

République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique

UNIVERSITE SAIDA - Dr. MOULAY Tahar

FACULTE : TECHNOLOGIE

DEPARTEMENT : INFORMATIQUE



MEMOIRE DE MASTER

OPTION :

Master 2 RISR

Thème

**A Stochastic Process based Routing Algorithm for
Wireless Ad Hoc Networks**

Présenté par :

HARTANI Ramdane Rached Anis.

AMMARI Khadidja.

Encadré par :

Dr. BENAMARA Djillali

Promotion : 2019-2020



Remerciement

*Nous remercions dieu tout puissant de
la patience et de la volonté qu'il nous a
donné pour réaliser ce projet de fin
D'étude.*

*Nous tenons à exprimer mes sincères remerciements à
notre encadreur monsieur Benamara Djilali pour son aide,
ses conseils, son encouragement et sa disponibilité dans ce
projet*

*Nous saisissons cette occasion pour exprimer notre
gratitude à tous les professeurs et tout le personnel du
Département pour leur aide et leur soutien.*

Dédicaces

Que ce travail témoigne de mes respects :

A mes parents

Pour l'encouragement, le soutien et l'attention incessants, ils ont pu créer le climat affectueux et propice à la poursuite de mes études.

Aucune dédicace ne pourrait exprimer mon respect, ma considération et mes profonds sentiments envers eux.

Je prie le bon Dieu de les bénir, de veiller sur eux, en espérant qu'ils seront toujours fiers de moi.

A mes sœurs et mes frères.

A la famille Hartani et Ammari.

Ils vont trouver ici l'expression de mes sentiments de respect et de Reconnaissance pour le soutien qu'ils n'ont cessé de me porter.

A tous mes professeurs :

Pour m'avoir apporté un soutien sans faille et des encouragements continus tout au long de mes années d'étude et de réalisation de ce projet

A tous mes amis et mes collègues :

Pour leur aimable coopération et leurs encouragements qui nous aident à mener à bien ce projet.

Table des matières

0.1	Introduction Générale	5
1	Réseaux Ad-hoc	6
1.1	Introduction	6
1.2	Les Catégories Des Réseaux Sans Fil	6
1.2.1	Selon La Zone De Couverture :	7
1.2.2	Selon L'infrastructure :	8
1.3	LE routage	13
1.3.1	Le routage dans les réseaux ad hoc :	13
1.3.2	Les défis de routage dans les réseaux ad hoc mobiles :	14
1.3.3	LES PROTOCOLES DE ROUTAGE AD-HOC :	14
1.3.4	Classification des protocoles de routage ad-hoc :	16
1.3.5	Algorithmes de routage :	19
1.4	Les Processus léatoires :	23
1.4.1	Différents types de processus stochastiques :	24
1.4.2	Processus de branchement :	25
1.5	Conclusion :	26
2	Cardinalité De Multidiffusion Et Sélection intelligente :	27
2.1	Introduction :	27
2.2	Contexte :	28
2.3	Travaux connexes :	28
2.3.1	Flooding :	28
2.3.2	la méthode des potins [the gossip method] :	28
2.3.3	Flooding probabiliste :	29
2.3.4	Le routage de multicast :	29
2.3.5	Description et hypothèses :	29
2.3.6	Processus de cardinalité multicast (MC) :	30
2.3.7	Processus de sélection intelligente (Smart Selection) :	32
2.3.8	Cardinalité De Multidiffusion Et Sélection Intelligente :	32
3	Implémentation	35
3.1	Introduction :	35
3.2	Outils de développement :	35

3.2.1	Eclipse (@ www.eclipse.org) :	35
3.2.2	Langage de programmation (Java) :	36
3.2.3	Syntaxe :	37
3.2.4	WindowBuilder :	38
3.3	Implémentation de l'application :	39
3.3.1	Interfaces de l'application :	39
3.3.2	Quelque exemple de code source :	40
3.4	Conclusion :	41

Table des figures

1.1	Classification des réseaux sans fils.	7
1.2	les réseaux sans fils selon la zone de couverture.. . . .	8
1.3	mode infrastructure et mode Ad Hoc.	8
1.4	domaines d'application des réseaux Ad Hoc.	12
1.5	Classification des protocoles de routage MANET.	16
1.6	Un réseau à six nœuds.	19
1.7	Algorithme Bellman-Ford (la ligne continue indique un lien direct ; la ligne pointillée indique la distance)..	21
1.8	Algorithme de Dijkstra	23
1.9	réalisation du processus stochastique	24
1.10	Un exemple de l'arborescence résultante du processus de branchement de Galton-Watson	25
3.1	fenetre de programmation Sur Eclipse..	36
3.2	architecture exécutable Code java..	37
3.3	projet Java Main..	38
3.4	Interface d'accueil d'application.	39
3.5	Code source pour dessiner des nœuds aléatoires..	40
3.6	Code source pour déplacer le nœud aléatoire.	40
3.7	l'exécution de l'algorithme.	41

Liste Des Abréviation

AODV	Ad hoc On-Demand Distance Vector
BLR	Boucle Locale Radio
BSS	Basic Service Set
DSDV	Destination Sequenced Distance Vector
DSR	Dynamic Source Routing
ESS	Extended Service Set
GPRS	General Packet Radio Services
GSM	Global System For Mobile Communication
LAN	Local-Area Network
LMR	Label-Based Multipath Routing.
M2M	Machine A Machine.
MANET	Mobile Ad Hoc Network
MC	Multicast Cardinality
MCSS	Multicast Cardinality And Smart Selection
MPRs	Multipoint Relays
MRL	Message Retransmission List
OLSR	Optimized Link State Routing
PAN	Personal Area Network
QoS	Quality Of Service
RIP	Routing Information Protocol
SIS	Spontaneous Information System.
SS	Smart Selection
WAP	Wireless Access Point
WLAN	Wireless Local Area Network
WMAN	Wireless Metropolitan Area Network
WRP	Wireless Routing Protocol.
WWAN	Wireless Wide Area Network

0.1 Introduction Générale

Les réseaux ad hoc sont devenus un sujet de recherche très actif ces dernières années, en raison du développement récent de systèmes et de protocoles sans fil améliorés. Un nombre croissant d'appareils tels que les PDA, les téléphones portables et les ordinateurs portables sont connectés à des systèmes sans fil, formant des réseaux ad hoc. Dans un réseau ad hoc, les nœuds agissent comme des clients et des serveurs, et aucune topologie fixe n'est appliquée. La flexibilité des réseaux ad hoc entraîne également une complexité accrue dans des domaines tels que le routage, la fiabilité et la sécurité, par exemple. De nombreux objectifs de conception associés peuvent être posés comme des problèmes d'optimisation combinatoire et bénéficieraient donc d'une étude détaillée de leur structure combinatoire. Le schéma de cardinalité et de sélection intelligente (MCSS) pour les réseaux ad hoc sans fil, basé sur un processus de branchement stochastique modifié, pour prendre en charge l'échange de données dans une topologie à commutation rapide.

Réseaux Ad-hoc

1.1 Introduction

Un réseau sans fil est un réseau informatique qui connecte différents systèmes ou postes entre eux par ondes radios. Le réseau sans fil peut associer à un réseau de télécommunication pour réaliser des interconnexions entre nœuds. Ces derniers permettent aux utilisateurs de profiter de tous les services traditionnels des réseaux indépendamment de leurs positions géographiques. Le rayonnement géographique des ondes est relativement limité étant donné la faible puissance d'émission des solutions matérielles actuelles. Pour cette raison, les réseaux sans fil se sont avant tout développés comme réseaux internes, propres à un bâtiment, soit comme réseau d'entreprise, soit comme réseau domestique.

1.2 Les Catégories Des Réseaux Sans Fil

Il existe différents critères pour concevoir et classer les réseaux sans fil. Les réseaux sans fil peuvent être classés selon : la zone de couverture du réseau et l'infrastructure. Au vu du premier critère il existe quatre catégories : les réseaux personnels, les réseaux locaux, le réseau métropolitain et les réseaux étendus. Le second critère qui est l'infrastructure peut diviser les réseaux sans fil en : réseaux avec infrastructures et réseaux sans infrastructure (Ad-hoc).[1].

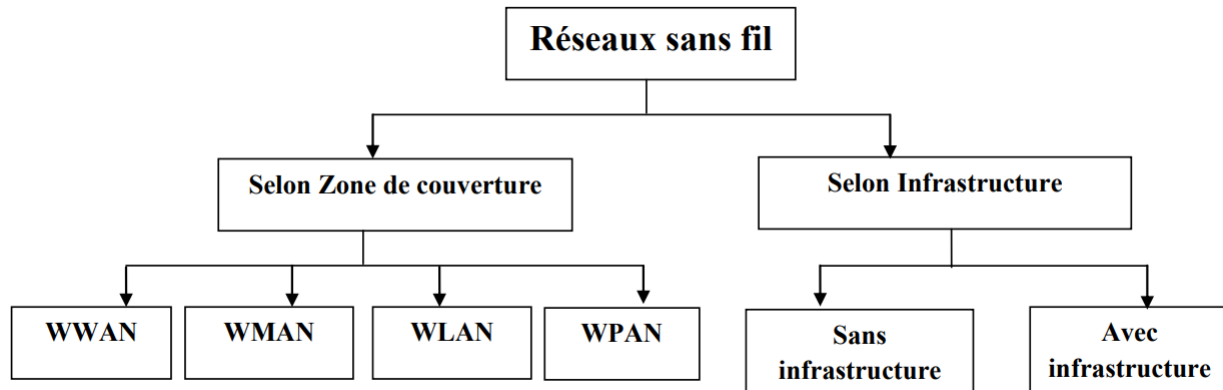


FIGURE 1.1 – Classification des réseaux sans fils.

1.2.1 Selon La Zone De Couverture :

a. Réseaux personnels ou PAN (Personal Area Network) :

C'est un réseau utilisés dans le cadre d'une utilisation personnelle. Parmi les technologies sans fil utilisées par les réseaux PAN, on note le Bluetooth, l'infrarouge (IR), ou le zigbee ...

b. Réseaux locaux sans fil (WLAN) :

Depuis le développement des normes qui offrent un haut débit, les réseaux locaux sans fil ou Wireless Local Area Network (WLAN) sont généralement utilisés à l'intérieur d'une entreprise, d'une université

c. Les réseaux métropolitains sans fil (WMAN) :

Les réseaux métropolitains sans fil ou Wireless Metropolitan Area Network (WMAN) sont aussi connus sous l'appellation de boucle locale radio (BLR). Les réseaux basés sur la technologie IEEE 802.16 ont une portée de l'ordre de quelques dizaines de kilomètres (50km de portée théorique annoncée) et un taux de transmission radio théorique pouvant atteindre 74 Mbit/s pour IEEE 802.16, plus connu sous le nom commercial de WIMAX.[2]

d. Les réseaux sans fil étendus (WWAN)

Un réseau étendu sans fil (WWAN) est un type spécifique de réseau qui envoie des signaux sans fil au-delà d'un seul bâtiment ou propriété. WWAN disponibles sont des technologies utilisant les satellites géostationnaires ou en orbite basse pour relayer l'information entre plusieurs points du globe [3]. Parmi les principales technologies, dans ce type de réseaux : GSM et GPRS.

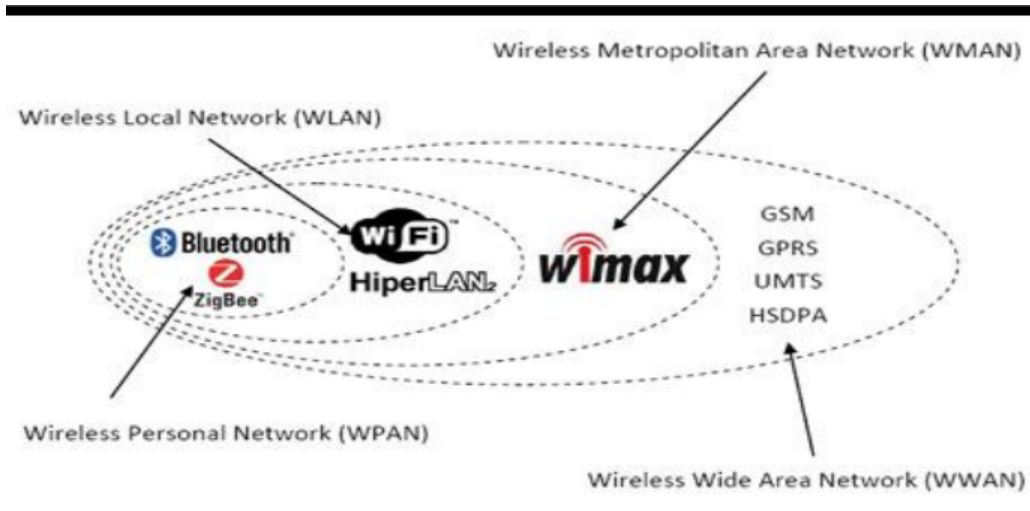


FIGURE 1.2 – les réseaux sans fils selon la zone de couverture..

