

Cálculo Científico Distribuido sobre Clientes Móviles Indeterminados en redes MANETs

Tesis Doctoral



Pablo José Iuliano

Director: Ing. Luis Marrone

Dr. Fernando G. Tinetti

Facultad de Informática
Universidad Nacional de La Plata

Informe presentado a fin de obtener el grado de
Doctor en Ciencias Informáticas

Octubre 2021

Quisiera dedicarle esta tesis a mi amada esposa Natalia, mis cariñosos hijos Emma y Juan, y a mi padre a quien extraño todos los días de mi vida.

Publicaciones

A continuación se enumeran y describen brevemente las publicaciones científicas internacionales realizadas en el marco de esta tesis doctoral:

1. Iuliano, Pablo José, Marrone, Luis Armando y Tinetti, Fernando Gustavo. Protocolo para el Cálculo Científico Distribuido sobre Clientes Móviles Indeterminados en MANET. *CONGRESO I^o GENIO 2021. CONGRESO LATINOAMERICANO DE INGENIERÍA, Medellín, Colombia*. Agosto 2021.

En este trabajo se presentó el diseño de un novel protocolo que combina un enfoque para lidiar con la indeterminación de clientes al momento de comenzar un cálculo científico distribuido basado en un diseño cross layer con un mecanismo que agrupa los dispositivos presentes en una red móvil ad hoc o MANET (Mobile Ad hoc Networks) en clusters, sacando rédito de la heterogeneidad inherente a este tipo de redes a fin de determinar que dispositivos serán los líderes de los agrupamientos conformados. La propuesta de solución implementada se basa en el protocolo de enrutamiento reactivo AODV (Ad hoc Ondemand Distance Vector), la cual modifica los formatos de mensajes para dar soporte a toda la nueva funcionalidad.

2. Iuliano, Pablo José, Marrone, Luis Armando y Tinetti, Fernando Gustavo. Analizando la viabilidad de cálculos científicos sobre redes móviles ad-hoc vía micro-benchmarks. *VII Congreso Internacional de Computación y Telecomunicaciones, Lima, Perú*. Septiembre 2015.

En este trabajo, se presentaron los resultados derivados de la aplicación de un conjunto de micro-benchmarks a diferentes dispositivos de una MANET destinada a realizar cálculos científicos en forma distribuida. Este artículo analiza los problemas característicos de los dispositivos que componen este tipo de redes y cómo, por medio de los micro-benchmarks, lograr alcanzar un perfilado adecuado de los mismos. La experiencia aquí presentada contribuye a identificar los ámbitos adecuados para el emplazamiento de una MANET dedicada al cómputo científico distribuido basándose en los perfiles generados de cada uno de los aspectos antes mencionados.

3. Iuliano, Pablo José, y Marrone, Luis Armando. Shannon Entropy in Distributed Scientific Calculations on Mobiles Ad-Hoc Networks (MANETs). *Communications and Network 5.03*. Septiembre 2013.

Este artículo aborda la problemática de proveer una métrica formal para estimar la incertidumbre al momento de comenzar un cálculo científico distribuido entre clientes que funcionan en redes móviles ad hoc. La forma utilizada para dimensionar la incertidumbre relacionada a la concreción exitosa de un cómputo distribuido se basa en la Teoría de la Evidencia Dempster-Shafer(TDS). La Entropía de Shannon es el mecanismo formal mediante el cual se estimó el conflicto en los escenarios planteados en el documento.

Agradecimientos

A mi esposa Natalia y mis hijos Juan y Emma, por su amor incondicional, comprensión y apoyo, que sin ellos no hubiese sido posible lograr finalizar esta ardua travesía.

A mis queridos padres a quienes les debo todo lo que he conseguido en mi vida y que gracias a sus numerosos sacrificios y privaciones lograron educarme como un ser humano de bien.

A mis dos directores por todo su esfuerzo y compromiso durante todo este proceso.

Finalmente, quisiera agradecer en forma especial a Luis por su soporte, apuntalamiento y guía, que sin estos el presente trabajo de investigación jamás se hubiera concretado.

Resumen

La tecnología wireless en los últimos años ha experimentado una expansión sin precedentes, alcanzando hasta el lugar más recóndito de la tierra. La tecnología inalámbrica permite la creación de redes sin la necesidad de cables, dando al usuario de este tipo de redes la libertad de moverse por su entorno sin la restricción de movilidad asociada a tener una conexión cableada. El mejor exponente del salto tecnológico que experimentan las redes wireless son las MANET (*Mobile Ad hoc Networks*, Redes móviles ad hoc) que se presentan como la arquitectura de comunicación más flexible que existe actualmente. Este tipo de redes está integrado por sistemas autónomos móviles con capacidades de enrutamiento y conectados a través de enlaces inalámbricos formando un grafo de comunicación arbitrario.

Utilizar las redes móviles ad hoc para desarrollar un esquema de computación distribuida sobre dispositivos heterogéneos que pueda explotar las crecientes capacidades de cómputo de estos, es la motivación principal del estudio plasmado en este documento. La mayor dificultad que se debe superar es encontrar un mecanismo para manejar la indeterminación de las unidades de red disponibles para balancear la carga en el procesamiento distribuido. Esta indeterminación es provocada mayoritariamente por la movilidad de los nodos que conforman estas redes.

En este trabajo se propone el diseño de un protocolo que combina un enfoque para lidiar con la indeterminación de clientes al momento de comenzar un cálculo científico distribuido basado en un diseño cross layer con un mecanismo que agrupa los dispositivos presentes en una MANET en clusters, sacando rédito de la heterogeneidad inherente a este tipo de redes a fin de determinar que dispositivos serán los líderes de los agrupamientos conformados. La propuesta de solución implementada se basa en el protocolo de enrutamiento reactivo AODV (*Ad hoc On-demand Distance Vector*), la cual modifica los formatos de mensajes para dar soporte a toda la nueva funcionalidad.

El método de validación de la propuesta presentada en esta investigación se delinea a través de la definición de diversas pruebas de rendimiento bajo diferentes valores en los parámetros que fijan: el número de dispositivos, la movilidad, el tamaño del área de circulación, entre otros; que al dimensionar los valores registrados vía métricas estándar permitirán arribar a los resultados de su desempeño.

Índice general

Índice de figuras	XV
Índice de tablas	XIX
1. Introducción e información general	1
1.1. Tecnología wireless	1
1.2. Cómputo simultaneo aplicado al cálculo científico	2
1.3. Clasificación de las redes inalámbricas	4
1.3.1. Clasificación basada en la arquitectura de red	4
1.3.2. Clasificación basada en el área de cobertura de la comunicación	5
1.3.3. Resumen de las redes inalámbricas basada en el área de cobertura de la comunicación	9
1.4. Redes móviles ad hoc	9
1.5. Objetivo	11
1.6. Solución propuesta	12
1.7. Estructura general de la tesis	13
2. Protocolos de enrutamiento, clustering y cross layer	15
2.1. Protocolos de enrutamiento para MANET	16
2.1.1. AODV (<i>Ad hoc On-demand Distance Vector</i>)	19
2.1.2. Resumen de los protocolos de enrutamiento	23
2.2. Clustering en MANET	25
2.2.1. Ventajas del clustering	25
2.2.2. Clasificación de los esquemas de clustering	26
2.2.3. Clustering en protocolos de enrutamiento sobre MANETs	27
2.2.4. Resumen de los protocolos de ruteo que utilizan clustering	30
2.3. Diseño cross layer	31
2.3.1. Comparación entre el diseño en capas y el cross layer	31

2.3.2.	Motivación del diseño cross layer	32
2.3.3.	Ventajas de utilizar una aproximación cross layer en los protocolos de enrutamiento	34
2.4.	Resumen	35
3.	PROCCDIMI: PROtocolo para el Cálculo Científico Distribuido sobre clientes Móviles Indeterminados en MANETs	37
3.1.	Problemas existentes en el funcionamiento de las MANET	38
3.2.	Soluciones a los problemas en los protocolos de red que funcionan sobre MANETs	40
3.3.	Descripción de PROCCDIMI	42
3.4.	Formatos de paquetes utilizados por PROCCDIMI	44
3.4.1.	Formatos de paquetes para el descubrimiento de clientes	44
3.4.2.	Formatos de mensajes del cómputo de distribuido	48
3.4.3.	Resumen de los tipos de paquetes que utiliza PROCCDIMI	49
3.5.	Tabla de clientes	50
3.6.	Generación de los mensajes CREQ, CREP y RERR	51
3.7.	Generación de los mensajes CDREQ, CDREP y RERR	58
3.8.	Diagrama de secuencia del descubrimiento de clientes	60
3.9.	Diagrama de secuencia del cómputo de distribuido	63
3.10.	Administración del clúster	63
3.11.	Resumen	65
4.	Simulación	67
4.1.	Selección del software de simulación	68
4.1.1.	NS2 (Network Simulator version 2)	69
4.1.2.	NS3 (Network Simulator version 3)	70
4.1.3.	GloMoSim (Global Mobile Information System Simulator)	71
4.1.4.	QualNet (QualNet® network simulation software)	71
4.1.5.	OPNET (Optimized Network Engineering Tools)	71
4.1.6.	MATLAB (MATrix LABoratory)	71
4.1.7.	Resumen del software de simulación	71
4.2.	Simulador de Red NS2	72
4.2.1.	Arquitectura de NS2	73
4.2.2.	Formato de las trazas de NS2	74
4.2.3.	Script de ejemplo en OTCL	77
4.3.	Modelos de Movilidad	81

4.3.1. Modelo Random Walk	81
4.3.2. Modelo Random Waypoint	82
4.3.3. Modelo Random Direction (RD)	83
4.3.4. Resumen de los modelos de movilidad	84
4.4. Nodos	85
4.4.1. Tipos de nodos	85
4.4.2. Densidad de nodos	86
4.5. Consideraciones del modelo de simulación	86
4.5.1. Topología	87
4.5.2. Tráfico	87
4.6. Escenarios de simulación	88
4.7. Resumen	89
5. Resultados experimentales	91
5.1. Métricas de rendimiento	92
5.1.1. Tasa de Entrega de Paquetes	92
5.1.2. Carga de enrutamiento normalizada	93
5.1.3. Jitter	93
5.1.4. Throughput	94
5.1.5. Costo energético promedio	94
5.2. Análisis del protocolo PROCCDIMI	94
5.2.1. Parámetros de la simulación	95
5.2.2. Modo descubrimiento de coordinador de cluster	96
5.2.3. Modo asignación de tareas	107
6. Conclusiones y Contribuciones	119
6.0.1. Resumen	120
6.1. Conclusión	122
6.2. Contribuciones	123
6.3. Líneas de trabajo futuras	125
Bibliografía	129
Apéndice A. Escenarios NS2	135
Apéndice B. Script AWK	145
Apéndice C. Visualización de la traza de simulación	151

Índice de figuras

1.1. Esquema de una red wireless sobre infraestructura ya existente	5
1.2. Esquema de una red wireless sin infraestructura	6
1.3. Esquema de una red wireless de área amplia	7
1.4. Esquema de una red wireless de área metropolitana	7
1.5. Esquema de una red wireless de área local	8
1.6. Esquema de una red wireless de área personal	9
1.7. Esquema simplificado de una red móvil ad hoc	10
1.8. Modelo de referencia TPC/IP para MANETs	11
2.1. Formato del paquete RREQ de AODV	20
2.2. Formato del paquete RREP de AODV	21
2.3. Formato del paquete RERR de AODV	21
2.4. Ejemplo de envío de RREQ en AODV	22
2.5. Ejemplo de envío de RREP de AODV	23
2.6. Ejemplo de envío de RERR de AODV	24
2.7. Ejemplo de una red dividida en clusters	25
2.8. Arquitectura en capas y el diseño cross layer	32
3.1. Formato del paquete CREQ	45
3.2. Formato del paquete CREP	46
3.3. Formato del paquete RERR de AODV	47
3.4. Formato del paquete CDREQ	48
3.5. Formato del paquete CDREP	49
3.6. Envío broadcast de CREQ	52
3.7. Respuesta CREP cuando el receptor es coordinador, es el destino del descubrimiento de ruta y no se ha respondido previamente con los datos de filiación a un cluster.	53

3.8. Respuesta CREP cuando el receptor es el destino del descubrimiento de ruta y la solicitud de clustering ya fue contestada	54
3.9. Respuesta CREP cuando el receptor no es coordinador, es el destino del descubrimiento de ruta y la solicitud de clustering no fue contestada aún. . .	55
3.10. Respuesta CREP cuando el receptor es un coordinador intermedio y la solicitud de clustering no fue contestada aún.	55
3.11. Respuesta CREP cuando el receptor es un coordinador intermedio y la solicitud de clustering ya fue contestada.	56
3.12. Respuesta CREP recibida por un nodo intermedio sin cluster asignado y el mensaje CREP transporta información de clustering.	57
3.13. Enlace caído hacia el coordinador	58
3.14. Envío unicast de CDREQ	59
3.15. Envío unicast de CDREP	60
3.16. Envío de RERR por coordinador fuera de alcance	61
3.17. Diagramas de secuencia del descubrimiento	62
3.18. Diagrama de secuencia del cómputo distribuido	63
3.19. Formato del paquete CDHello	64
4.1. Arquitectura de NS2	73
4.2. Arquitectura de NS2 para PROCCDIMI	74
4.3. Formato de traza inalámbrica	75
4.4. Ejemplo de una trayectoria de un nodo usando RW	82
4.5. Ejemplo de una trayectoria de un nodo usando RWP	83
4.6. Ejemplo de una trayectoria de un nodo usando RD	84
5.1. Tasa de entrega de paquetes (PDR) en función del tiempo y la cantidad de nodos. En modo solicitud de coordinador de cluster.	97
5.2. Tasa promedio de entrega de paquetes (PDR) en función de la la cantidad de nodos. En modo solicitud de coordinador de cluster.	98
5.3. Carga de enrutamiento normalizada en función del tiempo y la cantidad de nodos. En modo solicitud de coordinador de cluster.	100
5.4. Carga de enrutamiento normalizada en función de la la cantidad de nodos. En modo solicitud de coordinador de cluster.	101
5.5. Jitter en función del tiempo y la cantidad de nodos. En modo solicitud de coordinador de cluster.	102
5.6. Jitter en función de la la cantidad de nodos. En modo solicitud de coordinador de cluster.	103

5.7. Throughput en función del tiempo y la cantidad de nodos. En modo solicitud de coordinador de cluster.	104
5.8. Throughput en función de la la cantidad de nodos. En modo solicitud de coordinador de cluster.	105
5.9. Costo energético promedio en función del tiempo y la cantidad de nodos. En modo solicitud de coordinador de cluster.	106
5.10. Costo energético promedio en función de la la cantidad de nodos. En modo solicitud de coordinador de cluster.	107
5.11. Tasa de entrega de paquetes (PDR) en función del tiempo y la cantidad de nodos. En modo asignación de tareas.	108
5.12. Tasa promedio de entrega de paquetes (PDR) en función de la la cantidad de nodos. En modo asignación de tareas.	109
5.13. Carga de enrutamiento normalizada en función del tiempo y la cantidad de nodos. En modo asignación de tareas.	111
5.14. Carga de enrutamiento normalizada en función de la la cantidad de nodos. En modo asignación de tareas.	112
5.15. Jitter en función del tiempo y la cantidad de nodos. En modo asignación de tareas.	113
5.16. Jitter en función de la la cantidad de nodos. En modo asignación de tareas. .	114
5.17. Throughput en función del tiempo y la cantidad de nodos. En modo asignación de tareas.	115
5.18. Throughput en función de la la cantidad de nodos. En modo asignación de tareas.	116
5.19. Costo energético promedio en función del tiempo y la cantidad de nodos. En modo asignación de tareas.	117
5.20. Costo energético promedio en función de la la cantidad de nodos. En modo asignación de tareas.	118

Índice de tablas

1.1. Resumen de las redes inalámbricas basada en el área de cobertura	9
2.1. Resumen de los protocolos de enrutamiento	24
2.2. Clasificación de esquemas de clustering	27
2.3. Resumen de los protocolos de enrutamiento que utilizan clustering	31
3.1. Resumen de los tipos de paquetes que utiliza PROCCDIMI	50
3.2. Ejemplo de tabla de clientes para el coordinador del cluster 1	51
4.1. Resumen del software de simulación	72
4.2. Resumen de los modelos de movilidad	85
5.1. Números de nodos, clusters y coordinadores	95
5.2. Parámetros de simulación	96

Lista de abreviaturas

AODV	<i>Ad hoc On-demand Distance Vector</i>
AP	<i>Access Point Wireless, Punto de acceso wireless</i>
C-HELLO	<i>Cluster HELLO</i>
C-OLSR	<i>Cluster-based OLSR</i>
C-TC	<i>Cluster Topology Control</i>
CBR	<i>Constant Bit Rate, Tasa de bits constante</i>
CBRP	<i>Cluster Based Routing Protocol</i>
CDHELLO	<i>Calculation Distributed Hello</i>
CDREP	<i>Calculation Distributed Reply</i>
CDREQ	<i>Calculation Distributed Request</i>
CGSR	<i>Clusterhead Gateway Switching</i>
CLACR	<i>Core Location-Aided Cluster-based Routing</i>
CREP	<i>Coordinador Reply</i>
CREQ	<i>Coordinador Request</i>
DARPA	<i>Defense Advanced Research Projects Agency, Departamento de proyectos avanzados de investigación de los Estados Unidos</i>
DCL	<i>Dispositivo de capacidades limitadas</i>
DCS	<i>Dispositivo de capacidades superiores</i>
DEFER	<i>Deferred Routing</i>

DHCP	<i>Dynamic Host Configuration Protocol</i> , Protocolo de configuración dinámica de host
DNS	<i>Domain Name System</i> , Sistema de nombres de dominio
DOS	<i>Denial Of Service</i> , Denegación de servicio
DSDV	<i>Destination Sequenced Distance Vector</i>
DSR	<i>Dynamic Source Routing</i>
GLOMOSIM	<i>Global Mobile Information System Simulator</i>
GPS	<i>Global Positioning System</i> , Sistema de posicionamiento global
GUI	<i>Graphic User Interface</i> , Interfaz gráfica de usuario
IEEE	<i>U.S. Institute of Electrical and Electronics Engineers</i>
LCC	<i>Least Cluster Change</i>
MAC	<i>Media Access Control</i> , Control de acceso al medio
MANET	<i>Mobile Ad hoc Networks</i> , Red móvil ad hoc
MATLAB	<i>MATrix LABoratory</i>
MPRs	<i>Multipoint Relays</i>
NAM	<i>Network AniMator</i>
NRL	<i>Normalized Routing Load</i> , Carga de enrutamiento normalizada
NS2	<i>Network Simulator version 2</i>
NS3	<i>Network Simulator version 3</i>
OCRP	<i>On-Demand Clustering Routing Protocol</i>
OLSR	<i>Optimized Link State Routing</i>
OPNET	<i>Optimized Network Engineering Tools</i>
ORC	<i>On-Demand Routing-based Clustering</i>
OSPF	<i>Open Shortest Path First</i>

OTCL	<i>Object-oriented Tool Command Language</i>
PDA.....	<i>Personal Digital Assistant, Asistente Digital Personal</i>
PDR.....	<i>Packet Delivery Ratio, Tasa de Entrega de Paquetes</i>
PROCCDIMI ..	<i>PROtocolo para el Cálculo Científico Distribuido sobre clientes Móviles Indeterminados en MANETs</i>
QoS	<i>Quality Of Service, Calidad de servicio</i>
QRY	<i>Query</i>
QUALNET	<i>QualNet® network simulation software</i>
RD	<i>Random Direction</i>
RERR.....	<i>Route Error</i>
RRD	<i>Realistic Random Direction</i>
RREP	<i>Route Reply</i>
RREQ	<i>Route Request</i>
RW.....	<i>Random Walk</i>
RWP	<i>Random Waypoint</i>
TC	<i>Topology Control, Control de topología</i>
TCP.....	<i>Transmission Control Protocol, Protocolo de control de transmisión</i>
TLV	<i>Type-Length-Value</i>
TORA.....	<i>Temporally-Ordered Routing Algorithm</i>
TTL.....	<i>Time to live</i>
UPD.....	<i>Update</i>
VINT.....	<i>Virtual InterNetwork Testbed</i>
WEP	<i>Wired Equivalent Privacy, Privacidad equivalente a cableada</i>
WLANS	<i>Wireless Local Area Networks, Redes wireless de área local</i>

WMANS	<i>Wireless Metropolitan Area Networks</i> , Redes wireless de área amplia
WPA	<i>Wireless Protected Access</i> , Acceso inalámbrico protegido
WPANS	<i>Wireless Personal Area Networks</i> , Redes wireless de área personal
WWANS.....	<i>Wireless Wide Area Networks</i> , Redes wireless de área amplia
ZHLS	<i>Zone-based Hierarchical Link State</i>
ZRP	<i>Zone Routing Protocol</i>

Capítulo 1

Introducción e información general

1.1. Tecnología wireless

La tecnología wireless en los últimos años ha experimentado una expansión sin precedentes, llegando hasta el lugar más recóndito de la tierra. La tecnología inalámbrica permite la creación de redes sin la necesidad de cables, dando al usuario de este tipo de redes la libertad de moverse por su entorno sin la restricción de movilidad asociada a tener una conexión cableada.

De acuerdo al criterio dominante actualmente este tipo de redes tienden a ser centralizadas en torno a un dispositivo inalámbrico estacionario el cual ofrece el servicio de acceso a la red, sumado en la mayoría de los casos el acceso a internet. Sin embargo, los dispositivos de computación móvil continúan reduciendo sus precios e incrementando sus capacidades de cómputo; volviéndose cada vez más interesantes como alternativa barata de computación distribuida para cálculos científicos, sumado al hecho que los bajos precios han popularizado la computación móvil y en ese sentido la exploración en cuanto a la posibilidad de conformar de una MANET (*Mobile Ad hoc Networks*, Red móvil ad hoc) a gran escala destinadas al cómputo científico se está convirtiendo plausible.

Las redes móviles ad hoc tienen que ver, y están íntimamente relacionadas, con la evolución de los dispositivos electrónicos. Esta revolución se materializa a través de la miniaturización de los dispositivos y la mayor potencia de sus recursos. Así hoy es posible realizar simulaciones numéricas para estudiar y analizar fenómenos utilizando métodos computacionales, los que se volverían prohibitivos si se pretendiera realizarlos por métodos empíricos.

La existencia de numerosos y variados cálculos científicos en diversos campos de las ciencias se ha producido como consecuencia del crecimiento exponencial de las propiedades

y capacidades de los semiconductores que fueran desarrollados por la industria en los últimos años.

De acuerdo a la ley de Moore [79], el número de transistores, y por ende la capacidad de almacenamiento y cálculo de un circuito; se duplica cada uno o dos años.

Cuando el objetivo de una investigación científica es obtener información válida de algún entorno con el cual se interactúa, la mayoría de las veces es prácticamente imposible establecer vínculos fijos permanentes, en otras palabras enlaces cableados normales. Para estos escenarios se ha establecido una nueva tecnología, basada en un nuevo paradigma para los sistemas de comunicación, que han venido a denominarse MANET.

Las MANETs han trascendido el ámbito académico-científico, habiéndose vuelto una realidad en cuanto a la utilidad práctica de la tecnología. El alcance y diversidad de las aplicaciones de este tipo de redes, los requerimientos, diseños y plataformas son virtualmente ilimitados.

1.2. Cómputo simultáneo aplicado al cálculo científico

Este apartado introduce sucintamente las principales alternativas para desarrollar aplicaciones que hagan uso de la potencia de cálculo de varios dispositivos con capacidades computacionales trabajando en forma simultánea para resolver un problema de cómputo. En primer lugar, se define una serie de conceptos básicos de la computación paralela y distribuida, como lo son: a) la computación concurrente, b) los procesos, c) los hilos y d) la sincronización. Otro concepto que se presenta es el de sistemas de computación de altas prestaciones heterogéneos junto con los clusters en relación a la computación distribuida y el ejemplo más significativo de su aplicación; como así los paradigmas de programación distribuida más utilizados en los ámbitos de aplicación. Finalmente se introducen las alternativas actuales más relevantes para el desarrollo de aplicaciones sobre sistemas de computación heterogéneos.

La computación paralela ha sido utilizada ampliamente a lo largo de la historia para la resolución de problemas en cálculos científicos debido a que acortaban significativamente los tiempos de cómputo, ya que separaban el problema en partes menos complejas, las cuales se distribuían y se resolvían en forma simultánea. Esto mismo puede ser alcanzado de forma virtual o real. La primera se da cuando varias tareas están asociadas a un solo procesador y se les va asignando slots de tiempo para que puedan ir avanzando en la resolución de su parte del problema. Con lo cual el paralelismo de cómputo es virtual ya que en un mismo instante de tiempo solo una instrucción se está ejecutando; a este tipo de soluciones se las conoce como computación concurrente. La segunda se da cuando existen varios procesadores

para realizar los cálculos y por ello en un mismo instante de tiempo existe más de una tarea ejecutando cálculos; a estas tareas se las conocen como procesos.

Un proceso es una abstracción de un programa en ejecución. Los procesos son unidades independientes de ejecución que contienen su propia información de estado, utilizan sus propios espacios de direcciones y sólo interactúan entre sí a través de mecanismos de comunicación entre procesos administrados por el sistema operativo.

Los hilos o threads son la unidad más atómica para lograr concurrencia/paralelismo. Un solo proceso podría contener múltiples hilos, todos los hilos dentro de un proceso comparten el mismo estado y espacio de memoria, y pueden comunicarse directamente entre sí, debido a que comparten las mismas variables.

Independientemente de, si se trata de procesos o hilos, la sincronización en la resolución de cálculos paralelos colaborativos es necesaria.

La sincronización se entiende como la espera de alguna señal o mensaje de parte de una tarea, proceso o hilo, en pos de obtener resultados correctos, o cuando existe una dependencia de una tarea que provea un resultado parcial a otra para que esta última pueda continuar su ejecución.

Una alternativa económica para implementar sistemas de computación en paralelo son los sistemas denominados “sistemas de computación de altas prestaciones heterogéneos o clústers”. Estos se componen de múltiples máquinas, a priori, de distintas características conectadas mediante una red de comunicación, las cuales trabajan en forma conjunta para resolver cálculos en forma distribuida. Los clusters están conformados por PCs y son soluciones económicas para obtener ambientes de altas prestaciones. Uno de los proyectos emblemáticos en el área de la computación distribuida sobre sistemas heterogéneos fue el proyecto SETI@home [51], el cual estaba dirigido por un grupo de investigadores del Laboratorio de Ciencias Espaciales de la Universidad de California, Berkeley, que intentaba utilizar la potencia de cálculo de miles de computadoras heterogéneas hogareñas para realizar una búsqueda de señales de radio sensibles de civilizaciones extraterrestres.

En cuanto a los paradigma podemos mencionar que: En el paradigma maestro-esclavo los problemas son divididos en pequeñas partes, en cuanto a complejidad, siempre y cuando la naturaleza del mismo lo permita, para luego ser asignadas a tareas independientes encargadas de resolverlas. Al no existir dependencias entre los datos, cada una de estas tareas pueden ser procesadas en paralelo, a dichas tareas se las denominan esclavos. Existe otra tarea, denominada maestro que tiene la responsabilidad de realizar la división, asignación y distribución del trabajo a las tareas esclavas. Una vez que todas las tareas tienen asignado su trabajo particular a realizar, comienzan inmediatamente a procesar para obtener un resultado. Al finalizar el procesamiento, cada esclavo devuelve el resultado obtenido al maestro. La

última instancia es la compilación de todos resultados parciales en el resultado del cómputo a resolver, esta responsabilidad recae sobre la tarea maestro.

En lo que refiere al paradigma de programación pipeline, podemos mencionar, que es necesario particionar el conjunto de tareas en n partes, cada una de las cuales se denominan etapas del algoritmo. Una vez establecidas estas etapas, se reparten y resuelven en paralelo, manteniendo una configuración de arreglo unidimensional de nodos computacionales paralelos. Cada etapa i en el arreglo unidimensional, se alimenta de datos que provienen de la tarea encargada de resolver la etapa $i-1$ que son enviados al nodo computacional que se encuentra a su derecha. Es habitual encontrar configuraciones con una estructura de anillo unidireccional en el que se virtualiza el pipeline, esto es debido a que la cantidad de tareas a realizar supera a la cantidad de nodos computacionales paralelos disponibles.

Como mencionamos anteriormente, las computadoras pertenecientes a un sistema de cómputo distribuido que se encuentran interconectadas mediante una red local conforman los clusters; mientras que por otro lado, si los equipos que pertenecen a un sistema de cómputo distribuido se encuentran distantes unos de otros y utilizan canales de comunicación que abarcan distancias mucho mayores, como en Internet, se estaría hablando de computación Grid [41] o Internet Computing [77]. A partir de estos últimos tipos de sistemas computacionales distribuidos surge un nuevo modelo que propone ofrecer recursos de cálculo como un servicio, a esto último se lo conoce como computación en la nube o Cloud Computing [84] [6].

Las redes móviles ad hoc [30] se presentan como la arquitectura de comunicación más flexible que existe actualmente, en base a esta gran ventaja se motiva la exploración de montar una solución a fin de obtener un esquema para la computación distribuida sobre dispositivos heterogéneos que pueda explotar las crecientes capacidades de cómputo de los nodos que conforman este tipo de red y manejar la indeterminación existente asociada a la movilidad de los mismos.

1.3. Clasificación de las redes inalámbricas

De acuerdo al criterio que se elija las redes wireless se pueden clasificar, basándose en su arquitectura o basándose en el área de cobertura de la comunicación.

1.3.1. Clasificación basada en la arquitectura de red

Cuando clasificamos las redes inalámbricas en base a su arquitectura surgen dos grupos [10] y estos son:

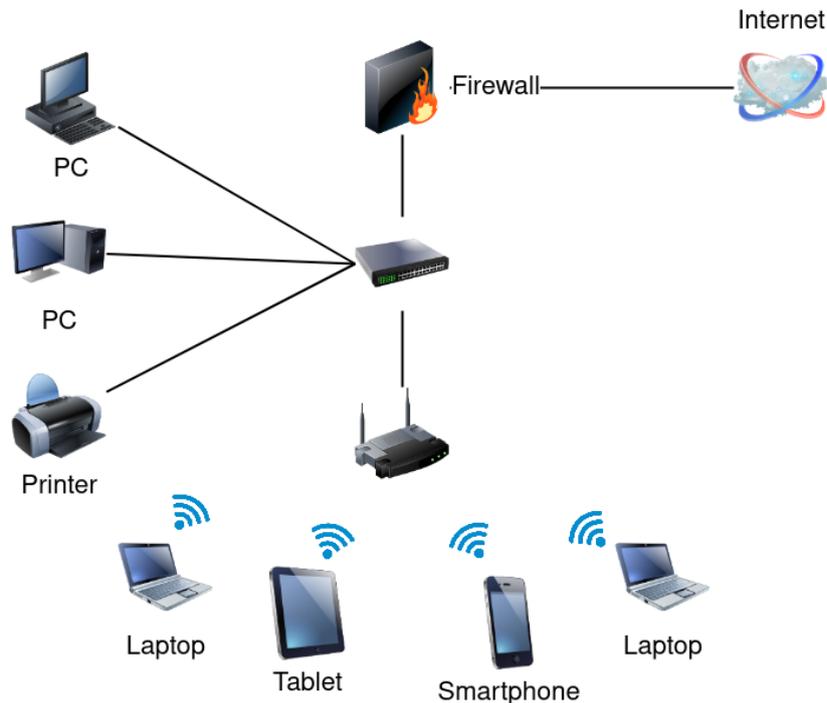


Figura 1.1 Esquema de una red wireless sobre infraestructura ya existente

1. Redes inalámbricas montadas sobre infraestructura ya existente: Son redes que tienen una infraestructura preconstruída a partir de una red fija o cableada. Los requerimientos de servicios de red son delegados a la estructura preestablecida. La Figura 1.1 provee un ejemplo de lo anteriormente desarrollado.
2. Redes sin infraestructura: Una red sin infraestructura es una que se forma dinámicamente a través de la cooperación de un conjunto arbitrario de dispositivos inalámbricos. Usualmente los nodos poseen capacidades de enrutamiento de paquetes y por consiguiente puede tomar decisiones de forma independiente basándose en el estado de la red. En la Figura 1.2 se ejemplifica de lo anteriormente mencionado.

1.3.2. Clasificación basada en el área de cobertura de la comunicación

Las redes inalámbricas se pueden clasificar en cuatro grupos específicos según el área de aplicación y el alcance de la señal, los cuales se detallan a continuación [10] [74].

WWANs (*Wireless Wide Area Networks*, Redes wireless de área amplia)

Están basadas en infraestructura de red que se construyen a partir de otras ya existentes permitiendo al usuario móvil establecer conexión inalámbrica con redes remotas. Estas

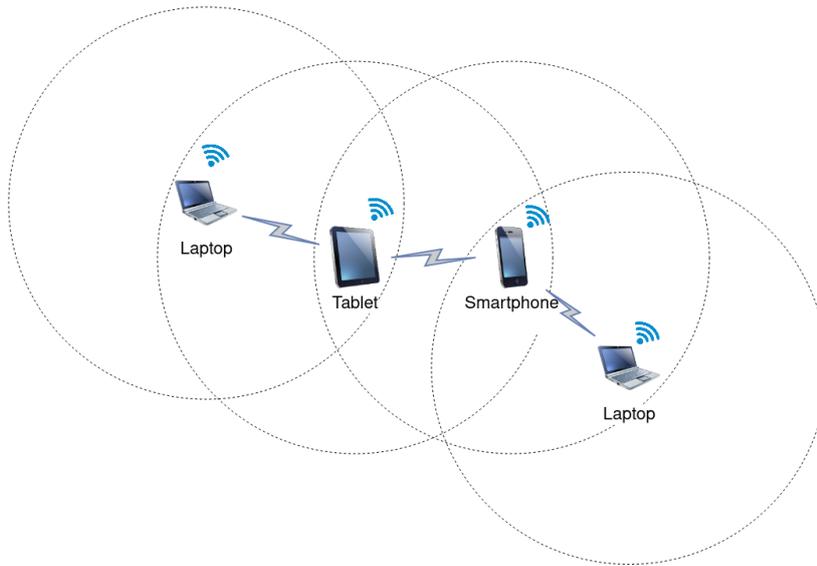


Figura 1.2 Esquema de una red wireless sin infraestructura

conexiones usualmente pueden abarcar áreas geográficas amplias a través del uso de múltiples antenas o satélites suministrados por el proveedor del servicio inalámbrico. En la Figura 1.3 se provee un ejemplo del esquema que podría tener una red de este tipo.

WMANs (*Wireless Metropolitan Area Networks, Redes wireless de área amplia*)

Al igual que las WMANs son redes basadas en infraestructura y posibilitan al usuario de las mismas establecer conexiones inalámbricas de banda ancha a lo largo de múltiples ubicaciones dentro de un área metropolitana. La transmisión de datos puede ser implementada a través del uso de ondas de radio o luz infrarroja. El IEEE (*U.S. Institute of Electrical and Electronics Engineers*) definió el estándar 802.16 para soportar el desarrollo e implementación de las WMANs. La Figura 1.4 muestra un esquema posible para una red WMANs.

WLANs (*Wireless Local Area Networks, Redes wireless de área local*)

Las conexiones en este tipo de redes tienen cobertura dentro de un área local, típicamente podrían ser dentro del edificio de una empresa, campus universitario o espacio público. El rango de cobertura de las conexiones tiene como máximo una distancia de 100 m. Las WLANs pueden estar basadas en infraestructura o no tener infraestructura alguna. En el primer caso las estaciones wireless se conectan a un AP (*Access Point Wireless, Punto de acceso wireless*) que sirven como puentes entre las estaciones inalámbricas y el backbone. En el segundo tipo se presentan múltiples nodos inalámbricos ubicados dentro de un área acotada que temporalmente forman una red sin la necesidad de APs. En este caso los nodos

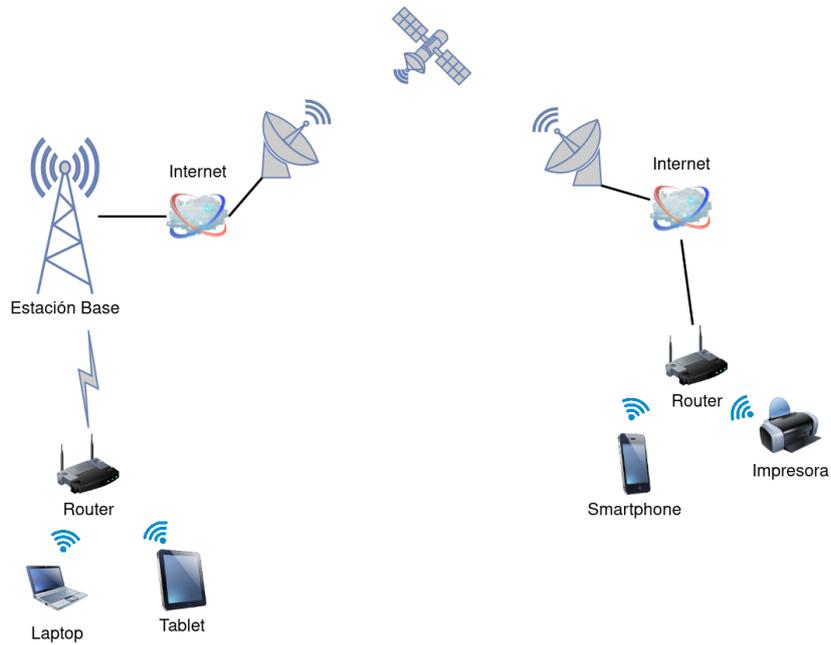


Figura 1.3 Esquema de una red wireless de área amplia

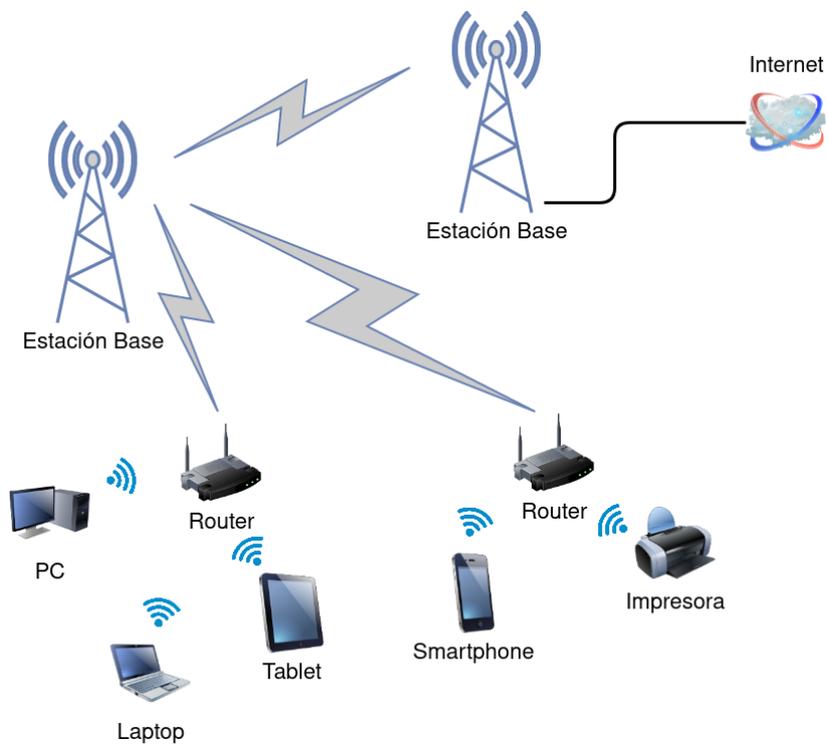


Figura 1.4 Esquema de una red wireless de área metropolitana

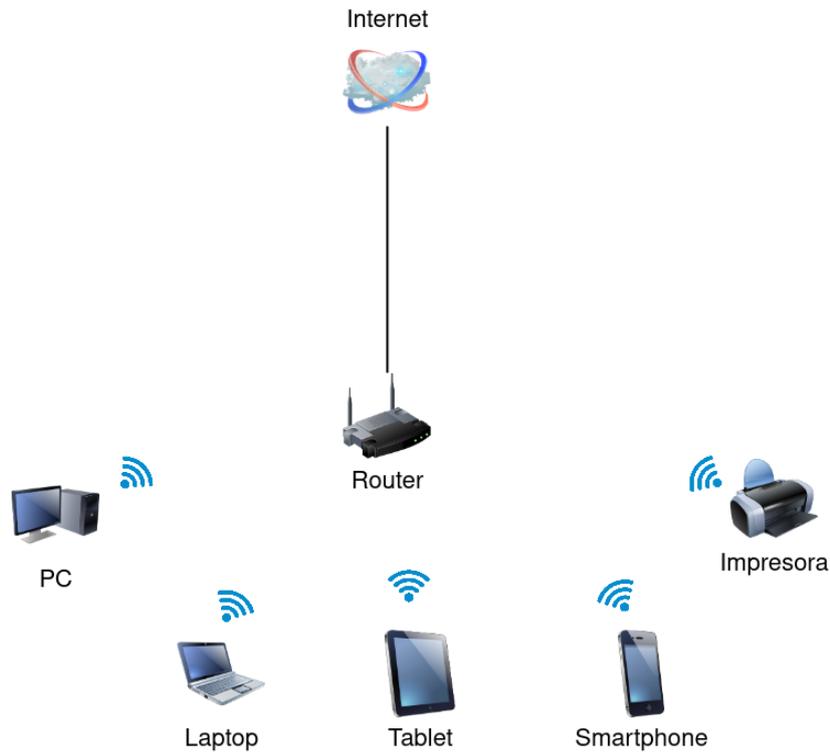


Figura 1.5 Esquema de una red wireless de área local

no necesitan tener acceso a recursos externos a la red. El ejemplo más común para este tipo de red es graficado en la la Figura 1.5.

WPANs (*Wireless Personal Area Networks*, Redes wireless de área personal)

Las WPANs facilita a los usuarios de las mismas establecer conexión ad hoc entre distintos dispositivos inalámbricos de uso personal como PDA (*Personal Digital Assistant*, Asistente Digital Personal), smartphone, smartwatches, tablets, etc. situados dentro de un espacio operacional personal. Una WPAN opera sin infraestructura y en el rango de cobertura de los 10 m. Las tecnologías que se utiliza para el establecimiento de conexión entre los nodos son el bluetooth, la Figura 1.6 muestra un ejemplo de WPAN conectada vía bluetooth, y la luz infrarroja. La implementación de este tipo de red es usualmente de baja complejidad, bajo consumo de energía y son interoperables con redes 802.11.

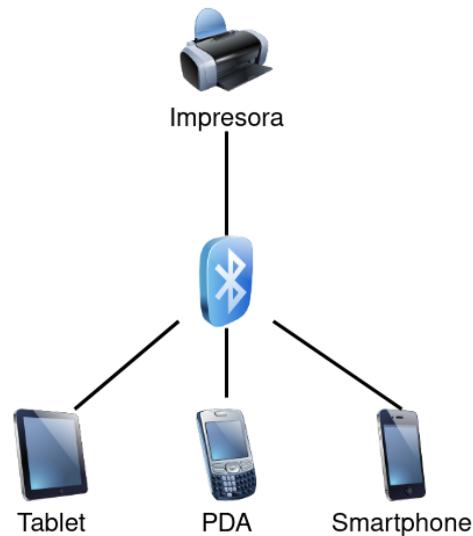


Figura 1.6 Esquema de una red wireless de área personal

Red	Tipo	Cobertura	Tecnología
WWAN	Área amplia	más 50 Km	Telefonía móvil - Satélites
WMAN	Área metropolitana	100m - 50 Km	Torres - Antenas
WLAN	Área local	10m - 100m	Routers inalámbricos o Accesos Points
WPAN	Área personal	hasta 10m	Dispositivos de uso personal (Ej: PDAs)

Tabla 1.1 Resumen de las redes inalámbricas basada en el área de cobertura

1.3.3. Resumen de las redes inalámbricas basada en el área de cobertura de la comunicación

En la Tabla 1.1 se presentan las redes inalámbricas clasificadas según el área de cobertura de la comunicación que fueron analizadas en el apartado 1.3.2, junto con sus características más sobresalientes.

1.4. Redes móviles ad hoc

Formalmente las redes móviles ad hoc o MANET se pueden definir como sistemas autónomos de nodos móviles con capacidades de enrutamiento y conectados a través de enlaces inalámbricos formando un grafo de comunicación arbitrario.

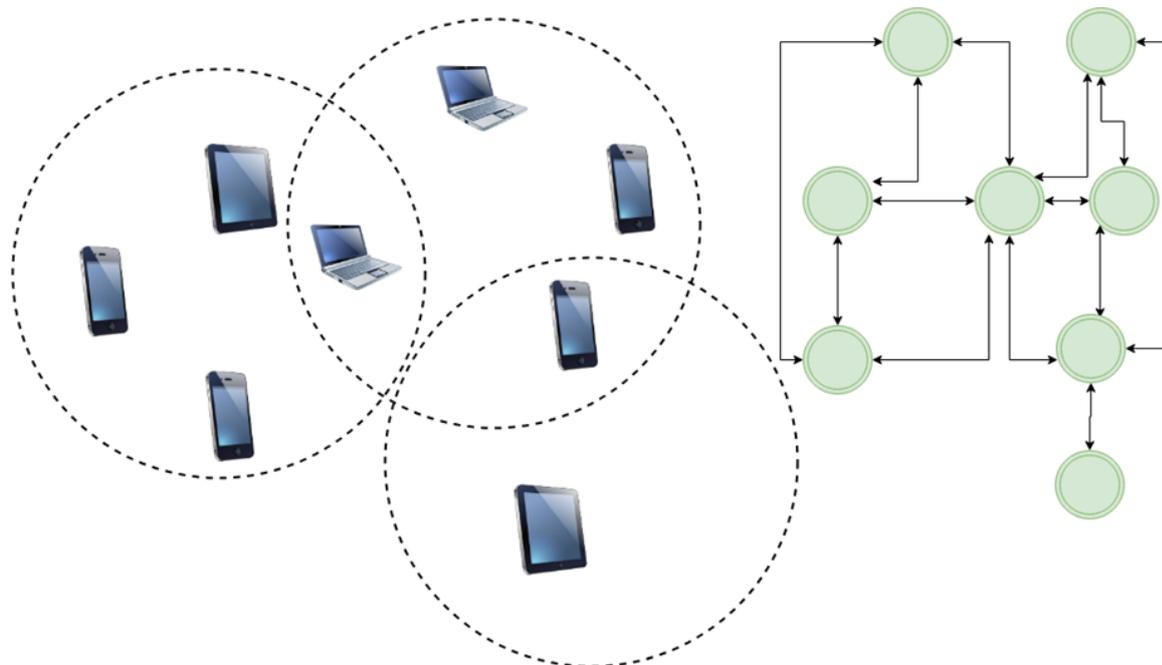


Figura 1.7 Esquema simplificado de una red móvil ad hoc

Con respecto al modelo de referencia TCP/IP, las redes móviles ad hoc definen una capa adicional entre la capa de red y las tecnologías inalámbricas subyacentes. Esta capa es necesaria debido a que este tipo de redes deben mantener un solo dominio de difusión en entornos donde la visibilidad del nodo está limitada por el rango de transmisión de la interfaz inalámbrica.

En este tipo de redes la topología varía dinámicamente y libre a medida que los nodos se van desplazando por su entorno. Básicamente el modo de operación de estas redes es peer-to-peer multihop, donde los paquetes de información son transmitidos de manera de almacenamiento y reenvío desde un origen a un destino arbitrario pudiendo involucran varios o ningún intermediario para alcanzar al destino.

Los patrones de movilidad de los nodos que componen las MANET generalmente son diferentes entre sí; con lo cual se presentarán nodos con gran movilidad, mientras otros tendrán características mayoritariamente estacionarias; complicando la tarea de predecir patrones de movimiento de los dispositivos que conforman este tipo de red.

