base a estas condiciones, los nodos de apoyo solo fungen como puentes que permiten hacer uso de una de las principales herramientas de la red: el enrutamiento. De esta forma, cada uno de los nodos dentro de la red sabrá de la existencia de los otros nodos a pesar de que este no se encuentre dentro de su radio de cobertura. Vale la pena resaltar que la topología de la red del control no se ve afectada, ya que los únicos nodos que tienen participación son los móviles (1,2,3 y 4). Sin embargo, añadir más nodos a la red, demuestra su robustez ante el procesamiento de más datos, ya que en algún momento estos nodos de apoyo pudieran ser agentes móviles. Finalmente, la fig.5.30 muestra la animación en 3D de la simulación realizada, donde de manera más gráfica se demuestra como los drones cumplen con su misión.

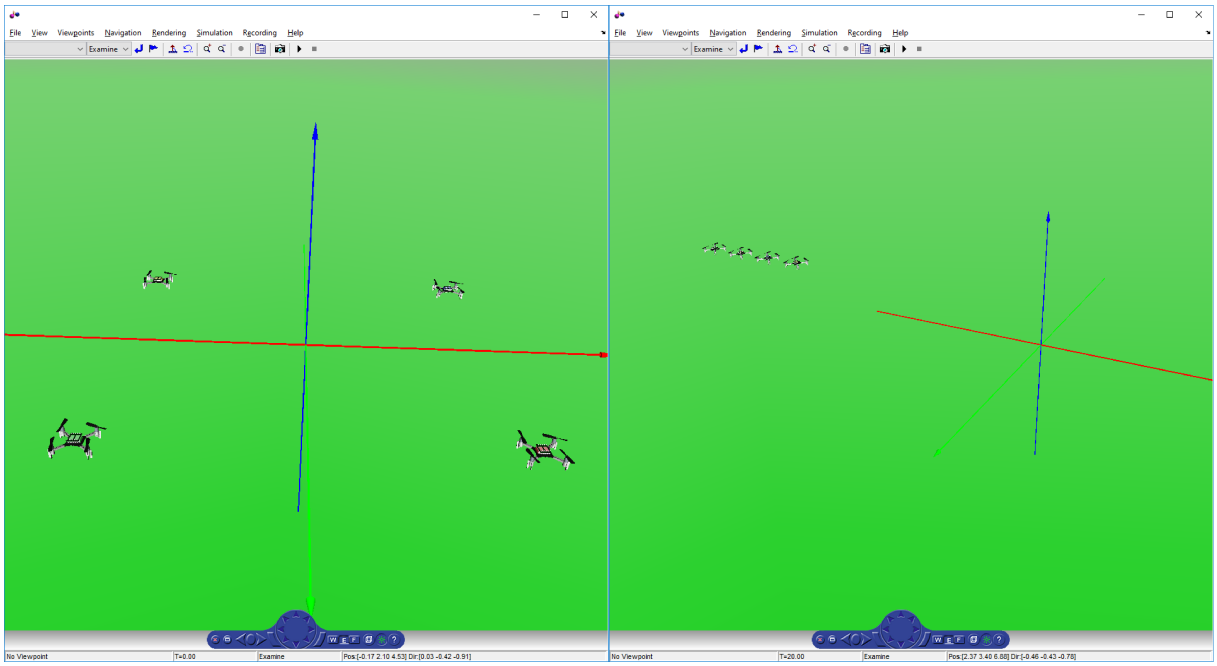


Figura 5.30. Simulación utilizando el toolbox VR.

5.2.2.1. Análisis de la comunicación

Además de mostrar los eventos de la red, esta plataforma genera tres vectores al término de la simulación: *Tabla de enrutamiento*, *Datos enviados* y *Datos recibidos* (véase la tabla 5.8). La idea de estos tres vectores, surgen con la necesidad de generar un reporte parecido al del ns2, con la finalidad de programar un algoritmo en *.m* que permita ejecutar las métricas de evaluación discutidas en la sección 3.2.

Tabla 5.8. Estructura del Vector *recibidos*

1	1	2	49.9942	20.004	5.69772	50	20	5.7	100	0.0002

Id: Id del paquete

Emisor: Indica el nodo emisor

Receptor: Indica el nodo receptor

Datos a enviar: Los siguientes seis campos llevan los datos requeridos por el control (*Posición* y *Velocidad*)

Tamaño del mensaje: Indica en bits el tamaño de los datos

Tiempo de entrega o envío: Muestra en segundos el tiempo en que se enviaron o se recibieron los datos

En base a los vectores generados en la simulación, se programaron en *.m* las métricas: tasa de paquetes entregados, retardo extremo a extremo y desempeño discutidas en 3.2. Para hacer uso de estas, solo basta teclear el comando : *Análisis* en la ventana de comandos de Matlab. Los resultados se muestran a continuación.

PDR [%]
La tasa de entrega de paquetes de la red es del 97.25 %

E2E [s]
Evaluación por nodo nodo1 = 35 ms , nodo2 =43.1 ms , nodo3 =39.7 ms, nodo4 =40.3 ms
Evaluación general de la red:

total =39.55ms

Thp [**Kbs**]
El Thp general de la red es de 30.498 Kbs

5.2.3. Simulación de una FANET para el consenso de 2 agentes (Micro-ambiente)

A continuación se muestra una simulación para un consenso de dos agentes, la cual será probado en el siguiente Capítulo de forma experimental. De esta manera, se busca tener una comparación de los resultados experimentales con los obtenidos en la simulación. Para esta parte solo se mostrarán los resultados obtenidos, teniendo en consideración que la plataforma ya ha sido descrita en secciones previas. Además, para esta simulación se toma en cuenta dos nodos móviles con un tercero fijo de apoyo (el tercer nodo simboliza la computadora con ROS, la cual sirve como puente para la comunicación entre los dos agentes).
La fig.5.9 muestra la formación inicial y final de esta simulación. Por otro lado, la posición deseada para el sistema es: x=0, y=0 y z=0.5 . Los parámetros considerados para esta simulación son los mismos mostrados en la tabla 5.26 (tomando en cuenta que el número de nodos es 3).

Tabla 5.9. Posiciones en metros de las formaciones realizadas para el consenso de dos agentes.

	Nodo 1		Nodo 2	
	x1	y1	x2	y2
Formación Inicial	0.2	-1.1	-0.5	0.5
Formación Final	0	0	-0.5	0

La fig. 5.31 muestra la trayectoria de los nodos desde el punto de vista de comunicaciones, misma que puede ser corroboradas con las gráficas de posición generadas durante la simulación o la interfaz de animación 3d, y cuyos resultados son mostrados en las figuras: 5.32 y 5.33 respectivamente.

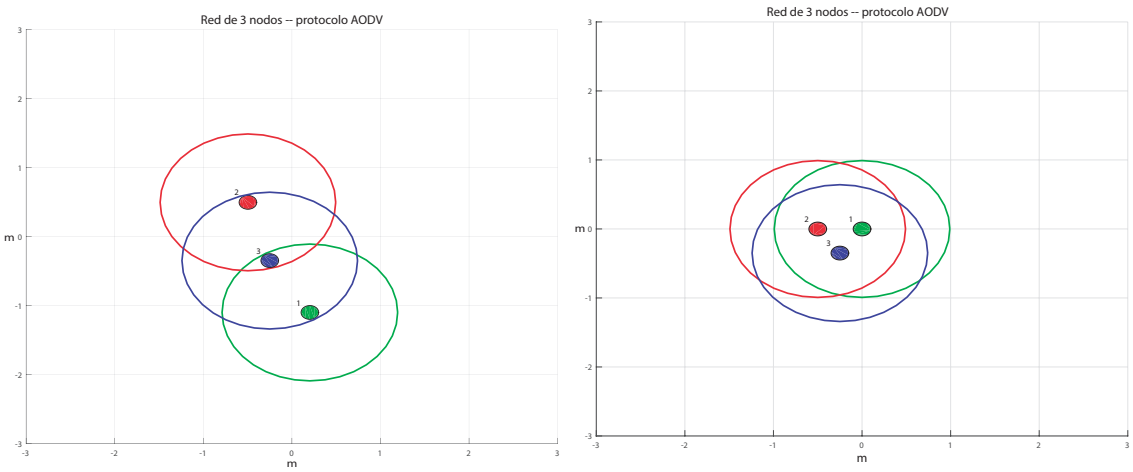


Figura 5.31. Animación del sistema de comunicaciones.

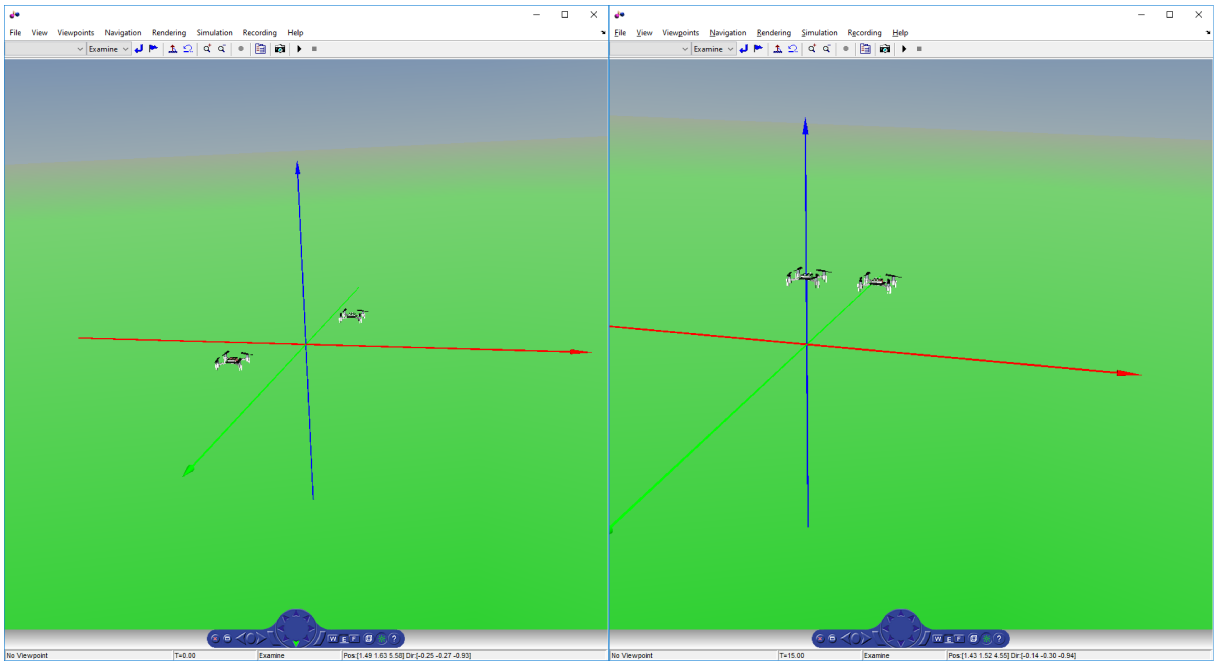


Figura 5.32. Simulación utilizando el toolbox VR.

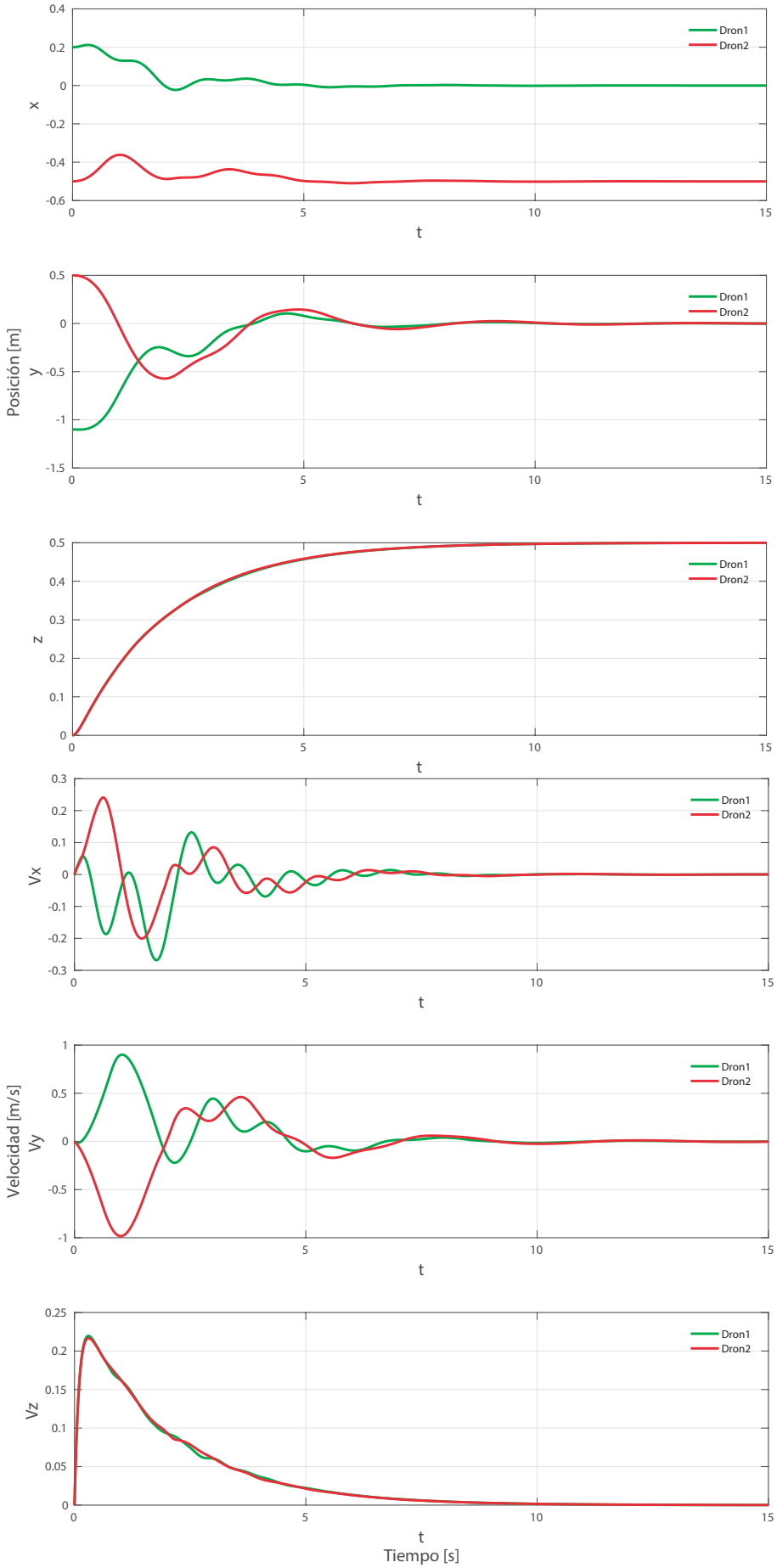


Figura 5.33. Señales de Posición y Velocidad (ejes x,y,z).

5.2.3.1. Análisis de la comunicación

En base a los vectores generados en la simulación, se programaron en *.m* las métricas: tasa de paquetes entregados, retardo extremo a extremo y desempeño discutidas en 3.2. Para hacer uso de estas, solo basta con teclear el comando : *Análisis* en la ventana de comandos de Matlab. Los resultados se muestran a continuación.

PDR [%]
La tasa de entrega de paquetes de la red es del 99 %

E2E [s]
Evaluación por nodo nodo1 = 5 ms , nodo2 =7.6 ms
Evaluación general de la red:

total =6.3ms

Thp [**Kbs**]
El Thp general de la red es de 116.89 Kbs

Los resultados obtenidos en ambas simulaciones han resultado satisfactorios. Esto se debe en gran parte que los tiempos en los que el sistema converge, se encuentran dentro de los estimados por el propio control, por lo que es posible decir que se cuenta con una buena plataforma de simulación que permite integrar sistemas de comunicaciones inalámbricas con sistemas de control y dar una buena aproximación a la realidad.

Capítulo 6

Desarrollo de una plataforma experimental para pruebas de algoritmos de Control de VANTs

En éste capítulo se presenta el desarrollo de una plataforma experimental para pruebas de algoritmos de Control de VANTs basada en ROS (Sistema Operativo Robótico). De esta manera es posible comprobar la propuesta mostrada en el capitulo anterior utilizando el control propuesto en el Capítulo 4.

6.1. ROS (Sistema Operativo Robótico)

ROS (Robot Operating System) provee librerías y herramientas para ayudar a los desarrolladores de software a crear aplicaciones para robots. ROS provee abstracción de hardware, controladores de dispositivos, librerías, herramientas de visualización, comunicación por mensajes, administración de paquetes y más. ROS está bajo la licencia open source, BSD. ROS es similar en algunos aspectos a los “frameworks de robots", como Player, YARP, Orocos, CARMEN, Orca, MOOS y Microsoft Robotics Studio [41]. Actualmente, ROS es la base fundamental de una comunidad en crecimiento. El total de líneas de código de más de 14 millones (puede ser utilizado con C ++ y Python), ha habido 2477 autores con 181509 confirmaciones (con un promedio de 73.3 por autor). Por otro lado, ROS-Industrial se creó para proporcionar beneficios clave para la industria. Desde 2007, cuando se lanzó el marco, la comunidad ROS ha crecido con al menos dos dígitos año tras año. La creciente popularidad ha traído varios grupos de intereses especiales para aplicaciones específicas. Uno de estos grupos es el ROS-Industrial Consortium [42].
El conjunto de herramientas y librerías de ROS proporciona[43]:

Mecanismos de comunicación entre programas. Es un estándar para comunicar entre sí diferente programas de un mismo sistema, ya sea en el mismo compu-

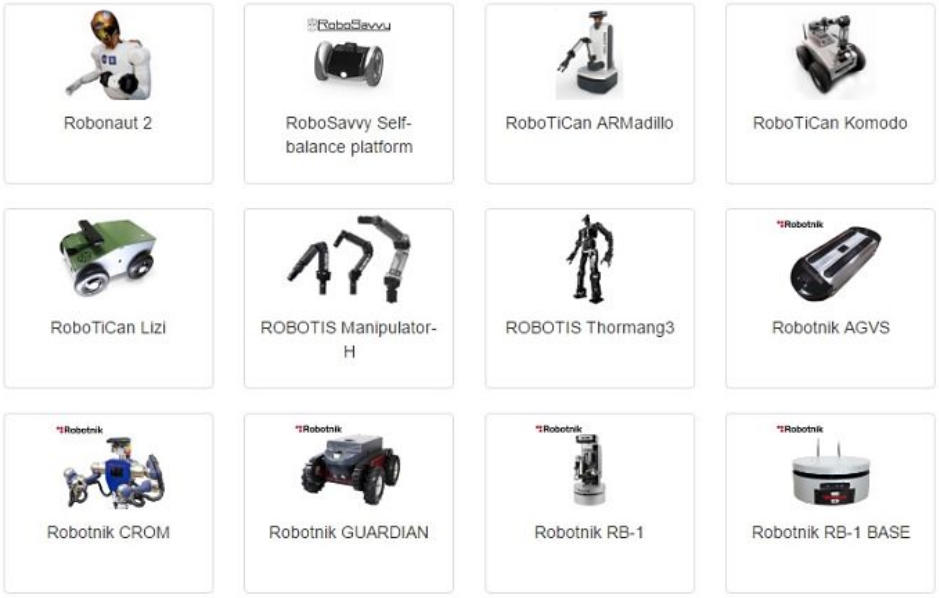


Figura 6.1. Aplicaciones con ROS.

tador o en varios computadores. La computación distribuida es un recurso muy común donde pequeñas partes de un robot trabajan por separado para conseguir un objetivo común.

Reusabilidad de código. Los paquetes estándar proporcionados en las distribuciones de ROS implementan muchos de los algoritmos comúnmente usado en robótica que ya han sido depurados y usados de forma estable. Además, el modelo de comunicación y gestión de mensajes de ROS se ha convertido en estándar y muchas de las plataformas robóticas ya implementan interfaces para ser usadas directamente desde ROS. Además de los paquetes estándar, existen muchos otros en la comunidad que implementan interfaces para sus sistemas (robots, sensores, librerías, etc).

Prueba rápida. Debido al diseño de comunicación por paso de mensajes, ROS nos permite simular muchos de los dispositivos con los que el sistema trabajará, de esta forma es posible aislar la funcionalidad del sistema del código de comunicación entre las diferentes partes del sistema, sensores y actuadores. Uno de los aspectos claves al desarrollar una aplicación es la capacidad de repetir los experimento y simular los sensores y actuadores que nos permite crear conjuntos de prueba.

En general, ROS fue diseñado específicamente para que distintos grupos pudieran colaborar y desarrollar conjuntamente el trabajo. Por ejemplo, un laboratorio puede tener expertos en el mapeo de ambientes interiores y podría contribuir con un sistema de clase mundial para producir mapas. Otro grupo podría tener expertos en el uso de mapas para navegar, y otro grupo podría haber descubierto un enfoque de visión por computadora que funciona bien para reconocer objetos pequeños en el desorden [41].

6.2. Conceptos

Ros tiene su propio acervo digital (*wiki*), el cual cuenta con una amplia documentación del sistema (que va desde la instalación del mismo hasta aplicaciones complejas como control colaborativo entre más de dos grupos) y en el cual pueden ser checados sus conceptos, librerías, herramientas visuales, etc. A continuación se describen algunos de los conceptos básicos que se consideran necesarios para comprender la plataforma desarrollada, sin embargo, para mayor información se recomienda visitar la pagina oficial de ROS (http://wiki.ros.org).

Nodo (node). Un nodo es un proceso o modulo que realiza algún tipo de computación en el sistema (programado en C++ o Python). Los nodos se combinan dentro de un grafo, compartiendo información entre ellos, para crear tareas complejas. Por ejemplo, un nodo puede controlar algún sensor, actuador u otro dispositivo necesario para un robot.

Mensaje. Básicamente, es una estructura de datos estrictamente tipada para comunicación entre nodos.

Por ejemplo, geometry_msgs/Twist expresa la velocidad en el espacio libre dividido en sus partes lineales y angulares de la forma:

- Vector3 linear
- Vector3 angular

donde *Vector3* es otro tipo de mensaje definido como:

- float64 x
- float64 y
- float64 z.