1.2.2 Selon L'infrastructure :

Les progrès de la technologie de réseau sans fil et des appareils d'information portables ont engendré un nouveau paradigme informatique, appelé informatique mobile, dans lequel les utilisateurs qui transportent des appareils portables ont accès à des services d'information via une infrastructure partagée, indépendamment de leur emplacement physique ou de leur comportement de mouvement. Un tel nouvel environnement introduit de nouveaux défis techniques dans le domaine de l'accès à l'information. Les réseaux mobiles ou sans fil, peuvent être classés en deux classes : les réseaux avec infrastructure et les réseaux sans infrastructure.

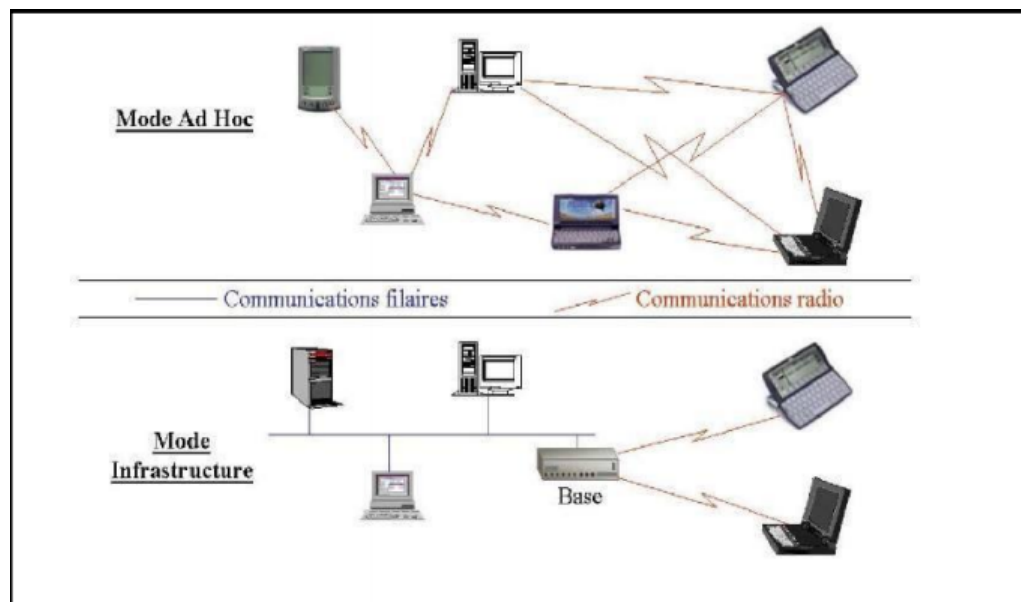


FIGURE 1.3 – mode infrastructure et mode Ad Hoc.

a. Réseaux sans fil avec infrastructure :

Les réseaux sans fil fonctionnant en mode infrastructure utilisent un ou plusieurs WAP (Wireless Access Point) pour connecter les nœuds du réseau sans fil de manière centralisée. Cette configuration est similaire à la topologie en étoile d'un réseau filaire. Vous utilisez également le mode infrastructure pour connecter des segments de réseau sans fil à des segments filaires. Si vous prévoyez de configurer un réseau sans fil pour un grand nombre de PC, ou si vous avez besoin d'un contrôle centralisé sur le réseau sans fil, le mode infrastructure est ce dont vous avez besoin. Un seul WAP desservant une zone donnée est appelé ensemble de services de base (BSS). Cette zone de service peut être étendue en ajoutant plus de points d'accès. Ceci est appelé, de manière appropriée, un ensemble de services étendus (ESS). [4]. Les réseaux sans fil fonctionnant en mode infrastructure nécessitent un peu plus de planification - comme l'emplacement des WAP pour fournir une couverture adéquate - que les réseaux en mode ad hoc, et ils fournissent un environnement stable pour les installations de réseau sans fil permanentes. Le mode infrastructure est mieux adapté aux réseaux d'entreprise ou aux réseaux qui doivent partager des ressources dédiées telles que des connexions Internet et des bases de données centralisées.

b. Réseaux sans infrastructure (ad hoc) :

Le terme " ad-hoc " implique que ce réseau est un réseau établi pour un service spécial, souvent improvisé, personnalisé aux applications. Donc, le réseau ad hoc typique est mis en place pour une période limitée. Les protocoles sont adaptés à l'application particulière (p. ex., envoyer un flux vidéo à travers le champ de bataille; déterminer si un incendie a commencé dans la forêt; établir une vidéoconférence entre trois équipes engagées dans un effort de sauvetage). L'application peut être mobile et l'environnement peut changer de façon dynamique. Par conséquent, les protocoles ad hoc doivent s'auto-configurer pour s'adapter à l'environnement, changements de trafic et de mission. Les réseaux ad hoc sont la frontière ultime de la communication sans fil. Cette technologie permet aux nœuds de réseau de communiquer directement entre eux à l'aide d'émetteurs-récepteurs sans fil sans avoir besoin d'une infrastructure fixe. Il s'agit d'une caractéristique très distinctive des réseaux ad hoc par rapport aux réseaux sans fil plus traditionnels, tels que les réseaux cellulaires et les réseaux sans fil, dans lesquels les nœuds (par exemple, les utilisateurs de téléphones mobiles) communiquent entre eux par le biais de stations de base (antennes radio filaires).

b.1- Caractéristiques des réseaux ad hoc :

Mobilité : le fait que les nœuds peuvent être rapidement repositionnés et/ou déplacés est la raison d'être des réseaux ad hoc. Un déploiement rapide dans des zones sans infrastructure implique souvent que les utilisateurs doivent explorer une zone et peut-être former des équipes/essaims qui se coordonnent entre eux pour créer un groupe de travail ou une mission. Nous pouvons avoir une mobilité

aléatoire individuelle, une mobilité de groupe, un mouvement le long des itinéraires planifiés, etc. Le modèle de mobilité peut avoir un impact majeur sur le choix d'un schéma de routage et peut donc influencer la performance.

Multihopping : un réseau multi-hop est un réseau où le chemin de la source à la destination traverse plusieurs autres nœuds. Les réseaux spéciaux présentent souvent de multiples avantages pour la négociation des obstacles, la réutilisation du spectre et la conservation de l'énergie.

Auto-organisation : le réseau ad hoc doit déterminer de manière autonome ses propres paramètres de configuration, y compris : adressage, routage, regroupement, identification de position, contrôle de puissance, etc. Dans certains cas, nœuds spéciaux (par exemple, nœuds de base mobiles) peut coordonner leur mouvement et répartit dynamiquement dans la zone géographique pour assurer la couverture des îles déconnectées.

Conservation de l'énergie : la plupart des nœuds spéciaux (p. ex., ordinateurs portatifs, ANP, capteurs, etc.) ont une alimentation électrique limitée et ne peuvent produire leur propre énergie (p. ex., panneaux solaires). La conception de protocoles éconergétiques (p. ex., routage, découverte de ressources, etc.) est essentielle à la longévité de la mission.

Évolutivité : dans certaines applications (par exemple, les grands capteurs environnementaux, les réseaux de véhicules urbains, etc.), le réseau ad hoc peut atteindre plusieurs milliers de nœuds. Pour l'évolutivité du réseau d'infrastructure sans fil, il suffit d'une structure hiérarchique. La mobilité limitée des réseaux d'infrastructure peut également être facilement gérée à l'aide de l'IP mobile ou des techniques de transfert local.

Sécurité : les défis de la sécurité sans fil sont bien connus capacité des intrus à l'écoute et brouiller le canal. Une grande partie du travail effectué dans les réseaux d'infrastructure sans fil en général s'étend au domaine ad hoc. Cependant, les réseaux ad hoc sont encore plus vulnérables aux attaques que les réseaux d'infrastructure. Des attaques actives et passives sont possibles. Un attaquant actif a tendance à perturber les opérations (par exemple, un imposteur se présentant comme un nœud légitime intercepte le contrôle et les paquets de données ; réintroduit de faux paquets de contrôle ; endommage les tables de routage au-delà de la réparation ; déclenche des attaques par déni de service, etc.).

Les attaques passives sont uniques aux réseaux ad hoc et peuvent être encore plus insidieuses que les réseaux actifs. L'attaquant actif est finalement découvert et physiquement désactivé/éliminé. L'attaquant passif n'est jamais découvert par le réseau. Comme un " bug " , il est placé dans un champ de détection ou à un coin de rue. Il surveille les données et contrôle les habitudes de circulation. Avec une faible énergie et une faible probabilité de détection. La défense contre les attaques passives exige de puissantes techniques de cryptage innovantes couplées à des protocoles de réseau soigneusement conçus.

b.2- Diversité des réseaux ad hoc :

b.2.1- Réseaux Peer to Peer ou SIS (Système d'information spontané) :

Sont des réseaux dont le fonctionnement est décentralisé entre les différents utilisateurs du réseau, ou les machines jouent le rôle des clients et serveurs (et aussi routeur, en passant les messages de recherche voire les données vers leur(s) destinataire(s)).

b.2.2- Réseaux de capteurs et communications M2M (Machine à Machine) :

Ils se composent des nœuds intégrant une unité de mesure chargée de capter des grandeurs physiques (chaleur, humidité, vibrations) et de les transformer en grandeurs numériques.

b.2.3- Réseaux en mouvement (informatique embarquée et véhicules communicants) :

Les voitures de nos jours embarquent de plus en plus de technologie et ont de plus en plus besoin de communiquer avec l'extérieur. Les voitures équipées par des capteurs dans les toits et/ou les pare-chocs sont capables de créer des plateformes de réseaux mobile Ad-hoc et de relier en réseau les automobiles passant à proximité les unes des autres. Des prototypes ont déjà été développés pour les véhicules d'urgence (les ambulances, les voitures des pompiers, etc.).

b.3- Exemple d'applications réseau ad hoc :

b.3.1- Véhicules autonomes sans pilote :

: certaines des applications de réseau ad hoc populaires nécessitent des composants robotisés sans pilote. Tous les nœuds d'un réseau générique sont bien entendu capables de fonctionner en réseau autonome. Lorsque la mobilité autonome est également ajoutée, il se présente des possibilités très intéressantes pour le réseautage et le mouvement combinés.

b.3.2- Connexion à Internet :

: LAN sans fil peut être étendu au besoin (au garage, la voiture garée dans la rue, la maison du voisin, etc.) avec des routeurs portables. Ces extensions opportunistes prennent de plus en plus d'importance et constituent la voie d'évolution la plus prometteuse vers les applications commerciales.

b.3.3-Communications des véhicules :

les communications sans fil à courte portée seront utilisées pour surveiller et contrôler les composants mécaniques du véhicule ainsi que pour connecter le

casque d'écoute du conducteur au téléphone cellulaire. Un autre ensemble d'applications novatrices découle des communications avec d'autres voitures sur la route. Les applications potentielles comprennent les messages de sécurité routière, la navigation coordonnée, les jeux vidéo en réseau et d'autres interactions entre pairs.

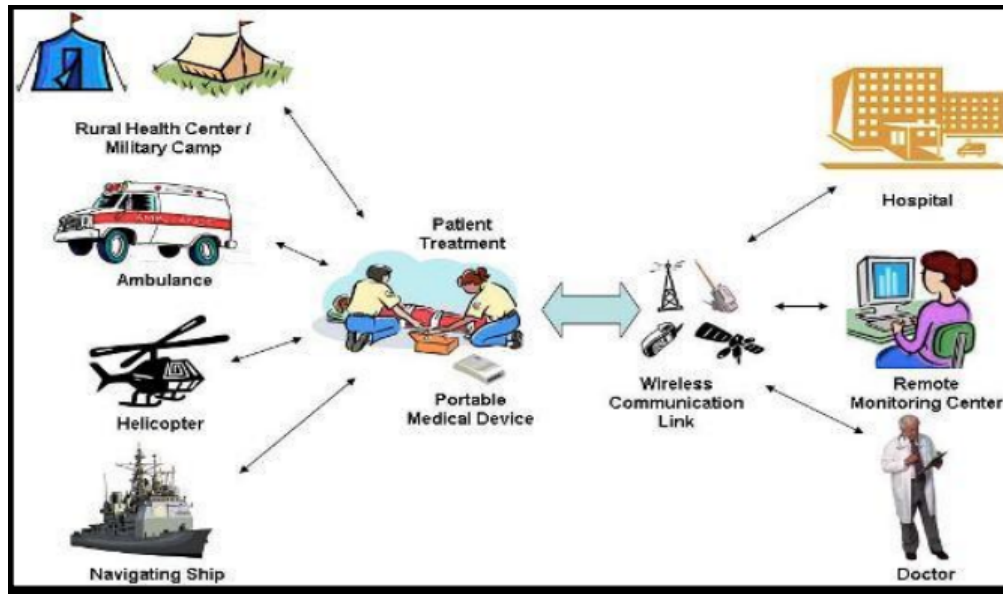


FIGURE 1.4 – domaines d'application des réseaux Ad Hoc.

c. Réseaux ad hoc Vs Réseaux sans fil avec infrastructure :

AD hoc est un type de réseau décentralisé, les appareils sans fil communiquent directement entre eux, plutôt que d'utiliser un pont, comme un point d'accès. Ad hoc est peer-to-peer et ce type appelé architecture peer-to-peer, et ad hoc est créé sur l'impulsion du moment. Il n'a pas besoin de configuration et d'administration d'infrastructure complexes. Ce serait bon marché et rapide. L'infrastructure est plus courante avec les entreprises et les organisations. Et c'est un réseau sans fil centralisé où les appareils sans fil communiquent avec un point d'accès et non directement entre eux.