Las problemáticas asociadas a las cuestiones energéticas deben ser consideradas debido a que, muchos o todos los nodos que componen las redes móviles ad hoc utilizan baterías o alguna otra forma de alimentación agotable. Con lo cual para estos nodos se debe considerar

Capa de Aplicación	Http, Https, FTP, SNMP, etc.
Capa de Transporte	TCP, UDP
Capa de Internet	IP
Capa Ad Hoc	AODV, DSR, BATMAN, etc.
Capa Física	

Figura 1.8 Modelo de referencia TPC/IP para MANETs

la presencia de medias de preservación de la energía. El ancho de banda es un problema que está presente en todos los tipos de redes actuales y particularmente en las MANET en donde se hace más evidente esta problemática. Con lo cual el throughput real es menor que la tasa máxima de transmisión de los canales de radio, esta merma radica en el efecto del multihop, el ruido ambiental, la atenuación, etc. Cuestiones de seguridad como posibles ataques de *eavesdropping*, *spoofing* y *denial of service* deben ser abordados cuidadosamente debido a que las redes móviles inalámbricas son más proclives a ser vulneradas a nivel físico que las cableadas por la naturaleza intrínseca de las mismas. Usualmente las redes móviles ad hoc están construidas en ausencia de roles claramente definidos como routers, servidores DNS (*Domain Name System*, Sistema de nombres de dominio) o DHCP (*Dynamic Host Configuration Protocol*, Protocolo de configuración dinámica de host), etc. Esta condición se presenta por la naturaleza intermitente de la existencia de los nodos dentro de la MANET y para proveer un servicio el nodo que publica el mismo debe estar en funcionamiento permanentemente. Aún así dentro de las MANET hay servicios, como por ejemplo el enrutamiento de paquetes que son soportados de manera distribuida y de esta manera la falla de un solo dispositivo no debería afectarlo significativamente.

1.5. Objetivo

El principal objetivo de esta tesis doctoral plasmada en este documento, es el estudio de la resolución de cálculos científicos paralelizables sobre un número de clientes móviles, a priori indeterminados, pensados para funcionar sobre MANET (*Mobile Ad hoc Networks*, Red móvil ad hoc). La contribución de este trabajo será proveer un entendimiento del efecto de la heterogeneidad y la indeterminación de nodos en sistemas distribuidos orientados al cálculo científico, suministrar técnicas para adaptarse y tomar ventaja sobre ello.

Una de las principales problemáticas a resolver será como se manejará la incertidumbre del número de los nodos disponibles dentro de una MANET al momento de comenzar con la resolución de un cálculo científico. Este problema se presenta debido a que los nodos tienen una naturaleza intermitente en cuanto a conectividad y permanencia dentro de este tipo de redes. El esquema computacional propuesto, el cual está dispuesto bajo una filosofía colaborativa, es otra de las causas que complejiza la resolución en el manejo de la indeterminación de nodos. Finalmente, la solución propuesta deberá funcionar de tal manera que los recursos que son aportados por cada una de las componentes del sistema se aprovechen al máximo con el fin de lograr la mayor eficiencia posible.

La heterogeneidad presente en el modelo propuesto, será otra de las problemáticas principales con la cual habrá que lidiar a lo largo de este estudio. Esto último se hace presente al incrementarse en forma considerable el grado de heterogeneidad de los dispositivos que constituyen la red de comunicación, lo cual en la mayoría de los casos resulta en la degradación de la performance general del sistema.

Concretamente se plantean los siguientes objetivos que se intentarán alcanzar para la conclusión de este trabajo de investigación:

- Formar clusters de nodos con el fin de optimizar el consumo de recursos, principalmente el uso de energía en cada dispositivo, y disminuir el tráfico de red.
- Poder diferenciar entre las distintas capacidades de cómputo con que cuenta cada dispositivo y así clasificarlo lidiando con la heterogeneidad presente en este tipo de redes.
- Detectar el arribo y partida de un nodo de la red a fin de manejar la indeterminación del número de nodos con que dispone la red.
- Determinar la herramienta de simulación más apropiada para trabajar en escenarios con nodos móviles, dinámicos, heterogéneos y aplicando modelos de movilidad reales con patrones de movimiento adecuados a la problemática que se analiza.

1.6. Solución propuesta

En este documento se pretende combinar el cálculo científico con clientes móviles pertenecientes a una red móvil ad hoc junto con la práctica de ensayos experimentales para así contrastar los resultados obtenidos.

Los clientes móviles colaborativos funcionarán sobre una MANET con el fin de resolver un cálculo científico en forma distribuida. La computación distribuida estará soportada por

la implementación de un nuevo protocolo de red que será interpretado por todos los nodos participantes de la red. El protocolo de red antes mencionado tendrá en cuenta principalmente las características de intermitencia de conexión y permanencia de los participantes en esta clase de redes.

1.7. Estructura general de la tesis

La presente tesis doctoral se compone de seis capítulos, a continuación se describirá brevemente los cinco capítulos restantes.

En el capítulo 2 se aborda las soluciones más relevantes en el área de los protocolos de enrutamiento que funcionan sobre MANET. Luego se analiza en detalle el protocolo reactivo seleccionado como sustrato sobre el cual se construirá la solución propuesta en este documento. Las distintas técnicas de clustering disponibles a ser aplicadas sobre MANET son presentadas a continuación junto con los principales protocolos de enrutamiento que utilizan clustering analizándolos en detalle. La última sección esta dedicada al diseño cross layer, debido a que el mismo es utilizado para el descubrimiento de nuevos clientes a nivel de la capa de red y no de la capa de aplicación, concluyendo con el listado de los beneficios de utilizar esta técnica.

En el capítulo 3 se introduce la idea principal de la tesis, que es la propuesta planteada para solucionar los problemas existentes en el cálculo científico distribuido con clientes indeterminados. Se continua describiendo los esquemas que basan la propuesta, diseño cross layer e implementación de técnicas de clustering. También se presenta el diseño general del nuevo protocolo, se explica el funcionamiento analítico, la gestión de grupos y por último se pormenoriza el mecanismo para el descubrimiento de nuevos clientes y asignación de tareas.

En el capítulo 4 se listan los diferentes simuladores existentes y se realiza una clasificación de acuerdo a sus rendimientos referente al entorno de utilización y al número de nodos en las simulaciones. Una vez seleccionado el simulador a utilizarse en el trabajo se ahonda en las características del mismo. Luego se abordan los modelos de movilidad seleccionados para ser utilizados en las pruebas de rendimiento y son explorados exhaustivamente describiendo sus características principales. A continuación se presenta la forma de representar la heterogeneidad de los nodos dentro de los escenarios de simulación y se introduce el concepto de densidad de nodos. Por último, se detallan los componentes finales de los escenarios de simulación: la topología y el generador de trafico. El capítulo concluye puntualizando los tipos de escenarios de simulación a utilizarse: escenarios con bajo, medio bajo, medio, medio alto y alto número de nodos.

En el capítulo 5 se presentan el estudio de rendimiento y los resultados obtenidos. Al principio del apartado, se describen las métricas utilizadas para evaluar el rendimiento de la solución aquí planteada, luego se describen todos los parámetros involucrados en la definición de los escenarios y se continúa con el análisis de los resultados obtenidos de las simulaciones realizadas en los dos modos de operación del protocolo propuesto.

Finalmente en el capítulo 6 se discuten los resultados obtenidos, se presentan las contribuciones realizadas y se proponen tareas a ser realizadas en el futuro sobre la tecnología analizada en la presente investigación.

Capítulo 2

Protocolos de enrutamiento, clustering y cross layer

En el capítulo 1 se presentó sucinta y globalmente las tecnologías wireless existentes conjuntamente con las alternativas más difundidas al momento de encarar un cálculo científicos distribuidos sobre una infraestructuras de red tanto homogénea como heterogénea. Se definió y clasificó las redes inalámbricas, centrando la atención especialmente en las redes móviles ad hoc. Finalmente se planteó el objetivo de esta investigación, para luego introducir la solución propuesta y concluir el capítulo detallando la estructura de esta tesis doctoral.

En este capítulo se revisan y clasifican los protocolos de encaminamiento que se utilizan en redes MANET. Se realiza una breve comparativa entre ellos y se escoge el protocolo a utilizar como sustrato de la solución aquí planteada por ser el que mejor se adapta a las necesidades de la red según los criterios esbozados. El estudio antes descrito se desarrolla a lo largo de la sección 2.1.

La sección 2.2 define el concepto de clustering en MANET y cuales son las ventajas de aplicar esta técnica. Se clasifican los esquemas de agrupamiento lógico más relevantes. La sección concluye detallando los protocolos de enrutamiento que utilizan o están pensados para redes divididas en clusters y que entran dentro del ámbito de interés del presente estudio.

Finalmente introduce el concepto de diseño cross layer y se lo compara con la arquitectura de red en capas tradicional. Se justifica la motivación de utilizar dicho diseño y se concluye con la evaluación de las ventajas de utilizar un enfoque cross layer en la implementación de nuevos protocolos de enrutamiento.

2.1. Protocolos de enrutamiento para MANET

En el estudio de las redes el establecimiento de rutas entre los nodos que pertenecen a las mismas ha sido un t3pico muy importante. Las dos aproximaciones consisten en tener definidas las rutas de manera est3tica o que estas se generen autom3ticamente. Dentro de este 3ltimo grupo, los dos modelos m3s utilizados son vector distancia y estado de enlace [55]. El ruteo por vector distancia esta basado en el algoritmo de Bellman-Ford [70], el cual define que en cada nodo se mantenga una tabla de enrutamiento que almacene informaci3n de cual es la distancia que hay que transitar para llegar a cada destino alcanzable. Esta informaci3n es peri3dicamente actualizada. Cuando un dispositivo de la red tiene que enviar un paquete de datos a un destino, busca dentro de su tabla de ruteo la ruta m3s corta y env3a el paquete a trav3s de ella. En el ruteo por estado de enlace, cada nodo propaga el estado de sus enlaces indicando todos los dispositivos alcanzables. Cuando el estado de un enlace cambia, se debe informar mediante un broadcast el evento y la tabla de ruteo debe ser actualizada.

Claramente en las redes m3viles ad hoc, tanto la topolog3a y los estados de los enlaces entre los dispositivos que la componen son dos propiedades que son impredecibles y que constantemente se van modificando, conllevando a que la eficiencia de las t3cnicas antes descritas de vea seriamente degradada. Como consecuencia directa de esta 3ltima contrariedad se han desarrollado varios protocolos de ruteos espec3ficos para este tipo de redes y caen dentro de una de estas tres categor3as: Protocolos proactivos, reactivos o h3bridos [48].

- Los protocolos proactivos mantienen la informaci3n de enrutamiento incluso antes de que se necesite. Todos y cada uno de los nodos en la red mantienen informaci3n de las rutas hacia todos los dem3s nodos. La informaci3n de las rutas se mantiene en las tablas de enrutamiento y se actualizan peri3dicamente a medida que cambia la topolog3a de la red. Las actualizaciones generan una excesiva sobrecarga en la red debido a los cambios frecuentes en las rutas por lo din3mico de este tipo de redes y el ancho de banda limitado que estas poseen.
- Los protocolos reactivos no mantienen informaci3n o generan actividad de ruteo en los nodos de la red si no es necesario establecer comunicaci3n entre ellos. Un nodo al enviar un paquete a un destino, provoca que se busque la ruta y se establezca la conexi3n. El procedimiento anterior es realizado a demanda. El descubrimiento de ruta por lo general se produce inundando toda la red con paquetes de petici3n de ruta. Estos protocolos experimentan cierto retardo asociado a la b3squeda de las rutas o porque las rutas pueden no ser v3lidas durante el tiempo suficiente. Por otro lado, utilizan eficientemente el ancho de banda de la red y el consumo de energ3a, debido a que busca las rutas s3lo cuando se necesitan.

- Los protocolos híbridos son una derivación de los dos primeros, combinan técnicas de los protocolos proactivos y de los protocolos reactivos.

A continuación se detallan los protocolos de enrutamiento que funcionan sobre redes MANET más relevantes para esta investigación.

DSDV (*Destination Sequenced Distance Vector*)

DSDV [34] [49] es típicamente un protocolo de ruteo proactivo el cual en cada nodo mantiene una tabla de ruteo para todos los destinos alcanzables. Las actualizaciones se realizan periódicamente mediante la generación de peticiones broadcast.

DSDV lleva un número de secuencia que se utiliza para evitar la generación de ciclos en el encaminamiento de paquetes. Cuando una ruta entre dos nodos se encuentra caída, se realizan los siguientes pasos:

1. La ruta es borrada de la tabla de ruteo.
2. Se incrementa el correspondiente número de secuencia.
3. La distancia hacia el destino es fijada en infinito.
4. Esta información es comunicada vía broadcast a los demás nodos.

DSDV utiliza dos tipos de mensajes de actualización, uno llamado *full-dump* y otro llamado *incremental*. Los mensajes incrementales se emplean para actualizaciones intermedias entre envíos periódicos *full-dump* de toda la información de enrutamiento.

DSR (*Dynamic Source Routing*)

DSR [44] [46] un protocolo reactivo que establece rutas a demanda. Se inicia un proceso de requerimiento de ruta cuando la forma de llegar a un destino es desconocida. Cuando se recibe un paquete de requerimiento, RREQ (*Route Request*), si el nodo intermedio conoce la ruta al destino responde con un paquete de respuesta, RREP (*Route Reply*), en caso contrario agrega su propia dirección al paquete RREQ y lo reenvía. Este procedimiento se repite hasta que se alcanza el destino o expire el tiempo de vida del paquete. El camino reverso se utilizará cuando el destino o un intermediario no posea información del origen en su tabla de enrutamiento. En el caso que una ruta no se encuentre disponible, un paquete de error será generado por el nodo que detecte dicho evento y la entrada correspondiente será eliminada de la tabla de ruteo.

TORA (*Temporally-Ordered Routing Algorithm*)

TORA [56] [42] es un protocolo reactivo que intenta minimizar la sobrecarga en el enrutamiento controlando el alcance de recepción de los mensajes que envía cuando la topología cambia, además es un protocolo que funciona bajo demanda. En TORA, cada nodo tiene asignado una altura y todos los mensajes tiene que fluir corriente abajo como sucede con el agua, o sea de un nodo con mayor altura a otro de menor altura. Cuando un origen tiene paquetes para un destino, pero no se tiene un enlace descendiente, el origen genera un broadcast con un paquete de consulta, QRY (*Query*), que será reenviado hasta que el mismo alcance un nodo que conozca una ruta valida, o sea el destino. Cada nodo realizará un broadcast enviando un paquete de actualización, UPD (*Update*), conteniendo su propia altura. Otros nodos recibirán el paquete UPD y asignaran su propia altura con valores mayores comparados con el del paquete UPD y enviaran otra vez por broadcast este nuevo valor. De esta manera la ruta queda establecida.

En TORA solo una ruta es descubierta aunque existan múltiples disponibles debido a que cada nodo solo tiene un solo valor de altura que inicialmente esta basado en la distancia hacia el destino.

OLSR (*Optimized Link State Routing*)

OLSR [1] es un protocolo de ruteo proactivo y es una optimización del protocolo de estado de enlace OSPF (*Open Shortest Path First*) adaptado para MANET. OLSR utiliza mensajes de Hello y de TC (*Topology Control*, Control de topología) para descubrir e intercambiar la información del estado del enlace basándose en que los nodos individuales informen acerca de cual es el nodo en el próximo salto para los distintos destinos.

Al ser un algoritmo proactivo, el tiempo de establecimiento de rutas es corto desde que las rutas son conocidas antes de ser usadas. Pero por otro lado OLSR tiene tiempos elevados de convergencia y sobrecarga debido a la información extra en los broadcast periódicos.

ZRP (*Zone Routing Protocol*)

ZRP [11] [83] utiliza esquemas de enrutamiento de los protocolos proactivos, con el fin de mantener actualizada la topología. Opera dentro de una zona limitada a nodos vecinos. Los nodos vecinos son aquellos que se encuentran a determinadas cantidad de enlaces de distancia. Y usa esquemas de ruteo reactivos para nodos que se encuentran fuera de esta zona. Las rutas hacia los nodos dentro de la zona son disponibles de manera inmediata. Para destinos fuera de la zona, ZRP realiza un descubrimiento de ruta, en el que utiliza la información local de las zonas. En ámbitos locales ZRP opera como un protocolo proactivo,

mientras que para ámbitos no locales cambia su modo de operación y adopta una estrategia de protocolo reactivo. Debido a que las zonas son sobrepuestas, ZRP se clasifica como un protocolo de ruteo plano. El comportamiento depende de la topología de la red como de la movilidad de los usuarios de la misma.

ZHLS (*Zone-based Hierarchical Link State*)

ZHLS [43] [86] es un protocolo de enrutamiento híbrido. Los nodos de la red son conscientes de su ubicación geográfica, mediante el uso de sistemas de localización, como el GPS (*Global Positioning System*, Sistema de posicionamiento global). La red es dividida en varias zonas no superpuestas de acuerdo a la información geográfica, basándose en un esquema jerárquico. Cada zona de la red tiene asignado un ID de zona y los nodos con un ID de nodo dentro de cada una de ellas. Cada nodo es capaz de recuperar su zona de acuerdo con su ubicación física, así como el mapa de zonas global es conocido por todos los nodos dentro de la red. ZHLS maneja dos tablas separadas para la información de enrutamiento, una para los datos intra zona y la otra para mantener los datos inter zona. Antes de la transmisión de paquetes, los orígenes primero verifican su tabla de enrutamiento intra zona. Si el destino se encuentra en la misma zona, el paquete es entregado directamente, en caso contrario se envía una solicitud de ubicación a todos los gateways de zona de la red. El gateway de la zona en la que el destino se encuentra envía una respuesta de ubicación al origen conteniendo el ID de zona. Finalmente, el nodo de origen una vez que ha recuperado el ID de zona del destino transmite los datos necesarios.

2.1.1. AODV (*Ad hoc On-demand Distance Vector*)

Como consecuencia directa de que el protocolo de enrutamiento AODV (*Ad hoc On-demand Distance Vector*) es el sustrato en el que se basa la propuesta aquí planteada, se realizará una descripción minuciosa de sus características en este apartado.

AODV [66] [13], es un protocolo de ruteo reactivo para redes móviles ad hoc basado en vector distancias que funciona bajo demanda. Solo comienza el descubrimiento de rutas cuando es requerido y además las rutas se almacenan en tablas de ruteo. Estas últimas son mantenidas y almacenadas en cada nodo de la red. Debido a que las rutas se crean únicamente cuando son requeridas se minimiza el consumo de ancho de banda, la latencia de transmisión y el consumo de energía.

AODV mejora el rendimiento de DSDV en cuanto a creación y mantenimiento de rutas para redes MANETs [69], así mismo es uno de los protocolos reactivos más ampliamente

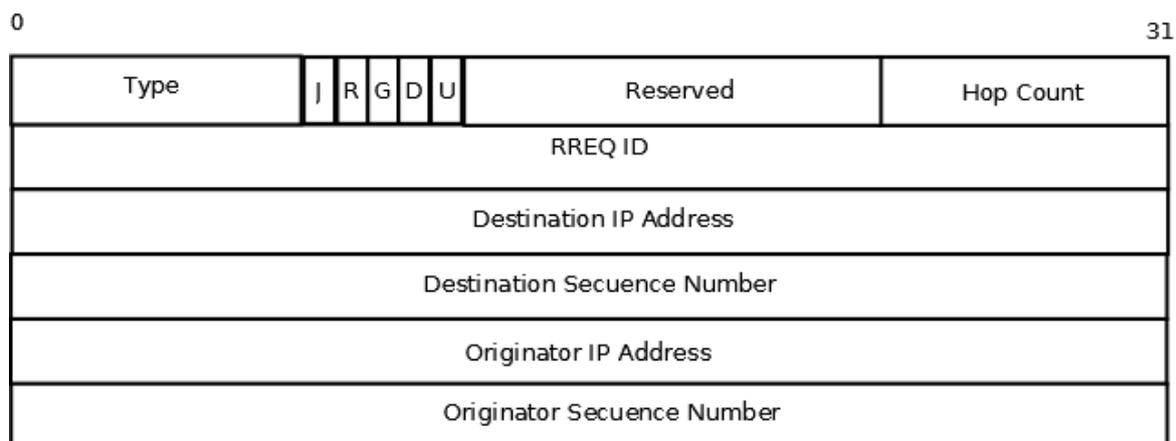


Figura 2.1 Formato del paquete RREQ de AODV [66]

utilizados, por lo cual lo convierte en la elección indicada para que funcione como el substrato en la implementación de nuevos protocolos de enrutamiento cross layers.

Las tablas creadas por AODV guardan información solamente de las rutas hacia los destinos con los cuales se mantiene comunicación, lo cual genera poca sobrecarga en la red y las rutas se van descubriendo conforme se van necesitando.

AODV soporta tres tipos de transmisión: unicast, multicast y broadcast; no obstante, AODV descubre las rutas basándose en peticiones broadcast y las respuestas son enviadas en modo unicast [85]. Inicialmente la tabla de ruteo solamente contiene entradas para los nodos vecinos. Si un origen necesita alcanzar un destino que no es vecino, debe obtener una ruta que le indique como hacerlo. Para ello, el nodo origen envía por broadcast mensajes de descubrimiento de ruta (RREQ). En la Figura 2.1 se muestran los campos que componen dicho mensaje. Si un nodo que no es el destino recibe este paquete, verifica los números de secuencia a fin de determinar si lo ha recibido previamente. En caso de que no lo haya recibido, lo reenvía incrementando el número de saltos y crea el camino inverso para la respuesta RREP, en la Figura 2.2 se muestran los campos del mensaje RREP. La respuesta del establecimiento de ruta RREP es generada por los nodos intermedios que cuentan con una ruta hacia el destino o el mismo destino una vez que el mensaje RREQ le haya llegado. Los nodos envían estas respuestas RREP a las unidades de red que le habían enviado el mensaje RREQ. La ruta se confirma una vez que el RREP llega al origen que había comenzado con el descubrimiento de ruta.

Es necesario mantener el enlace activo entre el origen y el destino después de establecerse una ruta, ya que ésta se mantiene válida durante un periodo de tiempo. Si los nodos que tienen rutas activas realizan envíos periódicos de mensajes RREP ajustando a 1 el TTL, es decir, mediante la difusión local del mensaje a solo los dispositivos situados a 1 salto de

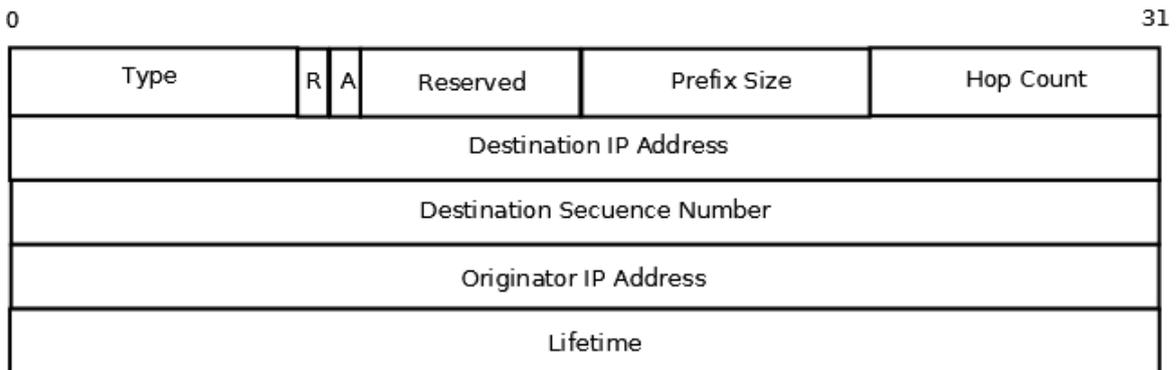


Figura 2.2 Formato del paquete RREP de AODV [66]

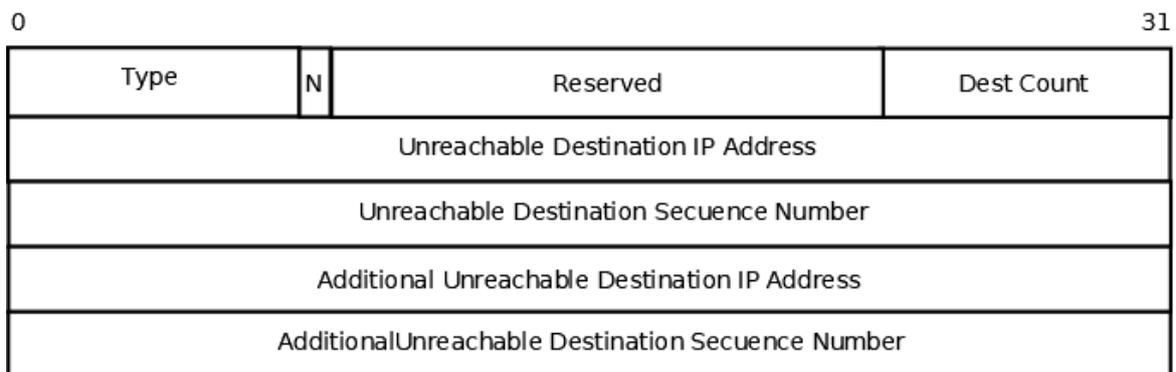


Figura 2.3 Formato del paquete RERR de AODV [66]

distancia este objetivo se asegura. Otro mecanismo adicional para garantizar la validez de las rutas consiste en enviar periódicamente mensajes HELLO entre los nodos vecinos a fin de determinar si se encuentran activos en ese instante en la red y así mantener actualizadas las tablas de rutas y conservar la conectividad entre ellos. Aunque para reducir el volumen de estos mensajes solo debe permitirse su envío a los nodos que estén transmitiendo datos. Si luego de algún tiempo no se reciben mensajes HELLO desde un vecino, se supone que este ya no está disponible, por lo que se desecha la ruta y se borra de la tabla. De esta forma, cuando un nodo detecta la ausencia de un vecino mediante los mensajes HELLO o se produce un error de envío, se origina un mensaje RERR (*Route Error*), que se emplea para invalidar una ruta, cuyos campos se muestran en la Figura 2.3. De modo que, el mensaje RERR llegará hasta el origen de la ruta y este decidirá o no si da comienzo nuevamente al proceso de descubrimiento de ruta.

A continuación se presenta un ejemplo del funcionamiento del protocolo de ruteo AODV. Como se puede apreciar en la Figura 2.4, Figura 2.5 y Figura 2.6 se trata de una red MANET compuesta de ocho dispositivos, los cuales se identifican con las letras A, B, C,..., H. El término RREQ representa una petición de ruta, el RREP representa una respuesta de ruta y

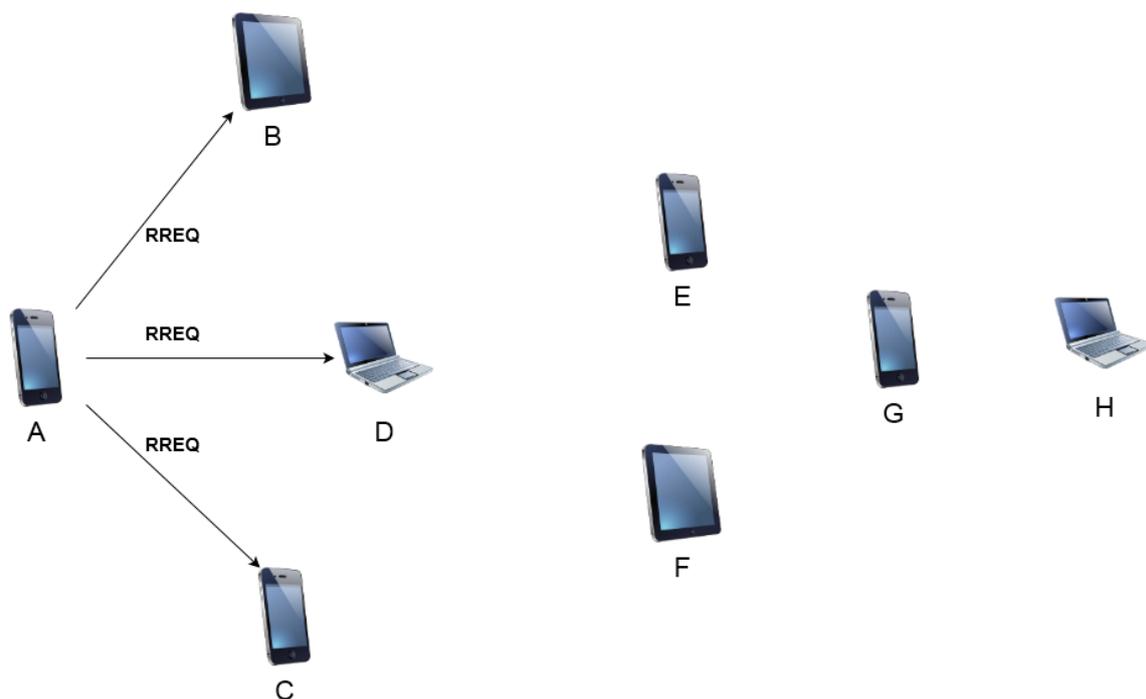


Figura 2.4 El origen A envía por broadcast un mensaje RREQ para descubrir una ruta para alcanzar el destino H. Los nodos intermedios reenvían el mensaje RREQ.

por último el termino RERR representa un error de ruta. El origen A difunde por broadcast un mensaje (RREQ), ver Figura 2.4, en busca de una ruta para alcanzar al destino H. El destino H recibe el mensaje RREQ y responde por unicast enviando un mensaje RREP al origen A para informar que ha recibido el mensaje RREQ, ver Figura 2.5. A partir de ese momento se ha establecido una ruta entre A y H. Si después de un tiempo ocurre algún corte de un enlace intermedio en esa ruta, entonces el A recibirá un mensaje RERR (generado en el ejemplo por G) en el que se informa que ha ocurrido una rotura de enlace, ver Figura 2.6. Cuando ocurre un fallo de este tipo, entonces el origen A decidirá si comienza nuevamente el proceso de descubrimiento de ruta para llegar al destino H.

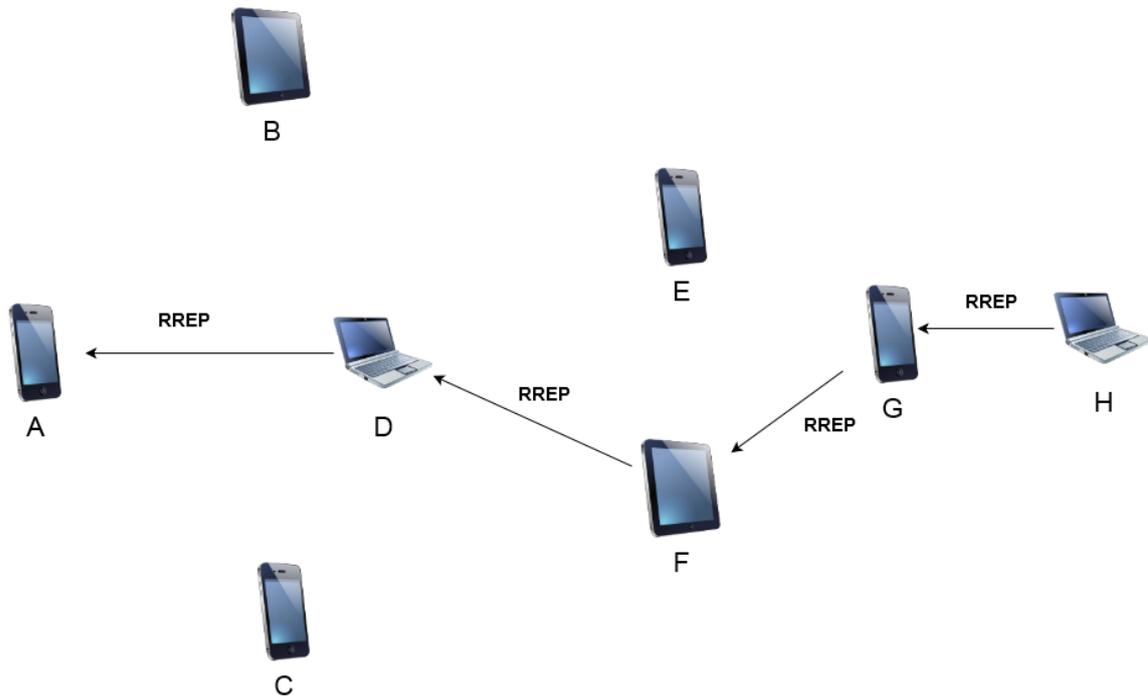


Figura 2.5 El destino H devuelve por unicast un mensaje RREP al origen A con la finalidad de informar que ha recibido el mensaje RREQ. Los nodos intermedios participan en el descubrimiento de ruta reenviando el mensaje RREP.

2.1.2. Resumen de los protocolos de enrutamiento

En la Tabla 2.1 se presentan todos los protocolos de enrutamiento analizados en el apartado 2.1, junto con sus características más sobresalientes. Principalmente se especifica si se trata de un protocolo de ruteo reactivo, proactivo o utiliza una aproximación híbrida. También se indica si funciona de manera distribuida, si requiere de mucha energía para su operación y por último si consume o no mucho ancho de banda.

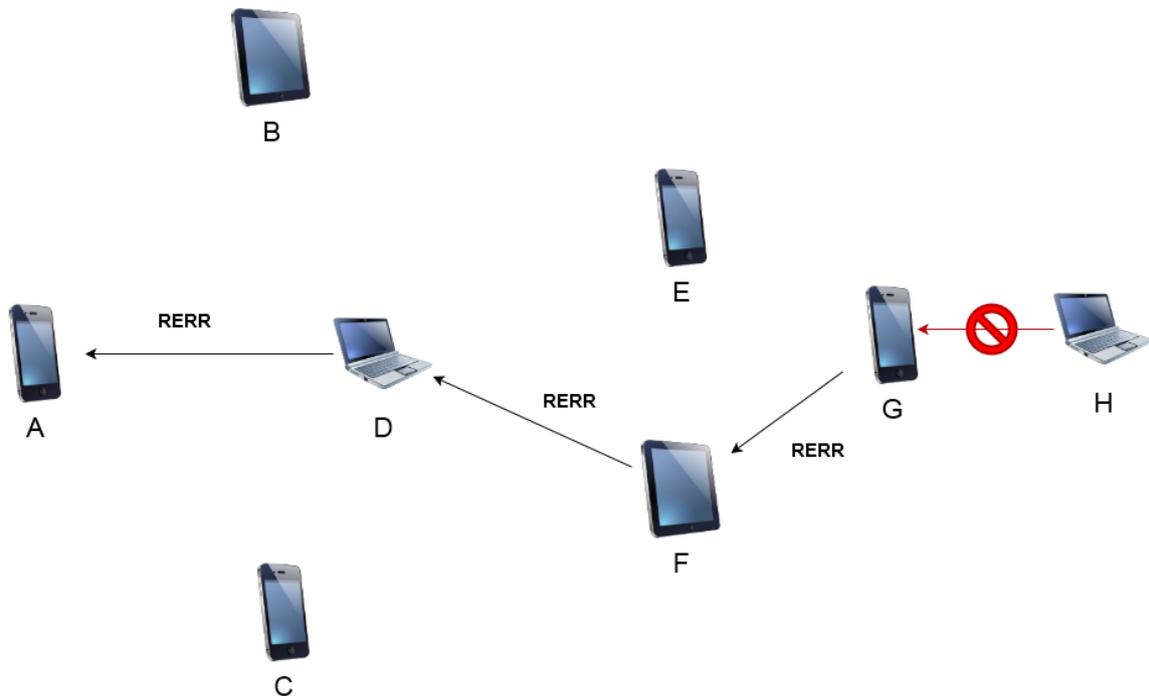


Figura 2.6 Cuando se rompe el enlace entre A y H, el nodo intermedio G avisa al origen A que ha ocurrido un fallo en el enlace. G es el que genera el mensaje RERR y a su vez los nodos intermedios reenvían el mensaje RERR.

Protocolo	Tipo	Distribuido	Uso de energía	Uso de ancho de banda
AODV	Reactivo	Si	Bajo	Bajo
DSDV	Proactivo	Si	Alto	Alto
DSR	Reactivo	Si	Bajo	Bajo
TORA	Reactivo	Si	Bajo	Bajo
OLSR	Proactivo	Si	Alto	Alto
ZRP	Híbrido	Si	Medio	Medio
ZHLS	Híbrido	Si	Medio	Medio

Tabla 2.1 Resumen de los protocolos de enrutamiento

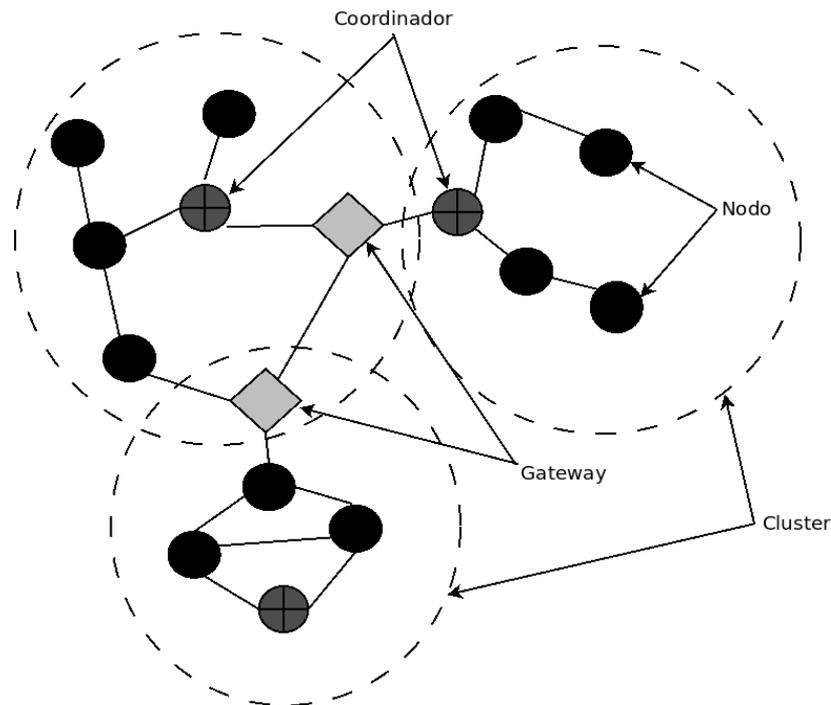


Figura 2.7 Ejemplo de una red dividida en clusters

2.2. Clustering en MANET

El proceso mediante el cual se divide una red en subestructuras interconectadas se denomina agrupamiento o clustering y las subestructuras resultantes se las denomina grupos lógicos o clusters. En estos escenarios los nodos de la red son divididos y ubicados dentro de grupos lógicos. Una estructura típica de clusters es la que se muestra en la Figura 2.7.

Los nodos normalmente en este tipo de esquemas asumen roles diferentes, como cabeza o coordinador, gateway o simplemente miembro del cluster. Los coordinadores mantienen y administran toda la actividad dentro del cluster al que pertenecen. Un nodo gateway se encarga de toda la comunicación entre clusters y existen esquemas donde este último tipo es omitido y por lo tanto no existe la diferenciación entre tráfico intra del inter cluster.

2.2.1. Ventajas del clustering

Una topología dividida en clusters mejora la performance de una MANET y los beneficios son enumerados a continuación:

- Reduce la información topológica debido a que como el número de nodos dentro de un cluster es menor al número total de nodos de la red, se simplifica el proceso de

incorporar nueva información de la topología. De esta manera cada nodo solo requiere almacenar una porción reducida de la información de ruteo de toda la red.

- Eficiencia en el ruteo, como en una arquitectura de red plana cada nodo tiene igual responsabilidad en el proceso de enrutamiento, una gran cantidad de paquetes inundan la red durante el proceso de descubrimientos de rutas. La división de la red en grupos lógicos mejora la eficiencia y hace que el descubrimiento de rutas sea más sencillo.
- Escalabilidad. Desde la perspectiva de un nodo que compone a una MANET la red aparenta ser mas pequeña y si un nodo se desconecta o cambia de grupo debido a la movilidad solamente los nodos restantes del grupo deben modificar sus estructuras de datos.

Existen otras ventajas resultantes del clustering como el mejor aprovechamiento del ancho de banda, el consumo de energía y el rendimiento también se mejoran.

2.2.2. Clasificación de los esquemas de clustering

Los esquemas de clustering se pueden clasificar de acuerdo a diferentes aproximaciones. Por ejemplo en esquemas donde nodos especiales son requeridos o en los que estos no lo son. Otro criterio podría ser tomando en cuenta la distancia entre pares de nodos, resultando en esquemas de clusters donde se agrupa a los nodos de un salto de distancia o esquemas de agrupamiento de múltiples saltos de distancia.

Dados los amplios tipos y propósitos de las soluciones de agrupamiento existentes, este estudio realiza una clasificación según sus principales objetivos. Siguiendo este criterio, los esquemas de agrupamiento en MANET se agrupan en cinco categorías, como se describe en la Tabla 2.2.

En el agrupamiento basado en el ID-Neighbour, un único identificador es asignado a cada nodo y cada nodo conoce el identificador de sus vecinos. El coordinador es seleccionado según un criterio que involucre a los identificadores, como por ejemplo el que posea el identificador más bajo o bien el más alto.

El agrupamiento basado en la conectividad considera la topología de red que circunda a los nodos. El grado de conectividad de un nodo es usualmente el criterio por el cual las decisiones de agrupamiento son tomadas.

El clustering según el contexto de movilidad toma en consideración los patrones de movilidad que exhiben los nodos de la red. De esta manera los nodos con patrones similares son agrupados juntos dentro del mismo grupo lógico.

Clasificación	Objetivos
Basados en el ID-Neighbour	Crea clusters y selecciona al coordinador mediante decisiones en base a los identificadores.
Basados en la conectividad	Considera el grado de conectividad de los nodos para la elección del líder del cluster.
Contexto de movilidad	Toma en cuenta los patrones de movilidad de los nodos para la creación del cluster.
Uso eficiente de la energía	Evitar o equilibrar el consumo de energía de los nodos móviles para extender la vida útil de la red.
Por ponderaciones combinadas	Considera varias métricas en la construcción del clúster y ajusta estas para diferentes escenarios de aplicación.

Tabla 2.2 Clasificación de esquemas de clustering

El clustering basado en uso eficiente de la energía se concentra especialmente en prolongar la vida útil de los dispositivos móviles en cuanto al uso de las baterías.

El agrupamiento por ponderaciones combinadas conjuga múltiples parámetros de las categorías anteriores. Una ventaja de este enfoque es la configuración individualizada de los pesos en la toma de decisiones, teniendo la posibilidad de ajustarse de forma adaptativa ante diferentes escenarios.

2.2.3. Clustering en protocolos de enrutamiento sobre MANETs

A diferencia de los protocolos de enrutamiento normales, los protocolos de enrutamiento basados en clústers crean o manejan en una topología de red jerárquica. Dichos protocolos realizan enrutamiento jerárquico entre clústers, típicamente eligiendo un líder de clúster y utilizan los nodos gateway para comunicación entre clusters.

A continuación se describen los protocolos de enrutamiento basados en agrupamientos lógicos más relevantes.

Protocolos de ruteo proactivos

En este apartado se describen los protocolos de enrutamiento proactivos basados en clustering más relevantes.

CGSR (*Clusterhead Gateway Switching*)

CGSR es una combinación entre el protocolo de enrutamiento DSDV con el algoritmo de agrupamiento LCC (*Least Cluster Change*) [2], CGSR agrupa los nodos de la red en clusters y cada uno de ellos contiene un líder [20] [62]. En LCC un coordinador es vuelto a elegir solamente cuando ocurre un cambio en la red que cause que existan dos *clusterheads* en un mismo grupo o que el *clusterhead* pierda la conectividad con el resto de los nodos del agrupamiento lógico. Esta elección se realiza gracias a la ejecución de un algoritmo distribuido.

En CGSR cuando un origen quiere realizar una transmisión, este envía los paquetes a su líder de cluster, el cual los reenvía a un gateway que conecta con el próximo líder de cluster de acuerdo a la ruta del destino. El gateway entrega los paquetes al próximo *clusterhead* y el proceso se repite hasta alcanzar el *clusterhead* del destino. Finalmente, este último transmite la información directamente al destino. Cada dispositivo envía periódicamente por broadcast su tabla de miembros a través del protocolo DSDV.

C-OLSR (*Cluster-based OLSR*)

C-OLSR [71] es una extensión del protocolo de enrutamiento OLSR para que funcione sobre una topología de red dividida en clusters. C-OLSR asume que la red ya se encuentra dividida en clusters y restringe la propagación de los mensajes de control de topología dentro de cada cluster. La generación de información de ruteo inter cluster está basada en MPRs (*Multipoint Relays*) a nivel de cluster y estos últimos son vistos como súper nodos. Los mensajes C-HELLO (*Cluster HELLO*) y de control de topología de cluster C-TC (*Cluster Topology Control*) son los encargados de mantener la información de ruteo entre los clusters.

Este protocolo puede ser configurado para que se ejecuten dos instancias simultáneamente de OLSR, una para el manejo de la información intra cluster, mientras que la segunda se encarga de la información inter cluster.

DefeR (*Deferred Routing*)

DefeR [63] provee una aproximación novedosa para el manejo eficiente del reenvío de mensaje en topologías de red que utilizan clusters. Arma un árbol binario de múltiples niveles de clusters virtuales en busca de reducir el impacto de la movilidad de los nodos. Mientras que los grupos lógicos reales son administrados por un algoritmo de agrupamiento externo, los virtuales están definidos por DefeR. Los agrupamientos reales siempre son representados por las hojas del árbol binario multinivel y los virtuales se representan vía los nodos que no son hojas. Esta estructura jerárquica minimiza el impacto de la movilidad de los nodos

e incrementa la escalabilidad en el ruteo. Si por ejemplo un dispositivo de un cluster real, clusterA, se mueve a uno hermano, la actualización de la información de las tablas de ruteo solamente será circunscripta a los dispositivos que componen el cluster virtual padre del clusterA. De esta manera se reduce drásticamente la sobrecarga en el ruteo.

Protocolos de ruteo reactivos

Este apartado esta destinado a la descripción de los protocolos de enrutamiento reactivos que utilizan clustering para su operación y que son de interés para esta investigación.

CLACR (Core Location-Aided Cluster-based Routing)

En CLACR [75] toda la red se divide en grupos cuadrados que no se superponen. Cada grupo está asignado a un *clusterhead* responsable del enrutamiento. Cuando la red se inicia y se particiona, cada nodo se asigna a sí mismo como *clusterhead* y envía un broadcast con un paquete HEAD INIT que anuncia su cluster ID y la ubicación obtenida mediante el GPS. Los nodos del mismo clúster, al recibir este paquete compara su ubicación con la del emisor. Si la ubicación del emisor está más cerca del centro geométrico del cluster, entonces se convierte en *clusterhead* del mismo.

CLACR es un protocolo de ruteo reactivo que utiliza el algoritmo de Dijkstra para el calculo de rutas, pero no proporciona enrutamiento de múltiples saltos, entonces para que funcione correctamente la red debe estar dividida en una gran cantidad de pequeños clusters.

CBRP (Cluster Based Routing Protocol)

CBRP [36] divide la red en grupos lógicos superpuestos o disjuntos. La comunicación directa entre lideres de grupos no esta permitida, en cambio esta tiene que hacerse a través de los gateways los cuales tienen a su alcance dos o más *clusterheads* como vecinos, todo esto cuando existe superposición de clusters o al menos tiene un líder de grupo y otro gateway, cuando los cluster son disjuntos. Como resultado de este proceso CBRP forma grupos con una distancia máxima de dos saltos.

CBRP utiliza el algoritmo LCC [2] y selecciona como cabeza de grupo al nodo con el ID más bajo. El descubrimiento de rutas se realiza inundando la red con mensajes RREQ. Estos mensajes son pasados entre los *clusterheads* y gateways, pero estos últimos no generan broadcasts; con ellos solo los retransmiten al siguiente *clusterhead*. Este proceso minimiza la cantidad de tráfico durante el descubrimiento de rutas.

OCRP (*On-Demand Clustering Routing Protocol*)

Para construir y mantener una topología jerárquica, los mecanismos de agrupamiento existentes consumen un cantidad significativa de comunicación. OCRP [38] introduce un paradigma novedoso en la forma en que se crean y son mantenidos los clusters. No existe fase de agrupamiento inicial, es decir, los clústeres se crean improvisadamente cuando los nodos tienen que enviar información.

Cuando un nodo origen inicia un broadcast para la búsqueda de ruta, cada nodo receptor está listo para ser *clusterhead*. El primer vecino que retransmite el paquete de broadcast se convierte en el líder del grupo y todos los demás nodos que reciben el primer reenvío se convierten en nodos ordinarios. Una vez que otro nodo, del mismo grupo, se convierte en *clusterhead* también se convierte en gateway y de esta manera se determinan los límites de los clústers. Al llegar los datos al destino, todos los nodos en la ruta pertenecen al mismo clúster.

ORC (*On-Demand Routing-based Clustering*)

De manera similar a OCRP, el esquema de ORC [37] crea la estructura de clusters y hace el descubrimiento de rutas bajo demanda. Sin embargo con la adición del mantenimiento del cluster. Mientras que en OCRP los clusters son creados solamente analizando los mensajes RREQ, ORC se basa en paquetes adicionales, cluster RREQ (C-RREQ), para crear la topología de clústers.