Tópico (topic). Son canales de información entre los nodos (donde se publican los mensajes). De esta manera, un nodo puede publicar o suscribirse a un tópico. Por ejemplo, se tiene un nodo llamado *posición*, el cual emite un tópico que es la odometría del robot. De esta manera, cualquier nodo se puede suscribir, sin embargo, el nodo que emite no controla quién está suscrito. La información es, por tanto, unidireccional (asíncrona). Si lo que queremos es una comunicación síncrona (petición/respuesta) debemos usar servicios.

Paquete (package). ROS está organizado en paquetes. Un paquete puede contener un nodo, una librería, conjunto de datos, o cualquier cosa que pueda constituir un módulo. Los paquetes pueden organizarse en pilas (stacks).

6.3. Descripción de la arena de vuelo

La fig.6.2 muestra el diagrama general de la arena de vuelo desarrollada. Está consta de cuatro cámaras OptiTrack para robots, cuatro drones crazyflie2.0, una computadora con los software Motive y Ros índigo (Ubuntu 14.04 LTS).

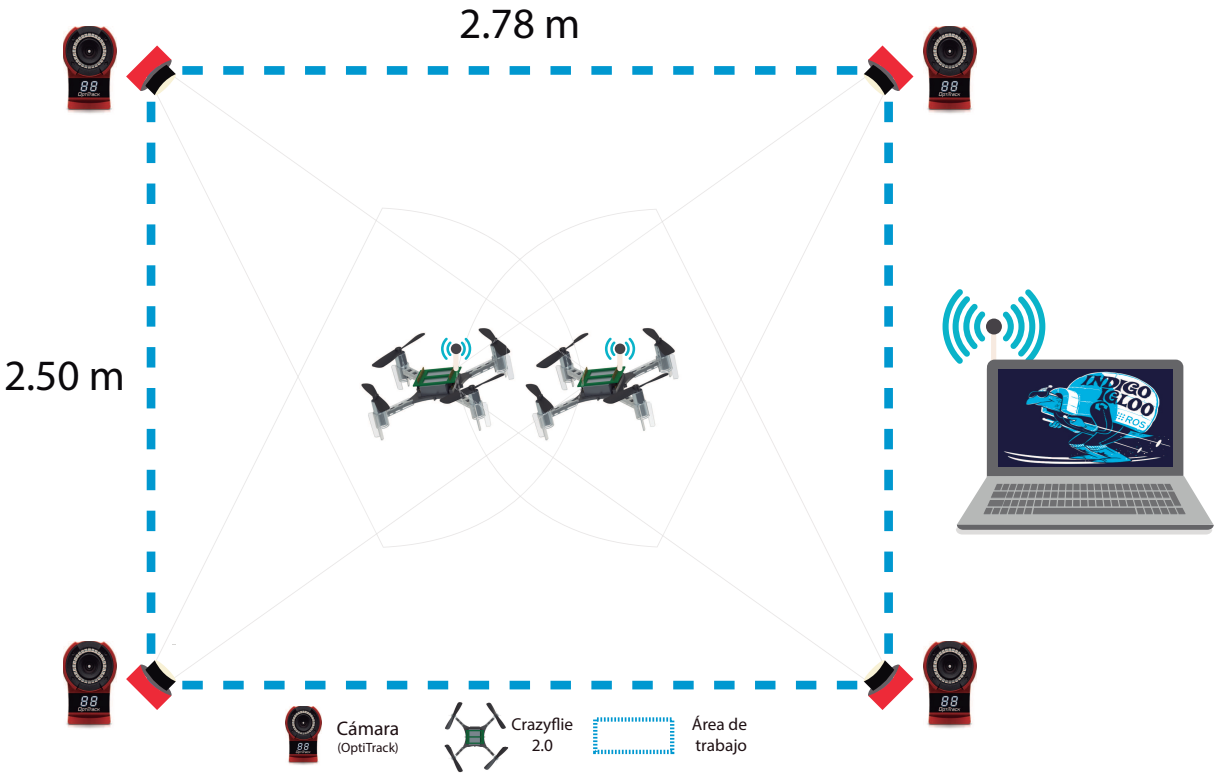


Figura 6.2. Arena de vuelo desarrollada.

6.3.1. OptiTrack

OptiTrack™ es el proveedor de captura de movimiento más grande del mundo y ofrece un seguimiento óptico de alto rendimiento a los precios más asequibles de la industria. La línea de productos OptiTrack incluye software de captura de movimiento y cámaras de rastreo de alta velocidad, así como servicios de ingeniería por contrato. Utilizados por instalaciones de todo el mundo en una variedad de mercados que van desde películas y juegos hasta entrenamiento deportivo y biomecánica, los clientes de OptiTrack incluyen lo mejor en sus respectivos campos. Ademas, OptiTrack ofrece una opción para robótica con una buena precisión en el seguimiento de objetos en el interior de seis grados de libertad (6 DoF) aérea y en tierra (UAV) (véase fig. 6.3)[44].

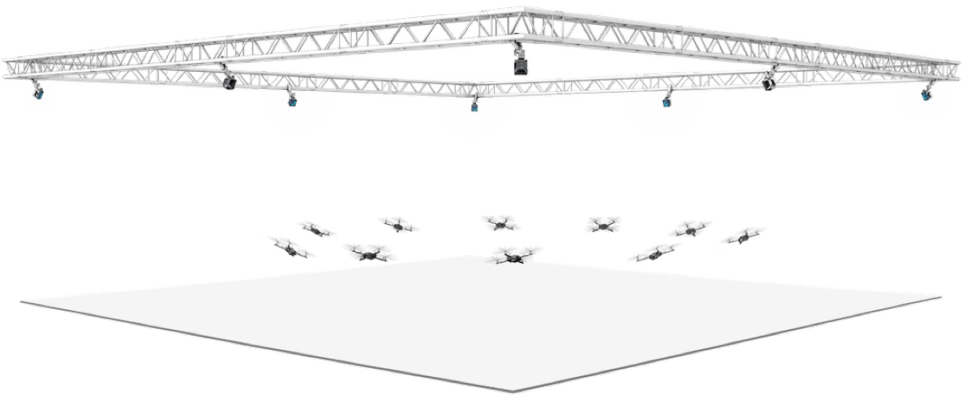


Figura 6.3. Sistema OptiTrack.

6.3.1.1. Motive

Es el software oficial del OptiTrack, el cual captura de movimiento óptico a partir de algunos marcadores colocados en el objeto (cuerpo) que se desee capturar. La fig.6.4 muestra el software y la configuración utilizada para la plataforma desarrollada. Los campos más relevantes en esta son:

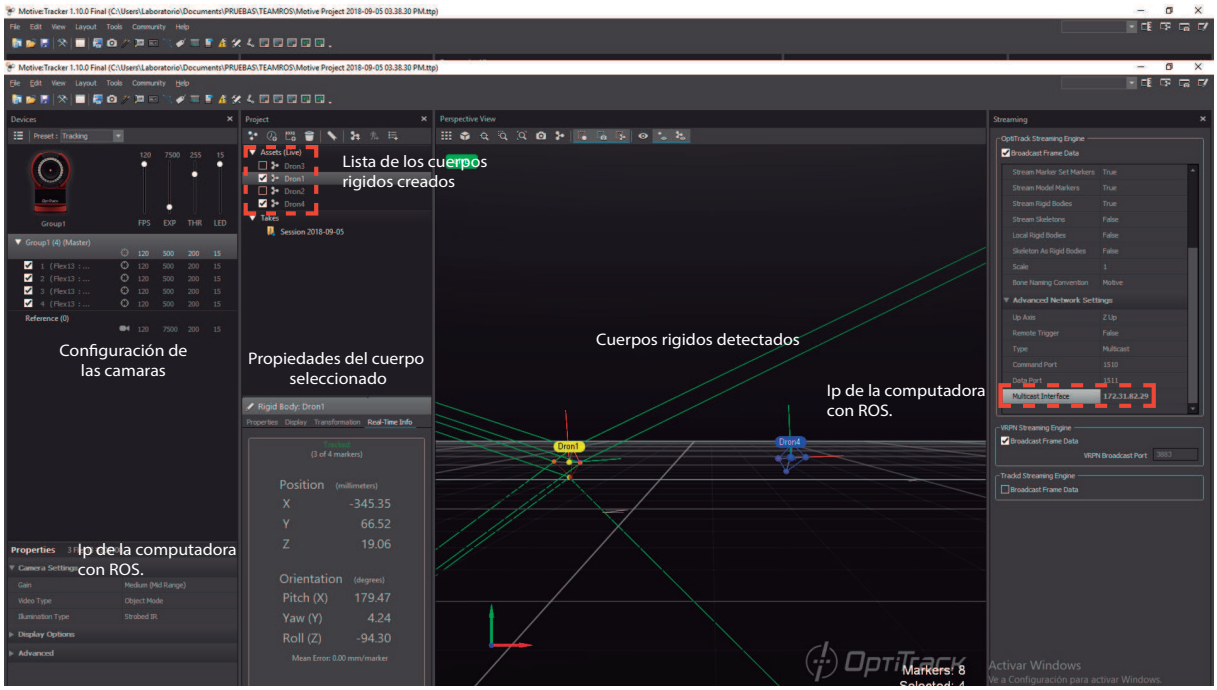


Figura 6.4. Software Motive.

Configuración de cámaras. Permite variar la sensibilidad, resolución y la tasa fps (frames per second) de cada una de las cámaras.Una vez configuradas correctamente es posible empezar la calibración del sistema.

Cuerpos rígidos Una vez inicializado el programa, éste detectará los marcadores dentro del área de cobertura, sin embargo no creara un cuerpo rígido. Para hacer esto

es necesario seleccionar un grupo de marcadores, dar un clic derecho y seleccionar la opción: *create rigidBody*. Una vez creado el cuerpo, en el lado izquierdo del programa se genera una lista de los mismo, además de que los marcadores ahora aparecen agrupados, con un sistema de referencia, una etiqueta y un color único. Una de las características de este listado es que permite solo visualizar los cuerpos con los que desea trabajar, haciendo que los demás permanezcan ocultos para el programa. La fig.6.5 muestra los marcadores utilizados en cada uno de los drones. Vale la pena señalar que la posición de los marcadores debe de variar en cada dron, ademas de ser asimétrica. Esto con la finalidad de evitar errores en la captura de la posición de los mismos.

Figura 6.5. Marcadores puestos en el dron.

Transmisión de datos Desafortunadamente, Motive se encuentra disponible solo para el sistema operativo Windows 7 y versiones posteriores. Por esta razón, este ofrece la posibilidad de transmitir los datos a través de una red local. Por lo cual es necesario activar el panel *Streaming* en la siguiente ruta: *View >Data Streaming*. A continuación, en el panel se selecciona la opción: *Broadcast Frame Data* y en *Multicast Interface*, se coloca la dirección IP de la computadora con la que se realizará la transmisión de los datos (véase fig. 6.4).

6.3.2. Crazyflie2.0

Crazyflie 2.0 es un dron pequeño cuadrirrotor de apenas 92 x 92 x 29 mm. y un peso de 31 gramos. También, cuenta con un sensor de medida de inercia (IMU) de 10 grados de libertad distribuidos (10DoF) de la siguiente manera: acelerómetro (3DoF), giroscopio (3DoF). magnetómetro (3DoF) y un barómetro (1DoF).
Una de las ventajas de trabajar con Crazyflie2.0 es que su programación es de código abierto, por lo cual existe una gran cantidad de información disponible en lenguajes: Java, Ruby, C/C++, C# y Javascript. Ademas, este cuenta con dos ARM: Cortex M4 (STM32F405) y M0 (nRF51822). El primero es el encargado de ejecutar los algoritmos de control para los actuadores del dron, mientras que el segundo es el encargado del sistema de comunicación de mismo (véase fig.6.6).

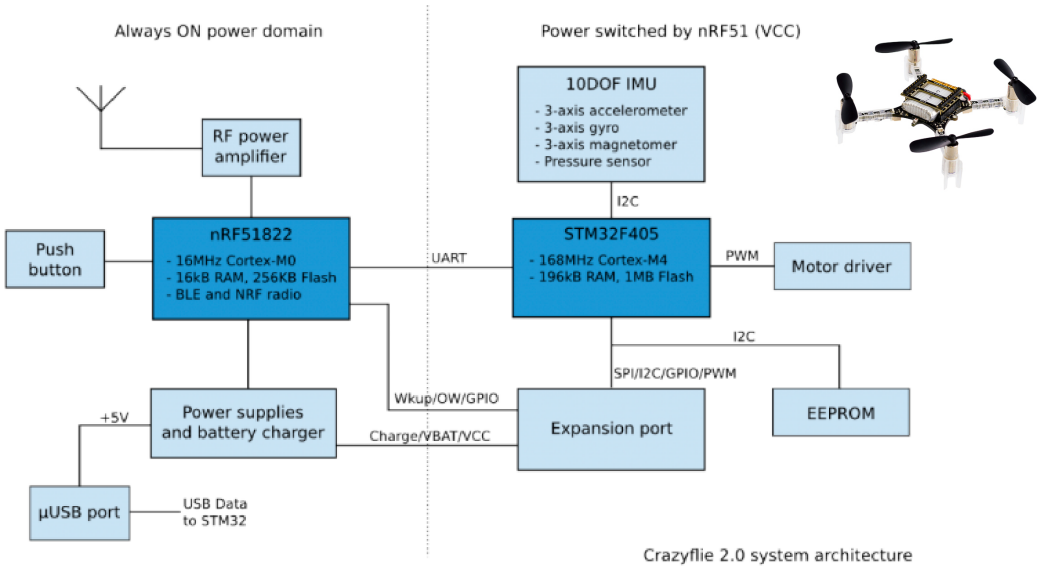


Figura 6.6. Arquitectura del Crazyflie2.0.

6.3.2.1. Conectividad

Crazyflie 2.0 ofrece dos opciones: La primera se trata de un Bluetooth de bajo consumo (BLE), que permite conexión con dispositivos iOS o Android mediante aplicación desarrollada por la empresa o bien alguna propuesta por el usuario. La segunda, es un transreceptor de 2.4GHz de alto rendimiento (basado en el nRF24LU1) llamado Crazyradio PA (véase fig. 6.7). Este cuenta con un amplificador de 20 dBm, que le permite una linea de visión de hasta 1 km, una tasa de velocidad de hasta 2 Mbps y conexión con 6 dispositivos a la vez con direcciones únicas [45]. En base a estas características, el Crazyradio PA fue el sistema de comunicación inalámbrica elegido para el desarrollo de la plataforma. Más información de este dispositivo puede obtenerse en la referencia [45].

Figura 6.7. Crazyradio PA.

6.3.3. Ros índigo

Una vez instalado Ros es necesario crear el espacio de trabajo *crazyflie_ws*, además de instalar el paquete de la Red Periférica de Realidad Virtual (VRPN) y la pila *crazyflie_ros*. Una es para obtener los valores obtenidos con el Motive y la otra contiene los drivers para

la comunicación con el Crazyflie2.0 respectivamente [46]. Estos paquetes, fueron instalados de la siguiente manera:

```
$ mkdir -p ~/crazyflie_ws/src
$ cd ~/crazyflie_ws/src
$ catkin_init_workspace
$ git clone https://github.com/whoenig/crazyflie_ros.git
$ cd ~/crazyflie_ws
$ catkin_make

$ cd ~/crazyflie_ws/src
$ sudo apt-get install ros-indigo-vrpn-client-ros
$ cd ~/crazyflie_ws
$ catkin_make
$ source ~/crazyflie_ws/devel/setup.bash
```

El objetivo de esta rutina es crear el espacio de trabajo *crazyflie_ws* y la ruta *crazyflie_ws >src*, donde deben de aparecer las carpetas: *crazyflie_ros* y *vrpn_client_ros*,tal y como se muestra en la fig. 6.8.

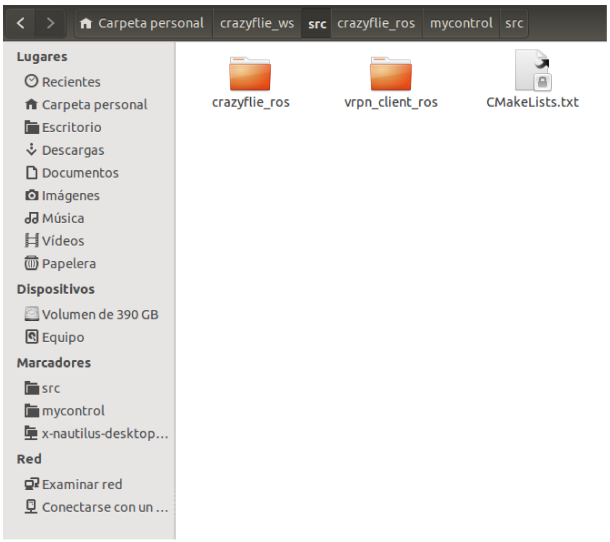


Figura 6.8. Paquete VRPN.

Generando el paquete *vrpn_client_ros*, es posible establecer la conexión con el Motive para obtener los valores de los cuerpos rígidos que se encuentren dentro de la arena. Para hacer esto, es necesario ejecutar el comando:

```
$ roslaunch vrpn_client_ros sample.launch server:= 192.168.0.123
```

donde *server* es la Ip de la computadora en la que se encuentra instalado el Motive. La fig.6.9 muestra la conexión con el Motive y la visualización de los datos a través de tres ventanas.

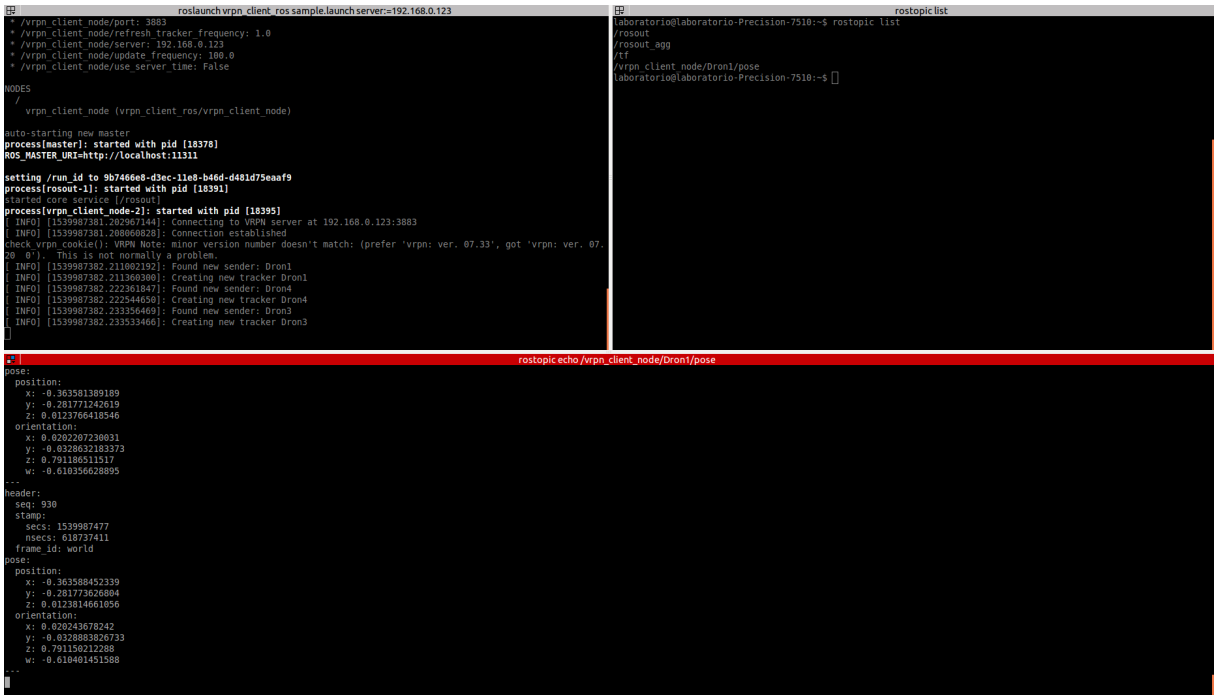


Figura 6.9. Conexión VRPN y visualización de los tópicos en ROS.

La primer ventana presenta el reporte de conexión con el Motive y la lista de los cuerpos rígidos. Como la conexión se realizó de manera exitosa, es posible visualizar los tópicos a los cuales se tiene acceso, para esto se ocupa el comando **$ rostopic list** mostrado en la segunda ventana. En este caso, se observa que el programa tiene acceso a cuatro tópicos, sin embargo el de mayor importancia es **/vrpn_client_node/Dron1/pose**, debido a que contiene la información de la posición y orientación en cuaterniones del Dron1. La última ventana, muestra el contenido del mensaje publicado en este tópico, siendo visualizado a través del comando: **$ rostopic echo** seguido del tópico que se desea leer. En este caso seria: **$ rostopic echo /vrpn_client_node/Dron1/pose**. Además, ROS posee una herramienta visual que te permite observar el estado de la conexión mediante un reporte detallado. Para esto, solo es necesario ejecutar el comando: **rqt_tf_tree** teniendo como resultado el diagrama de árbol mostrado en la fig. 6.10.

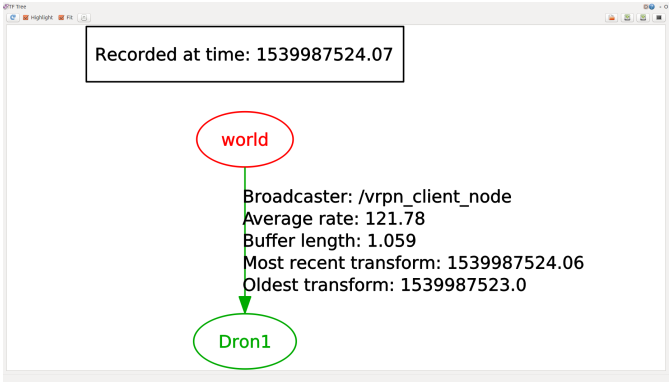


Figura 6.10. Estado de la conexión VRPN a través de rqt_tf_tree.

Después de visualizar los datos recibidos por el Motive, se percibieron dos problemas. El primero, fue que el motive presenta un sistema de referencia inercial diferente al proporcionado por el IMU del Crazyflie y el segundo, que algunos valores como de la posición u orientación no corresponden con lo mostrado en el Motive (estaban invertidos o con signo contrario).