Réseaux ad hoc	Réseaux sans fil avec infrastructure
Facile	Réseau sans fil centralisé
Rapide	Plan stratégique
Réseau décentralisé	Configuration complexe
	administration constante

1.3 LE routage

Est l'acheminement des informations à la bonne destination à travers un réseau de connexion donné. L'intérêt est de trouver le chemin optimal au sens d'un certain critère de performance (bande passante, délai, etc.).

1.3.1 Le routage dans les réseaux ad hoc :

Le développement du support pour le routage est l'un des défis les plus importants dans les réseaux ad hoc et est essentiel pour les opérations de réseau de base. Certaines combinaisons uniques de caractéristiques rendent le routage dans les réseaux ad hoc intéressant. Premièrement, les nœuds d'un réseau ad hoc peuvent se déplacer de façon incontrôlée. Une telle mobilité de nœud résulte en un réseau très dynamique avec des changements topologiques rapides causant des défaillances fréquentes d'itinéraire. Un bon protocole de routage pour cet environnement de réseau doit s'adapter dynamiquement à l'évolution de la topologie du réseau. Deuxièmement, le canal sans fil sous-jacent offre une bande passante beaucoup plus faible et plus variable que les réseaux câblés. Le canal sans fil fonctionnant comme un support partagé rend la bande passante disponible par nœud encore plus faible. Troisièmement, les nœuds fonctionnent avec des batteries dont l'alimentation en énergie est limitée. Pour que les nœuds restent et communiquent pendant de plus longues périodes, il est souhaitable qu'un protocole de routage soit également éconergétique. Cela fournit également une autre raison pour laquelle les frais généraux doivent être maintenus bas. Par conséquent, les protocoles de routage doivent répondre aux objectifs contradictoires de l'adaptation dynamique et des frais généraux peu élevés pour offrir une bonne performance globale. Si une seule destination est impliquée dans la communication, alors il s'agit d'un "routage unicast", si encore tous les nœuds du réseau ou juste un sous ensemble sont concernés par la réception des données alors on parle du "broadcast" et du "routage multicast", respectivement. [5]

3.1.1 Broadcast (Flooding) :

Flooding (ou diffusion à l'échelle du réseau) est le moyen le plus simple de transmettre des données d'un nœud à un autre nœud du réseau. En cas de flooding, la source transmet simplement le paquet de données à ses nœuds voisins via un mécanisme de diffusion en couche MAC. Chaque nœud entendant la diffusion pour la première fois la retransmet. Ainsi, la diffusion se propage en couches vers l'extérieur à partir de la source, se terminant finalement lorsque chaque nœud a entendu le paquet et l'a transmis une fois. La règle (chaque nœud ne transmet qu'une seule fois) garantit la fin de la procédure et évite également les boucles.

3.1.2 Efficient Flooding Techniques :

Flooding efficace vise essentiellement à éliminer les diffusions redondantes, tout en assurant que tous les nœuds du réseau reçoivent le paquet. Dans la plus simple de toutes les techniques, chaque nœud (autre que la source) retransmet le paquet de données seulement avec une certaine probabilité p [6]. La probabilité p détermine l'efficacité de cette technique, une très petite valeur empêche flooding d'atteindre chaque nœud tandis qu'une très grande valeur entraîne de nombreuses diffusions redondantes. La bonne valeur de p dépend de plusieurs facteurs, y compris le degré de nœud moyen. Déterminer la bonne valeur de p reste un problème difficile. D'autres techniques sont également possibles. Par exemple, quand un nœud entend le paquet de diffusion, il ne le transmet pas immédiatement, mais attend une brève période pour voir s'il entend à nouveau le même paquet. Si elle l'entend plusieurs fois (K fois) dans cette période, elle suppose que tous ses voisins doivent avoir entendu ce paquet, et s'abstient de le transmettre [7]. La détermination des valeurs appropriées K et du délai d'attente devient complexe.

1.3.2 Les défis de routage dans les réseaux ad hoc mobiles :

Le développement du support pour le routage est l'un des défis les plus importants dans les réseaux ad hoc et est essentiel pour les opérations de réseau de base. Certaines combinaisons uniques de caractéristiques rendent le routage dans les réseaux ad hoc intéressant. Premièrement, les nœuds d'un réseau ad hoc peuvent se déplacer de façon incontrôlée. Une telle mobilité de nœud résulte en un réseau très dynamique avec des changements topologiques rapides causant des défaillances fréquentes d'itinéraire. Un bon protocole de routage pour cet environnement de réseau doit s'adapter dynamiquement à l'évolution de la topologie du réseau. Deuxièmement, le canal sans fil sous-jacent offre une bande passante beaucoup plus faible et plus variable que les réseaux câblés. Le canal sans fil fonctionnant comme un support partagé rend la bande passante disponible par nœud encore plus faible. Troisièmement, les nœuds fonctionnent avec des batteries dont l'alimentation en énergie est limitée. Pour que les nœuds restent et communiquent pendant de plus longues périodes, il est souhaitable qu'un protocole de routage soit également éconergétique. Cela fournit également une autre raison pour laquelle les frais généraux doivent être maintenus bas. Par conséquent, les protocoles de routage doivent répondre aux objectifs contradictoires de l'adaptation dynamique et des frais généraux peu élevés pour offrir une bonne performance globale.

1.3.3 LES PROTOCOLES DE ROUTAGE AD-HOC :

L'objectif principal des protocoles de routage est l'établissement et la maintenance des chemins, pour que les données soient correctement délivrées dans le réseau [8].

Pour être réellement opérationnel dans un environnement mobile, le protocole de routage prend en compte trois phases :

- Dissémination de l'information de routage :

elle permet de connaître suffisamment d'éléments sur la topologie pour choisir un chemin atteignant le nœud de destination. Suivant la quantité d'informations échangées, les nœuds obtiennent une vue plus ou moins précise de la topologie du réseau. Le protocole de routage se voit dans l'obligation d'optimiser l'envoi de ces informations, car elles sont fortement consommatrices en bande passante. [8]

- Sélection du chemin :

une fois les informations de routage obtenues, le protocole de routage peut sélectionner une route parmi l'ensemble obtenu en fonction de certains critères. Pour les protocoles Meilleur effort (Best Effort), le critère est de minimiser le nombre de sauts du chemin. Ainsi, parmi l'ensemble des routes qui lui sont proposées, le protocole choisit celle traversant le plus faible nombre de nœuds. Les routes choisies doivent être dépourvues de boucles. La présence de boucles rend inefficace le chemin sélectionné puisque le paquet ne pourra pas atteindre la destination consommant inutilement de la bande passante. En effet, un paquet de données transitant sur un chemin, possédant une boucle, va tourner en rond tant que la boucle est présente. Pour éviter qu'un paquet de données tourne indéfiniment, le paquet est détruit lorsqu'il atteint la limite imposée par le champ TTL présent dans le protocole IP. Un protocole de routage peut créer deux sortes de boucles : les boucles temporaires et les boucles permanentes [10]. Les premières ont lieu pendant le transfert d'un message de routage. Durant ce temps, des stations peuvent être mises à jour et d'autres non, d'où la possible apparition d'une boucle. Elle dure au maximum la durée de traversée du réseau par un message de routage. Les boucles permanentes, quant à elles, sont dues au phénomène du bouclage à l'infini [11]. Ces boucles peuvent consommer énormément de bande passante.

- Maintenance des routes :

dans un environnement mobile, la topologie du réseau ne cesse d'évoluer avec le temps. De fait, les routes sont amenées à changer avec le déplacement des nœuds. Une route doit éviter de rester longtemps interrompue, car les paquets ne pourraient atteindre leur destination. Le protocole de routage doit donc tenir compte de ces changements et mettre à jour les routes qui viennent à être coupées. [8]

1.3.4 Classification des protocoles de routage ad-hoc :

Les protocoles de routage pour les réseaux ad hoc peuvent être classés de différentes manières. Elle peut dépendre de :

- Le principe de conception : proactif, réactif, ou hybride. ;
- La structure du réseau : qui peut être uniforme ou non uniforme ;
- L'état des informations : obtenues par chaque nœud dans le réseau comme pour les protocoles de routage orientés topologie ou les protocoles de routage orientés destination..

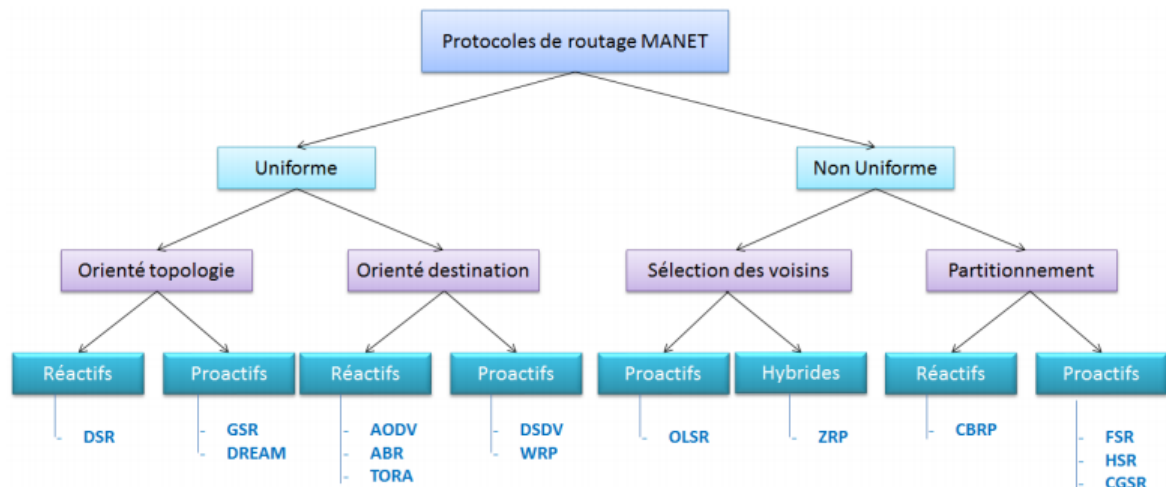


FIGURE 1.5 – Classification des protocoles de routage MANET.

3.4.1 Routage proactif :

Les protocoles proactifs maintiennent les routes unicast entre toutes les paires de nœuds, que toutes les routes soient utilisées ou non. Par conséquent, lorsque le besoin se présente (c.-à-d., lorsqu'une source de trafic commence une session avec une destination éloignée), la source de trafic a une route facilement accessible et n'a pas à subir de retard pour la découverte d'itinéraire. Ces protocoles peuvent également trouver des itinéraires optimaux (chemins les plus courts) compte tenu d'un modèle de coûts de liaison.

- Vecteur de distance séquencé par destination (DSDV) :

Est développé sur la base d'Algorithme de routage Bellman – Ford [11] avec quelques modifications. Dans ce protocole de routage, chaque nœud mobile du réseau conserve une table de routage. Chacune des tables de routage contient la liste de toutes les destinations disponibles et le numéro de houblon à chacun. Chaque entrée de table est étiquetée avec un numéro de séquence, qui est généré par le nœud de destination. Des transmissions périodiques des mises à jour des

tables de routage aident à maintenir les informations de topologie du réseau. S'il y a un nouveau changement significatif pour les informations de routage, les mises à jour sont transmises immédiatement. Alors les mises à jour des informations de routage peuvent être périodiques ou pilotées par des événements. Le protocole DSDV requiert chaque nœud mobile dans le réseau pour annoncer sa propre table de routage à ses voisins actuels. La publicité se fait soit par radiodiffusion ou multidiffusion.

- Protocole de routage sans fil (WRP) :

WRP appartient à la classe générale des algorithmes de recherche de chemin définis comme l'ensemble du chemin le plus court distribuer algorithmes qui calculent les chemins en utilisant des informations concernant la longueur et l'avant-dernier saut du chemin le plus court à chaque destination. WRP réduit le nombre de cas dans lesquels une boucle de routage temporaire peut se produire. Aux fins de routage, chaque nœud maintient quatre choses :

- 1) Un tableau des distances.
- 2) Une table de routage.
- 3) Un tableau des coûts de liaison.
- 4) Une liste de retransmission des messages (MRL).

WRP utilise des transmissions de messages de mise à jour périodiques aux voisins d'un nœud. Les nœuds de la liste de réponses du message de mise à jour (qui est formé à l'aide de MRL) doivent envoyer des accusés de réception. S'il n'y a pas de changement par rapport au dernier mise à jour, les nœuds de la liste de réponses doivent envoyer un message " Hello " inactif pour garantir la connectivité. Un nœud peut décider s'il faut mettre à jour sa table de routage après avoir reçu un message de mise à jour d'un voisin et toujours il cherche une meilleure chemin en utilisant les nouvelles informations. Si un nœud obtient un meilleur chemin, il relaie ces informations aux nœuds d'origine afin qu'ils peuvent mettre à jour leurs tables. Après avoir reçu l'accusé de réception, le nœud d'origine met à jour sa LMR. Donc, chaque fois que la cohérence des informations de routage est vérifiée par chaque nœud de ce protocole, ce qui aide à éliminer les boucles de routage et essaie toujours de trouver la meilleure solution de routage dans le réseau Envoyer des commentaires Historique Enregistré Communauté.