Un paquete C-RREQ es una variación del paquete RREQ y este es creado cuando un nodo recibe un paquete RREQ. Contiene información adicional, como el Cluster-ID (CID) y los IDs de los nodos vecinos. Esta información permite que ORC soportar la agrupación en clústeres multisalto con clusters superpuestos.

2.2.4. Resumen de los protocolos de ruteo que utilizan clustering

En la Tabla 2.3 se presentan todos los protocolos de enrutamiento que utilizan técnicas de agrupamiento lógico de nodos analizados en el apartado 2.2.3, junto con sus características más sobresalientes. Principalmente se especifica la clasificación según el criterio utilizado para conformar clusters, para más detalle referirse al apartado 2.2.2. También se indica si funciona con o sin líder de cluster, si admite superposición de clusters y por último si extiende a un protocolo de enrutamiento ya existente.

Protocolo	Tipo	Clusterhead	Superposición	Extiende
CGSR	ID-Neighbour	Si	Si	DSDV
C-OLSR	-	-	-	OLSR
DefeR	Virtual clusters	No	No	OLSR
CLACR	Basados en la ubicación	Si	No	-
CBRP	ID-Neighbour	Si	Si	-
OCRP	A demanda	Si	No	-
ORC	A demanda	Si	Si	-

Tabla 2.3 Resumen de los protocolos de enrutamiento que utilizan clustering

2.3. Diseño cross layer

En esta sección se explicará el diseño en capas y el cross layer comparándolos entre sí. Se presentarán las ventajas y desventajas de ambos enfoques para implementar soluciones enmarcadas en las MANET.

2.3.1. Comparación entre el diseño en capas y el cross layer

Históricamente las arquitecturas de red asumen que las funciones de comunicación están organizadas en capas y la entrega de paquetes son llevadas a cabo por protocolos que operan en distintas capas [28]. De esta manera las funcionalidades de la red en su conjunto son llevadas a cabo a través del modelo en capas. Así cada capa dentro de este modelo define un aspecto particular de la operatoria de la red del cual se encargara de resolver y proveer servicios a la capa inmediatamente superior. Otra característica de las capas dentro de este modelo es que son modulares y cada una de ellas implementa una tarea en especifica.

La arquitectura en capas prohíbe la comunicación directa entre capas no adyacentes, mientras que las comunicación entre las adyacentes funciona gracias a la utilización de interfaces preestablecidas y estandarizadas [80].

Alternativamente, existen protocolos que son diseñados explícitamente para ignorar la arquitectura en capas, permitiendo la interacción e intercambio de datos entre capas que no se encuentran dispuestas en forma adyacentes. Este tipo de diseño se utiliza para obtener mejores prestaciones cuando se construyen nuevas soluciones en particular con los nuevos protocolos de enrutamiento [8] [14].

En la Figura 2.8 se puede apreciar la arquitectura en capas y el diseño cross layer que permite la interacción e intercambio de datos entre capas no adyacentes, por ejemplo, en el diseño cross layer es posible que parámetros de la capa de aplicación puedan tener

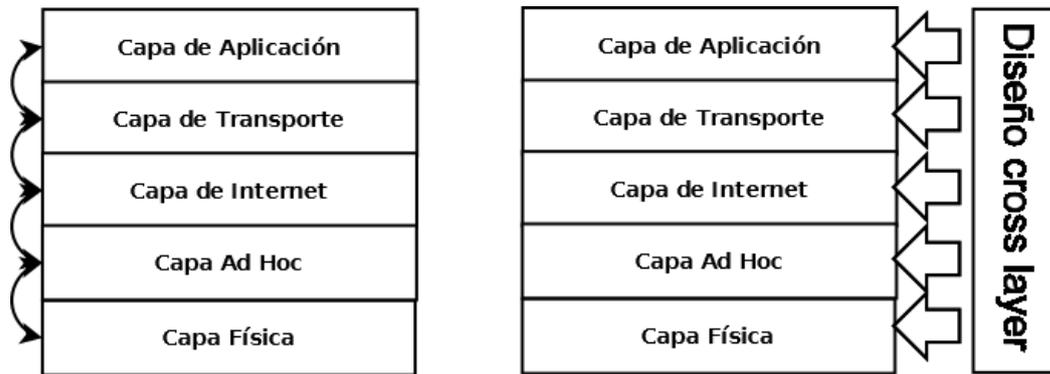


Figura 2.8 Arquitectura en capas y el diseño cross layer

comunicación directa con la capa de enlace. Bajo el enfoque cross layer, un capa puede adaptarse a los requerimientos y condiciones particulares de las otras capas.

2.3.2. Motivación del diseño cross layer

La motivación para adoptar un diseño cross layer en un protocolo que opere en una MANET se puede fundamentar principalmente en tres tópicos: Soportar la necesidad de que sea adaptable a las condiciones dinámicas de este tipo particular de red, satisfacer requisitos específicos del ámbito de aplicación y atender restricciones en cuanto al consumo de energía y niveles de seguridad que debe alcanzar la solución a desarrollar.

■ Adaptabilidad y auto-organización

Los protocolos de red que funcionan sobre MANETs deben ser adaptables a las condiciones volátiles de conectividad que este tipo de red presenta debido a la movilidad de los dispositivos que la conforman y todo este nivel de detalle debe ser transparente para las capas superiores. De este modo los nodos en estos sistemas dinámicos experimentan una serie de condiciones, dentro de las cuales se incluyen: cambios frecuentes en la topología, diferentes patrones en el tráfico y las distribuciones de los nodos, entre otras. La adaptabilidad puede ser implementada de la siguiente manera:

1. Conciencia del contexto o *context awareness*: Poseer conocimiento de los factores que afectan al estado de la red (condición del canal, congestión, demandas de tráfico, etc.).
2. Ajuste del protocolo: Tener la posibilidad de que cada protocolo ajuste su comportamiento de acuerdo al estado actual de la red.

Un ejemplo de esto son los protocolos de enrutamiento que toman decisiones durante su funcionamiento de acuerdo al estado del contexto, un parámetro comúnmente utilizado para ajustar su comportamiento son los niveles de energía presentes en los nodos componentes del sistema [31].

El *context awareness* en la mayoría de las ocasiones requiere practicar un rediseño de la manera en que los protocolos interactúan entre sí. Los diseños cross layers garantizan que los protocolos cooperen compartiendo información del estado de la red mientras se mantiene la separación en capas [22].

- Conservación del uso de la energía

El manejo eficiente de la energía es un factor determinante para que una solución que funciona sobre redes móviles ad hoc sea exitosa o no. Esto se debe a que los nodos que componen este tipo de redes dependen de fuentes de energía portátiles y limitadas. De hecho la conservación de la energía se convierte un un reto crucial y extremadamente difícil de lograr en entornos multi-hop, donde los nodos deben destinar parte de su energía para el enrutamiento de paquetes y garantizar de esta manera la conectividad de la red.

En la capa de red, por ejemplo, el proceso de selección de ruta debe ser realizado de tal manera que se reduzca el consumo de energía involucrado en el ruteo de extremo a extremo [12]. Para que esto sea posible el protocolo de enrutamiento debe tener acceso a la información de la energía involucrada en la retransmisión de paquetes y el nivel de batería del dispositivo.

- Seguridad

Los nodos en las MANET se comunican unos con otros vía un canal de broadcast abierto y compartido, por lo cual es extremadamente permeable a sufrir ataques a su seguridad . Además, la comunicación multi-hop implica que la red debe depender de soluciones individuales implementadas en cada nodo, resultando en que el entorno se vuelve vulnerable a infiltraciones, interferencias, ataques por DoS (*Denial Of Service*, Denegación de servicio), vulneraciones a la privacidad de los datos transmitidos, etc [29] .

Actualmente existen muchas investigaciones avocadas a proveer seguridad en el re- envío de datos en estos entornos abiertos y compartidos. Asegurando los protocolos de enrutamiento de modo de hacer frente a los ataques que intentan perturbar la información topológica [4]. También se puede mencionar a las soluciones que buscan asegurar el enlace de datos y que se implementan como parte de los estandar wireless

WEP (*Wired Equivalent Privacy*, Privacidad equivalente a cableada)/WPA (*Wireless Protected Access*, Acceso inalámbrico protegido) para la 802.11 [54] con el fin de proveer autenticación y privacidad en las redes wireless single-hop [3].

Todo lo antes expuesto tiene sentido para diseños cross layer de protocolos que trabajen en la capa MAC (*Media Access Control*, Control de acceso al medio), de transporte y enrutamiento que permitan tener en cuenta las cuestiones de seguridad en todas las etapas del diseño del protocolo.

- Requisitos específicos del ámbito de aplicación o QoS (*Quality Of Service*, Calidad de servicio)

QoS es una garantía de cierta performance en términos de los siguientes parámetros: el uso del ancho de banda, la demora de entrega de paquetes, el jitter o la tasa de entrega de paquete, etc. Generalmente, el diseño cross layer permite que las capas intercambien información de estado para obtener ganancias de rendimiento en términos de los parámetros antes enumerados [18].

2.3.3. Ventajas de utilizar una aproximación cross layer en los protocolos de enrutamiento

Actualmente se están utilizando técnicas que consisten en el intercambio de información entre las distintas capas para alcanzar nuevos niveles de optimización por este camino distinto al tradicional. El concepto de diseño cross layer promueve el intercambio de información en pos de obtener mejoras significativas en el rendimiento general de la red. En términos generales los diseños de este tipo involucran a la capa física, la de control de acceso al medio (MAC), la de red y de transporte. En la mayoría de los casos, las retroalimentaciones entre capas es utilizada para permitir el flujo de información del estado de la red desde las capas superiores a las inferiores y viceversa, mientras que la estructura tradicional en capas es conservada.

- Interacción cross layer entre la capa física y la de red: El impacto que tiene la capa física en los protocolos de enrutamiento es significativa, como por ejemplo cuando se calcula los caminos, utilizar un contador de saltos podría no ser una métrica óptima, en lugar de tomar en cuenta el estado actual del canal como la calidad del enlace [27].
- Interacción cross layer entre la capa física y la de transporte: El control de energía usualmente tiene influencia en las tasas de transmisión de los nodos móviles. La posibilidad de mejorar la comunicación multi-hop mediante el balanceo del control de

la energía en la capa física junto con el control de congestión mejoran notablemente el throughput y el consumo de energía en la red [21]. El modelo analítico para lograr esto indica que durante cada intervalo de transmisión de paquetes la potencia de transmisión del emisor aumentará proporcionalmente al retraso de cola de paquetes y disminuirá proporcionalmente a su nivel de energía.

- Interacción cross layer entre la capa MAC y la de transporte: La imposibilidad de TCP (*Transmission Control Protocol*, Protocolo de control de transmisión) de distinguir entre la pérdida de paquetes originada por congestión y la causada por otros factores, como por ejemplo la movilidad de los nodos, fluctuaciones en los enlaces inalámbricos, entre otros, es uno de los motivos para la pobre performance de TCP en ambientes MANETs. Una solución cross layer otorga mayor rendimiento mediante la recopilación de información de la capacidad del ancho de banda, así como la del retraso en la capa de enlace [50].

2.4. Resumen

En este capítulo se presentaron y abordaron las soluciones más relevantes en el área de los protocolos de enrutamiento que funcionan sobre MANET, teniéndose siempre en cuenta si se tratan de protocolos proactivos, reactivos o híbridos. Se analizó en detalle el protocolo reactivo AODV, por ser éste en el cual se basa la solución propuesta en este documento. Se describió los formatos de mensajes que empleados por AODV para su funcionamiento, junto a un ejemplo del funcionamiento de AODV.

La sección 2.2 trató sobre las distintas técnicas de clustering disponibles a ser aplicadas sobre MANET, así como los principales protocolos de enrutamiento que se utilizan actualmente. Se mencionan las ventajas de utilizar agrupamiento de nodos en redes móviles ad hoc y de esta manera se establece como la solución planteada en esta tesis doctoral, la cual se nutrirá de los beneficios antes mencionados, buscando el uso optimizado de recursos de manera escalable.

La última sección esta dedicada al diseño cross layer, debido a que la propuesta de solución que se plantea en este documento emplea este tipo de diseño para descubrir nuevos clientes en la capa de red y no en la capa de aplicación. Finalmente se detallan los beneficios de utilizar un diseño cross layer en el proceso de descubrimiento de nuevas rutas en MANET.

Capítulo 3

PROCCDIMI: PROtocolo para el Cálculo Científico Distribuido sobre clientes Móviles Indeterminados en MANETs

En el capítulo 2 se ha descrito las principales características de los protocolos de enrutamiento en MANET, se presentó los exponentes más apropiados al tema que estamos tratando y que aportan distintos enfoques de como solucionar los problemas relacionados con el ruteo de paquetes.

Se introdujo la idea de clustering, cuales son sus ventajas más interesantes y se listó las implementaciones más relevantes dentro de los protocolos de ruteo que explotan los beneficios asociados a la construcción de agrupamientos lógicos de nodos. El diseño cross layer también se trató en el capítulo anterior indicando en qué consiste el mismo y cuál es el criterio para adoptarlo en la construcción de nuevas soluciones en el área de enrutamiento; sin embargo, existen problemas importantes relacionados con la indeterminación de clientes en las MANET que aún están por resolverse.

En este capítulo se revisan los problemas latentes en los protocolos que funcionan sobre este tipo de infraestructura y se aporta una visión de cómo resolver algunas dificultades para luego plantear una propuesta de solución novel.

Una vez delineado el planteo, se define en detalle las extensiones a los formatos de mensajes de AODV que utiliza el nuevo protocolo. Finalmente se describe la generación de los mensajes dentro de este, el mecanismo de administración de grupos y las consideraciones finales.

3.1. Problemas existentes en el funcionamiento de las MANET

En esta sección se enumeran los principales problemas que son más comúnmente encontrados al momento de trabajar en una MANET y se proponen distintas estrategias para arribar a soluciones aceptables.

- Limitaciones de hardware.- Por lo general los nodos que participan en las redes móviles ad hoc son unidades limitadas, ligeras, reducidas en capacidades y pobres en recursos. De modo que, un factor a considerar es la limitación de recursos de los dispositivos móviles. Muchas de las soluciones que funcionan en este tipo de redes no consideran las limitaciones de los dispositivos tanto de batería, memoria, comunicación y computo. Por consiguiente, se deben diseñar soluciones que se adapten a las restricciones de los nodos inalámbricos y reduzcan el consumo de batería, por nombrar un recurso limitado. Estos diseños deben ir orientados hacia soluciones que operen con las restricciones típicas de hardware de los dispositivos que funcionan dentro de este tipo de redes.
- Coordinación centralizada.- En algunos de los protocolos existentes, un dispositivo funciona como coordinador central que se encarga de registrar toda la información acerca de los nodos que constituyen y así poder tomar las decisiones adecuadas y dar las ordenes necesarias para que los clientes móviles puedan operar dentro de la red, generando así una fuerte dependencia hacia el coordinador de la red. Para eliminar esta dependencia, se requiere un esquema de diseño descentralizado o un sistema alternativo que garantice en todo momento la disponibilidad de la información del estado de la red y un mecanismo para la toma de decisiones, y de esta manera ser tolerante a fallos eliminando de este modo el único punto de falla. Con el uso en cada dispositivo de una memoria cache local se logra obtener un esquema distribuido, descentralizado y autónomo en donde cada nodo de la red dispone de la información que antes solo manejaba el coordinador.
- Tormenta de broadcast.- Las aproximaciones que prefieren un abordaje descentralizado de la información sobre los clientes en la red, suelen generar un exceso de retransmisiones durante su operatoria. Este exceso de transmisiones origina un alto consumo de batería y la energía es un recurso muy crítico en la mayoría de los dispositivos limitados. Este tipo de soluciones consumen demasiados recursos, principalmente ancho de banda; envían una considerable cantidad de mensajes, ya sea a través de broadcast, multicast o unicast, dando lugar al problema conocido como tormenta de

broadcast. Problema originado en parte debido a la movilidad de los dispositivos y a las características inherentes al medio y al entorno. Es conveniente encontrar un equilibrio entre el envío de novedades y peticiones, la periodicidad de los envíos de mensajes y el tiempo de vida de las novedades.

- Escalabilidad.- La escalabilidad puede convertirse en un problema cuando se pretende escalar una red y el rendimiento se ve seriamente comprometido. Algunas soluciones definen coordinadores, que centralizan la asignación de las tareas que deben realizar los clientes dentro de un ámbito de cómputo distribuido. En MANETs ésta práctica no es recomendable, debido a que son redes altamente dinámicas y en algunos casos, los dispositivos miembros de este tipo de redes tienen capacidades de almacenamiento muy limitadas, por lo que es muy difícil y costoso que uno de estos dispositivos asuma el rol de coordinador central.
- Uso de caches.- Una estrategia altamente utilizada para minimizar el número de retransmisiones es el empleo de memorias caches. De esta forma cada dispositivo almacena localmente en ellas la información que necesita para operar, de modo que cuando se busca algún tipo de información, antes de realizar una transmisión sobre la red, el dispositivo verifica dentro de la cache y si encuentra la información que necesita, entonces no se transmite ningún mensaje. En ocasiones el problema que conlleva el uso de caches, consiste en que algunas de las entradas almacenadas se correspondan con información desactualizada, es decir, que ya no reflejen la realidad en la red. Este problema se agudiza en redes altamente dinámicas, para ello se deben implementar mecanismos de consistencia de caches que no aumenten en exceso el tráfico que esto pueda llegar a generar. Esto último comúnmente se resuelve con el uso de números de secuencia y contadores de tiempo de vida o *lifetime*, que indican el período de tiempo que se debe esperar antes de solicitar una actualización de la información almacenada en la cache.
- Heterogeneidad.- Por lo general los elementos participantes en las redes móviles ad hoc son dispositivos reducidos en capacidades como ya se indicó, o dispositivos móviles con una mayor disponibilidad de recursos como por ejemplo las notebooks. Esto genera un cierto grado de heterogeneidad, el cual se encuentra presente en todas las redes de este tipo. Las principales diferencias que ameritan mencionarse son: en el hardware que poseen, en que protocolos de comunicaciones utilizan y en que sistemas operativos tienen instalados.

3.2. Soluciones a los problemas en los protocolos de red que funcionan sobre MANETs

A continuación se enuncian distintas alternativas que se plantean para solucionar de manera dinámica los problemas planteados anteriormente. Estas soluciones centran sus esfuerzos en minimizar el uso de ancho de banda, en reducir el consumo de recursos (energía, memoria, cómputo), en disminuir el uso excesivo de retransmisiones y en la eliminación de la idea de la centralización de información.

- Intermediadores.- Este esquema de solución consiste en la existencia de nodos que desempeñan el rol de intermediarios entre clientes y coordinadores. De esta manera los coordinadores descubren la existencia de los nuevos clientes a través de los intermediarios y lo mismo sucede cuando un cliente desea identificar un coordinador en la red. Sin estos intermediarios los clientes no son capaces de descubrir a los coordinadores existentes en la red y viceversa. En primera instancia, tanto los coordinadores como los clientes se registran con uno o varios intermediarios. Si tomamos por ejemplo a los clientes, estos solicitan a uno o varios intermediarios la búsqueda del coordinador más cercano. Los intermediarios buscan en sus registros, si alguno se corresponde con lo solicitado, luego el intermediario devuelve al cliente la información correspondiente. En esta solución tanto clientes como coordinadores descubren dinámicamente a los intermediarios.
- El descubrimiento pasivo o *push*.- En esta solución los coordinadores del cómputo distribuido se anuncian periódicamente. Los clientes por su parte, almacenan estos anuncios en una memoria cache local, para que posteriormente cuando un cliente necesite realizar una solicitud, sea su cache local el primer sitio en el que compruebe si se tiene que comunicar con un coordinador anunciado previamente. Con este método de descubrimiento se mantiene información en casi todo momento sobre la presencia en la red de los coordinadores; pero cada nodo recibe y procesa una enorme cantidad de mensajes en forma de *overhead* en un momento determinado, ocasionando un alto consumo de energía, memoria y ancho de banda. En el modo *push* los coordinadores envían los anuncios de acuerdo a un tiempo de vida que previamente informan al cliente, cuando el cliente no recibe un anuncio dentro del tiempo previsto, el anuncio es eliminado de la caché local. Durante el envío de dos anuncios consecutivos del coordinador, podría ocurrir la caída del propio coordinador, a lo que el cliente no la detectará hasta el próximo anuncio. Una solución a esto último, es anunciarse

con mayor frecuencia, conllevando un mayor consumo de ancho de banda con el consecuente problema en la escalabilidad que esto trae aparejado.

- En el método de descubrimiento activo (denominado *pull*), un cliente que demanda un coordinador difunde mensajes (por broadcast o bien por multicast) a los nodos de la red en busca de respuesta a su petición y responden por unicast los coordinadores cercanos. Este método tiene problemas de escalabilidad en redes con gran número de nodos debido a un alto consumo de energía, memoria y sobre todo, de ancho de banda. El método de descubrimiento *pull* introduce cachés de coordinadores en los clientes, así, un cliente almacena localmente los coordinadores que va descubriendo. De tal forma que cuando un cliente necesite contactarse con un coordinador, antes de enviar un mensaje, primero comprueba en su caché si tiene almacenada información que necesita para comunicarse con el coordinador. Los coordinadores se anuncian incluyendo un número de secuencia y un tiempo de vida asociado al anuncio, que sirve para indicar al cliente el tiempo máximo que la información puede estar almacenado en la caché. Lo anterior es para evitar la incertidumbre por saber si el coordinador almacenado en la caché continúa estando disponible en la red.

Es cardinal gestionar adecuadamente los recursos de los dispositivos móviles, por eso existe la inquietud de lograr una óptima gestión en el envío mensajes. Si bien no se puede desatender las actualizaciones periódicas de las cachés, tampoco se puede abusar del uso descontrolado del envío de mensajes. A continuación se proponen algunas mejoras en el funcionamiento de los métodos anteriores:

- Que el método centralizado forme clústers con el objetivo de aumentar la escalabilidad, aun cuando se sabe que el punto débil de este método es la dependencia a un único punto de fallo, puesto que la caída de este hace imposible que los clientes puedan descubrir coordinadores y a los coordinadores a los nuevos clientes. Para evitar la situación anterior, es recomendable la presencia de uno o varios líderes en cada clúster.
- Que el método *push* no inunde la red de mensajes y que no permita transcurrir demasiado tiempo sin refrescar las cachés; para ello debe implementar una agrupación o segmentación de nodos atendiendo criterios geográficos o algún otro criterio más apropiado de acuerdo a la situación.
- Que el método *pull* solicite coordinadores solamente a los nodos integrantes de grupos específicos o realice búsquedas más selectivas en relación a la información registrada en su caché, o bien, si existen grupos o zonas, enviar peticiones a los nodos pertenecientes a la zona o grupo.

3.3. Descripción de PROCCDIMI

Las MANETs debido a su naturaleza intrínseca lidian con un alto grado de heterogeneidad, conllevando a manejar distintos equipos con capacidades variadas en cuanto a autonomía energética, memoria, potencia de cómputo, etc. Dentro de estos dispositivos heterogéneos, se encuentran algunos cuantos que poseen capacidades superiores, DCS (Dispositivo de capacidades superiores), como por ejemplo las notebooks o algunas tablets con prestaciones similares a PCs de escritorio [81].

En este documento se propone una arquitectura de cómputo distribuido para redes móviles ad hoc, donde se parte de la premisa de la existencia de ciertos dispositivos con capacidades superiores que participan de la MANET que contendrán algunos de ellos una cola de pedidos de procesamiento, los coordinadores, donde cada pedido de procesamiento se resolverá en forma conjunta y colaborativamente. Para esto, se definirán grupos lógicos o clusters de nodos dentro de los que se circunscribirá la asignación de tareas para la resolución de cada pedido de procesamiento. Cada solicitud deberá declarar a priori el tipo de problema que se debe resolver de las posibles opciones que ofrece el sistema y los conjuntos de datos a ser utilizados; tanto los posibles procedimientos computacionales como los datos a ser seleccionados por un eventual pedido, se encuentran almacenados en cada nodo de la red o al menos estos tienen indicado a donde ir a buscarlos.

El modelo aquí propuesto descansa sobre la idea de no tener un coordinador centralizado, para optar por organizarse dinámicamente en clústers agrupando los dispositivos alrededor de un coordinador o líder de grupo a fin de proveer los recursos necesarios para que cada pedido de las distintas colas pueda ser atendido.

Las agrupaciones de dispositivos son dinámicamente creadas, desagrupadas y vueltas a crear según se necesite. Las componentes de estos grupos son aquellas unidades de red a las cuales el coordinador del cluster las enroló primero, pudiéndose pensar que el criterio subyacente de conformación de grupos es por cercanía geográfica, o expresado de otra manera, los nodos que se ubican lo más próximos unos de otros pertenecerán al mismo cluster, aunque esto no es necesariamente cierto.

Si la topología de red y la carga de trabajo de los nodos se mantienen invariantes al momento de haber sido incorporados al cluster y como la política de enrolamiento es “el primero que responde es mi coordinador” se obtendría una reducción de los costos asociados a la latencia y la comunicación. Esta política de conformación de grupos de dispositivos en la MANET no implica de ningún modo que todos los nodos se encuentren a un *single-hop* de distancia unos de otros, con lo cual podría presentarse el escenario donde paquetes generados

en un origen deban atravesar varios nodos intermedios hasta alcanzar el destino en el afán de establecer comunicación entre ambos extremos.

El diseño propuesto combina el enfoque basado en clústers con uno de tipo cross layer. Esta combinación tiene como propósito involucrar el descubrimiento de nuevos clientes o agentes en el proceso de encaminamiento, y como ya se explicó, los clientes o agentes junto al coordinador son los que dan el soporte al cómputo distribuido colaborativo.

El protocolo propuesto se denomina PROCCDIMI (PROtocolo para el Cálculo Científico Distribuido sobre clientes Móviles Indeterminados en MANETs), el cual establece la creación de clústers dinámica y distribuidamente, donde el envío de mensajes relacionados con el cómputo únicamente se circunscribe al clúster. Esta última cualidad tiene por objetivo el de optimizar el consumo de energía, memoria, potencia de cómputo y ancho de banda, gracias al uso eficiente de recursos.

PROCCDIMI implementa un balanceo de cargas dinámico basándose en multi-agentes y se fundamenta en los protocolos de enrutamiento de vector distancia diseñados para las MANETs, particularmente en AODV [66]. También se contempla la heterogeneidad existente en este tipo de redes y es por ello que se incorpora la idea en esta propuesta de aprovechar la existencia de algunos dispositivos que poseen capacidades superiores (en cómputo, en comunicación ó almacenamiento de energía) con el objetivo de ser elegidos como los líderes de los clústers, o en este caso como coordinadores del cómputo distribuido, a fin de llevar a cabo la conformación del grupo, la asignación de tareas dentro del mismo y realizar el proceso de descubrimiento de nuevos clientes. Por ello es necesario distinguir entre dispositivos con capacidades superiores y dispositivos con capacidades reducidas. Una vez hecha la distinción, es posible una clasificación en base a sus capacidades.

Las capacidades de los dispositivos se obtienen a partir del hardware que ellos mismos poseen, es decir, inicialmente un dispositivo conoce sus propias características respecto a la cantidad de memoria, energía, capacidad de cómputo y potencia de señal; por lo tanto, la capacidad de un dispositivo se asigna en función del tipo de dispositivo que se trata.

El criterio para la elección de coordinador del clúster se basa en la selección de los dispositivos con capacidades superiores, en este caso la capacidad de un dispositivo se determina por medio de una estrategia sencilla que asigna un identificador binario a cada dispositivo; es decir, si se trata de un dispositivo con capacidades superiores es un DCS, de lo contrario es un DCL (Dispositivo de capacidades limitadas) [81].

Para definir la capacidad de un dispositivo se tiene en cuenta las siguientes suposiciones, cada dispositivo es consciente del grado de la capacidad que posee y este puede ser solo de dos tipos: alta (DCS) o limitada (DCL). En PROCCDIMI los nodos que son líderes de clústers son todos ellos DCS, no implicando que el recíproco sea siempre verdadero.

3.4. Formatos de paquetes utilizados por PROCCDIMI

Las extensiones que presentan los formatos de mensajes de PROCCDIMI usan el formato TLV (*Type-Length-Value*) de tipos de 8-bits que siguen los lineamientos que se presentan en el Internet-Draft de AODV (*Ad hoc On-demand Distance Vector*) [17]. Es decir, las extensiones de los campos deben definirse primero por un *Type* indicando a que tipo de datos pertenece el valor almacenado en el campo. Luego el *Length* que guarda la longitud de la extensión y por último *Value* que es el valor del campo.

3.4.1. Formatos de paquetes para el descubrimiento de clientes

A continuación se describen los formatos utilizados para construir los paquetes que se intercambiarán durante el proceso de descubrimiento de clientes y de enrutamiento que se realizan de forma simultanea.

Formato del paquete CREQ (*Coordinador Request*)

PROCCDIMI clasifica a los dispositivos de la red según si poseen capacidades superiores o limitadas de acuerdo al nivel de recursos con los que cuenta. Esta discriminación tiene por función la de proveer de una métrica para que el coordinador sepa con qué tipo de dispositivo cuenta a la hora de encarar un cómputo distribuido. De este modo, el coordinador tendrá un medio que le permitirá lograr un efectivo balanceo de cargas dentro del clúster. A tal efecto, PROCCDIMI añade un nuevo campo a los paquetes RREQ (*Route Request*) extendidos de AODV, el campo *Nodo Coeficiente*. Con lo cual a estos nuevos paquetes se los denominarán CREQ (*Coordinador Request*) a partir de ahora. El campo *Nodo Coeficiente* va acompañado de los correspondientes campos *Type* y *Length* siguiendo las recomendaciones del Internet-Draft de AODV .

El campo *Nodo Coeficiente* también se encuentra presente en cada entrada de la tabla de clientes que cada coordinador lleva localmente.

Otro campo que se agrega al RREQ (*Route Request*) de AODV (*Ad hoc On-demand Distance Vector*) es *Cluster ID Origen* también haciendo uso del formato TLV (*Type-Length-Value*). La función de *Cluster ID Origen* es una vez que se generó la respuesta para el emisor del RREQ (*Route Request*) con los datos del cluster al que pertenecerá a partir de la recepción de la misma, no se generen respuestas duplicadas o innecesarias. Este campo ayuda a mantener la sobrecarga de la red asociada al enrutamiento bajo control.

El formato del mensaje CREQ (*Coordinador Request*) que se observa en la Figura 3.1 tiene por propósito llevar una petición a fin de lograr que al nodo solicitante se le identifique

0						31	
Type	J	R	G	D	U	Reserved	Hop Count
RREQ ID							
Destination IP Address							
Destination Sequence Number							
Originator IP Address							
Originator Sequence Number							
NodoCoefType	NodoCoefLength			Nodo Coeficiente			
CIIDType	CIIDLength			Cluster ID Origen			

Figura 3.1 Formato del paquete CREQ

cual será su cluster de pertenencia junto con el coordinador que lo administra. Al mismo tiempo que mantiene su objetivo original que es el descubrimiento de rutas a los posibles destinos.

Un mensaje puede pasar por varios nodos intermedios antes de alcanzar el nodo destino, y es por ello que en el paquete CREQ se transportan datos de control útiles para la retransmisión de mensajes por parte de los nodos intermedios y para el envío de vuelta de la respuesta.

El formato de mensaje que se está presentando en este apartado contiene los siguientes campos cabecera: *Type* es el tipo de dato. *J* es la *Join flag*, reservado para multicast. *R* es la *Repair flag*, reservado para multicast. *G* es el *Gratuitous RREP flag*, indica si se debe enviar un *gratuitous RREP* en modo unicast al nodo indicado en el campo dirección IP destino. *D* es la bandera de destino, se indica solo cuando el destino puede responder el CREQ actual. *U* es un flag que indica que el número de secuencia es desconocido. *Reserved* se envía como 0, ignorado en recepción. *Hop Count* es el número de saltos desde la dirección IP origen al nodo que gestiona la petición. *RREQ ID* es un número de secuencia que identifica únicamente el CREQ cuando es tomado en conjunto con la dirección IP del nodo origen. *Destination IP Address* es la dirección IP del destino para la ruta deseada. *Destination Sequence Number* es el último número de secuencia recibido en el pasado por el origen para cualquier ruta hacia el destino. *Originator IP Address* es la dirección IP del nodo que origina el CREQ. *Originator Sequence Number* es el número actual de secuencia del origen. *NodoCoefType* es el tipo de dato y *NodoCoefLength* es la longitud de *Nodo Coeficiente*. El campo *Nodo Coeficiente* almacena si el emisor es:DCS o DCL. *CIIDType* es el tipo de dato y *CIIDLength*

0 31

Type	R	A	Reserved	Prefix Size	Hop Count
Destination IP Address					
Destination Sequence Number					
Originator IP Address					
Lifetime					
CIDType	CIDLength		Cluster ID		
IPCoordType	IPCoordLength		Coordinador IP Address		
			IPMCType	IPMCLength	
IPMulticastCluster					

Figura 3.2 Formato del paquete CREP

es la longitud de *Cluster ID Origen*. Finalmente, *Cluster ID Origen* es el que indica si al momento de recepción de este paquete un nodo anterior generó una respuesta con los datos de afiliación de un grupo lógico.

Formato del paquete CREP (*Coordinador Reply*)

El formato de mensaje CREP (*Coordinador Reply*) ilustrado en la Figura 3.2 tiene un propósito dual, en primer lugar comunicar la ruta hacia el destino del mensaje CREQ (*Coordinador Request*) que genera la respuesta encapsulada en el paquete CREP y en segundo termino llevar la información referente a la respuesta de la petición de coordinador. Este paquete puede ser generado por el coordinador del clúster, el destino del CREQ o un nodo intermedio que se ubique entre el origen y el destino que este en condiciones de responder una o las dos interrogantes planteadas por el mensaje CREQ que requieren respuesta. Esta decisión responde a mantener la sobrecarga de enrutamiento lo más reducida posible.

El paquete CREP contiene los siguientes campos de cabecera: *Type* es el tipo de dato. *R* es la *Repair flag*, usado para multicast. *A* cuando se requiere un mensaje de *Acknowledgment* o reconocimiento de peticiones y se activa cuando se quiere comprobar si un enlace es unidireccional. *Reserved* si se envía 0, se ignora cuándo es recibido. *Prefix Size* si no es cero, este prefijo especifica que el siguiente salto puede ser usado por cualquier nodo con el mismo prefijo de encaminamiento. *Hop Count* es el número de saltos desde la dirección IP del origen hasta la dirección IP del destino. *Destination IP Address* es la dirección IP

0		31
Type	N	Reserved
Unreachable Destination IP Address		
Unreachable Destination Sequence Number		
Additional Unreachable Destination IP Address		
Additional Unreachable Destination Sequence Number		

Figura 3.3 Formato del paquete RERR de AODV

del destino. *Destination Sequence Number* es el número de secuencia del destino asociado a la ruta. *Originator IP Address* es la dirección IP del nodo que ha originado el CREQ. *Lifetime* es el tiempo en milisegundos de los que los nodos reciben el CREP considerando que la ruta es válida. *CIIDType* es el tipo de dato y *CIIDLength* es la longitud de *Cluster ID*. *Cluster ID* es el campo que almacena el ID del cluster al cual pertenecerá el destino del CREP. Otros campos del paquete destinados a guardar información de clustering son: *IPCoordType*, *IPCoordLength* y *Coordinador IP Address* los cuales sirven para transportar la IP del coordinador del cluster al que se agregará el destino. *IPMCType* es el tipo de dato y *IPMCLength* es la longitud de *IPMulticastCluster*. Por último, *IPMulticastCluster* es el campo que maneja el coordinador para indicar cuál va a ser la dirección IP multicast que se utilizará para realizar anuncios globales dentro del cluster y así eliminar la necesidad de utilizar envío de mensajes de broadcast para realizar este tipo de comunicaciones.

Formato del paquete RERR (*Route Error*)

El formato de mensaje RERR (*Route Error*) mostrado en la Figura 3.3 es otro de los paquetes usados por PROCCDIMI, pero este se utiliza tal cual es propuesto por AODV, es decir, no se ha añadido ningún campo extra, por lo que se conserva y se utiliza igual a como se lo emplea AODV.

El formato de la cabecera de RERR tiene como propósito invalidar una ruta. El paquete RERR llega hasta el origen de la ruta con el fin de que se comience de nuevo el proceso de descubrimiento. El mensaje RERR se envía cuando algún enlace roto ocasiona que algunos destinos lleguen a ser inalcanzables por algunos de sus nodos vecinos. El paquete RERR contiene los siguientes campos de cabecera: *Type* es el tipo de dato. *N* es una bandera de no eliminación, se coloca cuando un nodo ha ejecutado una reparación local de un enlace y los nodos siguientes no deben eliminar la ruta. *Reserved* se envía 0, es ignorado en recepción.

Type	J	R	G	D	U	Reserved	Hop Count
RREQ ID							
Destination IP Address							
Destination Sequence Number							
Originator IP Address							
Originator Sequence Number							
TNType	TNLength			Task Number			

Figura 3.4 Formato del paquete CDREQ

DestCount es el número de destino inalcanzable incluido en el mensaje, debe tener como valor al menos un 1. *Unreachable Destination IP Address* es la dirección IP del destino que ha llegado a ser inalcanzable debido a un enlace roto. *Unreachable Destination Sequence Number* es el número de secuencia en la entrada de la tabla de rutas para el destino.

3.4.2. Formatos de mensajes del cómputo de distribuido

A continuación se describen los formatos utilizados para construir los paquetes que se intercambiarán durante el proceso de asignación de tareas a los clientes y la posterior recuperación de los resultados de las ejecuciones de las mismas.

Formato del paquete CDREQ (*Calculation Distributed Request*)

PROCCDIMI, al igual que en el caso de los CREQ (*Coordinador Request*), añade un campo a los mensajes RREQ (*Route Request*) de AODV, el campo *Task Number*, los cuales a partir de ahora se denominarán CDREQ (*Calculation Distributed Request*).

El campo *Task Number* es definido siguiendo el formato TLV (*Type-Length-Value*) de tipos de 8-bits y por lo tanto es acompañado por: *TNType* es el tipo de dato de *Task Number*, *TNLength* es la longitud de *Task Number* y *Task Number* es el campo para la identificación de cual tarea debe resolver el cliente.

La extensión *Task Number* cumple con la función de indicarles a los clientes cual será la tarea que deberán realizar dentro de un nuevo cómputo científico distribuido. El formato del mensaje CDREQ que se observa en la Figura 3.4 tiene como propósito llevar la petición

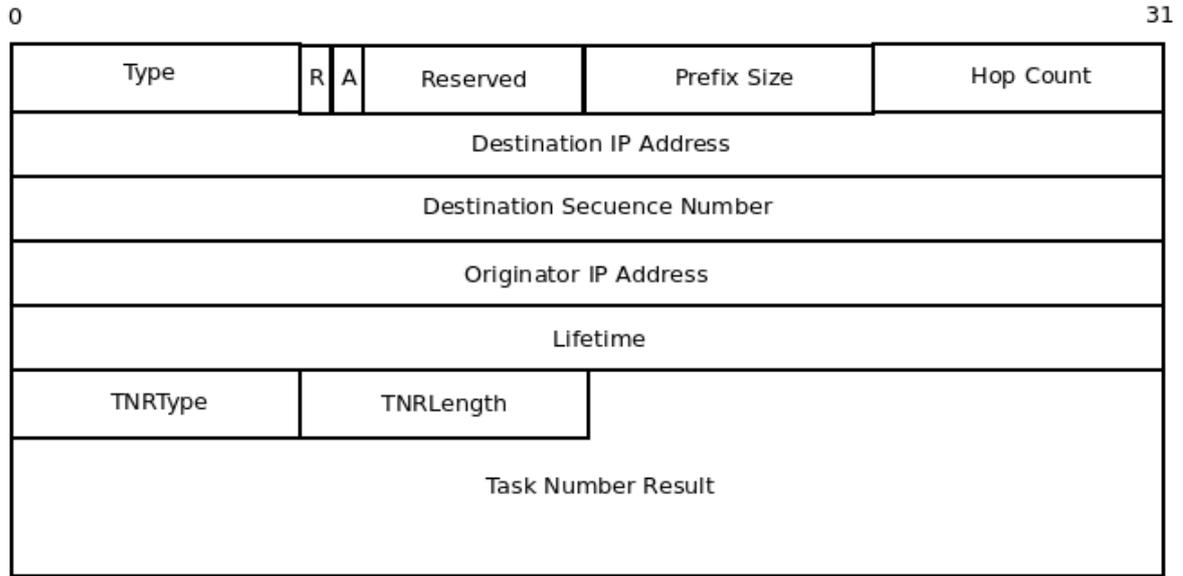


Figura 3.5 Formato del paquete CDREP

del coordinador a los clientes indicándoles que formarán parte de un nuevo procesamiento colaborativo y que parte les corresponde resolver de este cómputo mancomunado que se desarrollará dentro del grupo.

Formato del paquete CDREP (*Calculation Distributed Reply*)

El formato de mensaje CDREP (*Calculation Distributed Reply*) ilustrado en la Figura 3.5 tiene como propósito llevar el resultado del cálculo de la tarea encomendada al cliente en lo referente a la respuesta de la petición del coordinador CDREQ. Este paquete es generado por cada cliente una vez finalizada su ejecución y tiene como nodo destino el coordinador del clúster.

El paquete CDREP contiene los mismos campos de cabecera que el paquete RREP de AODV, con la incorporación del campo *TTNResult* que se utiliza para indicar el resultado del cálculo. También se adicionan los siguientes campos: *TNRType* es el tipo de dato de *TNResult*. *TNRLength* es la longitud de *TNResult*.

3.4.3. Resumen de los tipos de paquetes que utiliza PROCCDIMI

En la Tabla 3.1 se presentan todos los tipos de paquetes que utiliza el protocolo propuesto y que fueron analizados en el apartado 3.4.1 y en el apartado 3.4.2. En la tabla también se proporciona de una breve descripción de cuales son los propósitos de cada uno de los paquetes utilizados por PROCCDIMI.

Paquete	Funcionalidad
CREQ (<i>Coordinador Request</i>)	Petición de descubrimiento de ruta hacia un destino dentro de la red y simultáneamente solicita la adición del origen a un cluster.
CREP (<i>Coordinador Reply</i>)	Respuesta a una solicitud CREQ. Pudiendo responder una o las dos interrogantes planteadas por el mensaje CREQ
RERR (<i>Route Error</i>) de AODV	Paquete utilizado para invalidar una ruta.
CDREQ (<i>Calculation Distributed Request</i>)	Petición de cálculo científico distribuido a desarrollarse dentro del cluster. Esta solicitud contiene el identificador de la tarea que debe resolver el receptor de este tipo de paquete.
CDREP (<i>Calculation Distributed Reply</i>)	Respuesta a una solicitud CDREQ que contiene el resultado del cálculo científico distribuido.

Tabla 3.1 Resumen de los tipos de paquetes que utiliza PROCCDIMI

3.5. Tabla de clientes

En PROCCDIMI los dispositivos de red que cumplen el rol de coordinador utilizan, mantienen y actualizan una tabla de clientes o agentes. La tabla de clientes esta fusionada con la tabla de enrutamiento, expresado de otra manera, existe una sola estructura de datos en donde se guardan tanto la información asociada al ruteo como la que se utiliza cuando es necesario realizar un balanceo de cargas; aunque esta estructura de datos se encuentra presentes en todos los dispositivos de la red que utilizan PROCCDIMI, los únicos que hacen uso de la extensión del registro de clientes son los coordinadores de cluster.

La tabla de ruteo extendida contendrá las rutas calculadas a demanda hacia otros dispositivos dentro de la MANET como a que cluster pertenecen y de que dispositivo se trata, DCS, DCL o coordinador de cluster. Como se comentó anteriormente los únicos nodos que utilizarán la información de clustering y de capacidad del nodo son los lideres de grupo, que cuando necesiten saber con cuantos clientes cuenta dentro de su cluster realizará una consulta sobre la tabla de clientes colectando las entradas de la misma que posean el mismo cluster ID que el del coordinador que esta realizando la consulta. El resultado de la consulta anterior, variará significativamente de acuerdo al momento en que se realiza, esto se debe a que la información que se guarda en esta tabla es volátil y debe ser actualizada constantemente debiéndose principalmente a la movilidad de los nodos. Una vez que el coordinador cuenta con la información de cuantos clientes existen actualmente dentro del cluster que lidera, a partir de ese momento esta en condiciones de poder realizar el balanceo de carga, para de este modo, calcular distribuidamente la computación científica entre los clientes del cluster.

Ip Address	Nodo Coeficiente	Cluster ID	Task Assign Number	Lifetime
Ip1	Coordinador	2	-	-
Ip2	DCL	1	1	180000
Ip3	DCS	1	2	90000
Ip4	DCL	1	3	360000
Ip5	DCS	1	-	-
Ip6	DCL	3	-	-

Tabla 3.2 Ejemplo de tabla de clientes para el coordinador del cluster 1

En el Tabla 3.2 se muestra, a modo de ejemplo, la información que podría contener la tabla de clientes de un hipotético coordinador que contiene el ID de cluster igual a 1. Solamente los campos de la tabla de enrutamiento extendida que corresponde a la información de los clientes son mostrados en el ejemplo, omitiéndose el resto.

Como se mencionó, PROCCDIMI añade el campo *Nodo Coeficiente* a los formatos de mensajes de AODV y por tanto también se añade el mismo campo a la tabla de clientes para realizar el balanceo de carga más adecuado por parte de los coordinadores de cluster.

Cada entrada de la estructura de datos que se esta analizando en este apartado está compuesta por campos útiles para el desarrollo de una computación distribuida y un balanceo de carga adecuado con los clientes con que se cuentan activos en el agrupamiento lógico al momento de la consulta.

Las entradas de la tabla de clientes son: Dirección IP (IP Address), Coeficiente del nodo o *Nodo Coeficiente* que define las prestaciones del nodo, Identificador de cluster o *Cluster ID* es un valor numérico para identificar a que grupo lógico pertenece un cliente, el *Task Assign Number* que sirve para identificar que tarea tiene asignada un cliente cuando exista un contexto de la ejecución de computo científico distribuido entre los clientes del cluster y el *lifetime* indica durante que período de tiempo se debe esperar el mensaje CDREP por parte del cliente con la respuesta de su parte del computo; manejar un tiempo de expiración es necesario debido a la volatilidad de la permanencia de los nodos en este tipo de redes.

3.6. Generación de los mensajes CREQ, CREP y RERR

Cuando un nuevo dispositivo ingresa a una MANET que funciona sobre PROCCDIMI y tiene intenciones de formar parte de un cluster abocado al cómputo distribuido, con el primer descubrimiento de ruta comienza el proceso de filiación a un cluster. Tanto el pedido

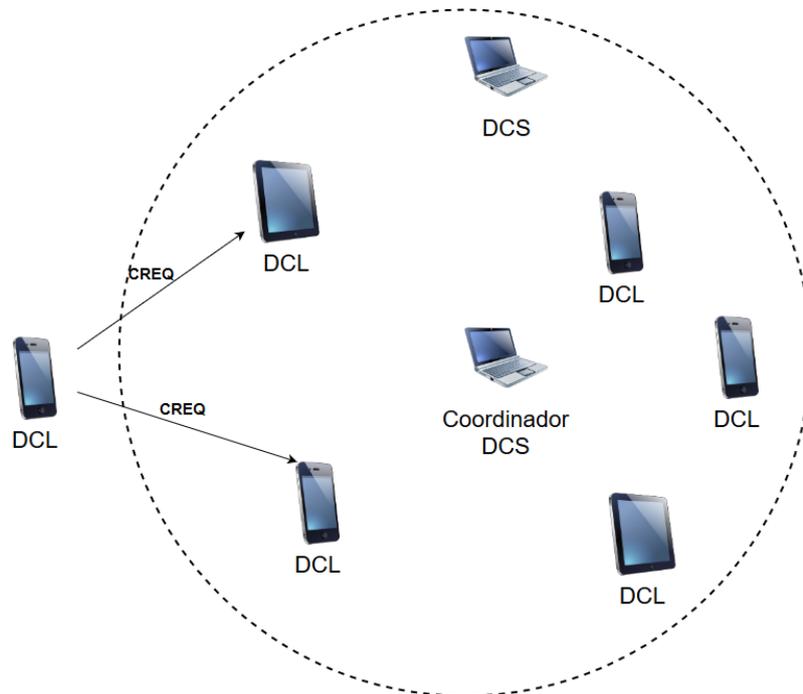


Figura 3.6 Envío broadcast de CREQ

de descubrimiento de ruta como el ingreso de un nodo a un cluster inicia con una solicitud CREQ (*Coordinador Request*) y ambos pedidos se resuelven en forma simultánea.

