Analysis and Design of Protocols for Clustering in Mobile Ad Hoc Networks

Dissertation submitted to the department of

Computer Science and Engineering

of

National Institute of Technology Rourkela

in partial fulfilment of the requirements
for the degree of

Doctor of Philosophy

by

Suchismita Chinara

(Roll- 50606001)

under the supervision of

Prof. Santanu Kumar Rath

Department of Computer Science and Engineering

National Institute of Technology Rourkela

Rourkela- 769008, India

2011

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, India. `www.nitrkl.ac.in`

Dr. Santanu Kumar Rath
Professor

July 27, 2011

Certificate

This is to certify that the work in the thesis entitled *Analysis and Design of Protocols for Clustering in Mobile Ad Hoc Networks* by *Suchismita Chinara* is a record of an original research work carried out by her under my supervision and guidance in partial fulfilment of the requirements for the award of the degree of *Doctor of Philosophy* in *Computer Science and Engineering.* Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Santanu Kumar Rath

Acknowledgment

This dissertation would not have been possible without the constant encouragement I have received from my supervisor *Prof. Santanu Kumar Rath.* He has constantly encouraged me to remain focused on achieving the goal. His observations and comments helped me to establish the overall direction of the research and to move forward with investigation in depth. No words can express my gratitude to one who has been a guru in the truest sense.

I am very much indebted to the Doctoral Scrutinity committee members *Prof. Ashok Kumar Turuk*, *Prof. Bansidhar Majhi* and *Prof. (Mrs.) Sushmita Das* for their time to provide more insightful opinions into my research. I am also thankful to *Prof. Sanjay Kumar Jena* and all the Professors of the department for their in time support, advise and encouragement .

I am really thankful to all my fellow research colleagues. My sincere thanks to Mr.Saroj Panigrahi who has spared his time to help me in the documentation of the thesis, I am truly indebted.

I do acknowledge the academic resources that I have received from NIT Rourkela. I also thank the administrative and technical staff members of the Computer Science Department for their intime support.

I feel proud to acknowledge my parents for their throughout support and motivation in my career for whom I am today and always.

Last, but not the least, I would like to dedicate this thesis to my husband and my children *Pratik* and *Gulmini* for their unconditional love, patience and cooperation.

Suchismita Chinara

Abstract

Communication in mobile ad hoc networks (MANET) without having any fixed infrastructure has drawn much attention for research. The infrastructure based cellular architecture sets up base stations to support the node mobility. Thus, mapping the concepts of base stations into MANET could meet its challenges like limited battery power, scalability, available band width etc.. This leads to the design of logical clusters, where the cluster heads in every cluster play the role of base station. The cluster heads also form the virtual back bone for routing the packets in the network. In this thesis, simulation based survey has been made to study the strengths and weaknesses of existing algorithms that motivated for the design of energy efficient clustering in MANET.

Neighbour Detection Protocol (NDP) has been designed to help the nodes to probe their immediate neighbours. In this protocol, every node broadcasts its own information to the network, so that it is received by a node that lies within its transmission range. The receiver senses its neighbours and updates its neighbour table from time to time. This protocol is validated through simulation by using Colour Petri Nets (CPN) prior to its implementation.

Topology Adaptive Clustering Algorithm (TACA) has been proposed, that uses the node mobility and its available battery power for calculating the node weights. A node having the highest weight among its immediate neighbours declares itself as the volunteeer cluster head. As the current head consumes its battery power beyond a threshold, non-volunteer cluster heads are selected locally. The algorithm aims to utilise the battery power in a fairly distributed manner so that the total network life time is enhanced with reduced cluster maintenance overhead.

During the process of clustering, some isolated heads without having any members are formed. This increases the delay in communication as the number of hops in the routing back bone is increased. A Transmissiion Range Adjustment Protocol (TRAP) has been proposed, that allows the isolated nodes to adjust their ranges to remain connected with existing cluster heads. The results show that, TRAP reduces the delay in communication by reducing the number of cluster heads in the network.

Validation for the base protocol NDP and algorithm TACA are made through simulation by using the CPN tools. Each of the proposed work is evaluated separately to analyse their performances and compared with the competent results.

***Keywords*:** Neighbour detection, cluster maintenance, volunteer head, non-volunteer head, re-affiliation, re-election, network life time, topology control, coloured petri nets.

Contents

List of Abbreviations

Ack	Acknowledgement
ACKSTORE	Acknowledgement Store
AMPS	Advance Mobile Phone Services
BID	Battle Field Information Distributed System
CCI	Cluster Contention Interval
CTAB	Cluster Table
CPN	Coloured Petri Nets
CBTC	Cone Based Topology Control
CDS	Connected Dominating Set
CTR	Critical Transmission Range
DCA	Distributed Clustering Algorithm
DMAC	Distributed Mobility Adaptive Algorithm
GDMAC	Generalized Distributed Mobility Adaptive Algorithm
GPS	Global Positioning System
HF	High Frequency
HF ITF	High Frequency Intra Task Force
HC	Highest Connectivity
IDS	Independent dominating Set
I/O Port	Input Output Port
LAA	Link Activation Algorithm
LCA	Linked Cluster Algorithm
LILT	Local Information Local Topology

LINT	Local Information No Topology
LID	Lowest ID
MAC	Medium Access Control
MSGSTORE	Message Store
MANET	Mobile Ad hoc Network
MOBIC	Mobility Metric Based Algorithm
MFR	Most Forward with Fixed Radious
MVR	Most Forward with Variable Radious
NFP	Nearest with Forward Progress
NC	Neighbour Confirmation
NDPAK	Neighbour Detection Packet
NDP	Neighbour Detection Protocol
NRQ	Neighbour Request
NTAB	Neighbour Table
NAC	Nighbor Acknowledgement
PRNET	Pcaket Radio Network
PDA	Personal Digital Assistant
QoS	Quality of Service
RF	Radio Frequency
TDMA	Time Division Multiple Access
TACA	Topology Adaptive Clustering Algorithm
TRAP	Transmission Range Adjustment Protocol
TRCP	Transmission Range Control Protocol
VCA	Vertual Cellular Architecture
VCB	Vertual Cellular Backbone
WBCA	Weight Based Clustering Algorithm
WCA	Weighted Clustering Algorithm
WSN	Wireless Sensor Network

List of Symbols

$\lvert V \rvert$	Cardinality of set V
$\rightarrow$	With Respect To
δ	Threshold
$\sum$	Summation
@	Symbol for Timed Token in CPN
$\in$	Belongs To
$'$	Symbol for Current marking in CPN
#	Symbol to extract a field from the place in CPN
Δ	A Difference Factor
Γ	Set of Neighbors
α	A Weighing Factor
β	A Weighing Factor
γ	A Weighing Factor
μ	Micro
π	Pie
τ	A Small Time Period

List of Figures

Chapter 1

Introduction

Over the decades, the use of personnal communication devices like mobile phones, personal digital assistants($PDAs$) and mobile computers have taken an exponential growth. This tendency is reinforced when the cost of these small devices are reduced and further equipped with one or more wireless interfaces. The wireless interfaces allow the devices to get connected with the access points available in various location such as air ports, railway stations, restaurants, city centers etc.. At the same time, they also enable the devices to interconnect directly with each other in a decentralized way and *self-organise* into "*Ad Hoc Networks*" [1].

The term *ad hoc* is a Latin adjective which means *special purpose.* This term was borrowed by few computer scientists to characterise a special purpose network called the *ad hoc network.* The researchers have provided different definition to the ad hoc networks. As per The Institute of Electrical and Electronics Engineers (IEEE) 802.11, " An ad hoc network is a network composed of communication devices within mutual transmission range of each other via the wireless medium" [2]. The communication devices are termed as nodes in this literature. As the mobility of the nodes are taken into consideration, the same is redefined by Gerla [3] as:

"*A mobile ad hoc network (MANET) is a collection of mobile nodes that* *dynamically self organise into peer-to-peer wireless network without using any pre-existing infrastructure.*"

The term *self-organise* is a key term in this definition. This term mostly refers to the routing of the packets in the network in the absence of any fixed infrastructure. The nodes of the MANET organise themselves to route the packets of the neighboring nodes by creating a multi-hop networking scenario while on-the-fly. Thus, the specially designed nodes should have the capability of a router to forward the packets in addition to its normal job of a transmitter or receiver. The term *self-organise* is also equally important when the topology control is taken into consideration. In this context, the nodes try to adjust their transmission ranges to remain connected to each other in the dynamic network.

1.1 Technologies of Mobile Ad Hoc Networks

The present day communication system demands a high speed and reliable network where a wired backbone can be connected to several wireless networks as in figure 1.1. The category of wireless networks could be cellular networks, wireless personal area network (WPAN), wireless local area network (WLAN) or mobile ad hoc network (MANET). Most wireless technologies operate in the Unlicensed Industrial Scientific and Medical (ISM) 2.4 GHz band. For this reason, the network may suffer interference from microwave ovens, cordless telephones, baby monitors and similar other appliances that use almost the same band.

In the earlier versions of mobile ad hoc networks, the packet radios sponsored by DARPA were used for communication. However, currently three main communication standards with ad hoc capabilities are used to address a specific range of commercial applications. They are the *IEEE 802.11* family of protocols, the *high-performance LAN* (HiperLAN) protocols and *Bluetooth* [2].

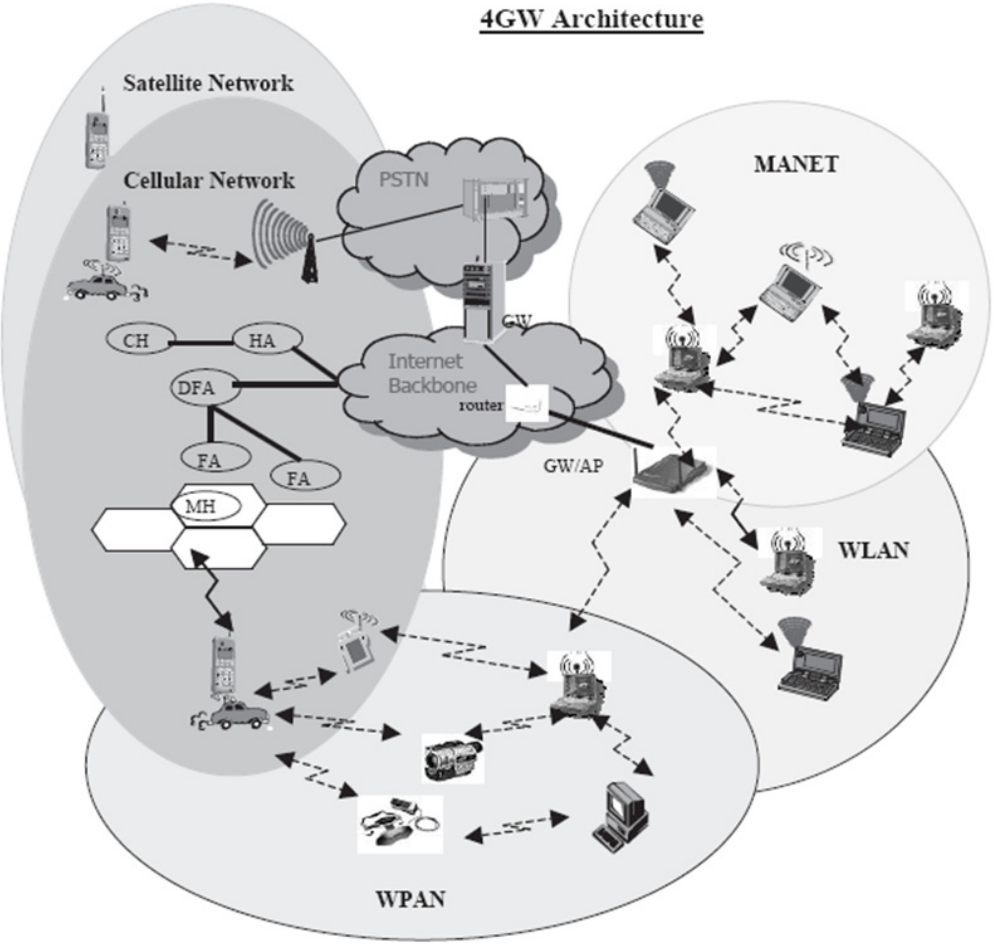


Figure 1.1: Current generation networks [4]

1.1.1 IEEE 802.11

In 1997, The Institute of Electrical and Electronics Engineers (IEEE) adopted the first digital wireless data transmitting standard, named IEEE 802.11 [5]. The purpose of the IEEE 802.11 standard was to foster industry product compatibility between WLAN product vendors. The most popular and widely used Wi-Fi networking technology is based on the IEEE 802.11 specifications.

- This IEEE 802.11 family has many extended versions, one of which is the 802.11a. The IEEE 802.11a is the only task group that works in 5GHz band and data rates upto 54 Mb/s plus error correction code. It uses

Orthogonal Frequency Division Multiplexing (OFDM) air interface.

- Another task group the IEEE 802.11b operates in 2.4 GHz band with data rates upto 11Mb/s. This is the most used Wi-Fi standard which has been heavily studied in the framework of MANET.

Under the IEEE 802.11 standard, the mobile communicating devices in a network can work in two different modes. They are the infrastructure mode and ad hoc mode. Infrastructure mode wireless networking connects a wireless network to a wired network. It also supports central connection points for WLAN clients. A wireless access point is required for infrastructure mode wireless networking. Hence it offers the advantage of scalability, centralized security management and better connectivity. In the ad hoc mode of wireless networks, the nodes can directly communicate with each other without using any access point. To set up an ad hoc wireless network, each wireless adapter must be configured for ad hoc mode versus the infrastructure mode. An ad hoc network tends to feature a small group of devices all in very close proximity to each other. In this network the performance degrades as the number of nodes increases. Ad hoc networks can not bridge to wired LANs or the the internet without the presence of a special purpose gateway node.