- Routage d'état de liaison optimisé (OLSR) :

Le protocole OLSR (Optimized Link State Routing) est une variante du routage d'état de liaison traditionnel, modifié pour un fonctionnement amélioré dans les réseaux ad hoc. La principale caractéristique d'OLSR est son utilisation de relais multipoints (MPR) pour réduire la surcharge de floods du réseau et la taille des mises à jour de l'état des liaisons. Chaque nœud calcule ses MPR à partir de son ensemble de voisins. Le paramètre MPR est sélectionné de telle sorte que lorsqu'un nœud diffuse un message, la retransmission de ce message

par le paramètre MPR garantit que le message est reçu par chacun de ses deux voisins de hop. Par conséquent, chaque fois qu'un nœud diffuse un message, seuls les voisins dans son MPR mettent rediffuser le message. Les autres voisins qui ne sont pas dans le set MPR traitent le message, mais ne le retransmettent pas. En outre, lors de l'échange d'informations de routage d'état de liaison, un nœud ne liste ses connexions aux voisins qui l'ont sélectionné comme MPR. Cet ensemble de voisins est appelé les sélecteurs MPR.

3.4.2 Routage réactif :

Le protocole de routage réactif est également appelé protocole de routage à la demande. Dans ce protocole la route est découverte chaque fois que cela est nécessaire. Les nœuds lancent la découverte de la route sur demande. Le nœud source voit son itinéraire cache pour l'itinéraire disponible de la source à la destination si l'itinéraire n'est pas disponible, il lance la découverte de l'itinéraire processus.

3.4.2.1 Routage de source dynamique (DSR) :

Est un protocole réactif basé sur l'approche de la route source. Dans dynamique Source Routage (DSR), le protocole est basé sur l'algorithme d'état de liaison dans lequel la source initie la route découverte sur demande. L'expéditeur détermine l'itinéraire de la source à la destination et il inclut l'adresse du nœud intermédiaires vers l'enregistrement de route dans le paquet. DSR a été conçu pour les réseaux multi-sauts pour les petits diamètres Il s'agit du protocole sans balise dans lequel aucun message HELLO n'est échangé entre les nœuds pour les informer de leurs voisins dans le réseau.

3.4.2.2 Routage ad hoc à distance sur demande (AODV) :

Est fondamentalement une amélioration du DSDV. Mais, AODV est un protocole de routage réactif au lieu de proactif. Il minimise le nombre de diffusions en créant des itinéraires basés sur demande, ce qui n'est pas le cas pour DSDV. Lorsqu'un nœud source souhaite envoyer un paquet vers une destination, il diffuse un paquet de demande d'itinéraire (RREQ). Les nœuds voisins diffusent à leur tour le paquet à leurs voisins et le processus continue jusqu'à ce que le paquet atteigne la destination. Pendant le processus de transfert de l'itinéraire demande, les nœuds intermédiaires enregistrent l'adresse du voisin d'où provient la première copie du paquet de diffusion reçu. Cet enregistrement est stocké dans leurs tables de routage, ce qui aide à établir un chemin inverse. Si des copies supplémentaires du même RREQ sont reçus ultérieurement, ces paquets sont rejetés. La réponse est envoyée en utilisant le chemin inverse. Pour l'itinéraire maintenance, lorsqu'un nœud source se déplace, il peut relancer un processus de découverte d'itinéraire. Si un nœud intermédiaire se déplace au sein d'un itinéraire particulier, le voisin du nœud dérivé peut détecter la défaillance de la liaison et envoie une défaillance de la liaison notification à son voisin en amont.

Ce processus se poursuit jusqu'à ce que la notification d'échec atteigne le nœud source. Sur la base des informations reçues, la source peut décider de relancer la phase de découverte de l'itinéraire.

1.3.5 Algorithmes de routage :

Une exigence importante d'un réseau de communication est de faire circuler ou acheminer le trafic d'un nœud source vers un nœud de destination. Pour ce faire, nous devons déterminer une route, qui est un chemin du nœud source au nœud de destination. Un itinéraire peut certainement être configuré manuellement ; une telle route est connue sous le nom de route statique. En général, cependant, il est souhaitable d'utiliser un algorithme de routage pour déterminer un itinéraire. La fonction de base d'un algorithme de routage est de trouver un chemin optimal d'une source à une destination. Le chemin optimal est généralement un chemin avec la longueur la plus courte, bien que d'autres critères d'optimisation comme le délai minimum ou le débit maximum soient également possibles. Des exemples bien connus d'algorithmes de routage sont les algorithmes de Dijkstra et de Bellmanford.

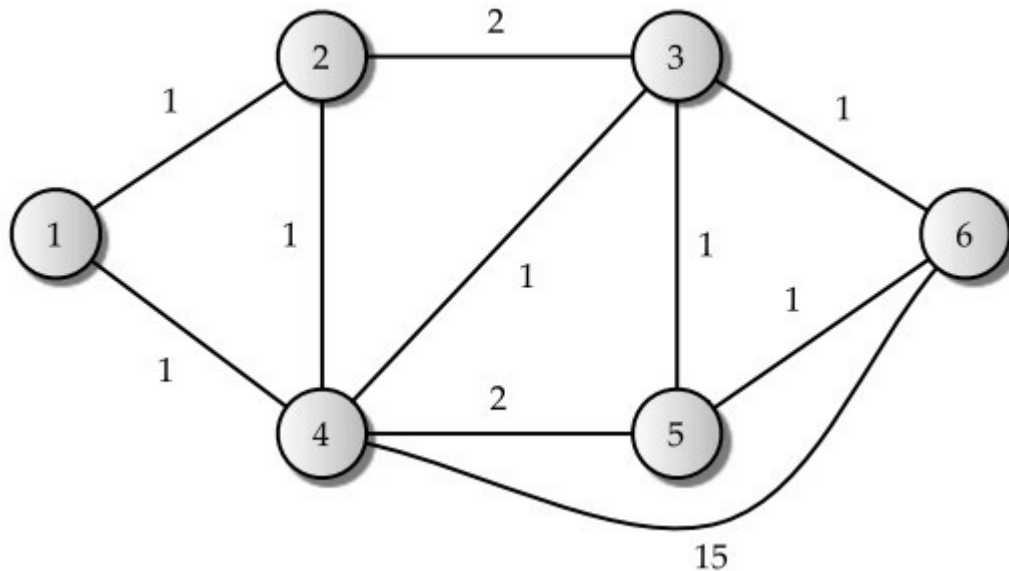


FIGURE 1.6 – Un réseau à six nœuds.

3.5.1 Algorithme de Bellman-Ford :

L'algorithme Bellman-Ford utilise une idée simple pour calculer le chemin le plus court entre deux nœuds de façon centralisée. Deux nœuds génériques, nœud i et nœud j , dans un réseau de nœuds N . Ils peuvent être reliés directement par un lien comme le lien 4-6 avec les nœuds d'extrémité 4 et 6 (voir la figure 6). Comme on peut le voir à la figure 1.3, de nombreux nœuds ne sont pas

directement connectés, par exemple, les nœuds 1 et 6 ; dans ce cas, pour trouver la distance entre ces deux nœuds, nous devons recourir à d'autres nœuds et liens. La notion de coût entre deux nœuds, qu'ils soient directement connectés ou non. Deux notations importantes :

d_{ij} = liaison entre les nœuds i et j

\bar{D}_{ij} = chemin de coût minimum calculé du nœud i au nœud j .

Si deux nœuds sont directement connectés, alors la liaison d_{ij} prend une valeur finie. Considérons à nouveau la figure 1.3. Ici, les nœuds 4 et 6 sont directement connectés avec la liaison 15 ; ainsi, nous pouvons écrire $d_{46} = 15$. D'autre part, les nœuds 1 et 6 ne sont pas directement connectés donc, $d_{16} = \infty$. Du nœud 4 au nœud 6, le coût minimum est 2, qui prend le chemin 4-3-6, c'est-à-dire $\bar{D}_{46} = 2$ et $d_{46} = 15$. Pour les nœuds 1 et 6, nous trouvons que $\bar{D}_{16} = 3$ tandis que $d_{16} = \infty$. Comme on peut le voir, un chemin de coût minimum peut être obtenu entre deux nœuds dans un réseau indépendamment du fait qu'ils soient directement connectés ou non, aussi longtemps que l'un des nœuds finaux n'est pas complètement isolé du reste du réseau. Pour calculer le coût minimum entre deux nœuds dans un réseau ; considérons un nœud générique dans le réseau qui est directement connecté à l'un des nœuds finaux ; nous supposons que k est directement connecté au nœud destination j , ce qui signifie que d_{kj} a une valeur finie. Les équations suivantes, appelées équations de Bellman, doivent être satisfaites par le chemin le plus court entre le nœud i et le nœud j :

$$\begin{aligned} \bar{D}_{ii} &= 0, \text{ Pour tous } i, \\ \bar{D}_{ij} &= \min_{k \neq j} \{ \bar{D}_{ik} + d_{kj} \}, \text{ Pour } i \neq j \end{aligned}$$

Le terme pour le coût minimum en termes de nombre de sauts h comme suit :

$\bar{D}_{ij}^{(h)}$ = coût du chemin de coût minimal du nœud i au nœud j jusqu'à h nombre de sauts sont pris en compte.

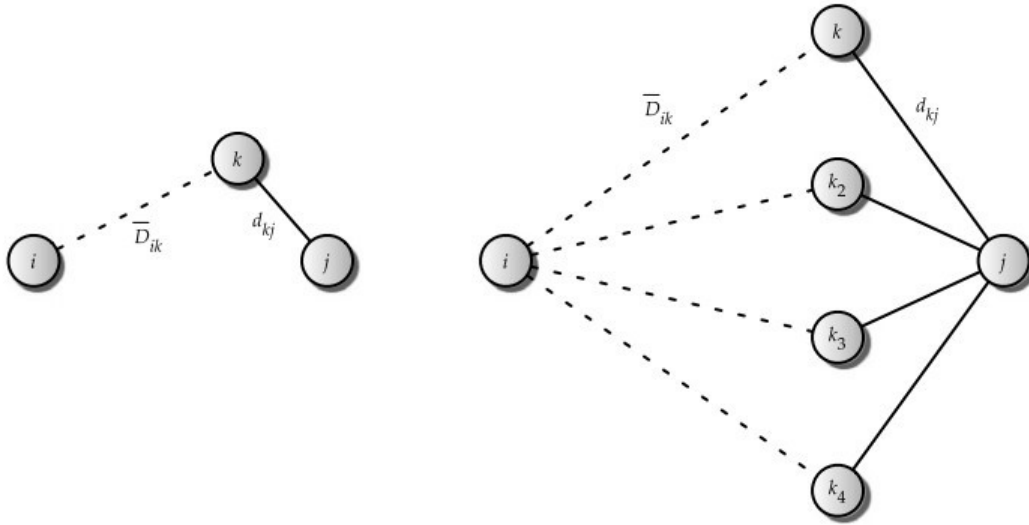


FIGURE 1.7 – Algorithme Bellman-Ford (la ligne continue indique un lien direct ; la ligne pointillée indique la distance)..

3.5.2 Algorithme de Dijkstra :

L'algorithme de Dijkstra est un autre algorithme de routage de chemin le plus court bien connu. L'idée de base de l'algorithme de Dijkstra est très différente de l'algorithme de Bellman-Ford. Il fonctionne sur la notion d'un ensemble de nœuds voisins candidats ainsi que sur le calcul de la source pour identifier le chemin le plus court vers une destination. Une autre propriété intéressante sur l'algorithme de Dijkstra est qu'il calcule les chemins les plus courts vers toutes les destinations à partir d'une source, au lieu de simplement pour une paire spécifique de nœuds source et destination à la fois, ce qui est très utile, en particulier dans un réseau de communication, car un nœud veut calculer le chemin le plus court vers toutes les destinations.

Considérez un nœud générique i dans un réseau de nœuds N . La liste de nœuds N sera indiquée par $N = \{1, 2, \dots, N\}$. Un nœud générique de destination sera désigné par j ($j \neq i$). Nous utiliserons les deux termes suivants :

d_{ij} = liaison entre les nœuds i et j

\underline{D}_{ij} = chemin de coût minimum calculé du nœud i au nœud j .

L'algorithme de Dijkstra divise la liste N des nœuds en deux listes : il commence par la liste permanente S , qui représente les nœuds déjà considérés, et la liste provisoire S' , pour les nœuds non encore pris en compte. Au fur et à mesure que l'algorithme progresse, la liste S s'élargit avec de nouveaux nœuds inclus tandis que la liste S' rétrécit lorsque les nœuds nouvellement inclus dans S , et sont supprimés de cette liste ; l'algorithme s'arrête lorsque la liste S' devient vide. Initialement, nous avons $S = \{i\}$ et $S' = N \setminus \{i\}$ (c.-à-d., tous les nœuds dans N sauf le nœud i). Le noyau de l'algorithme a deux parties :

(1) comment étendre la liste S

(2) comment calculer le chemin le plus court vers les nœuds qui sont voisins des nœuds de la liste S (mais les nœuds ne sont pas encore dans cette liste). La liste S est développée à chaque itération en considérant un nœud voisin k de nœud i avec le chemin le moins coûteux de nœud i.