El paquete CREQ posee un número de secuencia o campo *RREQ ID* que se incrementa cada vez que se genera un nuevo CREQ. El número de secuencia es utilizado para los nodos intermedios para decidir si reenvían el mensaje o lo descartan si es que ya lo han retransmitido anteriormente.

El mensaje CREQ se envía por broadcast al vecindario dentro del clúster, como muestra la Figura 3.6. El mensaje CREQ lleva la dirección IP del nodo origen, junto con el campo *Nodo Coeficiente* que como se explicó, tiene por objetivo proveerle al coordinador de una métrica para que este pueda realizar un balance de cargas efectivo.

En el momento que un nodo intermedio reenvía un CREQ, añade una ruta inversa en su tabla, esta ruta referencia hacia el nodo origen emisor del CREQ.

Mientras el mensaje CREQ es recibido por nodos que no tienen una de las respuestas para las que fue generado, el paquete solamente es reenviado conforme al proceso descrito en las líneas superiores.

A continuación se analizarán los casos en que un dispositivo de la red esta en condiciones de generar un mensaje CREP al origen:

- El nodo receptor es coordinador, la solicitud de descubrimiento de ruta es para alcanzarlo a él y el campo *Cluster ID Origen* del CREQ no tiene un valor válido. Cuando

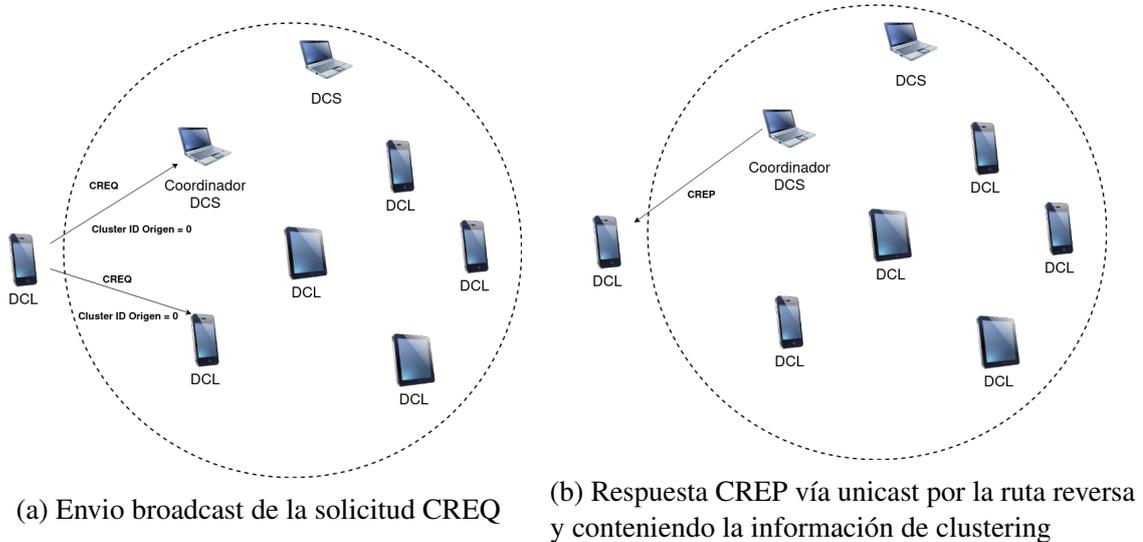


Figura 3.7 Respuesta CREP cuando el receptor es coordinador, es el destino del descubrimiento de ruta y no se ha respondido previamente con los datos de filiación a un cluster.

se dan estas condiciones el receptor responde las dos interrogantes para las cuales se creó el paquete CREQ devolviendo un mensaje CREP al origen que transita la ruta inversa por donde le ha llegado el mensaje CREQ. Además en los campos *Cluster ID* y *Coordinador IP Address* se completan con el identificador del cluster del coordinador que recibió el paquete y la dirección IP del mismo respectivamente. El envío del CREP toma lugar posteriormente a que el coordinador receptor actualice su tabla de clientes agregando al nuevo integrante del cluster, el nodo emisor del CREQ. En la Figura 3.7 se grafica el escenario antes planteado.

- El nodo receptor puede o no ser coordinador, la solicitud de descubrimiento de ruta es para alcanzarlo a él y el campo *Cluster ID Origen* del CREQ tiene un cluster ID válido. Lo que significa que el receptor solo debe responder por la información de ruteo solicitada. Ya que lo correspondiente al clustering ya fue resuelto por un nodo que recibió el CREQ en un estadio anterior. De este modo el destinatario devuelve un mensaje CREP al origen, a través de la ruta inversa por donde le ha llegado el mensaje CREQ. La Figura 3.8 muestra un ejemplo de lo descrito anteriormente.
- El nodo receptor no es coordinador, la solicitud de descubrimiento de ruta es para alcanzarlo a él y el campo *Cluster ID Origen* del CREQ no tiene un cluster ID válido. Bajo estas condiciones el receptor debe:
 1. Responder por la información de ruteo solicitada del mismo modo que se explicó anteriormente.

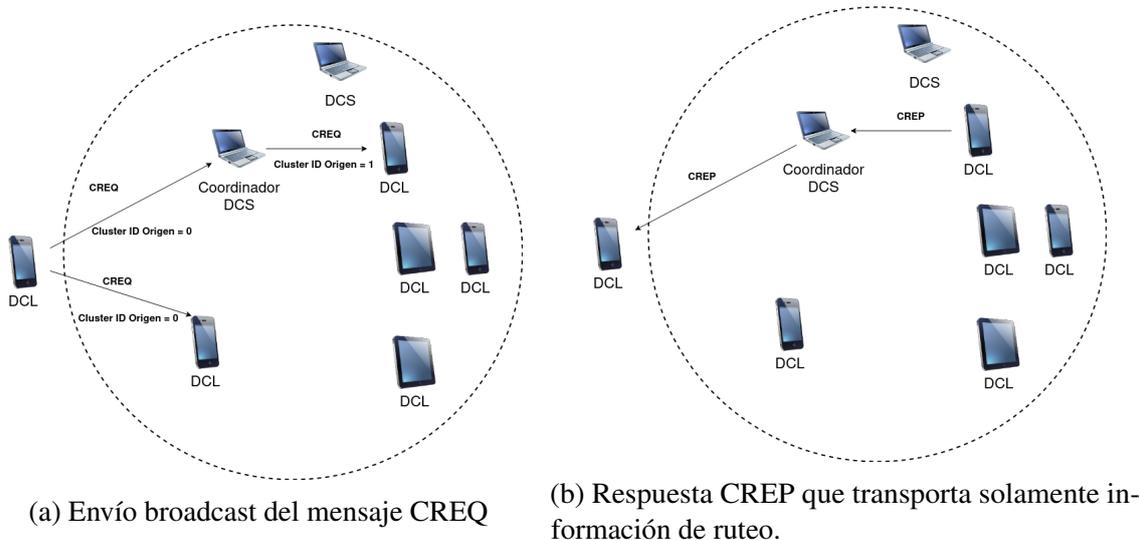


Figura 3.8 Respuesta CREP cuando el receptor es el destino del descubrimiento de ruta y la solicitud de clustering ya fue contestada

2. Si el nodo receptor pertenece a un cluster entonces junto con la información de ruteo se completan los campos *Cluster ID* y *Coordinador IP Address* del paquete CREP, para luego reenviar el paquete CREQ con el campo *Cluster ID Origen* asignado al cluster ID del nodo receptor como se muestra en la Figura 3.9. Esto último tiene por objetivo que el coordinador de cluster del receptor actualice su tabla de clientes con los datos del nodo emisor.

Si el receptor no pertenece a ningún cluster solamente responde con un CREP con la información de ruteo.

- El receptor es un coordinador intermedio y el campo *Cluster ID Origen* del CREQ no tiene un cluster ID válido. Bajo las condiciones anteriores, el coordinador receptor ingresa al emisor del CREQ a su tabla de cliente y genera un mensaje CREP conteniendo la información de clustering. Luego el nodo actualiza el campo *Cluster ID Origen* del CREQ con su cluster ID y finalmente lo reenvía como se puede ver en la Figura 3.10.

Otro escenario de generación de paquetes distinto a los antes enumerados, se produce cuando el receptor es un nodo coordinador intermedio y el campo *Cluster ID Origen* del CREQ tiene un cluster ID válido. En este caso, como se muestra en la Figura 3.11 no se genera ninguna respuesta CREP y se procede a verificar que el cluster ID sea el mismo que del coordinador receptor. Si esto último ocurre entonces el coordinador actualiza la tabla de clientes agregando al nodo emisor del CREQ. Si resulta que el cluster ID no coincide con el del receptor entonces el paquete es reenviado.

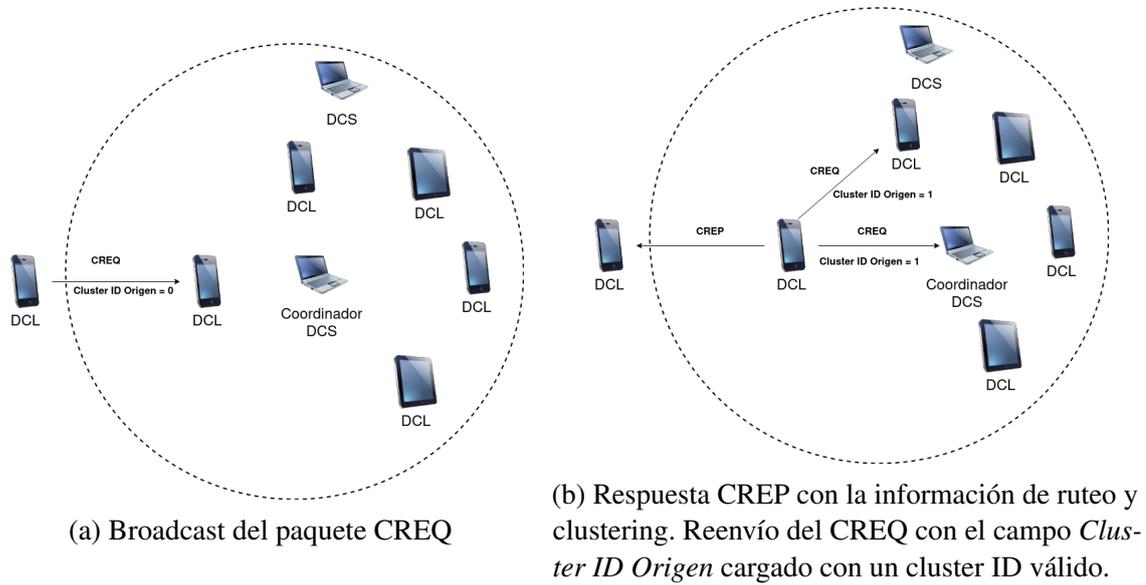


Figura 3.9 Respuesta CREP cuando el receptor no es coordinador, es el destino del descubrimiento de ruta y la solicitud de clustering no fue contestada aún.

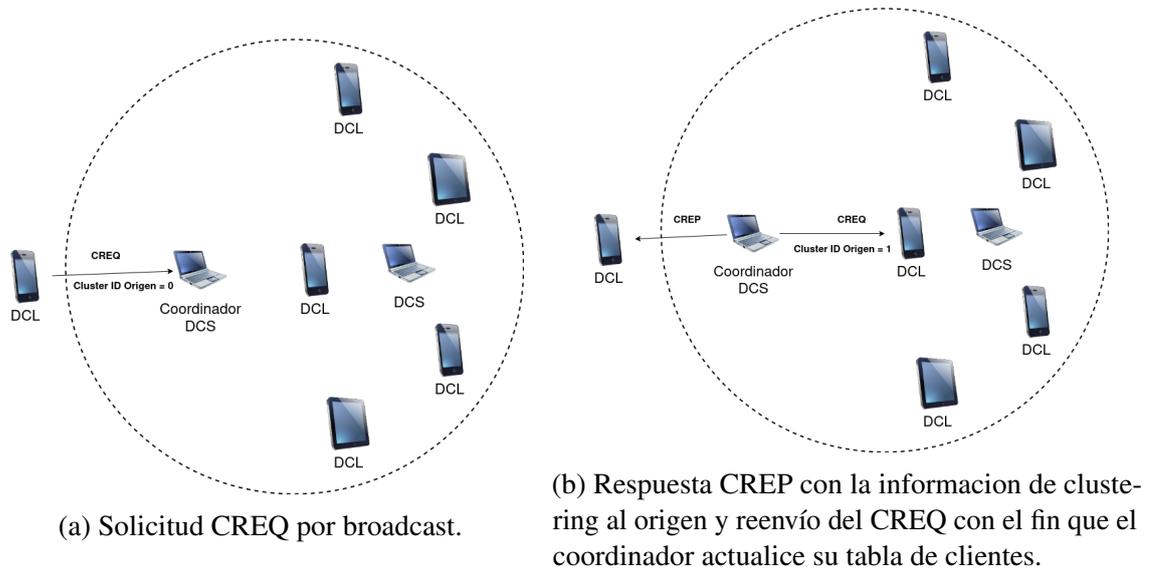
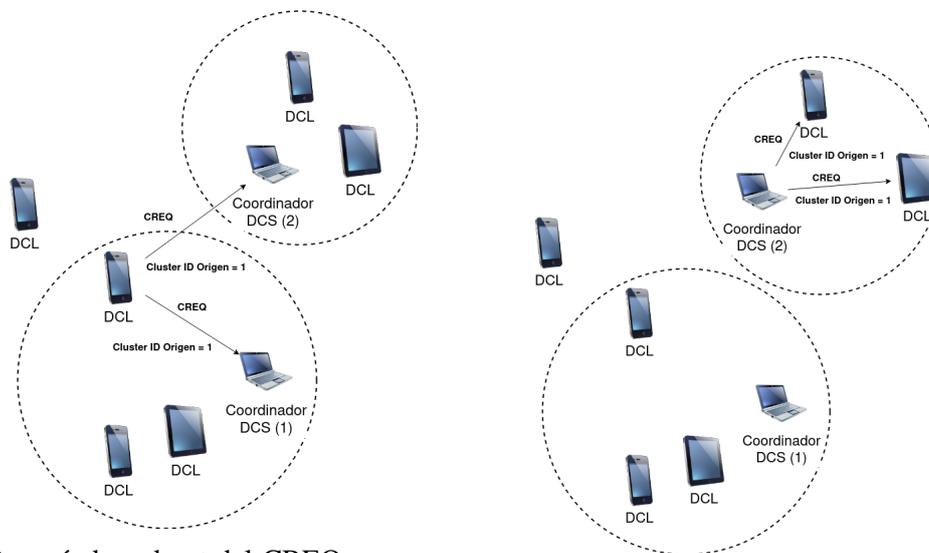


Figura 3.10 Respuesta CREP cuando el receptor es un coordinador intermedio y la solicitud de clustering no fue contestada aún.



(a) Reenvío broadcast del CREQ a un coordinador con el mismo cluster ID que el valor cargado en el *Cluster ID Origen* y a otro con distinto cluster ID.
 (b) Reenvío broadcast del paquete CREQ por cargado en el *Cluster ID Origen* y a otro con el mismo cluster ID.

Figura 3.11 Respuesta CREP cuando el receptor es un coordinador intermedio y la solicitud de clustering ya fue contestada.

En las siguientes líneas se analizarán los distintos escenarios que podrían acontecer cuando un mensaje CREP generado por un origen dado pasa por un nodo intermedio en el camino al destino o directamente llega a destino:

- El nodo receptor es el destino y no tiene asignado cluster. En este caso si el CREP contiene información de la ruta y del clustering este se actualiza con ambas, caso contrario actualizará la información que sea provista por el mensaje, pudiendo ser está tanto la de la ruta o la del cluster y el coordinador.
- El nodo receptor es el destino y ya tiene asignado cluster, en este caso se actualiza la información de la ruta solamente y el resto se descarta.
- El receptor es un nodo intermedio y tiene asignado cluster de pertenencia, aquí solamente se actualiza la información de la ruta solamente y se reenvía el paquete.
- El receptor es un nodo intermedio y no tiene asignado cluster, en esta situación el receptor realiza las siguientes operaciones:
 1. Actualiza la información de la ruta.
 2. Reenvía el mensaje CREP.

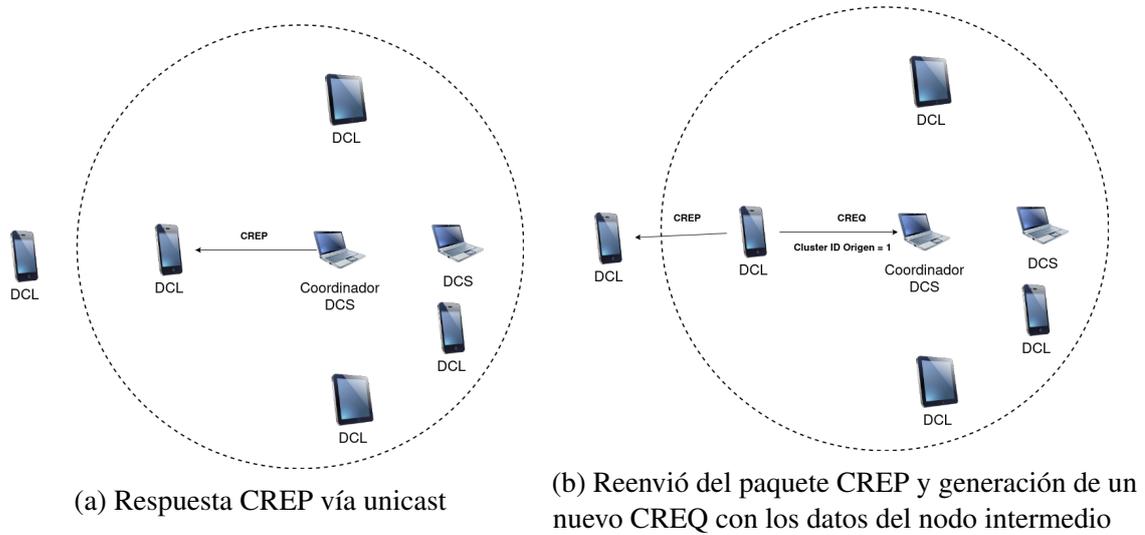


Figura 3.12 Respuesta CREP recibida por un nodo intermedio sin cluster asignado y el mensaje CREP transporta información de clustering.

- Si el mensaje CREP transporta información de clustering entonces el receptor genera un nuevo CREQ destinado al coordinador conteniendo en el campo *Cluster ID Origen* el cluster ID que es transportado por el mensaje CREP. La Figura 3.12 grafica la situación planteada en las líneas anteriores.

En caso que el mensaje CREP no transporte información de clustering entonces no hace ninguna operación adicional.

Una vez establecida una ruta entre ambos nodos, origen y destino, la ruta es válida solo durante un período de tiempo. Ese tiempo se encuentra registrado en el campo *lifetime*. El *lifetime* es el tiempo de expiración de la validez de la ruta y es necesario debido a la volatilidad de los enlaces en este tipo de redes, ya que más adelante esa ruta podría resultar no válida.

El mantenimiento de las rutas sirve precisamente para resolver la situación antes descrita. Si el origen que ha enviado el mensaje se mueve y como consecuencia modifica la topología, entonces debe reiniciar un nuevo proceso de descubrimiento de ruta. Por otro lado, si el nodo que se mueve es un nodo intermedio o se trata del destino, ver Figura 3.13, y existe algún mensaje dirigido a él, entonces el nodo que detecta el fallo genera un mensaje de error y lo envía al nodo que ha originado la comunicación. Este mensaje de error es denominado RERR (*Route Error*) y tiene como propósito invalidar una ruta.

Todos los nodos intermedios por los que pasa el mensaje RERR (*Route Error*), cancelan las rutas hacia el nodo que se ha vuelto inalcanzable. Una vez que el mensaje RERR llega a

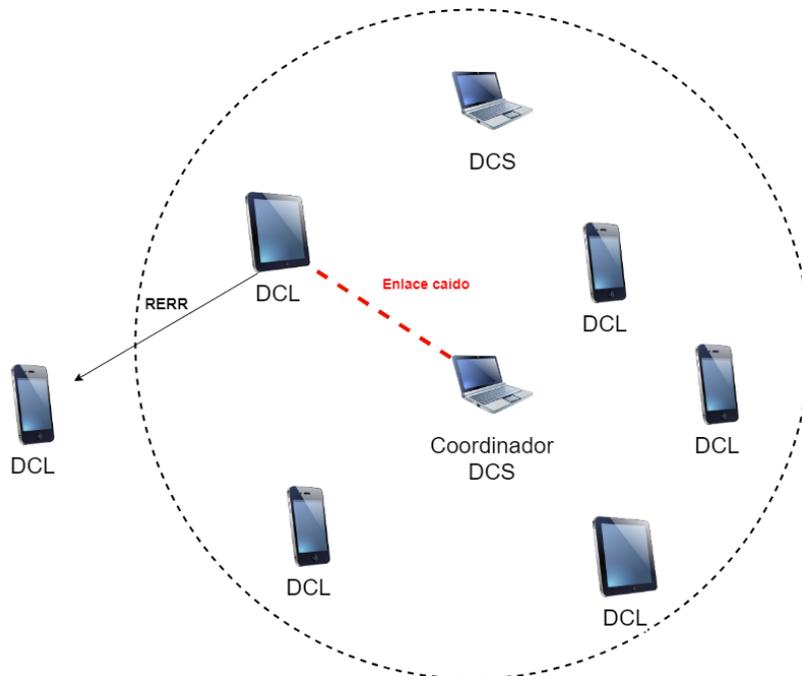


Figura 3.13 Enlace caído hacia el coordinador

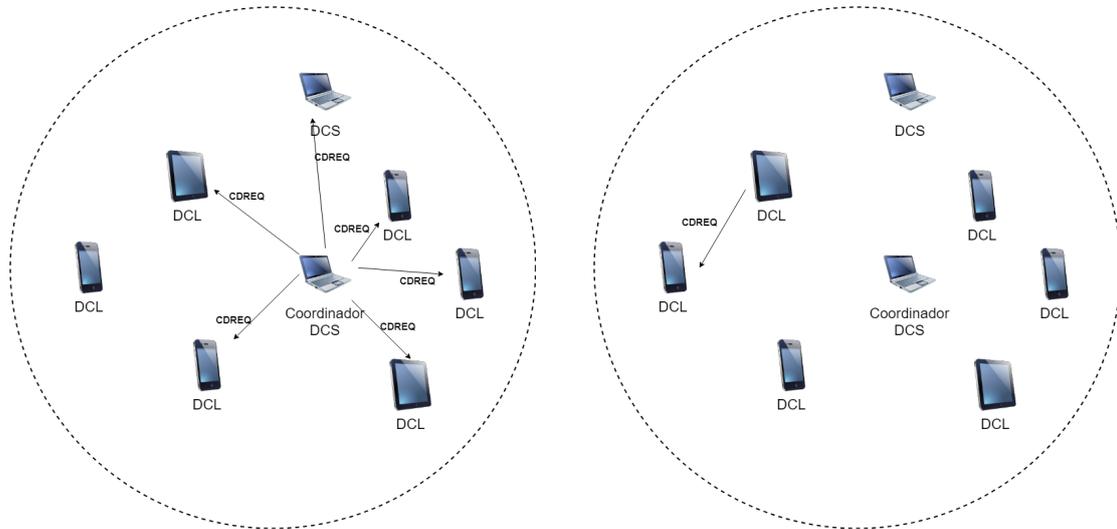
su destino, el origen decide dar por terminado el envío o volver a enviar un nuevo un mensaje CREQ.

3.7. Generación de los mensajes CDREQ, CDREP y RERR

Para comenzar con un nuevo cómputo distribuido sobre un cluster ubicado dentro de una MANET, el coordinador generará un paquete CDREQ (*Calculation Distributed Request*) para cada uno de los clientes que estarán involucrados al cálculo distribuido. El paquete CDREQ, al igual que en el caso anterior, posee un número de secuencia en el campo *RREQ ID*, el cual se utiliza de manera idéntica a como se describió en la sección 3.6.

Los mensajes CDREQ se envían por unicast por el coordinador del clúster y este envía uno específico para cada cliente como se muestra en la Figura 3.14. Este mensaje contiene el campo *Task Number* cuya funcionalidad es indicarles a los clientes cual es la tarea que deben realizar dentro del cómputo científico distribuido.

Cuando un mensaje CDREQ alcanza al destinatario, este inspecciona el campo *Task Number*, busca dentro de un catalogo de tareas cual es la que corresponde con el identificador almacenado en el campo anterior, ejecuta la tarea que le fue asignada y devuelve un mensaje CDREP al origen con el resultado, a través de la ruta inversa por donde le ha llegado



(a) El Coordinador envía mensaje CDREQ a los (b) Dispositivo intermedio reenvía por unicast CDREQ a clientes

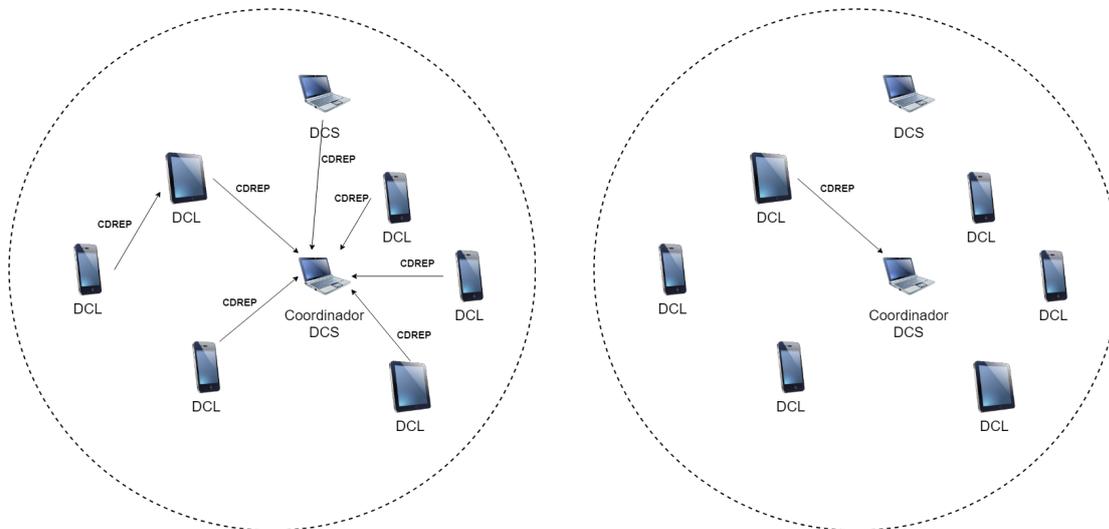
Figura 3.14 Envío unicast de CDREQ

el mensaje CDREQ. En la Figura 3.15 se grafica el envío de los mensajes CDREP al coordinador.

Una vez que el coordinador envía el CDREQ a un nodo, este espera el mensaje CDREP por parte del cliente solamente durante un período de tiempo; ese tiempo se encuentra registrado en el campo *lifetime*. Se maneja un tiempo de expiración debido a la volatilidad de la permanencia de los nodos en este tipo de redes.

Una vez que el coordinador recolecta todos los paquetes CDREP de los clientes que fueron afectados a la computación; en el campo *TNResult* de cada uno de ellos se encuentra el resultado del cálculo. Luego se compagina cada una de las partes del cómputo y se obtiene el resultado final. Si por el contrario, alguno de los paquetes CDREP no llegará al coordinador antes de que se venza el *lifetime*, el coordinador entenderá que esa parte del cómputo se perdió y dará por fallido el mismo, pudiendo más adelante volver a intentar calcular en forma distribuida las partes faltantes con los clientes que registra como activos dentro del clúster y para eso volviendo a generar mensajes CDREQ para la computación de dichas partes.

En las líneas anteriores, se analizó si un cliente alcanzado por un CDREQ se mueve quedando inaccesible y como consecuencia modifica la topología, entonces el coordinador debe anular el cálculo distribuido o reasignar el computo que debía realizar el cliente que ahora es inalcanzable. Por otro lado, si el nodo que se mueve es el coordinador y existe algún computo en proceso, entonces el nodo que detecta el fallo genera un mensaje de error denominado RERR (*Route Error*) y lo envía por multicast a todos los nodos del clúster. El



(a) Los clientes envían mensajes CDREP al coordinador (b) Dispositivo intermedio reenvía por unicast un CDREP

Figura 3.15 Envío unicast de CDREP

mensaje RERR tiene como propósito en estos caso invalidar el compute en cada cliente como se muestra en la Figura 3.16.

Todos los nodos por los que pasa el mensaje RERR que a su vez han sido afectados formando parte en el compute distribuido cancelan la ejecución de dicho compute; una vez que el mensaje RERR llega a todos los nodos, estos inician nuevamente el proceso para establecer una nueva afiliación de cluster y de este modo saber que nodo será su nuevo coordinador.

3.8. Diagrama de secuencia del descubrimiento de clientes

La Figura 3.17 describe cómo funciona el proceso de descubrimiento de clientes en esta propuesta. Dicho patrón de interacción está integrado por dos o tres objetos dependiendo el caso, una serie de eventos y mensajes. Uno de los objetos que intervienen es el nuevo cliente que ingresa al grupo de cómputo y que solicita se le asigne un coordinador, siendo el nuevo cliente en este caso un DCL (Dispositivo de capacidades limitadas). El anuncio de la solicitud de coordinador se transmite por broadcast.

La Figura 3.17 muestra otro objeto que es el coordinador del grupo, que es esencialmente un DCS (Dispositivo de capacidades superiores). Tanto el primer objeto como el segundo son mostrados en las cajas superiores de las figuras: Figura 3.17a y Figura 3.17b. Esta última

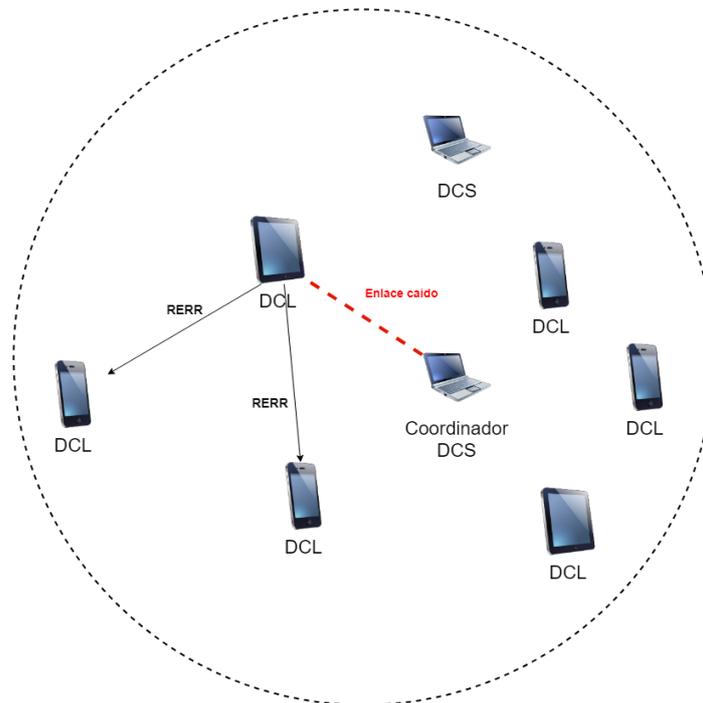
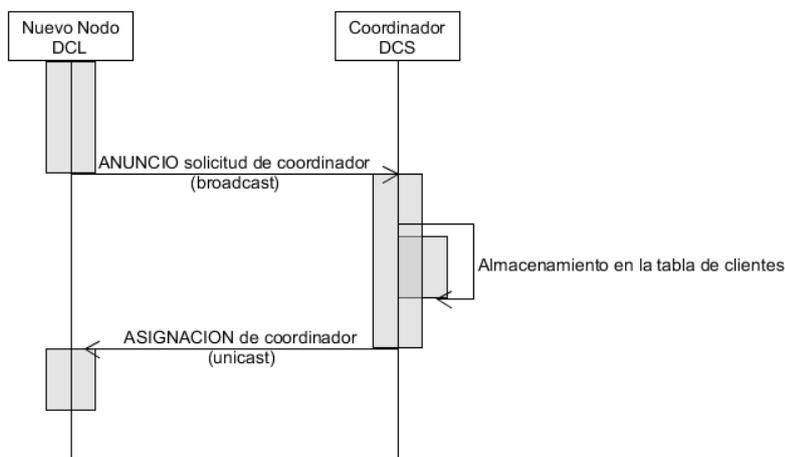


Figura 3.16 Envío de RERR por coordinador fuera de alcance

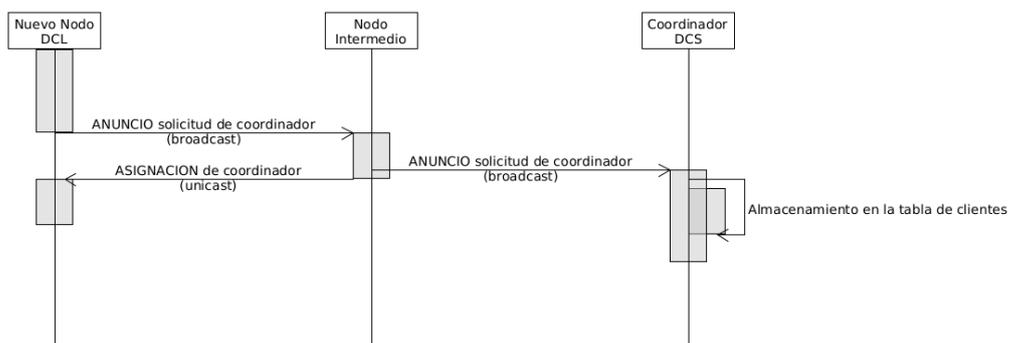
imagen presenta un tercer objeto que es un nodo intermedio entre el primer objeto y el segundo.

El objeto que comienza teniendo el control es el DCL, que es el que envía el primer mensaje con la descripción del evento: ANUNCIO de solicitud de coordinador (broadcast). A partir de este punto pueden ocurrir dos secuencias de eventos e intercambio de mensajes distintas:

- El mensaje ANUNCIO de solicitud de coordinador (broadcast) es recibido por el objeto coordinador, quien ejecuta el evento Almacena el ANUNCIO en la tabla de clientes de PROCCDIMI (PROtocolo para el Cálculo Científico Distribuido sobre clientes Móviles Indeterminados en MANETs); las tablas de clientes son los repositorios en los que los coordinadores registran los clientes con que cuenta el coordinador para llevar a cabo un cálculo distribuido dentro del cluster que administran. Una vez registrado el nuevo cliente en la tabla, el objeto coordinador tiene el control e invoca el evento ASIGNACION de coordinador (unicast) devolviendo la respuesta, en la cual se encuentra la información de que coordinador se le asignó. Todo este último procedimiento está graficado en la Figura 3.17a.
- El mensaje ANUNCIO de solicitud de coordinador (broadcast) es recibido por el objeto nodo intermedio, quien ejecuta el evento mensaje ANUNCIO de solicitud



(a) Diagrama de secuencia del descubrimiento del coordinador vía enlace directo



(b) Diagrama de secuencia del descubrimiento del coordinador vía nodo intermedio

Figura 3.17 Diagramas de secuencia del descubrimiento de clientes

de coordinador (broadcast) con los datos del objeto iniciador e invoca el evento ASIGNACION de coordinador (unicast) devolviendo la respuesta al objeto DCL, en la cual se encuentra la información de que coordinador se le asignó; cuando el mensaje ANUNCIO de solicitud de coordinador (broadcast) del nodo intermedio es recibido por el objeto coordinador ejecuta el evento Almacena el ANUNCIO en la tabla de clientes de PROCCDIMI (PROtocolo para el Cálculo Científico Distribuido sobre clientes Móviles Indeterminados en MANETs) y finaliza la ejecución. Este procedimiento esta graficado en la Figura 3.17b.

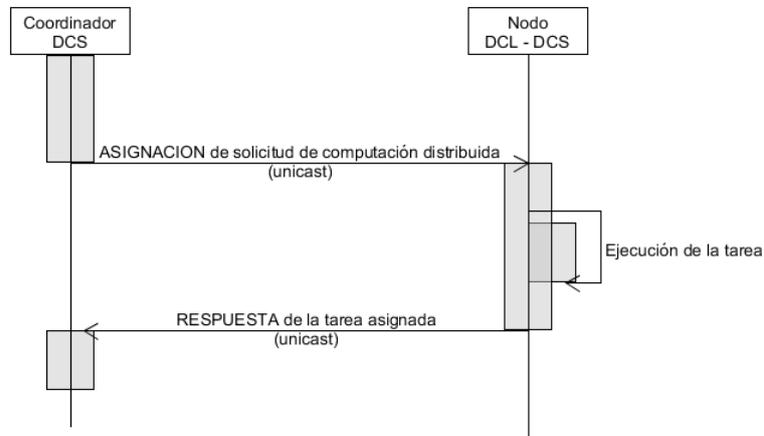


Figura 3.18 Diagrama de secuencia del cómputo distribuido

3.9. Diagrama de secuencia del cómputo de distribuido

En la Figura 3.18 se describe la manera de cómo funciona el proceso de asignación de tareas y recolección de resultado en un cómputo distribuido aquí propuesto. Dicho patrón de interacción está integrado por dos únicos objetos y una serie de eventos y mensajes. Uno de los objetos que intervienen es el coordinador y como se mencionó anteriormente, el coordinador es un DCS (Dispositivo de capacidades superiores), el cual solicita a los clientes que estarán involucrados en el cómputo distribuido que resuelvan una tarea particular, la ASIGNACION se realiza por unicast. Los otros objetos presentes son los clientes, que son esencialmente dispositivos DCL (Dispositivo de capacidades limitadas). Tanto el primer objeto descrito como el segundo son mostrados en las cajas superiores de la Figura 3.18. El objeto que comienza teniendo el control es el DCS, que es el que envía el primer mensaje con la descripción del evento: ASIGNACION de solicitud de computación distribuida (unicast). Este mensaje es recibido por los objetos clientes, quienes realizan el evento ejecución de la tarea asignada. Una vez finalizado este evento, los objetos clientes llevan a cabo el evento RESPUESTA de la tarea asignada (unicast) devolviendo la respuesta al coordinador.

3.10. Administración del clúster

En la solución aquí planteada, un nodo con capacidades superiores asume el rol de coordinador, pudiendo haber más de uno en cada clúster, esto con el fin de dar robustez al clúster ante eventuales *off line* o fuera de alcance de la red.

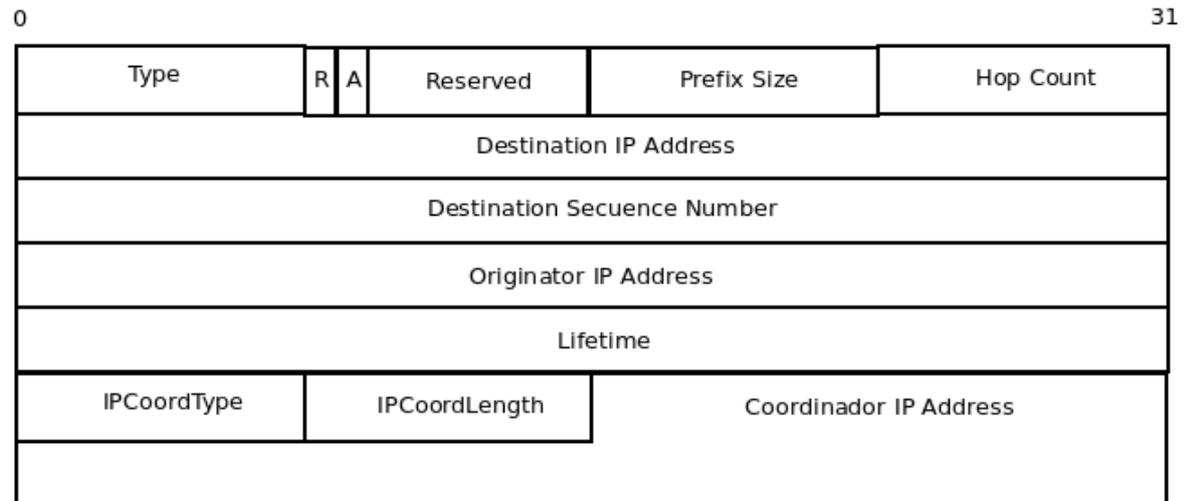


Figura 3.19 Formato del paquete CDHello

El formato de mensaje CDHELLO (*Calculation Distributed Hello*) que se muestra en la Figura 3.19 es esencialmente un mensaje CREP (*Coordinador Reply*), del que ya sea desarrollado el contenido de su cabecera en la sección 3.4.1. La primera diferencia que rápidamente se aprecia radica en que el formato de mensaje CDHELLO (*Calculation Distributed Hello*) tiene en el campo TTL (*Time to live*) el valor de 1 salto, esto con el objetivo de que el mensaje solo se difunda en la cercanía del vecindario. Y en segunda instancia se notará que los campos: *IPMCType*, *IPMCLength*, *IPMulticastCluster*, *CIIDType*, *CIIDLength*, *Cluster ID* y *Cluster ID* han sido eliminados mientras que los campos *IPCoordType*, *IPCoordLength* y *Coordinador IP Address* se han mantenido. Esta modificación del paquete original tiene por objetivo mantener actualizada en todo momento la dirección IP del nodo que actualmente desempeña el rol de coordinador en los nodos que se encuentran dentro del grupo. Cualquier modificación respecto a esta última condición es rápidamente comunicada a todos los nodos mediante los envíos periódicos de los paquetes CDHELLO entre vecinos. De esta manera se acortan significativamente los tiempos de notificación y también se evita las notificaciones broadcast o multicast con el consecuente ahorro en la utilización de los recursos de los dispositivos pertenecientes al cluster.

Cada cierto lapso determinado de tiempo se envían mensajes CDHELLO a los vecinos, con objeto de avisar que el nodo que ha enviado el mensaje aún se encuentra activo. Con la llegada de este mensaje, los nodos vecinos actualizan los *lifetime* asociados a ese nodo o en caso contrario eliminan la entrada de la tabla de enrutamiento de los nodos que no respondan.

El mecanismo de mantenimiento de los clusters, consiste específicamente en el envío periódico de mensajes CDHELLO anunciando la presencia de un nodo en el vecindario. Y al mismo tiempo, los nodos en el clúster mantienen actualizada la identidad del coordinador.

Con la extensión del campo *Coordinador IP Address* en el mensaje *HELLO* de AODV, la cantidad de mensajes enviados en la red no se incrementa, ni se modifica para nada su propósito original de determinar la conectividad entre vecinos, solamente agrega una funcionalidad, la de mantener actualizada la información del coordinador de cluster.

Si algún nodo vecino al coordinador recibe un mensaje CDHELLO de este último y no vuelve a recibir un nuevo de ese mismo emisor, entonces se puede suponer que se ha perdido el enlace y pudiéndose atribuir a varias causas: que no este activo, que se ha desplazado lejos del alcance, que se le han agotado sus recursos entre otras.

Si los nodos del clúster no perciben la presencia del coordinador, suponen que no se encuentra disponible en ese momento, y dado que en un clúster existe más de un DCS, entonces del resto de los DCS activos, uno asume en ese momento el papel de coordinador y se continúa trabajando con las peticiones que comenzará a encolar el novel líder del grupo.

3.11. Resumen

Para cerrar el capítulo se resumirán las características más relevantes de PROCDDIMI (PROtocolo para el Cálculo Científico Distribuido sobre clientes Móviles Indeterminados en MANETs). Este es un protocolo de descubrimiento y asignación de tareas a clientes que pertenece a los esquemas que caen en la categoría de arquitectura de multi-agente distribuida con diseño cross layer en un esquema de clustering. Además, parte de la premisa que una MANET está compuesta por unidades de red heterogéneas y que dentro de esta variedad de clases de dispositivos, se pueden distinguir a los dispositivos que tienen capacidades superiores y los que no. De este modo se logra una clasificación con el objetivo de poder elegir coordinadores de cluster idóneos para dicha función dentro de la red.

Es una solución que utiliza diseño cross layer, porque descubre clientes y asigna tareas en la capa de red y no en la capa de aplicación como muchas otras soluciones. Utilizando el protocolo de encaminamiento reactivo AODV (*Ad hoc On-demand Distance Vector*) como protocolo de red subyacente. Para ello, durante el proceso de encaminamiento se insertan varios campos dentro de la cabecera de algunos mensajes de AODV, estos campos pueden contener tanto información de la potencia del nodo, como información de clustering o de la tarea que se le asignó al mismo o el resultado de la misma. La información del tipo de nodo tiene como fin distinguir de qué clase de dispositivo se trata, bien sea un DCS (Dispositivo de capacidades superiores) o bien sea un DCL (Dispositivo de capacidades limitadas). Y respecto a que emplea técnicas de clustering, porque agrupa los nodos de un área determinada y les establece coordinadores para el soporte del proceso de descubrimiento de clientes y administración de tareas.

El criterio para elegir coordinadores consiste esencialmente en diferenciar y seleccionar los dispositivos que poseen capacidades superiores de aquellos dispositivos limitados en recursos. Una vez seleccionados los dispositivos con capacidades superiores, de estos emergerán los elegidos para desempeñar la tarea de coordinador.

Capítulo 4

Simulación

En el capítulo anterior se formuló la propuesta de solución para encarar el cómputo científico distribuido en redes MANET con clientes indeterminados. A continuación se validará la misma. En general, con este tipo de tesis, se cuenta con dos maneras de validación de las propuestas que en ellas se realizan. La primera mediante el modelado formal matemático del protocolo, a través de teoría de probabilidades mayoritariamente, formulando teoremas matemáticos junto con sus correspondientes demostraciones validando la propuesta a través de la mejor muestra de orientaciones de cotas inferiores y superiores de los parámetros a mensurar. También cabe mencionar a las Redes de Petri [60] como una posible opción para llevar a cabo un modelado formal, éstas se construyen mediante el uso de una técnica gráfica fundamentada matemáticamente que permite el estudio de sistemas informáticos complejos no deterministas, paralelos y distribuidos. Una red de Petri se compone de lugares, transiciones, arcos dirigidos y marcas. Sin embargo para modelar protocolos destinados a funcionar sobre redes se utiliza un subtipo particular, las Redes de Petri Coloreadas [82] que se caracterizan por proveer el soporte necesario para la construcción de modelos formales que logran representar todas las interacciones presentes en los protocolos de red multi-agente. La segunda forma es realizar simulaciones, reproduciendo en computadora los procedimientos del protocolo y ajustando valores a los parámetros. Los resultados de las simulaciones son interpretados como conocimiento empírico. Este último es el método heurístico de validación elegido, basándose esta decisión en el hecho de que cuando se realizó la investigación de las distintas formas de comprobación aplicables en este estudio, arrojó que los modelos analíticos para las redes móviles inalámbricas de múltiples saltos resultaban ser considerablemente imprecisos [5], convirtiéndolos en inadecuados para validar con un grado razonable de precisión el novel protocolo aquí propuesto. De las herramientas de simulación examinadas se adoptó NS2 (*Network Simulator version 2*) porque una de sus mayores ventajas son todas las facilidades que provee para implementar e incorporar extensiones al mismo. Facilidades que

fueron extremadamente útiles para implementar a PROCCDIMI, extender la versión estándar de NS2 (*Network Simulator version 2*) con este y simular sobre los escenarios diseñados. Una de las tareas que requirió un porcentaje considerable del esfuerzo imprimido en esta investigación consistió en desarrollar desde cero hasta su totalidad a PROCCDIMI bajo el marco de desarrollo que especifica la herramienta de simulación para nuevas extensiones y así poder realizar las pruebas de rendimiento al nuevo protocolo. Esta capacidad de poder crear nuevas soluciones e incorporarlas al entorno utilizado, no todos los simuladores lo permiten, en algunos casos no se dispone de la suficiente documentación para poder llevar a cabo el desarrollo o el proceso de inclusión de nuevas soluciones es muy complejo. Todas estas complicaciones enumeradas anteriormente no se encuentran presentes en NS2.

Este apartado introduce el contexto y los lineamientos de las simulaciones realizadas cuyos resultados serán utilizados en los capítulos posteriores. La primera sección aborda la selección del software para llevar a cabo las simulaciones, junto con una descripción exhaustiva de las fortalezas y debilidades de los distintos candidatos examinados. La sección 4.2 describe minuciosamente la herramienta de simulación seleccionada junto con su arquitectura de funcionamiento, los formatos de las trazas de salida y la ejemplificación de que parámetros se pueden configurar en la herramienta por medio de un script. En la sección 4.3 se revisan los modelos de movilidad más relevantes para este estudio, describiendo como es su funcionamiento y cual se utilizará en las simulaciones a realizarse. A continuación en el apartado 4.4 se analizan los tipos y la densidad de nodos que serán necesarios tener presentes en los escenarios de simulación. Luego en la sección 4.5 se describen la topología y el generador de tráfico que serán utilizados, para luego realizar el abordaje del método de validación y así concluir el apartado. La sección 4.6 presenta los escenarios de simulación diseñados para esta tesis y el capítulo concluye con las consideraciones finales expuestas en la sección 4.7.