1.1.2 HiperLAN

The European counterparts to the IEEE 802.11 standards are the high performance radio LAN (HIPERLAN) standards defined by the European Telecommunication Standards Institute (ETSI). While IEEE 802.11 standards can use either radio access or infrared access, the HIPERLAN standards are based on radio access only [6]. Four standards have been defined for wireless networks by the ETSI.

- **HIPERLAN/1:** It is a wireless radio LAN (RLAN) without a wired infrastructure, based on one-to-one and one-to-many broadcasts. It is well studied for both ad hoc and infrastructure based networks. The standard covers the physical layer and the media access control part of the data link layer. A new sub layer is associated with it called the channel access and control layer (CAC). This sub layer deals with the access requests to the channel which is accomplished depending on the usage of the channel and priority of the request. The features of the standard includes the transmission range of about 50m at a rate of 23 Mb/s. It also supports asynchronous and synchronous traffic with slow mobility of 1.4 mts/s.

- **HIPERLAN/2:** It has a transmission range of about 200mts for wireless Asynchronous Transfer Mode (ATM) networks. It offers a wide range of data rates from 6 Mbps to 54 Mbps and uses 5GHz radio frequency. It supports centralised and direct modes of operation. The former is used in the cellular network topology where as the later is used in ad hoc network topology. Basic services in HiperLAN/2 are data, sound and video transmission with emphasis given on the quality of service (QoS). This standard covers physical, data link control and convergence layers. The convergence layer takes care of service dependent functionality between data link control and network layer. This layer can also be used on the physical layer to connect IP and ATM networks. This feature makes HiperLAN/2 suitable for the wireless connection of various networks.

- **HIPERLAN/3:** It is also called as HIPERACCESS network that enables establishment of outdoor high speed radio access networks poviding fixed radio connections to customer premises. It has a range of 500mts and provides a data rate of 25Mbps. This network can be used for wireless local loop (WLL) communications.

- **HIPERLAN/4:** It is also called as the HIPERLINK standard that provides high speed radio links for point-to-point static interconnections. The transmission has a range of about 200mts and operates on the 17GHz frequency range. It provides a data rate of 155 Mbps.

1.1.3 Bluetooth

Bluetooth is a technology for wireless body area network (WBAN) and wireless personal area network (WPAN) that provides short range radio links between portable devices such as mobile PCs and mobile phones. Bluetooth specifies 10mts radio range and supports upto seven devices in a master slave mode. The master permits slaves to transmit by allocating slots for voice or data traffic. Bluetooth uses a combination of circuit switching and packet switching [7].

Bluetooth wireless networks are classified into two network topologies named *piconets* and *scatternets*. In a piconet, two or more slave devices can share the same frequency hopping sequence. A *piconet* comprises one master station and up to seven active slaves can participate in data exchange. So it can form a point-to-point or point-to-multipoint design. A direct link can exist between a master and a slave but not between slaves. So data exchange between the slaves has to be routed through the master. Independent piconets that have overlapping coverage areas may form a *scatternet*. A scatternet exists when a station is active in more than one piconet at the same time. A slave can communicate with the different piconets it belongs to, in a time-multiplexing mode. When two piconets overlap, the master of one piconet serves as the slave of the other piconet. No device can serve as a master of two piconets. When more piconets overlap with each other, one master serves as a slave of two piconets. Thus multihop communication can be achieved through the scatternet concept, where several masters from different piconets can set up links among each other. The bluetooth applications span from wireless headset to PDAs, networked computer peripherals like printers, scanners, digital cameras etc. Bluetooth uses frequency hopping spread spectrum (FHSS) technique that breaks the data into small packets for data communication.

1.2 Applications of Mobile Ad Hoc Networks

The deployment of a mobile ad hoc network is easy due to the absence of setting up any infrastructure for communication. Mostly such kind of network is required in militaly application and emergency rescue operations. But slowly it has been emerged into the areas of gaming, sensing, conferencing and collaborative and distributed computing [6]. This dynamic network is yet to capture most of the commercial applications. Research is still going on in this direction so that the MANET can be deployed in any area where a faster and cheaper network can be setup instantly for data communication. Mostly, the distributed applications where the entities are strongly coupled, are not adaptive to the connection disruptions. In this regard the mobile ad hoc networks, where the nodes are highly dynamic in nature, can be most suitable for the loosly distributed applications. Here are some of the applications of mobile ad hoc networks:

- **Military Services** Military services are one of the most discussed and common application area of mobile ad hoc networks where installation of any fixed infrastructure is not possible in the enemy territories or inhospitable terrains. In this environment MANET provide the required communication mechanism in no time. Here, the soldiers are considered to be the mobile nodes. So the network is required to remain connected even though the soldiers move freely. This support is provided by the MANET. Another application in this area can be the coordination of the military objects and the personnel in the battle field. For example, the leader of a group of soldiers may want to pass a message to all the soldiers or a group of soldiers involved in the operation. In this situation, a secure and reliable routing protocol should be able to do the job.
- **Emergency Services** In certain situations that are unexpected and unavoidable like search and rescue, crowd control, disaster recovery and commando operations, the use of mobile ad hoc networks is very much

suitable. The major factors that favour the deployment of MANET in such situations are its self configuration with minimum overhead, unavailability of fixed or centralised infrastructure as well as freedom and flexibility of mobility of the nodes. Since the ad hoc networks require minimum initial network configuration, it can also be deployed in situations where conventional infrastructure based communication is disturbed due to natural calamity or any other reason.

- **Collaborative and Distributed Computing** The use of mobile ad hoc network is very much necessary in such situation where a group of researchers want to share their research findings or share their research materials during a conference or on-the-fly. The reliability of data transfer is much important in this case than the data security. So MANET should be able to provide reliability in data communication.

- **Sensing and Gaming** Sensor network is a special case of ad hoc networks where mobility is generally not considered. However the battery power is a key factor in sensors. Each sensor is equipped with a transceiver, a small micro-controller and an energy source. The sensors relay information from other devices to transport data to a central monitor. The sensors are used to sense the environmental condition such as temperature, pressure, humidity etc. In this case they form an ad hoc network to collect intended information. The mobility can also be incorporated into the sensor network where they are meant to study the behaviour of tornados or to study the behaviour of patients in the hospital.

- **Personal Area Networking** Personal communicating devices like laptops, PDAs, mobile phones create a network to share data among each other called the personal area network (PAN). The PAN cover a very short range for communication and can be used for ad hoc communication among the devices or for connecting to a backbone network.

1.3 Challenges of Mobile Ad Hoc Networks

Though the deployment of a mobile ad hoc network does not take much time or price, but its functionality faces few major challenges. The following points are discussed with regards to the MANET challenges:

- **Routing in Dynamic Topology:** In MANET, the presence of node mobility changes the link of connectivity between the nodes very frequently. The existing conventional Bellman Ford routing algorithm or classic Link State algorithms are not applicable for such dynamic network where the topology changes with the free movement of the nodes. So new routing protocols are required for MANET that can be adaptive with the topology changes as well as frequent neighbor changes.

- **Scalability:** Scalability of a network can be defined as the ability to provide an acceptable level of service even with a large number of nodes. In mobile ad hoc networks, the nodes are constrained with limited battery power, computation capability and storage capacity. As the network size increases, the number of packets forwarded by each node also increases. This drains the node resources fast, making it dead in a short period. Similarly, topology maintenance overhead in a scalable dynamic network is another challenging issue. This ultimately affects the QoS of the network.

- **Energy Efficiency:** The conventional network protocols never consider power consumption as an issue since they assume unlimited power supply for the static nodes and the routers. However, the portable mobile devices are mostly operated by the batteries whose life span is very limited. Further, the nodes in the MANET have to perform the role of an end system (transmitter or receiver) as well as an intermediate system (forwarding packets of other nodes) which causes more battery drainage. So, the design of the protocols for MANET must be energy efficient.

- **Security and Privacy:** Mobility implies higher security risks such as peer-to-peer network architecture or a shared wireless medium accessible to both legitimate users and malicious attackers. Through neighbor identity authentication, the nodes can know if the neighbor is friendly or hostile. But still, maintaining privacy and security in a network where multiple nodes operate in a self organised manner are challenging issues.
- **Lack of central Infrastructure:** There exist several solution in a cellular network to handle the mobility of the nodes while routing is the major concern. But, the lack of any central facility in mobile ad hoc network decreases the routing efficiency as well as the throughput. Thus, in order to enhance the routing efficiency and resource management of the network, design of the network with virtual central infrastructure could be a wise solution.

1.4 Clustering in Mobile Ad Hoc Networks

The research is still going on to meet the above challenges in mobile ad hoc network. Routing in MANET can take place either in a flat structure or in a hierarchical structure [1]. It has been proved that the packet delivery delay is reduced in a hierarchical structure than that of in the flat structure resulting in a better routing efficiency [8]. In large networks, the flat routing structure produces excessive information flow which can saturate the network. Hierarchical routing is an interesting solution for handling such a scalable network where only selected nodes take the responsibility of routing [9]. Thus, topology management plays a vital role prior to the actual routing in MANET. Cluster based structure, which is a typical synonym of the hierarchical structure in network topology, has become popular since last few decades to improve the routing efficiency in a dynamic network. This process deals with the formation of a virtual cellular backbone (VCB), where the cluster heads map to the base stations in cellular architecture [10].

1.4.1 Definition

Clustering in MANET can be defined as the virtual partitioning of the dynamic nodes into various groups. Groups of the nodes are made with respect to their nearness to other nodes. Two nodes are said to be neighbor of each other when both of them lie within their transmission range and set up a bidirectional link between them.

Clusters in MANET can be categorised as overlapping clusters or non-overlapping clusters as is shown in figure 1.2. The small circles represent the wireless nodes in the network. The lines joining the nodes denote the connectivity among them. Cluster control structure forms the virtual backbone of communication where cluster heads are the communication hot spots. The cluster head works as the local coordinator for its member nodes and does the resource management among them similar to a base station of cellular architecture. These cluster heads are responsible for inter cluster and intra-cluster communication. Inter cluster communication is made possible through the gateway nodes. A gateway node is a node that works as the common or

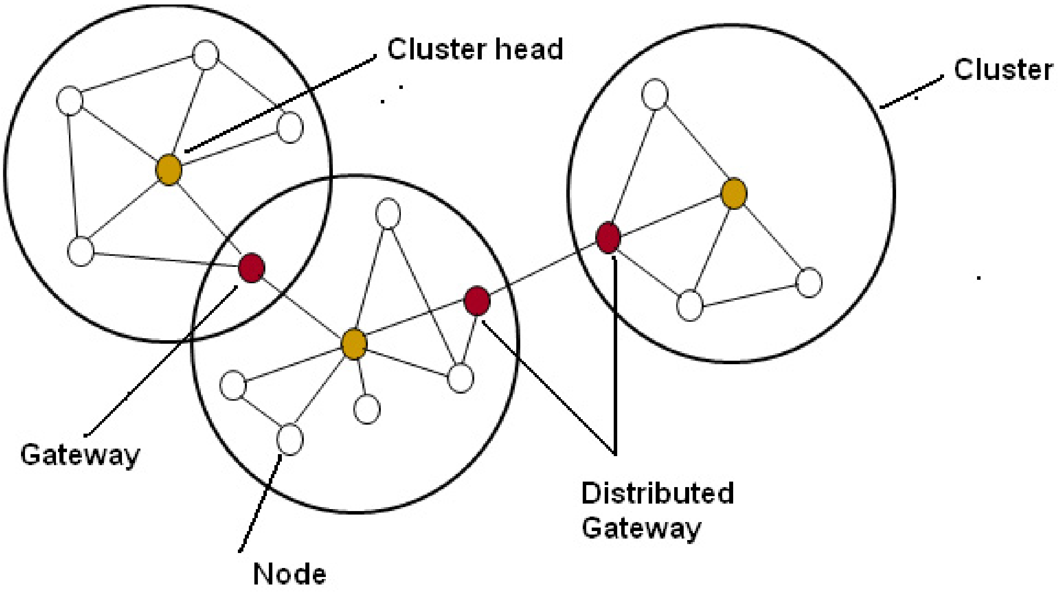


Figure 1.2: Overlapping and non-overlapping clusters

distributed access point for two cluster heads. When a node lies within the transmission range of two cluster heads and supports inter cluster communication, it is called the ordinary gateway for two corresponding clusters. A node having one cluster head as an immediate neighbor in addition to which it can reach a second cluster head in two hops is a distributed gateway that is linked to another distributed gateway of other cluster. Both of the distributed gateways provide the path for the inter-cluster communication as well. The ordinary nodes of the cluster are the immediate neighbors of the cluster heads. They have the capability of serving as either a head or a gateway whenever selected to do so.

Depending on the diameter of the clusters, there exist two kinds of cluster control architectures, known as one-hop clusters and multi-hop (d-hop) clusters. In one-hop clusters, every member node is at most 1-hop distance away from the central coordinator called as the cluster head. Thus, all the member nodes remain at most two hops distance away from each other within a logical cluster. But in multi hop clusters, the constraint of immediate neighborhood of members from the head is eliminated by allowing the nodes to be present at most d-hop distance away from each other to form a cluster [3, 11, 12].