La solución para el primero consiste, en una rotación de ejes desde la opción de calibración del Motive. La fig.6.11 muestra su configuración por defecto y la actual (con la que se trabaja). De forma que (x_0, y_0, z_0) y (x_1, y_1, z_1) tengan el mismo sentido. En este caso, el eje z mantiene la convención de la regla de la mano derecha (positivo hacia arriba). Otra solución que puede aplicarse seria la implementación de matrices de rotación, sin embargo estas ultimas pudieran demandar mas tiempo de procesamiento al control propuesto.

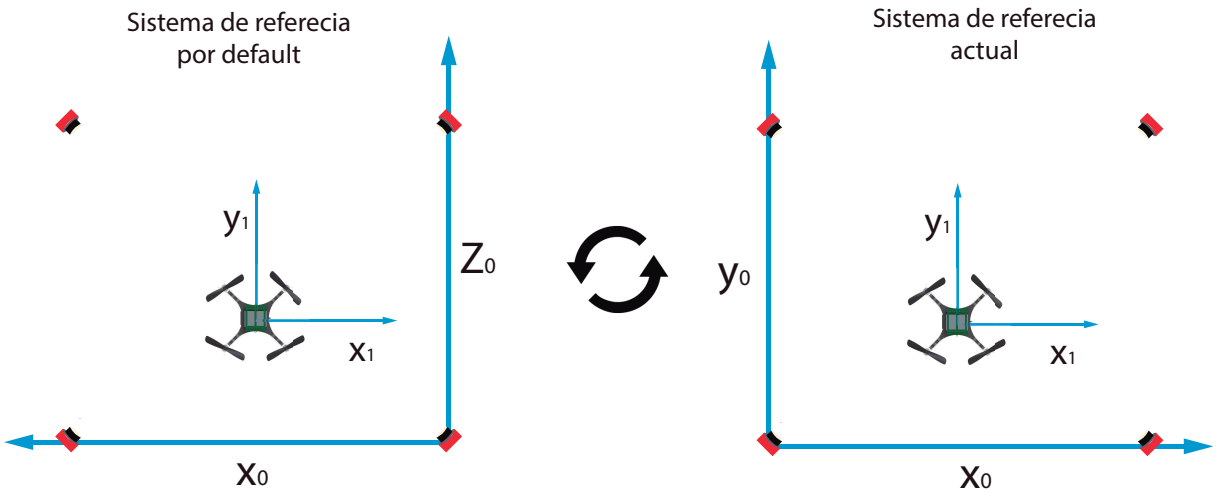


Figura 6.11. Sistema de referencia inercial de la arena.

En cuanto al segundo problema, es necesario realizar algunas modificaciones en la estructura del mensaje. Esto se hace en el archivo *vrpn_client_ros.cpp*, el cual puede ser encontrado en la ruta: *home >crazyflie_ws >src >vrpn_client_ros.*

6.3.4. Plataforma LabCA-FCE: ROS + Crazyflie2.0

LabCA-FCE: ROS + Crazyflie2.0 es una plataforma experimental basada en ROS, OptiTrack y Crazyflie2.0 que permite ejecutar algoritmos de control sobre VANTs. La fig.6.12 muestra el diagrama general a bloques de la misma.

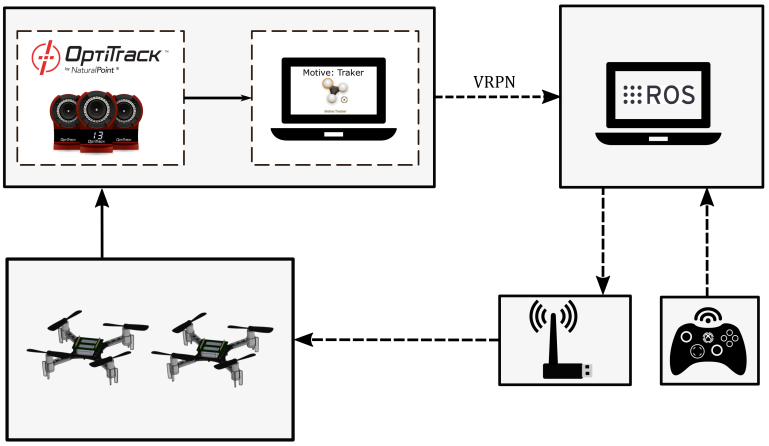


Figura 6.12. Diagrama general de plataforma.

Una de las ventajas que presenta esta plataforma, es que cuenta con un paquete llamado *mycontrol* (este es el núcleo de la plataforma, fig. 6.13). Este contiene los códigos y algoritmos del control, las ganancias, la comunicación con el Crazyflie y el Motive, etc. Sin embargo, los archivos más importantes de este paquete son: *Collaborativepid.hpp* y *controller.cpp*, ambos programados en c++. El primero contiene el algoritmo de control colaborativo tipo PID propuesto, mientras que el segundo es el encargado de realizar las suscripciones a los tópicos: VRPN, Joystick, Crazyflie, etc.

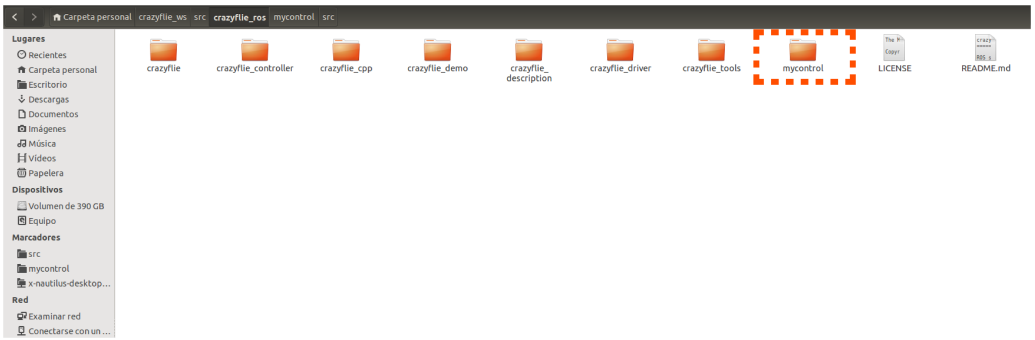


Figura 6.13. Paquete *mycontrol*.

El hecho de que estas dos funciones se encuentren separadas ayuda a que una infinidad de controles sean probados, es decir, si se desea cambiar el control solo bastará con remplazar el archivo: *Collaborativepid.hpp* por el control que se desee ejecutar. A continuación

se muestra la implementación de un control colaborativo tipo PID para dos agentes utilizando esta plataforma. El sistema trabaja de la siguiente manera: Dos Crazyflie2.0 son puestos en cualquier posición dentro de la arena. Un control de Xbox360 da la orden de inicio y final de la prueba, es decir, controla el despegue o aterrizaje de los drones. Si el botón X es presionado, los Crazyflie ejecutarán su tarea programada, por otra parte si el botón A es presionado estos interrumpen su tarea y aterrizan. Para iniciar la prueba, solo es necesario ejecutar en siguiente comando:

```
$ roslaunch crazyflie_demo Collaborative2.launch uri1:=radio
    ://0/100/2M/E7E7E7E701 frame1:=Dron1 uri2:=radio://0/100/2M/
    E7E7E7E704 frame2:=Dron4 x1:=0 y1:=0 z1:=0.4 server:=
    192.168.0.123
```

donde *uri1* y *uri2* son las direcciones únicas de cada Crazyflie para su comunicación con el *Crazyradio PA*. *Frame1* y *Frame2* son los nombres de las tramas que serán leídas por el VRPN (nombre de los cuerpos rígidos). *X1,Y1,Z1* son la posición deseada y por último *sever* es la dirección Ip de la computadora donde se encuentra instalado el Motive. Por otro lado, el archivo *Collaborative2.launch* asigna los frames leídos por el VRPN al nodo de su dron correspondiente, establece el punto de referencia (objetivo), crea los nodos controladores de cada dron. Básicamente, es el encargado de describir los nodos principales que serán ocupados.

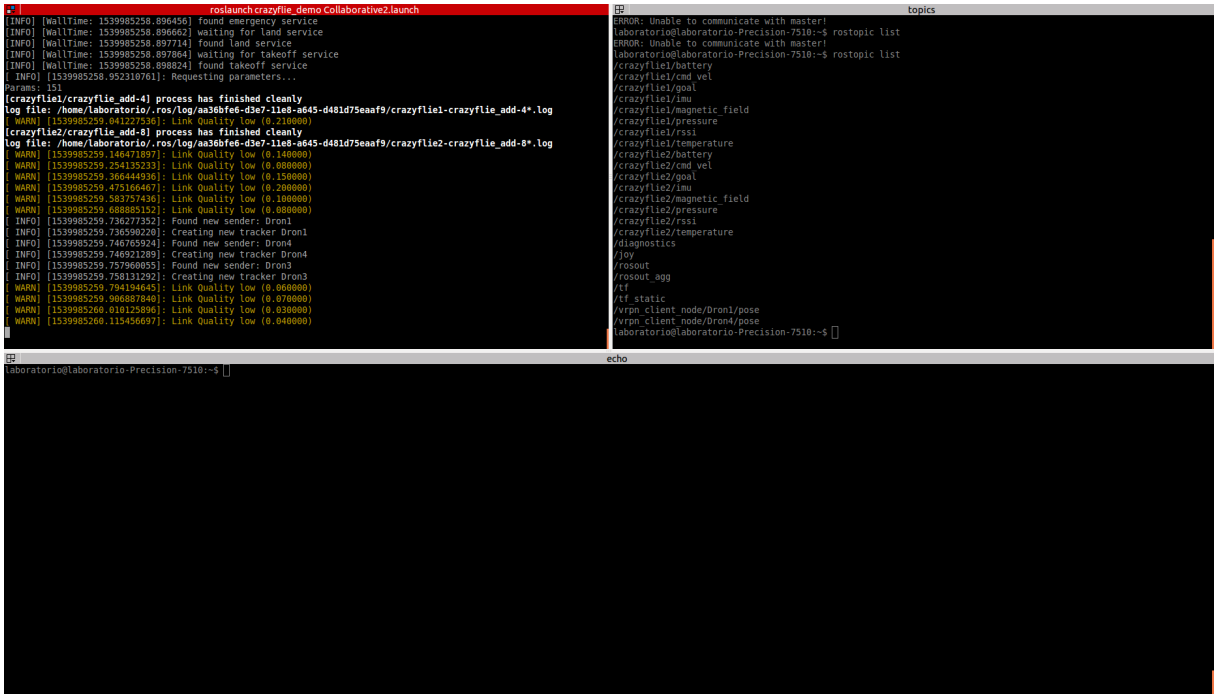


Figura 6.14. Ejecución del control colaborativo.

La fig.6.14 muestra la ejecución del programa y los tópicos a los que se tiene acceso. La tercera pantalla se ocupa para volver a compilar el programa en caso de que se requiera

un cambio a través del comando: *Catkin_make*.

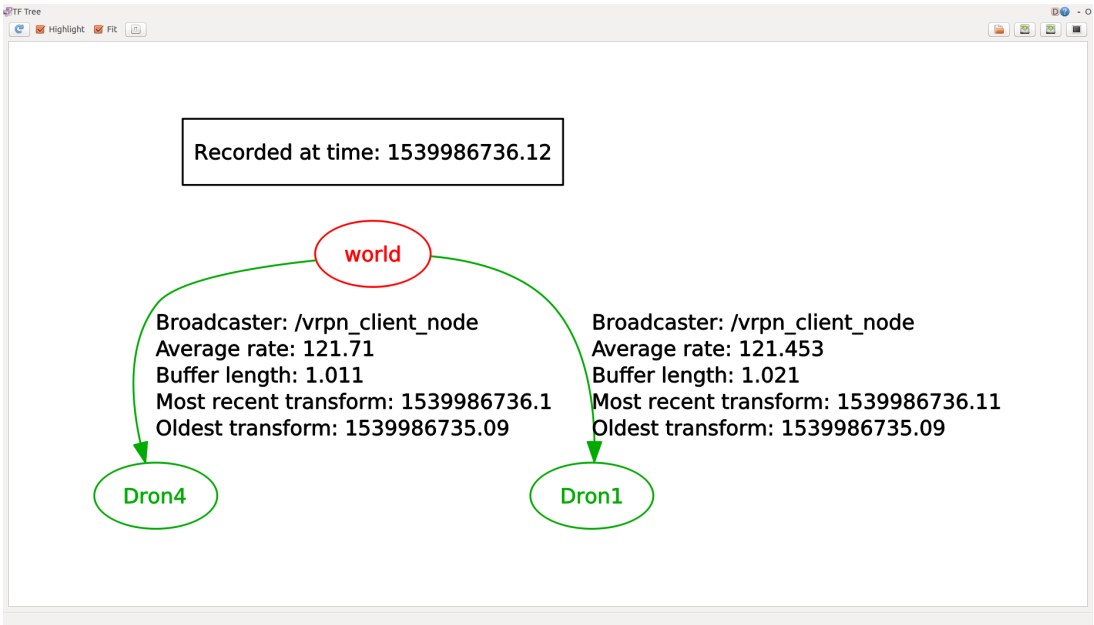


Figura 6.15. Estado de la conexión VRPN a través de rqt_tf_tree para dos nodos.

La fig. 6.15 muestra el estado de la conexión con el motive, mientras que la figura 6.16 muestra la distribución computacional de ROS. Esta permite visualizar los nodos y los tópicos involucrados en la simulación.

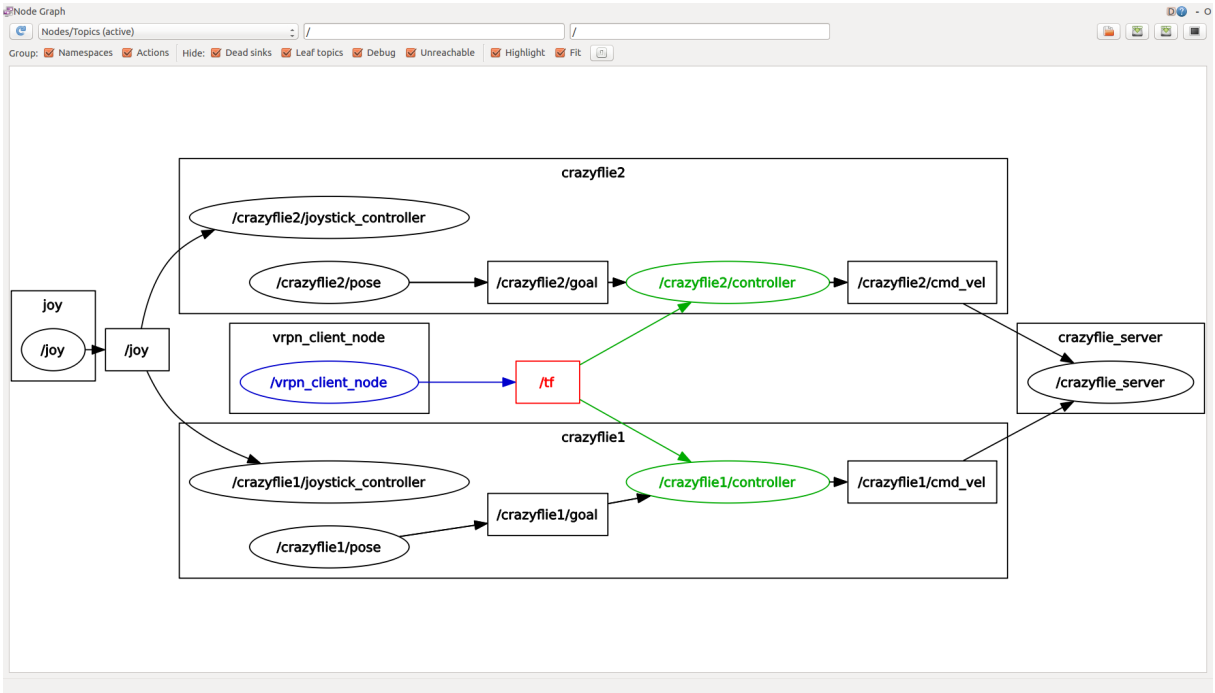


Figura 6.16. Distribución computacional del programa en ROS.

De esta forma, es posible observar el nodo *vrpn_client_node*, el cual se encarga de publicar la posición de los cuerpos rígidos en su tópico *tf*. Este a su vez cuenta con dos nodos subs-

criptores: *crazyflie2/controller* y *crazyflie2/controller*. En el nodo *Controller* se encuentra el algoritmo de control de cada agente (para esta simulación se aplicó el control mostrado en el Capítulo 5). Este necesita varios parámetros para su ejecución. De este bloque se puede que nodos cuentan con más suscripciones que otros, en otras palabras, estos son los nodos más importante del sistema. Una vez que el nodo de control ha implementado su algoritmo, este publica los valores de *roll*, *pitch*, *yaw*, y *thrust* en un tópico llamado *crazyflie/cmd_vel*, el cual se cuenta con un subscriptor (el nodo *crazyflie_server*). Este último nodo, es el encargo de establecer la comunicación con el Crazyflie, el cual recibe las posiciones deseadas de los parámetros *roll*, *pitch*, *yaw*, y *thrust* y ejecuta un control para obtenerlas. Este segundo control, se encuentra en el Crazyflie y se rige por su IMU. De aquí la importancia en el ajuste del sistema inercial de ambos sistemas.

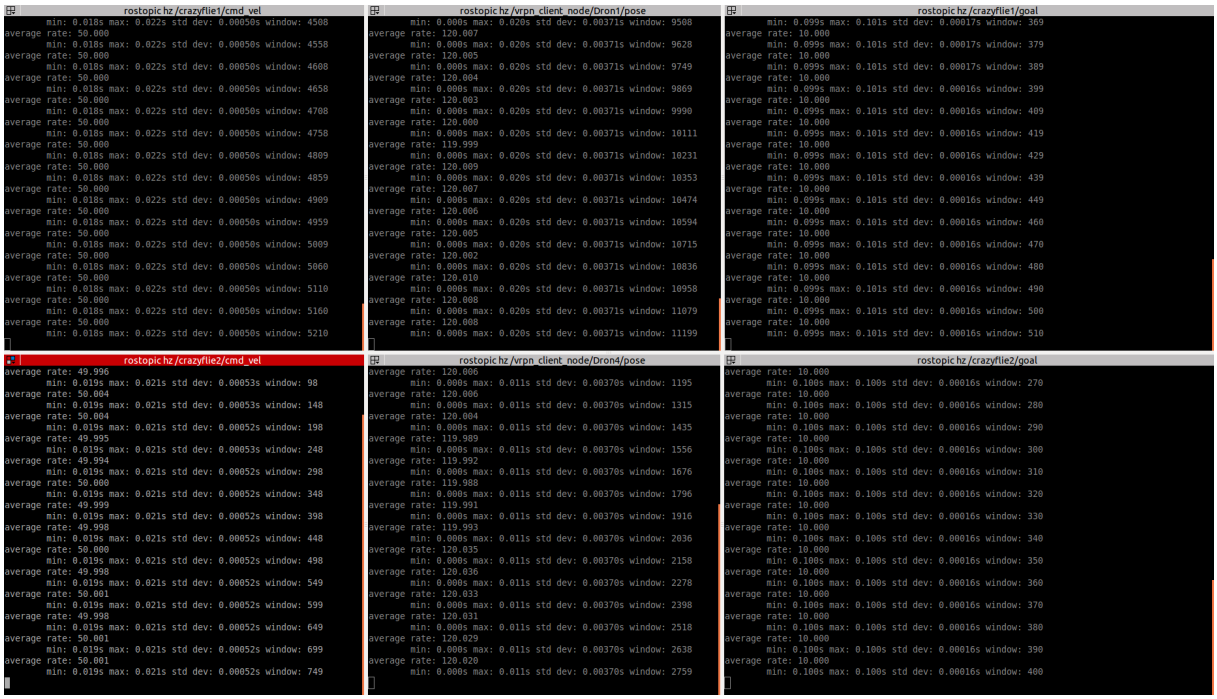
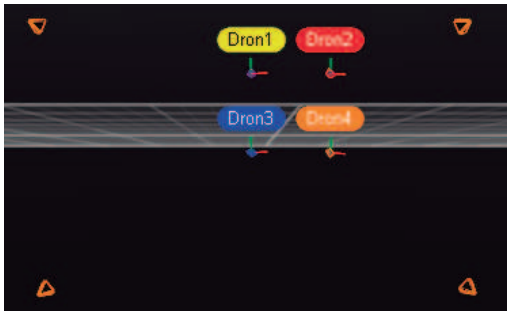


Figura 6.17. Frecuencia con la que llegan los mensajes a los tópicos.

Además, se realizó un análisis de tiempo para saber con que frecuencia se tienen los mensajes en los principales tópicos. La fig. 6.17 muestra los cuatro nodos fundamentales en ROS: *crazyflie1/cmd_vel*, *crazyflie2/cmd_vel*, *vrpn_client_node/Dron1/pose* y *vrpn_client_node/Dron2/pose*. En base a esto, es posible observar que los mensajes son publicados con una frecuencia de 50 hz (0.02 s), 49.996 hz (0.02 s) , 120 hz (0.008 s) y 120 hz (0.008 s), respectivamente. Asimismo, los nodos *crazyflie1/cmd_vel* y *crazyflie2/cmd_vel* son los tópicos donde se colocan los valores para el control del Crazyflie y de acuerdo con la frecuencia obtenida, es posible afirmar que esta satisface los 50 hz requeridos por el control (Véase Sección 3.1).