A chaque itération, l'algorithme considère alors les nœuds voisins de k, qui ne sont pas déjà en S, pour voir si le coût minimum change par rapport à la dernière itération. Nous allons illustrer l'algorithme de Dijkstra en utilisant le réseau indiqué dans la figure 6. Supposons que le nœud 1 veuille trouver les chemins les plus courts vers tous les autres nœuds du réseau. Puis, initialement, $S = \{1\}$, et $S' = \{2, 3, 4, 5, 6\}$, et les chemins les plus courts à tous les nœuds qui sont voisins directs du nœud 1 peuvent être facilement trouvés tandis que pour le reste, le coût reste à ∞ , c-à-d ,

$$\underline{D}_{12} = 1, \underline{D}_{14} = 1, \underline{D}_{13} = \underline{D}_{15} = \underline{D}_{16} = \infty .$$

Pour la prochaine itération, nous notons que le nœud 1 a deux voisins directement connectés : le nœud 2 et le nœud 4 avec $d_{12} = 1$ et $d_{14} = 1$, respectivement ; tous les autres nœuds ne sont pas directement connectés au nœud 1, et donc, le "direct" coût pour ces nœuds reste à ∞ . Puisque les deux nœuds 2 et 4 sont voisins avec le même coût minimum, nous pouvons choisir l'un ou l'autre. Pour notre illustration, nous choisissons le nœud 2, et ce nœud devient intermédiaire, k. Ainsi, nous avons maintenant $S = \{1, 2\}$, et S' devient la liste $\{3, 4, 5, 6\}$. Ensuite, du nœud 2 à ses voisins directs qui ne sont pas déjà dans l'ensemble S. Les voisins du nœud 2 sont le nœud 3 et le nœud 4 :

$$\begin{aligned} \underline{D}_{13} &= \min\{\underline{D}_{13}, \underline{D}_{12} + d_{23}\} = \min\{\infty, 1 + 2\} = 3 \\ \underline{D}_{14} &= \min\{\underline{D}_{14}, \underline{D}_{12} + d_{24}\} = \min\{1, 1 + 1\} = 1 \end{aligned}$$

Il n'y a pas d'amélioration du coût pour le nœud 4. Pour le nœud 3, le chemin le plus court, 1 - 2 - 3. Pour le reste des nœuds, le coût reste à ∞ . Ceci termine cette itération. L'itération suivante et constater que le prochain intermédiaire est $k = 4$, et le processus se poursuit comme avant.

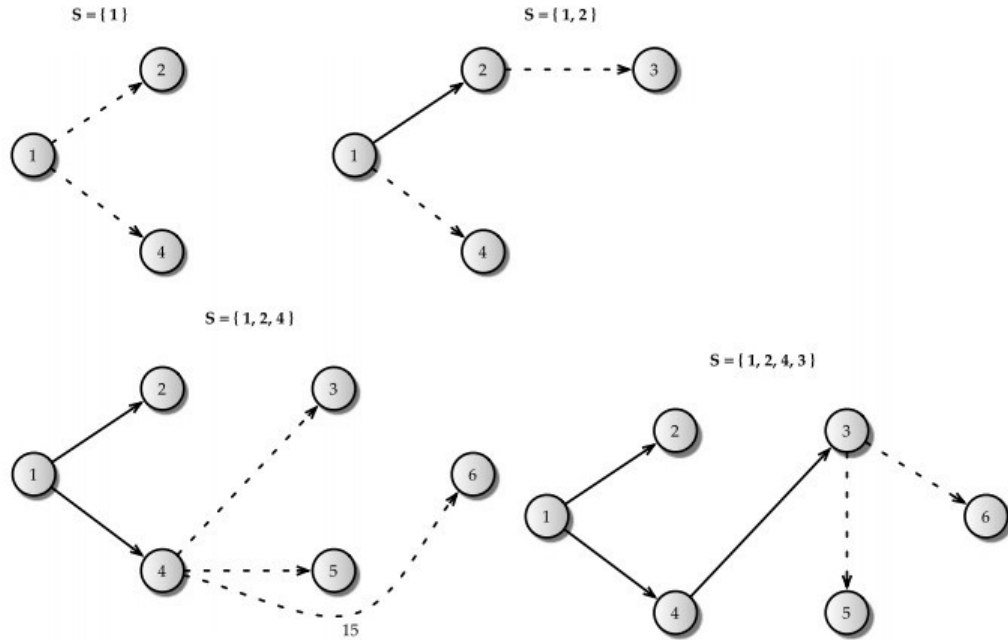


FIGURE 1.8 – Algorithme de Dijkstra

3.5.3 Les limites des algorithmes déterministe :

Supposons que le nœud (u) envoie un message au nœud (n) exactement à la vitesse, à la distance et dans les mêmes conditions. Chaque fois que nous exécutons cette expérience, le message sera reçu exactement dans le même nœud (n) : nous pouvons prédire exactement ce qui va se passer. Ceci est un exemple d'un système déterministe. Le problème est que lorsque nous avons beaucoup de nœuds, plus nous en avons, plus le problème se pose et il y aura plus d'un nœud pour envoyer le message exactement au même nœud de destination, ici nous aurons une collision. Pour éviter et résoudre ce problème, nous devons passer au processus aléatoires. L'aléatoire est l'opposé du déterminisme : un phénomène aléatoire est celui qui peut donner des résultats différents dans des expériences répétées, même si nous utilisons exactement les mêmes conditions dans chaque expérience. Par exemple, si nous lançons une pièce, nous savons à l'avance qu'elle reviendra en tête-à-tête ou en queue, mais nous ne pouvons pas prédire avant une expérience donnée lequel de ces résultats se produira. Dans notre projet, nous développons un algorithme pour raisonner précisément sur les phénomènes aléatoires.

1.4 Les Processus aléatoires :

la description mathématique d'un phénomène aléatoire tel qu'il change dans le temps est un processus stochastique. Depuis le siècle dernier, on s'est rendu davantage compte que les modèles stochastiques (ou non déterministes) sont plus

réalistes que les modèles déterministes dans de nombreuses situations. Les observations prises à différents moments plutôt que celles prises à une période de temps fixe ont commencé à attirer l'attention des scientifiques. Les physiciens et les ingénieurs en communication ont joué un rôle de premier plan dans le développement de l'indéterminisme dynamique. De nombreux phénomènes se produisant dans les sciences physiques et les sciences de la vie sont étudiés non seulement comme un phénomène aléatoire, mais aussi comme évoluant avec le temps ou l'espace. Des considérations similaires sont également faites dans d'autres domaines tels que les sciences sociales, l'économie et les sciences de gestion, etc. Le champ d'application des processus stochastiques qui sont fonction du temps ou de l'espace ou des deux est en constante augmentation.

Un processus stochastique est une famille de variables aléatoires X_t , où t prend des valeurs dans l'ensemble d'index T [14]. Les valeurs de X_t sont appelées l'espace d'état et seront désignées par S .

Si T est dénombrable, le processus stochastique est appelé une séquence stochastique. Si S est dénombrable, alors le processus stochastique est appelé processus d'état discret.

Si S est un sous-ensemble de la ligne réelle, le processus stochastique est appelé processus à valeur réelle. Si S est un sous-ensemble de la ligne réelle, le processus stochastique est appelé processus à valeur réelle. Si T prend un nombre indénombrable de valeurs comme $(0, +\infty)$ ou $(-\infty, +\infty)$, Le processus stochastique est appelé processus temporel continu. Si T prend un nombre indénombrable de valeurs comme $(0, +\infty)$ ou $(-\infty, +\infty)$, Le processus stochastique est appelé processus temporel continu. Pour souligner sa dépendance de t et du point d'échantillonnage w , nous noterons le processus stochastique par $X(t, w)$, $t \in T, w \in \Omega$. c'est-à-dire que pour chaque $w \in \Omega$, c'est-à-dire que pour chaque $w \in W$, $X_t = X(t, w)$ est fonction de t .



FIGURE 1.9 – réalisation du processus stochastique

1.4.1 Différents types de processus stochastiques :

Voici les types de processus stochastiques les plus importants que nous rencontrons :

1. Séquence stochastique indépendante (processus temporel discret)

$T = [1, 2, 3, \dots]$ et $\{X_t, t \in T\}$ sont des variables aléatoires indépendantes.

2. Processus de renouvellement (processus à temps discret)

$$\text{Ici } T = [0, 1, 2, 3, \dots], S = [0, +\infty].$$

Si X_n sont des variables aléatoires non négatives et $S_n = X_1 + \dots + X_n$ alors $\{S_n\}$ forme un temps discret (*processus de renouvellement*).

3. Processus d'incrémentatation indépendant (processus de temps continu)

$T = [t_0, +\infty]$, où t_0 est n'importe quel nombre réel (+ou-). Pour chaque :

$$t_0 < t_1 < \dots < t_n, t_i \in T, i = 1, 2, \dots, n \quad (1.1)$$

Si $X_{t_0}, X_{t_1} - X_{t_0}, X_{t_2} - X_{t_1}, \dots, X_{t_n} - X_{t_{n-1}}$ sont indépendants pour tous les choix possibles de (1,1), alors le processus stochastique

4. Processus de Markov

Si $P[X_{t_{n+1}} \in A | X_{t_n} = a_n, X_{t_{n+1}} = a_{n+1}, \dots, X_{t_0} = a_0] = P[X_{t_{n+1}} \in A | X_{t_n} = a_n]$ valable pour tous les choix de $t_0 < t_1 < t_2 < \dots < t_{n+1}, t_i \in T, i = 0, 1, 2, \dots, n+1$ (1,2) Et $A \in B$, le champ Borel de l'espace d'état S , alors $\{X_t, t \in T\}$ est appelé un processus de Markov.

1.4.2 Processus de branchement :

Processus de branchement est un type de processus stochastique, le processus de Galton-Watson [15] est le processus de branchement le plus fondamental. Un exemple de l'arbre résultant produit par ce processus de branchement est illustré à la Figure 1.10.

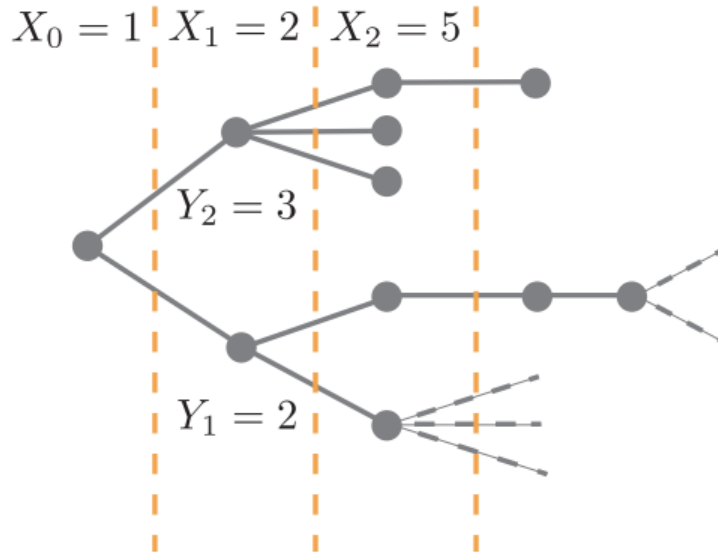


FIGURE 1.10 – Un exemple de l'arborescence résultante du processus de branchement de Galton-Watson

Le processus démarre à partir d'un seul nœud, est appelé la graine, et chaque nœud génère de nouveaux nœuds au hasard. Le nombre de nouveaux nœuds générés à partir du nœud i , Y_i , est une variable stochastique. Supposons que Y_i est indépendant et identique c'est-à-dire

$$\mathbf{P} \{Y_i = K\} = P_k$$

Chaque nœud est généré en tant que nœud actif et une fois qu'il génère de nouveaux nœuds, il devient inactif. Par conséquent, le processus de branchement se poursuit jusqu'à ce qu'aucun nœud ne génère un nouveau nœud. La distance d du nœud germe appelée génération. Le nombre de nœuds de la d ième génération est noté X_d . Le processus commence à partir d'une seule graine, $X_0 = 1$. Lorsque le nombre de nœuds dans la d ième génération est i ($X_d = i$), la probabilité de transition que le nombre de nœuds soit j dans le $(d + 1)$ génération ($X_{d+1} = j$) est donnée par :

$$\mathbf{p} (j|i) = \mathbf{p} (X_{d+1} = j|X_d = i) = \mathbf{p} \sum_{K=1}^i Y_K = j$$

1.5 Conclusion :

Ce chapitre nous a permis d'avoir une vue globale sur les réseaux de capteurs sans fil qui présentent un intérêt considérable vu à l'évolution connue récemment dans ce domaine. Par conséquent, nous avons remarqué que ces réseaux ont plusieurs contraintes, dont le problème majeur est la consommation d'énergie. Dans le prochain chapitre nous allons définir les différentes classifications des protocoles de routage.