The nodes in the MANET can be either in the flat structure or in hierarchical structure. Typical mobile ad hoc network is shown in figure 1.3 and figure 1.4 with flat and hierarchical structure respectively. It may be noted from the figures that in the flat architecture of MANET, every node bears equal responsibility to act as a router for forwarding the packets to every other node. So a great amount of message flooding takes place to achieve efficient routing. In return, such message flooding reduces the MAC layer efficiency to certain extent. Cluster control structure can be one possible solution to improve such MAC layer efficiency [1]. But, in the hierarchical structure, nodes are assigned with different functionalities while acting as cluster head, cluster gateway or cluster member as shown in the figure 1.4.

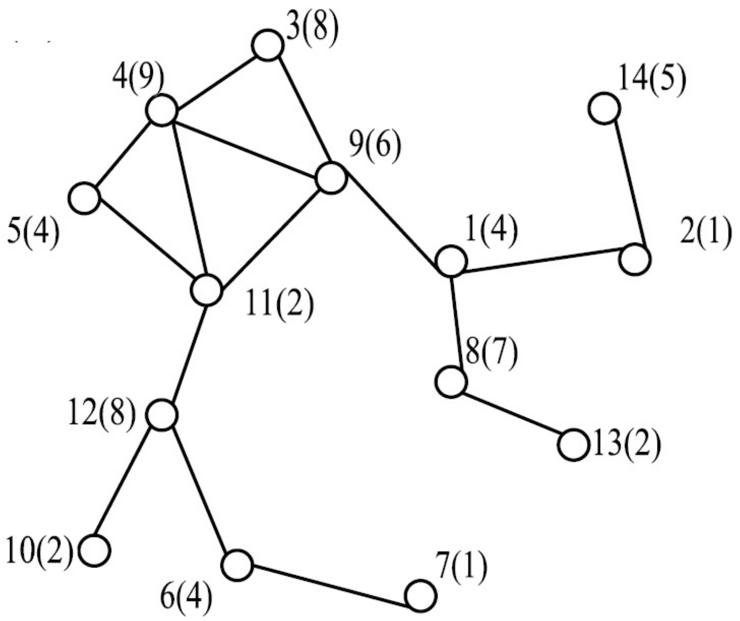


Figure 1.3: Nodes in Flat Structure

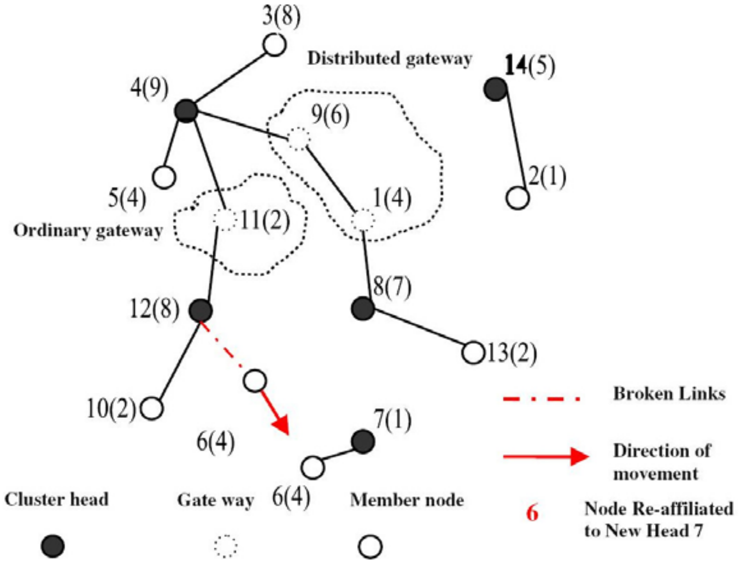


Figure 1.4: Nodes in Hierarchical Structure

1.4.2 Cluster Requirement

Qualitative information flow among various nodes is the prime goal of any communication network. This type of information includes

- The control information that is exchanged between the source and the destination prior to the actual data transmission. (i.e. Polling)

- The data information that is routed by various routers to reach the destination.

The overhead of information passing increases considerably when the routing nodes are mobile and the network topology changes frequently. This causes instability in the pre-established routes. By the cluster control structure, the topology updating information and routing information can be reduced to the exchange of aggregated information between various clusters and the exchange of detailed topology information to single clusters [13]. This concept localises the traffic of updating and control messages leading to efficient bandwidth utilisation in the ad hoc network.

In cellular network, the mobile nodes communicate directly with the fixed base station reducing the wireless part of communication to a single hop problem. A number of solutions have been proposed for handling of mobility of the nodes by this base station. Thus, the mapping of cellular architecture into peer-to-peer network leads to the concept of clustering [14]. In such a virtual cellular architecture (VCA), cluster heads are selected to play the role of base stations of the cellular structure. The cluster head along with its one-hop members form the virtual cells retaining the merits of cellular architecture. Scalability and bandwidth utilisation in an ad hoc network have considerable effect on routing efficiency. Keeping track of neighbors in a dynamic network for efficient routing, further increases the storage cost in a flat structure. The spatial reuse of major resource like bandwidth is also reduced in case of a scalable network. Clustering is a proven solution for such situations. The cluster heads being responsible for storing and forwarding the packets need to keep the information regarding their neighbor heads or gateways within their close proximity. This enables to increase the scalability of the network. The bandwidth utilisation is also increased as limited nodes take part in transmitting the packets. Energy issue is still an unsolved problem for wireless nodes

[15]. Specially, in a wireless sensor network limited battery power is a major constraint. The cluster control structure provides solution for efficient energy management [16]. Here, limited nodes are selected as communication centres for forwarding data of other nodes. At that time, the remaining nodes can save their battery power as they don't take part in packet forwarding.

1.4.3 Phases of Clustering

The process of clustering can be visualised as a combination of two phases, i.e., **cluster formation** and **cluster maintenance**. The cluster formation phase deals with the logical partition of the mobile nodes into several groups and selection of a set of suitable nodes to act as heads in every group. In mobile ad hoc network, where the topology changes frequently, selection of optimum number of cluster heads is a NP-hard problem [17]. There exists some representative algorithms that use the parameters like node identity number, mobility, battery power, degree of connectivity etc. as the factors to decide its suitability for cluster head [18]. Even some researchers combine multiple node parameters to select these set of routers in an efficient manner. These selected nodes are responsible for routing as well as node management in the mobile network and collectively called as the dominant set in graph theory terminology [19].

The objective of cluster maintenance is to preserve the existing clustering structure as much as possible. In one hop clustering, since every node is directly connected to a cluster head, the mobility of either the member node or the cluster head may drive them away from each other. There exists a bidirectional link between these two nodes till both of them are within their transmission range. When any of them moves away from the other, there occurs a link failure and the member node searches for another new head within its transmission range to get affiliated. This kind of situation is called as re-affiliation to a new head node. A typical example is shown for node 6 in figure 1.3. Here, the node 6 moves (shown in the arrow mark) away from its current cluster head 12. When it is out of the transmission range of head 12, it finds the new head node 7 within its transmission range and gets itself affiliated to node 7 and becomes its new member. Thus the head nodes 12 and 7 update their member lists accordingly. A single re-affiliation causes several update messages to flow between both the old and new cluster heads.

The requirement for the reelection of cluster heads arises when the current heads fail to cover all the nodes in the network. Sometimes a node may move away from the transmission range of all the current cluster heads and becomes an orphan node. This demands a reelection of cluster heads. Even at times any of the cluster heads may drain out of energy or may even fail to work due to any fault occurrence and needs a head reelection process. However, such an unavoidable reelection increases the computation cost and the message complexity.

1.5 Contributions

In order to proceed with the process of clustering in MANET, the thesis starts with the detail investigations of the existing algorithms in chapter 2. Simulation survey has been carried out for all the algorithms with respect to their cluster setup and cluster maintenance. A comparison of the results are also made to study the flaws and strengths of the algorithm. [dissemination 1]

Proposal of a new clustering algorithm to meet the challanges studied in chapter 2 starts with the design of a Neighbour Detection Protocol (NDP) in chapter 3. This protocol enables the nodes to keep track of the one hop neighbours in the network. The proposed protocol is validated by using the Coloured Petri Nets (CPN) to ensure its success in handling the data and control flow.[dissemination 3]

The next contribution of the thesis is the proposal of a Topology Adaptive Clustering Algorithm (TACA) in chapter 4. This algorithm maintains the linked cluster architecture by carefully designing the setup and maintenance phase.Two types of cluster heads are proposed to enhance the network life time and cluster stability.[dissemination 4,5,6,7]

The subsequent contribution of the thesis is the design of a topology Control protocol TRAP in chapter 5. This protocol aims to reduce the cluster maintenance overhead by allowing the nodes to adjust their transmission ranges as and when required.[dissemination 2]

1.6 Organisation of the thesis

The rest of the thesis is organised as follows:

Chapter 2 makes a thorough review of the existing clustering algorithms for MANET. A complete simulation survey presents the strengths and weaknesses of the existing algorithms. The simulation results are discussed with respect to various cluster maintenance overheads. The survey motivates to move in the direction of proposing new energy efficient clustering algorithms for longer network life time and reduced maintenance overhead.

Chapter 3 discusses a well known modelling tool, the Coloured Petri Nets (CPNs). This tool has been used to validate some of our proposed protocols and algorithms. To start with, a neighbor detection protocol (NDP) has been proposed that probes the one-hop neighbors of the nodes in the network. The protocol has been validated through simulation by using CPN tools. The flow of the data and control throughout the simulation is discussed in this chapter.

Chapter 4 discusses the proposed topology adaptive clustering algorithm (TACA) that uses the NDP for neighbor detection. This algorithm has two folds. First, it selects a set of volunteer cluster heads by considering the node mobility and the available battery power as the selecting parameters. Eventually, as a cluster head drains its battery power below a threshold value it invites one of its members having the maximum battery power and minimum mobility to take up the role of the cluster head. This head is called as the non-volunteer cluster head. The process of selecting the cluster heads among their one-hop neighbors is validated through the CPN simulation. The simulation ensures that the flow of the control and data follows the requirements of the algorithm. The performance results of the proposed algorithm are analysed and compared with existing competent algorithms.

Chapter 5 presents the proposed transmission range adjustment protocol that allows the nodes to increase or decrease their transmission range so that they stay connected to the cluster heads in spite of making frequent re-affiliations to the other heads. The frequency of cluster head selections are reduced by the application of this protocol. The number of cluster heads forming the backbone of communication is also reduced by this range adjustment protocol. The performance results are shown and discussed at the end of the chapter.

Chapter 6 concludes the thesis. In this chapter, the work done is summarised, the contributions are highlighted and suggestion for the future work has been discussed.

Chapter 2

Literature Survey on Clustering

2.1 Introduction

The concept of partitioning the dynamic network into logical clusters was initially proposed by Baker et. al. [20]. They have designed a self-starting, distributed algorithm to establish and maintain a connected architecture even with the node mobility and node failure. This algorithm is best suited for High Frequency Intra Task Force (HF ITF) communication network along with other communication networks such as packet radio network (PRNET), advanced mobile phone service (AMPS) network and battle field information distributed system (BID). The HF ITF network is a mobile and widely dispersed general purpose network that provides extended line of sight (ELOS:50-1000km) communication for naval task force units. Here the nodes are linked via radio waves from the HF band (2-30MHz). The variation in antenna pattern of the nodes, noise level, propagation losses and ground wave / sky wave interference change the connectivity among the nodes. This connectivity is further hampered by the node movement, node failure and new node addition into the network. So there is a need for a self organizing reliable network structure that can be maintained under changing connectivity without the support of a central controller. Due to the wide disperse of the nodes, the classical hidden terminal phenomena of wireless communication increases the radio channel transmission conflict. Thus, the authors of [20] proposed a new architecture called the linked cluster architecture where the network is organized into a set of node clusters and each node belongs to at least one cluster. Cluster heads of every cluster remain in direct communication range of member nodes resolving the hidden terminal problem by the existing busy tone multiple access (BTMA) technique [21].

The major five tasks associated with the implementation of linked cluster architecture [22] are **topology sensing**, **cluster formation**, **cluster linkage**, **link activation** and finally **routing**. To meet the above tasks without relying on a central coordinator, following three algorithms were developed:

- the **Linked Cluster Algorithm** (LCA)
- the **Link Activation Algorithm** (LAA)
- the **Routing Algorithm**

2.2 Linked Cluster Algorithm (LCA)

The linked cluster algorithm (LCA) performs the job of initial three tasks such as topology sensing, cluster formation and cluster linkage. Where as the link activation algorithm (LAA) performs the job of link activation between the nodes in the network. The routing algorithm covers the details of the routing operations for packet communication. The objective of the current work is to focus on the basis of neighborhood detection in changing topology and cluster formation.