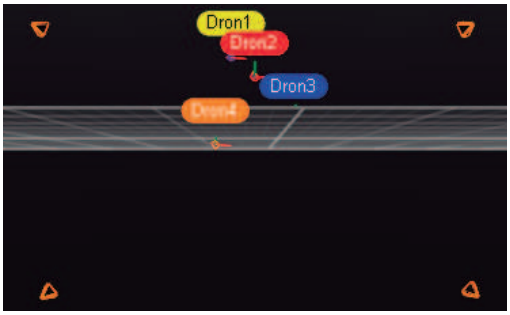
6.3.5. Enrutamiento de la LabCA-FCE

Además de ejecutar algoritmos de control, esta plataforma genera una tabla de enrutamiento en cada dron (nodo). De esta manera, el sistema sabe el camino que debe de seguir para el envío de los datos demandado por el control. El enrutamiento se probó en una región de $\simeq 7\ m^2$, por lo que se consideraron distancias máximas entre nodos de $1\ m$. Los modelos de propagación se ajustaron de acuerdo a las condiciones del interior de la arena pero estos pueden ser sustituidos si la implementación se cambia al exterior y considerando el respectivo incremento en las distancias. Para la implementación de las estrategias de enrutamiento y en virtud de que se esta omitiendo el uso de gps's, se cuenta con un sistema de rastreo visual mediante cámaras OptiTrack. De tal forma que se logra el proceso de enrutamiento nodo a nodo en la red bajo la topología especificada por el sistema de control como se observa en la fig. 5.20.



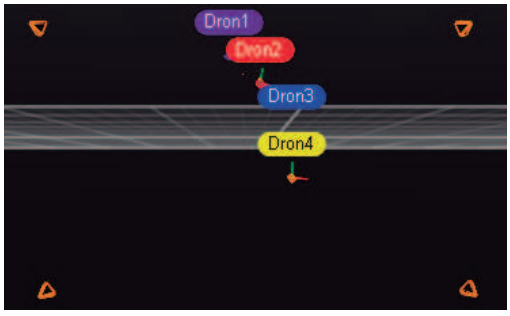
Dron1		Dron2		Dron3		Dron4	
x1	y1	x2	y2	x3	y3	x4	y4
-0.3	0.78	0.7	0.78	-0.3	-0.22	0.7	-0.22
Nodo2	1	Nodo1	1	Nodo1	1	Nodo1	1.41421356
Nodo3	1	Nodo3	1.41421356	Nodo2	1.41421356	Nodo2	1
Nodo4	1.414213562	Nodo4	1	Nodo4	1	Nodo3	1

(a) Posición inicial.



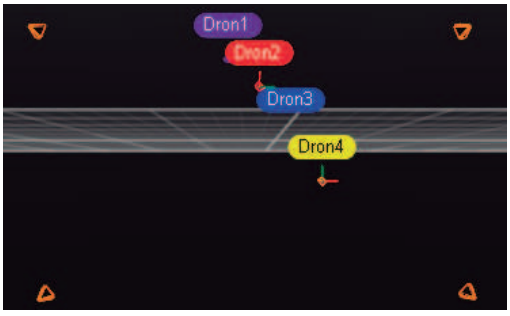
Dron1		Dron2		Dron3		Dron4	
x1	y1	x2	y2	x3	y3	x4	y4
-0.6	1.28	-0.3	0.78	0.2	0.28	-0.8	0.08
Nodo2	0.583095189	Nodo1	0.58309519	Nodo1	1.28062485	Nodo1	1.21655251
Nodo3	1.280624847	Nodo3	0.70710678	Nodo2	0.70710678	Nodo2	0.86023253
Nodo4	1.216552506	Nodo4	0.86023253	Nodo4	1.0198039	Nodo3	1.0198039

(b) Paso intermedio 1.



Dron1		Dron2		Dron3		Dron4	
x1	y1	x2	y2	x3	y3	x4	y4
-0.6	1.28	-0.2	0.68	0.2	0.08	0.2	-0.52
Nodo2	0.721110255	Nodo1	0.72111026	Nodo1	1.44222051	Nodo1	1.96977156
Nodo3	1.44222051	Nodo3	0.72111026	Nodo2	0.72111026	Nodo2	1.26491106
Nodo4	1.96977156	Nodo4	1.26491106	Nodo4	0.6	Nodo3	0.6

(c) Paso intermedio 2.

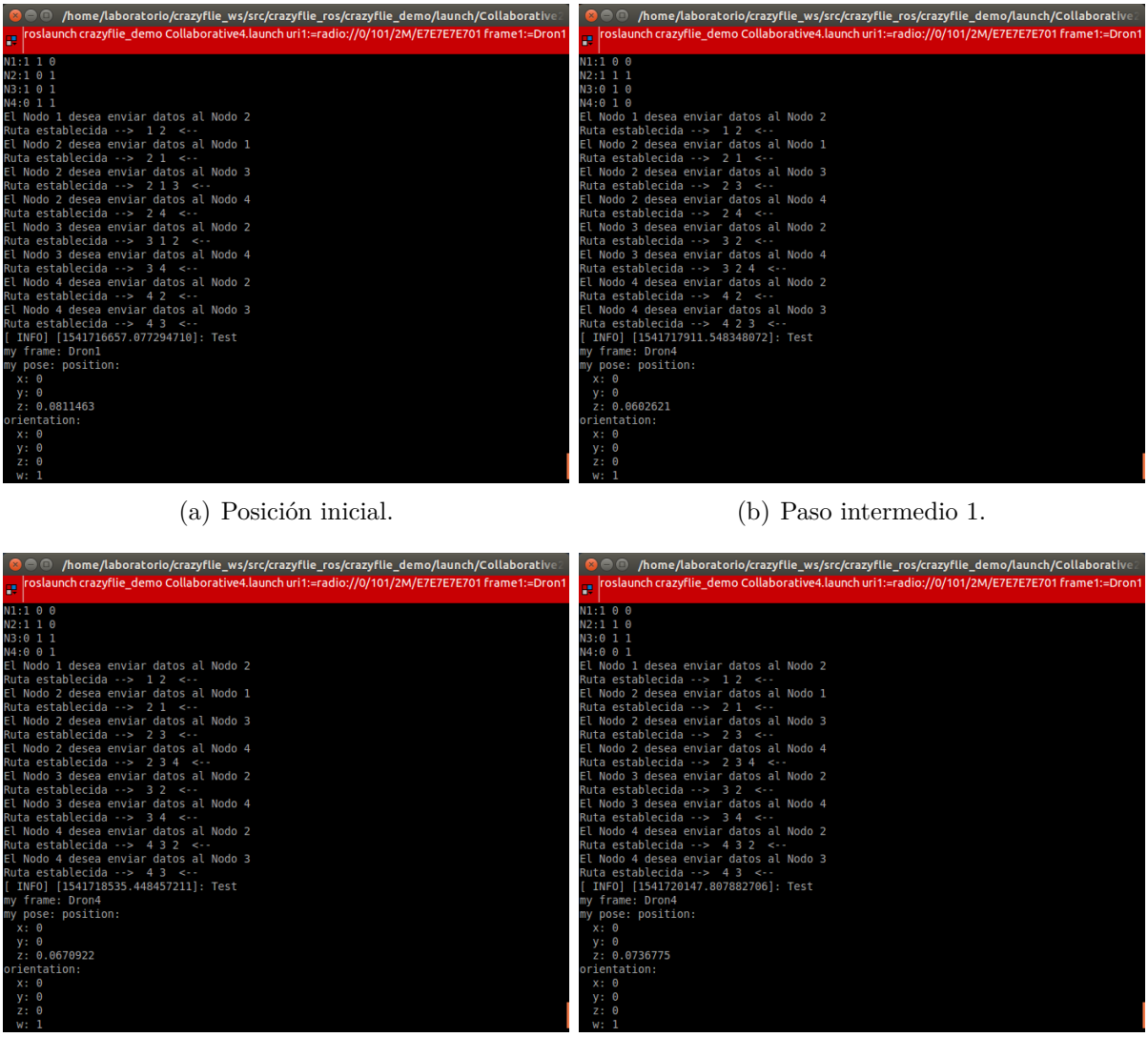


Dron1		Dron2		Dron3		Dron4	
x1	y1	x2	y2	x3	y3	x4	y4
-0.6	1.28	-0.2	0.68	0.2	0.08	0.6	-0.52
Nodo2	0.721110255	Nodo1	0.72111026	Nodo1	1.44222051	Nodo1	2.16333077
Nodo3	1.44222051	Nodo3	0.72111026	Nodo2	0.72111026	Nodo2	1.44222051
Nodo4	2.163330765	Nodo4	1.44222051	Nodo4	0.72111026	Nodo3	0.72111026

(d) Posición final.

Figura 6.18. Formación en diagonal con sus paso intermedios y tablas de distancia entre nodos.

La fig. 6.18 se muestra una formación en diagonal realizada con cuatro agentes, con la finalidad de mostrar el sistema de enrutamiento de la LabCA-FCE, misma que se encuentra relacionada con la fig. 6.19, la cual ilustra los mensajes desplegados por el control programado en ROS en cada paso de la formación.



(a) Posición inicial.

(b) Paso intermedio 1.

(c) Paso intermedio 2.

(d) Posición final.

Figura 6.19. Ruteos realizados por la plataforma.

Las impresiones que realiza el control son: tablas de enrutamiento (Nx: 1 1 0) y una posible ruta de envío. De acuerdo con la topología mostrada en la fig. 5.20; en cada uno de los VANTs se cumplen las siguientes conexiones.

Dron 1, mantiene una conexión con el dron 2.

Dron 2, mantiene una conexión con los drones 1, 3 y 4.

Dron 3, mantiene una conexión con los drones 2 y 4.

Dron 4, mantiene una conexión con los drones 2 y 3.

De la capturas mostradas en la fig. 6.19 es posible observar una variable llamada Nx ($x = 1, 2, 3, 4$), esta es la tabla de enrutamiento generada para cada nodo durante la formación en curso, donde 0 significa que no existe comunicación con el nodo en esa casilla y 1 que si. El formato de esta variable es el siguiente: $Nx = |nodoa|nodob|nodoc|$, donde $a, b\ y\ c$ se encuentran ordenados de menor a mayor de izquierda a derecha. Por ejemplo, $N1 = |nodo2|nodo3|nodo4|$.
El algoritmo de enrutamiento se ejecuta en el nodo de ROS *Crazyfliex/controller*. Para esta prueba $x = 1, 2, 3$ y 4 (4 Crazyflie2.0) y la distribución computacional en ROS se mantiene como el de la fig. 6.16, solo que se deben considerar otros dos bloques (Crazyflie 3 y 4).

6.3.6. Resultados experimentales

Debido a factores como: un espacio de trabajo reducido para OptiTrack, fallas mecánicas en dos Crazyflie que no permitieron una adecuada caracterización del sistema, una considerable cantidad de perturbaciones en el laboratorio para el PA del Crazyflie (varias redes wifi, celulares, repetidoras, entre otras.) y la falta de baterías hicieron que solo se pudieran volar dos Crazyflie. Además, de tener un sistema de control colaborativo diseñado para dos agentes. A continuación se muestran los resultado experimentales obtenidos del Crazyflie 2.0 utilizando la arena descrita anteriormente con el control colaborativo discutido en el Capítulo 4. Cabe mencionar que debido al extenso trabajo de tesis se utiliza el control de orientación que se encuentra en el Crazyflie 2.0, mientras que el control de posición y colaborativo son programados en ROS (estos dos últimos son explicados de manera detallada en la referencia [38]).
La fig. 6.20 muestra el diagrama a bloques del control colaborativo implementado en la plataforma experimental, donde P_i y V_i representa la posición y velocidad lineal del agente 1, respectivamente. P_j y V_j representa la posición y velocidad lineal del agente 2. De esta forma, el agente 1 conoce su propia posición y velocidad, la posición y velocidad del agente vecino (agente 2) y la posición deseada impuesta por el agente virtual. Por lo tanto, el agente 2 conoce su propia posición y la posición de su vecino (agente 1) sin saber la posición deseada del agente virtual, el agente 2 utilizando esta información el agente debe de respetar dicha formación.

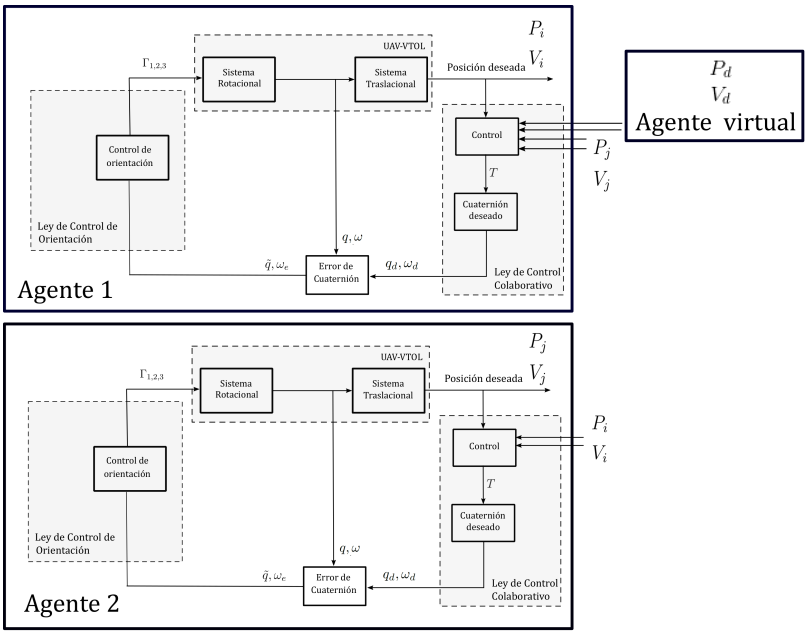


Figura 6.20. Diagrama de bloques del control colaborativo.

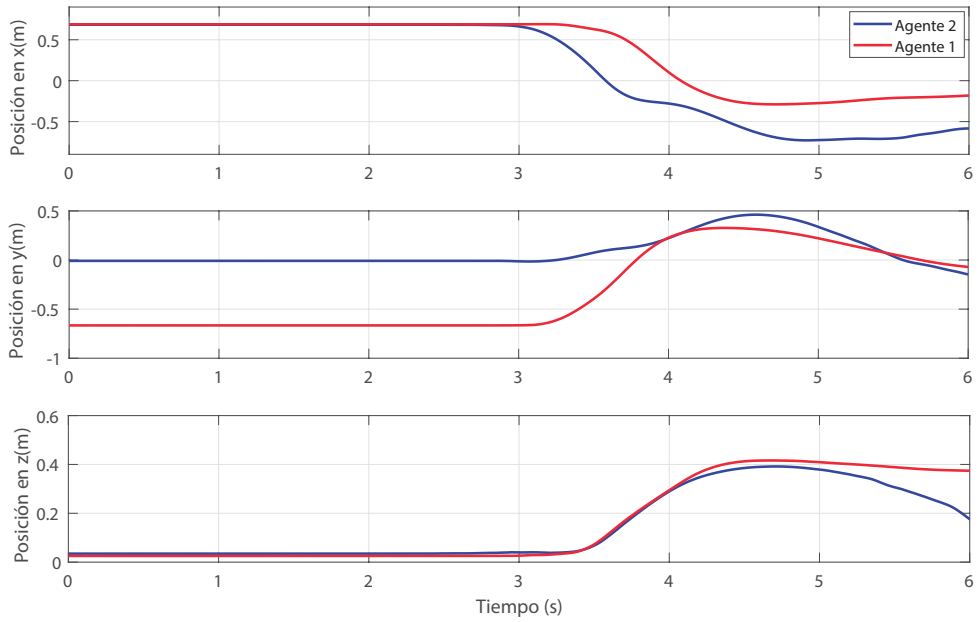


Figura 6.21. Evolución de la posición de los VANTs.

En la figura 6.21 se presentan tres gráficas que muestran la evolución de la posición de los agentes 1 y 2. Se puede observar que los agente comienzan en una posición inicial $P1 = (0.66, 0.68, 0)$ m y $P2 = (0, 0.68, 0)$ m, respectivamente. La posición deseada $Pd = (0, 0, 0.4)$ del agente 1 está impuesta por un agente virtual que le indica cual debe ser su posición, y al mismo tiempo el agente 2 debe respetar dicha formación a través de los datos que son enviados del agente 1. De esta manera, el consenso entre ambos agentes es verificada en la implementación experimental del control colaborativo en 4.12 y 4.13 donde $\Delta_{xij}, \Delta_{xij} = 0$ y $\Delta_{yij}, \Delta_{yij} = 0$ representan la distancia de cada agente en el eje x

y y. La fig. 6.22 muestra una vista en el plano $x - y$ la posición inicial y final de cada agente.

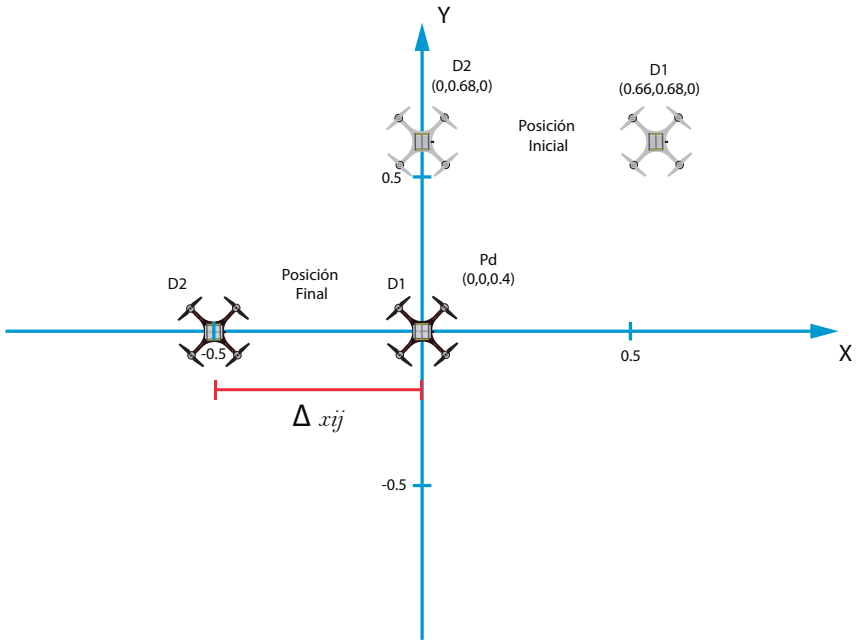


Figura 6.22. Vista en el plano $x - y$ del control colaborativo.

En base a los resultados experimentales obtenidos, el promedio del error entre la posición deseada y la que se obtuvo es de $\pm 0.09\ m$ (Véase fig. 6.23).

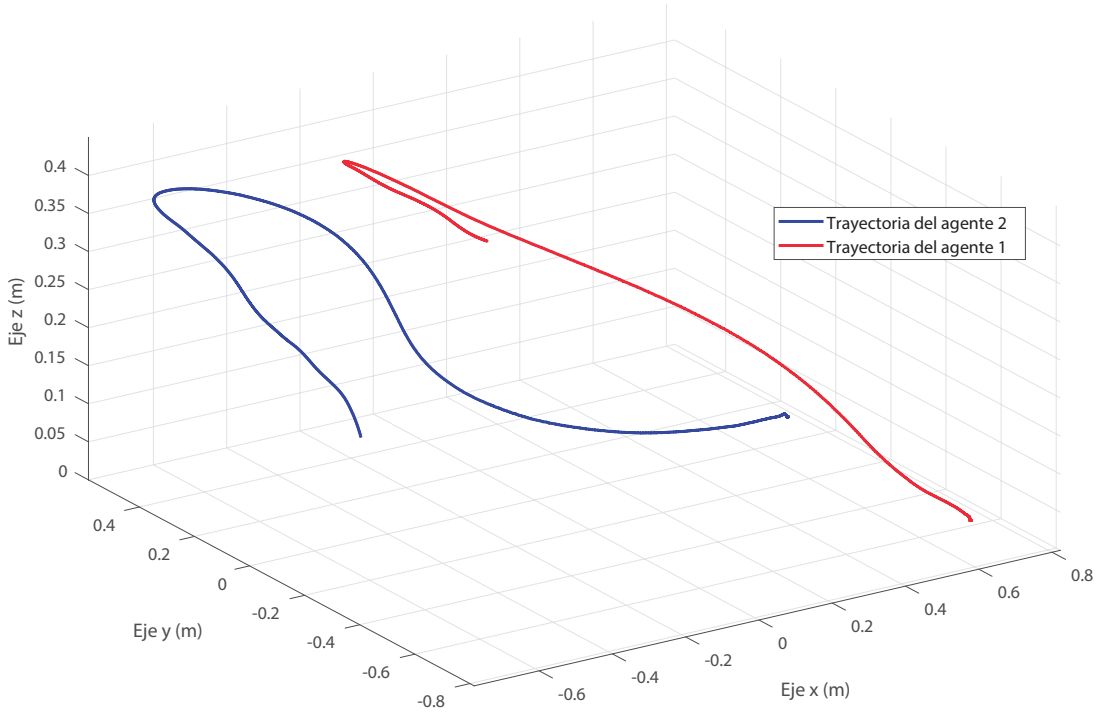
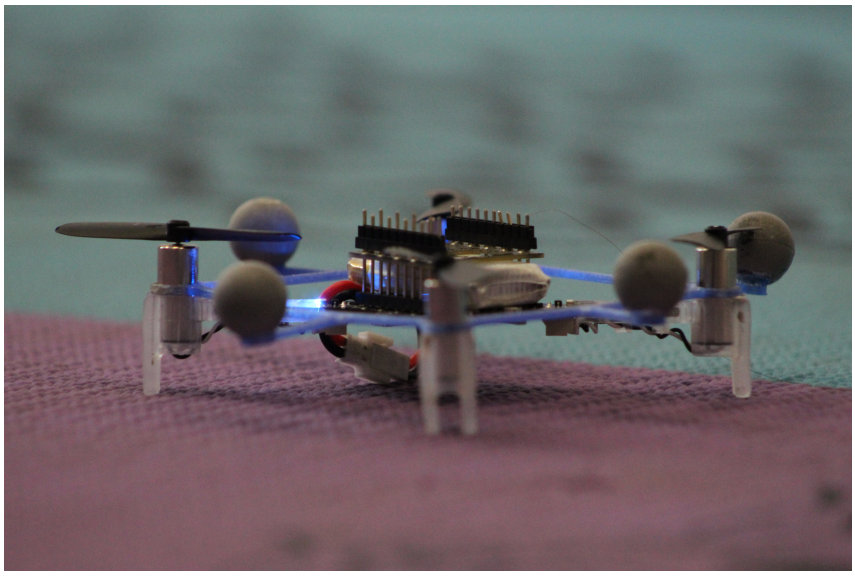


Figura 6.23. Trayectoria de los agentes realizando control colaborativo de forma experimental.