Cardinalité De Multidiffusion Et Sélection intelligente :

2.1 Introduction :

Alors que les réseaux de communication deviennent de plus en plus répandus, la recherche pour améliorer leur conception est devenue à la fois plus globale et plus spécifique à l'application. Les problèmes de protocole tels que la priorité du service, les schémas de retransmission, le contrôle de flux et le routage sont désormais étudiés pour les grands systèmes distribués et pour une variété de types de canaux de communication. En outre, il est reconnu depuis longtemps que l'intégration de divers types de sources, telles que la voix, la vidéo, le courrier électronique, l'accès au Web et les données de transactions commerciales dans un réseau transparent se traduit à la fois par la plus grande efficacité et la facilité d'utilisation. Dans de tels réseaux, comme la capacité à fournir une qualité de service (QoS) appropriée à chaque type de service devient primordiale. Un autre défi est posé par les réseaux sans fil, où ces objectifs de service doivent être atteints avec des canaux peu fiables et variant dans le temps, et où de nouvelles préoccupations, telles que la consommation d'énergie et les interférences de canaux multi-utilisateurs, imposent des contraintes supplémentaires. Le terme ad-hoc est appliqué au réseau dans lequel il n'y a pas de contrôleur de réseau central, chaque nœud peut lui-même agir comme stockage et retransmission. Un réseau ad-hoc contient un certain nombre de paquets, chacun étant destiné à un ensemble de nœuds de destination. Le problème général de routage est de définir une politique qui, compte tenu des historiques de trajectoire de tous les paquets, choisit quels nœuds devraient ensuite transmettre quel paquet. Il est généralement souhaitable que cette politique soit mise en œuvre sous une forme distribuée, afin que les décisions de transmission puissent être décidées localement sans connaissance des autres parties du réseau.

2.2 Contexte :

Nous abordons le problème de la multidiffusion de données dans un réseau ad hoc sans fil opportuniste d'objets en mouvement qui délivre un message avec un nombre minimal de nœuds encore atteint un taux de livraison élevé, c'est-à-dire une accessibilité, dans une topologie hautement dynamique où les nœuds se déplacent à différentes vitesses. Ce fut un sujet de recherche difficile à développer des protocoles de routage efficaces dans les réseaux ad hoc. Ce problème est d'une grande importance pour diverses applications car il facilite la diffusion fiable des alertes d'urgence ou de la proximité d'événements imprévus ou d'un champ de bataille.

2.3 Travaux connexes :

La première catégorie de travaux consiste à appliquer la méthode des inondations, l'un des mécanismes les plus déployés.

2.3.1 Flooding :

Chaque nœud diffuse le message à tous les nœuds voisins dans la plage de communication sans aucun processus de sélection basé sur un local topologie. Par conséquent, il utilise un grand nombre de nœuds pour envoyer le message au nœud de destination via plusieurs houblons, ce qui entraîne un problème de tempête de diffusion mais garantit grande accessibilité. Le problème de la tempête de diffusion est toujours difficile dans les réseaux ad hoc en raison de la difficulté de prévoir et de gérer la topologie du réseau à commutation rapide tout en limiter la redondance des données.

Algorithme 1 : Figure 2. Un aperçu de l'algorithme flooding simple pour la diffusion dans les MANETs

- 1 **Algorithme flooding** (m) Réception de protocole ()
 - 2 Lors de la réception d'un message diffusé m au nœud A , procédez comme suit :
 - 3 **if** *le message m est reçu pour la première fois* **then**
 - 4 Broadcast (m) il s'agit de la primitive de diffusion locale de base pour les nœuds
 à portée seulement
-

Pour atténuer les effets de ce problème, un certain nombre de solutions ont été suggérées. L'un des efforts pour réduire le nombre de nœuds utilisés dans la transmission est la méthode des potins [the gossip method].

2.3.2 la méthode des potins [the gossip method] :

Le Gossiping [16] ou Bavardage en français est une version légèrement améliorée du Flooding où le nœud émetteur sélectionne aléatoirement un nœud parmi

ses voisins pour lui transmettre les données, une fois le nœud voisin reçoit ces données, il choisit un autre nœud d'une façon aléatoire pour lui expédier ces données à son tour, jusqu'à ce que les données atteignent la station de base. Gossip évite le problème de l'implosion avec le choix probabiliste des nœuds pour envoyer le paquet plutôt que de le diffuser. Cependant, cette méthode génère un retard dans la propagation des données dans les nœuds.

2.3.3 Flooding probabiliste :

En Flooding probabiliste, la probabilité p de transmettre le message à chaque nœud est ajustée localement. Alshaer et coll. et Bani-Yassein et al ont utilisé la probabilité de rediffusion en fonction de la densité du voisinage. Si le nombre de voisins est supérieur au nombre moyen de voisins dans les réseaux, une faible probabilité est attribuée au nœud intermédiaire et vice versa. Lors de la réception d'un message de diffusion pour la première fois, un nœud rediffuse le message avec une probabilité prédéterminée p de sorte que chaque nœud a la même probabilité de rediffuser le message, quel que soit son nombre de voisins. Dans les réseaux denses, plusieurs nœuds partagent une plage de transmission similaire. Par conséquent, ces probabilités contrôlent la fréquence des rediffusions et peuvent ainsi économiser les ressources du réseau sans affecter les taux de livraison. Notez que dans les réseaux clairsemés, il y a beaucoup moins de couverture partagée ; ainsi, certains nœuds ne recevront pas tous les paquets de diffusion, sauf si le paramètre de probabilité est élevé. Donc, si la probabilité de rediffusion p est fixée à une valeur beaucoup plus petite, l'accessibilité sera mauvaise. D'un autre côté, si p est défini sur une valeur beaucoup plus grande, de nombreuses rediffusions redondantes seront générées. Le besoin d'ajustement dynamique augmente donc. La probabilité de rediffusion devrait être élevée sur les hôtes dans les zones clairsemées et faible sur les hôtes dans les zones plus denses. Une diffusion directe par flooding est généralement très coûteuse et entraînera une redondance, des conflits et des collisions graves, (broadcast storm problem)[17].

2.3.4 Le routage de multicast :

Si une seule destination est impliquée dans la communication, alors il s'agit d'un "routage unicast", si encore tous les nœuds du réseau ou juste un sous ensemble sont concernés par la réception des données alors on parle du "broadcast" et du "routage multicast", respectivement. [5]

2.3.5 Description et hypothèses :

Le voisinage : Le nœud voisin v' d'un nœud est un nœud dans la plage de transmission (r_v), c'est-à-dire $(v, v') < r_v$. Les nœuds voisins sont classés en deux groupes : les nœuds actifs et les nœuds inactifs. Les nœuds qui n'ont pas reçu le message sont appelés nœuds actifs, sinon nœuds inactifs. Avant de transmettre le message, le nœud envoie une requête à tous les nœuds voisins et reçoit s'ils sont

actifs ou non. $N'(v)$ ensemble de nœuds voisins actifs du nœud v . Nous considérons un réseau ad hoc sans fil à sauts multiples où un nombre fixe de nœuds sont connectés via un réseau dynamique et les nœuds sont autorisés à se déplacer dans un domaine rectangulaire bidimensionnel fini. Chaque nœud (v) se déplace avec une vitesse constante (\vec{u}_v) et une plage de transmission constante (r_v) . La distribution de la vitesse est supposée normale pour reproduire des scénarios plus réalistes, y compris des appareils mobiles aux capacités diverses. La direction de chaque vitesse est initialement donnée au hasard, mais lorsque le nœud atteint la limite du domaine, il rebondit à l'intérieur du domaine avec un angle de réflexion aléatoire de sorte que le nombre total de nœuds dans le domaine reste constant. La vitesse de nœud correspondante $(|\vec{u}_v|)$ indique à quel point un nœud peut se déplacer proportionnellement à (r_v) pendant un saut. La plage de transmission est supposée fiable, offrant une connectivité à 100% , et est généralement plus petite que la plage de communication maximale en localisation distribuée qui est liée à l'énergie de transmission et aux caractéristiques du canal [20]. Dans le réseau que nous considérons, ni collision ni retransmission entre les nœuds n'est autorisée. Nous supposons que la communication est interrompue si la destination se trouve dans la plage de transmission actuelle. Un nombre de génération est déterminé sur la base de la façon dont un nœud est lié au nœud source, c'est-à-dire le nombre de sauts. Une transmission est initiée lorsque le nœud source dans la génération initiale (G_0) diffuse un message à ses nœuds voisins dans la génération suivante (G_1) à l'intérieur et continue jusqu'à la génération suivante et ainsi de suite. Une transmission est considérée comme un succès si le message atteint le nœud de destination. Supposons que la transmission circule toujours vers une génération supérieure et que le message n'est ni transmis inversement en génération ni retransmis.

2.3.6 Processus de cardinalité multicast (MC) :

Le schéma de cardinalité multicast (MC) basé sur un processus de ramification stochastique provient de l'idée qu'il existe similitudes entre épidémie et diffusion des données. Le processus de ramification stochastique est l'un des modèles les plus fondamentaux pour décrire le stade initial d'une épidémie de maladie infectieuse lorsque le nombre de patients infectés reste nettement inférieur à celui de l'ensemble de la population [21]. Il permet de suivre la transmission dynamique plus individuellement, ce qui est différent des autres modèles de compartiment. Une partie importante du processus de ramification est le nombre reproducteur qui est le nombre d'infections secondaires. Ce nombre est une statistique de la structure du réseau et utilisé pour prédire et classer une épidémie en épidémie majeure ou mineure. Le degré de sommet d_v d'un nœud v est défini comme le nombre de voisins actifs dans $N'(v)$, représentant la densité du voisinage. La distribution de probabilité du degré k est p_k , qui est la fraction du nombre de nœuds avec un degré de k dans le réseau. Les premier et deuxième moments

d'infection moyens sont définis comme :

$$\langle k \rangle = \sum_{k=1}^{\infty} k P_k, \quad \langle k^2 \rangle = \sum_{k=1}^{\infty} k^2 P_k \quad (2.1)$$

et le nombre reproductif de base est calculé comme suit :

$$R = \frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle} \quad (2.2)$$

qui représente le ratio entre le nombre moyen de voisins secondaires et le nombre moyen de voisins (directs).

Proposition 1. Le nombre reproductif dans (2.2) peut être utilisé comme un indicateur avec lequel une extinction ultime de la transmission est prévue pour $R \leq 1$, et une épidémie majeure avec une probabilité positive de $R > 1$. Le nombre reproducteur à un nœud v dans G_n est calculé de la même manière que dans l'équation. (2.2) mais limité au trajet de transmission comme suit :

$$R_v = \frac{\langle k^2 \rangle_v - \langle k \rangle_v}{\langle k \rangle_v} \quad (2.3)$$

Un paramètre local, ε_n pour $n \geq 0$ comme :

$$\varepsilon_{n+1} = \frac{R_{v'}/R_v}{d_{v'}/d_v} = \frac{R_{v'} \cdot R_v}{d_{v'} \cdot d_v} \quad (2.4)$$

Où v est le nœud d'envoi en G_n et v' est le nœud récepteur en G_{n+1} . Ce paramètre local indique à quel point le nombre reproductif statistique change à mesure que la densité change de G_n à G_{n+1} comme nœud v envoie les données au nœud voisin actif v' . La probabilité de transmission initiale (p_0) au nœud source dans G_0 est fixé sur 1 pour diffuser le message à tous les nœuds voisins et la probabilité de transmission (p_{n+1}) au nœud v' en G_{n+1} est déterminé en fonction de la probabilité de transmission précédente (p_n) au nœud d'envoi et ajusté en fonction du long du chemin :

$$p_{n+1} = p_n + p_n(1 - \varepsilon_{n+1}) \quad (2.5)$$

Si p_{n+1} ($n > 0$) est supérieur à 1, nous le limitons à 1, c'est-à-dire la diffusion. Si p_{n+1} est inférieur à 0, nous le définissons sur $1/d_{v'}$ à un nœud v' envoyer les données à un seul nœud voisin. La probabilité de distribution en Eq. (2.5) augmente si la densité actuelle est supérieure à la densité au nœud précédent, ce qui signifie qu'il y a de meilleures chances de diffuser efficacement le message lorsqu'il y a relativement plus de voisins. Cependant, la probabilité est contrôlée en fonction des nombres reproductifs, c'est-à-dire que si le nombre reproductif statistique est supérieur à celui du nœud précédent, il supprime la distribution au nœud actuel.