4.1. Selección del software de simulación

Para implementar y medir la performance de los nuevos protocolos mediante herramientas de simulación hay que advertir con que fortalezas y desventajas se deberá lidiar al elegir una de ellas. Las características a las cuales se le prestó especial atención al momento de seleccionar el simulador son listadas a continuación [76]:

- El comportamiento general que tiene el sistema previamente a la simulación, durante y en la recolección de los datos producidos.
- Que los resultados sean precisos en general con el modelo analítico.

- Que los fenómenos inexplicables sean relativamente fáciles de encontrar.
- El experimento de análisis hipotético sea fácil de realizar y que sea de bajo costo en comparación con el experimento práctico.
- Que sea de open source, aunque esta característica no esta listada en [76] tiene un peso determinante en la selección del simulador.

También del mismo artículo se tomaron las limitaciones consideradas:

- El nivel de dificultad que presenta al momento de la construcción del modelo de simulación.
- Que tan complejo resulta la conducción de la distintas simulaciones que deben llevarse a cabo a fin de arribar a un resultado.
- La dificultad en recolectar e interpretar los resultados.

Como ya se mencionó anteriormente las MANET se presentan como una tecnología de comunicación móvil de rápida conformación que ofrece servicios para los más diversos campos, como por ejemplo: los ámbitos militar, servicios de emergencia, entretenimiento, redes de sensores, hogar, redes vehiculares, entre otros. Al tener nodos móviles con administración descentralizada, la experimentación requiere mucho tiempo y es costosa. Para superar estos inconvenientes, se encuentran disponibles varias herramientas de simulación que principalmente se utilizan para la implementación y experimentación de nuevos tipos de protocolos. Dentro de sus usos principales se puede mencionar también la medición del rendimiento de nuevos protocolos mediante la aplicación de diferentes métricas cualitativas de rendimiento y de parámetros de configuración. Esta sección examina y describe brevemente los diferentes simuladores de red que se utilizan en MANET y otras redes. Los principales son: NS2 [39], NS3 (*Network Simulator version 3*) [16], GloMoSim (*Global Mobile Information System Simulator*) [9], OPNET (*Optimized Network Engineering Tools*) [19], QualNet (*QualNet® network simulation software*) [26] y MATLAB (*MATrix LABORatory*) [35].

4.1.1. NS2 (Network Simulator version 2)

NS2 (*Network Simulator version 2*) es un simulador de eventos discretos. Es la más popular y libre herramienta de simulación utilizada en el campo de investigación de redes MANETs [39]. Fue desarrollado por la Universidad de California en Berkeley y es ampliamente utilizado por la comunidad científica y educativa [76]. Está equipado con una gran

variedad de protocolos y modelos. También posee una gran comunidad del ámbito del open source que ofrece una fuente sustancial de soporte técnico [24].

NS2 se encuentra dividido en dos partes en cuanto a lenguaje de programación refiere, la que opera con el lenguaje OTCL (*Object-oriented Tool Command Language*) y la que opera con el lenguaje C++. El primero es un interprete usado para ejecutar comandos y facilita la configuración de los objetos; mientras que el segundo refleja fiel y eficientemente todo lo correspondiente a los objetos definidos en OTCL. Es relevante comentar que con C++ es como los protocolos son implementados. Esto posibilita tener una detallada descripción de la operación de los protocolos, paquetes y colas de procesamiento disponibles en NS2. Entre los dos lenguajes existe un puente de comunicación que los conecta permitiendo que OTCL puede usar objetos C++ a través de un *OTCL Linkage*. Este mecanismo crea concordancias entre las estructuras de datos definidas en OTCL y los objetos C++.

Se eligió NS2 principalmente porque es una herramienta de simulación probada y utilizada mayoritariamente en muchos estudios previos sobre MANET [53], además de haber sido validada y verificada en numerosos trabajos de investigación revisionistas sobre las distintas alternativas de simulación existentes para este tipo de redes [45] [40]. Cabe señalar que NS3 (*Network Simulator version 3*) no es una versión mejorada o actualizada de NS2 (*Network Simulator version 2*), sino que muy por el contrario es una herramienta totalmente distinta enfocada más a la didáctica que a la modelización rigurosa del funcionamiento de todas las partes componentes de una MANET (*Mobile Ad hoc Networks*, Red móvil ad hoc) y destinada al uso dentro de la comunidad académica tal como sucede con NS2.

Por todas las características antes mencionadas NS2 es la herramienta de simulación elegida para llevar a cabo la experimentación de PROCCDIMI. En el apartado 4.2 se continua profundizando sobre el tema.

4.1.2. NS3 (Network Simulator version 3)

NS3 (*Network Simulator version 3*) es un simulador de eventos discretos principalmente orientado a la didáctica y con propósitos educativos. La iniciativa fue impulsada en 2006 [67] [16] y a diferencia de lo que se puede suponer no es una extensión de NS2. NS3 es un simulador distinto de NS2 aunque comparten similitudes entre ambos, la principal sería que los dos están escritos en C++ pero NS3 no soporta la API de NS2 lo cual es un gran limitante porque la comunidad que trabaja con NS2 es mucho más amplia que la de NS3. Otro problema que presenta NS3 con respecto a NS2 es la portabilidad, ya que se puede realizar una migración de NS2 a NS3 pero el camino inverso no es posible.

4.1.3. GloMoSim (Global Mobile Information System Simulator)

GloMoSim (*Global Mobile Information System Simulator*) es una herramienta que funciona como un mecanismo de eventos discretos, posee algunas características interesantes como el diseño modular así como la posibilidad de ser escalable [9] [64]. Sin embargo, la pila de protocolos en esta herramienta es relativamente simple y carece de algunos protocolos de enrutamiento muy difundidos como TORA y OLSR para los cuales no se ofrece implementación alguna.

4.1.4. QualNet (QualNet® network simulation software)

QualNet (*QualNet® network simulation software*) es un simulador comercial que extiende la pila de protocolos que maneja GloMoSim y además provee una GUI (*Graphic User Interface*, Interfaz gráfica de usuario) al usuario del mismo. También posee la cualidad de ser escalable y soporta redes con una gran cantidad de nodos (Por ejemplo redes de 2500 nodos) [26].

4.1.5. OPNET (Optimized Network Engineering Tools)

OPNET (*Optimized Network Engineering Tools*) es una herramienta de simulación comercial que ofrece una GUI. Posee muchos componentes los cuales se encuentran bien organizados, como por ejemplo patrones de movilidad, modelos de propagación de señales, protocolos que operan en la capa MAC y muchos protocolos de enrutamiento.

4.1.6. MATLAB (MATrix LABoratory)

MATLAB (*MATrix LABoratory*) crea entornos de computación numérica que habilita a los usuarios a realizar tareas de cómputo intensivo más rápido que la manera tradicional, además posee algunas librerías especiales o *toolboxes* para telecomunicaciones y movilidad pero su alcance es limitado.

4.1.7. Resumen del software de simulación

En la Tabla 4.1 se presentan todas las herramientas de simulación analizadas en el apartado 4.1, junto con sus características más sobresalientes. Principalmente se especifica cual es la interface de programación que utiliza para llevar acabo las simulaciones y la adición de nuevos protocolos. También se indica que tan extensible es, cual es el grado de adopción por la comunidad científica y por último si es open source o no.

Herramienta	Lenguaje	Extensibilidad	Uso académico	Open source
NS2	C++/OTcl	Excelente	Muy Alto	Si
NS3	C++/Phyton	Buena	Alto	Si
GloMoSim	Parsec(C)	Buena	Bajo	No
QualNet	Parsec	Buena	Bajo	No
OPNET	C/C++	Buena	Alto	No
MATLAB	Híbrido	Buena	Medio	No

Tabla 4.1 Resumen del software de simulación

4.2. Simulador de Red NS2

Como se comentó en el apartado 4.1.1, el rol de NS2 (*Network Simulator version 2*) es tan importante en la comunidad científica que investiga las MANETs que es el simulador de referencia de-facto [47], además la flexibilidad que posee se interpreta como una conveniencia sumado a la cualidad de ser open source contribuyen al éxito del mismo y por lo tanto la herramienta de simulación seleccionada para la presente tesis doctoral. Una de las principales razones para utilizar esta herramienta es que el protocolo sobre el que se sustenta PROCCDIMI ya se encuentra programado y por consiguiente se considera como válido la reutilización de estos objetos, dado que se ha contrastado suficientemente dicho código y se asume como correcto plantear una simulación en estos términos.

NS2 es un simulador open source de eventos discretos desarrollado en 1989 por la Universidad de California en Berkeley y que desde ese punto ha ganado un enorme interés desde todos los ámbitos relacionados a las redes, desde el gubernamental hasta el académico siendo este el nicho donde más relevancia tiene. A partir de 1995 DARPA (*Defense Advanced Research Projects Agency*, Departamento de proyectos avanzados de investigación de los Estados Unidos) dio soporte a NS2 a través del proyecto de prueba de VINT (*Virtual InterNetwork Testbed*)¹ [39]. Desde su creación hasta el día de hoy ha ido creciendo sus capacidades de simular entornos de red mediante la incorporación de nuevas implementaciones de protocolos de ruteo, transporte y aplicación; junto con esto último también se incorporó la posibilidad de configurar entornos de simulación para nuevos tipos de redes como por ejemplo las redes móviles ad hoc. Esta última ampliación fue desarrollada por el grupo Monarch de CMU a fin de permitir el análisis de las simulaciones sobre redes inalámbricas.

¹El proyecto VINT (*Virtual InterNetwork Testbed*) es la unión de esfuerzos de gentes de UC Berkeley, USC/ISI, LBL y Xerox PARC. El proyecto es soportado por DARPA (*Defense Advanced Research Projects Agency*, Departamento de proyectos avanzados de investigación de los Estados Unidos).

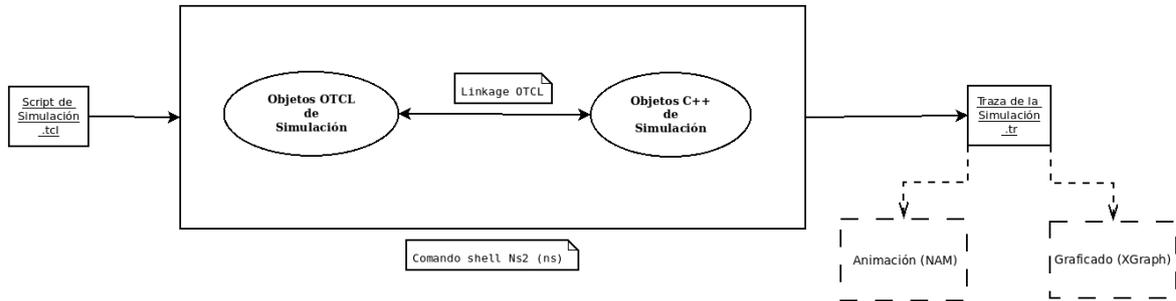


Figura 4.1 Arquitectura básica de NS2

El código fuente de NS2 se distribuye principalmente bajo la filosofía del open source, lo que permite que en el simulador se encuentren aportes de varios centros de investigación de todo el mundo.

4.2.1. Arquitectura de NS2

En la Figura 4.1 se muestra muy sintéticamente la arquitectura de NS2. El simulador es provisto mediante un ejecutable llamado *ns* que toma como entrada un archivo *.tcl* que contiene la configuración necesaria para llevar a cabo la experimentación. La salida una vez finalizado todo el proceso se almacena en un archivo de traza que suele ser a su vez la entrada de otras herramientas asociadas al procesamiento de texto y generación de gráficos cartesianos.

NS2 usa dos lenguajes de programación, C++ y OTCL, mientras que el primero se utiliza para definir todo lo que a mecanismos internos refiere, el segundo es utilizado para la configuración del entorno a simular y como se puede apreciar en la Figura 4.1 la comunicación entre los dos lenguajes la lleva a cabo el *Linkage OTCL*. Este último se encarga del mapeo de objetos C++ y variables al dominio OTCL.

NS2 provee una gran cantidad de clases C++ en forma nativa, las cuales pueden ser configuradas vía scripts de simulación *.tcl*. Sin embargo en algunas ocasiones los parámetros que se pueden configurar son insuficientes, en estos casos está previsto que se pueda definir nuevos atributos en las clases C++ y usar las interfaces de configuración OTCL para hacerlos accesibles al usuario. Del mismo modo, NS2 cuenta con la capacidad de crear aplicaciones y protocolos personalizados. De forma nativa se dispone de varios protocolos implementados que pueden ser ampliados o implementar nuevos a partir de ciertas especificaciones. Esto último se lleva a cabo mediante la programación en el lenguaje C++.

Cuando se ejecuta *ns* tomando como entrada un archivo *.tcl* y este proceso concluye sin errores, se genera un archivo de texto basado en los resultados de la simulación. A este archivo se lo denomina archivo de traza *.tr*. Para interpretar los resultados volcados en el

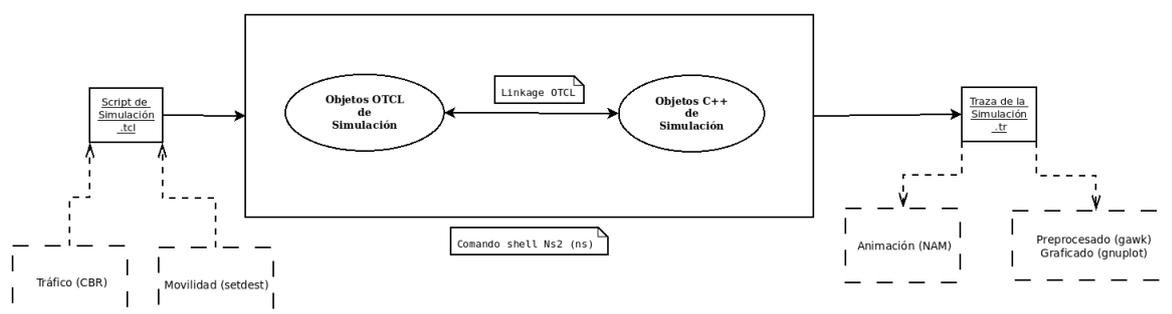


Figura 4.2 Arquitectura básica de NS2 para PROCCDIMI

archivo de traza, tanto gráficamente como interactivamente, se dispone de dos herramientas: NAM (*Network AniMator*) con el ejecutable *nam* que es el animador de la simulación y *XGraph* con el ejecutable *xgraph* que se utiliza para la creación de gráficos cartesianos.

Arquitectura de NS2 ajustada a PROCCDIMI

Se puede observar en la Figura 4.2 la arquitectura utilizada para realizar las simulaciones de PROCCDIMI (PROtocolo para el Cálculo Científico Distribuido sobre clientes Móviles Indeterminados en MANETs) que se conserva el núcleo de NS2 sin modificación alguna. También con un mayor nivel de detalle se distingue el mecanismo que resuelve los aspectos relacionados a la de generación de tráfico y en particular es resuelto mediante el uso de un generador CBR (*Constant Bit Rate*, Tasa de bits constante). Además se individualiza la herramienta *setdest*, utilizada para simular el comportamiento de movilidad de las unidades de red, de acuerdo con los modelos de movilidad más utilizados en estos ámbitos.

La principal diferencia entre la arquitectura básica de NS2 y la utilizada para PROCCDIMI es que al concluir la simulación los resultados que se registran en el archivo de traza, se les realiza un preprocesamiento con el lenguaje *awk*, en particular con la implementación GNU llamada *gawk*, y posteriormente se visualizan los mismos con la herramienta *gnuplot*.

4.2.2. Formato de las trazas de NS2

El producto final de toda simulación sobre NS2 (*Network Simulator version 2*) es un archivo de traza, donde cada línea o registro de traza posee un formato predefinido. Aunque existen tres tipos principales de formato, y estos son:

- Formato de traza normal: Está asociado con la simulación de red cableada.

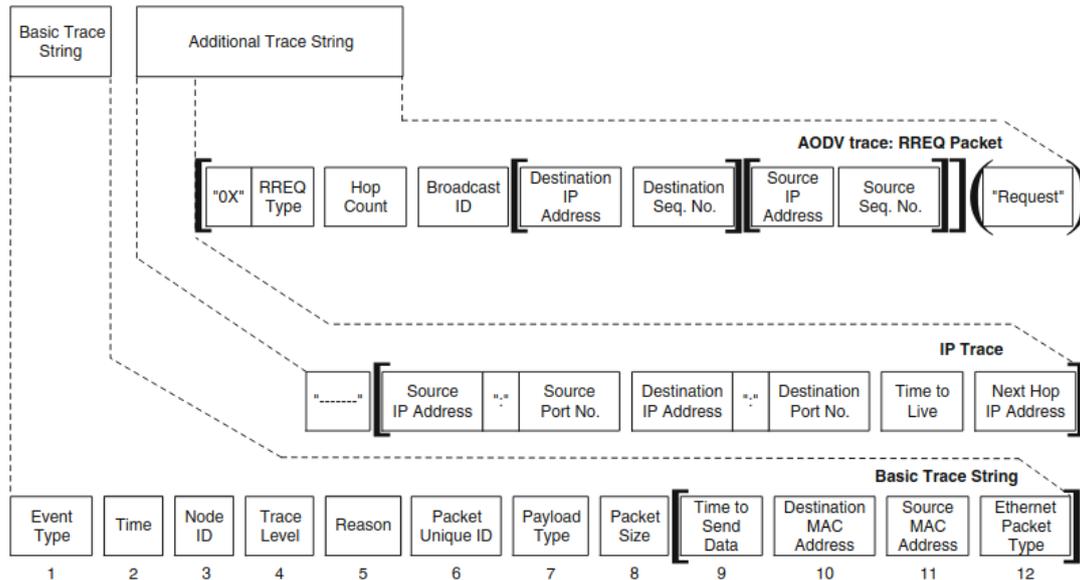


Figura 4.3 Formato de traza inalámbrico. Imagen extraída de [39]

- Formato de seguimiento de paquetes inalámbricos: Esta fijado como predeterminado cuando se ejecuta una simulación de red inalámbrica. También conocido como formato antiguo o de traza CMU.
- Formato nuevo de traza inalámbrico: Esta integrado en NS2 y es mucho más completo que su predecesor. Para utilizar el nuevo formato de traza hay que activarlo mediante la siguiente instrucción OTCL: `ns use-newtrace` donde `ns` es el objeto que representa al simulador.

Los formatos inalámbricos, antiguo y nuevo, difieren en que el primero sigue estructuras predefinidas y la interpretación se basa principalmente en las posiciones de las cadenas en un registro de trazas, mientras que el segundo tipo no tiene estructura y la interpretación se basa en la etiqueta ubicada antes de una cadena de rastreo. De la diferencia analizada en las líneas anteriores resulta evidente que la estructura del formato antiguo de traza se ajusta más a las necesidades existentes en esta investigación.

Como se ve en la Figura 4.3, el formato de traza de paquetes wireless se compone de dos partes: la primera *Basic tracestring* que es obligatoria y aparece en cada registro del archivo y la segunda que es opcional utilizada específicamente para el protocolo, pudiendo variar su contenido según del protocolo que se trate pudiendo ser: TCP, DSR, AODV, entre otros. La Figura 4.3 también muestra dos ejemplos de formato de traza opcional, a saber, *IPtrace* y *AODV*.

Los registros de trazas comienzan siempre con un tipo de evento, que puede ser *send(s)*, *receive(r)*, *drop(d)* o *forward(f)* seguidos del instante de tiempo en que el registro fue creado y que nodo es el que lo origina. A continuación se listan los campos restantes:

- *Trace level*: Los niveles de traza más comunes son: *AGT* para indicar que pertenece a un agente, *RTR* para traza de ruteo y *MAC* para traza MAC (*Media Access Control*, Control de acceso al medio).
- *Reason*: La razón de creación del registro. Por ejemplo, *NRTE* para indicar *No Route Entry*.
- *Time to Send Data*: Duración esperada para transmitir un paquete a través del canal inalámbrico según lo indicado por el protocolo MAC subyacente.
- *Ethernet Packet Type*: Existen solamente dos tipos: IP y ARP.
- *RREQ Type*: Tipo que indica que se trata de información relacionada con el protocolo de enrutamiento.

Un registro típico en un archivo de traza *.tr* posee la siguiente estructura:

```
s 0.000000000 _10_ RTR --- 0 AODV 48 [0 ffffffff 17 800] ---
----[23:255 -1:255 30 0] [0x2 1 1 [25 0] [23 4]] (REQUEST)
```

La interpretación del registro anterior es la siguiente: El nodo *_10_* envía en el tiempo 0.000000000 segundos. El nivel es de traza de ruteo y el paquete tiene un ID único de 0 conteniendo información del protocolo de enrutamiento AODV y posee un tamaño de 48 bytes. El protocolo de ruteo asume que la demora sobre el canal inalámbrico es 0 segundos. El origen y destino de la transmisión son: 0 y ffffffff respectivamente. Finalmente es un paquete IP que va sobre una red Ethernet.

Para el formato IP de la traza, este paquete es etiquetado con origen y destino 23 y 1 respectivamente y ambos utilizan el puerto 255. El número de saltos máximos o TTL (*Time to live*) es 30 y la dirección del próximo salto es 0.

Finalmente, el formato de la traza que corresponde a AODV (*Ad hoc On-demand Distance Vector*) señala que es un paquete RREQ (*Route Request*), indicado en la traza por el valor 0x2, el contador de saltos es 1 y el identificador de broadcast es 1. La dirección IP de destino y número de secuencia es 25 y 0 respectivamente. Mientras que la dirección IP de origen y número de secuencia es 23 y 4 respectivamente. Por último, la etiqueta *REQUEST* confirma que se trata de un paquete AODV RREQ.

4.2.3. Script de ejemplo en OTCL

Este apartado tiene como objetivo dar una idea de la potencia de OTCL (*Object-oriented Tool Command Language*), mediante el análisis de un script de ejemplo.

En el inicio del archivo de configuración se define que la simulación consta de 10 nodos móviles trasladándose por un área plana de 500 metros de ancho por 500 metros de largo. Otro parámetro que vale la pena comentar es el tiempo total de simulación, que para este caso es de 10 segundos.

Una vez que comienza el experimento se producen los siguientes eventos:

- A los 0.01 segundos, los 10 nodos aparecen sobre el escenario en sus correspondientes posiciones iniciales y comienzan a moverse libremente sobre el área de 500 x 500 metros.
- En el instante 0.05 segundos los 10 nodos establecen comunicación entre ellos y comienza el intercambio de paquetes.
- A los 10 segundos el sistema avisa a todos los nodos que el tiempo ha terminado, para luego proceder a finalizar la ejecución del simulador.

El script completo en OTCL que ejecuta las acciones antes descritas se muestra a continuación y se ha incorporado líneas de comentarios para facilitar su comprensión:

```
# Se definen todas las variables globales de configuración

set opt(chan)      Channel/WirelessChannel    ;# tipo de canal
set opt(prop)     Propagation/TwoRayGround   ;# modelo de propagación de radio
set opt(netif)    Phy/WirelessPhy           ;# tipo de interface de red
set opt(mac)      Mac/802_11                 ;# tipo de MAC
set opt(ifq)      Queue/DropTail/PriQueue    ;# tipo de cola de paquetes
set opt(ll)       LL                          ;# tipo de capa de enlace
set opt(ant)      Antenna/OmniAntenna        ;# modelo de antena
set opt(x)        500                         ;# Dimensión en el eje X
set opt(y)        500                         ;# Dimensión en el eje Y
set opt(ifqlen)   50                          ;# paquete máximo en ifq
set opt(nn)       10                          ;# cantidad de nodos móviles
set opt(connections) 5                          ;# Nro de conexiones por los nodos
set opt(stop)     10                          ;# tiempo de finalización
```

```
set opt(dataRate)      [expr 1.0*256*8]      ;# tamaño de paquete (256 bytes)
set opt(adhocRouting)  AODV                  ;# protocolo de ruteo

# Creación del simulador.
set ns_ [new Simulator]

# Creación del objeto topográfico
set topo [new Topography]

# Nombre para los archivos de traza y de animación.
set opt(fn) "wireless_AODV"

# Crea el archivo de trazas y se abre para escritura (w) wireless_AODV.tr
set tracefd [open $opt(fn).tr w]

# Crea el archivo de animación y se abre para escritura (w) wireless_AODV.nam
set namtrace [open $opt(fn).nam w]

$ns_ trace-all $tracefd ;# el trazado se envía al archivo
$ns_ namtrace-all-wireless $namtrace $opt(x) $opt(y)

# Declaración de la función que se ejecuta cuando el programa termina.
proc finish {} {
  global ns_ tracefd namtrace
  $ns_ flush-trace
  close $tracefd
  close $namtrace
  exit 0
}

# Definición de la topología
$topo load_flatgrid $opt(x) $opt(y)
```

```

# Crea al God(Generate Operations Director)
set god_ [create-god $opt(nn)]

# Define como deben ser creados y configurados los nodos móviles.
# Estas variables han sido declaradas al principio de este archivo.
$ns_ node-config -adhocRouting $opt(adhocRouting) \
    -llType $opt(ll) \
    -macType $opt(mac) \
    -ifqType $opt(ifq) \
    -ifqLen $opt(ifqlen) \
    -antType $opt(ant) \
    -propType $opt(prop) \
    -phyType $opt(netif) \
    -energyModel "EnergyModel" \
    -initialEnergy 3.0 \
    -txPower 0.9 \
    -rxPower 0.5 \
    -idlePower 0.45 \
    -sleepPower 0.05 \
    -channelType $opt(chan) \
    -topoInstance $topo \
    -agentTrace ON \
    -movementTrace ON \
    -routerTrace ON \
    -macTrace ON

# Se crean los 10 nodos que intervienen en la simulación.
for {set i 0} {$i < $opt(nn) } {incr i} {
set node_($i) [$ns_ node]
}

# Se les indican a los 10 nodos sus posiciones iniciales.
for {set i 0} {$i < $opt(nn) } {incr i} {
    $node_($i) set X_ [expr rand()*500]
}

```

```

    $node_($i) set Y_ [expr rand()*500]
    $node_($i) set Z_ 0
}

#Se crean las conexiones UDP y agentes de aplicación CBR
# para que los 10 nodos establezcan comunicación entre ellos y
# se define el instante de tiempo en que debe comenzar el envío de paquetes.
for {set i 0} {$i < $opt(connections)} {incr i} {

    #Creación y configuración de la conexión UDP
    set udp_($i) [new Agent/UDP]
    $ns_ attach-agent $node_($i) $udp_($i)
    set null_($i) [new Agent/Null]
    $ns_ attach-agent $node_([expr $i+2]) $null_($i)
    $ns_ connect $udp_($i) $null_($i)

    #Creación de agentes, aplicación CBR sobre UDP
    set cbr_($i) [new Application/Traffic/CBR]
    $cbr_($i) attach-agent $udp_($i)
    $cbr_($i) set type_ CBR
    $cbr_($i) set packet_size_ 256
    $cbr_($i) set rate_ $opt(dataRate)
    $cbr_($i) set random_ false

    $ns_ at 0.05 "$cbr_($i) start" ;# se indica que comience al 0.05 segs.
}

# Asigna destinos aleatorios a los 10 nodos de la simulación.
for {set j 0} {$j < 10} {incr j} {
    for {set i 0} {$i < $opt(nn)} {incr i} {
        set xx_ [expr rand()*$opt(x)]
        set yy_ [expr rand()*$opt(y)]
        $ns_ at .01 "$node_($i) setdest $xx_ $yy_ 15.0"
    }
}

```

```
# Cuando se alcanza el tiempo limite de la simulación se indica a
# todos los nodos que la misma ha finalizado.
for {set i 0} {$i < $opt(nn) } {incr i} {
    $ns_ at $opt(stop) "$node_($i) reset";
}

# Cuando se alcanza el tiempo 10 segundos se ejecuta la función finish.
$ns_ at $opt(stop) "finish"

$ns_ run
```

4.3. Modelos de Movilidad

Diferentes modelos de movilidad han sido seleccionados para alcanzar mayor realismo en las simulaciones. Estos son implementados por el simulador NS2 (*Network Simulator version 2*) cuando se define el archivo *.tcl* a cada nodo se le asigna su trayectoria y el modelo de movilidad gobierna el movimiento de los nodos.

Como se mencionó anteriormente, el comportamiento de una MANET (*Mobile Ad hoc Networks*, Red móvil ad hoc) varía según el modelo de movilidad utilizado y las simulaciones se verán afectadas según el algoritmo escogido. Varios trabajos evalúan distintos modelos de movilidad típicos en la simulaciones de redes móviles ad hoc y demuestran la importancia a la hora de escoger uno [78].

Los modelos de movilidad más usados para las simulaciones de redes MANET son detallados a continuación.

4.3.1. Modelo Random Walk

El RW (*Random Walk*) es un modelo de movilidad muy sencillo pero ampliamente utilizado para realizar simulaciones en redes móviles ad hoc [7]. El nodo móvil se traslada hacia cualquiera de las posibles direcciones con la misma probabilidad independientemente de cual fuese el movimiento anterior. Se escoge una velocidad y dirección aleatoria y el nodo se mueve en línea recta. Cuando alcanza el límite de la zona de trabajo retoma el movimiento con un ángulo distinto al utilizado en el intento anterior. Lo que luego de un tiempo provoca que los dispositivos móviles se concentren en la zona central. El nodo deja de moverse al cabo de un tiempo o tras una distancia recorrida, según se defina.

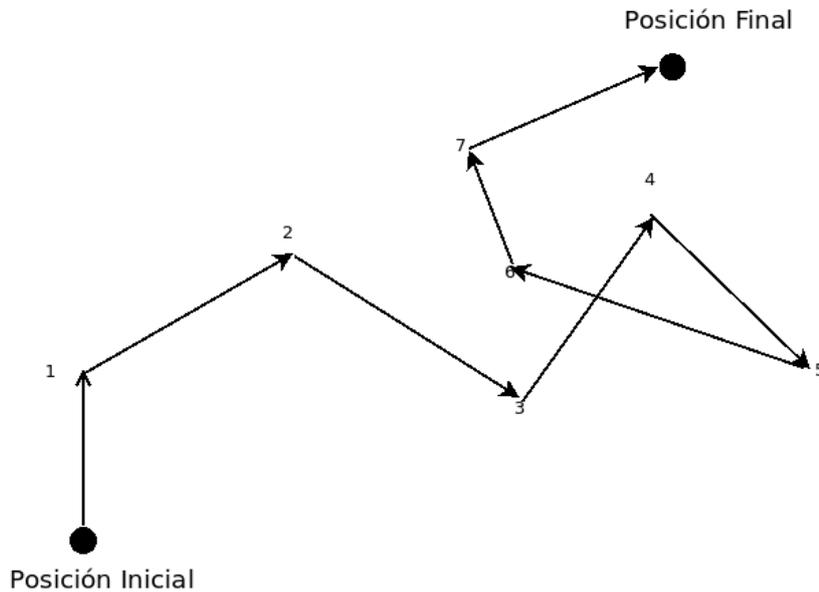


Figura 4.4 Ejemplo de una trayectoria de un nodo usando RW

En la Figura 4.4 se muestra un ejemplo de la trayectoria que sigue un nodo realizando 7 cambios de velocidad y dirección utilizando el modelo de movilidad RW.

4.3.2. Modelo Random Waypoint

RWP (*Random Waypoint*) es uno de los más utilizados en las simulaciones de redes móviles inalámbricas y es usado en muchos de los trabajos donde se evalúa el comportamiento de protocolos de enrutamiento [7] [32] [68]. Los nodos están distribuidos dentro del área de simulación, que luego de pasado un cierto período de tiempo en una posición determinada se escoge aleatoriamente un nuevo destino al que se dirigirán con una velocidad uniformemente distribuida dentro de un rango que define la máxima y mínima velocidad posible que podrán desarrollar los dispositivos móviles. En ese momento los nodos se mueven hasta los destinos particulares con la velocidad escogida. La diferencia con el modelo RW (*Random Walk*), descrito en el apartado 4.3.1, es que escoge un destino aleatorio y no una dirección. Este modelo incluye tiempos de pausa entre los cambios de dirección y/o velocidad.

En la Figura 4.5 se puede apreciar un ejemplo de una trayectoria que sigue un nodo utilizando el modelo de movilidad RWP.

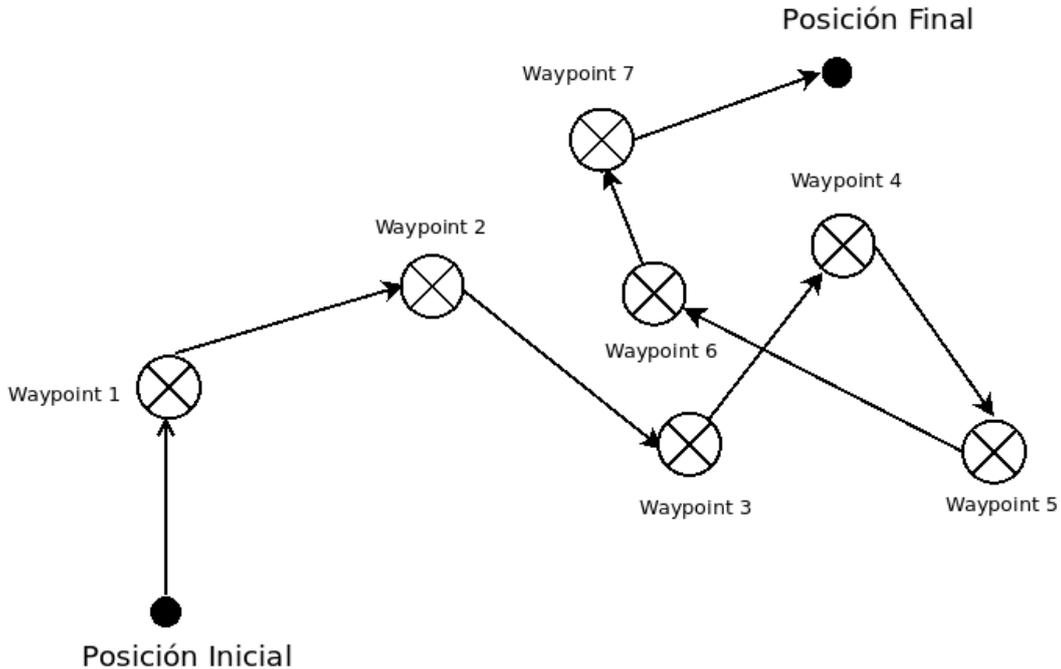


Figura 4.5 Ejemplo de una trayectoria de un nodo usando RWP

4.3.3. Modelo Random Direction (RD)

En RD (*Random Direction*), el nodo selecciona una dirección a la cual dirigirse al igual que en el modelo RW (*Random Walk*), descrito en el apartado 4.3.1, y se mueve hasta el borde del área de simulación siguiendo el rumbo elegido. Por lo que los nodos se trasladan por el límite del área de simulación antes de cambiar de dirección y velocidad. Una vez alcanzado el borde, el nodo realiza las siguientes acciones: deja de moverse durante un tiempo especificado, escoge otra dirección angular entre 0 y 180° y continúa trasladándose.

El modelo de movilidad RRD (*Realistic Random Direction*) [72], es una variación que no fuerza al nodo a viajar hasta el límite del área antes de parar para escoger nuevo rumbo, sino que puede hacerlo antes de alcanzar el borde del escenario de simulación.

En la Figura 4.6 se muestra un ejemplo de la trayectoria que sigue un nodo utilizando el modelo de movilidad RD.

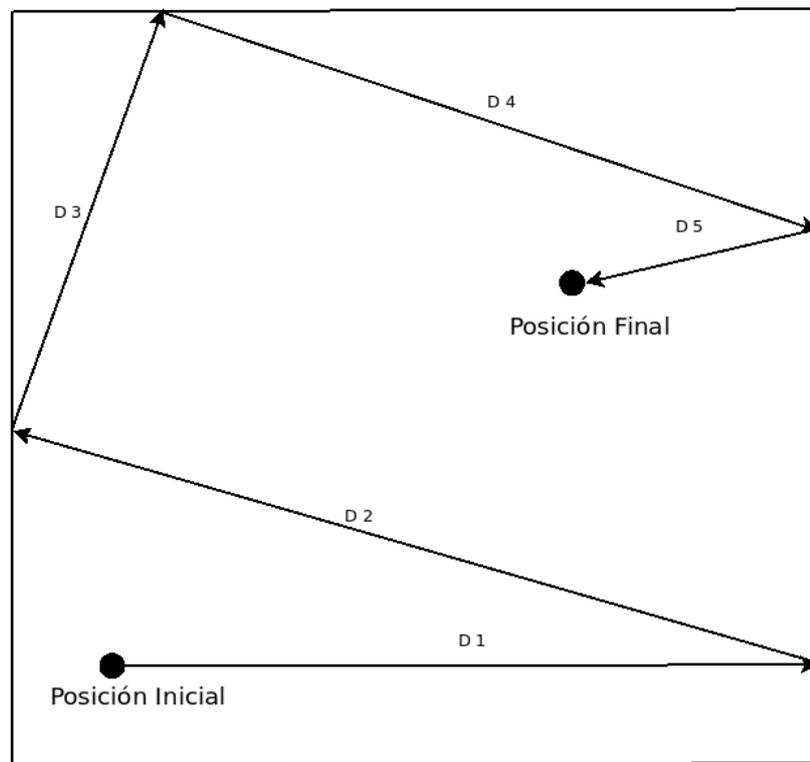


Figura 4.6 Ejemplo de una trayectoria de un nodo usando RD

En este trabajo únicamente se ha utilizado un único modelo de movilidad, en particular el RWP (*Random Waypoint*), con distintas variaciones de velocidad para estudiar como afectan al comportamiento del protocolo propuesto en esta tesis doctoral. La velocidad de los nodos utilizada en la mayoría de los experimentos presentados en este documento es de 1.38 m/s, un valor equivalente a 5 Km/h que corresponde a la velocidad de una persona caminando a un ritmo normal. Sin embargo en otros casos se ha utilizado un rango de velocidades más amplio (5, 10, 15 Km/h), aunque algunas de ellas son muy elevadas para considerarse que se trata de una persona moviéndose a pie, igualmente se han utilizado para evaluar como influye la velocidad en el comportamiento de PROCCDIMI.

4.3.4. Resumen de los modelos de movilidad

En la Tabla 4.2 se presentan los modelos de movilidad seleccionados y analizados en el apartado 4.3, junto con sus características más sobresalientes. Principalmente se especifica a que tipo pertenece, si utiliza waypoints o no, si se utiliza en entornos científicos o no y si fue seleccionado para las simulaciones.

Modelo	Tipo	Waypoints	Uso académico	Seleccionado
Random Walk (RW)	Aleatorio	No	Si	No
Random Waypoint (RWP)	Aleatorio	Si	Si	Si
Random Direction (RD)	Aleatorio	No	Si	No

Tabla 4.2 Resumen de los modelos de movilidad

4.4. Nodos

Se llevarán a cabo múltiples simulaciones a fin de validar la propuesta presentada en esta tesis doctoral, la cual toma la forma de un protocolo que descubre clientes basándose en una aproximación de diseño cross layer. Conformar clusters donde en ellos se diferencia los dispositivos con capacidades superiores de los dispositivos con capacidades limitadas. Y que establece como coordinadores de clusters a los dispositivos con capacidades superiores. Validar un protocolo con las características antes mencionadas implica tener en cuenta los tipos y la densidad de nodos presentes en los escenarios de simulación. Estos dos parámetros son abordados y detallados en la continuación.

4.4.1. Tipos de nodos

Como se mencionó anteriormente PROCCDIMI (PROtocolo para el Cálculo Científico Distribuido sobre clientes Móviles Indeterminados en MANETs) toma en cuenta las características heterogéneas de los nodos que conforman las redes móviles ad hoc y supone la existencia de dos tipos de dispositivos: dispositivos DCS (Dispositivo de capacidades superiores) y dispositivos DCL (Dispositivo de capacidades limitadas). Solamente los DCS serán designados como coordinadores, pero no todos ellos se desempeñaran como coordinadores de clusters.

El tipo de nodo se obtiene en base a una ponderación de que tan potentes son los recursos de hardware con los cuales cuenta el dispositivo que se esta inspeccionando. El resultado de este examen clasificará al nodo como DCS o DCL. Una vez establecido el tipo de dispositivo para todas y cada una de las unidades de la red, un subconjunto de los que resultan ser DCS se desempeñaran como coordinadores y tendrán como principal tarea el descubrimiento de nuevos clientes que pertenecerán al cluster que administran junto con la asignación de tareas de computo distribuido y recolección de resultados.

Como se comentó anteriormente, la clasificación de DCS la obtiene un nodo, a partir de la capacidad que éste tenga respecto a las prestaciones de hardware como con que

capacidad de memoria cuentan, duración de la batería, potencia de cómputo, entre otras. En esta investigación se trabaja con el supuesto que una unidad de red es consciente de las prestaciones que posee. De esta forma, a un dispositivo que participa de una red móvil ad hoc se le asigna un identificador para determinar a qué tipo de nodo pertenece y el cual admite solamente tres valores posibles:

- Si el identificador toma el valor 0 se trata de un DCL.
- Si el identificador toma el valor 1 se trata de un DCS.
- Si el identificador toma el valor 2, entonces el nodo es un coordinador de cluster.

Existen trabajos de investigación precedentes [81], donde se utilizan mecanismos similares para determinar las capacidades de un nodo. Algunos son estáticos tomando la forma de un identificador y otros son dinámicos [59] donde se utiliza una computación en la unidad de red a fin de determinar las capacidades de la misma. Esta última aproximación tiene la desventaja que involucra un consumo de energía considerable que va en detrimento de la duración de la batería impactando directamente sobre la autonomía del dispositivo.

4.4.2. Densidad de nodos

La densidad de nodos se compone de tres parámetros: el número total de nodos, área y rango de transmisión. En una red muy esparcida, el número de posibles conexiones entre cualquier par de nodos es muy limitado [25]. Para medir la densidad de nodos se usa la fórmula del grado de conectividad d la cual es:

$$d = \frac{(N \times \pi \times r^2)}{A}$$

Donde N representa el número total de nodos, r indica el rango de transmisión de los nodos y A el área de simulación. De este cálculo, surge que variar cualquiera de los parámetros, el rango de transmisión, área de simulación o el número de nodos, puede lograr el mismo efecto a fin de ajustar la densidad de nodos de la red.

Para el estudio que se propone en esta tesis doctoral, se decide variar el rango de transmisión en las simulaciones a fin de lograr una densidad de nodos adecuada [73].

4.5. Consideraciones del modelo de simulación

Probar la viabilidad de un protocolo con las características de PROCCDIMI implica tener en cuenta los siguientes criterios a la hora de realizar las simulaciones. Primero, es

necesario generar escenarios con nodos móviles heterogéneos altamente dinámicos, aplicar modelos de movilidad reales con patrones de movimientos aproximados al desplazamiento de una persona a pie. Segundo, tener presente que hipotéticamente en una MANET pueden existir DCS (Dispositivo de capacidades superiores) y DCL (Dispositivo de capacidades limitadas). Y tercero, una vez identificados los DCS establecer cuales de ellos officiarán como coordinadores.

Los parámetros que corresponden al número de nodos, número de clústers, número de coordinadores y movilidad son aplicados a simulaciones que plantean escenarios que pueden ubicarse en espacios abiertos como por ejemplo un campus universitarios o al desarrollo de un experimento de campo en un lugar remoto con un número acotado de científicos donde cada uno de ellos cuenta con un dispositivo móvil que participarán de un cálculo científico distribuido. En este último escenario hipotético, los dispositivos serán agrupados temporalmente y posiblemente coordinados por el dispositivo de un profesor o investigador de grado superior.

4.5.1. Topología

De acuerdo a D. Kotz [52], el diámetro de movilidad en un campus es menos que 50 m. Por simplicidad pero sin perder generalidad y en aras de la simulación, el coordinador estará ubicado en forma aleatoria dentro un área de simulación rectangular de 500 m x 500 m.

4.5.2. Tráfico

Para la generación de tráfico se encuentran disponibles cuatro posibles alternativas:

- Los generadores de tráfico CBR (*Constant Bit Rate*, Tasa de bits constante) que producen ráfagas de datos de tamaño fijo cada cierto intervalo fijo de tiempo.
- Los generadores de tráfico exponencial encendido/apagado que se caracterizan por tener un funcionamiento similar a los CBR durante los períodos donde se encuentra encendido y no genera ningún paquete de datos durante los períodos de apagado.
- Los generadores de tráfico Pareto encendido/apagado que trabajan de manera similar al exponencial encendido/apagado pero los períodos de encendido y apagado se realizan según lo que dicta la distribución de Pareto.
- Los generadores de tráfico de traza que realizan las ráfagas de paquetes de acuerdo a un archivo de traza dado.

Todas las simulaciones realizadas para validar la propuesta de esta tesis doctoral utilizaron un generador de tráfico CBR como fuente de datos. Éste hace un uso mas exhaustivo del ancho de banda de la red móvil y el sentido de esta elección es estresar la red hasta su límite y observar el rendimiento de PROCCDIMI cuando es sometidos a semejante prueba de esfuerzo.

4.6. Escenarios de simulación

Es muy importante probar y validar la viabilidad de llevar a cabo cómputo científico en MANET en escenarios reales. Por lo cual se realizarán diferentes simulaciones en escenarios diseñados con un alto grado de realismo. A continuación se presentan cinco tipos de escenarios:

- Escenario 1, con bajo número de nodos (10 nodos).
- Escenario 2, con un número medio bajo de nodos (20 nodos).
- Escenario 3, con un número medio de nodos (50 nodos).
- Escenario 4, con un número medio alto de nodos (100 nodos).
- Escenario 5, con gran número de nodos (150 nodos).

Los escenarios anteriores esta diseñados para probar si el protocolo cuenta con:

- La habilidad de asignar correctamente un cluster de pertenencia a un nuevo nodo que ingresa a la red.
- La habilidad de comenzar un nuevo calculo científico distribuido y posteriormente coleccionar los resultados del mismo.
- Mantener controlado la carga dentro de la red.
- Tener medidas de performance aceptables y un bajo consumo energético.
- Controlar correctamente los cambios en la topología de la red debido a la movilidad de los nodos.

4.7. Resumen

En este capítulo se presentaron y abordaron las soluciones más relevantes en el área de simulación de protocolos de enrutamiento que funcionan sobre MANET, teniéndose en cuenta: la adopción dentro de la comunidad académica, si se trata de un desarrollo open source o comercial y si admite la posibilidad de extender los protocolos de enrutamiento disponibles en la misma con nuevas implementaciones. Luego se estableció que la herramienta para llevar a cabo los estudios de rendimiento a realizarse en el capítulo 5 será NS2 (*Network Simulator version 2*). La arquitectura de NS2 y como esta se ajusta a la solución propuesta en esta tesis doctoral fue descrita en profundidad en el apartado 4.2.1. La sección 4.2 finaliza exhibiendo los formatos de las trazas que genera NS2 y analizando la estructura un script OTCL (*Object-oriented Tool Command Language*).

La sección 4.3 aborda los modelos de movilidad seleccionados para ser utilizado en las simulaciones. Cada uno de estos, son explorados exhaustivamente describiendo sus características de funcionamiento y como estos gobiernan el movimiento de los nodos.

En la sección 4.4 trata, en primera instancia, sobre la manera de representar la heterogeneidad de los nodos dentro de los escenarios de simulación, detallando que en estos últimos estarán presentes nodos de tipo DCL (Dispositivo de capacidades limitadas), DCS (Dispositivo de capacidades superiores) y coordinadores de cluster. Un vez concluido el tema anterior, se presenta el concepto de densidad de nodos, con el cual vía un cálculo matemático se busca asegurar la conectividad entre los dispositivos en las simulaciones, ya que en una red muy esparcida el número de posibles conexiones entre cualquier par de nodos es muy poco probable.

Las dos últimas secciones están dedicadas a presentar los componentes finales de los escenarios de simulación. La topología y el generador de tráfico son especificados. En el primer caso se detalla las dimensiones que tendrá el área de simulación y en el segundo caso se expone un listado de las posibilidades con que se cuenta en NS2, para luego definir cual se usará. El tema final del capítulo define los tipos de escenarios que se utilizarán, determinando que se contarán con cinco tipos distintos: escenarios con bajo, medio bajo, medio, medio alto y alto número de nodos.