Topology sensing is the initial step in implementing the linked cluster architecture where the nodes discover their neighbors by the method of probing as described in [22]. Figure 2.1 shows an example of linked cluster architecture for a fixed communication range of the nodes in a particular HF sub-band. Here the cluster heads are connected directly or via the gateway nodes. Due to the

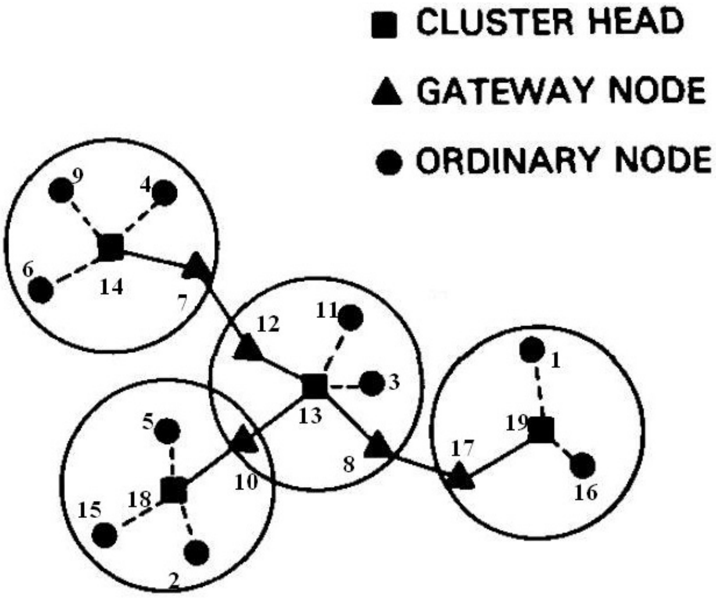


Figure 2.1: Linked Cluster Architecture for one sub-band [22].

variation in the HF communication range with frequency, separate connectivity maps are formed based on the communication range and connectivity within that sub-band. For every sub-band LCA works in two TDMA frames where every frame is subdivided into N slots as shown in figure 2.2. Each node is identified by a unique integer from 1 to N and is allocated to transmit control channel messages related to the algorithm during its own slot in each frame.

During the i^{th} slot of the 1^{st} Frame, node i broadcasts its probe message (for example, announcing its own identity) and acknowledges the receipt of previously transmitted probe messages that it has heard (by announcing the IDs of those nodes it has heard from so far) during the earlier slots of this frame. A binary CONNECTIVITY matrix stores a value of 1 in the *(i,j)* position to indicate the existence of a link between i and j where as a 0 indicates a lack of connectivity. Thus, at the end of this frame, node i fills in some of the entries of its CONNECTIVITY matrix.

During 2^{nd} Frame each node broadcasts acknowledgements for probe messages that it has received since the transmission of its own slot of 1^{st} frame.

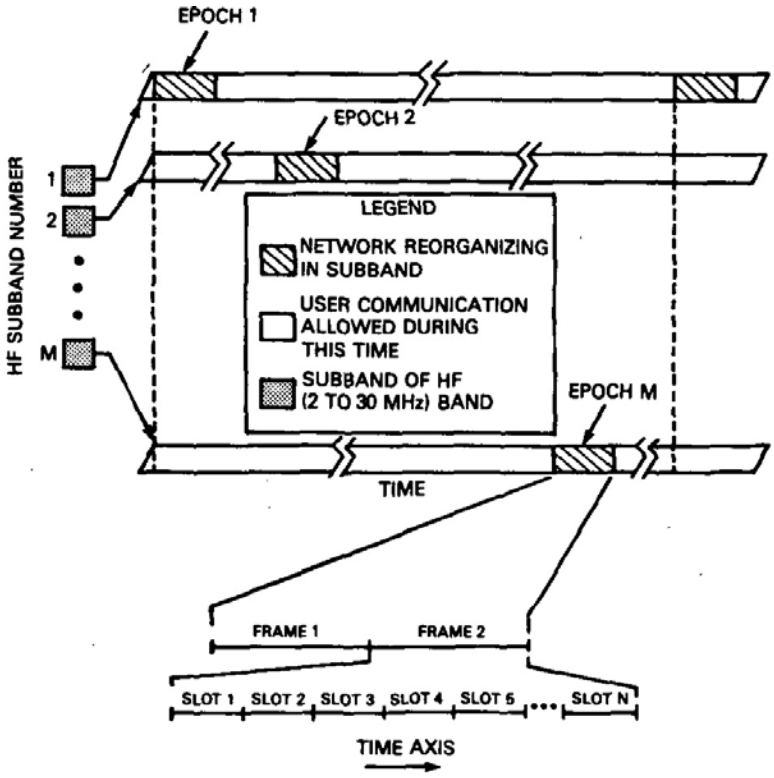


Figure 2.2: Timing of network reorganization. Each node is assigned to one of N transmission slots [22].

Thus, by the time of i^{th} slot of 2^{nd} frame transmission, node i fills in the i^{th} row of the connectivity matrix and has the complete knowledge of its neighbors.

During the **cluster formation** phase of LCA, it uses the rule that the node with the highest identity number among its neighbors is the first candidate to claim cluster head status as in figure 2.1. A node i first checks its connectivity row in the matrix and if no neighbor with higher identity number is found, it becomes the cluster head. If one or more neighbors exist with higher identity number, the highest numbered neighbor becomes the cluster head. Also, the node i checks whether it is the highest neighbor of other node j . Thus, at the end of the 2^{nd} frame the nodes know their NODESTATUS and broadcast it for the implementation of the next task of link cluster architecture, i.e. linking of clusters by LAA.

The LCA uses a very simple cluster head selection strategy for its implementation. However, this TDMA approach suffers from the fact that every node needs to maintain an accurate global time among them which is possible only by the presence of a central timer. Moreover, partitioning the frames into number of slots is typically not possible as the node numbers in a dynamic network can not be known a priori. The identity of the nodes being the deciding factor for the cluster heads, the method of number assignment to the nodes become very crucial part of the proposed organisation. The higher numbered nodes have a greater tendency to become heads unless it is completely covered by another cluster head. Further, the situation of node and/or link addition and deletion in the clusters (i.e. the cluster maintenance) is not being considered by the authors. In a nutshell, LCA could not meet certain criteria of the ad hoc network, but could become the base algorithm for other benchmark algorithms.

Once the clusters are formed in the network, it is very much required to maintain the cluster structure in the network. The node mobility frequently change the network topology as well as the node connectivity. A proper cluster maintenance strategy would be required, so that the LCA is retained. The following parameters may be considered with regard to the cluster maintenance:

- **Node Re-affiliation:** In the cluster formation phase of the LCA, cluster heads are selected for every cluster. The one-hop neighbors of the head become the cluster members. A non-head node having connectivity to multiple cluster heads can select only one to get affiliated as its member. By the virtue of mobility, either the cluster head or the cluster member may go out of range of each other breaking the link between them. At this time, the non-head node try to connect with another head in its proximity. This process is defined as the node re-affiliation [23]. Node re-affiliation is very much required to retain the node connectivity in the network. However, frequent node re-affiliation increases the number of messages to be exchanged by the nodes consuming the communication bandwidth. It also degrades the cluster stability and routing stability by changing the path very often. Thus, to reduce the cluster maintenance overhead, the frequency of node re-affiliation should be as less as possible.

- **Node Re-election:** In the dynamic topology of MANET, the change in link between the nodes occur very frequently. After the initial cluster formation phase, any of the existing nodes may move to an isolated area, where it may not establish the connectivity to other nodes. In such a situation, the node has to act as an isolated head without having any cluster members. This changes the set of cluster members, or in the graph theory terminology the members of the dominant set is changed. Further, the light weight mobile devices are battery operated, which is a major constraint. An existing head may consume its battery power completely at any time and may become dead. This requires another head to get selected in the network resulting a change in the dominant set. So it is understood that, cluster formation phase must be followed by a maintenance phase so that as much as of the clustering structure is retained. However, a single cluster head re-election leads to several node re-affiliations resulting higher communication and computation overhead due to message exchanges.

- **Average No. of Cluster Heads:** This parameter decides the length of communication backbone. In the LCA, communication of packets take place through the virtual backbone consisting of cluster heads. To reduce the end-to-end delay in communication, the number of hops in the virtual back bone should be as less as possible [24].

All the above cluster maintenance parameters are the overheads, that need to be minimised. The following section focuses on the cluster formation phase of the linked cluster architecture while explaining the existing clustering algorithms with their basis of cluster formation as well as the cluster maintenance in the presence of node mobility.

2.3 Lowest ID Algorithm (LID)

A small variation to LCA [20] was proposed by Ephremides et. al.in [25] and it was named as the Lowest ID (LID) algorithm. In this algorithm, every node is assigned with a unique non-negative identification number which is the deciding factor for the status of a node. In a mobile packet radio network, a node has no a priori knowledge of the locations of other nodes as well as the connectivity of the network. So, as a first task when the network comes up, the connectivity among the nodes is discovered by every other node. This is accomplished by every single node that broadcasts its own *ID* to its neighbors. At the same time it also receives the same from its neighbors. (The term neighbor defines the set of nodes that can hear the messages directly.) If a node listens to all the *ID*s that are higher than its own *ID* , then it declares itself as the cluster head among its immediate neighbors. And the neighbor nodes whose status is not yet decided become the members of the newly selected head. This process is repeated till all the nodes are assigned with the role of a head or a member of a cluster. This algorithm provides a solution to avoid the hidden terminal problem of wireless communication as well as how to make a quick recovery from the node or link failure in the dynamic network. This algorithm does not allow two cluster heads to be neighbors at any point of time. So the gateway nodes provide the path for inter cluster communication. TDMA based probing message exchange is used here to ensure a contention free communication. It does not target to achieve minimum number of clusters, but ensures to produce a connected network.

LID retains its utility as a benchmark, for producing reasonably stable cluster control architecture as discussed by Gerla et. al. in [26] where the stability of clusters is much better than that of the contemporary algorithms [8]. However, as node *ID* is the only deciding factor for it to be a cluster head, the lower *ID* nodes are biased to become the heads all the time resulting in faster energy drainage of those nodes which in turn perturbs the cluster stability.

2.3.1 Result Discussion of LID

The performance results for Lowest ID (LID) algorithm are shown in this section. Figure 2.3 depicts the average number of clusters formed with the variation of the transmission range. Here, the number of nodes N is considered as 30, 40, and 50. The population of the cluster heads is shown in figure 2.3 whose value reduces with the increase in transmission range. As discussed earlier, the LID algorithm does not allow two cluster heads to remain as neighbors of each other. So,one node is forced to resign from its current status of cluster head and joins as the member of the other. This increases the frequency of cluster head updations (i.e. reelection) as well as the frequency of re-affiliation by the member nodes as shown in figure 2.4 and figure 2.5 respectively.

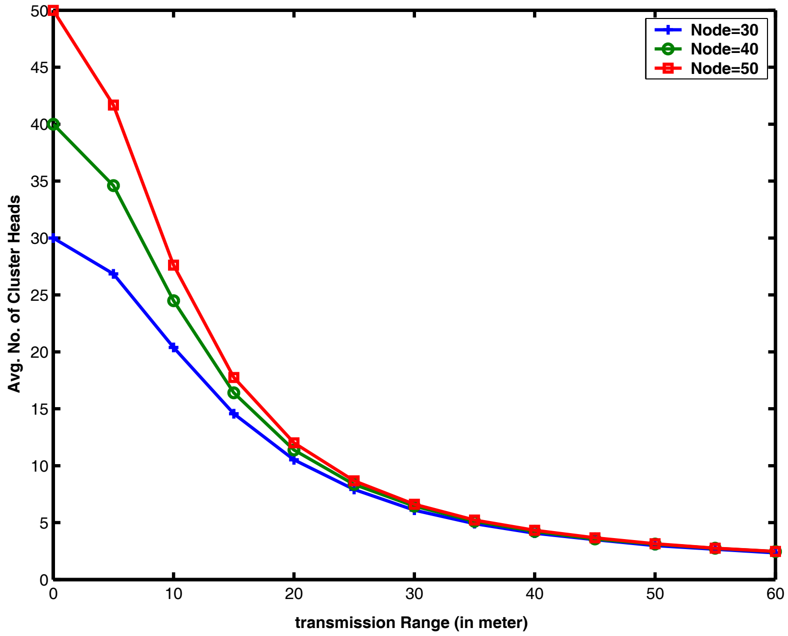


Figure 2.3: Avg. No. of Cluster Heads for LID.

There occurs considerable rise in the frequency of reaffiliation for the lower transmission ranges where as it decreases slowly with increase in transmission

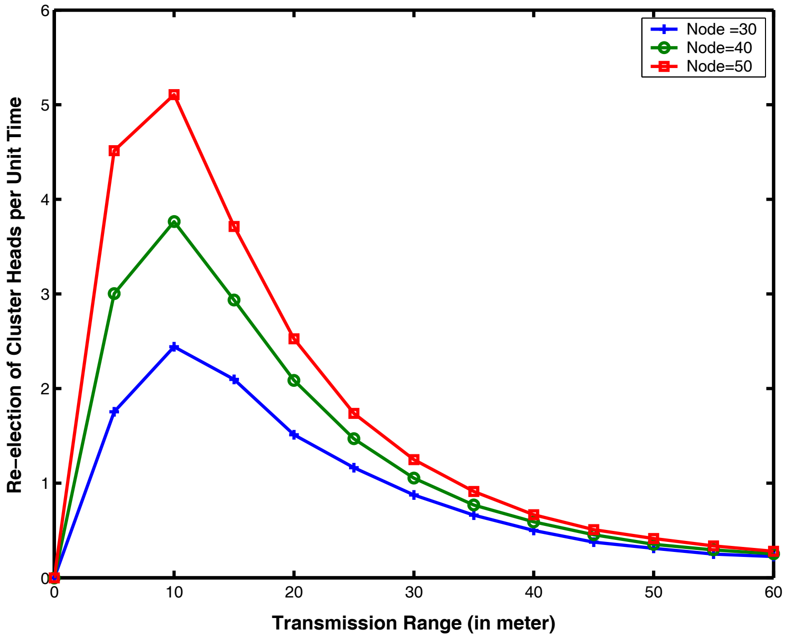


Figure 2.4: Frequency of Cluster Head Re-elections for LID.