2.3.7 Processus de sélection intelligente (Smart Selection) :

Le nœud source lance la diffusion des données en diffusant le message à tous ses voisins. Une fois la probabilité de transmission locale déterminée à un nœud intermédiaire comme dans la sous-section précédente, le message sera diffusé à $p_n \cdot d_v$ nœuds voisins actifs. Nous priorisons et sélectionnons les nœuds voisins actifs de deux manières différentes selon le scénario de réseau. La première option de sélection est basée sur la distance, $dist(v, v')$, entre l'émetteur v dans G_n et le récepteur v' dans G_{n+1} , dans la plage de transmission de v , où le nœud voisin le plus éloigné a la priorité la plus élevée. La sélection basée sur la distance a été l'une des affectations les plus courantes car le message est plus susceptible de se propager à un plus grand voisinage. La deuxième option est basée sur une corrélation négative entre les vitesses de l'émetteur et du récepteur, c'est-à-dire que nous priorisons le nœud voisin actif s'éloignant relativement plus rapidement du nœud émetteur. Pour nos besoins, nous utilisons l^2 -norme pour mesurer la différence entre \vec{u}_v et $\vec{u}_{v'}$:

$$|\vec{u}_v - \vec{u}_{v'}| = \sqrt{(\vec{u}_{v,x} - \vec{u}_{v',x})^2 + (\vec{u}_{v,y} - \vec{u}_{v',y})^2}$$

2.3.8 Cardinalité De Multidiffusion Et Sélection Intelligente :

L'algorithme détaillé de l'algorithme Cardinalité De Multidiffusion Et Sélection Intelligente (MCSS) est :

A. Algorithme MCSS :

Algorithme 2 : A. Algorithme MCSS

1 **Phase1.** Phase de cardinalité multicast MCSS au nœud v dans G_{n-1}
2 **Entrée :** $\langle k \rangle_w$, $\langle k^2 \rangle_w$, et d_w passant de w dans G_{n-1} . || Notez qu'il peut être vide si v dans G_n est la source, c'est-à-dire $n = 0$.
3 **Sortie :** Phase de cardinalité multicast MCSS au nœud v dans l'ensemble de voisins actifs est calculée pour former G_{n+1} .
4 **if** $v = D$ (*destination*) **then**
5 | arrêtez la transmission . || La destination reçoit le message.
6 **else**
7 | former un ensemble $N'(v)$ de tous les voisins actifs, v' , à travers la norme ZigBee [22] de telle sorte que $\text{dist}(v, v') < r_v$ et trouve
8 | $d_v = |N'(v)|$.
9 **end**
10 **if** $D \in N'(v)$ **then**
11 | envoyer le message à D et arrêter. || La destination est à un saut. $n = 0$.
12 **else**
13 | **if** $n = 0$ **then**
14 | | $p_n = 1$ || La destination a reçu le message.
15 | **else**
16 | | Calculez $\langle K \rangle_v$ et $\langle K^2 \rangle_v$ selon l'équation.(2.1).
17 | **end**
18 **end**
19 Calculez R_v et R_v^2 selon l'équation. (2.3) et (2.4) .
20 Calculez P_n selon l'équation. (2.5).
21 **Phase2.** Phase de Smart sélection MCSS au nœud v dans G_n
22 **Entrée :** L'ensemble $N'(v)$ et p_n dans **Phase1**..
23 **Sortie :** Sortie : La sélection basée sur l'un des deux critères de sélection.
24 **if** la sélection est basée sur la distance **then**
25 | trier $N'(v)$ par ordre décroissant de $\text{dist}(v, v')$ pour tous $v' \in N'(v)$
26 | et former le nouvel ensemble de voisins $N'(v)$ || Sélection MCSS– Distance.
27 **else**
28 | la sélection est basée sur la vitesse, trier $N'(v)$ dans l'ordre décroissant de la différence de vitesse (2.2) pour tous $v' \in N'(v)$ et former le nouvel ensemble de voisins $N'(v)$. || Sélection de vitesse MCSS.
29 **end**
30
31 Sélectionnez le premier p_n , d_n les nœuds de la triée $N'(v)$ former $N'(v)$.
32 Communiquez avec tous les nœuds de $N'(v)$.

B. Analyse de la complexité temporelle et de la complexité des messages

Nous analysons la complexité temporelle et la complexité du message à travers les deux théorèmes suivants :

Theorem 1 *La complexité temporelle du MCSS est $\theta(d_m)$ où $d_m = \max_n d_n$.*

Preuve : Pour la phase 1, la complexité temporelle est déterminée par former l'ensemble voisin actif, ce qui peut être fait en $\theta(d_m)$ où

$$d_m = \max_n d_n .$$

À partir des lignes 24 à 31 de la phase 2, il suffit de sélectionner des nœuds p_n , d_n pour appliquer l'algorithme de sélection, qui ne nécessite qu'un temps linéaire. Par conséquent, la complexité temporelle totale peut être $\theta(dm)$.

Par conséquent, la complexité temporelle totale est $\theta(d_m)$ où

$$d_m = \max_n d_n .$$

Puisque notre algorithme fonctionne de manière distribuée, il est important d'analyser la complexité du message. La complexité du message peut être déterminée par le théorème suivant.

Theorem 2 *La complexité des messages de l'algorithme est $\theta(|V|^2)$ et peut être réduite $\theta(|V|)$ où V est l'ensemble des nœuds du réseau.*

Preuve. Dans la phase 1, lorsqu'un nœud forme le voisin actif ensemble, il peut recevoir $\theta(|V|)$ nombre de messages de voisins dans le pire des cas. Par conséquent, la complexité du message pour former des ensembles voisins actifs est $\theta(|V|^2)$ où V est l'ensemble des nœuds du réseau.

Si seuls les voisins actifs répondent au message, la complexité du message peut être réduite à $\theta(|V|)$ car chaque nœud ne peut être voisin actif qu'une seule fois. Par conséquent, la complexité du message pour former des ensembles voisins actifs peut être réduite à $\theta(|V|)$. Comme pour la phase 2, dans le pire des cas, chaque nœud recevra le message provenant du nœud source jusqu'à ce qu'il atteigne la destination, et la complexité du message sera $\theta(|V|)$. Ainsi, la complexité totale du message sera $\theta(|V|^2)$ et peut être réduite à $\theta(|V|)$ en concevant soigneusement le message pour former l'ensemble voisin actif.

Implémentation

3.1 Introduction :

Ce chapitre est consacré à la réalisation et la mise en œuvre de notre application, nous allons présenter les outils de développement adoptés, soit l'environnement utilisé qui est Eclipse IDE, WindowBuilder que les langages de programmation (JAVA).

3.2 Outils de développement :

3.2.1 Eclipse (@ www.eclipse.org) :

Est un environnement de développement intégré (IDE) open source pris en charge par IBM. Eclipse est populaire pour le développement d'applications Java (Java SE et Java EE) et les applications Android. Il prend également en charge C / C++ , PHP, Python, Perl et d'autres développements de projets Web via des plug-ins extensibles. Eclipse est multiplateforme et fonctionne sous Windows, Linux et mac OS.

La plate-forme Eclipse qui fournit la base de l'IDE Eclipse est composée de plug-ins et est conçue pour être extensible à l'aide de plug-ins supplémentaires. Développée à l'aide de Java, la plateforme Eclipse peut être utilisée pour développer des applications client riches, des environnements de développement intégrés et d'autres outils. Eclipse peut être utilisé comme IDE pour tout langage de programmation pour lequel un plug-in est disponible. [23]

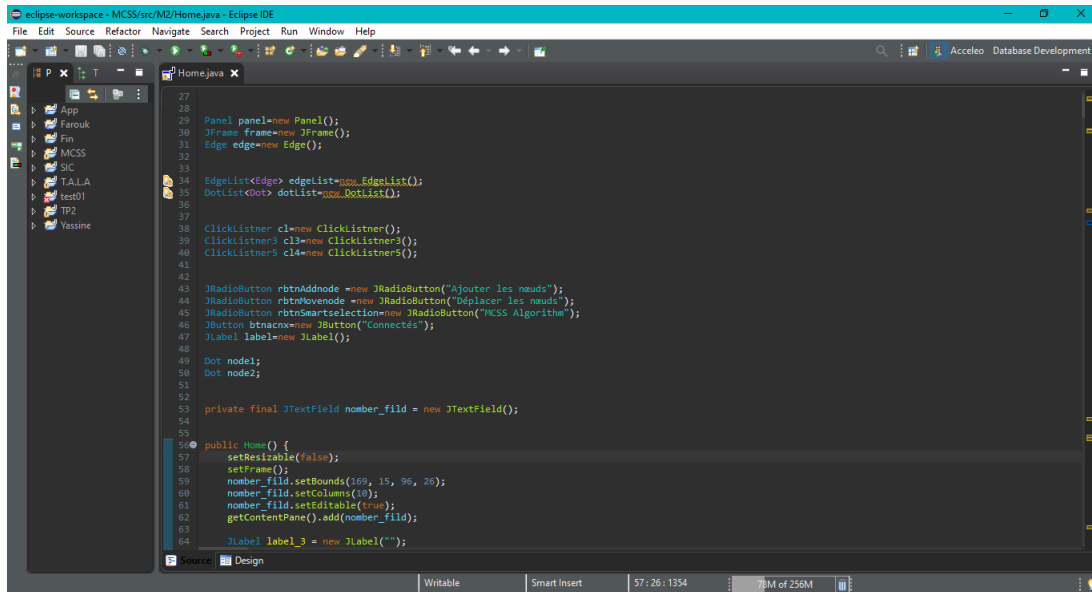


FIGURE 3.1 – fenetre de programmation Sur Eclipse..

3.2.2 Langage de programmation (Java) :

Java est un langage de programmation informatique à usage général qui est simultané, basé sur une classe, orienté objet et spécialement conçu pour avoir le moins de dépendances d'implémentation possible. Il est destiné à permettre aux développeurs d'applications « d'écrire une fois, de s'exécuter n'importe où » (WORA), ce qui signifie que le code Java compilé peut s'exécuter sur toutes les plates-formes qui prennent en charge Java sans avoir besoin de recompilation.

Java a été initialement développé par James Gosling chez Sun Microsystems et publié en 1995 en tant que composant central de la plate-forme Java de Sun Microsystems. Le langage tire une grande partie de sa syntaxe du C et du C ++, mais il a moins de fonctionnalités de bas niveau que l'un ou l'autre [24].

L'implémentation est présentée dans deux distributions différentes :

1. 1. Java Runtime Environment (JRE) [25] : qui contient les parties de la plate-forme Java SE (Java Standard Edition) requises pour exécuter des programmes Java et est destinée aux utilisateurs finaux.
2. Java Development Kit (JDK) [26] : qui est destiné aux développeurs de logiciels et comprend des outils de développement tels que le compilateur Java, Jar et un débogueur.

Environnement Java :

L'environnement d'exécution Java (JRE) est un ensemble d'outils logiciels pour le développement d'applications Java. Il combine la machine virtuelle Java (JVM), les classes de base de la plate-forme et les bibliothèques de support.

JRE fait partie du kit de développement Java (JDK), mais peut être téléchargé séparément. JRE a été initialement développé par Sun Microsystems Inc., une filiale en propriété exclusive d'Oracle Corporation.

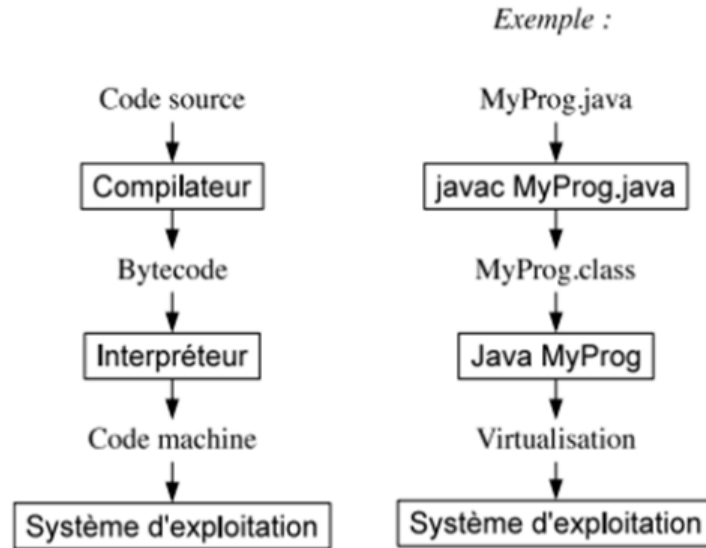


FIGURE 3.2 – architecture exécutable Code java..

3.2.3 Syntaxe :

La syntaxe de Java est largement influencée par C ++. Contrairement à C ++, qui combine la syntaxe pour une programmation structurée, générique et orientée objet, Java a été construit presque exclusivement comme un langage orienté objet. Tout le code est écrit à l'intérieur des classes et chaque élément de données est un objet, à l'exception des types de données primitifs (c'est-à-dire des entiers, des nombres à virgule flottante, des valeurs booléennes et des caractères), qui ne sont pas des objets pour des raisons de performances. Java réutilise certains aspects populaires de C ++ (tels que la méthode printf).

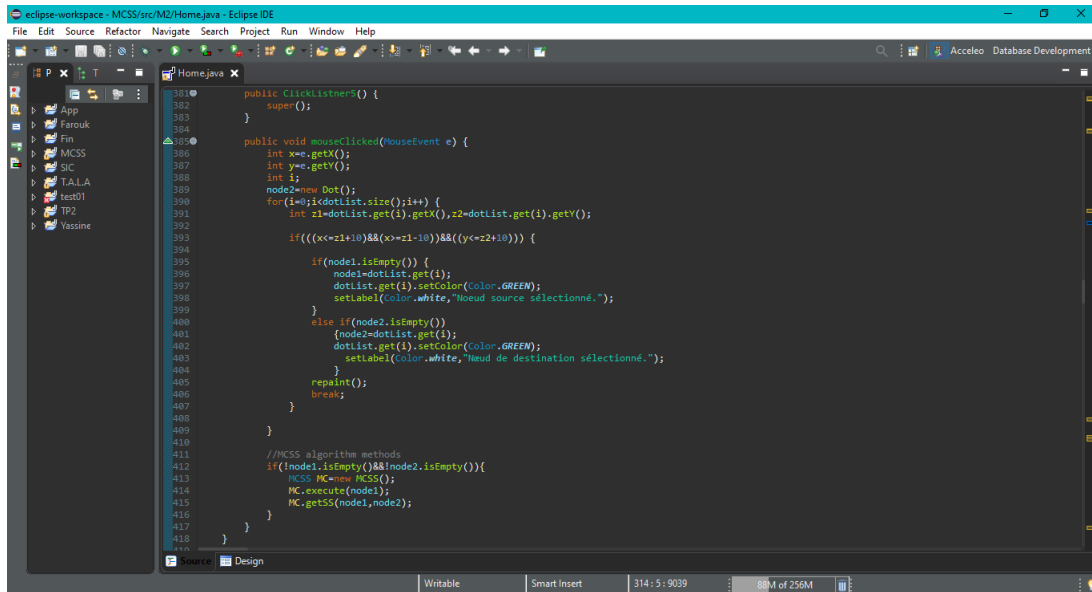


FIGURE 3.3 – projet Java Main..