Capítulo 5

Resultados experimentales

En el capítulo 4 se describió la metodología para validar la propuesta detallada en el capítulo 3 a través del uso de simulación. Se ha presentado la herramienta a utilizar, NS2 (*Network Simulator version 2*), conjuntamente con los elementos más importantes contenidos en los escenarios de simulación. Todo lo expuesto anteriormente esta relacionado con la preparación para el proceso de comprobación que se llevará a cabo en este apartado.

A lo largo del presente capítulo se tratarán todos los aspectos relacionados con la evaluación de rendimiento de PROCCDIMI (PROtocolo para el Cálculo Científico Distribuido sobre clientes Móviles Indeterminados en MANETs) especificando los diferentes valores en los parámetros; analizando en forma separada según el modo de operación cual es el comportamiento dependiendo del número de dispositivos, la movilidad, la cantidad de coordinadores, entre otras variables.

Las métricas elegidas servirán para comprobar si PROCCDIMI permite tener controlado el número de mensajes transmitidos en busca de un coordinador, si la tasa de entrega de paquetes es aceptable, si la sobrecarga en el enrutamiento que se emplea para descubrir coordinadores no es tan elevada, si la variación o fluctuación en el retardo de fin a fin entre el envío de dos paquetes es tolerable, si la cantidad de energía consumida por los nodos es suficiente baja como para creer que es viable el nuevo protocolo. Todo lo anterior se vuelve a ver buscando identificar si se mantiene razonablemente controlado cuando el protocolo cambia el modo de operación implicando esto último que además de realizar operaciones de descubrimiento comienza a enviar mensajes para asignar y recuperar los resultados de tareas previamente asignadas en un cómputo científico distribuido.

Dada la naturaleza de ser una propuesta novel, PROCCDIMI, no es posible trazar una comparativa justa debido a que se requiere disponer de al menos otro protocolo que tenga características similares, es decir, encontrar uno que realice las mismas tareas que PROCCDIMI sobre una MANET (*Mobile Ad hoc Networks*, Red móvil ad hoc) basándose en un diseño

cross layer, que su operación corresponda a la de un protocolo de encaminamiento reactivo como AODV (*Ad hoc On-demand Distance Vector*), que forme clusters de nodos, que distinga DCS (Dispositivo de capacidades superiores) de DCL (Dispositivo de capacidades limitadas) y que los DCS oficien de coordinadores de los clusters de la red para realizar las asignaciones de tareas distribuidas y recolecciones de resultados. De las soluciones disponibles actualmente ninguna cumple con todas las características de PROCCDIMI, por lo que no es posible realizar ninguna comparativa al respecto. Con lo cual estas últimas se realizarán contra los valores que arrojen los protocolos de enrutamiento más ampliamente utilizados en este tipo de redes; a la hora de realizar el análisis del desempeño esto último será tenido en cuenta.

Este capítulo está estructurado de la siguiente manera: En la sección 5.1 se describen los parámetros o métricas empleadas para evaluar el rendimiento del protocolo: La tasa de entrega de paquetes, la carga de enrutamiento normalizada, el Jitter, el Throughput, y el consumo energético promedio. A continuación, en la sección 5.2 se analiza los resultados obtenidos de las mediciones de las métricas antes mencionadas para los dos modos de operación.

5.1. Métricas de rendimiento

La evaluación del rendimiento, como así el diseño y el desarrollo de los protocolos de ruteo en MANET requieren que sean examinados meticulosamente bajo parámetros estandarizados dentro de la materia de estudio. Estos últimos son seleccionados de acuerdo a la IETF RFC 2701 [57], la cual recomienda el uso de un conjunto de métricas destinadas al estudio y evaluación de rendimiento de diferentes protocolos pensados para trabajar en redes MANET.

Cuando se evalúa la performance en los protocolos de ruteo o de otro tipo destinados a trabajar sobre redes móviles ad hoc es importante contrastarlos contra ciertos parámetros para dimensionar correctamente su rendimiento.

5.1.1. Tasa de Entrega de Paquetes

PDR (*Packet Delivery Ratio*, Tasa de Entrega de Paquetes) se define como el número de paquetes entregados al nodo destino sobre el total enviados por el nodo origen. Con lo cual esta tasa tomará valores entre 0 y 100 por ciento.

En esta interpretación, un paquete que no puede ser ruteado debido a que el nodo origen no conoce ninguna ruta hacia el destino es contado como un paquete enviado pero no entregado, con lo cual se penaliza al protocolo interpretando que la actualización de su conocimiento

de la topología de la red es no es óptima. Por lo tanto, una tasa menor a 100 significa que los paquetes se perdieron en el camino o que no se pudo encontrar ninguna ruta al destino. Claramente es deseable tener una alta proporción de entrega de paquetes en cualquier tipo de red y más en las redes wireless móviles [58]. Esta métrica dimensiona la confiabilidad del protocolo en cuanto a la transmisión de información [15].

5.1.2. Carga de enrutamiento normalizada

La carga de enrutamiento lleva cuenta de las cantidad de veces que el origen debió reenviar paquetes que corresponden al protocolo de enrutamiento durante el tiempo que tomó realizar la simulación. Estos paquetes reenviados pueden tomar la forma de mensajes de solicitud de establecimiento de ruta, respuestas o de control.

La carga de enrutamiento es una medida importante de la eficiencia respecto al uso de ancho de banda disponible y la escalabilidad del protocolo que se esta evaluando [15].

NRL (*Normalized Routing Load*, Carga de enrutamiento normalizada) es la tasa definida como el cociente entre la cantidad de paquetes con información sobre la cantidad de paquetes adicionales transmitidos para asegurar que un byte de datos se entrega con éxito en el destino. Esta métrica tiene en cuenta cada retransmisión de un paquete de control, contabilizándolo como un nuevo paquete.

Conforme esta medida se aproxima a la unidad, mejor es el desempeño, puesto que para entregar un paquete de datos sería preciso transmitir dicho paquete y no serían necesarios los mensajes de control. Lo anterior implica una óptima eficiencia en la entrega de paquetes de datos.

Una cualidad deseable en un protocolo, es que genere poco tráfico durante el proceso de enrutamiento y así mantener más ancho de banda disponible para los paquetes de datos.

5.1.3. Jitter

Jitter se define como la variación o fluctuación en el retardo de fin a fin entre el envío de dos paquetes. Los tiempos de arribo de paquetes serán tal cual los esperados si el Jitter es bajo.

Esta métrica reviste cierto grado de importancia porque afecta directamente la performance de toda la red, principalmente a los requisitos de *buffering* que debe tener para alcanzar un uso razonable de la misma. Además, un valor excesivo de Jitter puede provocar problemas de sincronización hasta la pérdida de paquetes debido al *overflow* o *underflow* del buffer de paquetes [33]. Con lo cual el Jitter es muy importante para el dimensionado de la performance de la red en su conjunto.

5.1.4. Throughput

Throughput es definida como la tasa en que se entregan todos los datos al destino. Es el total de todos los bits exitosamente entregados a destinos individuales y es expresado como bytes por segundo (byte/sec) o bits por segundo (bit/sec). También puede verse como el tiempo que le toma al destino recibir el último mensaje con información [61]. Por consiguiente, es una medida de efectividad de un protocolo.

5.1.5. Costo energético promedio

El costo energético promedio dimensiona el consumo de energía promedio de los nodos que conforman la red al utilizar el protocolo que se esta evaluando durante la simulación. Este valor se calcula como la suma de las diferencias registradas entre energía inicial del nodo menos la energía que posee el mismo luego de terminada la simulación dividida la cantidad de nodos que componen la red.

La energía es un recurso escaso y un mal manejo de la misma por parte de un protocolo podría desencadenar el agotamiento de la batería de nodos críticos en términos de conectividad y afectar la disponibilidad de la misma, es decir, dejando unidades de red aisladas debido a que las que proveían conectividad de encuentra *off line* debido a que su batería no posee carga. Por lo tanto, es de suma importancia utilizar la energía de manera eficiente [65].

5.2. Análisis del protocolo PROCCDIMI

En esta sección, se analizaron los resultados obtenidos de realizar las pruebas de rendimiento presentándose en forma de gráficos que han sido generados con los datos de las simulaciones realizadas con NS2. Además se diseñaron cinco escenarios con 10, 20, 50, 100 y 150 nodos en cada uno de ellos; y se evaluó a PROCCDIMI en sus dos modos de operación contrastándolo con los protocolos AODV (*Ad hoc On-demand Distance Vector*), DSR (*Dynamic Source Routing*) y DSDV (*Destination Sequenced Distance Vector*) bajo los parámetros descritos en la sección 5.1.

Los tres protocolos de enrutamiento previamente nombrados solamente se remiten al encaminamiento de paquetes sobre las redes MANET y no realizan ninguna operación adicional a esta última. Mientras que PROCCDIMI resuelve el ruteo junto con todas las tareas involucradas en la gestión del cómputo científico distribuido y la administración de los clientes que resuelven estos. Dada esta disparidad de responsabilidades y a pesar de la misma, las pruebas de rendimiento se llevaron a cabo utilizando los protocolos tradicionales, ya que en los escenarios habituales lo único común que puede haber son justamente los

No. de nodos	Nro. de clusters	Nro. de coordinadores	Nro. de DCS's	Nro. de DCL's
10	2	2	6	4
20	4	4	12	8
50	10	20	30	20
100	20	60	60	40
150	50	100	100	50

Tabla 5.1 Números de nodos, clusters y coordinadores

protocolos de ruteo, debido a que el resto es muy dependiente de cada proceso, de cada forma de construcción, de cada configuración y hasta de cada plataforma, con lo cual es imposible poder realizar una evaluación en los mismos términos. Al no disponerse de un mecanismo, aplicación o herramienta unificadora comparable a la que en esta tesis se propone, que resuelva tanto el ruteo de paquetes como la indeterminación de los clientes de la red MANET más las gestión simultánea la asignación de tareas y recolección de resultados, entonces se prefirió utilizar los protocolos de enrutamiento establecidos para realizar la comparaciones con PROCCDIMI donde se tiene presente siempre que este último realiza operaciones adicionales además del enrutamiento de paquetes y evaluándose la performance incluyendo tareas propias del ruteo y lo antes mencionado. Pese a estas diferencias de procesos involucrados los resultados analizados son similares en cuanto a las métricas típicas de la operación de los protocolos de red refiere y en lo concerniente a la métrica del consumo de energía, PROCCDIMI que exhibe un comportamiento superior al resto de los protocolos de enrutamiento con los que se lo contrastó.

5.2.1. Parámetros de la simulación

Las extensas y numerosas simulaciones llevadas a cabo para la validación de la propuesta que se plantea en esta tesis se realizaron sobre el simulador NS2 (*Network Simulator version 2*) como se detalló anteriormente conjuntamente con los detalles del mismo. Los parámetros utilizados para la confección de los escenarios de simulación son detallados en las tablas: Tabla 5.1 y Tabla 5.2 con sus nombres y valores respectivos.

Los nodos son ubicados aleatoriamente sobre un área rectangular de 500 x 500 metros con una duración de observación máxima de 200 segundos. Los nodos desarrollarán distintos valores de velocidad (5, 10, 15 y 20 Km/h) y poseerán con un rango de comunicación de 100 metros usando la norma IEEE 802.11b como tecnología de transmisión inalámbrica. La energía inicial será de 100 Julios y por cada intento de comunicación realizado se irá

Parámetro	Valor
Nro. de dispositivos	10, 20, 50, 100, 150
Nro. de dispositivos DCSs	6, 12, 30, 60, 100
Nro. de dispositivos DCLs	4, 8, 20, 40, 50
Nro. de clusters	2, 4, 10, 20, 50
Coordinadores por cluster	2, 4, 20, 60, 100
Tiempo de simulación	200 segundos
Área total de simulación	500 x 500 mts
Modelo de movilidad	RWP
Protocolos	PROCCDIMI, AODV, DSR, DSDV
Protocolo MAC	IEEE 802.11b
Rango de transmisión (Tx)	100 mts.
Velocidad	5, 10, 15 Km/h
Pausas	Ninguna
Energía inicial	100.0 Joules

Tabla 5.2 Parámetros de simulación

consumiendo la carga de la batería. Se emplea el modelo de movilidad RWP (*Random Waypoint*).

La distribución de los coordinadores dentro del área de simulación es inicialmente aleatorias. Luego comenzarán a desplazarse hacia un destino fijo dentro del área y este desplazamiento es libre, aleatorio y sin pausas.

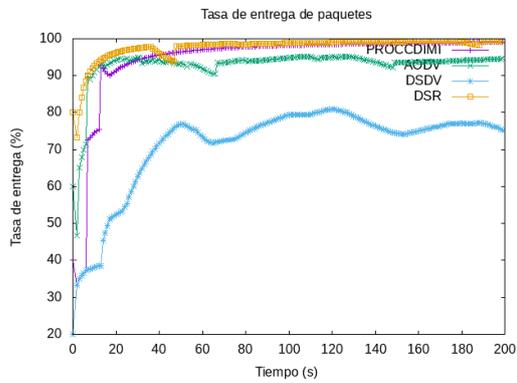
5.2.2. Modo descubrimiento de coordinador de cluster

Para averiguar la eficiencia de PROCCDIMI, es necesario evaluarlo separadamente en cada uno de sus modos de operación, cuando esta descubriendo nuevos dispositivos y asociándolos a un cluster y cuando esta asignando tareas a ser resueltas distribuidamente entre los clientes asociados a cada coordinador.

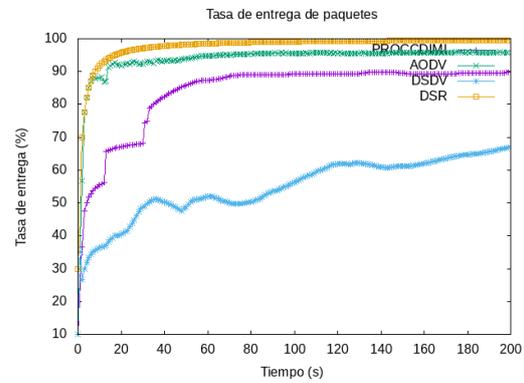
En este apartado se empleará todos los parámetros mencionados en las tablas: Tabla 5.1 y Tabla 5.2 para llevar acabo las simulaciones sobre NS2 para luego realizar el análisis de rendimiento con las métricas oportunamente seleccionadas.

Tasa de Entrega de Paquetes

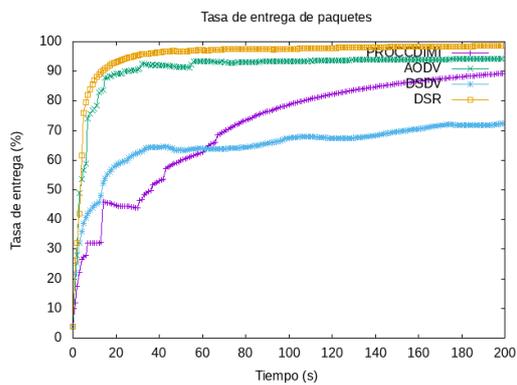
PDR (*Packet Delivery Ratio*, Tasa de Entrega de Paquetes) se calcula como el cociente entre la cantidad de paquetes que llegaron a destinos y la cantidad total de paquetes generados.



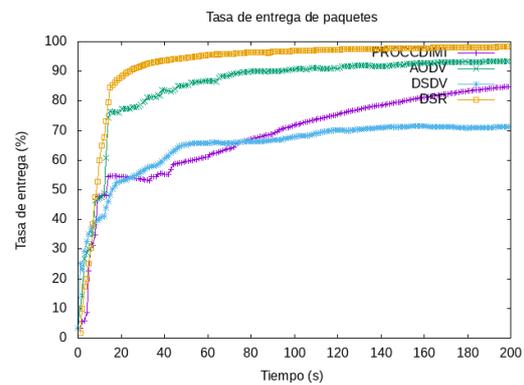
(a) Experimento realizado con 10 nodos



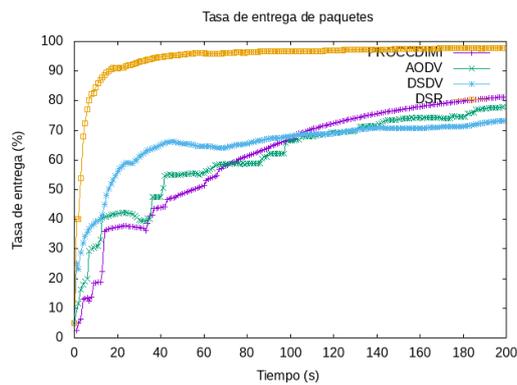
(b) Experimento realizado con 20 nodos



(c) Experimento realizado con 50 nodos



(d) Experimento realizado con 100 nodos



(e) Experimento realizado con 150 nodos

Figura 5.1 Tasa de entrega de paquetes (PDR) en función del tiempo y la cantidad de nodos. En modo solicitud de coordinador de cluster.

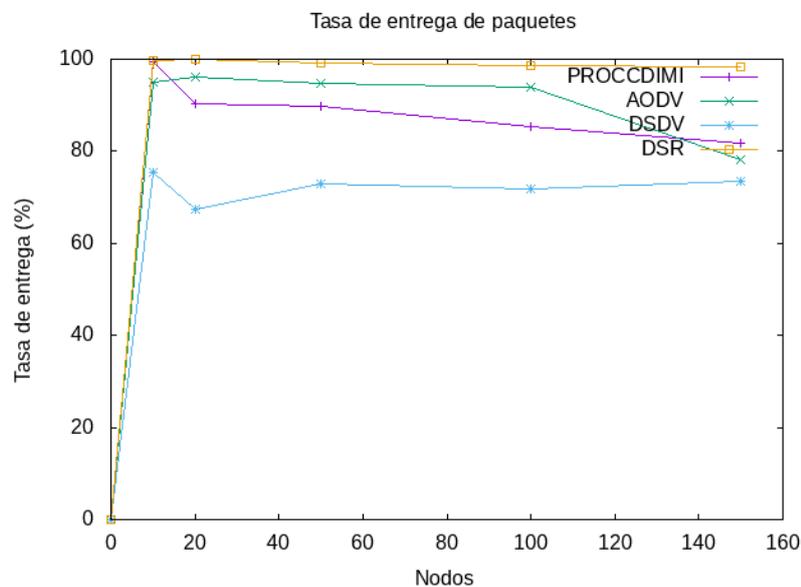


Figura 5.2 Tasa promedio de entrega de paquetes (PDR) en función de la la cantidad de nodos. En modo solicitud de coordinador de cluster.

Esta tasa representa el porcentaje de paquetes exitosamente entregados y dimensionando de esta manera la fiabilidad del protocolo.

En las simulaciones realizadas destinadas al análisis del rendimiento de PROCDDIMI en modo de descubrimiento de nuevos clientes con distintas cantidades de nodos, partiendo de los escenarios descritos en la sección 5.2.1, se puede apreciar que la velocidad de desplazamiento que desarrollan los dispositivos varía entre 5 y 15 Km/h, además de que los protocolos de ruteo contra los cuales se contrasta PROCDDIMI son AODV, DSR y DSDV. Todo lo anterior, junto a la cantidad de nodos utilizados en las simulaciones y el tiempo de simulación tienen el objetivo de dimensionar como varía el PDR.

Se observa claramente en las figuras: Figura 5.1 y Figura 5.2 la influencia del tiempo y del número de nodos sobre la tasa de entrega de paquetes. Conforme aumenta el número de nodos el valor de la tasa de entrega de paquetes se ve afectado significativamente. Aunque las curvas se solapan en algunos puntos, esto se ve claramente en la imagen 5.1b en el instante 190, y aumentan progresivamente hasta conseguir el último valor en el instante 200. La diferencia entre las curvas muestra que conforme evoluciona el tiempo de operación de la red y el número de nodos, se va desacelerando a su vez el crecimiento de las curvas, el PDR se va estabilizando. La explicación es la siguiente, mientras más tiempo este operando la red existe mayor posibilidad de que los enlaces puedan romperse, ocasionando un mantenimiento de rutas más frecuente.

En las figuras antes mencionadas, las curvas presentan diferencias significativas. Si bien las curvas tienen tendencia a describir una trayectoria ascendente en la Figura 5.1 y una descendente en la Figura 5.2, PROCCDIMI muestra una tasa de entrega de paquetes notoriamente superior cuando se trata de redes con una cantidad de nodos que oscila entre 10 y 20 nodos y por ende convirtiéndolo en un protocolo más confiable respecto a los demás analizados. Cuando se analiza PROCCDIMI en redes con un número de nodo superior a 20 nodos exhibe un PDR superior al %80 convirtiéndolo en un protocolo que otorga un nivel de confiabilidad que rivaliza con los otros protocolos con cuales se lo contrastó.

Carga de enrutamiento normalizada

Como ya se mencionó, la carga de enrutamiento normalizada es la tasa de bytes adicionales transmitidos, para asegurar que un byte de datos sea entregado con éxito en el destino y a medida que esta se aproxima a la unidad, mejor es el desempeño.

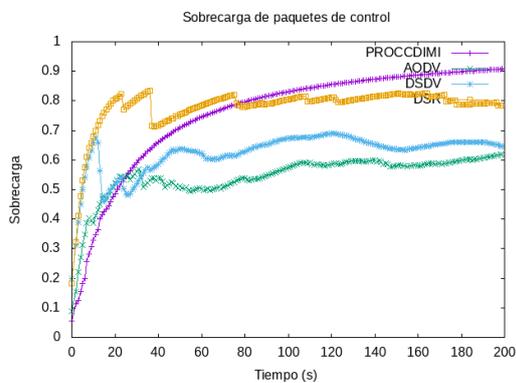
Los resultados que se pueden apreciar en las figuras: Figura 5.3 y Figura 5.4 indican que cada protocolo desarrolla una evolución singular y distintiva en sus curvas de desempeño. Todos ellos muestran una mejor respuesta en las curvas que se corresponden a las simulaciones realizadas con números de nodos relativamente pequeños, 10 y 20, puesto que en todos los casos se registran los valores más cercanos a la unidad de todas las simulaciones realizadas. Para redes más grandes en términos de número de nodos, los rendimientos se tornan menos eficientes, aunque PROCCDIMI se mantiene en el rango medio de desempeño, no siendo el que registra ni la mejor, ni la peor carga de enrutamiento.

Hay que tener siempre en cuenta que el protocolo aquí propuesto además de resolver cuestiones de ruteo de paquetes como los demás otros protocolos contra los que se lo compara, resuelve el descubrimiento de nuevos clientes y asignación de clusters de pertenecía, todo lo anterior implica el envío de paquetes de control. Por lo tanto que se mantenga en valores de NRL (*Normalized Routing Load*, Carga de enrutamiento normalizada) similares a los otros protocolos que solo se encargan de enrutar paquetes es realmente significativo.

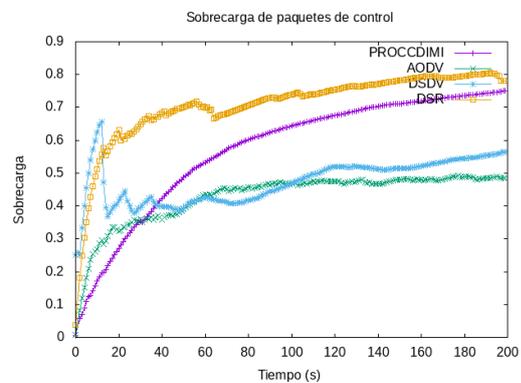
Jitter

Jitter se define como la variación o fluctuación en el retardo de fin a fin entre el envío de dos paquetes. Los tiempos de arribo de paquetes serán tal cual los esperados si le Jittes es bajo.

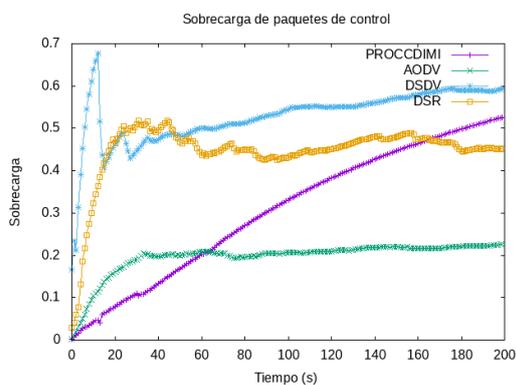
Esta métrica reviste gran importancia porque afecta directamente la performance de toda la red, principalmente los requisitos de *buffering* que debe cumplir para alcanzar un uso razonable de la misma. Además, un valor excesivo de Jitter puede provocar problemas de



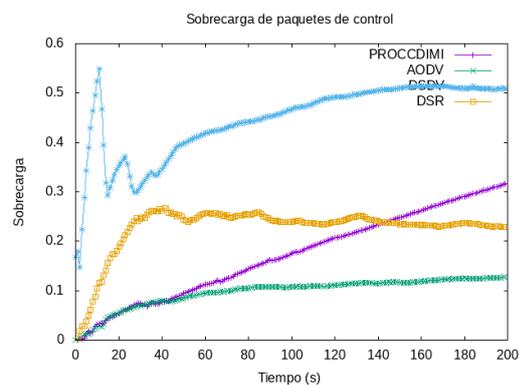
(a) Experimento realizado con 10 nodos



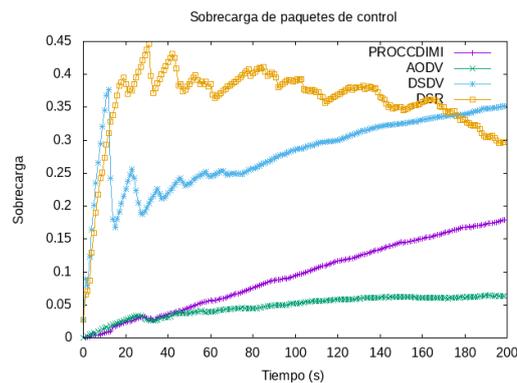
(b) Experimento realizado con 20 nodos



(c) Experimento realizado con 50 nodos



(d) Experimento realizado con 100 nodos



(e) Experimento realizado con 150 nodos

Figura 5.3 Carga de enrutamiento normalizada en función del tiempo y la cantidad de nodos. En modo solicitud de coordinador de cluster.

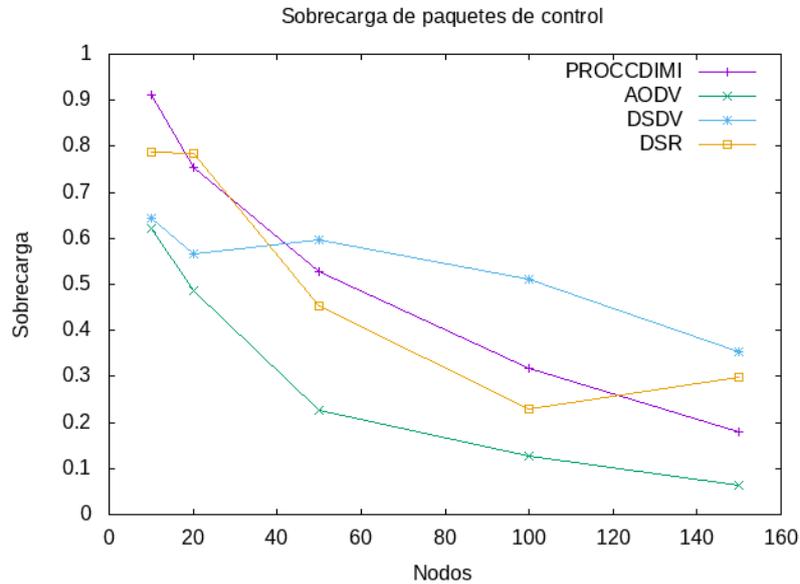


Figura 5.4 Carga de enrutamiento normalizada en función de la la cantidad de nodos. En modo solicitud de coordinador de cluster.

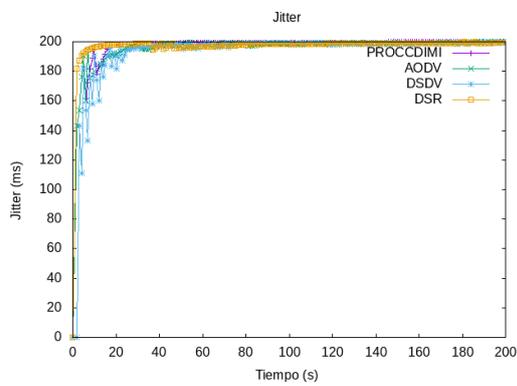
sincronización hasta la pérdida de paquetes debido al *overflow* o *underflow* del buffer de paquetes [33]. Con lo cual esta métrica es importante para dimensionar la performance de la red en su conjunto.

En la Figura 5.5 se puede apreciar que existen diferencias significativas entre las curvas de desempeño solamente al comienzo y que luego de transcurrido 60 segundos de simulación todas ellas tienden a estabilizarse y solaparse unas con otras. Este solapamiento se ve claramente en la Figura 5.6 donde todas las curvas describen la misma evolución. Para el caso particular de PROCCDIMI, este muestra un desempeño similar al resto de los protocolos evaluados.

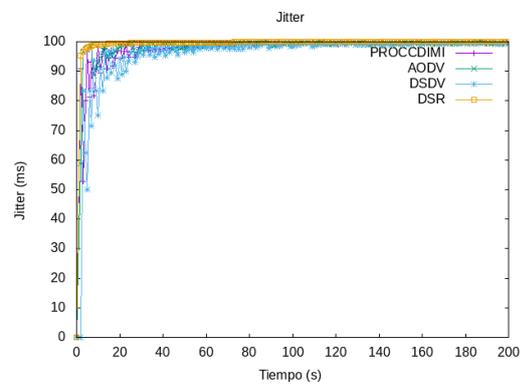
Throughput

Throughput es la tasa en que se entregan todos los datos al destino. Es el total de todos los bits exitosamente entregados a los respectivos destinos individuales y es expresado como bytes por segundo (byte/sec) o bits (bit/sec) por segundo.

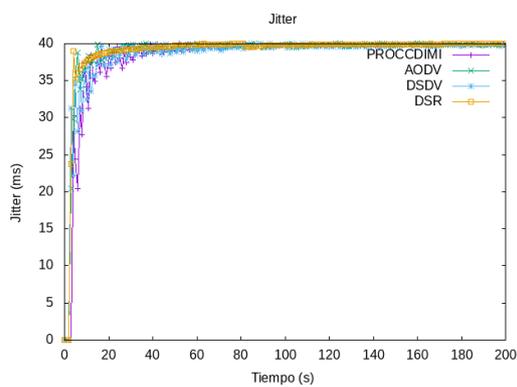
Los resultados que se pueden apreciar en las figuras: Figura 5.7 y Figura 5.8 indican que no hay diferencias significativas entre las curvas de desempeño, sobre todo en las curvas que se corresponden a las simulaciones realizadas con números de nodos relativamente pequeños 10 y 20, puesto que la velocidad de transmisión de todos los protocolos ronda los 10kbps en el primer casos y 20kbps en el segundo, con la salvedad de DSDV que esta cercano a



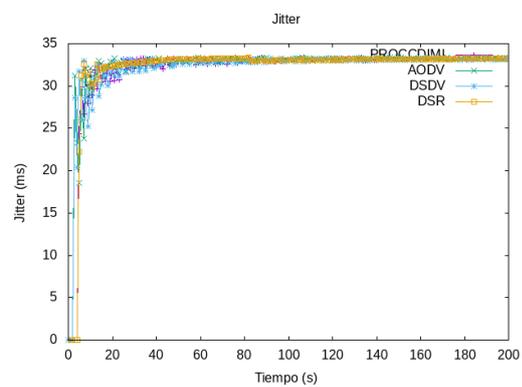
(a) Experimento realizado con 10 nodos



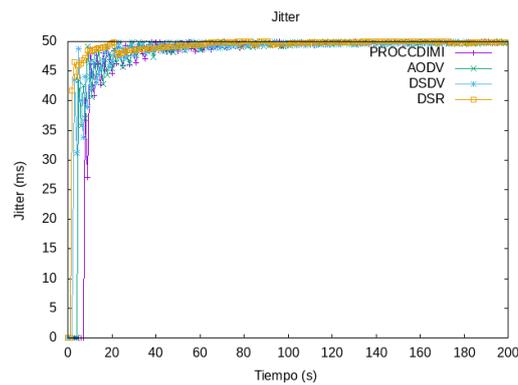
(b) Experimento realizado con 20 nodos



(c) Experimento realizado con 50 nodos



(d) Experimento realizado con 100 nodos



(e) Experimento realizado con 150 nodos

Figura 5.5 Jitter en función del tiempo y la cantidad de nodos. En modo solicitud de coordinador de cluster.

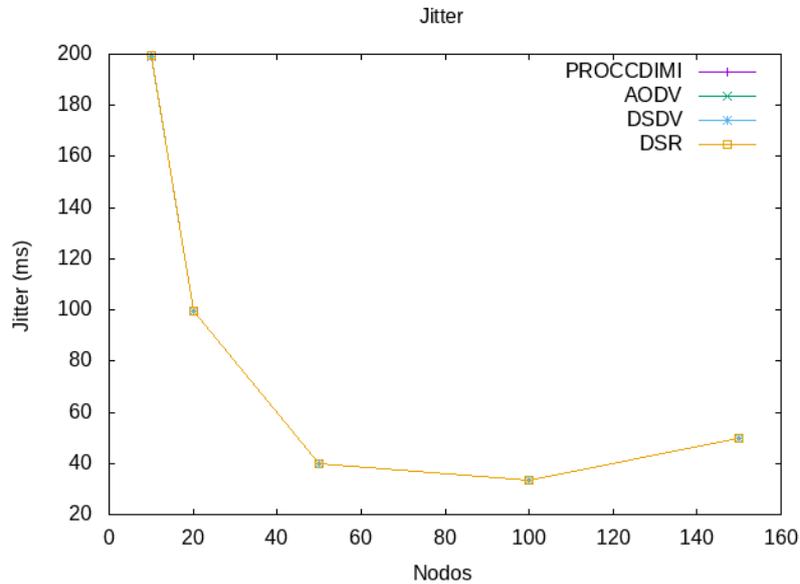
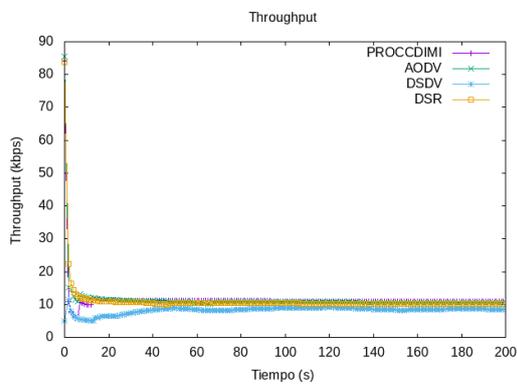


Figura 5.6 Jitter en función de la la cantidad de nodos. En modo solicitud de coordinador de cluster.

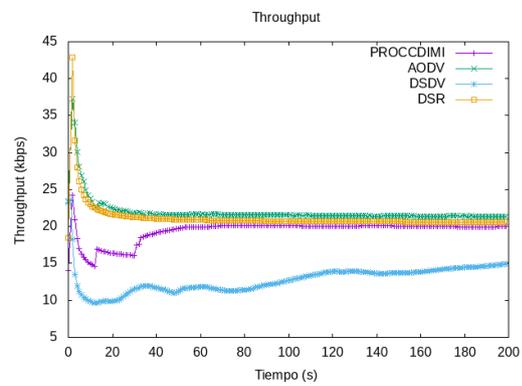
los 15kbps. Para redes más grandes en términos del número de nodos, los rendimientos se tornan más heterogéneos aunque PROCCDIMI se mantiene en el rango medio de desempeño, no siendo el que registra ni la mejor, ni la peor velocidad de transmisión. Si se tiene en cuenta que el protocolo aquí propuesto además de resolver cuestiones de enrutamiento de paquetes como los demás protocolos contra los que se lo compara, resuelve el descubrimiento de nuevos clientes y asignación de clusters de pertenecía, que se mantenga en velocidades de entrega similares a los otros protocolos que solo se encargan de enrutar paquetes es realmente significativo. Este aumento en la tasa de Throughput se explica por el diseño de clustering de PROCCDIMI que otorga una ventaja sobre la capa de acceso al medio ayudando a incrementar la rapidez de la conexión y por tanto, a conseguir mayor tasa de transmisión de datos [23].

Costo energético promedio

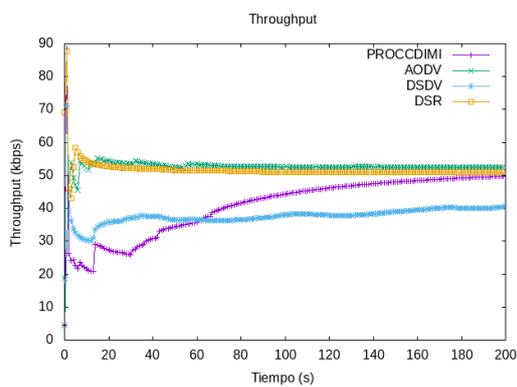
En una MANET (*Mobile Ad hoc Networks*, Red móvil ad hoc) el costo energético promedio es uno de los valores más complicados de calcular y son muchos los factores que intervienen en el mismo. Estos factores pueden estar relacionados a diferentes actividades de la red como: transmisión y recepción de mensajes, a factores relacionados con la manipulación de los protocolos de red, donde el protocolo MAC (*Media Access Control*, Control de acceso



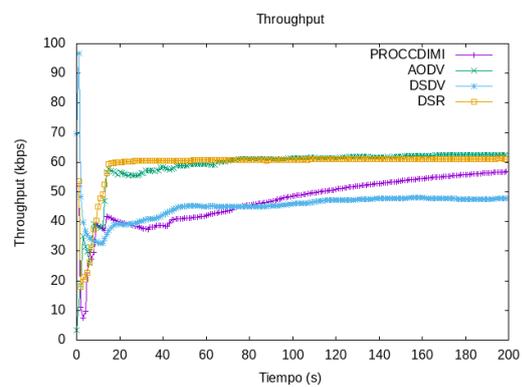
(a) Experimento realizado con 10 nodos



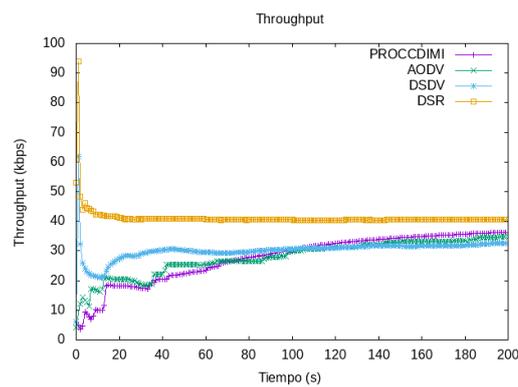
(b) Experimento realizado con 20 nodos



(c) Experimento realizado con 50 nodos



(d) Experimento realizado con 100 nodos



(e) Experimento realizado con 150 nodos

Figura 5.7 Throughput en función del tiempo y la cantidad de nodos. En modo solicitud de coordinador de cluster.

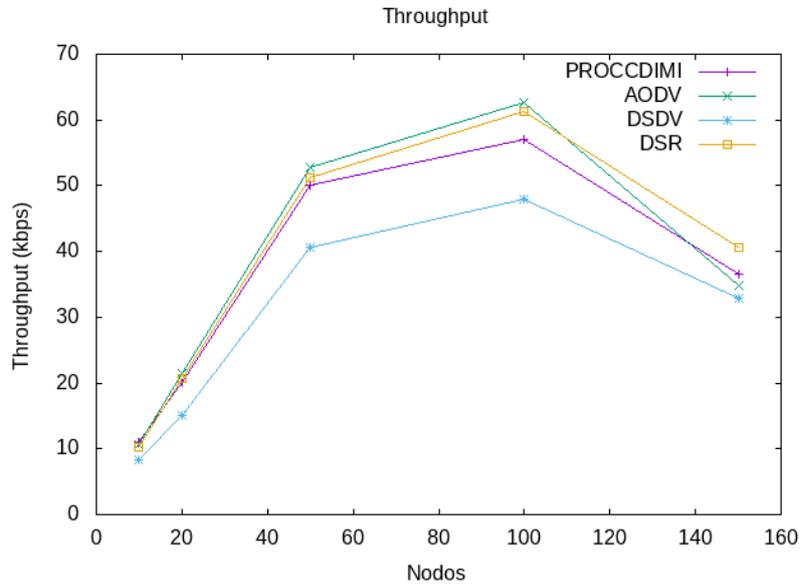
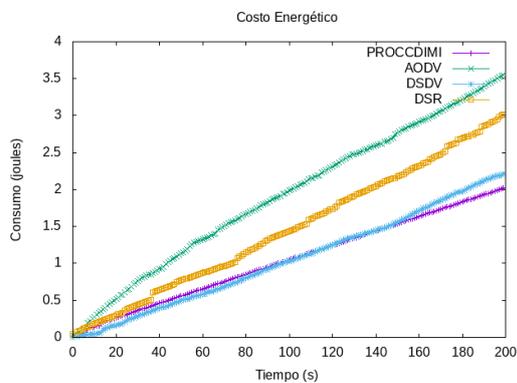


Figura 5.8 Throughput en función de la la cantidad de nodos. En modo solicitud de coordinador de cluster.

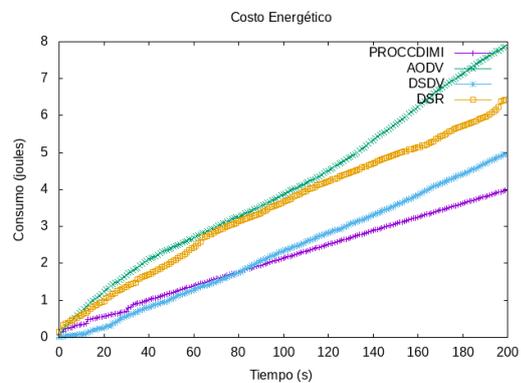
al medio) es uno de los más relevantes, y a factores dinámicos relacionados con la carga de tráfico [65].

En este estudio se ha evaluado el consumo de energía con el objetivo de conocer en qué medida PROCCDIMI minimiza el consumo de este recurso contrastandolo contra los protocolos de enrutamiento AODV, DSR y DSDV. A tal efecto, se han realizado dos mediciones: La primer medida toma el promedio de la energía consumida en función del tiempo, mientras que la segunda medida toma la media de la energía consumida en función del número de nodos.

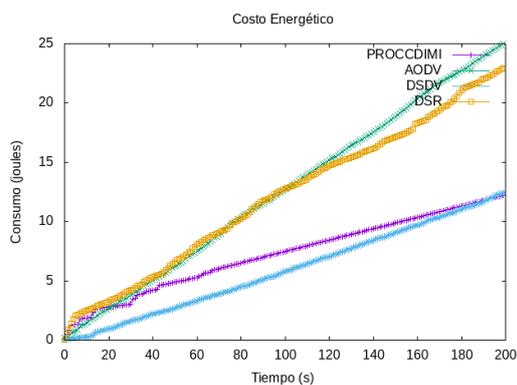
Se muestran en las figuras: Figura 5.9 y Figura 5.10 los detalles de la medición correspondiente a consumo energético vs. tiempo y a consumo energético vs. número de nodos. Se utiliza esta métrica para cuantificar el consumo de energía y de esta forma, conocer cuál de los protocolos optimiza mejor este recurso. Se puede apreciar que la curva correspondiente a PROCCDIMI asciende por debajo de las demás curvas, lo que indica que este protocolo consume menos energía que el resto, salvo en el caso cuando el tiempo de operación es bajo, donde la curva asciende por encima de las demás. El hecho de que PROCCDIMI tenga menor gasto energético, es producto de la estrategia de agrupamiento de los nodos, dado que uno de los beneficios que se esperan al formar clusters, es la reducción del consumo de energía [65].



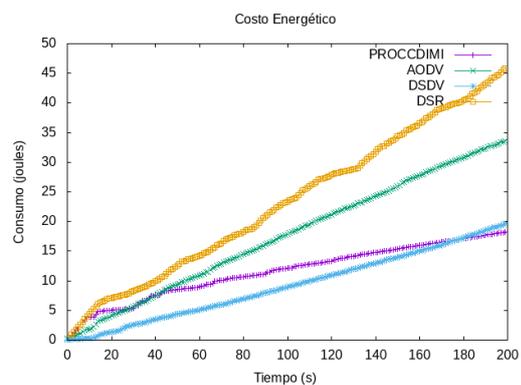
(a) Experimento realizado con 10 nodos



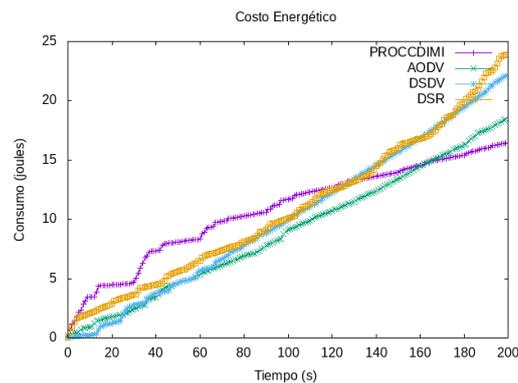
(b) Experimento realizado con 20 nodos



(c) Experimento realizado con 50 nodos



(d) Experimento realizado con 100 nodos



(e) Experimento realizado con 150 nodos

Figura 5.9 Costo energético promedio en función del tiempo y la cantidad de nodos. En modo solicitud de coordinador de cluster.

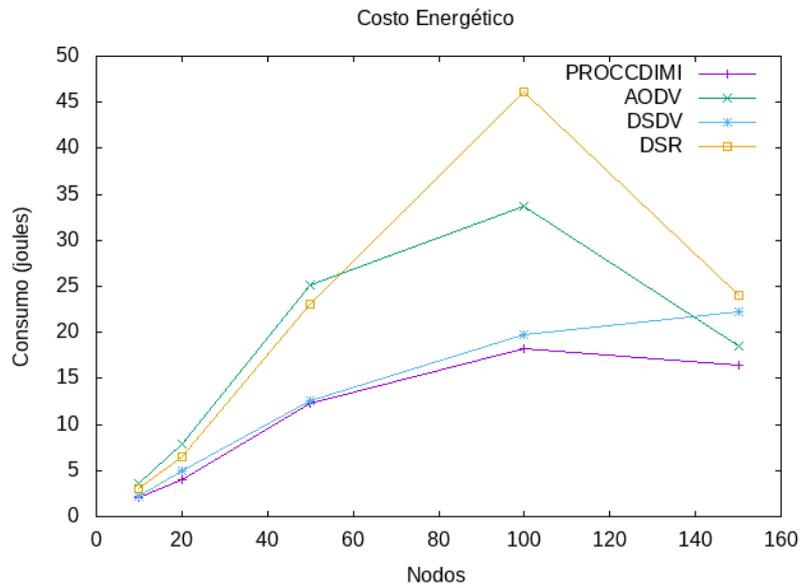


Figura 5.10 Costo energético promedio en función de la la cantidad de nodos. En modo solicitud de coordinador de cluster.

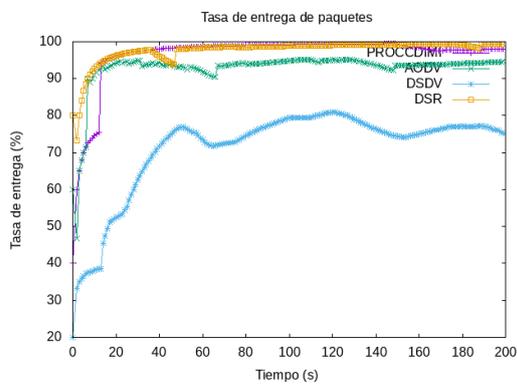
5.2.3. Modo asignación de tareas

Como se comentó anteriormente, para averiguar la eficiencia del protocolo PROCCDIMI (PROtocolo para el Cálculo Científico Distribuido sobre clientes Móviles Indeterminados en MANETs) es necesario evaluarlo separadamente en cada uno de sus modos de operación. El modo de operación donde se descubre nuevos dispositivos y se los asocia a un cluster, fue evaluado en la sección 5.2.2. Solo restando analizar el rendimiento del modo de operación donde se asigna tareas a ser resueltas distribuidamente entre los clientes seleccionados por el coordinador de cluster.

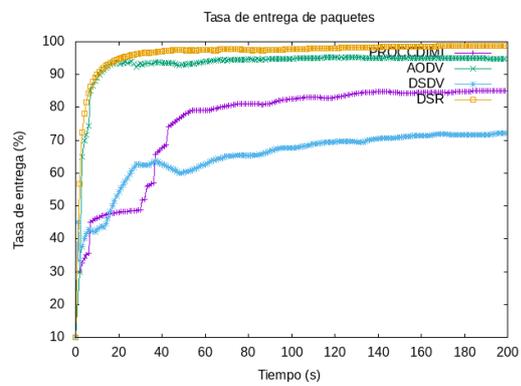
Siguiendo la misma metodología aplicada en la sección 5.2.2 en este apartado se emplearán todos los parámetros mencionados en las tablas: Tabla 5.1 y Tabla 5.2 para llevar a cabo las simulaciones sobre NS2 (*Network Simulator version 2*) para luego realizar el análisis de rendimiento con las métricas oportunamente descritas en la sección 5.1.