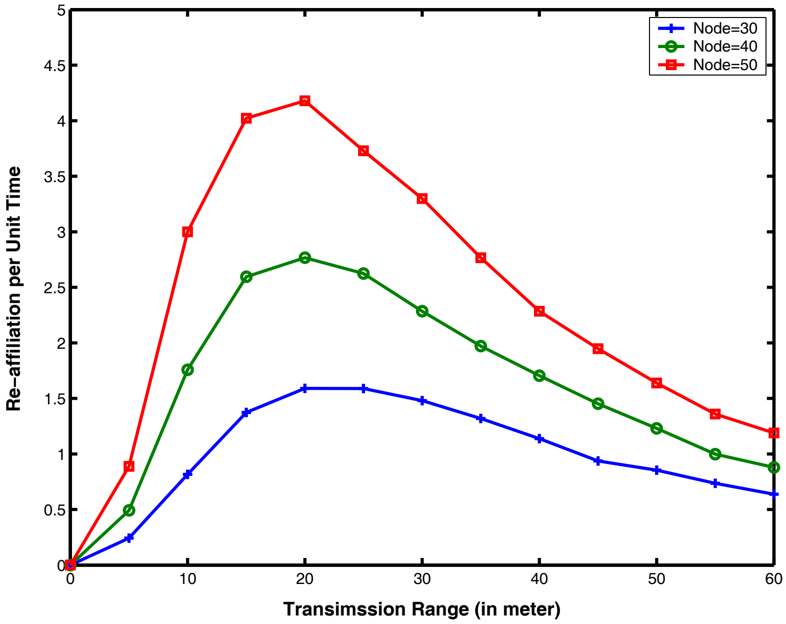


Figure 2.5: Frequency of node re-affiliations for LID.

range. This happens because the lower transmission range reduces the size of the cluster zone keeping most of the nodes at the boundary, so that even for a slight movement they leave the current cluster boundary and join other clusters as their new members. With the increase in the range, the cluster head periphery is increased to accommodate more members within it.

The frequency of cluster head reelections that takes place at lower transmission range is comparatively more than that at high transmission range as in figure 2.4. The reason is, at lower transmission range, the number of cluster heads is high and the probability that two cluster heads will come closer to each other is quite high. Thus a considerable amount of node head resignation (due to the non-neighborhood constraint) occurs resulting the number of reelections (or updations) of the heads to take place.

2.4 Highest Connectivity Algorithm (HC)

Highest Connectivity Algorithm that was proposed by Parekh in [27] is a variant of LCA. This algorithm aims to reduce the number of clusters in the network. In every cluster there exists a cluster head that belongs to the dominating set. If N_i represents the set of adjacent nodes of a particular node i , then the degree of connectivity of i is represented as $D_i = |N_i|$, where $|N_i|$ is the cardinality of i . In the HC algorithm, a node having highest degree of connectivity is selected as the cluster head. And the adjacent node whose status is not yet decided becomes the member of the selected cluster head. A higher degree of connectivity ensures efficient service to the member nodes by minimizing the number of heads. Here the efficiency means lowering the delay in communication through the head nodes.

However, this algorithm results in low throughput. This happens because every cluster is assigned with some resources that are shared among all the nodes in the cluster in a near equal manner. So, an increased number of nodes in a cluster reduce the throughput and finally the system performance is degraded. Moreover, the mobility of nodes changes the degree of connectivity of the node very frequently leading to more number of cluster head re-elections as well as link updations. Thus, the maintenance cost of the clusters is worse than the configuration counterpart. The poor cluster stability of the algorithm decreases its application in real world situation in spite of its reduced delay in packet communication.

2.4.1 Result Discussion of HC

The performance result for the highest connectivity algorithm (HC) has been depicted in this section where figure 2.6 shows that the number of clusters in the network reduces as the transmission range of the nodes increases. This happens because the higher transmission range enhances the degree of connectivity of the nodes and more number of nodes can emerge within a single cluster. Being connectivity based algorithm, this algorithm suffers from higher cluster head re-election rate and frequency of node re-affiliations to the cluster heads. It is true because the mobility of the nodes changes their degree of connectivity, so that for a single connectivity change of the node, the weight of the head may

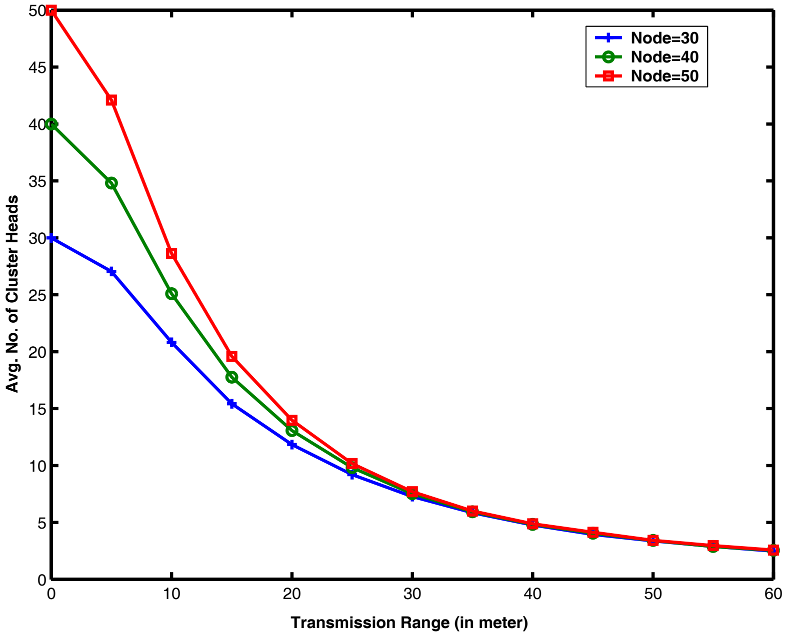


Figure 2.6: Avg. No. of Cluster Heads for HC

change enforcing re-election and subsequent re-affiliations to occur. It could be observed from figure 2.7 and figure 2.8 respectively.

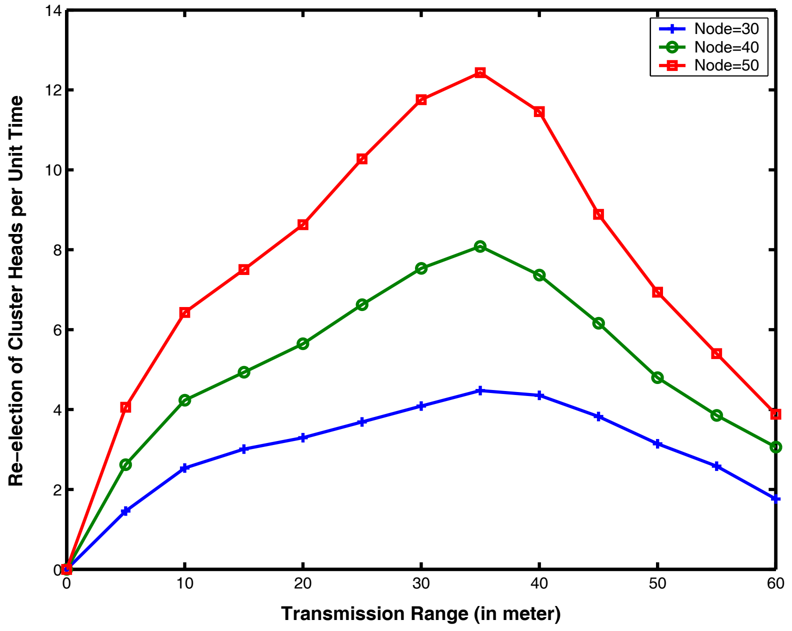


Figure 2.7: Frequency of Cluster Head Re-elections for HC.

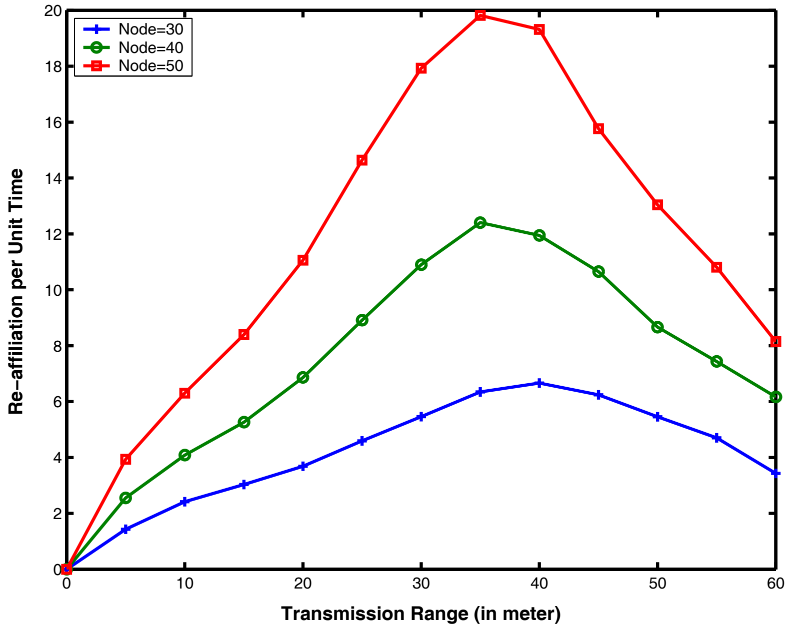


Figure 2.8: Frequency of node re-affiliations for HC.

2.5 Mobility Metric Based Algorithm (MOBIC)

A mobility metric based version of lowest ID algorithm MOBIC was proposed by Basu et al. in [28]. The algorithm uses mobility based metric as cluster formation basic and calculation of weights of the nodes in the network. To compute the relative motion of a node with respect to its neighbors, the authors have proposed to use the ratio of two consecutive signal strengths received by a node. Thus the relative mobility metric which is denoted as:

$$M_Y^{rel}(X) = 10 \log_{10} \frac{R_x P_r^{new} X \to Y}{R_x P_r^{old} X \to Y} \tag{2.1}$$

at a node Y with respect to X, gives either a positive or negative value depending on the value of the numerator. Here $R_x P_r$ is the received signal power received from node X. When $(R_x P_r^{new} X \to Y) > (R_x P_r^{old} X \to Y)$ the result gives a positive value indicating that both the nodes are approaching each other. Similarly, when $(R_x P_r^{new} X \to Y) < (R_x P_r^{old} X \to Y)$ the logarithm of the ratio gives a negative value indicating that the nodes are moving away from each other. Thus, a node having N number of neighbors will have N such values of M_Y^{rel} . The aggregate local mobility M_Y of a node Y is calculated by taking the variance of the entire set of relative mobilities. That is, $M_Y = var(M_Y^{rel}(X_1), M_Y^{rel}(X_2), ..., M_Y^{rel}(X_l))$ where the variance is taken with respect to 0. The motivation behind calculating the variance of relative mobility metric with respect to each neighbor is that, a lower value of M_Y indicates Y to be less mobile with respect to its neighbors. Hence, choosing a relatively low mobile node to act as a cluster head yields a better cluster stability. This heuristic is most applicable in a group mobility model where the nodes move in a group and is easy to find the relative mobility of a node with respect to its neighbors. But the situation where nodes move independently, it is difficult to find the relative mobility metric for every individual node. The relative mobility needs two consecutive "hello" messages to be transmitted. This further degrades the MAC efficiency by increasing the message exchanges.

Once the relative mobility metric for every node is decided, MOBIC is called upon the nodes. MOBIC works almost same as the Lowest ID algorithm, where the node IDs are replaced by the relative mobility metrics of each node. A node with the lowest value of M_Y amongst its neighbors becomes the cluster head; else it becomes a cluster member. When two cluster heads accidentally come within their transmission range, re-clustering is deferred for Cluster-Contention-Interval (CCI) period [8]. If they remain within the range even after the CCI period, then re-clustering is invoked and the node with higher mobility metric resigns from its present status. In MOBIC the need of collecting the relative speed information from the neighbors degrades its performance, because continuous movement of the nodes in MANET may provide inaccurate mobility information during cluster set up time.

2.5.1 Result Discussion of MOBIC

The result of the MOBIC algorithm is depicted in this section. The result for the average number of cluster heads is shown in figure 2.9. This is almost same as that of the LID algorithm except for the rate of decay in values which is slower in comparison to that of the former one. The reason is that, the two neighbor cluster heads are allowed to retain their role for a Cluster Contention Interval (CCI) period. As the reelection is deferred till the expiry of a CCI time period, the frequency of re-election of cluster heads is reduced to a great extent as in figure 2.10. For the number of re-affiliations by the nodes to the cluster heads as in figure 2.11, it is quite high like HC algorithm. And it converges to that of LID only when the transmission range is very high [28].