Finalmente, la fig. 6.24 muestra la arena de vuelo montada y el dron utilizado para este trabajo.

(a) Área de trabajo de la arena de vuelo.



(b) Crazyflie 2.0 con guarda.

Figura 6.24. Ruteos realizados por la plataforma.

Conclusiones

A lo largo de este trabajo se diseñó e implementó una red de comunicaciones para el control de formación de vehículos aéreos no tripulados. Como resultado final, se obtuvieron dos plataformas: una de simulación y una experimental.
La plataforma de simulación está basada en Simulink del software MATLAB, en donde es posible algoritmos de control (especialmente colaborativos) haciendo uso de una red de comunicaciones con un protocolo de enrutamiento AODV. El protocolo de enrutamiento es uno de los puntos claves en esta red de comunicaciones. El análisis comparativo realizado en el Capítulo 5, permitió seleccionar el AODV, debido a que su estrategia de operación bajo demanda lo hace adecuado para el sistema de control (disparado por eventos), además de que el sistema de comunicaciones solo trabaja cuando una bandera de envío es activada. De esta manera, el canal de comunicación mantiene un canal libre de tráfico, mismo que se encuentra siempre presente en un protocolo proactivo (DSDV). Por otra parte, el acoplamiento de los tiempos y los métodos de evaluación se podrían enunciar como los principales retos de este diseño, en el que ambos sistemas funcionan perfectamente por separados, y se tuvieron problemas al momento de la unión de todo el sistema.
En base a los resultados de las simulaciones realizadas en la Sección 5.2.2 se puede observar que la red presenta un PDR del 99 % lo que indica que se están recibiendo casi en su totalidad los datos que se envían. Igualmente, el E2ED general de la red es de 6.3 ms y ninguno de los nodos excede los 20 ms. Con esta métrica se comprueba que la red cumple con el requerimiento de 20 ms por parte del control. Así mismo, el thp de la red es de $30.498Kbs$. Si se compara este último con los obtenidos en el NS2 resulta ser muy pequeño. Esto se debe a que este toolbox cuenta con una trama muy corta (Heder+Datos+CRC <100bytes) en comparación con la del otro simulador.
Se puede afirmar que la plataforma de simulación diseñada es la adecuada para la evaluación, analisis y puesta en marcha de un sistema de control colaborativo, donde el sistema de comunicaciones propuesto en la sección 5.1 fue evaluado con resultados satisfactorios para este trabajo.
Por otro lado, la plataforma experimental *LabCA-FCE:ROS+Crazyflie2.0* permite el análisis de algoritmos de control colaborativo de manera experimental. En base a la prueba realizada en la Sección 6.3.6 es posible observar que los agentes logran el consenso. Sin embargo; este presenta un error de ± 0.09 m, el cual comparado con los 80 ms de latencia

y un error milimétrico de OptiTrack se considera un buen resultado. La principal ventaja de esta plataforma es que cuenta con un paquete dedicado al control, por esta razón, si se desea evaluar otro algoritmo de control solo basta sustituir el código por default del sistema por el nuevo. Además, el código del control se encuentra escrito en C++, esto reduce el riesgo de que el usuario cometa un error en la estructura del programa debido a un mala declaración en algún archivo principal de conexión.

Al probarse la funcionalidad de ambas plataformas, esto da pie a desarrollar nuevas estrategias de control que permitan optimizar tiempos, rendimientos, etc. Por ejemplo, un control colaborativo que cambie su topología de red en base a su tabla de enrutamiento. Por la parte de comunicaciones, se pueden agregar nuevos parámetros de evaluación de redes, evaluar nuevas topologías y nuevos protocolos de enrutamiento.

Finalmente, la plataforma experimental soporta hasta 6 agentes, sin embargo, esta pudiera incrementarse con la adquisición de mas cámaras y un espacio de trabajo más amplio.

Bibliografía

[1] Forbes México. Los drones invaden méxico, November 2015. www.forbes.com.mx.

[2] Grand View Research. Commercial uav industry outlook, January 2016. http://www.grandviewresearch.com.

[3] M. S. H. a. M. A. Md. Hasan Tareque. On the routing in flying ad hoc networks. *Computer Science and Information Systems (FedCSIS), 2015 Federated Conference on*, pages 1 – 9, September 2015. DOI: 10.15439/2015F002.

[4] Vega Alonzo Argel. Desarrollar una estrategia de control colaborativo disparado por eventos aplicado al problema del formación de la orientación de un grupo de vehículos aéreos no tripulados. Master's thesis, BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA, Puebla,México, dec 2016.

[5] R. J. a. G. V. Lav Gupta. Survey of important issues in uav communication networks. *IEEE Communications Surveys and Tutorials*, 18:1123 – 1152, November 2015. DOI: 10.1109/COMST.2015.2495297.

[6] Carlos Ernesto Carrillo Arellano. Análisis de protocolos de encaminamiento para redes inalámbricas tipo malla en modo infraestructura. Master's thesis, UNIVERSIDAD AUTÓNOMA METROPOLITANA, México,México, sep 2011.

[7] P. M. a. S. V. Krishnamurthy. *AD HOC NETWORKS Technologies and Protocols*. Springer, United States of America, 2005.

[8] Ramin Hekmat. *Ad-hoc Networks: Fundamental Properties and Network Topologies*. Springer, Dordrecht, The Netherlands, 2006.

[9] Unai Aguilera Irazabal. *NUEVOS PROTOCOLOS PARA EL DESCUBRIMIENTO Y LA COMPOSICIÓN AUTOMÁTICA DE SERVICIOS EN REDES MÓVILES AD HOC*. PhD thesis, UNIVERSIDAD DE DEUSTO, Bilbao, España, jan 2013.

[10] Bekmezci A. Lker Sahingoz. Ozgur KorayTemel. Samil. Flying ad-hoc networks (fanets): A survey. *Ad-Hoc Networks*, 11:1254–1270, 2013. DOI: https://doi.org/10.1016/j.adhoc.2012.12.004.

[11] Stallings William. *Comunicaciones y Redes de Computadores*. Prentice Hall, Madrid, 6 edition, 2000.

[12] Feng Zhao. David Tipper. Jinmei Tatuya. Pei Zheng et al. *Wireless Networking Complete*. Morgan Kaufmann, United States of America, 2009.

[13] Erdal Cayirci and Chunming Rong. *Security in Wireless Ad Hoc and Sensor Networks*. Wiley, United Kingdom, 2009.

[14] IEEE. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE Computer Society, New York, NY, 2012.

[15] University of Southern California. The network simulator ns-2. http://www.isi.edu/nsnam/ns/.

[16] RIVERBED. Riverbed modeler. https://www.riverbed.com.

[17] Amandeep Kaur Hardeep Singh2. A study of secure routing protocols. *International Journal of Application or Innovation in Engineering and Management (IJAIEM)*, 2:176–179, 2013. ISSN:2319-4847.

[18] Sergio Hernán Rocabado Moreno. Caso de estudio de comunicaciones seguras sobre redes móviles ad hoc. Master's thesis, Universidad Nacional de La Plata, Buenos Aires,Argentina, dec 2013.

[19] Yan Zhang Jijun Luo Honglin Hu. *WIRELESS MESH NETWORKING Architectures, Protocols and Standards*. Auerbach Publications, USA, 2007.

[20] L. Koszalka A. Zakrzewska and I. Pozniak-Koszalka. Performance study of routing protocols for wireless mesh networks. *Systems Engineering, 2008. ICSENG '08. 19th International Conference on*, pages 331–336, 2008. DOI: 10.1109/ICSEng.2008.49.

[21] I. Pozniak-Koszalka A. Zakrzewska, L. Koszalka and A. Kasprzak. Analysis of routing protocol performance in wireless mesh networks. *2010 International Conference of Computational Science and Its Applications*, pages 307–310, 2010. DOI: 10.1109/ICCSA.2010.72.

[22] PARUL SHARMA ARVIND KALIA and JAWAHAR THAKUR. Performance analysis of aodv, dsr and dsdv routing protocols in mobile ad-hoc network (manet). *Journal of Information Systems and Communication*, 3:322–326, 2012. ISSN:0976-8742.

[23] S. Taneja and A. Kush. A survey of routing protocols in mobile ad hoc networks. *International Journal of Innovation, Management and Technology*, 1:279–285, 2010.

[24] G. Jayakumar and G. Gopinath. Ad hoc mobile wireless networks routing protocols, a review. *Journal of Computer Science ,Science Publications*, 3:574–582, 2007.

[25] M. Teshnehlab S. Tabatabaei and S. J. Mirabedin. Ad hoc mobile wireless networks routing protocols, a review. *Wireless Pers Commun*, pages 1766–1778, 2015. DOI: 10.1007/s11277-015-2475-2.

[26] A. Patel S. Mohseni, R. Hassan and R. Razali. Comparative review study of reactive and proactive routing protocols in manets. *4th IEEE International Conference on Digital Ecosystems and Technologies*, pages 304–309, 2010.

[27] A. Kalia P. Sharma and J. Thakur. Performance analysis of aodv, dsr and dsdv routing protocols in mobile ad-hoc network (manet). *Journal of Information Systems and Communication*, 3:322–326, 2012.

[28] M. Gaballah Q. Razouqi A. Boushehri and L. Alsaleh. Extensive simulation performance analysis for dsdv, dsr and aodv manet routing protocols. *27th International Conference on Advanced Information Networking and Applications Workshop*, pages 335–342, 2013. DOI: 10.1109/WAINA.2013.239.

[29] Bekmezci A. Lker Sahingoz. Ozgur KorayTemel. Samil. Flying ad-hoc networks (fanets): A survey. *Ad-Hoc Networks*, 11:1254–1270, 2013.

[30] Alamsyah and Purnomo H. Performance of the routing protocols aodv, dsdv and olsr in health monitoring using ns3. *International Seminar on Intelligent Technology and Its Application*, pages 323–328, 2016.

[31] Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. *SIGCOMM '94 Proceedings of the conference on Communications architectures, protocols and applications*, pages 234–244, 1994.

[32] V.Ellappan A.Sureshkumar and K. Manivel. A comparison analysis of dsdv and aodv routing protocols in mobile ad hoc networks. *IEEE Conference on Emerging Devices and Smart Systems (ICEDSS 2017)*, pages 234–237, 2017.

[33] D. Shanmugasundaram and Dr. A. R. Md. Shanavas. Analysis of cosmic dust attack in ad-hoc on demand distance vector routing (aodv) protocol. *World Congress on Computing and Communication Technologies (WCCCT)*, pages 11–15, 2017.

[34] E. Altman and T. Jiménez. *NS Simulator for Beginners*. Morgan and Claypool, Berkeley, USA, 2012. DOI: 10.2200/S00397ED1V01Y201112CNT010.

[35] GNU. The gnu awk userś guide. https://www.gnu.org/software/gawk/manual/gawk.html#Invoking-Gawk.

[36] Larry Wall. Perl: Practical extraction and report language. http://www-cgi.cs.cmu.edu/cgi-bin/perl-man.

[37] Lund University. Truetime: Simulation of networked and embedded control systems. http://www.control.lth.se/truetime/.

[38] Pulido Flores Araceli. Desarrollo de una estrategia de control colaborativo asíncrono para el transporte de cargas mediante un sistema de multi vants. Master's thesis, BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA, Puebla,México, dec 2018.

[39] A.N.A.M.V. RODRÍGUEZ, F.A. MARTÍN, F.G. COUSO, M.L. GONZÁLEZ, G.P. VERA, and C.V. MARTÍN. *Teoría de grafos. Ejercicios y problemas resueltos.* Paraninfo, 2014.

[40] M. G. Villarreal-Cervantes J. P. Sánchez-Santana, J. F. Guerrero-Castellanos and S. Ramírez-Martínez. Control distribuido y disparado por eventos para la formación de robots móviles tipo (3,0). *Congreso Nacional de Control Automático 2017*, pages 493–498, 2017.

[41] ROS. Ros. http://www.ros.org/.

[42] intorobotics. 15+ reasons to use the robot operating system (ros). https://www.intorobotics.com/15-reasons-to-use-the-robot-operating-system-ros/.

[43] Moodle UA 2015-16. Manual de ros. https://moodle2015-16.ua.es/moodle/mod/book/tool/print/index.php?id=82546.

[44] OptiTrack. Optitrack. https://www.optitrack.com/.

[45] NORDIC SEMICONDUCTOR. nrf24lu1 product specification. https://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24LU1.

[46] Honig W. and Ayanian N. *Flying Multiple UAVs Using ROS*. Springer, Cham, 2017. https://doi.org/10.1007/978-3-319-54927-9_3.

Apéndice A

Algoritmo de Dijkstra y Bellman-Ford

Los algoritmos y ejemplos de Dijkstra y Bellman-Ford mostrados en este apéndice, fueron tomados de [11].

Algoritmo de Dijkstra

El algoritmo de Dijkstra se puede enunciar como sigue: encontrar las rutas más cortas entre un nodo origen dado y todos los demás nodos, desarrollando los caminos en orden creciente de longitud. El algoritmo actúa en etapas. Tras el paso o etapa $k - ésima$ se han determinado los caminos más cortos a los k nodos más cercanos (de menor costo) al nodo origen especificado; estos nodos se almacenan en el conjunto T. En el paso $(k + 1)$ se añade a la lista T aquel nodo que presente el camino más corto desde el nodo origen y que no se encuentre ya incluido en dicha lista.
A medida que se incorporan nuevos nodos a T, se define su camino desde el origen. El algoritmo se puede describir formalmente como sigue. Definamos:

N = conjunto de nodos de la red.

s = nodo origen.

T = lista o conjunto de nodos añadidos o incorporados por el algoritmo.

$w(i, j)$ = costo del enlace desde el nodo i al nodo j; $w(i, j) = 0$; $w(i, j) = \infty$ si los dos nodos no se encuentran directamente conectados; $w(i, j) \geqslant 0$ si los dos nodos están directamente conectados.

$L(n)$ = costo en curso obtenido por el algoritmo para el camino de mínimo costo del nodo s al nodo n; al finalizar el algoritmo, este costo corresponde al del camino de mínimo costo de s a n en el grafo de la red.

El algoritmo consta de tres pasos, repitiéndose los pasos 2 y 3 hasta que $T = N$; es decir, hasta que las rutas finales han sido asignadas a todos los nodos en la red:

1. [**Inicialización**]
   $T = s$ El conjunto de nodos incorporados sólo consta del nodo origen s.
   $L(n) = (s, n), para n \neq s$ El costo inicial de las rutas a los nodos vecinos es el asociado a los enlaces.

2. [**Obtención del siguiente nodo**]
   Se busca el nodo vecino que no esté en T con el camino de menor costo desde s y se incorpora a T; también se incorporará el enlace desde ese nodo hasta un nodo de T que forma parte del camino. Esto se puede expresar como

   $$\text{Encontrar } x \text{ tal que } L(x) = \min_{j \notin T} L(j)$$

   Añadir x a T, incorporando también el enlace desde x que contribuye a $L(x)$ como la componente de menor costo (es decir, el último salto en la ruta).

3. [**Actualización de los caminos de mínimo costo**]

   $$L(n) = min[L(n), L(x) + w(x, n)] para todo n \notin T$$

Si el último término es el mínimo, el camino desde s hasta n es ahora el camino desde s hasta x concatenado con el enlace desde x hasta n.

El algoritmo concluye cuando todos los nodos han sido añadidos a T. Al final, el valor $L(x)$ asociado a cada nodo x es el costo (longitud) de la ruta de mínimo costo de s a x. Además, T define la ruta de mínimo costo desde s hasta cualquier otro nodo.
Cada iteración de los pasos 2 y 3 incorpora un nuevo nodo a T y define el camino de mínimo costo desde s hasta ese nodo, atravesando dicha ruta sólo nodos incluidos en T. Para comprender mejor esto considérese el siguiente razonamiento. Tras k iteraciones existen k nodos en T, habiéndose obtenido además el camino de mínimo costo desde s hasta cada uno de esos nodos.

Algoritmo de Bellman-Ford

El algoritmo de Bellman-Ford se puede enunciar así: encontrar los caminos más cortos desde un nodo origen dado con la condición de que éstos contengan a lo sumo un enlace; a continuación encontrar los caminos más cortos con la condición de que contengan dos enlaces como máximo, y así sucesivamente. Este algoritmo actúa también en pasos, pudiéndose describir formalmente como sigue. Definamos:

s = nodo origen.

h = número máximo de enlaces en un camino en el paso actual del algoritmo.

$w(i,j)$ = costo del enlace desde el nodo i al nodo j; $w(i,j) = 0$; $w(i,j) = \infty$ si los dos nodos no se encuentran directamente conectados; $w(i,j) \geqslant 0$ si los dos nodos están directamente conectados.

$L_h(n)$ = costo del camino de mínimo costo desde el nodo s hasta el nodo n con la condición de que no haya más de h enlaces.

1. [**Inicialización**]

$$L_o(n) = \infty, \forall n \neq s$$
$$L_h(s) = 0, \forall h$$

2. [**Actualización**]
   Para cada sucesivo $h \geqslant 0$:
   Para cada $n \neq s$, calcular

$$L_{h+1}(n) = \min_j = [L_h(j) + w(j,n)]$$

Conectar n con el nodo predecesor j de mínimo costo y eliminar todas las conexiones de n con un nodo predecesor diferente obtenido en una iteración anterior. El camino de s a n finaliza con el enlace de j a n. Para la iteración del paso 2 con $h = K$, y para cada nodo de destino n, el algoritmo compara las rutas potenciales de longitud $K+1$ desde s hasta n con el camino existente al final de la iteración anterior. Si el camino más corto previo tiene un costo inferior, se guarda; en caso contrario, se define un nuevo camino de longitud $K+1$ de s a n, el cual consiste en una ruta de longitud K de s a algún nodo j más un salto directo desde el nodo j hasta el nodo n.

Ejemplo del Algoritmo de Dijkstra y Bellman-Ford

Una vez conocido los algoritmos de Dijkstra y Bellman-Ford, se presenta un ejemplo de su uso, el cual, aplica ambos algoritmos al diagrama de una red mostrada en la figura A.1

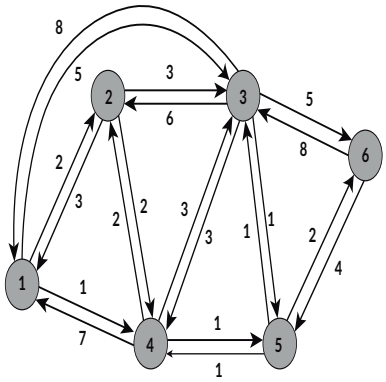


Figura A.1. Diagrama de una red de 6 nodos

Algoritmo de Dijkstra aplicado al diagrama

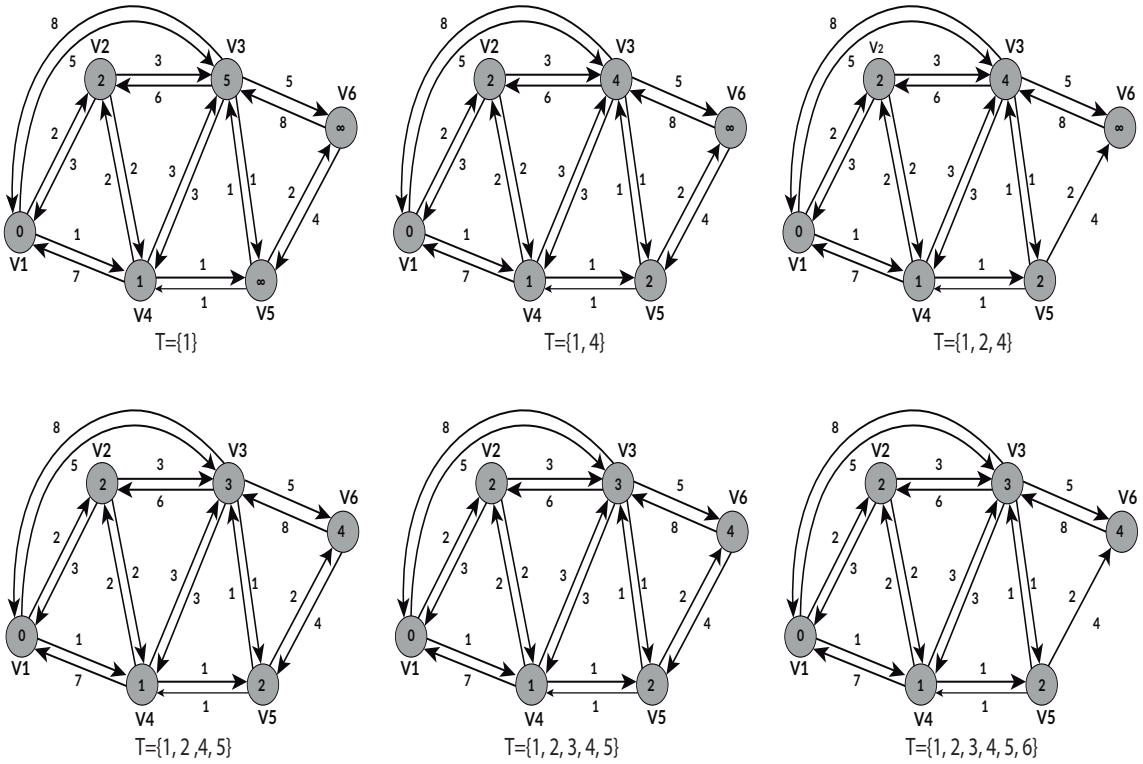


Figura A.2. Algoritmo de Dijkstra aplicado al diagrama de la figura A.1

Iteración	T	L(2)	Ruta	L(3)	Ruta	L(4)	Ruta	L(5)	Ruta	L(6)	Ruta
1	1	2	1-2	5	1-3	1	1-4	∞	–	∞	–
2	1, 4	2	1-2	4	1-4-3	1	1-4	2	1-4-5	∞	–
3	1, 2, 4	2	1-2	4	1-4-3	1	1-4	2	1-4-5	∞	–
4	1, 2, 4, 5	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6
5	1, 2, 3, 4, 5	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6
6	1, 2, 3, 4, 5, 6	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6