Tasa de Entrega de Paquetes

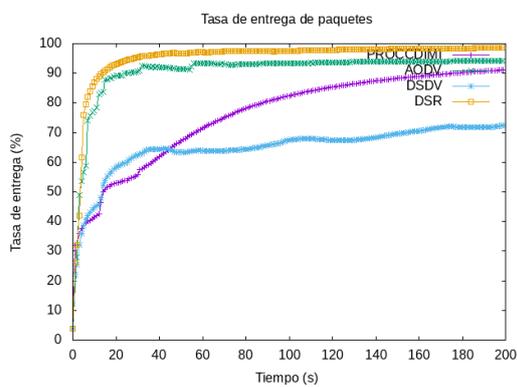
Se observa claramente en las figuras: Figura 5.11 y Figura 5.12 la influencia que en el tiempo y el número de nodos presentan sobre la métrica aquí tratada. Conforme aumenta el número de nodos el valor de la tasa de entrega de paquetes se ve afectado significativamente. Aunque las curvas se solapan en algunos puntos, la diferencia entre las curvas muestra que a medida que evoluciona el tiempo de operación y el número de nodos, va desacelerando a su



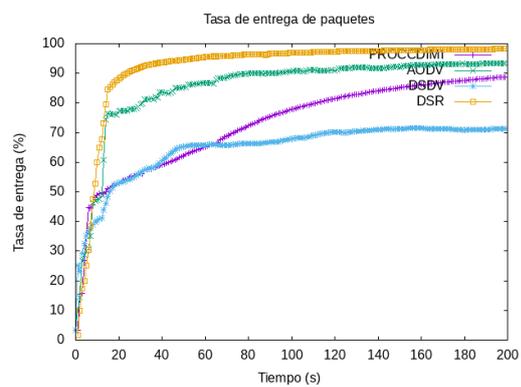
(a) Experimento realizado con 10 nodos



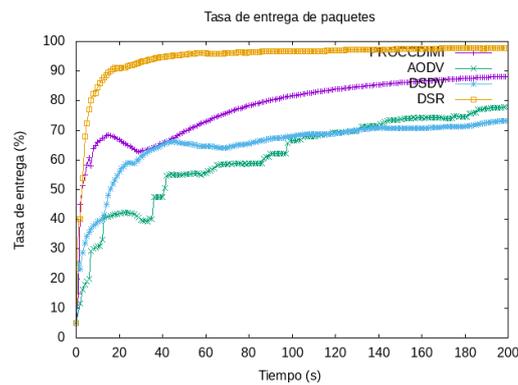
(b) Experimento realizado con 20 nodos



(c) Experimento realizado con 50 nodos



(d) Experimento realizado con 100 nodos



(e) Experimento realizado con 150 nodos

Figura 5.11 Tasa de entrega de paquetes (PDR) en función del tiempo y la cantidad de nodos. En modo asignación de tareas.

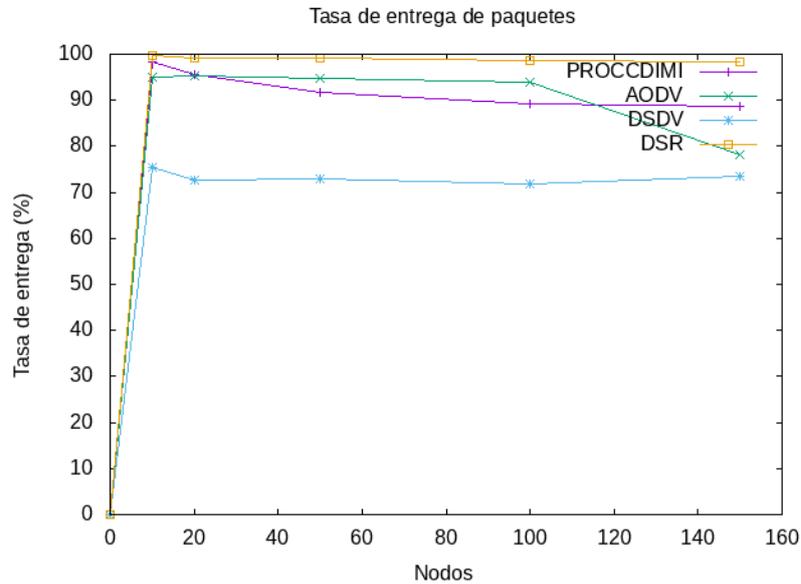


Figura 5.12 Tasa promedio de entrega de paquetes (PDR) en función de la la cantidad de nodos. En modo asignación de tareas.

vez el crecimiento de las curvas, el PDR (*Packet Delivery Ratio*, Tasa de Entrega de Paquetes) se va estabilizando. La explicación es la siguiente, a medida que pasa el tiempo las rutas que se conformaron en algún instante previo y de forma dinámica, por efecto de la movilidad de los nodos se incrementa la posibilidad de que estas puedan destruirse, ocasionando un mantenimiento de rutas más frecuente.

En las figuras antes mencionadas, las curvas presentan diferencias significativas. Si bien las curvas tienen tendencia a describir una trayectoria ascendente en la Figura 5.11 y una descendente en la Figura 5.12, PROCCDIMI muestra una tasa de entrega de paquetes notoriamente superior al 85% y por ende convirtiéndolo en un protocolo más confiable respecto de, por ejemplo, DSDV y AODV.

Al realizar un análisis en términos más generales, PROCCDIMI exhibe un PDR comparable al resto de los protocolos contra los cuales se lo contrasta y por ende un nivel de confiabilidad similar.

Carga de enrutamiento normalizada

Los resultados que se pueden apreciar en las figuras: Figura 5.13 y Figura 5.14 indican que cada protocolo muestra una evolución singular y distintiva en sus curvas de desempeño. Todos ellos muestran una mejor respuesta en las curvas que se corresponden a las simulaciones realizadas con números de nodos relativamente pequeños, 10 y 20, puesto que en todos los

casos se registran los valores más cercanos a la unidad, del mismo modo que ocurría en el análisis realizado en la sección 5.2.2.

Para redes más grandes en términos de número de nodos, los rendimientos se tornan menos óptimos, aunque PROCCDIMI se mantiene en el rango medio de desempeño, no siendo el que registra ni el mejor, ni el peor NRL (*Normalized Routing Load*, Carga de enrutamiento normalizada).

Hay que tener siempre en cuenta que el protocolo aquí propuesto además de resolver cuestiones de enrutamiento de paquetes como los demás protocolos contra los que se lo compara, resuelve el descubrimiento de nuevos clientes, asignación de clusters de pertenencia, asignación de tareas y recolección de resultados de las misma. Y todo lo anterior implica el envío de paquetes de control, por lo que mantener valores de NRL similares a los otros protocolos es por demás relevante.

Jitter

Los resultados que se pueden apreciar en las figuras: Figura 5.15 y Figura 5.16 determinan que existen diferencias significativas entre las curvas de desempeño solamente al comienzo de las simulaciones y una vez transcurrido 60 segundos todas ellas tienden a estabilizarse y solaparse unas con otras. Este solapamiento se ve claramente en el Figura 5.16 donde todas las curvas describen la misma evolución.

Para el caso particular de PROCCDIMI, este ha mostrado un desempeño similar al resto de los protocolos evaluados con esta métrica.

Throughput

Los resultados que se pueden apreciar en las figuras: Figura 5.17 y Figura 5.18 indican que no hay diferencias significativas entre las curvas de desempeño, sobre todo en las curvas que se corresponden a las simulaciones realizadas con números de nodos relativamente pequeños 10 y 20, puesto que la velocidad de transmisión de todos los protocolos ronda los 10kbps en el primer casos y 20kbps en el segundo, con la salvedad de DSDV que esta cercano a los 15kbps. Para redes más grandes en términos del número de nodos, los rendimientos se tornan más dispares aunque PROCCDIMI se mantiene en el rango medio de desempeño, no siendo el que registra ni la mejor, ni la peor velocidad de transmisión.

En base a lo arrojado por las figuras: Figura 5.17 y Figura 5.18, aquí cabe el mismo análisis que en la sección 5.2.2, concluyéndose que es un protocolo adecuado para redes de porte relativamente pequeño y cuando se analiza PROCCDIMI en redes con mayor porte

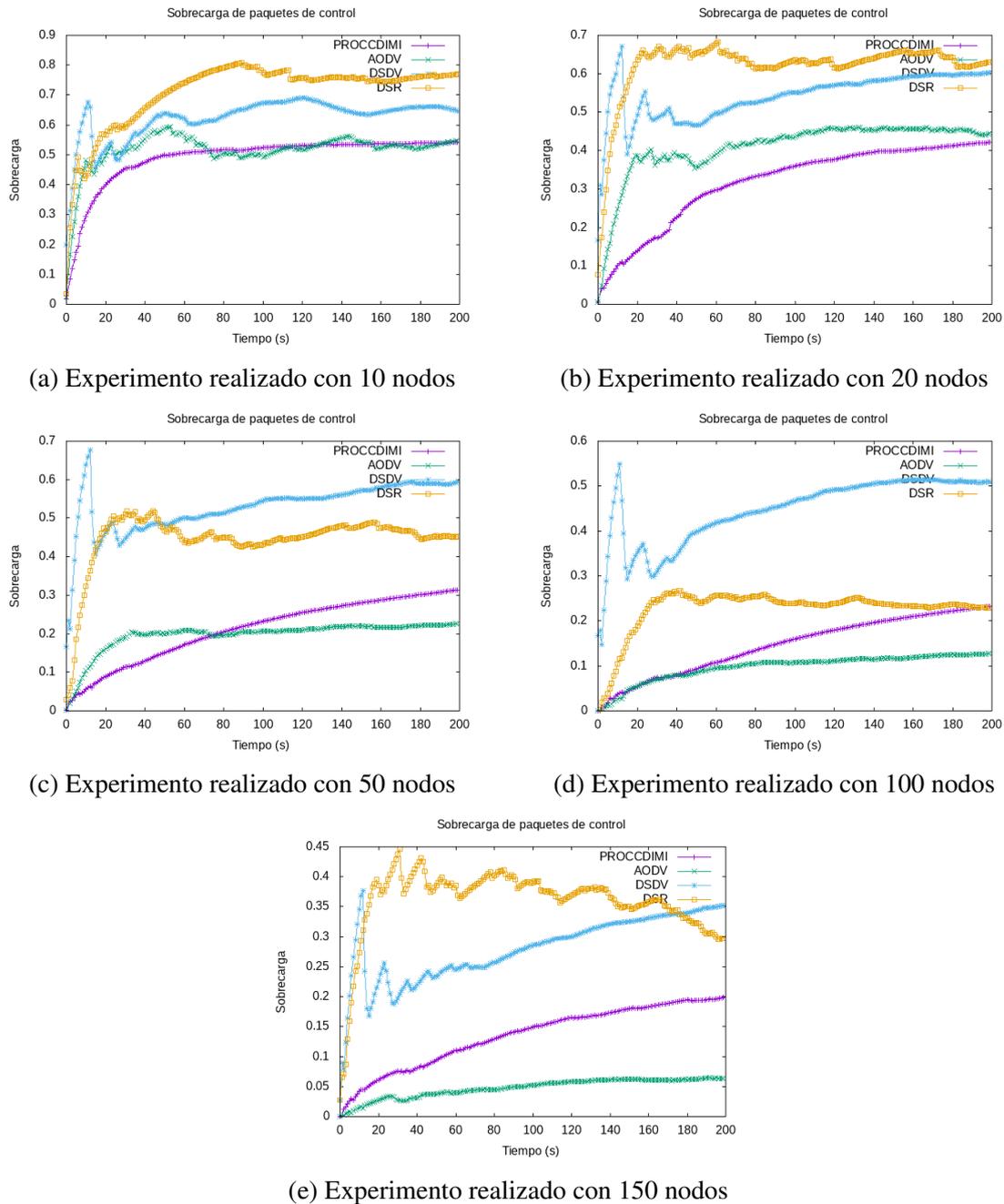


Figura 5.13 Carga de enrutamiento normalizada en función del tiempo y la cantidad de nodos. En modo asignación de tareas.

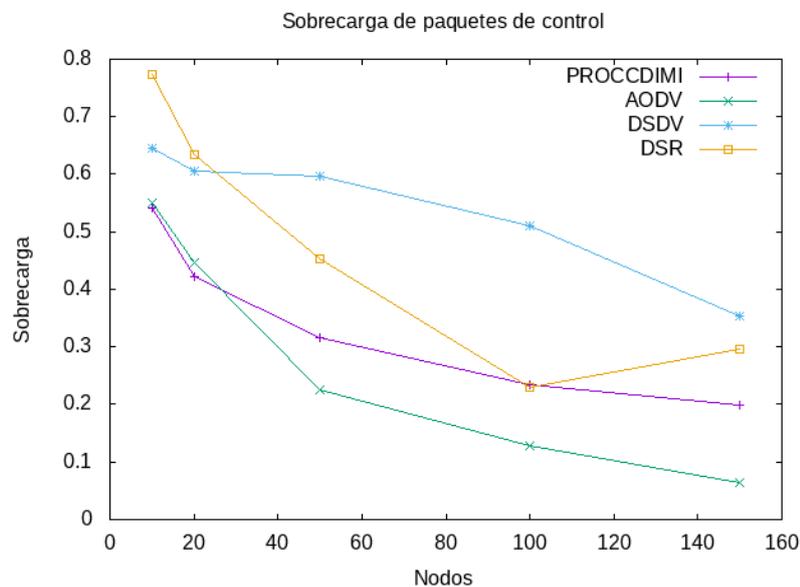


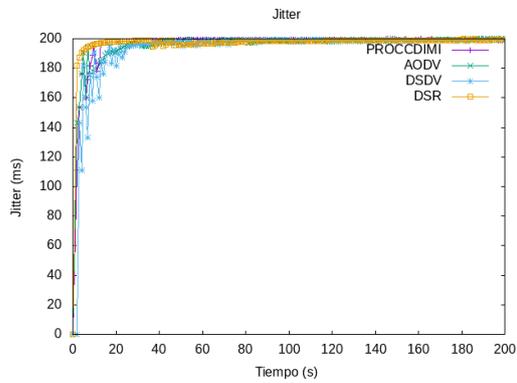
Figura 5.14 Carga de enrutamiento normalizada en función de la la cantidad de nodos. En modo asignación de tareas.

exhibe un desempeño levemente inferior que DSR y AODV, pero superior que DSDV en cuanto a la velocidad de transmisión refiere.

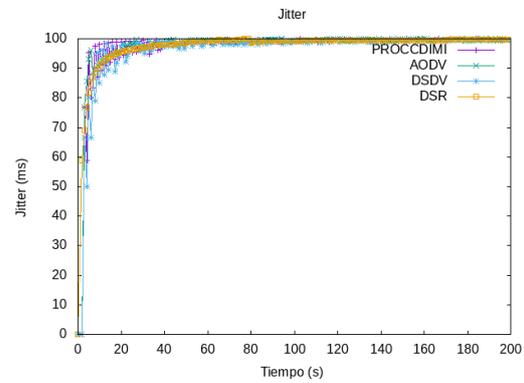
Costo energético promedio

En las figuras: Figura 5.19 y Figura 5.20 se muestran los detalles de la medición correspondiente al consumo energético vs. tiempo y al consumo energético vs. número de nodos. Esta métrica se utiliza para cuantificar el consumo de energía y de esta forma, conocer cuál de los protocolos optimiza mejor este recurso.

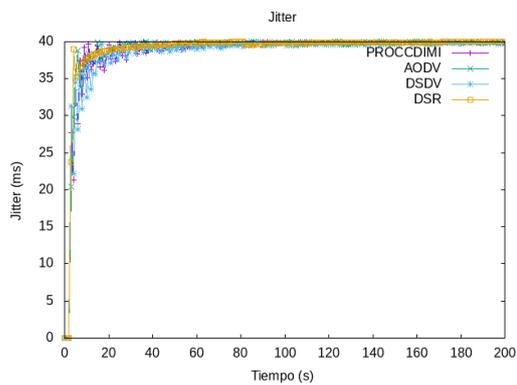
A diferencia de lo que ocurría en la sección 5.2.2, la adición de asignar tareas y la recolección posterior de resultados que son tareas extras que debe realizar PROCCDIMI en este apartado que no las realizaba anteriormente, incrementan el consumo de energía. Se puede apreciar que la curva correspondiente a PROCCDIMI asciende por encima de la curva de DSDV y por debajo de las de AODV y DSR para redes inferiores a los 130 nodos. Lo que indica que este protocolo consume menos energía que AODV y DSR, pero más que DSDV en redes de hasta 130 nodos. Las graficas también exhiben en los casos que el tiempo de operación es bajo, la curva de PROCCDIMI asciende por encima de las demás curvas indicando un consumo mayor de energía. Para redes de más de 130 nodos, PROCCDIMI vuelve a mostrar el mejor desempeño en cuanto al consumo de energético, esto se explica debido a que las tareas extras que el protocolo aquí propuesto realiza no consumen tanta



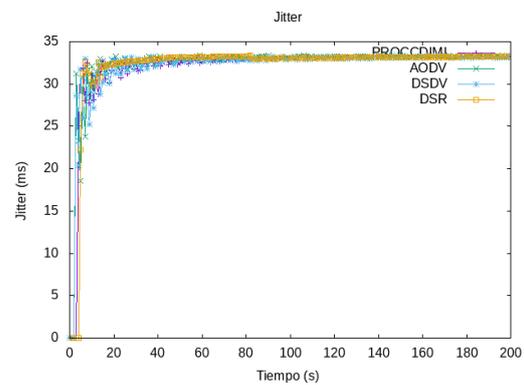
(a) Experimento realizado con 10 nodos



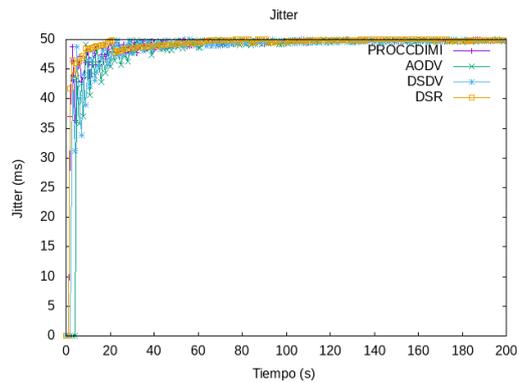
(b) Experimento realizado con 20 nodos



(c) Experimento realizado con 50 nodos



(d) Experimento realizado con 100 nodos



(e) Experimento realizado con 150 nodos

Figura 5.15 Jitter en función del tiempo y la cantidad de nodos. En modo asignación de tareas.

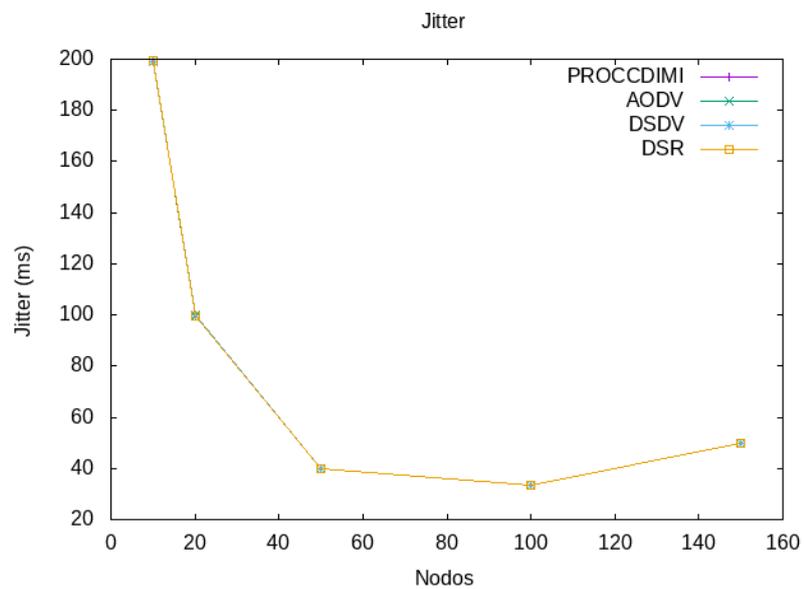
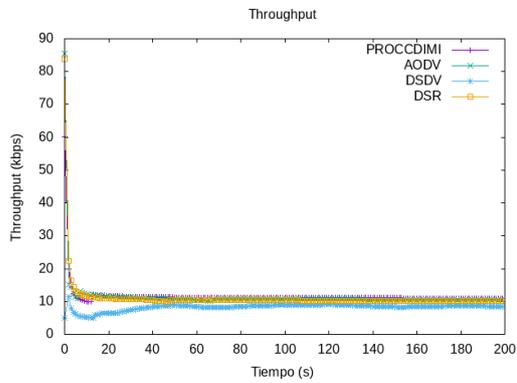
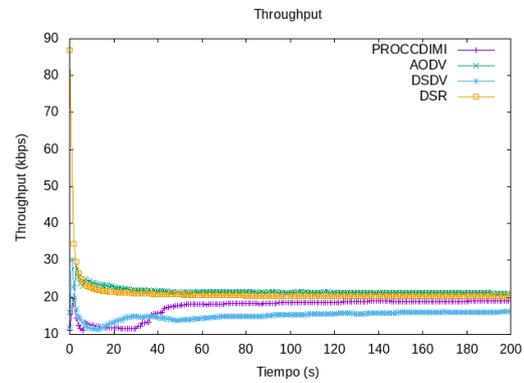


Figura 5.16 Jitter en función de la la cantidad de nodos. En modo asignación de tareas.

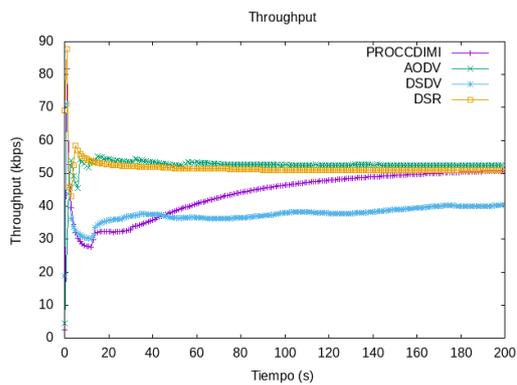
energía como la que se emplea para las tareas de ruteo por parte de los demás protocolos, ya que estas últimas involucran múltiples envíos de broadcast, mientras que las asignaciones de tareas y las recolecciones de resultados se resuelven vía unicast.



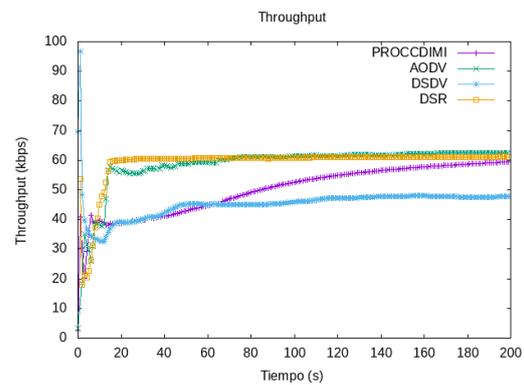
(a) Experimento realizado con 10 nodos



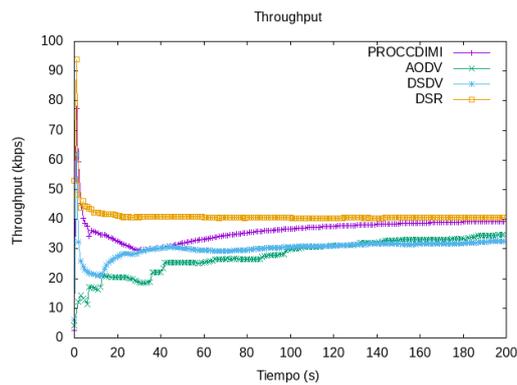
(b) Experimento realizado con 20 nodos



(c) Experimento realizado con 50 nodos



(d) Experimento realizado con 100 nodos



(e) Experimento realizado con 150 nodos

Figura 5.17 Throughput en función del tiempo y la cantidad de nodos. En modo asignación de tareas.

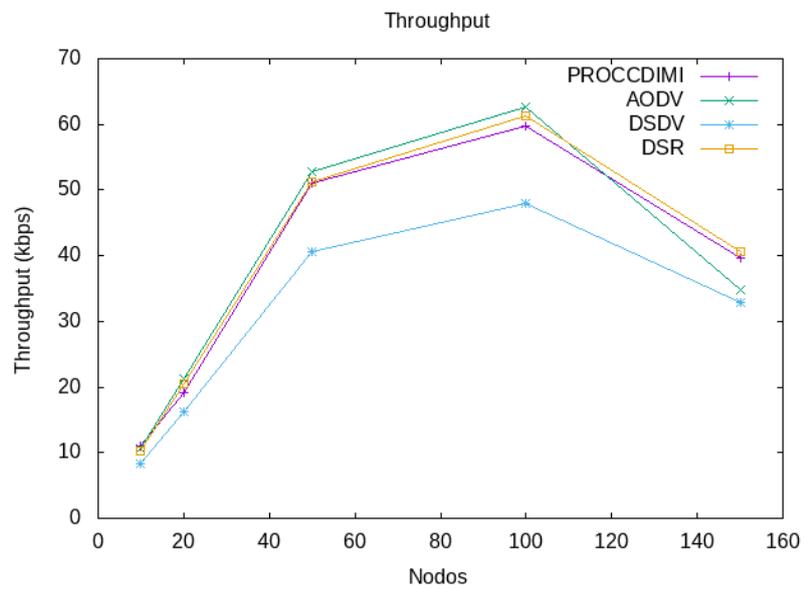


Figura 5.18 Throughput en función de la la cantidad de nodos. En modo asignación de tareas.

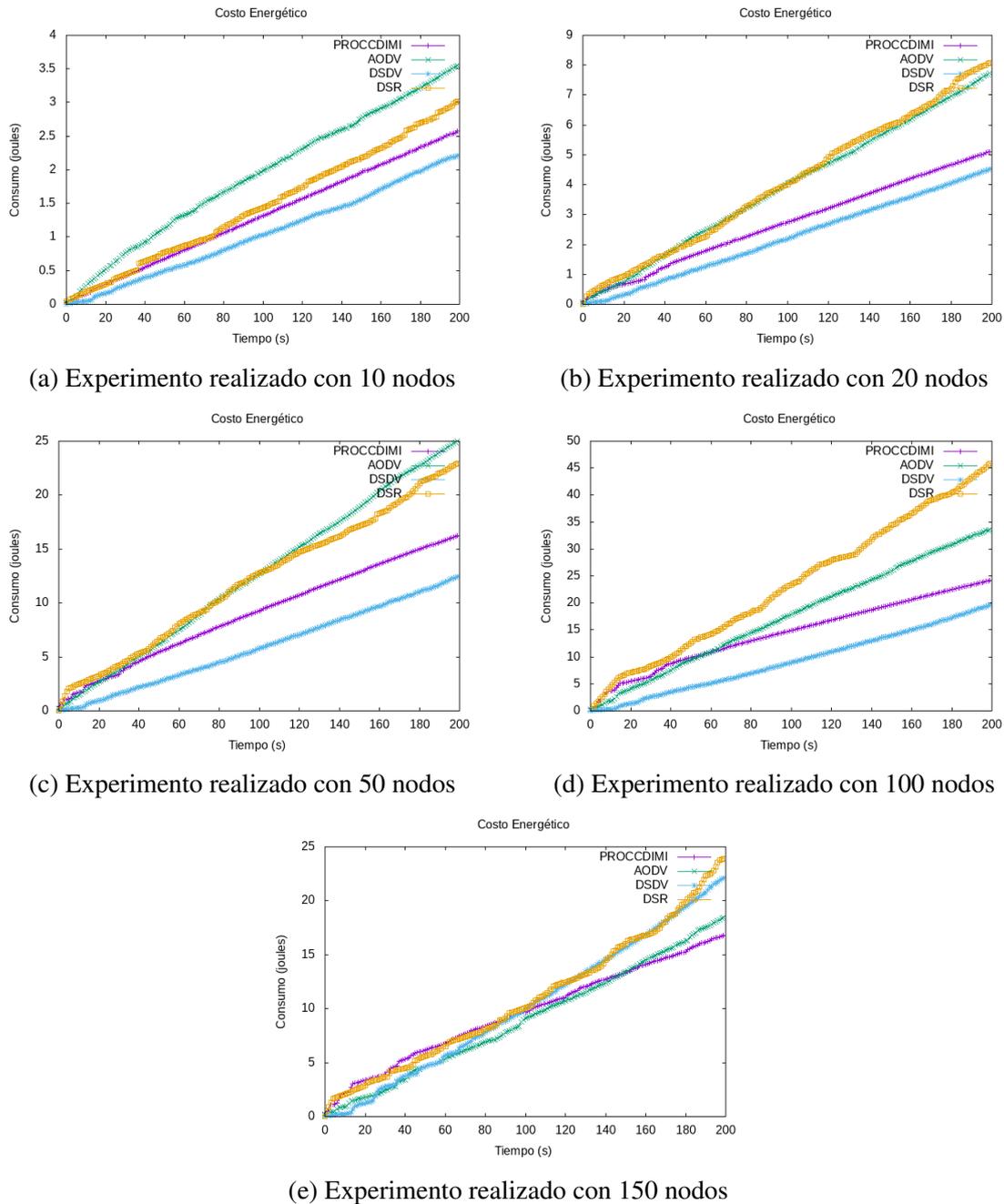


Figura 5.19 Costo energético promedio en función del tiempo y la cantidad de nodos. En modo asignación de tareas.

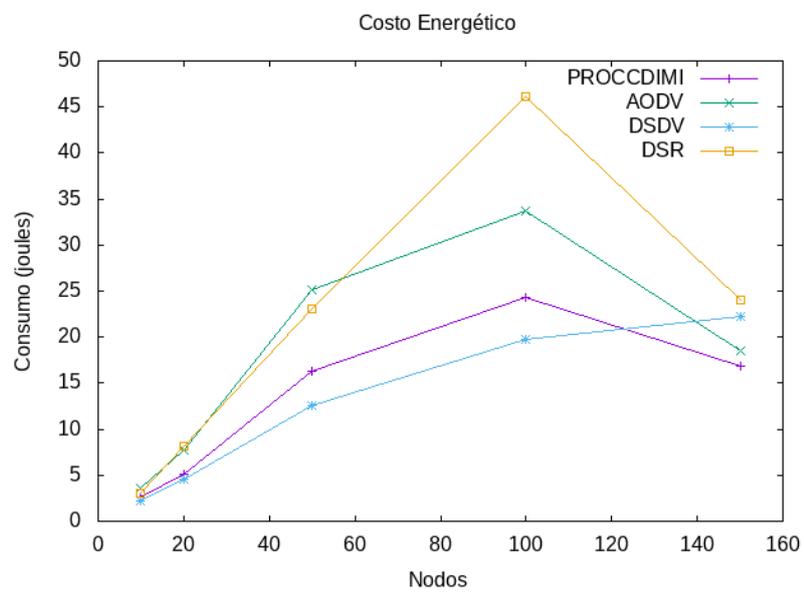


Figura 5.20 Costo energético promedio en función de la la cantidad de nodos. En modo asignación de tareas.

Capítulo 6

Conclusiones y Contribuciones

En la presente tesis doctoral se han relevados las tecnologías que hacen posible lidiar con el cálculo científico distribuido en redes móviles ad hoc sobre clientes indeterminados. Se han repasado las distintas soluciones disponibles actualmente, detallando e identificado los problemas latentes en cada una de ellas. Se ha planteado un esquema de solución que contribuye a resolver los problemas identificados para luego llevar a cabo las simulaciones necesarias a fin de validar la propuesta de solución aquí presentada. Concluyendo con el análisis de rendimiento de los resultados que arrojaron de las simulaciones.

En este capítulo se explican las contribuciones más importantes resultantes de esta investigación. En la sección 6.1 se detallan las principales conclusiones obtenidas al término del estudio. Posteriormente en la sección 6.2 se detallan todas las contribuciones que ha realizado esta tesis doctoral en base a los conocimientos volcados en la misma. Para finalmente en la sección 6.3 mencionar las líneas de investigación abiertas para realizar trabajos futuros.

6.0.1. Resumen

En la primera parte del presente documento se ofrece una visión general de las áreas de estudio que son abarcadas por el propósito de esta tesis, los objetivos, problemas a resolver y la estructura de la misma.

El segundo capítulo trata sobre los protocolos de enrutamiento, técnicas de clustering y diseño cross layer junto con los lineamientos de los conceptos básicos aplicados en el contexto de esta investigación. La revisión aplicada durante el desarrollo de este capítulo identifica los protocolos de enrutamientos más importantes que funciona en las MANET (*Mobile Ad hoc Networks*, Red móvil ad hoc) haciendo foco en sus cualidades objetivas e identificando a AODV (*Ad hoc On-demand Distance Vector*) como el que más se ajusta a las necesidades que hay que cubrir en esta investigación y determinándose como el candidato para ser ampliado en sus capacidades. Una vez establecido AODV como el sustrato para la implementación de la solución aquí propuesta, se pasa a estudiar los mecanismos existentes para la creación, administración y mantenimiento de clusters sobre redes móviles ad hoc. El capítulo concluye con el estudio comparativo entre el modelo en capas y el cross layer, determinándose que para este desarrollo en particular el camino más adecuado es la aplicación del último de los mencionados.

El tercer capítulo presenta el diseño de un protocolo que combina una solución para lidiar con la indeterminación de clientes al momento de comenzar un cálculo distribuido basado en un diseño cross layer con un mecanismo que agrupa los dispositivos presentes en la red en clusters, determina cuales de los dispositivos DCS (Dispositivo de capacidades superiores) serán los coordinadores de los clusters para dar soporte a la asignación de tareas a los clientes y la posterior recolección de resultados. La propuesta de solución implementada se basa en el protocolo de ruteo AODV y construida siguiendo los lineamientos del diseño cross layer. El resultado del proceso anterior toma la forma de una nueva extensión de AODV, la cual modifica los formatos de mensajes de AODV añadiendo nuevos campos a las cabeceras de los mensajes para dar soporte a toda la nueva funcionalidad de este nuevo protocolo.

En el cuarto capítulo a fin de validar la viabilidad y el rendimiento de PROCCDI-MI (PROtocolo para el Cálculo Científico Distribuido sobre clientes Móviles Indeterminados en MANETs) se investigó cuales son la herramientas de simulación actualmente vigentes y que nivel de idoneidad revisten para el tema de investigación de esta tesis doctoral. De este estudio surgió como la herramienta seleccionada para llevar a cabo las simulaciones NS2 (*Network Simulator version 2*). Se ajustaron los scripts OTCL (*Object-oriented Tool Command Language*) a las necesidades particulares del protocolo propuesto, para posteriormente vía scripts awk procesar las trazas resultantes del proceso de simulación con el objetivo de dimensionar los rendimientos en las áreas de interés.

También en el capítulo cuarto se presenta la necesidad de la creación de escenarios apropiados para las simulaciones con la herramienta NS2, dichos escenarios determinan las condiciones en las cuales los dispositivos móviles desarrollarán sus actividades imitando lo que sucede en la realidad. Al momento de la definición de un escenario de las características mencionadas anteriormente, se consideró las posiciones iniciales de los dispositivos, la cantidad de estos últimos que tomarán parte en la simulación, la cantidad de cada tipo de dispositivos presentes en el escenario y el patrón de movimiento que deben seguir los nodos de la MANET.

En el capítulo 5, la primera parte determina las métricas a utilizarse para evaluar el rendimiento de PROCCDIMI y estas son: tasa de entrega de paquetes, carga de enrutamiento normalizada, Jitter, Throughput y consumo energético promedio. En la segunda parte se realiza el análisis de rendimiento bajo distintas configuraciones de la red y los resultados bajo las métricas seleccionadas en la primera parte del capítulo también son analizados. Se examinaron minuciosamente los dos modos de operación de PROCCDIMI, en el primer caso donde cada dispositivo en la red inicialmente no posee coordinador asociado realiza una solicitud de descubrimiento de coordinador de cluster. Y el segundo caso, cuando el coordinador del cluster debe atender a las solicitudes de descubrimiento de coordinador tanto como realizar la asignación de tareas distribuidas para luego recolectar los resultados. Todo este procesamiento realizándose de forma simultánea. Los resultados simulados de los dos modos de operación antes comentados muestran que, aunque las comparaciones realizadas sean con protocolos de ruteo que solamente tienen esa única función, PROCCDIMI exhibió un rendimiento similar a estos y en algunos casos tuvo un desempeño mejor como por ejemplo en el consumo energético.

6.1. Conclusión

La propuesta presentada en esta tesis doctoral organiza los nodos de la MANET en clusters y elige coordinadores de estos agrupamientos lógicos a los dispositivos con capacidades superiores, para que estos asuman el rol de asignar los clusters de pertenencia a las unidades de red que lo solicitan, tomar el cálculo científico entre los cliente que llevaran a cabo dicho computo en forma distribuida y finalmente recolectar todos los resultados parciales, compilarlos y devolver el resultado del calculo científico. Ciertamente los coordinadores tendrán una mayor carga de trabajo. La adopción de un diseño cross layer que emplea al protocolo de enrutamiento AODV (*Ad hoc On-demand Distance Vector*) como mecanismo de red subyacente, permite llevar a cabo: el proceso de ruteo, de descubrimiento de nuevos clientes, asignación de tareas y recolección de resultados, todo en forma simultánea extendiendo los mensajes de control originales de AODV con el fin de minimizar el número de mensajes necesarios en este proceso.

Los resultados obtenidos de analizar a PROCCDIMI (PROtocolo para el Cálculo Científico Distribuido sobre clientes Móviles Indeterminados en MANETs) mediante las simulaciones llevadas a cabo sobre la herramienta NS2 (*Network Simulator version 2*) fueron sopesados y se arribaron a las siguientes conclusiones:

- PROCCDIMI es notoriamente superior en términos de confiabilidad que el resto de los protocolos de enrutamiento contra los cuales se lo comparó cuando se trata de redes relativamente pequeñas en cuanto a cantidad de nodos.
- PROCCDIMI posee una carga de enrutamiento comparable a AODV, DSR (*Dynamic Source Routing*) y DSDV (*Destination Sequenced Distance Vector*) sin perder de vista que además se encarga de tareas asociadas a la administración de los clusters y el computo distribuido, con lo cual es un resultado excelente del desempeño del protocolo.
- El Jitter que mostró PROCCDIMI durante las simulaciones es comparable a los demás protocolos examinados, con lo cual la afectación que tendrá sobre la performance de toda la red, principalmente en los requisitos de *buffering* que se deben tener para alcanzar un uso razonable serán exactamente iguales al resto de los protocolos comparados. Este conocimiento emergente es muy importante ya que establece que la migración de una red que esta actualmente funcionando con AODV, DSR o DSDV a PROCCDIMI es viable en términos de la dimensiones de buffers ya establecidos y que se encuentran actualmente en funcionamiento.

- El Throughput o velocidad de transferencia de datos que mostró PROCCDIMI durante las simulaciones permiten concluir que es un protocolo adecuado para redes de porte relativamente pequeño en cuanto a cantidad de nodos.
- PROCCDIMI es el mejor en cuanto al manejo de la utilización de la energía ya que es el que consumió menos de esta última durante las simulaciones realizadas, mientras que el resto registró niveles de consumo superiores al de la propuesta de esta tesis doctoral. El hecho de que PROCCDIMI tenga menor gasto energético es producto de la estrategia de conformar grupos lógicos de nodos, dado que es uno de los beneficios que se espera obtener al formar agrupamientos lógicos, es la reducción del consumo de energía. De todos modos que la solución aquí propuesta tenga un consumo de energía menor al resto de los protocolos contra los que se lo comparó y teniendo en cuenta toda la actividad extra que realiza, es realmente sorprendente.
- El diseño cross layer, la organización de nodos en grupos lógicos y la utilización de protocolos de ruteo reactivos, como lo es AODV, proveen la plataforma adecuada en términos de confiabilidad, velocidad de transmisión de datos y consumo de energía para la implementación de nuevas soluciones con prestaciones similares y en algunos casos mejores que los protocolos de ruteos ya establecidos en el ámbito de las MANET.

6.2. Contribuciones

Las contribuciones de la presente investigación se dan en términos de ofrecer un mecanismo para lidiar con la indeterminación del número de clientes con que se cuenta a la hora de comenzar un cálculo científico distribuido, proveer un mecanismo cross layer para la gestionar esto último y poner en relevancia la heterogeneidad existente en los nodos que conforman las MANET y poder sacarle provecho a esta disparidad de características para que los dispositivos con capacidades superiores se encarguen de la coordinación del computo distribuido y la gestión de los clusters. Además de proponer modificar a AODV para que realice lo antes enumerado y tener prestaciones de red comparables a los protocolos existentes.

Los aportes de este documento de investigación son informados a continuación:

- Desarrollo de un nuevo modelo que utiliza un diseño cross layer y de organización de nodos en agrupamientos lógicos que gestiona el cómputo y asigna tareas diferenciales de acuerdo a las prestaciones del nodo para lidiar con la indeterminación de clientes al momento de realizar cálculos científicos sobre MANET.

- Utilizar el protocolo de ruteo reactivo AODV como sustrato para construir una extensión orientada a descubrir los clientes disponibles en una red móvil ad hoc para llevar a cabo un cómputo distribuido.
- Desarrollo de un nuevo protocolo que emerge como solución de los problemas asociados a la indeterminación de clientes al momento de realizar cálculos científicos sobre MANET y la gestión del cómputo distribuido.
- Pruebas de rendimiento de la solución propuesta en esta tesis doctoral que proveen un análisis cuantitativo del comportamiento de PROCCDIMI con respecto a los demás protocolos contra los que se lo comparó.

6.3. Líneas de trabajo futuras

Los resultados obtenidos en esta tesis suponen un avance en el tema de cálculo científico distribuido sobre clientes móviles indeterminados en MANET (*Mobile Ad hoc Networks*, Red móvil ad hoc), sin embargo es un tema aún abierto al que se le pueden añadir propuestas que inicien nuevas líneas de investigación. En este sentido y continuando con la misma línea de recurrir a un diseño cross layer para descubrir nuevos clientes, asociarlos a un coordinador, asignarles tareas y recolectar los resultados durante el proceso de enrutamiento mediante la formación de clusters coordinados por dispositivos de capacidades superiores, se proponen los siguientes trabajos:

- Probar el desempeño de PROCCDIMI en entornos más reales: Modelar la generación de tráfico en la red de forma más realista, para que no sea únicamente la información transmitida entre los nodos de la ruta establecida la que circule por la red, si no que pueda haber interferencias debidas a otras comunicaciones. Dicho en otras palabras la incorporación de ruido en los canales de comunicación. El principal problema en pruebas reales es la interferencia que pueda haber en el ambiente durante la realización del experimento. El ruido del entorno permite que las pruebas realizadas en el laboratorio puedan considerarse similares a las que se podrían llevar a cabo en un escenario académico o científico donde los demás equipos provocan interferencias y problemas de comunicaciones.

Al margen del problema de las interferencias, es interesante el estudio del protocolo en exteriores con distancias reales entre los nodos en lugar del uso en ámbitos simulados. De esta forma se podrían realizar pruebas que consideren la potencia de transmisión y como afecta al desempeño de PROCCDIMI bajo las métricas antes mencionadas para su evaluación. Del mismo modo que para el caso anterior y utilizando la misma metodología se podría observar el desempeño que exhibe la solución aquí propuesta utilizando otros modelos de generación de tráfico, como el de Pareto que permite acercarse a una distribución de tráfico autosimilar revistiendo este estudio un gran interés en el área de los protocolos de red.

- Implementar un mecanismo de seguridad: Otro aspecto a evaluar en el protocolo PROCCDIMI es la incorporación de medidas que provean seguridad-privacidad. En este área de investigación se analiza cuestiones de gestión de identidades de usuarios y dispositivos en el proceso de enrutamiento y el descubrimiento de nuevos clientes. También el manejo del acceso a la red y la detección y prevención de intrusiones serian líneas de investigación para extender PROCCDIMI. Dado el ámbito para el cual esta

pensada la solución propuesta en esta tesis doctoral, asegurar que el intercambio de información sea llevado a cabo mediante un mecanismo confiable indicará el éxito o fracaso de la misma. Con lo cual este último tópico también debe ser investigado dentro de la línea de investigación que intenta el proveer una extensión segura de PROCCDIMI.

- Implementar la automatización de elección del coordinador de cluster: En las simulaciones realizadas los escenarios definían estáticamente los tipos de dispositivos y esta definición era realizada ad hoc. Es decir, los nodos que han de ser de tipo DCS (Dispositivo de capacidades superiores), DCL (Dispositivo de capacidades limitadas) o coordinador de cluster son fijados en los scripts OTCL (*Object-oriented Tool Command Language*) utilizados por la herramienta de simulación NS2 (*Network Simulator version 2*). Con el propósito de hacer más sólida y estricta la estrategia de elección de coordinador es preciso emplear algoritmos ampliamente probados como por ejemplo *Highest-Degree heuristic*, *Lowest-ID heuristic*, *Node-weight heuristic*, *Weighted Clustering algorithm* o *Distributed Weighted Clustering Algorithm*. De esta forma en los escenarios de aplicación reales la elección del coordinador se hará en base a cualquiera de los algoritmos mencionados. Estas estrategias de clustering habitualmente eligen sus líderes de grupo en base a cierto criterio de selección, bien sea por medio de sopesar la conectividad o de acuerdo al mayor-menor valor de identificador (ID) o en función del número de saltos de distancia que existe entre los nodos.

Es interesante el estudio de llevar a cabo la implementación de un mecanismo que automáticamente identifique el tipo de dispositivo. Y cuando una unidad de red cumpla con las características esperadas para officar de coordinador, esta sea elegida como tal. Para ello se sugiere utilizar *Weighted Clustering Algorithm* que es un algoritmo distribuido basado en pesos. *Weighted Clustering Algorithm* elige a los líderes de cluster realizando una valoración del peso que posee cada nodo teniendo en cuenta aspectos relacionados a las prestaciones del dispositivo. Lo anterior es lo suficientemente aproximado al esquema que se propone en PROCCDIMI en cuanto al dispositivo que coordina el cluster, el cual debe tener características superiores en cuanto a la capacidad de memoria, potencia de cómputo, prestaciones de comunicación y duración de la batería refiere.

- Probar PROCCDIMI con otro protocolo de enrutamiento: A la luz de los resultados obtenidos en las simulaciones y a la naturaleza intrínseca de los protocolos de enrutamiento que funcionan sobre las MANET, se puede decir que los protocolos reactivos introducen poca carga de paquetes de control, como resultado se optimiza el ancho

de banda y se obtienen altas tasas de entrega de paquetes durante el proceso de descubrimiento de nuevos clientes; pero en cambio se registran tiempos de latencias muy elevados. Por tal razón resultaría de interés evaluar a PROCCDIMI con un protocolo de ruteo proactivo como sustrato para realizar toda su operatoria reemplazando a AODV, con el objetivo de dimensionar en qué medida mejoran los tiempos de latencias.

Bibliografía

- [1] Adjih, C., Clausen, T., Jacquet, P., Laouiti, A., Minet, P., Muhlethaler, P., Qayyum, A., and Viennot, L. (2003). Optimized link state routing protocol. rfc 3626. *IETF*.
- [2] Agarwal, R., Motwani, D., et al. (2009). Survey of clustering algorithms for manet. *arXiv preprint arXiv:0912.2303*.
- [3] Aijaz, A., Bochow, B., Dötzer, F., Festag, A., Gerlach, M., Kroh, R., and Leinmüller, T. (2006). Attacks on inter vehicle communication systems-an analysis. *Proc. WIT*, pages 189–194.
- [4] Akbani, R., Korkmaz, T., and Raju, G. (2012). Mobile ad-hoc networks security. In *Recent advances in computer science and information engineering*, pages 659–666. Springer.
- [5] Alizadeh-Shabdiz, F. and Subramaniam, S. (2006). Analytical models for single-hop and multi-hop ad hoc networks. *Mobile Networks and Applications*, 11(1):75–90.
- [6] Antonopoulos, N. and Gillam, L. (2010). *Cloud computing*. Springer.
- [7] Arya, H. and Chauhan, A. (2020). Survey on various routing protocols and mobility models used in mobile ad hoc network. *Current Trends in Information Technology*, 9(3):40–45.
- [8] Awang, A., Husain, K., Kamel, N., and Aissa, S. (2017). Routing in vehicular ad-hoc networks: a survey on single-and cross-layer design techniques, and perspectives. *IEEE Access*, 5:9497–9517.
- [9] Bajaj, L., Takai, M., Ahuja, R., Tang, K., Bagrodia, R., and Gerla, M. (1999). Glomosim: A scalable network simulation environment. *UCLA computer science department technical report*, 990027(1999):213.
- [10] Baydal Cardona, M. E. (2018). Clasificación de las redes inalámbricas.
- [11] Beijar, N. (2002). Zone routing protocol (zrp). *Networking Laboratory, Helsinki University of Technology, Finland*, 9:1–12.
- [12] Belding-Royer, E. M., Basagni, S., Conti, M., Giordano, S., and Stojmenovic, I. (2004). Routing approaches in mobile ad hoc networks. *Mobile ad hoc networking*, 1(1):275–300.
- [13] Belding-Royer, E. M. and Perkins, C. E. (2003). Evolution and future directions of the ad hoc on-demand distance-vector routing protocol. *Ad Hoc Networks*, 1(1):125–150.