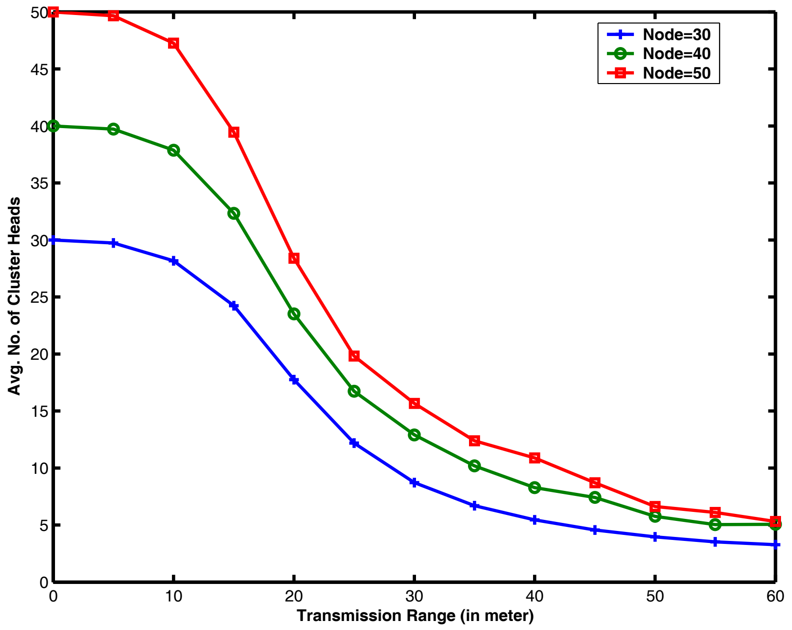


Figure 2.9: Avg. No. of Cluster Heads for MOBIC

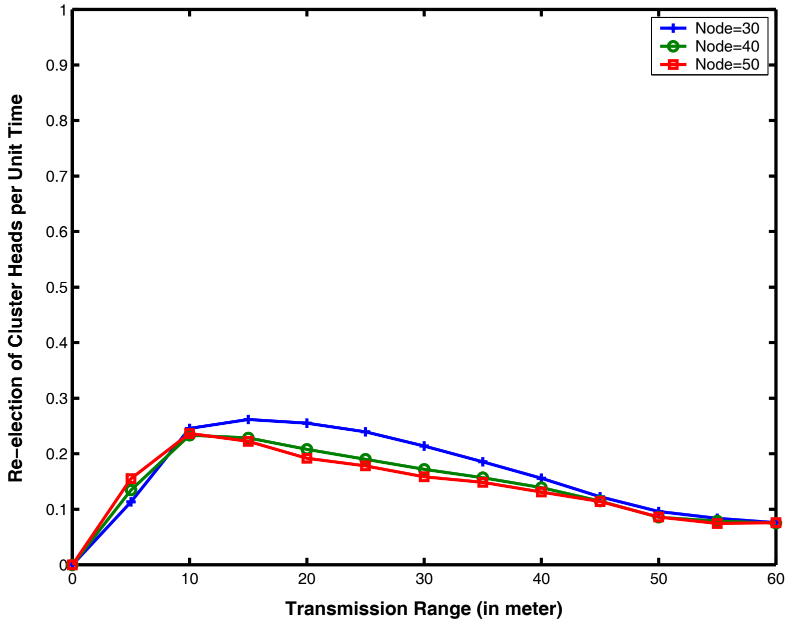


Figure 2.10: Frequency of Cluster Head Re-elections for MOBIC

2.6 Distributed Mobility Adaptive Algorithm (DCA, DMAC)

Basagni et. al. presented an extension of [14] in [29] as a distributed clustering algorithm (DCA) that is mobility adaptive and truly distributed in nature.

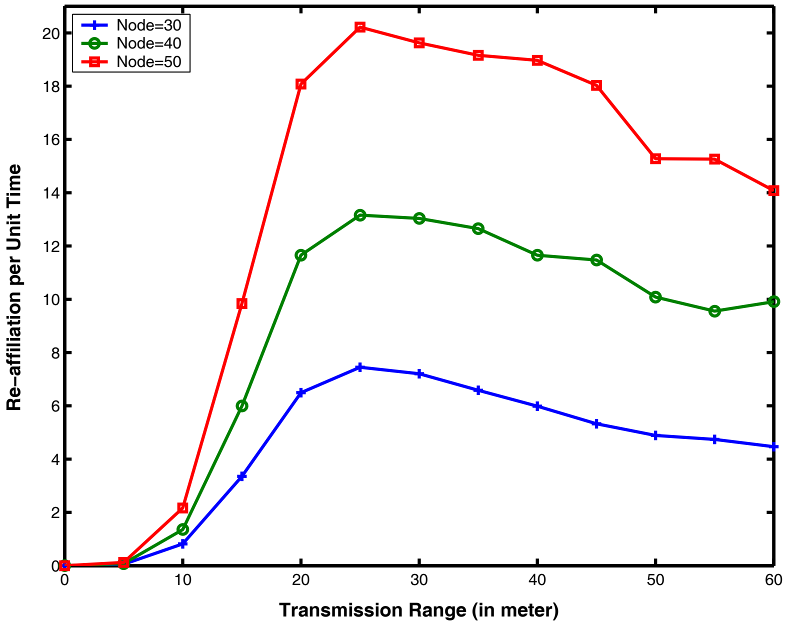


Figure 2.11: Frequency of node re-affiliations for MOBIC

This algorithm is a generic weight based cluster formation algorithm. Here each node is associated with a parameter called the weight (i.e. ≥ 0) that decides the role of a node. The weight of a node may be the function of its transmission range or node mobility.

DCA does not allow the change in network topology during the execution of the algorithm. A node having bigger weight among all its one-hop neighbors is selected as the cluster head (ties are broken by using LID). An ordinary node opts to join a cluster head with the biggest weight when it comes across several other heads in its proximity. This algorithm explains well for the cluster formation where as the maintenance of the clusters in the presence of node mobility is not specified by the authors. DCA is mostly applicable for a static or a quasi static network.

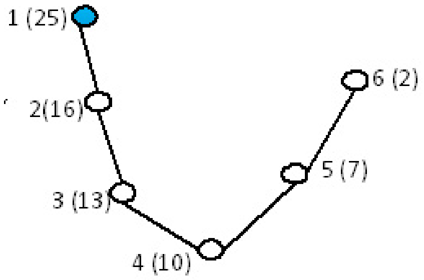
In the distributed and mobility adaptive clustering algorithm DMAC [30] proposed by the same authors, the cluster formation process is almost same as that of DCA. However, the non-mobility of nodes during the execution of the algorithm is eliminated here, making it truly mobility adaptive. DMAC claims to be the most suitable algorithm for the cluster formation and maintenance in the presence of node mobility. It starts with the assumption that every node knows its own *ID*, *weight* and status in the network as well as the same for its one-hop neighbors. This proves that the cluster head is selected only with the knowledge of its local topology.

The major weakness of this algorithm lies with the lower weighted nodes. A lower weighted node decides its role only after all the 1-hop neighbor nodes with higher weight have decided their role. The worst case scenario occurs when the network topology contains a chain of nodes whose weights are in sorted order as shown in figure 2.6(a,b,c). Here, after the first iteration of the algorithm, only node 1 is able to decide its role where as that of the other nodes remain undecided. Node 3 can decide its role only when node 2 joins to 1 as a member node. Hence, node 6 has to wait for a long way to decide its own role as a cluster head or a member node.

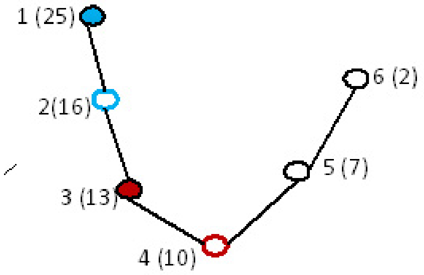
Further, Vilzmann et. al. explains the chain activity in DMAC, which is the major drawback of the algorithm, when a higher weighted node is added to the network or a link failure occurs between a cluster head and one of its member nodes [31].

2.6.1 Result Discussion of DMAC

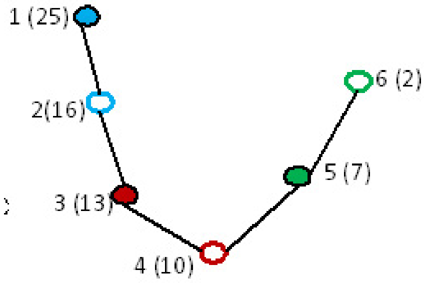
The simulation result for DMAC is discussed in this section. The results for average number of cluster heads are shown in figure 2.13. It can be clearly observed that the result for the average number of cluster heads in the network remains almost same to that of figure 2.3 and figure 2.6. The reason is that the lowest ID algorithm chooses the node with lowest identification as the cluster head and the highest connectivity algorithm chooses a node with higher connectivity as the cluster head. When the network scenario remains same for the simulation of both the algorithms, the number of cluster heads selected for both of them remains same. Similarly, the authors of DMAC algorithm have



(a) Node 1 decides its role



(b) Node 3 decides its role after 2 has become member of 1



(c) Node 5 decides its role after 4 has become member of 3

Figure 2.12: Worst case scenario of DMAC algorithm to decide the node status

not specified the criteria for the cluster head selection procedure. So the current simulation considers the lower ID method of cluster head selection. However, the maintenance strategy is completely different for all three algorithms which is clearly visible in the figures for the frequency of re-affiliation and frequency of re-election. The following two criteria of DMAC increase the re-affiliation and reelections per unit time:

- No two cluster heads can be neighbors of each other, i.e. whenever two cluster heads come within their range of transmission, the head with the

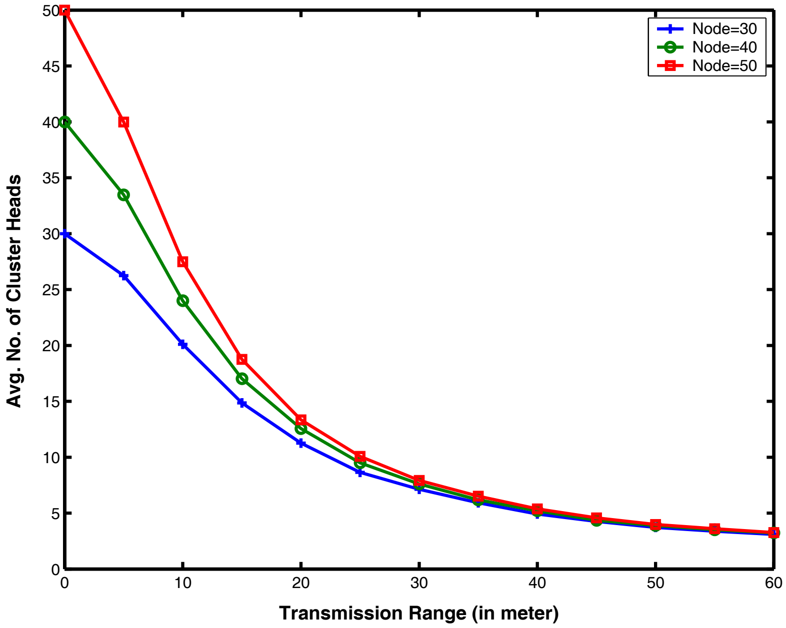


Figure 2.13: Avg. No. of Cluster Heads for DMAC

lower weight value has to resign from its current role and join the head with the higher weight. Thus, there occurs a reelection and several re-affiliations by the members of the previous head by a chain activity as discussed earlier.

- When a member node comes within the transmission range of another head node whose weight is more than its current head, then it withdraws its affiliation from the current head and joins the new head resulting in further re-affiliation (i.e. every node tries to be connected with the most suitable node in its proximity).

The result of the frequency of re-election of cluster heads and re-affiliations by the nodes are shown in figure 2.14 and figure 2.15 respectively.

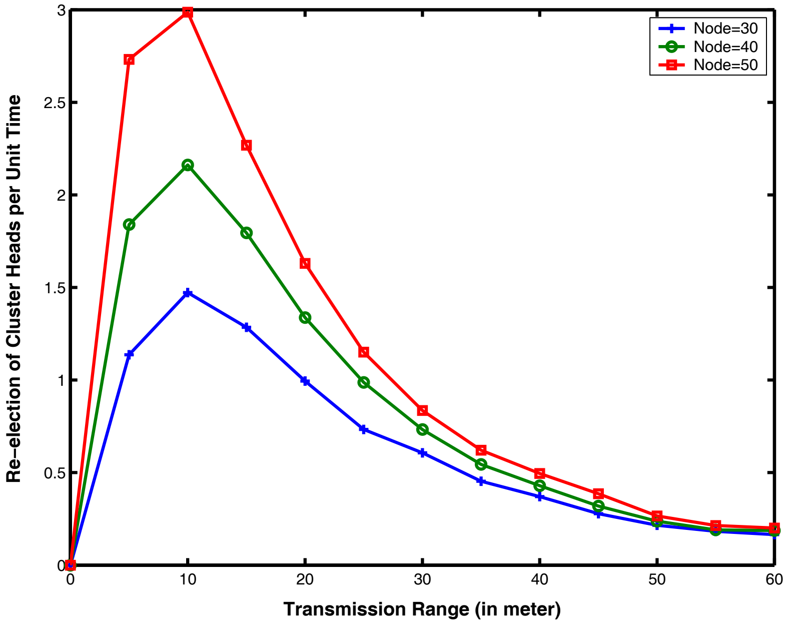


Figure 2.14: Frequency of Cluster Head Re-elections for DMAC

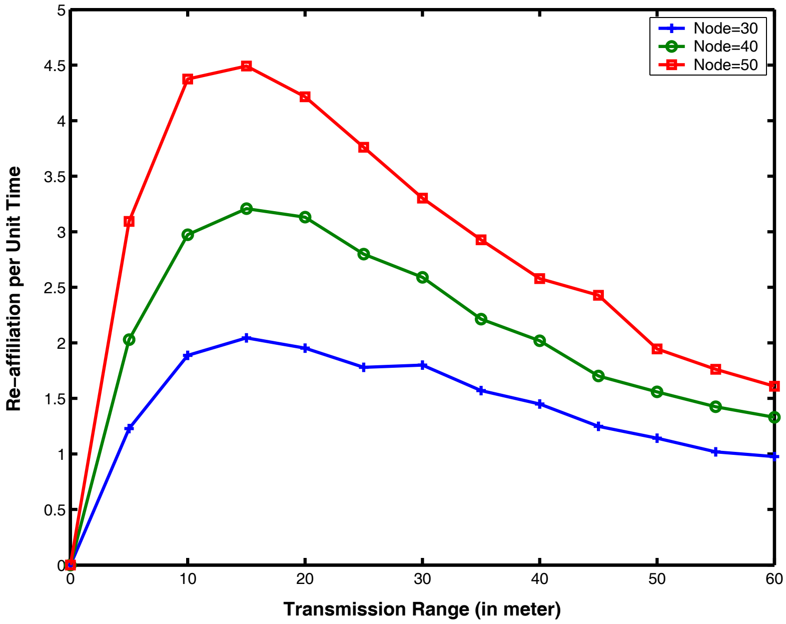


Figure 2.15: Frequency of node re-affiliations for DMAC

2.7 Weighted Clustering Algorithm (WCA)

Das et. al. have proposed a weighted clustering algorithm WCA in [23,32,33] where the re-election takes place with the occurrence of certain events i.e., when there is a demand for it. Node parameters like degree of connectivity, mobility, transmission power and available battery power are considered for selection of a cluster head and are given different weights depending on the network scenario. For example, sensor networks where energy is a major constraint, battery power could be given higher weight. The combined weight of every node is calculated as $W_c = w_1 D_v + w_2 M_v + w_3 T_v + w_4 P_v$, where D_v is the difference of the cardinality of the neighbors of a node v represented as $|N_v|$ and a scenario based threshold δ which limits the upper bound for the number of members in a cluster, i.e. $D_v = ||N_v| - \delta|$. M_v is the average speed of a node since the last reelection taken place and $T_v = \sum dist(v, \bar{v})$, where $\bar{v}$ is the set of neighbor nodes of v . P_v is the cumulative time for which the node remains as the cluster head. All these parameters are normalised to a predetermined value and the weighing factors are chosen so that