Tabla A.1. Tabla de enrutamiento del Algoritmo de Dijkstra para $s = 1$

Algoritmo de Bellman-Ford aplicado al diagrama

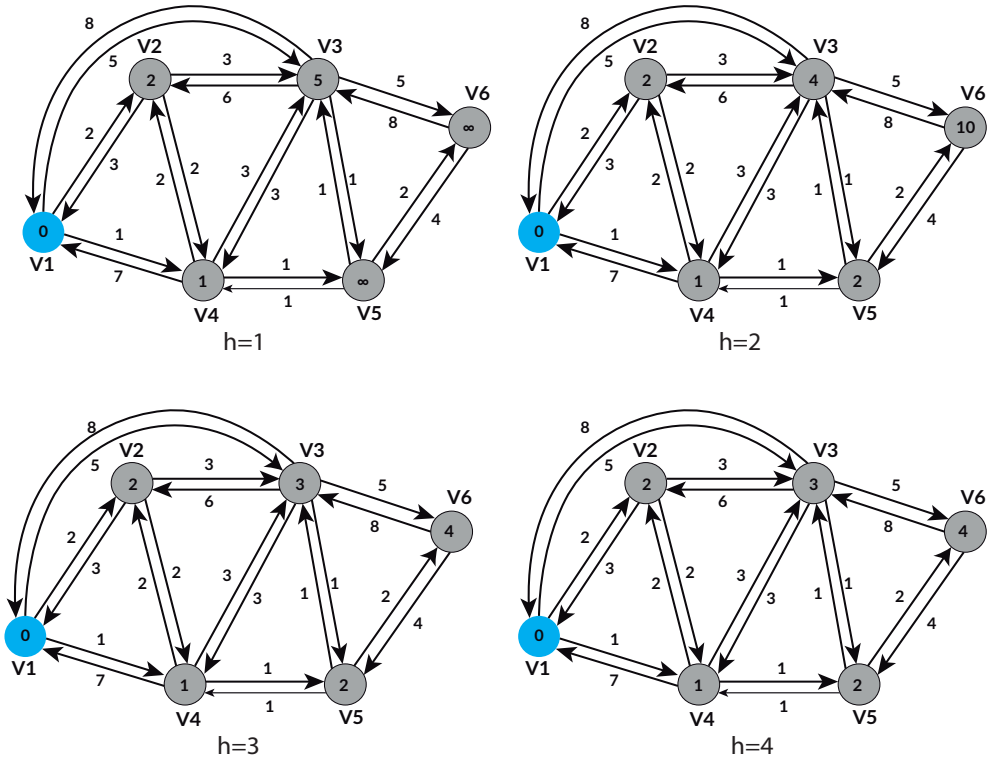


Figura A.3. Algoritmo de Bellman-Ford aplicado al diagrama de la figura A.1

h	$L_h(2)$	Ruta	$L_h(3)$	Ruta	$L_h(4)$	Ruta	$L_h(5)$	Ruta	$L_h(6)$	Ruta
0	∞	–	∞	–	∞	–	∞	–	∞	–
1	2	1-2	5	1-3	1	1-4	∞	–	∞	–
2	2	1-2	4	1-4-3	1	1-4	2	1-4-5	10	1-3-6
3	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6
4	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6

Tabla A.2. Tabla de enrutamiento del Algoritmo de Bellman-Ford $s = 1$

Apéndice B

MATLAB/SIMULINK

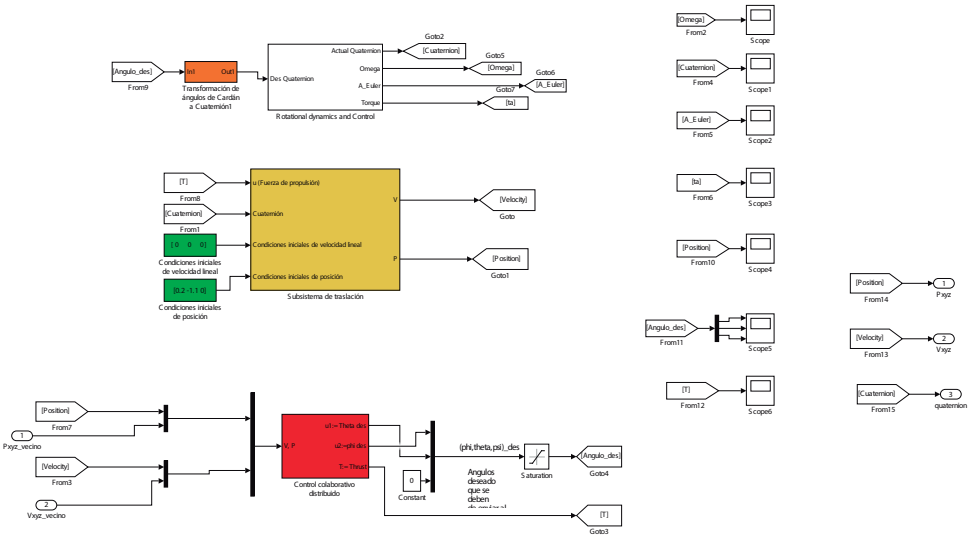


Figura B.1. Diagrama a bloques del control colaborativo para cuatro agentes en Simulink.

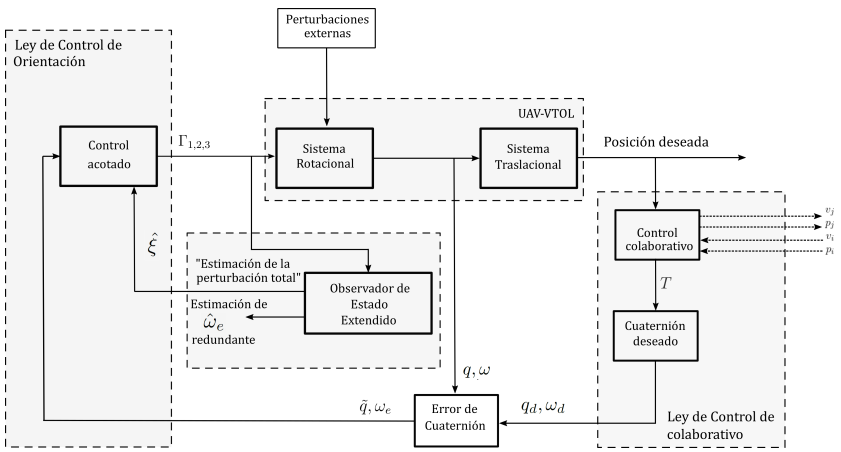


Figura B.2. Diagrama de bloques del control colaborativo para la formación de un conjunto de VANTs.

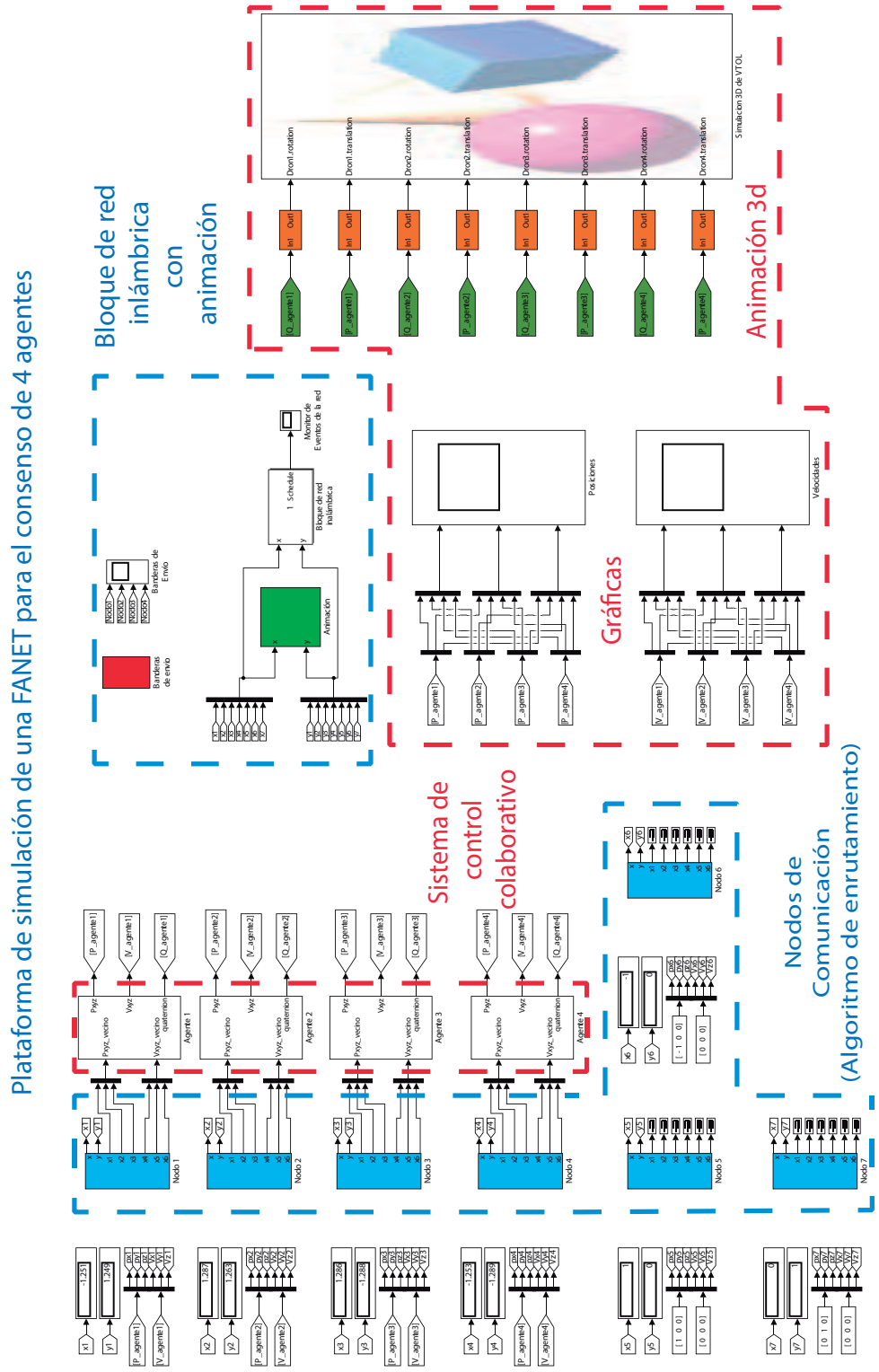


Figura B.3. Diagrama a bloques de la simulación para cuatro agentes en Simulink.

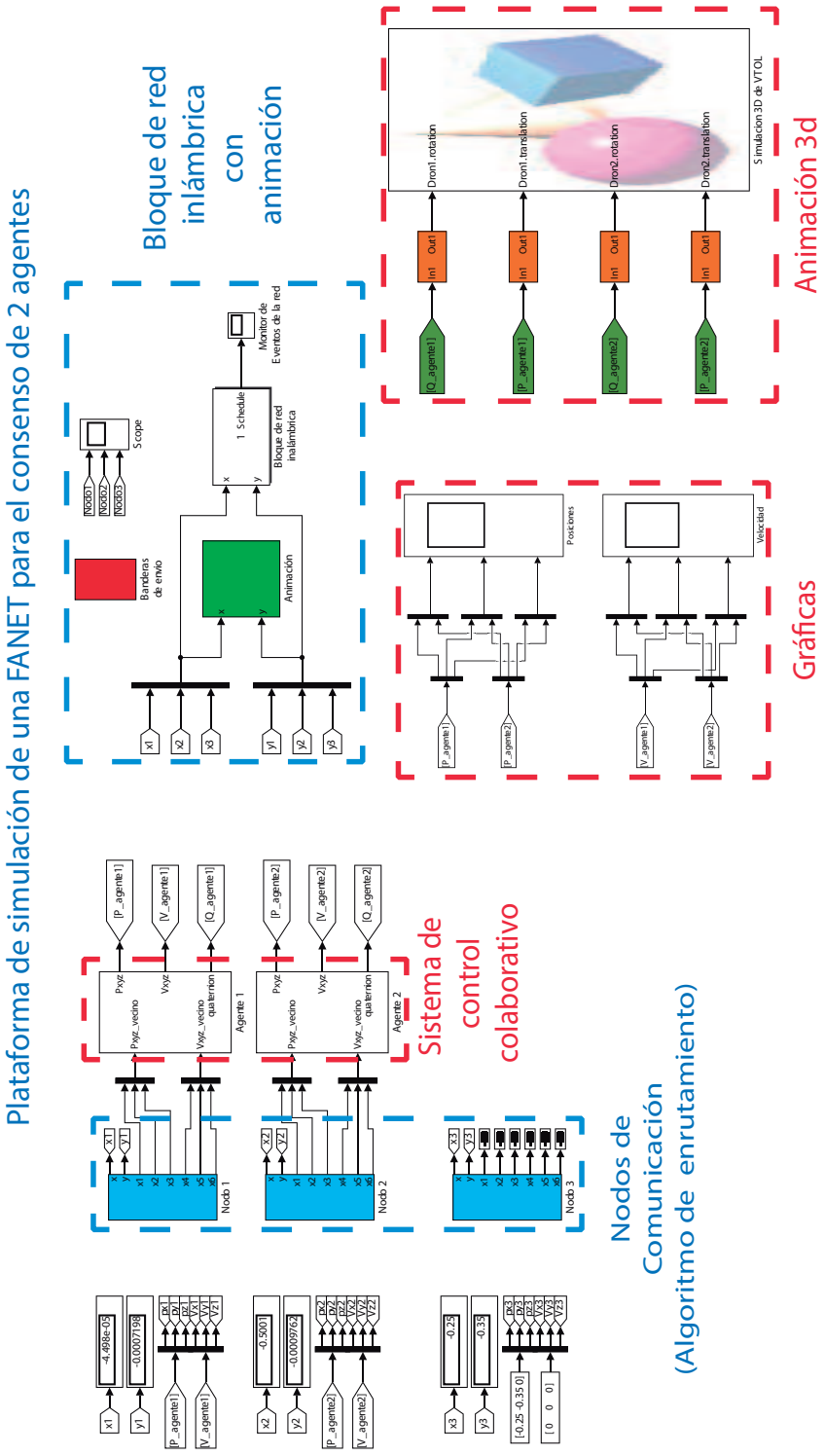


Figura B.4. Diagrama a bloques de la simulación para dos agentes en Simulink.

TrueTime2.0 pseudocódigo AODV

Algoritmo 1 Receptor

```
1: if Recibió datos then
2:     Actualiza el tiempo de expiración de la ruta de regreso hacia el nodo emisor
3:     if es el nodo destino then
4:         Pasa el mensaje a la aplicación
5:     else
6:         Envía los datos al siguiente salto establecido en la ruta
7:         Actualiza el tiempo de expiración de la ruta entrante
8:     end if
9: else
10:    switch Tipo de mensaje do
11:        case RREQ
12:            if Primer RREQ recibido then
13:                Almacena el RREQ en el cache
14:                Crea o actualiza la ruta de regreso al emisor
15:                Verifica una ruta hacia el nodo destino en la tabla de enrutamiento
16:                if Existe una ruta then
17:                    EnvÃa mensaje RREP al nodo emisor
18:                else
19:                    Actualiza y reenvía el mensaje RREQ
20:                end if
21:            end if
22:        case RREP
23:            Verifica una ruta hacia el nodo destino en la tabla de enrutamiento
24:            if No existe una ruta then
25:                Crea una ruta hacia el destino
26:            else if Existe la ruta pero necesita actualizarse then
27:                Actualiza la ruta hacia el destino
28:            end if
29:            if Es el nodo emisor then
30:                Notifica a la tarea de envío sobre la nueva ruta
31:            else if Es una nueva ruta o fue actualizada then
32:                Actualiza la ruta de regreso hacia el nodo emisor
33:                Envía el mensaje RREP al siguiente salto hacia el nodo emisor
34:            end if
35:        case RERR
36:            Encuentra e invalida todas las rutas afectadas
37:            Propaga el mensaje RERR por todos los saltos previos de la ruta
38: end if
```

Algoritmo 2 Emisor

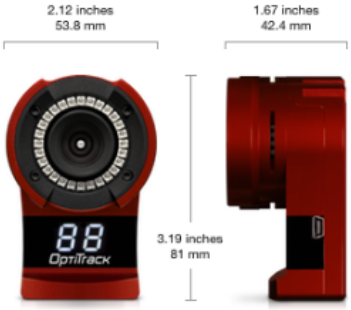
```
1: if Mensaje de solicitud de envío then
2:     Checa si existe una ruta al destino
3:     if Existe una ruta valida then
4:         Envía los datos al siguiente salto marcado en la ruta
5:         Actualiza el tiempo de expirado de la ruta
6:     else
7:         Inicializa el descubrimiento de nuevas rutas con el menaje RREQ
8:         Almacena datos en un buffer hasta encontrar una ruta
9:     end if
10: else if Notificación de ruta establecida then
11:     Envía los datos almacenados en el buffer
12: end if
```

Apéndice C

OptiTrack

Camera Body

- Width: 2.12 inches (53.8 mm)
- Height: 3.19 inches (81 mm)
- Depth: 1.67 inches (42.4 mm)
- Weight: 6.6 ounces (187 g)
- Mounting: 1/4"-20 tripod thread
- Status Indicators:
  - 2 digit numeric LEDs
  - 1 bicolor status LED



LED Ring

- 28 LEDs
- 850 nm IR
- Adjustable brightness
- Strobe or Continuous Illumination
- Removable

Lens & Filter

- Stock Lens: 5.5 mm F#1.8
  - Horizontal FOV: 56°
  - Vertical FOV: 46°
- Optional Lens: 8 mm F#1.8
  - Horizontal FOV: 42°
  - Vertical FOV: 34°
- M12 Lens Mount
- Adjustable focus w/ spring assist
- 800 nm IR long pass filter
- Optional: 800 nm IR long pass filter w/ Filter Switcher

Image Sensor

- Imager Size : 6.144 mm × 4.9152 mm
- Pixel Size : 4.8 µm × 4.8 µm
- Imager Resolution : 1280 × 1024 (1.3 Megapixels)
- Frame Rate: 30-120 FPS (adjustable)
- Latency: 8.3 ms
- Shutter Type: Global
- Shutter Speed:
  - Default: 500 µs
  - Minimum: 20 µs
  - Maximum: 7.5 ms (at 120 FPS)

Image Processing Types

- Object
- Segment
- Precision Grayscale
- MJPEG Grayscale
- Raw Grayscale

Input/Output & Power

- Data: USB 2.0
- Camera Sync: USB 2.0 (via OptiSync)
- Power: USB 2.0 @ 1 A

System Requirements

Recommended system requirements can vary significantly based on camera model and count. Please visit the **system builder tool** to find system recommendations tailored to your specific setup.

Note: Flex 13 cameras must be connect to the PC via an OptHub 2 in order for the LED Ring to be powered. Connecting the camera directly to the PC will result in the LEDs being disabled.

In the Box

- 1 Flex 13 camera Part number: FL13
- 1 Quick Start guide

Figura C.1. Características del OptiTrack.

Apéndice D

Publicaciones

D.0.1. Congresos

En base a los resultados obtenidos en este trabajo, se realizaron dos artículos para presentación en congresos internacionales. Ambos fueron aceptados y presentados.

Artículo 1. *Evaluation of AODV and DSDV Routing Protocols for a FANET: Further results towards robotic vehicle networks*, publicado en el 2018 IEEE 9th Latin American Symposium on Circuits & Systems (LASCAS), en Puerto Vallarta, México. del 25 al 28 de febrero de 2018. ISBN: 978-1-5386-2311-4.

Artículo 2. *Simulation platform for a VANET using the TrueTime toolbox: further result toward Cyber-Physical Vehicle Systems*, publicado en el 2018 IEEE 88th Vehicular Technology Conference: VTC2018Fall, en Chicago, USA, del 27 al 30 de agosto de 2018. ISBN: En espera.

D.0.2. Estancia

Se realizo una estancia de investigación en Institut Polytechnique des Sciences Avancées (IPSA) París, Francia. En un periodo con duración de tres meses (01 de mayo de 2018 al 27 de julio de 2018) con el proyecto *Implementation of robotics vehicle networks using ROS (Robot operating System)*, bajo la supervisión del Dr. Juan Antonio Escareno. Esta estancia me sirvió para aprender ROS y entender conceptos de control que posteriormente fueron sintetizados en código.

2018 IEEE 9th Latin American Symposium on Circuits & Systems (LASCAS)

Evaluation of AODV and DSDV Routing Protocols for a FANET: Further results towards robotic vehicle networks

Alejandro Garcia-Santiago, Josefina Castaneda-Camacho, Jose F. Guerrero-Castellanos and Gerardo Mino-Aguilar
Faculty of Electronics, Benemérita Universidad Autónoma de Puebla (BUAP)
Av. San Claudio y 18 sur Ciudad Universitaria Puebla Pue, Mexico. Email: see http://www.ece.buap.mx

***Abstract*—Flying Ad-Hoc Network (FANET) is basically a special form of Mobile Ad-Hoc Network (MANET) which is formed by a group of wireless nodes, which can dynamically form a network to exchange information without the necessity of using a fixed network infrastructure. Its good performance and its low cost, make it suitable for use in several applications such as environment sensors, vehicular communications, disaster rescue operations, air/land/navy defense and so on. In the present work the performance evaluation of Ad hoc on Demand Distance Vector (AODV) and Destination-Sequenced Distance-Vector (DSDV) routing protocols for a FANET using the software package NS2. The metrics of delivery rate, end-to-end delay and throughput are calculated for both protocols in order to evaluate its performance in the network. Besides, an energy analysis is shown for each protocol since energy is a key technical challenge for embedded and mobile applications. These results represent the base for future development of collaborative control protocols for robotic vehicle networks.**

***Index Terms*—AODV, DSDV, FANET, NS2 simulation, performance analysis, routing protocols, MANET.**

I. Introduction

Unmanned Aerial Vehicle (UAV) systems can fly independently or can be operated distantly. The use of single drone system is very common. However, simple functions of single drone system restrict its further applications. Currently, the need for building multi-drone system to improve the operational efficiency through the cooperation of multi-drones has become very important [1]. Nowadays, wireless mobile networks and devices are becoming increasingly popular due to they provide users access, communication anytime and anywhere. The communication among UAVs can be through a ground base station (GBS); others can connect to satellites, thereby realizing the UAV-to-UAV communication through the infrastructure [2]. However, they need to be connected to GBS or other infrastructure. This limits the operational range of the drones and requires a higher degree of coordination between them [1], [2].