3.2.4 WindowBuilder :

WindowBuilder est disponible sur Eclipse WindowBuilder. Il est composé de WindowBuilder Engine, SWT, eRCP, XWT & Swing Designer. WindowBuilder Engine fournit une API riche pour créer des concepteurs d'interface utilisateur. Il est très modulaire avec des dizaines de points d'extension. Il fournit un support enfichable pour différentes langues et analyseurs. Il fournit à la fois des cadres d'interface utilisateur basés sur Java (Swing, SWT / RCP, eRCP, GWT) et XML (XWT, GWT UiBinder, Android).

WindowBuilder est conçu comme un plug-in pour Eclipse et les différents IDE basés sur Eclipse (RAD, RSA, MyEclipse, JBuilder, etc.). Le plug-in crée une arborescence de syntaxe abstraite (AST) pour naviguer dans le code source et utilise GEF (Graphical Editing Framework) pour afficher et gérer la présentation visuelle. Le code généré ne nécessite aucune bibliothèque personnalisée supplémentaire pour être compilé et exécuté : tout le code généré peut être utilisé sans avoir installé WindowBuilder Pro. WindowBuilder Pro peut lire et écrire presque n'importe quel format et inverser la plupart des codes GUI (Graphical User Interface) Java manuscrits. Il prend également en charge l'édition de code de forme libre (apportez des modifications n'importe où... pas seulement dans des zones spéciales) et la plupart des reclassements utilisateur (vous pouvez déplacer, renommer et subdiviser des méthodes sans problème).

3.3 Implémentation de l'application :

3.3.1 Interfaces de l'application :

Cette page offre un aperçu de l'application :



FIGURE 3.4 – Interface d'accueil d'application.

1. Pour créer un nombre spécifique de nœuds dans le réseau.
2. Créer des nœuds.
3. Pour changer la position aléatoire de n'importe quel nœud.
4. 4. Tous les nœuds du réseau se connecteront pour voir le voisin.
5. Pour exécuter l'algorithme MCSS.
6. Numéro de reproduction : qui représente le rapport entre le nombre moyen de voisins secondaires et le nombre moyen de voisins (directs).
7. Paramètre local indique à quel point le nombre reproductif statistique change à mesure que la densité change de G_n à G_{n+1} comme nœud v envoie les données au nœud voisin actif v' .
8. Premiers moments d'infection.
9. Deuxièmes moments d'infection.
10. Probabilité de transmission.

3.3.2 Quelques exemple de code source :

```
274 protected class ClickListener extends MouseAdapter{
275     ClickListener(){
276         super();
277     }
278
279     public void mouseClicked(MouseEvent e) {
280
281         for(int i=0;i<Integer.parseInt(number_fild.getText());i++){
282
283             Dot dot=new Dot();
284
285             int x=e.getX()+((int)(Math.random()*900);
286             int y=e.getY()+((int)(Math.random()*900);
287             dot.setColor(color.black);
288             if(x>1350){
289                 int dif=x-1350;
290                 x=x-(2*dif);
291             }else if(y>620){
292                 int dif=y-620;
293                 y=y-(2*dif);
294             }
295
296             dot.setX(x);
297             dot.setY(y);
298
299             dotList.add(dot);
300
301         }
302         repaint();
303     }
304 }
305
306
307
308
```

FIGURE 3.5 – Code source pour dessiner des nœuds aléatoires..

```
316 protected class ClickListener3 extends MouseAdapter{
317
318     int x,y,i=0;
319     int a,b,j;
320     boolean pressed;
321
322     ClickListener3() {
323         super();
324     }
325
326     public void mousePressed(MouseEvent e) {
327         x=e.getX();
328         y=e.getY();
329         for(this.i=0;this.i<dotList.size();this.i++) {
330             a=dotList.get(this.i).getX();
331             b=dotList.get(this.i).getY();
332             if(((x<a+10)&&(x>=a-10))&&((y<b+10))) {
333                 pressed=true;
334                 dotList.get(i).setColor(Color.GREEN);
335
336                 setLabel(Color.RED,"Faites glisser votre souris vers n'importe quel emplacement et relâchez pour déplacer le nœud.");
337                 repaint();
338                 break;
339             }
340         }
341     }
342
343     public void mouseReleased(MouseEvent e) {
344         if(pressed) {
345             x=e.getX();
346             y=e.getY();
347             dotList.get(this.i).setX(x);
348             dotList.get(this.i).setY(y);
349
350             setLabel(Color.GREEN,"Le nœud a été déplacé avec succès.");
351             this.moveEdge();
352             dot.setDefaultColor();
353             repaint();
354         }
355     }
356 }
```

FIGURE 3.6 – Code source pour déplacer le nœud aléatoire.

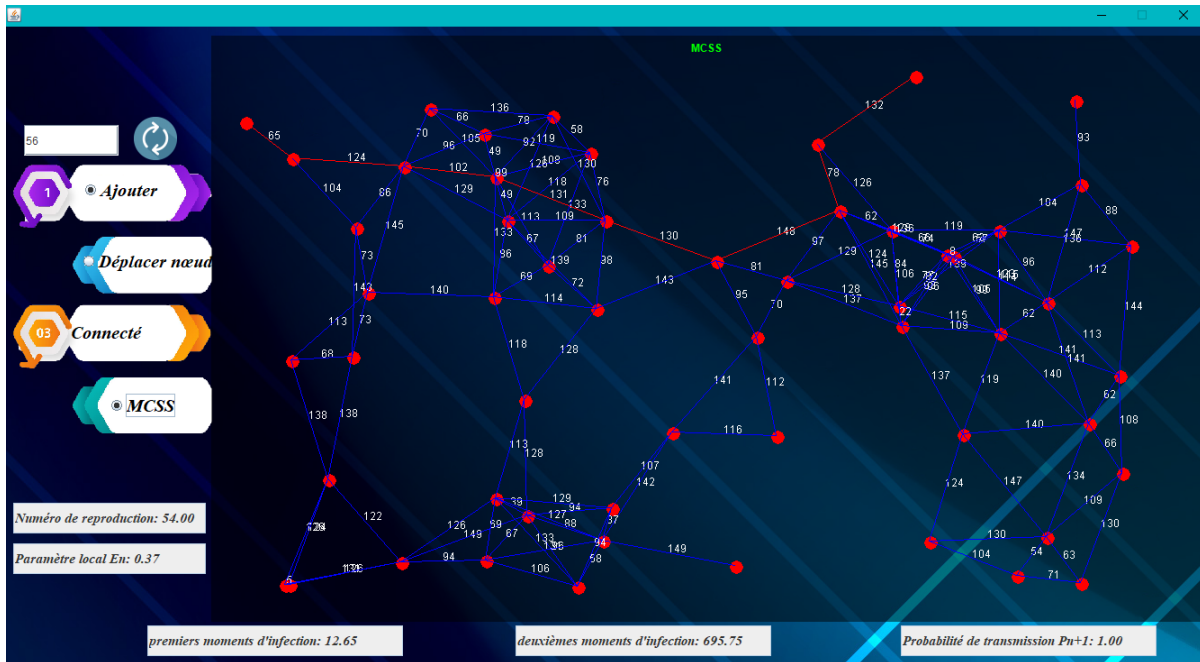


FIGURE 3.7 – l'exécution de l'algorithme.

3.4 Conclusion :

Dans ce projet, nous avons proposé un nouveau modèle de communication ad hoc sans fil, appelé Multicast Cardinality et Smart Selection, composé des processus Multicast Cardinality (MC) et Smart Selection (SS). Puisque nous sommes dans un réseau dynamique, cela signifie que les nœuds continuent de se déplacer de manière différente et à des vitesses différentes, cela signifie qu'il y aura de nombreux scénarios dans le réseau, notre algorithme facilite l'envoi / la réception d'informations entre les nœuds, pour prendre en charge les données échange en topologie à commutation rapide. Notre approche vise à équilibrer l'accessibilité, l'utilisation des nœuds et le facteur de branchement moyen dans la diffusion de données multi-sauts en régulant localement la probabilité de transmission et en sélectionnant de manière adaptative les nœuds voisins.

Bibliographie

- [1] Mohapatra P. (ed.), Krishnamurthy S.V. (ed.) - AD HOC Networks Technologies and Protocols-Springer (2004).
- [2] C. S. R. Murthy and B. Manoj, Ad Hoc Wireless Networks : Architectures and Protocols. Prentice Hall PTR, May 2004.
- [3] C. E. Perkins, Ad Hoc Networking. Addison Wesley Professional, December 2000.
- [4] J.Van der Meerschen. Hybridation entre les modes ad-hoc et infrastructure dans les réseaux de type Wi-Fi. Université Libre de Bruxelles. 2006
- [5] D. Balfanz, D. K. Smetters, P. Stewart, and H. C. Wong. Talking to strangers : Authentication in ad-hoc wireless networks. In Proceedings of the Network and Distributed System Security Symposium (NDSS), San Diego, CA, February 2002.
- [6] M. Zapata and N. Asokan. Securing ad hoc routing protocols. In Proceedings of the ACM Workshop on Wireless Security (WiSe 2002), Atlanta, GA, September 2002.
- [7] D. Balfanz, D. K. Smetters, P. Stewart, and H. C. Wong. Talking to strangers : Authentication in ad-hoc wireless networks. In Proceedings of the Network and Distributed System Security Symposium (NDSS), San Diego, CA, February 2002.
- [8] R. Meraihi ; "Gestion de la qualité de service et contrôle de topologie dans les réseaux Ad Hoc" ; Thèse de doctorat, École nationale supérieure des télécommunications, Paris, 2004.
- [9] David Espès, Protocoles de routage réactifs pour l'optimisation de bande passante et la garantie de délai dans les réseaux Ad Hoc mobiles THÈSE En vue de l'obtention du DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE, Le 27 novembre 2008.
- [10] 9 C. Cheng, R. Riley, S. Kumar et J.J. Garcia-Luna-Aceves. A Loop-Free BellmanFord Routing Protocol without Bouncing Effect. ACM SIGCOMM'89. Sept 1989.
- [11] G. Malkin. RIP Version 2-Carrying Additional Information. RFC 1388. Internet Engineering Task Force. January 1993.

-
- [12] (The electrical engineering handbook series) Mohammad Ilyas - The Handbook of Ad hoc Wireless Networks-CRC Press (2003).
- [13] D. Balfanz, D. K. Smetters, P. Stewart, and H. C. Wong. Talking to strangers : Authentication in ad-hoc wireless networks. In Proceedings of the Network and Distributed System Security Symposium (NDSS), San Diego, CA, February 2002.
- [14] A. K. Basu - Introduction to Stochastic Process-Alpha Science (2003).
- [15] J. L. Doob - Stochastic processes-Wiley-Interscience (1990).
- [16] A Gossip-Style Failure Detection Service, Dept. of Computer Science, Cornell University, 4118 Upson Hall, Ithaca, NY 14853.
- [17] <https://link.springer.com/article/10.1023/A:1013763825347> 14-04-2020.
- [18] Luo Junhai, Ye Danxia, et al., Research on topology discovery for IPv6 networks, IEEE, SNPD 2007 3 (2007) 804–809.
- [19] J. Kuruvila, A. Nayak, and I. Stojmenovic, “Hop Count Optimal Position-ased Packet Routing Algorithms for Ad Hoc Networks with a Realistic Physical Layer,” IEEE Journal in Selected Areas in Communications, vol. 23, no. 6, pp. 1267-275, 2005.
- [20] . F. Braucer and C. C. Chavez, “Mathematical Models in Population,” Springer Biology and Epidemiology, 2011.
- [21] V. Deep, T. Elarabi, “Efficient IEEE 802.15.4 ZigBee Standard Hardware Design for IoT Applications,” in Proc. IEEE International Conference on Signal and Systems, 2017.
- [22] <https://www.tutorialspoint.com/eclipse/index.htm> 8-5-2020.
- [23] <https://howtodoinjava.com/java/basics/what-is-java-programming-language/> 8-5-2020.
- [24] <https://www.techopedia.com/definition/5442/java-runtime-environment-jre>. 08-05-2020.
- [25] <https://www.techopedia.com/definition/5594/java-development-kit-jdk> 08-05-2020.