$$\sum_{k=1}^{4} W_k = 1. \tag{2.2}$$

Initially, the node having the smallest weight is selected as the initial cluster head and its 1 hop neighbors become the members of the cluster. These covered set of nodes are exempted from taking part in the further selection [17]. This process is repeated till all the nodes are allocated a status of either a head or an ordinary member. Fair distribution of load among the cluster heads are tried to make by restricting the upper limit for number of member nodes within a cluster. Such restriction of member nodes also improves the MAC layer efficiency. The cluster heads work in a dual power mode, i.e. it works in a low power mode for intra cluster communication and in a high power mode for inter cluster communication.

However, the limitation of the algorithm lies in yielding the global minima of weight values in the network. To have a distributed solution of the algorithm, a large number of information are stored and exchanged among the nodes to find the smallest weight. This becomes worse with the increase in network size. To continue with the limitations, the freezing time of mobility

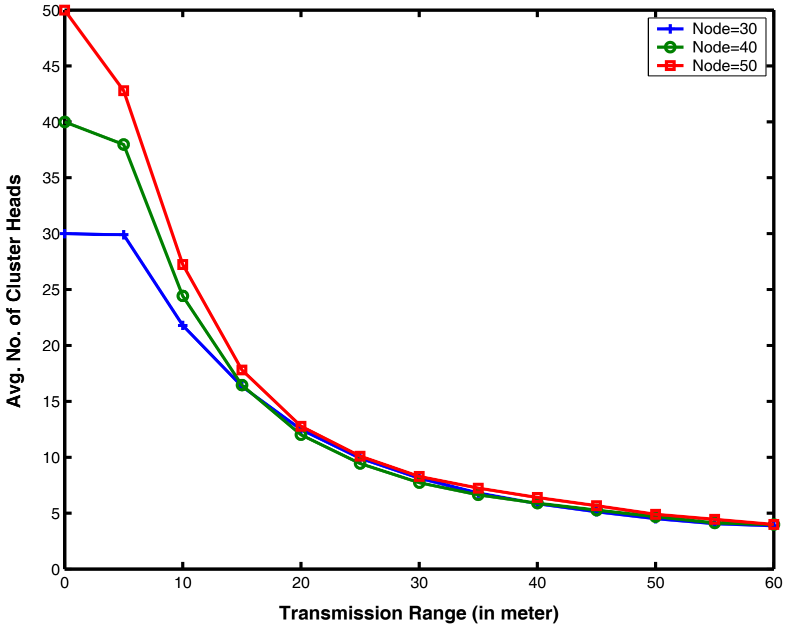


Figure 2.16: Avg. No. of Cluster Heads for WCA

of nodes is high for the cluster setup. This is due to the need for computing so much of information for every node to calculate the combined weight. Whenever a reelection takes place the combined weight of every node needs to be calculated, resulting in further increase of the computation cost. Finally, the major drawback of this algorithm lies with not retaining the property of the lowest weight value to be the cluster head. This happens as WCA does not re-cluster when a member node changes its attaching cluster head [27]. That is, there may be a situation when a low weight node may enter into a cluster whose head is of higher value than this newly entered node.

2.7.1 Result Discussion of WCA

This section discusses about the results of WCA algorithm. The variation in node population does not have any effect on the cluster population for higher transmission ranges as shown in figure 2.16. Moreover, the on-demand clustering algorithm defers the re-election of cluster heads till the current set of cluster heads become unable to cover all the nodes in the network, i.e. the reelection takes place for the following cases:

- Whenever a member node goes out of the transmission range of all the cluster heads
- Whenever a current head goes out of the range of its member nodes leaving them as orphan nodes.

This reduces the frequency of re-elections to great extent. The cluster heads in this algorithm remain as neighbors to enhance the backbone connectivity which in turn reduces the number of re-elections and re-affiliations as shown in figure 2.17 and figure 2.18 respectively. It can be observed that WCA outperforms any of the algorithms in terms of the re-affiliation and re-election rate of the mobile nodes in the dynamic network. WCA uses an upper threshold for the number of members in a cluster. This increases the re-affiliation rate at lower transmission range which is denoted in figure 2.18.

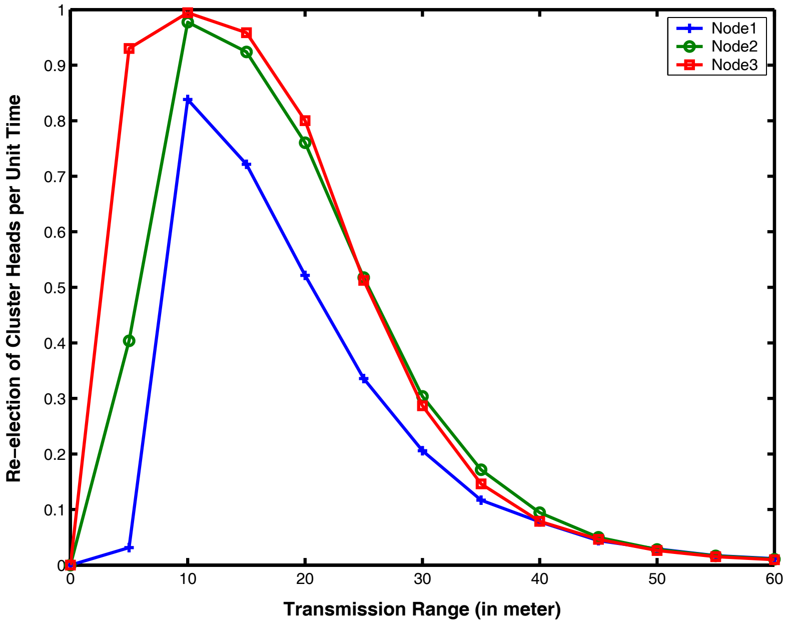


Figure 2.17: Frequency of Cluster Head Re-elections for WCA

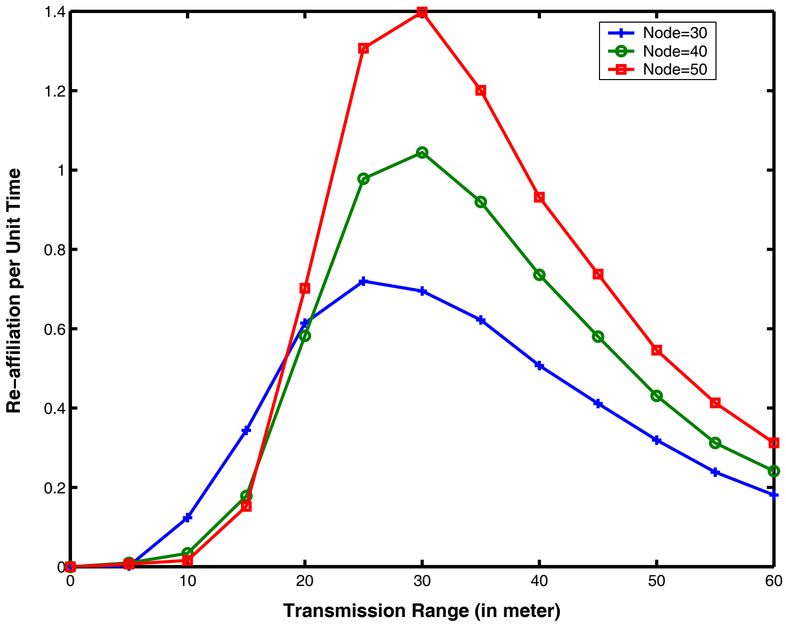


Figure 2.18: Frequency of node re-affiliations for WCA

2.8 Generalised Distributed Mobility Adaptive Algorithm (GDMAC)

A variation of DMAC algorithm is proposed by Ghosh et. al. in [34] known as a Generalised Distributed Mobility Adaptive Clustering (GDMAC) algorithm. The algorithm improves the performance of DMAC by eliminating the restrictions imposed on the nodes and the following changes are incorporated into it:

- In the clustered architecture of the mobile network, the maximum number of heads that are allowed to be neighbors at any time is **K**.
- A member node within a cluster re-affiliates to a newly arrived cluster head only when the weight of the new head exceeds by a threshold amount of **H** over the weight of its current head.

It is clear that the first condition reduces number of reelections that was present in DMAC while the lower weight head was forced to resign from its role when a higher weight node comes to its neighbor. The value of K needs to be optimized as a lower valued K definitely keeps the number of cluster heads to a restricted size, but in turn it may not be able to outperform over DMAC for the number of reelections. Similarly, a higher valued K might reduce the rate of reelections to a significant value, but at the outset it increases the cardinality of the dominant set, which may further decrease the routing efficiency. Thus, K can be defined as the cluster density control parameter for the whole network.

The second point introduces the threshold H which restricts the number of re-affiliations to take place when a member node comes in contact with multiple cluster heads. The higher the value of H, the lower is the chance that a node will switch to a new cluster. Thus, H can be defined as the member density control parameter for a single cluster. By nullifying both the values of H and K, GDMAC algorithm converges to the original DMAC algorithm.

2.8.1 Result Discussion of GDMAC

This section depicts the result of the GDMAC algorithm. It can be seen that the frequency of reelection of cluster heads shown in figure 2.19 has a remarkable improvement over that of DMAC.

As explained earlier, GDMAC algorithm basically depends on two factors such as K and H. In the current simulation K is considered as 3, i.e. a maximum of 3 cluster heads are allowed to be neighbors of each other. Similarly the weight difference threshold H, is considered to be 32 in this case. It is worth noting that, higher value of K could increase the cluster population as in figure 2.19 and decreases the frequency of reelection of cluster heads and frequency of re-affiliations as in figure 2.20 and 2.21. A comparison between DMAC and GDMAC for the cluster population is explained in [34]. But the value of H could change only the number of cluster changes by the nodes in the network. A higher H definitely causes lower re-affiliations as the nodes are allowed to stay connected with the existing head when the weight difference between the

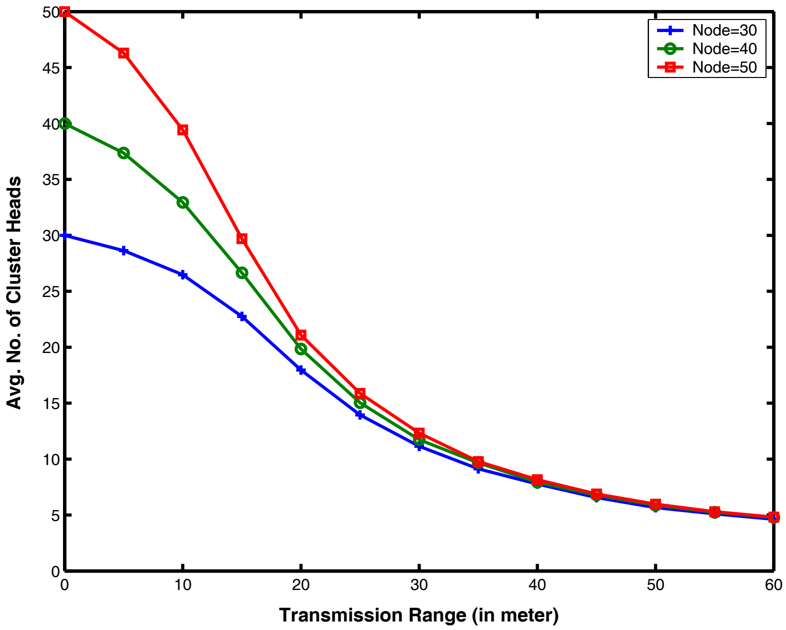


Figure 2.19: Avg. No. of Cluster Heads for GDMAC

current head and the other neighbor head is less than that of the threshold value.