The wireless network can be classified in two types: Infrastructured or Infrastructureless [3], [4].

Infrastructured wireless networks. The base stations are fixed and they have a specific range where the mobile nodes can move while communicating as long as they keep into the coverage area. However, when a node goes out of the range of a base station, it can get into the range of another base station whether there is one available, otherwise this node will be without connection. This kind of network can be seen as the conventional wireless mobile networks.

978-1-5386-2311-4/18/$31.00

Infrastructureless or MANETs. There are no fixed base stations and all the nodes in the network operates as routers. For this reason, the mobile nodes can move while communicating in anytime. They are changing from one to another position frequently and establishing a route among them to create their own network [4]. In MANETs, each node acts as a router and as a host due to the topology of the network changes suddenly. For this reason, ad hoc networks are useful in many application environments without infrastructure support such as sensors networks, vehicular communications, disaster rescue operations, air/land/navy defense, etc. [4], [5].

Currently, many studies are focused on MANETs due to their dynamic topology and no centralized administration. In this case, the communication problems faced by multi-drone systems can be solved with the application of ad-hoc networks among drones and GBS; this is called FANET which is basically ad hoc network between UAVs [1], [2]. This paper is focused on the routing protocols and mobility models that have been used in the FANET network to solve communication, cooperation and collaboration problem between UAVs. A multi-drone FANET is proposed and evaluated with two types of routing protocols in order to understand and to prove the importance of a good routing protocol for a good performance in the Network.

II. Routing protocols

A routing protocol is needed when a packet needs to be transmitted through different nodes in the network. These protocols find an adequate route for delivering the packets to the correct destination. The performance of FANETs is related to the efficiency of routing protocols. The efficiency depends on several factors such as convergence time after topology changes, bandwidth overhead to enable proper routing, and power consumption. The studies about routing protocols have been an active area of research for many years in fact; severals routing protocols have been proposed for FANETs [5], [6].

The Routing protocols can be classified into two types: Proactive and Reactive.

Proactive protocols. Every node maintains routing information of the other nodes in the network through the routing tables which are periodically updated as the network topology changes [7]. The different types of proactive protocols are: Destination sequenced Distance vector routing (DSDV), Wireless routing protocol (WRP), Fish eye State Routing protocol (FSR), Optimised Link State Routing protocol (OLSR), Cluster Gateway switch routing protocol (CGSR), Topology Dissemination Based on Reverse path forwarding (TBRPF), etc.

Reactive protocols. The routes are created when they are required. The route remains valid meanwhile the destination is reachable or until the route is no longer needed [7]. The different types of reactive protocols are: Ad hoc On Demand Distance Vector (AODV), Dynamic Source routing protocol (DSR), Temporally ordered routing algorithm (TORA), Associativity Based routing (ABR), etc.

This paper are specially focused on the evaluation of a FANET design (see the Fig. 1), using both types of protocols in order to find the one who offers the best characteristics and performance.

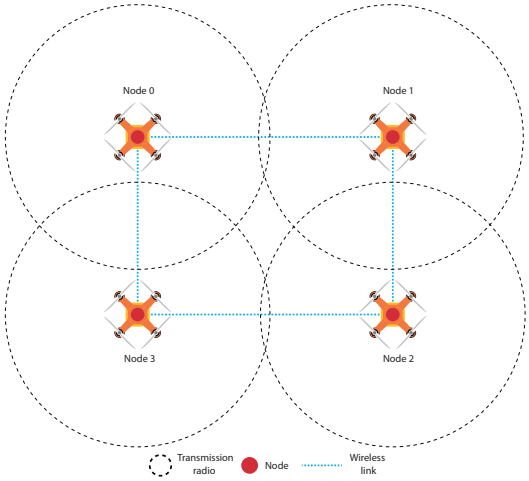


Fig. 1: FANET design for a drone communication network.

A. AODV protocol

Ad hoc on Demand Distance Vector (AODV) routing protocol was designed for FANET. It uses a reactive approach for finding routes and a proactive approach for identifying the path to their destination. The main advantages of this protocol is that it offers a high mobility rates, as well as a variety of data traffic levels, presents a low battery consumption, reduces the overhead on data traffic, improves scalability and performance [6]–[11].

B. DSDV protocol

Destination-Sequenced Distance-Vector (DSDV), is a Proactive routing protocol. Each node maintains a table consisting of the next-hop neighbor and the distance to the destination in terms of the number of hops. In this protocol, all the nodes periodically broadcast their tables to their neighboring nodes in order to maintain an updated view of the network. The tables can be updated in two ways, either incrementally or through a full dump [6]–[11]. An advantage of this protocol is that it guarantees loop free routes to each destination and find the optimal path based on an average settling delay which is a delay before advertising a route. A disadvantage is that the routing tables require regular updates battery power and a small amount of bandwidth even when the network is idle [6]–[11].

III. Evaluation metrics

According to [4]–[11], it is possible to get the basis metrics which are used for showing the performance of the routing protocols in a FANET.

- **Packet Delivery Rate (PDR)**
  It's shows how successful a protocol performs delivering packets from source to destination. It can be characterized as

$$PDR = \frac{R_p}{S_p} * 100 \tag{1}$$

  where R_p is the total packets received and S_p is the total packets sent.

- **End-to-End Delay (E2ED)** It is measured in seconds and calculates all possible delays occurred in the networks. The average E2ED can be defined as the time calculated for a data packet to be transmitted from source to destination across a FANET.

$$E2ED = T_R - T_S \tag{2}$$

  where T_R is receive time and T_S is sent time.

- **Throughput (Thp)** The average throughput is the average number of bits arrived per second at destination node. This is used as a measure of the reliability of the protocol under different conditions, hence the average throughput in the network needs to be as higher as possible [9]. Mathematically it can be defined as,

$$Average\ thp = \frac{N_p * packetsize * 8}{Seconds} \tag{3}$$

  where N_p is the number of bits received successfully by all destinations.

IV. Simulation results

The FANET analyzed in this paper is an Unmanned aerial vehicles (UAVs) network which consists of 4 drones (wireless mobile nodes) as shown in the Fig, 1. At the beginning, all of them are placed uniformly and forming a Mobile Ad-hoc Network in a 100m x 100m area. After 5 sec they start to moving and execute 4 mobility patterns with a velocity of 5m/s during 110 seconds of simulated time. Besides, the height of the antenna in every node is moved in order to show how much it affects the performance of the network.

In this part, the AODV and DSDV routing protocol simulations in a FANET are presented under various mobility scenarios and traffic characteristics using the NS2 and Ubuntu 14.04. Whereas, the performance evaluation metrics were programed in AWK and PERL and the results are plotted using Matlab [12] .

A. Simulation scenarios

The simulations are performed with a basic set of parameters which are shown in Table I.

TABLE I: Simulation Parameters

Parameters	Value
Channel type	Channel/Wireless channel
Radio-propagation model	Propagation/Two ray ground
Network interface type	Phy /wirelessphy
Highest antenna	0.5m - 3.5m
MAC type	Mac/802.11
Interface queue type	Queue/Drop Tail
Link Layer Type	LL
Antenna	Antenna/omni antenna
Area (mxm)	100x100
Number of mobile nodes	4
Node speed (m/sec)	5
Source type	TCP/NewReno
Simulation Time (sec)	110
Routing protocol	AODV, DSDV
Operating system	Ubuntu 14.04
NS2-2 Version	2.35

B. Packet delivery Rate.

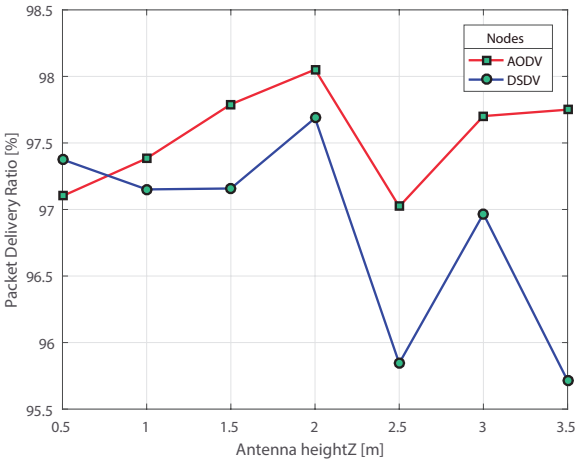
The Fig. 2 shows the packet delivery rate under TCP traffic for the AODV and DSDV routing protocols. Notice that both protocols present a good PDR. The AODV has a PDR=98.0514% and a PDR=97.0222% for the best and worst cases respectively. The DSDV has a PDR of 97.6867% and 95.7143% for the best and worst cases respectively. In line with this simulation, the PDR of AODV protocol results better than DSDV protocol in this simulation.

C. Average End-to-End Delay.

The Fig. 3 shows the average end-to-end delay under TCP traffic for the AODV and DSDV routing protocols.The AODV presents a small delay in comparison with the DSDV. The AODV has a AE2ED=58.0377ms in the best case and a AE2ED=165.4887ms in the worst scenario. Moreover, the DSDV has a AE2ED=50.9746ms in the best and a AE2ED=163.318ms for the worst cases. This results are expected because the DSDV is a proactive protocol then, it tends to be faster than AODV specially in a small networks.

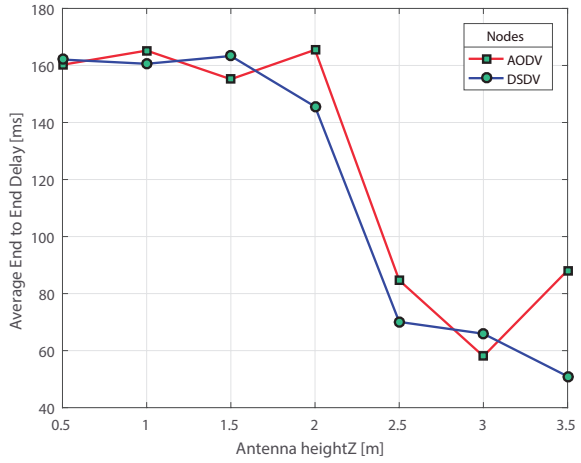
D. Average throughput.

The Figure 4 shows the average throughput under TCP traffic for the AODV and DSDV routing protocols. The AODV has a Thp=516.39Kbps in the best and a Thp=179.57Kbps in the worst cases. Otherwise, the DSDV has a Thp=524.74Kbps and Thp=109.95Kbps for the best and worst cases respectively.



	0.5	1	1.5	2	2.5	3	3.5
AODV	97.1008	97.384	97.7903	98.0514	97.0222	97.7019	97.7501
DSDV	97.3758	97.1509	97.1579	97.6867	95.8446	96.9669	95.7143

Fig. 2: Packet delivery Rate simulation.



	0.5	1	1.5	2	2.5	3	3.5
AODV	160.246	165.215	155.178	165.488	84.6672	58.0377	88.1378
DSDV	162.075	160.587	163.318	145.578	70.0709	65.9257	50.9746

Fig. 3: Average End-to-End Delay simulation.

According to this simulation, it can be appreciate clearly, how much affect the height of the antenna in the data transmission. In this case, as the antenna height is higher, the throughput in the routing protocols is affected.

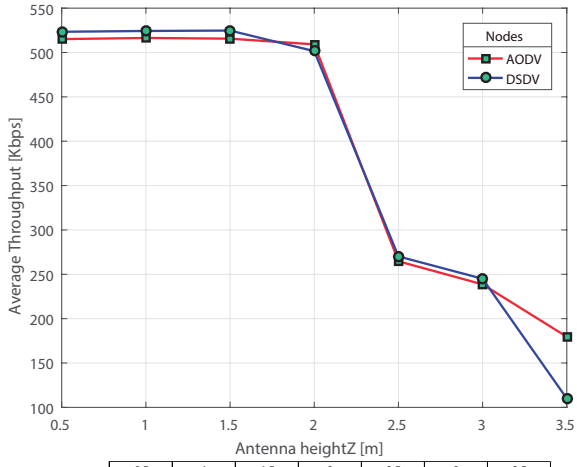
E. Energy consumption.

The Fig. 5 shows the Energy consumption under TCP traffic conditions for the AODV and DSDV routing protocols in every node of the network. This simulation adds the parameters considered in Table I.

According to the Fig. 5, the AODV gives an average energy consumption in their nodes of 13.7950915 Joules and the DSDV consumes 14.1981385 Joule.

V. Conclusions

In this paper was analyzed and compared the performance of two routing protocols: AODV and DSDV in a FANET . These



	0.5	1	1.5	2	2.5	3	3.5
AODV	515.11	516.39	515.59	509.16	264.64	238.62	179.57
DSDV	523.37	524.28	524.74	501.53	270.15	244.95	109.95

Fig. 4: Average throughput simulation.

TABLE II: Simulation Parameters

Parameters	Value
energyModel	EnergyModel
rxPower (W)	0.10
txPower (W)	0.25
idlePower (J)	0.05
initialEnergy (J)	1000

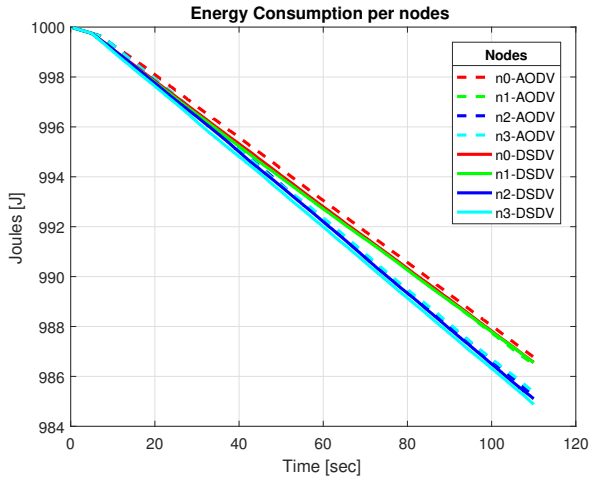


Fig. 5: Energy consumption simulation.

comparisons are based on different metrics such as packet delivery rate, average end-to-end delay, average throughput and energy consumption. Both protocols were selected according to their performance in a FANET [4]–[11].

The AODV presented an average PDR of 97.54%, which is 0.71% better than the DSDV. In the cases of the throughput simulation, the AODV shows an average of 391.29Kbps, being this 1.46% better than the other routing protocol. According to the energy consumption results, the AODV (13.79J) is lower than the DSDV (14.19J). However, the DSDV presents an average E2ED of 116.93ms, being this 6.66% faster than the AODV. In line with these results, it's recommended the use of the AODV routing protocol for the proposed FANET (UAV network of four nodes).

Finally, in accordance with the results obtained is possible to appreciate good performances and slight differences between the protocols chosen. This effect is observed because of the simulations were developed in a small network (4 nodes). In addition, the height of the antenna was moved inside typically ranges of operation of this UAV networks to illustrate the effects that it has into the FANET. A notable difference between them can be observed moving the speed of the nodes or adding more nodes. The two ray ground model (radio-propagation) was used however it has a drawback. The throughput proved to be the most sensitive metric to the parameters variations. It is worth mentioning that although the two ray ground model (radio-propagation) is used it has a drawback according the distance between the nodes reduces. Therefore, it is possible the use of another models to small distances.

ACKNOWLEDGMENT

The authors appreciate the contributions of CONACYT and BUAP VIEP projects to be possible this research.

REFERENCES

[1] K. Geon-Hwan, N. Jae-Choong and M. Imtiaz , *Multi-Drone Control and Network Self-Recovery for Flying Ad Hoc Networks.* Eighth International Conference on Ubiquitous and Future Networks (ICUFN), 2016, pp. 148-150.

[2] Md. Hasan Tareque, Md. Shohrab Hossain and M. Atiquzzaman*On the Routing in Flying Ad hoc Networks.* Proceedings of the Federated Conference on Computer Science and Information Systems, 2015, pp. 1-9.

[3] P. Mohapatra and S. V. Krishnamurthy, *AD HOC NETWORKS Technologies and Protocols.* Boston, USA: Springer Science and Business Media, 2005, pp. 185.

[4] S. Taneja and A. Kush, *A Survey of Routing Protocols in Mobile Ad Hoc Networks.* International Journal of Innovation,Management and Technology, Vol. 1, No. 3, August 2010, pp. 279285.

[5] G. Jayakumar, and G. Gopinath, *Ad Hoc Mobile Wireless Networks Routing Protocols A Review.* Journal of Computer Science ,Science Publications, Vol. 3, No. 8, 2007, pp. 574-582.

[6] S. Tabatabaei, M. Teshnehlab and S. J. Mirabedini, *A New Routing Protocol to Increase Throughput in Mobile Ad Hoc Networks.* Wireless Pers Commun, 2015, pp. 1766-1778, DOI: 10.1007/s11277-015-2475-2.

[7] S. Mohseni, R. Hassan, A. Patel, and R. Razali,*Comparative Review Study of Reactive and Proactive Routing Protocols in MANETs.* ,4th IEEE International Conference on Digital Ecosystems and Technologies, 2010, pp.304-309.

[8] P. Sharma, A. Kalia AND J. Thakur,*PERFORMANCE ANALYSIS OF AODV, DSR AND DSDV ROUTING PROTOCOLS IN MOBILE AD-HOC NETWORK (MANET).* Journal of Information Systems and Communication, Volume 3, Issue 1, 2012, pp.322-326.

[9] Q. Razouqi A. Boushehri, M. Gaballah and L. Alsaleh,*Extensive Simulation Performance Analysis for DSDV, DSR and AODV MANET Routing Protocols.* 27th International Conference on Advanced Information Networking and Applications Workshop, 2013, pp. 335-342, DOI 10.1109/WAINA.2013.239.

[10] Bekmezci A. Lker Sahingoz. Ozgur KorayTemel. Samil. *Flying ad-hoc networks (fanets): A survey.* Ad-Hoc Networks, Vol. 11, 2013, pp. 1254-1270.

[11] Alamsyah and Purnomo H., *Performance of The Routing Protocols AODV, DSDV and OLSR in Health Monitoring Using NS3.* International Seminar on Intelligent Technology and Its Application, 2016, pp. 323-328.

[12] Fall K. and Varadhan K. *The ns Manual*, November 4, 2011, [Online]. Available: https://www.isi.edu/nsnam/ns

Simulation platform for a VANET using the TrueTime toolbox: further result toward Cyber-Physical Vehicle Systems

Alejandro Garcia-Santiago, Josefina Castaneda-Camacho, Jose F. Guerrero-Castellanos, Gerardo Mino-Aguilar and V. Yair Ponce-Hinestroza
Faculty of Electronics, Benemérita Universidad Autónoma de Puebla (BUAP)
Av. San Claudio y 18 sur Ciudad Universitaria Puebla Pue, Mexico. Email: see http://www.ece.buap.mx

***Abstract*—Research on Mobile Ad-Hoc Network (MANET) has received special attention in the last few years due to their particular characteristics as self-organization, scalability, easy deployment and low cost. Vehicle Ad-Hoc Network (VANET) is basically a special form of MANET, which is formed by a group of wireless nodes, which can dynamically form a network to exchange information without the necessity of using a fixed network infrastructure, which make them ideal for use in several applications such as environment sensors, vehicular communications, disaster rescue operations, air/land/navy defense and so on. In this paper a simulation platform for a VANET which integrates communications and control system to achieve stability using TrueTime 2.0 is presented. It is able to use wireless communications, AODV routing protocol, distributed event-based control and gives a graphical representation of communication components (nodes). Besides, given that a trusty communication guarantee the stability of the system and it achieve a consensus, special attention is paid in the wireless communication performance and the metrics of delivery rate, end-to-end delay and throughput.**

The simulation results represent the base for future development of collaborative control protocols for Cyber-Physical Vehicle Systems and show the applicability and usability of the proposed simulation platform in future works.

***Index Terms*—AODV, control event, routing protocols, Simulink, TrueTime2.0, VANET.**

I. Introduction

A cyber-physical system (CPS) is the next generation of system that requires a solid integration of computing, communications, and control technologies to achieve stability, performance, reliability, robustness, and efficiency in dealing with physical systems of many application domains. A cyber-physical vehicle system (CPVS), ranging from automobile to aircraft and marine craft, is composed of tightly-coupled locomotion, computational and communications components. Historically, CPVS development has been driven by advances in closely-related (but not identical) autonomous vehicle research and cooperative vehicle control to increase system capacity and improve safety and efficiency. CPS research generally aims to synergistically integrate control, computing, communications and physical systems in novel ways that take advantage of the interdependent behavior [1].

Currently, because of the features of CPVS there has been an increment in the interest over the multi-agent systems (MAS) that are found related to a wide range of potential applications in areas as physic, biology and engineering. This is due to different sectors require the implementation of dynamic systems, integrated and coordinated by a cooperative, distributed and intelligent scheme. In addition, they can offer flexibility and reconfigurability to preserve their quality and productivity. Example of this is the Industry 4.0 [2].

The term *agent* concerns to the meaning of autonomy that means self-activity for its objectives achievement. Agents interact, collaborate, coordinate and negotiate in a system that was designed and implemented as a multi-agent scenario. A multi-agent architecture is based on cooperative intelligent entities and has been developed for the integration of design, manufacturing and shop-floor control activities [3]. The main coordination types are listed below.

a) Centralized multi-agent coordination (or coordinator agents). In which one central agent undertakes the collection of partial plans from agents, combines them in a plan and solves possible conflicts [3].

b) Decentralized multi-agent coordination (or autonomous agents). In which agents are not controlled by a central agent although they communicate with each other to creating their plans and the solution of possible arguments [3].

Unfortunately, due to the fact that a network is present in the feedback loop, it is possible to appreciate the impact in terms of performance due to delays and/or the losses of data in networks of shared communication. It avoids complying with the real-time specification which may cause the instability of the system [2], [3]. Under this approach, the wireless mobile networks are becoming increasingly popular due to they provide users access whenever they need (communication anytime and anywhere). Relevant features, wireless network can be classified in two types: Infrastructured or Infrastructureless [4], [5].