- [14] Brindha, M. (2017). A survey on cross layer distributed topology control in mobile adhoc network. *Bonfring International Journal of Networking Technologies and Applications*, 4 (1), 1, 3.
- [15] Broch, J., Maltz, D. A., Johnson, D. B., Hu, Y.-C., and Jetcheva, J. (1998). A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, pages 85–97.
- [16] Carneiro, G. (2010). Ns-3: Network simulator 3. In *UTM Lab Meeting April*, volume 20, pages 4–5.
- [17] Chakeres, I. D. and Belding-Royer, E. M. (2004). Aodv routing protocol implementation design. In *24th International Conference on Distributed Computing Systems Workshops, 2004. Proceedings.*, pages 698–703. IEEE.
- [18] Chander, D. and Kumar, R. (2018). Qos enabled cross-layer multicast routing over mobile ad hoc networks. *Procedia Computer Science*, 125:215–227.
- [19] Chang, X. (1999). Network simulations with opnet. In *WSC'99. 1999 Winter Simulation Conference Proceedings. Simulation-A Bridge to the Future' (Cat. No. 99CH37038)*, volume 1, pages 307–314. IEEE.
- [20] Chiang, C.-C., Wu, H.-K., Liu, W., and Gerla, M. (1997). Routing in clustered multihop, mobile wireless networks with fading channel. In *proceedings of IEEE SICON*, volume 97, pages 197–211.
- [21] Chiang, M. (2004). To layer or not to layer: Balancing transport and physical layers in wireless multihop networks. In *IEEE INFOCOM 2004*, volume 4, pages 2525–2536. IEEE.
- [22] Conti, M., Maselli, G., Turi, G., and Giordano, S. (2004). Cross-layering in mobile ad hoc network design. *Computer*, 37(2):48–51.
- [23] Correa, B. A., Ospina, L., and Hincapié, R. C. (2007). Survey of clustering techniques for mobile ad hoc networks. *Revista Facultad de Ingeniería Universidad de Antioquia*, 41:145–161.
- [24] de California, U. (1999). The network simulator - ns-2, <https://www.isi.edu/nsnam/ns/>.
- [25] Divecha, B., Abraham, A., Grosan, C., and Sanyal, S. (2007). Impact of node mobility on manet routing protocols models. *J. Digit. Inf. Manag.*, 5(1):19–23.
- [26] Documentation, Q. S. (2006). Qualnet 5.0 users guide. *Scalable Network Technologies, Inc., Los Angeles, CA, 90045*.
- [27] Doss, R. and Pan, L. (2006). A case for cross layer design: The impact of physical layer properties on routing protocol performance in manets. In *ATNAC, Australian Telecommunication Networks and Applications Conference, 2006.*, pages 409–413. Australian Telecommunication Networks and Applications Conference.

- [28] Edwards, J. and Bramante, R. (2015). *Networking self-teaching guide: OSI, TCP/IP, LANs, MANs, WANs, implementation, management, and maintenance*. John Wiley & Sons.
- [29] Gagandeep, A., Kumar, P., et al. (2012). Analysis of different security attacks in manets on protocol stack a-review. *International Journal of Engineering and Advanced Technology (IJEAT)*, 1(5):269–75.
- [30] Giordano, S. et al. (2002). Mobile ad hoc networks. *Handbook of wireless networks and mobile computing*, pages 325–346.
- [31] Golla, V., Jayanthi, G., and Shivashankar, H. (2014). Designing energy routing protocol with power consumption optimization in manet'. *IEEE Transactions on Emerging topics in Computing*, 2(2):192–197.
- [32] Gowrishankar, S., Basavaraju, T., and Sarkar, S. K. (2007). Effect of random mobility models pattern in mobile ad hoc networks. *International Journal of Computer Science and Network Security*, 7(6):160–164.
- [33] Hakak, S., Latif, S. A., Anwar, F., and Alam, M. K. (2014). Impact of key factors on average jitter in manet. In *First International Conference on Systems Informatics, Modeling and Simulation*. Computer Society, IEEE, pages 179–183.
- [34] He, G. (2002). Destination-sequenced distance vector (dsv) protocol. *Networking Laboratory, Helsinki University of Technology*, pages 1–9.
- [35] Higham, D. J. and Higham, N. J. (2016). *MATLAB guide*. SIAM.
- [36] Hollerung, T. D. (2003). The cluster-based routing protocol. *project group 'Mobile Ad-Hoc Networks Based on Wireless LAN'-2003/4*.
- [37] Hsu, C.-H. and Feng, K.-T. (2007). On-demand routing-based clustering protocol for mobile ad hoc networks. In *2007 IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1–5. IEEE.
- [38] Huang, C., Zhang, Y., Jia, X., Shi, W., Cheng, Y., and Zhou, H. (2006). An on-demand clustering mechanism for hierarchical routing protocol in ad hoc networks. In *2006 International Conference on Wireless Communications, Networking and Mobile Computing*, pages 1–6. IEEE.
- [39] Issariyakul, T. and Hossain, E. (2009). Introduction to network simulator 2 (ns2). In *Introduction to network simulator NS2*, pages 1–18. Springer.
- [40] Ivanov, S., Herms, A., and Lukas, G. (2007). Experimental validation of the ns-2 wireless model using simulation, emulation, and real network. In *Communication in Distributed Systems-15. ITG/GI Symposium*, pages 1–12. VDE.
- [41] Jacob, B., Brown, M., Fukui, K., Trivedi, N., et al. (2005). Introduction to grid computing. *IBM redbooks*, pages 3–6.
- [42] Jha, G. K., Kumar, N., Sharma, H., and Sharma, K. (2011). Improvement of qos performance in manet by qos-tora: A tora based qos routing algorithm. In *International Conference on Network Security and Applications*, pages 89–99. Springer.

- [43] Joa-Ng, M. and Lu, I.-T. (1999). A peer-to-peer zone-based two-level link state routing for mobile ad hoc networks. *IEEE Journal on selected areas in communications*, 17(8):1415–1425.
- [44] Johnson, D., Hu, Y.-c., Maltz, D., et al. (2007). The dynamic source routing protocol (dsr) for mobile ad hoc networks for ipv4. Technical report, RFC 4728.
- [45] Johnson, D. B. (1999). Validation of wireless and mobile network models and simulation. In *DARPA/NIST network simulation validation workshop*.
- [46] Johnson, D. B., Maltz, D. A., Broch, J., et al. (2001). Dsr: The dynamic source routing protocol for multi-hop wireless ad hoc networks. *Ad hoc networking*, 5(1):139–172.
- [47] Katkar, P. S. and Ghorpade, D. V. R. (2016). Comparative study of network simulator: Ns2 and ns3. *International Journal of Advanced Research in Computer Science and Software Engineering*, 6(3).
- [48] Kaur, H., Sahni, V., and Bala, M. (2013). A survey of reactive, proactive and hybrid routing protocols in manet: A review. *network*, 4(3):498–500.
- [49] Khan, K. U. R., Zaman, R. U., Reddy, A. V., Reddy, K. A., and Harsha, T. S. (2008). An efficient dsdv routing protocol for wireless mobile ad hoc networks and its performance comparison. In *2008 Second UKSIM European Symposium on Computer Modeling and Simulation*, pages 506–511. IEEE.
- [50] Kliazovich, D. and Granelli, F. (2006). Cross-layer congestion control in ad hoc wireless networks. *Ad Hoc Networks*, 4(6):687–708.
- [51] Korpela, E., Werthimer, D., Anderson, D., Cobb, J., and Lebofsky, M. (2001). Seti@home—massively distributed computing for seti. *Computing in science & engineering*, 3(1):78–83.
- [52] Kotz, D. and Essien, K. (2005). Analysis of a campus-wide wireless network. *Wireless Networks*, 11(1-2):115–133.
- [53] Kurkowski, S., Camp, T., and Colagrosso, M. (2005). Manet simulation studies: The current state and new simulation tools. *Mobile Computing and Communications Review*, 9(4):50–61.
- [54] Lashkari, A. H., Danesh, M. M. S., and Samadi, B. (2009). A survey on wireless security protocols (wep, wpa and wpa2/802.11 i). In *2009 2nd IEEE International Conference on Computer Science and Information Technology*, pages 48–52. IEEE.
- [55] Liu, C. and Kaiser, J. (2005). Survey of mobile ad hoc network routing protocols.
- [56] Liu, S., Zhang, D., Liu, X., Zhang, T., and Wu, H. (2020). Adaptive repair algorithm for tora routing protocol based on flood control strategy. *Computer Communications*, 151:437–448.
- [57] Malkin, G. (1999). Nortel networks multi-link multi-node ppp bundle discovery protocol. In *Technical Report*. September 1999. RFC 2701.

- [58] Manickam, P., Baskar, T. G., Girija, M., and Manimegalai, D. D. (2011). Performance comparisons of routing protocols in mobile ad hoc networks. *arXiv preprint arXiv:1103.0658*.
- [59] Marin-Perianu, R., Scholten, J., Havinga, P. J., and Hartel, P. H. (2008). Cluster-based service discovery for heterogeneous wireless sensor networks. *International Journal of Parallel, Emergent and Distributed Systems*, 23(4):325–346.
- [60] Marsan, M. A., Bobbio, A., and Donatelli, S. (1996). Petri nets in performance analysis: An introduction. In *Advanced Course on Petri Nets*, pages 211–256. Springer.
- [61] Nasipuri, A., Zhuang, J., and Das, S. R. (1999). A multichannel csma mac protocol for multihop wireless networks. In *WCNC. 1999 IEEE Wireless Communications and Networking Conference (Cat. No. 99TH8466)*, volume 3, pages 1402–1406. IEEE.
- [62] Niu, X., Tao, Z., Wu, G., Huang, C., and Cui, L. (2006). Hybrid cluster routing: an efficient routing protocol for mobile ad hoc networks. In *2006 IEEE International Conference on Communications*, volume 8, pages 3554–3559. IEEE.
- [63] Palma, D. and Curado, M. (2012). Onto scalable ad-hoc networks: deferred routing. *Computer Communications*, 35(13):1574–1589.
- [64] Pandey, A. K. and Fujinoki, H. (2005). Study of manet routing protocols by glomosim simulator. *International Journal of Network Management*, 15(6):393–410.
- [65] Pati, A. P., Kanth, K. R., Sharanya, B., Kumar, M. D., and Malavika, J. (2011). Design of an energy efficient routing protocol for manets based on aodv. *International Journal Of Computer Science Issues (IJCSI)*, 8(4):215.
- [66] Perkins, C., Belding-Royer, E., Das, S., et al. (2003). Ad hoc on-demand distance vector (aodv) routing.
- [67] project, N.-. (2012). Ns-3 simulator, ns-3 tutorial.
- [68] Qin, L. and Kunz, T. (2004). Survey on mobile ad hoc network routing protocols and cross-layer design.
- [69] Rahman, A. H. A. and Zukarnain, Z. A. (2009). Performance comparison of aodv, dsdv and i-dsdv routing protocols in mobile ad hoc networks. *European Journal of Scientific Research*, 31(4):566–576.
- [70] Rajkumar, G. and DURAISAMY, D. (2012). A review of ad hoc on-demand distance vector routing protocol for mobile ad hoc networks. *Journal of Theoretical and Applied Information Technology*, 36(1):134–144.
- [71] Ros, F. J. and Ruiz, P. M. (2007). Cluster-based olsr extensions to reduce control overhead in mobile ad hoc networks. In *Proceedings of the 2007 international conference on Wireless communications and mobile computing*, pages 202–207.
- [72] Roy, R. R. (2011). Realistic random direction mobility. In *Handbook of Mobile Ad Hoc Networks for Mobility Models*, pages 223–244. Springer.

- [73] Royer, E. M., Melliar-Smith, P. M., and Moser, L. E. (2001). An analysis of the optimum node density for ad hoc mobile networks. In *ICC 2001. IEEE International Conference on Communications. Conference Record (Cat. No. 01CH37240)*, volume 3, pages 857–861. IEEE.
- [74] Sharma, K. and Dhir, N. (2014). A study of wireless networks: Wlans, wpans, wmans, and wwans with comparison. *International Journal of Computer Science and Information Technologies*, 5(6):7810–7813.
- [75] Shih, T.-F. and Yen, H.-C. (2006). Core location-aided cluster-based routing protocol for mobile ad hoc networks. *pp223-228*.
- [76] Shrivastava, S. and Pargania, V. (2020). Survey of routing protocols, simulation, testing tools and mobility models in manet. 22:4–5.
- [77] Singh, M. P. (2004). *The practical handbook of internet computing*. CRC press.
- [78] Teotia, S., Sharma, S., et al. (2018). A survey on mobility models in mobile ad-hoc networks. *International Journal of Management, IT and Engineering*, 8(12):269–274.
- [79] Tirado Fernández, F. et al. (2017). Los procesadores y la computación en el final de la ley de moore.
- [80] Tolosa, G. (2014). Protocolos y modelo osi. *Recuperado de <http://www.tyr.unlu.edu.ar/TYR-publica/02-Protocolosy-OSI.pdf>*.
- [81] Wister Ovando, M. A. (2009). Arquitectura de descubrimiento de servicios en manet basada en dispositivos de capacidades superiores liderando clusters. *Proyecto de investigacio*.
- [82] Xiong, C., Murata, T., and Tsai, J. (2002). Modeling and simulation of routing protocol for mobile ad hoc networks using colored petri nets. In *Proceedings of the conference on Application and theory of petri nets: formal methods in software engineering and defence systems-Volume 12*, pages 145–153.
- [83] Yang, X., Chen, Q., Chen, C., and Zhao, J. (2018). Improved zrp routing protocol based on clustering. *Procedia computer science*, 131:992–1000.
- [84] Zhang, S., Chen, X., Zhang, S., and Huo, X. (2010). The comparison between cloud computing and grid computing. In *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*, volume 11, pages V11–72. IEEE.
- [85] Zhang, X. and Riley, G. F. (2005). Performance of routing protocols in very large-scale mobile wireless ad hoc networks. In *13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, pages 115–122. IEEE.
- [86] Zhu, S. (2001). *Performance analysis of a modified zone-based hierarchical link state routing*. PhD thesis, Concordia University.

Apéndice A

Escenarios NS2

En este apéndice se incluyen los scripts escritos en OTCL (*Object-oriented Tool Command Language*) utilizados para interactuar con NS2 (*Network Simulator version 2*) con los dos modos de operación de PROCCDIMI (PROtocolo para el Cálculo Científico Distribuido sobre clientes Móviles Indeterminados en MANETs). Estos inicializan todos los parámetros utilizados en las simulaciones según se detalló en el capítulo 4.

Modo de operación descubrimiento:

```
#=====
# Se definen todas las variables globales de configuración
#=====

set opt(chan)          Channel/WirelessChannel    ;# tipo de canal
set opt(prop)          Propagation/TwoRayGround   ;# modelo de propagación de radio
set opt(netif)         Phy/WirelessPhy           ;# tipo de interface de red
set opt(mac)           Mac/802_11                ;# tipo de MAC
set opt(ifq)           Queue/DropTail/PriQueue   ;# tipo de cola de paquetes
set opt(ll)            LL                        ;# tipo de capa de enlace
set opt(ant)           Antenna/OmniAntenna       ;# modelo de antena
set opt(x)             500                       ;# Dimensión en el eje X
set opt(y)             500                       ;# Dimensión en el eje Y
set opt(ifqlen)        50                       ;# paquete máximo en ifq
set opt(nn)            10                       ;# cantidad de nodos móviles
set opt(connections)   5                       ;# Nro de conexiones por los nodos
set opt(stop)         10                       ;# tiempo de finalización
```

```
set opt(dataRate)      [expr 1.0*256*8]      ;# tamaño de paquete (256 bytes)
set opt(rp)            PROCCDIMI            ;# protocolo de ruteo

set val(coords)       2      ;# Cantidad de coordinadores
set val(DCS)          6      ;# Cantidad de DCS
set val(DCL)          4      ;# Cantidad de DCL
set val(cd_mode)      0      ;# Modo de operacion (0 para descubrimiento y 1 para asign

#=====
#      Inicializacion
#=====
# Creación del simulador.
set ns_ [new Simulator]

# Creación del objeto topográfico
set topo [new Topography]

# Nombre para los archivos de traza y de animación.
set opt(fn) "wireless_PROCCDIMI"

# Crea el archivo de trazas y se abre para escritura (w) wireless_PROCCDIMI.tr
set tracefd [open $opt(fn).tr w]

# Crea el archivo de animación y se abre para escritura (w) wireless_PROCCDIMI.nam
set namtrace [open $opt(fn).nam w]

$ns_ trace-all $tracefd      ;# el trazado se envía al archivo
$ns_ namtrace-all-wireless $namtrace $opt(x) $opt(y)

# Declaración de la función que se ejecuta cuando el programa termina.
```

```
proc finish {} {
global ns_ tracefd namtrace
$ns_ flush-trace
close $tracefd
close $namtrace
exit 0
}

# Definición de la topología
$topo load_flatgrid $opt(x) $opt(y)

# Crea al God(Generate Operations Director)
set god_ [create-god $opt(nn)]

# Define como deben ser creados y configurados los nodos móviles.
# Estas variables han sido declaradas al principio de este archivo.
$ns_ node-config -adhocRouting $opt(adhocRouting) \
                -llType $opt(ll) \
                -macType $opt(mac) \
                -ifqType $opt(ifq) \
                -ifqLen $opt(ifqlen) \
                -antType $opt(ant) \
                -propType $opt(prop) \
                -phyType $opt(netif) \
                -energyModel "EnergyModel" \
                -initialEnergy 3.0 \
                -txPower 0.9 \
                -rxPower 0.5 \
                -idlePower 0.45 \
                -sleepPower 0.05 \
                -channelType $opt(chan) \
                -topoInstance $topo \
                -agentTrace ON \
                -movementTrace ON \
                -routerTrace ON \
                -macTrace ON
```

```
# Se crean los 10 nodos que intervienen en la simulación.
for {set i 0} {$i < $opt(nn)} {incr i} {
set node_($i) [$ns_ node]
}

# Se les indican a los 10 nodos sus posiciones iniciales.
for {set i 0} {$i < $opt(nn)} {incr i} {
$node_($i) set X_ [expr rand()*500]
    $node_($i) set Y_ [expr rand()*500]
    $node_($i) set Z_ 0

if { $i < $val(DCS) } {
$node_($i) set FLAG_ 1      ;# setea al nodo como DCS
} else {
$node_($i) set FLAG_ 0      ;# setea al nodo como DCL
}

}

#Se crean las conexiones UDP y agentes de aplicación CBR
# para que los 10 nodos establezcan comunicación entre ellos y
# se define el instante de tiempo en que debe comenzar el envío de paquetes.
for {set i 0} {$i < $opt(connections)} {incr i} {

    #Creación y configuración de la conexión UDP
    set udp_($i) [new Agent/UDP]
    $ns_ attach-agent $node_($i) $udp_($i)
    set null_($i) [new Agent/Null]
    $ns_ attach-agent $node_([expr $i+2]) $null_($i)
    $ns_ connect $udp_($i) $null_($i)

    #Creación de agentes, aplicación CBR sobre UDP
    set cbr_($i) [new Application/Traffic/CBR]
    $cbr_($i) attach-agent $udp_($i)
```

```
$cbr_($i) set type_ CBR
$cbr_($i) set packet_size_ 256
$cbr_($i) set rate_ $opt(dataRate)
$cbr_($i) set random_ false

set j [expr $i+ $val(connections)]
    $node_($j) set FLAG_ 2                ;# Le indica al nodo que es un coordinador
$node_($j) set CLUSTER_ID_ $i          ;# Le indica al nodo el cluster id
$node_($j) set CD_MODE_ $val(cd_mode) ;# Le indica el modo de operación al coordinador

    $ns_ at 0.0 "$cbr_($i) start" ;# se indica que comience a los 0.0 segs.
    $ns_ at $val(stop) "$cbr_($i) stop"
}

# Asigna destinos aleatorios a los 10 nodos de la simulación.
for {set j 0} {$j < 10} {incr j} {
    for {set i 0} {$i < $opt(nn)} {incr i} {
        set xx_ [expr rand()*$opt(x)]
        set yy_ [expr rand()*$opt(y)]
        $ns_ at .01 "$node_($i) setdest $xx_ $yy_ 15.0"
    }
}

# Cuando se alcanza el tiempo limite de la simulación se indica a
# todos los nodos que la misma ha finalizado.
for {set i 0} {$i < $opt(nn) } {incr i} {
    $ns_ at $opt(stop) "$node_($i) reset";
}

# Cuando se alcanza el tiempo 10 segundos se ejecuta la función finish.
$ns_ at $opt(stop) "finish"

$ns_ run
```

Modo de operación asignación de tareas:

```

#=====
# Se definen todas las variables globales de configuración
#=====

set opt(chan)          Channel/WirelessChannel    ;# tipo de canal
set opt(prop)          Propagation/TwoRayGround   ;# modelo de propagación de radio
set opt(netif)         Phy/WirelessPhy           ;# tipo de interface de red
set opt(mac)           Mac/802_11                ;# tipo de MAC
set opt(ifq)           Queue/DropTail/PriQueue   ;# tipo de cola de paquetes
set opt(ll)            LL                        ;# tipo de capa de enlace
set opt(ant)           Antenna/OmniAntenna       ;# modelo de antena
set opt(x)             500                       ;# Dimensión en el eje X
set opt(y)             500                       ;# Dimensión en el eje Y
set opt(ifqlen)        50                       ;# paquete máximo en ifq
set opt(nn)            10                       ;# cantidad de nodos móviles
set opt(connections)   5                       ;# Nro de conexiones por los nodos
set opt(stop)          10                       ;# tiempo de finalización
set opt(dataRate)      [expr 1.0*256*8]         ;# tamaño de paquete (256 bytes)
set opt(rp)            PROCCDIMI                ;# protocolo de ruteo

set val(coords)        2           ;# Cantidad de coordinadores
set val(DCS)           6           ;# Cantidad de DCS
set val(DCL)           4           ;# Cantidad de DCL
set val(cd_mode)       1           ;# Modo de operacion (0 para descubrimiento y 1 para asign
set val(max_tasks)     20          ;# Cantidad max. de tareas por nodo.

#=====
#           Inicializacion
#=====
# Creación del simulador.
set ns_ [new Simulator]

```

```
# Creación del objeto topográfico
set topo [new Topography]

# Nombre para los archivos de traza y de animación.
set opt(fn) "wireless_PROCCDIMI"

# Crea el archivo de trazas y se abre para escritura (w) wireless_PROCCDIMI.tr
set tracefd [open $opt(fn).tr w]

# Crea el archivo de animación y se abre para escritura (w) wireless_PROCCDIMI.nam
set namtrace [open $opt(fn).nam w]

$ns_ trace-all $tracefd ;# el trazado se envía al archivo
$ns_ namtrace-all-wireless $namtrace $opt(x) $opt(y)

# Declaración de la función que se ejecuta cuando el programa termina.
proc finish {} {
global ns_ tracefd namtrace
$ns_ flush-trace
close $tracefd
close $namtrace
exit 0
}

# Definición de la topología
$topo load_flatgrid $opt(x) $opt(y)

# Crea al God(Generate Operations Director)
set god_ [create-god $opt(nn)]

# Define como deben ser creados y configurados los nodos móviles.
# Estas variables han sido declaradas al principio de este archivo.
```

```

$ns_ node-config -adhocRouting $opt(adhocRouting) \
    -llType $opt(ll) \
    -macType $opt(mac) \
    -ifqType $opt(ifq) \
    -ifqLen $opt(ifqlen) \
    -antType $opt(ant) \
    -propType $opt(prop) \
    -phyType $opt(netif) \
    -energyModel "EnergyModel" \
    -initialEnergy 3.0 \
    -txPower 0.9 \
    -rxPower 0.5 \
    -idlePower 0.45 \
    -sleepPower 0.05 \
        -channelType $opt(chan) \
    -topoInstance $topo \
    -agentTrace ON \
        -movementTrace ON \
        -routerTrace ON \
        -macTrace ON

# Se crean los 10 nodos que intervienen en la simulación.
for {set i 0} {$i < $opt(nn)} {incr i} {
    set node_($i) [$ns_ node]
}

# Se les indican a los 10 nodos sus posiciones iniciales.
for {set i 0} {$i < $opt(nn)} {incr i} {
    $node_($i) set X_ [expr rand()*500]
    $node_($i) set Y_ [expr rand()*500]
    $node_($i) set Z_ 0

    if { $i < $val(DCS) } {
        $node_($i) set FLAG_ 1      ;# setea al nodo como DCS
    } else {

```

```

$node_($i) set FLAG_ 0          ;# setea al nodo como DCL
}

}

#Se crean las conexiones UDP y agentes de aplicación CBR
# para que los 10 nodos establezcan comunicación entre ellos y
# se define el instante de tiempo en que debe comenzar el envío de paquetes.
for {set i 0} {$i < $opt(connections)} {incr i} {

    #Creación y configuracion de la conexión UDP
    set udp_($i) [new Agent/UDP]
    $ns_ attach-agent $node_($i) $udp_($i)
    set null_($i) [new Agent/Null]
    $ns_ attach-agent $node_([expr $i+2]) $null_($i)
    $ns_ connect $udp_($i) $null_($i)

    #Creación de agentes, aplicación CBR sobre UDP
    set cbr_($i) [new Application/Traffic/CBR]
    $cbr_($i) attach-agent $udp_($i)
    $cbr_($i) set type_ CBR
    $cbr_($i) set packet_size_ 256
    $cbr_($i) set rate_ $opt(dataRate)
    $cbr_($i) set random_ false

set j [expr $i+ $val(connections)]
    $node_($j) set FLAG_ 2          ;# Le indica al nodo que es un coordinado
    $node_($j) set CLUSTER_ID_ $i    ;# Le indica al nodo el cluster id
    $node_($j) set CD_MODE_ $val(cd_mode) ;# Le indica el modo de operación al coordina
    $node_($j) set TASK_NRO_ $i      ;# Indica el nro de tarea a resolver distribu
    $node_($j) set TN_DATA_ $i       ;# Indica el dato
    $node_($j) set TN_LENGTH_ $i*32*8 ;# Indica el tamaño de dato
    $node_($j) set TN_TYPE_ $i       ;# Indica el tipo de dato
    $node_($j) set TASK_MAX_NRO_ $val(max_tasks) ;# Cantidad max. de tareas que resuelve

```

```
$ns_ at 0.0 "$cbr_($i) start" ;# se indica que comience a los 0.0 segs.
$ns_ at $val(stop) "$cbr_($i) stop"
}

# Asigna destinos aleatorios a los 10 nodos de la simulación.
for {set j 0} {$j < 10} {incr j} {
  for {set i 0} {$i < $opt(nn)} {incr i} {
    set xx_ [expr rand()*$opt(x)]
    set yy_ [expr rand()*$opt(y)]
    $ns_ at .01 "$node_($i) setdest $xx_ $yy_ 15.0"
  }
}

# Cuando se alcanza el tiempo limite de la simulación se indica a
# todos los nodos que la misma ha finalizado.
for {set i 0} {$i < $opt(nn) } {incr i} {
  $ns_ at $opt(stop) "$node_($i) reset";
}

# Cuando se alcanza el tiempo 10 segundos se ejecuta la función finish.
$ns_ at $opt(stop) "finish"

$ns_ run
```

Apéndice B

Script AWK

En el presente apéndice se detalla el script escrito en el lenguaje AWK que se utilizó para obtener los resultados expuestos en el capítulo 5. Este código procesa los valores para realizar las pruebas de rendimiento a partir de las trazas generadas por el simulador NS2 (*Network Simulator version 2*), el formato de las trazas es mostrada en el apéndice C.

```
function abs(v) {return v < 0 ? -v : v}
BEGIN {
    recvdSize = 0
    startTime = 0
    stopTime = 0
    lastTime = -1
    sent=0
    receive=0
    agtCount=0
    rtrCount=0
    jitter=0
    startCount=0
    endCount=0
    totalenergy=0.0
    initialenergy=100.0

    print "#protocol >> "throughput.txt";
    print "#protocol >> "loss.txt";
    print "#protocol >> "PDR.txt";
    print "#protocol >> "load.txt";
    print "#protocol >> "overhead.txt";
```

```
    print "#protocol >> "renergy.txt";
    print "#protocol >> "jitter.txt";
    print "#protocol >> "stats.txt";
    print "#protocol >> "stats_"protocol".txt";
}

{
    event = $1
    time = $2
    node_id = $3
    pkt_size = $8
    level = $4

    if (event == "r" || event == "d" || event == "s" || event
        == "f") {
        energy = $14
    }

    if (event == "N") {
        node_id = $5
        energy = $7
    }

    totalenergy=0.0
    finalenergy[int(node_id)] = energy
    for (i in finalenergy) {
        consumenergy[i] = initialenergy - finalenergy[i]
        totalenergy += consumenergy[i]
    }

    if (( $1 == "r" ) && ( $7 == "cbr" || $7 == "tcp" ) && ( $4
        == "AGT" )) agtCount++;

    if (( $1 == "s" ) && $4 == "RTR") rtrCount++;

    if (level == "AGT" && event == "s" && $7 == "cbr") {
```

```
sent++;
if (!startTime || (time < startTime)) {
    startTime = time
}
startTimeArr[startCount] = $2;
startCount++;
}

if (level == "AGT" && event == "r" && $7 == "cbr") {
    receive++;
    if (time > stopTime) {
        stopTime = time
    }
    recvdSize += pkt_size

    endTimeArr[endCount] = $2;
    prev_delay[endCount]= $2;
    endCount++;

count = 0;
jitter = 0;
for(i=0;i<startCount;i++) {
    if(endTimeArr[i] > 0){
        delay[i] = endTimeArr[i] - startTimeArr[i];
        if(prev_delay[i] > 0){
            jitter += abs(delay[i] - prev_delay
                [i]);
            prev_delay[i] = delay[i];
        }
        count++;
    }
    else
    {
        delay[i] = -1;
    }
}
}
```

```

jitter = jitter/count;

if (int(lastTime) < int(stopTime) && (recvdSize/(time
-startTime))*(8/1000) < 100 ) {

printf int(stopTime) " " (recvdSize/(time-
startTime))*(8/1000) "\n">> "throughput.txt";
printf int(stopTime) " " sent-receive "\n">> "loss
.txt";
printf int(stopTime) " " (receive/sent)
*100 "\n"> "PDR.txt";
printf int(stopTime) " " ((agtCount*1.0)/rtrCount)
"\n">> "overhead.txt";
printf int(stopTime) " " (jitter* 1000) "\
n">> "jitter.txt";
printf("%d %.7f\n", int(stopTime),
totalenergy/totalnodes ) >> "renergy.txt
";
lastTime=stopTime

}

}

if (level == "AGT" && event == "d" && $7 == "cbr") {
endTimeArr[$35] = -1;
}

}

END {
printf "\n\n" >> "throughput.txt";
printf "\n\n" >> "loss.txt";
printf "\n\n" >> "PDR.txt";
}

```

```

printf "\n\n" >> "load.txt";
printf "\n\n" >> "overhead.txt";
printf "\n\n" >> "jitter.txt";
printf "\n\n" >> "renergy.txt";

printf("Sent\t %d\n",sent) >> "stats_"protocol".txt";
printf("Received %d\n",receive) >> "stats_"protocol".txt
";
printf("Dropped %d\n",sent-receive) >> "stats_"protocol".
txt";
printf("PDR %.2f\n", (receive/sent)*100) >> "stats_"
protocol".txt";
printf("Average Throughput[kbps] = %.2f\tStartTime=%.2f\t
tStopTime = %.2f\n", (recvdSize/(stopTime-startTime))
*(8/1000),startTime,stopTime) >> "stats_"protocol".txt
";
printf("Normalized Load\t %0.3f\n",((agtCount*1.0)/
rtrCount)) >> "stats_"protocol".txt";
printf("Energy =%d %.7f\n", int(stopTime), totalenergy/
totalnodes ) >> "stats_"protocol".txt";
printf("jitter= %0.3f\n", (jitter* 1000)) >> "stats_"
protocol".txt";
printf "\n\n" >> "stats_"protocol".txt";

printf("%d %0.3f\n", totalnodes,((agtCount*1.0)/rtrCount)
) >> "../overhead_"protocol".txt";
printf("%d %.2f\n", totalnodes,(receive/sent)*100) >>
"../PDR_"protocol".txt";
printf("%d %.2f\n", totalnodes,((recvdSize/(stopTime-
startTime))*(8/1000))) >> "../Throughput_"protocol".
txt";
printf("%d %.7f\n", totalnodes,(totalenergy/totalnodes) )
>> "../renergy_"protocol".txt";
printf("%d %.7f\n", totalnodes, (jitter* 1000)) >> "../
jitter_"protocol".txt";

```

}

Apéndice C

Visualización de la traza de simulación

A continuación se muestra el formato de salida de las trazas generadas por PROCCDIMI (PROtocolo para el Cálculo Científico Distribuido sobre clientes Móviles Indeterminados en MANETs) al realizar las simulaciones sobre NS2 (*Network Simulator version 2*) según la configuración establecida en el capítulo 4 para poder realizar las pruebas de rendimientos definidas en el capítulo 5.

El archivo de texto que almacena este formato particular se obtiene a partir de la ejecución de los scripts OTCL (*Object-oriented Tool Command Language*) incluidos en el apéndice A.

```
r 30.120397091 _10_ MAC --- 0 PROCCDIMI 52 [0 ffffffff 1a
 800] [energy 96.505415 ei 0.000 es 0.000 et 0.175 er
 3.320] ----- [26:255 -1:255 28 0] [0x2 3 5 [38 0] [13
 12]] (REQUEST)
r 30.120397122 _35_ MAC --- 0 PROCCDIMI 52 [0 ffffffff 1a
 800] [energy 96.605260 ei 0.000 es 0.000 et 0.079 er
 3.315] ----- [26:255 -1:255 28 0] [0x2 3 5 [38 0] [13
 12]] (REQUEST)
r 30.120397131 _43_ MAC --- 0 PROCCDIMI 52 [0 ffffffff 1a
 800] [energy 96.548554 ei 0.000 es 0.000 et 0.078 er
 3.374] ----- [26:255 -1:255 28 0] [0x2 3 5 [38 0] [13
 12]] (REQUEST)
r 30.120397182 _42_ MAC --- 0 PROCCDIMI 52 [0 ffffffff 1a
 800] [energy 96.514417 ei 0.000 es 0.000 et 0.155 er
 3.331] ----- [26:255 -1:255 28 0] [0x2 3 5 [38 0] [13
 12]] (REQUEST)
r 30.120397209 _4_ MAC --- 0 PROCCDIMI 52 [0 ffffffff 1a
 800] [energy 96.484950 ei 0.000 es 0.000 et 0.221 er
```

```
3.294] ----- [26:255 -1:255 28 0] [0x2 3 5 [38 0] [13
12]] (REQUEST)
r 30.120397218 _31_ MAC --- 0 PROCCDIMI 52 [0 ffffffff 1a
800] [energy 96.534538 ei 0.000 es 0.000 et 0.109 er
3.356] ----- [26:255 -1:255 28 0] [0x2 3 5 [38 0] [13
12]] (REQUEST)
r 30.120397219 _11_ MAC --- 0 PROCCDIMI 52 [0 ffffffff 1a
800] [energy 96.506502 ei 0.000 es 0.000 et 0.172 er
3.321] ----- [26:255 -1:255 28 0] [0x2 3 5 [38 0] [13
12]] (REQUEST)
r 30.120397221 _34_ MAC --- 0 PROCCDIMI 52 [0 ffffffff 1a
800] [energy 96.535990 ei 0.000 es 0.000 et 0.106 er
3.358] ----- [26:255 -1:255 28 0] [0x2 3 5 [38 0] [13
12]] (REQUEST)
r 30.120397226 _14_ MAC --- 0 PROCCDIMI 52 [0 ffffffff 1a
800] [energy 96.439008 ei 0.000 es 0.000 et 0.324 er
3.237] ----- [26:255 -1:255 28 0] [0x2 3 5 [38 0] [13
12]] (REQUEST)
r 30.120397229 _15_ MAC --- 0 PROCCDIMI 52 [0 ffffffff 1a
800] [energy 96.553837 ei 0.000 es 0.000 et 0.175 er
3.271] ----- [26:255 -1:255 28 0] [0x2 3 5 [38 0] [13
12]] (REQUEST)
r 30.120397241 _48_ MAC --- 0 PROCCDIMI 52 [0 ffffffff 1a
800] [energy 96.559814 ei 0.000 es 0.000 et 0.126 er
3.314] ----- [26:255 -1:255 28 0] [0x2 3 5 [38 0] [13
12]] (REQUEST)
r 30.120397243 _23_ MAC --- 0 PROCCDIMI 52 [0 ffffffff 1a
800] [energy 96.566291 ei 0.000 es 0.000 et 0.158 er
3.276] ----- [26:255 -1:255 28 0] [0x2 3 5 [38 0] [13
12]] (REQUEST)
r 30.120397262 _47_ MAC --- 0 PROCCDIMI 52 [0 ffffffff 1a
800] [energy 96.512699 ei 0.000 es 0.000 et 0.158 er
3.329] ----- [26:255 -1:255 28 0] [0x2 3 5 [38 0] [13
12]] (REQUEST)
r 30.120397275 _33_ MAC --- 0 PROCCDIMI 52 [0 ffffffff 1a
800] [energy 96.519525 ei 0.000 es 0.000 et 0.143 er
```

```
3.337] ----- [26:255 -1:255 28 0] [0x2 3 5 [38 0] [13
12]] (REQUEST)
r 30.120397280 _25_ MAC --- 0 PROCCDIMI 52 [0 ffffffff 1a
800] [energy 96.515767 ei 0.000 es 0.000 et 0.152 er
3.333] ----- [26:255 -1:255 28 0] [0x2 3 5 [38 0] [13
12]] (REQUEST)
r 30.120397281 _40_ MAC --- 0 PROCCDIMI 52 [0 ffffffff 1a
800] [energy 96.532181 ei 0.000 es 0.000 et 0.115 er
3.353] ----- [26:255 -1:255 28 0] [0x2 3 5 [38 0] [13
12]] (REQUEST)
r 30.120397295 _46_ MAC --- 0 PROCCDIMI 52 [0 ffffffff 1a
800] [energy 96.536407 ei 0.000 es 0.000 et 0.105 er
3.359] ----- [26:255 -1:255 28 0] [0x2 3 5 [38 0] [13
12]] (REQUEST)
r 30.120397296 _28_ MAC --- 0 PROCCDIMI 52 [0 ffffffff 1a
800] [energy 96.532895 ei 0.000 es 0.000 et 0.113 er
3.354] ----- [26:255 -1:255 28 0] [0x2 3 5 [38 0] [13
12]] (REQUEST)
r 30.120397319 _1_ MAC --- 0 PROCCDIMI 52 [0 ffffffff 1a
800] [energy 96.499201 ei 0.000 es 0.000 et 0.189 er
3.312] ----- [26:255 -1:255 28 0] [0x2 3 5 [38 0] [13
12]] (REQUEST)
r 30.120397351 _0_ MAC --- 0 PROCCDIMI 52 [0 ffffffff 1a
800] [energy 96.526822 ei 0.000 es 0.000 et 0.127 er
3.347] ----- [26:255 -1:255 28 0] [0x2 3 5 [38 0] [13
12]] (REQUEST)
r 30.120397358 _45_ MAC --- 0 PROCCDIMI 52 [0 ffffffff 1a
800] [energy 96.575992 ei 0.000 es 0.000 et 0.080 er
3.344] ----- [26:255 -1:255 28 0] [0x2 3 5 [38 0] [13
12]] (REQUEST)
r 30.120397376 _22_ MAC --- 0 PROCCDIMI 52 [0 ffffffff 1a
800] [energy 96.511706 ei 0.000 es 0.000 et 0.161 er
3.328] ----- [26:255 -1:255 28 0] [0x2 3 5 [38 0] [13
12]] (REQUEST)
r 30.120397389 _13_ MAC --- 0 PROCCDIMI 52 [0 ffffffff 1a
800] [energy 96.543515 ei 0.000 es 0.000 et 0.089 er
```

```
3.367] ----- [26:255 -1:255 28 0] [0x2 3 5 [38 0] [13
12]] (REQUEST)
r 30.120397392 _21_ MAC --- 0 PROCCDIMI 52 [0 ffffffff 1a
800] [energy 96.505773 ei 0.000 es 0.000 et 0.174 er
3.320] ----- [26:255 -1:255 28 0] [0x2 3 5 [38 0] [13
12]] (REQUEST)
r 30.120397409 _32_ MAC --- 0 PROCCDIMI 52 [0 ffffffff 1a
800] [energy 96.641192 ei 0.000 es 0.000 et 0.062 er
3.297] ----- [26:255 -1:255 28 0] [0x2 3 5 [38 0] [13
12]] (REQUEST)
r 30.120397418 _39_ MAC --- 0 PROCCDIMI 52 [0 ffffffff 1a
800] [energy 96.549323 ei 0.000 es 0.000 et 0.095 er
3.356] ----- [26:255 -1:255 28 0] [0x2 3 5 [38 0] [13
12]] (REQUEST)
r 30.120397443 _19_ MAC --- 0 PROCCDIMI 52 [0 ffffffff 1a
800] [energy 96.484570 ei 0.000 es 0.000 et 0.222 er
3.294] ----- [26:255 -1:255 28 0] [0x2 3 5 [38 0] [13
12]] (REQUEST)
r 30.120397449 _30_ MAC --- 0 PROCCDIMI 52 [0 ffffffff 1a
800] [energy 96.539729 ei 0.000 es 0.000 et 0.098 er
3.363] ----- [26:255 -1:255 28 0] [0x2 3 5 [38 0] [13
12]] (REQUEST)
r 30.120397491 _20_ MAC --- 0 PROCCDIMI 52 [0 ffffffff 1a
800] [energy 96.534717 ei 0.000 es 0.000 et 0.109 er
3.356] ----- [26:255 -1:255 28 0] [0x2 3 5 [38 0] [13
12]] (REQUEST)
r 30.120397518 _29_ MAC --- 0 PROCCDIMI 52 [0 ffffffff 1a
800] [energy 96.531810 ei 0.000 es 0.000 et 0.115 er
3.353] ----- [26:255 -1:255 28 0] [0x2 3 5 [38 0] [13
12]] (REQUEST)
r 30.120397566 _49_ MAC --- 0 PROCCDIMI 52 [0 ffffffff 1a
800] [energy 96.585920 ei 0.000 es 0.000 et 0.057 er
3.357] ----- [26:255 -1:255 28 0] [0x2 3 5 [38 0] [13
12]] (REQUEST)
```

```
r 132.517204757 _25_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
 800] [energy 88.357113 ei 0.000 es 0.000 et 0.368 er
11.275] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
r 132.517204782 _7_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
 800] [energy 88.356758 ei 0.000 es 0.000 et 0.369 er
11.274] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
r 132.517204795 _3_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
 800] [energy 88.290341 ei 0.000 es 0.000 et 0.736 er
10.974] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
r 132.517204803 _11_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
 800] [energy 88.301639 ei 0.000 es 0.000 et 0.493 er
11.205] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
r 132.517204804 _2_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
 800] [energy 88.351391 ei 0.000 es 0.000 et 0.381 er
11.268] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
r 132.517204807 _10_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
 800] [energy 88.241208 ei 0.000 es 0.000 et 0.629 er
11.130] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
r 132.517204828 _0_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
 800] [energy 88.320292 ei 0.000 es 0.000 et 0.451 er
11.229] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
r 132.517204841 _5_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
 800] [energy 88.409271 ei 0.000 es 0.000 et 0.330 er
11.261] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
r 132.517204849 _20_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
 800] [energy 88.330563 ei 0.000 es 0.000 et 0.428 er
11.242] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
```

```
r 132.517204855 _30_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
 800] [energy 88.376394 ei 0.000 es 0.000 et 0.325 er
11.299] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
r 132.517204856 _42_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
 800] [energy 88.353759 ei 0.000 es 0.000 et 0.376 er
11.271] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
r 132.517204858 _33_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
 800] [energy 88.362475 ei 0.000 es 0.000 et 0.356 er
11.281] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
r 132.517204862 _13_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
 800] [energy 88.285673 ei 0.000 es 0.000 et 0.529 er
11.185] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
r 132.517204864 _39_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
 800] [energy 88.395142 ei 0.000 es 0.000 et 0.302 er
11.303] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
r 132.517204876 _8_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
 800] [energy 88.265056 ei 0.000 es 0.000 et 0.575 er
11.160] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
r 132.517204880 _12_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
 800] [energy 88.284642 ei 0.000 es 0.000 et 0.531 er
11.184] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
r 132.517204888 _17_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
 800] [energy 88.414297 ei 0.000 es 0.000 et 0.502 er
11.084] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
r 132.517204891 _38_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
 800] [energy 88.447675 ei 0.000 es 0.000 et 0.284 er
11.268] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
```

```
r 132.517204894 _1_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
800] [energy 88.200778 ei 0.000 es 0.000 et 0.720 er
11.079] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
r 132.517204911 _29_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
800] [energy 88.374396 ei 0.000 es 0.000 et 0.329 er
11.296] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
r 132.517204918 _46_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
800] [energy 88.384170 ei 0.000 es 0.000 et 0.307 er
11.309] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
r 132.517204919 _21_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
800] [energy 88.230372 ei 0.000 es 0.000 et 0.653 er
11.116] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
r 132.517204927 _9_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
800] [energy 88.297485 ei 0.000 es 0.000 et 0.502 er
11.200] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
r 132.517204937 _16_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
800] [energy 88.316461 ei 0.000 es 0.000 et 0.478 er
11.205] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
r 132.517204937 _28_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
800] [energy 88.379143 ei 0.000 es 0.000 et 0.319 er
11.302] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
r 132.517204938 _32_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
800] [energy 88.440623 ei 0.000 es 0.000 et 0.373 er
11.187] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
r 132.517204954 _4_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
800] [energy 88.193982 ei 0.000 es 0.000 et 0.735 er
11.071] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
```

```
r 132.517204956 _24_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
 800] [energy 88.295021 ei 0.000 es 0.000 et 0.508 er
11.197] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
r 132.517204980 _22_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
 800] [energy 88.310094 ei 0.000 es 0.000 et 0.474 er
11.216] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
r 132.517204981 _45_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
 800] [energy 88.422556 ei 0.000 es 0.000 et 0.285 er
11.293] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
r 132.517204984 _27_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
 800] [energy 88.385130 ei 0.000 es 0.000 et 0.305 er
11.310] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
r 132.517204990 _43_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
 800] [energy 88.391519 ei 0.000 es 0.000 et 0.291 er
11.318] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
r 132.517204997 _37_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
 800] [energy 88.331048 ei 0.000 es 0.000 et 0.427 er
11.242] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
r 132.517205001 _14_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
 800] [energy 88.152342 ei 0.000 es 0.000 et 0.829 er
11.019] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
r 132.517205002 _18_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
 800] [energy 88.296963 ei 0.000 es 0.000 et 0.503 er
11.200] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
r 132.517205003 _26_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
 800] [energy 88.385748 ei 0.000 es 0.000 et 0.304 er
11.311] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
```

```
r 132.517205004 _19_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
 800] [energy 88.279788 ei 0.000 es 0.000 et 0.542 er
11.178] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
r 132.517205010 _48_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
 800] [energy 88.385511 ei 0.000 es 0.000 et 0.378 er
11.236] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
r 132.517205015 _6_ MAC --- 0 PROCCDIMI 56 [0 ffffffff 29
 800] [energy 88.286892 ei 0.000 es 0.000 et 0.526 er
11.187] ----- [41:255 -1:255 30 0] [0x12 1 [127 13]
0.000000] (HELLO)
```