Infrastructured wireless networks. The base stations are fixed and they have a specific range where the mobile nodes can move while communicating as long as they keep into the coverage area. However, when a node goes out of the range of a base station, it can get into the range of another base station whether there is one available, otherwise this node will be without connection. This kind of network can be seen as the conventional wireless mobile networks.

978-1-5386-6358-5/18/$31.00 ©2018 IEEE

Infrastructureless or MANETs. There are no fixed base stations and the network nodes operate as routers. For this reason, the mobile nodes can move while communicating at anytime. They are changing from one to another position frequently and establishing a route among them to create their own network [5].

Currently, many studies are focused on MANETs due to their dynamic topology, scalability, easy deployment and no centralized administration. In this case, the communication problems faced by MAS can be solved with the application of ad hoc networks; this is called VANET which is basically an ad hoc network oriented to vehicles [4], [6].

Although there are widely information about simulators that allow to create and evaluate an infinity of designs either in control or communication systems (NS2, OMNET++, Matlab, etc.) only a few of them are able to perform simulations which take into account both areas [7], [8]. It means that if a communications system is evaluated using a typically network simulator, many metrics about its performance can be calculated from communications perspective (see section III). However, if the addition of a conventional control is required in order to improve or propose a more complete system, its implementation could be turn complicated due to the software was designed to networks evaluations. Taking this into account, one of the principals trouble that need to be faced by the new simulation are the operation times that each system (control and communication) demand, as well as the interaction between them.

Based on this, our paper present a simulation platform which integrates communications and control system to achieve stability using TrueTime 2.0. The principal goal of this platform is obtaining a better simulation with a close connection to the reality, hence it is able to use wireless communications, AODV routing protocol and distributed event-based control, gives a graphical representation of communication components and control responses. Besides, the principal metrics of evaluation networks can be obtained (see section III).

To evaluate and demonstrate the applicability and usability of the proposed platform, a simulation of four vehicles (MAS) that perform a formation under a control strategy called consensus and a VANET as form of communication between them is done and the results are shown in this manuscript.

II. Simulation platform

The platform was developed using the TrueTime2.0 toolbox that is a free Matlab /Simulink-based library for real-time control systems. This one facilitates co-simulation of controller task execution in real-time kernels, network transmissions, and continuous plant dynamics [9], [10]. Making use of these characteristics, the platform is able to integrate a communications and control system.

The fig. 1 shows a general view of the simulation done. It consists of 4 mobile nodes (vehicles), each node has its communication and control system. The first one is a distributed event-based control and the second is a VANET which use an AODV routing protocol to delivery their packets.

Due to the fact that the control use consensus to reach the desired position, this needs to know its position and that of its neighbors to generate the *event signal*. The control evaluate the system each 100 ms, in others words, the communication system has to guarantee that the requested from other nodes be received before 100 ms.

According to this requirements make it easy to see the importance of a good coordination of both systems to reach the consensus and guarantee the stability of the system. In this case, the communication system is regulated by the control. In fig. 1 is possible to see the event signal generated (continuous blue line) by the control. This indicate to the node that its data (position and velocity) need to be sent to another node. Here, the communication system evaluate if a link is available with the receiver node (nodes inside the transition radio) or a routing protocol is necessary to reach it.

The performance of a VANET is widely related to the efficiency of routing protocols which depends on several factors such as convergence time after topology changes, bandwidth overhead to enable proper routing, and power consumption. In the reference [11] authors evaluate two different protocols for a mobile network of four nodes. According to that results the AODV routing protocol is considered the most efficient. For this reason, the AODV routing protocol is considered in this platform.

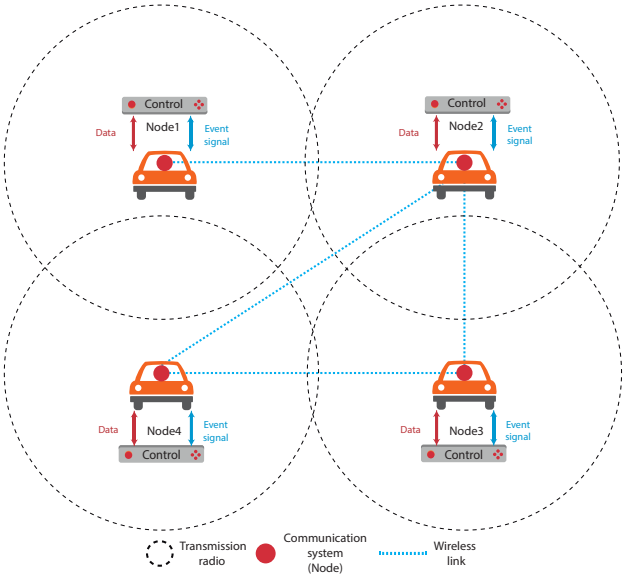


Fig. 1: VANET proposal for 4 robots vehicles.

The fig. 2 presents the simulation platform for a VANET. The blocks within the red line correspond to the control system. This receives the data ($rcvdata1$, $rcvdata2$, $rcvdata3$, and $rcvdata4$) of its neighbors according to the topology shown in fig. 1. According to the values obtained, it decides whether or not it is necessary to send the data to its neighboring nodes through to the *event signal* ($event1$, $event2$, $event3$ and $even4$).

On the other hand, the blocks within the blue line correspond to the communications system. It has four mobile nodes, an animation block and the wireless network block. The

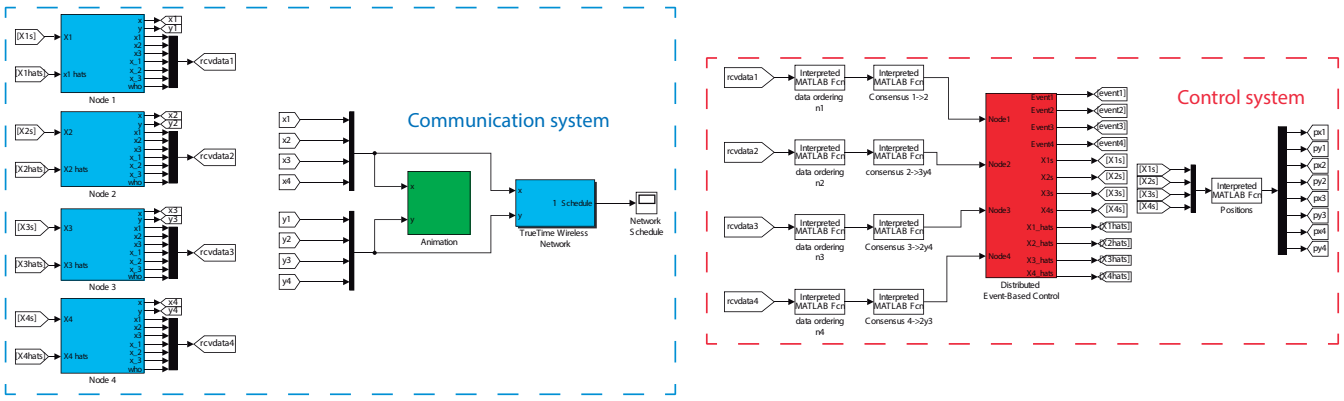


Fig. 2: Simulation platform for a VANET using the TrueTime toolbox.

system works as follows: each node has its position, identifier, data that must be sent (Xns and $Xn\ hats$) and four *event signal* ($eventx$) as shown in fig. 3. Once an *event signal* is received the node look all the nodes inside of its transmission radio. if the receiver node is here the communication is direct if not, it uses the AODV algorithm to find a route through other nodes. It means that the sender node can use others nodes as a router to sent its packets to the receiver even though this is far away of its transmission area.

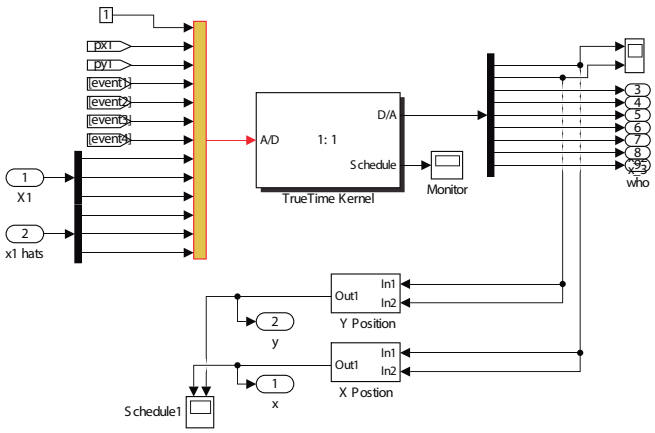


Fig. 3: Diagram of a node.

It is worth mentioning that the simplest and fastest data transfers are generated when a single node transmits in a time slot; however, this does not occurs the most part of the time. Given that there are time slots where multiple agents wish to transmit simultaneously (multiple flags in the same time slot), it is necessary to create a slot that allow each node to transmit at a certain time and thus avoid collisions. Giving solution to this problem, 11 critical cases were found, which are shown in table I. The sending sequence proposed gives the best results in terms of packet delays and delivery.

Fig.4 shows the simulation performed by the complete system on this platform. It consists of four mobile nodes, which start from initials positions and perform a formation over the x-axis (see table II) .

Finally, the table III shows the parameters considered by the communications system.

TABLE I: Possible sending requests generated by the control

Node1	Node2	Node3	Node4	Sending sequence
1	1	1	1	1-2-3-4
1	1	1	0	1-2-3
1	1	0	1	1-2-4
1	1	0	0	1-2
1	0	1	1	1-3-4
1	0	1	0	1-3
1	0	0	1	1-4
0	1	1	1	2-3-4
0	1	1	0	2-3
0	1	0	1	2-4
0	0	1	1	3-4

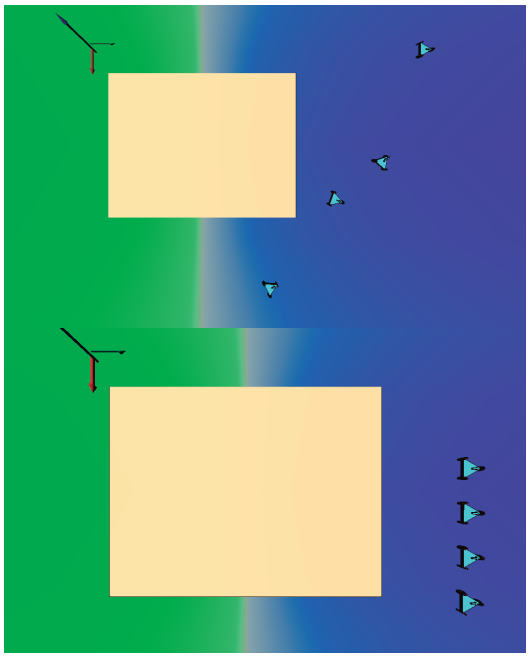


Fig. 4: Simulation of the distributed event-based control using Virtual Reality Toolbox.

TABLE II: Position in meters of the formation realize in the simulation.

	Node 1		Node 2		Node 3		Node 4	
	x1	y1	x2	y2	x3	y3	x4	y4
Inicial position	50	20	20	40	30	60	-40	80
Final position	16.5	50	15.51	49.99	14.5	50	13.5	50

TABLE III: Parameters considered in the TrueTime Wireless Network block.

Parameter	Value
Network type	802.11b
Network Number	1
Number of nodes:	4
Data rate (bits/s)	1M
Minimum frame size (bits)	512
Transmit power (dbm)	24.4994
Receiver signal threshold (dbm)	-45.68
Pathloss exponent	3.5
ACK timeout (s)	0.00004
Retry limit	5
Error coding threshold	0.03

III. Evaluation metrics for the VANET

According to [5] and [12]–[16], it is possible to get the basic metrics which are used to showing the performance of the routing protocols in a VANET.

- **Packet Delivery Rate (PDR)** shows how successful a protocol performs delivering packets from source to destination. It can be characterized as

$$PDR = \frac{R_p}{S_p} * 100 \tag{1}$$

where R_p is the total packets received and S_p is the total packets sent.

- **End-to-End Delay (E2ED)** It is measured in seconds and calculates all possible delays occurred in the networks. The average E2ED can be defined as the time calculated for a data packet to be transmitted from source to destination across a FANET.

$$E2ED = T_R - T_S \tag{2}$$

where T_R is receive time and T_S is sent time.

- **Throughput (Thp)** The average throughput is the average number of bits arrived per second at destination node. This is used as a measure of the reliability of the protocol under different conditions, hence the average throughput in the network needs to be as higher as possible [16]. Mathematically it can be defined as,

$$Average\ thp = \frac{N_p * packetsize}{Seconds} \tag{3}$$

where N_p is the number of bits received successfully by all destinations.

IV. Simulation Results

Once the simulation started, the system sends the data every time the control requires them. At the same time, the system generates a report of its events, which are displayed in the Matlab terminal (as shown in fig. 5).

The simulation takes about 10 sec to achieve the consensus. The fig. 6 shows the trajectory of the nodes generated by the $animationblock$. These positions can be verified with the figures 7 and 8, which indicate the trajectory of the agents on the x and y axes. This responses are obtained from the control block .

```
                                        .
                                        .
                                        .

Time: 2.0969| A new route has been established between Node#2 and Nodo#1
 ---> 2  1 <---
1 data messages in buffer
Sending buffered message 1 to  Node#1
Buffer emptied
Time: 2.0998| A new route has been established between Node#2 and Node#4
 ---> 2  4 <---
1 data messages in buffer
Sending buffered message 1 to  Node#4
Buffer vaciado
Application | Node#3 receiving data: 35  2 3 15.49891 50.00094 -0.0002353861
15.49921  50.00038 -0.0005164576 100 time: 2.1189
                                        .
                                        .
                                        .
```

Fig. 5: Report of events generated during the simulation.

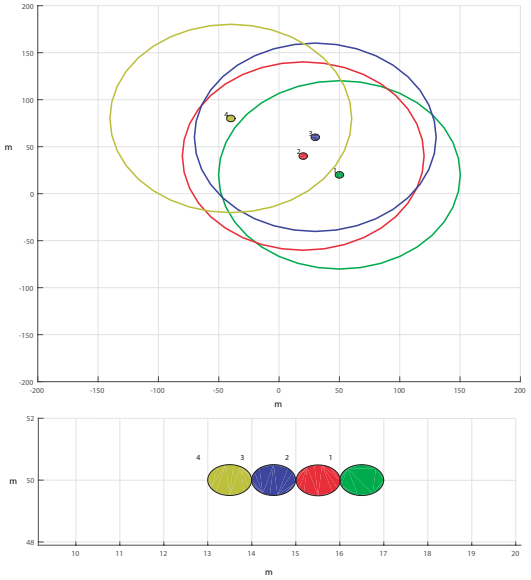


Fig. 6: Animation window of the simulation.

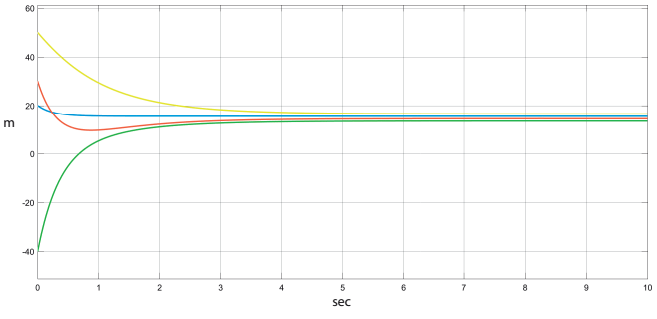


Fig. 7: Response of consensus convergence for x-axis

The platform besides to show the events of the network, it generates three vectors at the end of the simulation: Routing table, Sent data and Received data (see table IV). The idea of these three vectors, arise with the necessity of generate a

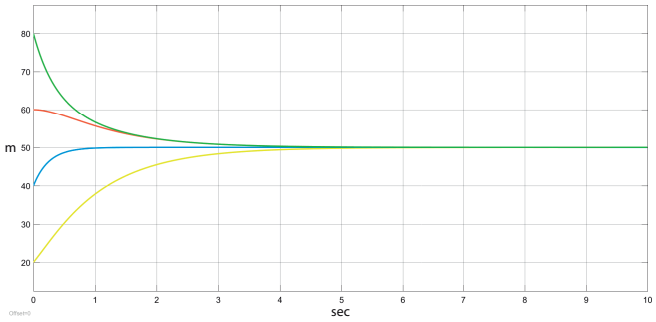


Fig. 8: Response of consensus convergence for y-axis

report similar to that of Ns2. Thanks to this, now it is possible to program an algorithm that allows executing the evaluation metrics discussed in section III.

TABLE IV: structure of $received$ vector

1	1	2	49.9942	20.004	5.69772	50	20	5.7	100	0.0002

Id: package Id
Sender: who send
Receiver: who receive
Data message: the next six field indicate the values requested by the control (Xns y $Xn\ hats$)
Message Size: data message size (bits)
Time (delivery or sending) : shows the time when the node send or delivery a data message

The metrics : Packet Delivery Rate, End-to-End Delay and Throughput (discussed in the sectionIII) are programmed in $.m$ and applicated to the reports generated by the platform. Thus, the performance of the network can be evaluated . The results are shown below.

PDR [%]: the PDR of the network is 97.22%
E2ED [sec]: the E2ED by node is: node1 = 35 ms , node2 =43.1 ms , node3 =39.7 ms, node4 =40.3 ms
The E2ED of the complete network is 39.55ms
Thp [Kbs]: the network presents a Thp of 10.8864 Kbs

According to this results, it can be observed a PDR of 97.22%, which indicates that almost the data are being received. Likewise, the general E2ED of the network is 39.55 ms. it shows that the nodes share their information in a time less than 100 ms. Thank to that, it is possible to say that the VANET proposed is complying with the requirements of the system.

V. Conclusions

A simulation platform for a VANET which integrates communications and control system to achieve stability using TrueTime 2.0 was evaluated successfully. In based on the results obtained in this simulation (figures 6, 7 and 8). It is possible to say that the system achieved a consensus. In others words, the interaction between the communication and control system is given in appropriated way. One of the principal challenges in this platform is the synchronization of the systems times, due to they work with different samples times. However, it was possible to figure out thanks to the execution of tasks and interrupt handlers given by the TrueTime 2.0. On another hand, one limitation of this platform is that can be unfriendly with the user, because of the TrueTime blocks have to be programing and some Simulink blocks has compatibility issues with them. So, it could be necessary to program a block that replace them. For this reason, it is important that the user has programming skills.

Acknowledgment

The authors appreciate the contributions of CONACYT and BUAP VIEP projects to be possible this research.

References

[1] Bradley, Justin M. and Atkins, Ella M., *Optimization and Control of Cyber-Physical Vehicle Systems.* Sensors 2015,15, 2015, pp. 23020-23049, doi:10.3390/s150923020.
[2] Sánchez-Santana, J. P., Guerrero-Castellanos, J. F., Villarreal-Cervantes, M. G. and Ramírez-Martínez, S., *Control distribuido y disparado por eventos para la formación de robots móviles tipo (3,0).* Congreso Nacional de Control Automático, 2017,Monterrey, Nuevo León, México, pp. 493-498.
[3] Georgios, A., Paraskevi, K., Kyriaki, N. and Konstantin-Dionysios, B., *Classification and Review of Multi-Agents Systems in the Manufacturing Section.* 24th DAAAM International Symposium on Intelligent Manufacturing and Automation, 2013 , pp. 282 290 .
[4] P. Mohapatra and S. V. Krishnamurthy, *AD HOC NETWORKS Technologies and Protocols.* Boston, USA: Springer Science and Business Media, 2005, pp. 185.
[5] S. Taneja and A. Kush, *A Survey of Routing Protocols in Mobile Ad Hoc Networks.* International Journal of Innovation,Management and Technology, Vol. 1, No. 3, August 2010, pp. 279285.
[6] Bekmezci A. Lker Sahingoz. Ozgur KorayTemel. Samil. *Flying ad-hoc networks (fanets): A survey.* Ad-Hoc Networks, Vol. 11, 2013, pp. 1254-1270.
[7] University of Southern California ,*The Network Simulator - ns-2*, https://www.isi.edu/nsnam/ns/
[8] OMNet++,*OMNeT++ Discrete Event Simulator*, https://www.omnetpp.org/
[9] Cervin, A., Henriksson, D. and Ohlin, M.,*TRUETIME 2.0 Reference Manual.* Department of Automatic Control Lund University, 2016.
[10] Lund University,*TrueTime: Simulation of Networked and Embedded Control Systems* Department of Automatic Control, http://www.control.lth.se/truetime/
[11] Garcia-Santiago, A., Castaneda-Camacho, J., Guerrero-Castellanos, J.F. and Mino-Aguilar, G., *Evaluation of AODV and DSDV Routing Protocols for a FANET: Further results towards robotic vehicle networks.* IEEE 9th Latin American Symposium on Circuits & Systems (LASCAS), 2018 .
[12] G. Jayakumar, and G. Gopinath, *Ad Hoc Mobile Wireless Networks Routing Protocols A Review.* Journal of Computer Science ,Science Publications, Vol. 3, No. 8, 2007, pp. 574-582.
[13] S. Tabatabaei, M. Teshnehlab and S. J. Mirabedini, *A New Routing Protocol to Increase Throughput in Mobile Ad Hoc Networks.* Wireless Pers Commun, 2015, pp. 1766-1778, DOI: 10.1007/s11277-015-2475-2.
[14] S. Mohseni, R. Hassan, A. Patel, and R. Razali,*Comparative Review Study of Reactive and Proactive Routing Protocols in MANETs.* ,4th IEEE International Conference on Digital Ecosystems and Technologies, 2010, pp.304-309.
[15] P. Sharma, A. Kalia AND J. Thakur,*PERFORMANCE ANALYSIS OF AODV, DSR AND DSDV ROUTING PROTOCOLS IN MOBILE AD-HOC NETWORK (MANET).* Journal of Information Systems and Communication, Volume 3, Issue 1, 2012, pp.322-326.
[16] Q. Razouqi A. Boushehri, M. Gaballah and L. Alsaleh,*Extensive Simulation Performance Analysis for DSDV, DSR and AODV MANET Routing Protocols.* 27th International Conference on Advanced Information Networking and Applications Workshop, 2013, pp. 335-342, DOI 10.1109/WAINA.2013.239.