2.9 Weight Based Clustering Algorithm (WBCA)

Another weight based clustering algorithm WBCA has been proposed by Yang et. al. in [35]. This algorithm is a modification over the WCA algorithm that takes the mean connectivity degree and battery power into consideration for calculating the weight of nodes. The mean connectivity degree of a node is calculated as

$$C_v = \frac{\sum_{i=1}^{N_v} N_{vi} + N_v}{N_v + 1} \tag{2.3}$$

where N_{vi} is the degree of connectivity of i^{th} neighbor of node v , and N_v is the degree of connectivity of node v . The consumed energy of a node is

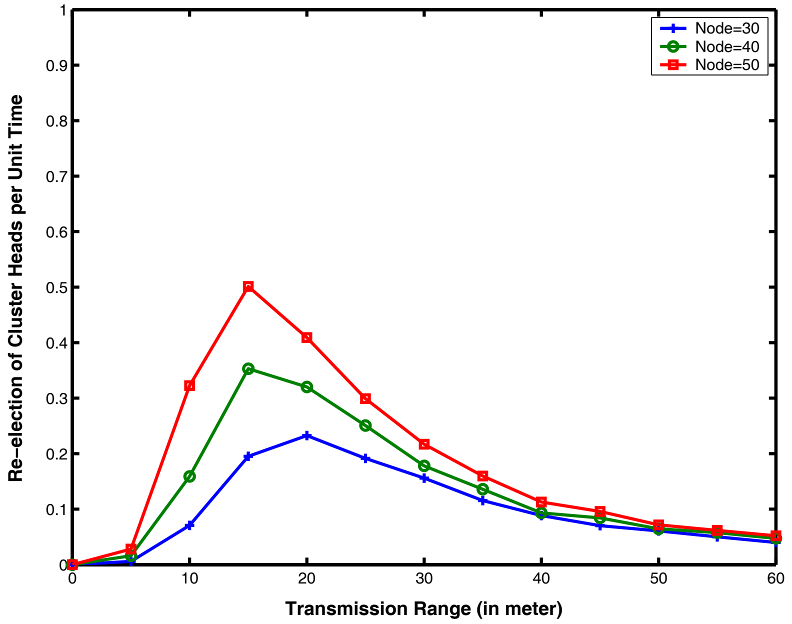


Figure 2.20: Frequency of Cluster Head Re-elections for GDMAC

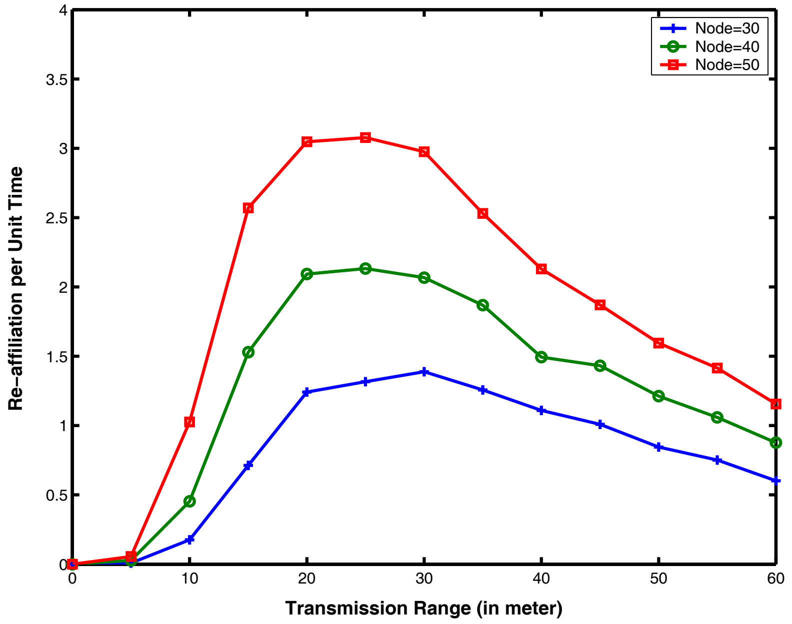


Figure 2.21: Frequency of node re-affiliations for GDMAC

calculated as

$$E_v = \sum_{i=1}^{q} N_{vi} \times e \tag{2.4}$$

where q is the time of period during which the node v acts as cluster head and N_{vi} is the degree of a node v acting as cluster head at i^{th} times. e is the consumed battery power per degree for clusterhead in a period. Finally, the combined weight is calculated as $W_v = w_1 DD_v + w_2 E_v$, where DD_v is the degree difference and is defined as $|N_v - C_v|$ for every node v . The values of w_1 and w_2 are the weighing factors that depend on the system requirements and $w_1 + w_2 = 1$. Unlike Lowest ID (LID) and Highest Connectivity (HC) algorithms, WBCA gives a uniform distribution of time for which the nodes act as cluster head. This also reduces the computation cost of cluster setup as it calculates only two values DD_v and E_v for calculating the combined weight. However, calculating the mean connectivity degree of a single node needs to know the degree of connectivity of all its neighbor nodes. This is typically an unpredictable situation in a dynamic network as the mobility of nodes frequently changes its degree of connectivity. Thus like WCA, this algorithm also needs a considerable amount of freezing time for the nodes before the actual cluster setup.

2.9.1 Result Discussion of WBCA

The results of WBCA for the number of cluster heads remain same to that of WCA algorithm as shown in figure 2.22. WBCA is the weighted variance of the WCA algorithm. That is, the weights of the nodes depend on different parameters in comparison to that of WCA. As a result, there occurs slight change in the simulation result for the frequency of reelection and frequency of re-affiliation of WBCA algorithm. the reelection rate as shown in figure 2.23 is the same as that of WCA. However, the frequency of cluster changes / re-affiliations by the nodes is shown in figure 2.24 is marginally lower than that of WCA.

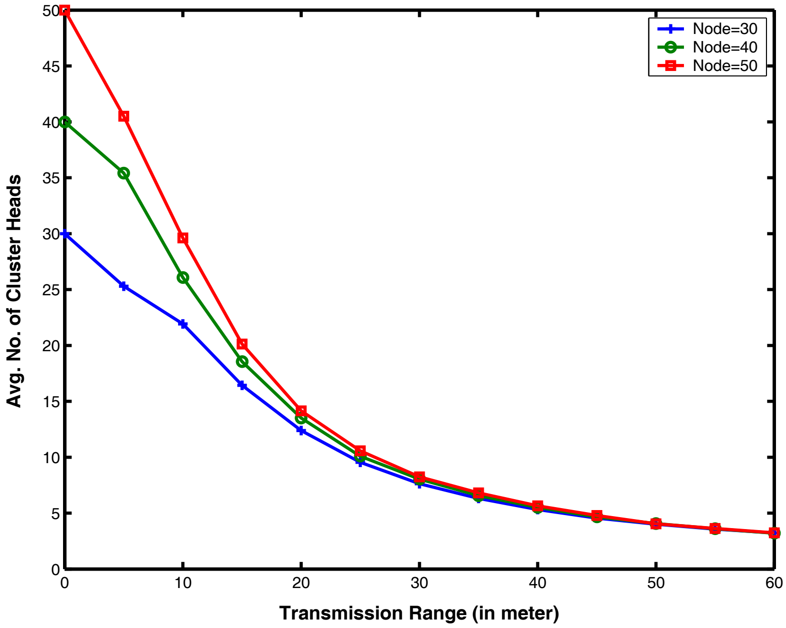


Figure 2.22: Avg. No. of Cluster Heads for WBCA

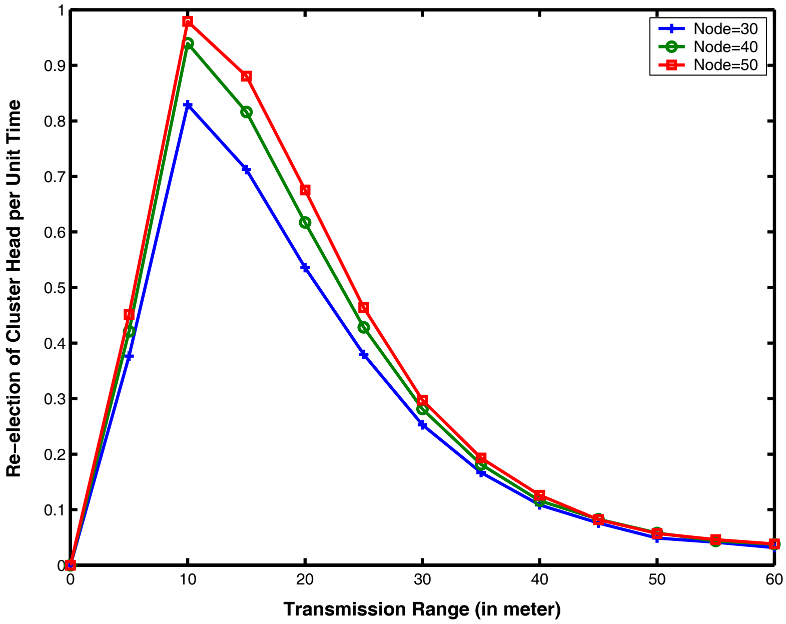


Figure 2.23: Frequency of Cluster Head re-election for WBCA

2.10 Comparison of the Algorithms

In the previous sections, we have presented the basics of cluster formation for different one-hop clustering algorithms along with their simulation result in

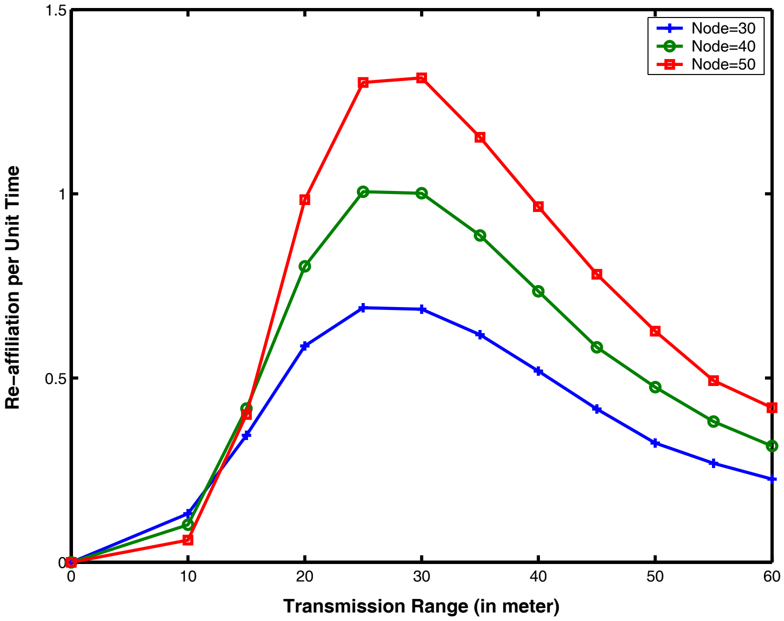


Figure 2.24: Average Cluster Changes by nodes for WBCA

the direction of cluster maintenance. The graphs comparing the results of all the above algorithms are presented in this section. It can be observed from the comparison of the graphs that, the pattern of the results are almost similar for most of the algorithms. Even for some algorithms the differences are marginal. But some algorithms like HC provides worse performance in some cases where as MOBIC has the worst result for some other maintenance parameters. So this can be concluded that each of these algorithms has its own strengths and weaknesses. Depending on the network condition and requirement, any of the algorithms can be chosen for implementation prior to the actual routing job. For example, LID results in an overall good performance among other algorithms in achieving the cluster population, re-affiliation overhead and number of reelections as shown in the figures. But the identification based weight calculation criteria biases the lower ID nodes to become the heads all the time. This may cause faster resource drainage or even node failure to such lower ID nodes.

Similarly HC succeeds in minimising the number of clusters as shown in

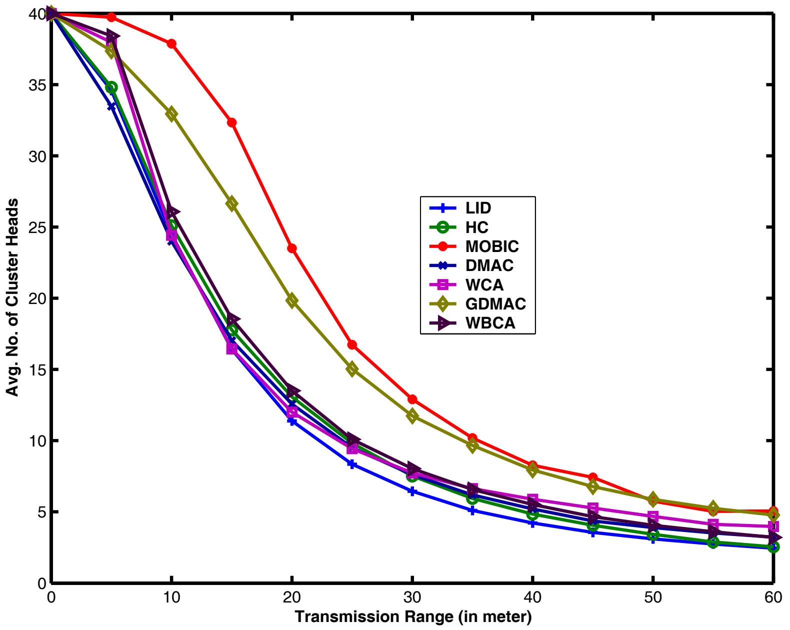


Figure 2.25: Comparison of algorithms for avg. no. of cluster heads

figure 2.25 so that the routing delay is minimised due to reduced number of routing heads. But at the same time the re-elections and the number of re-affiliations are compromised as shown in figures 2.26 and figure 2.27 which does not encourage a designer to choose this algorithm for implementation.

MOBIC and GDMAC have a higher cluster population as shown in figure 2.25 because they allow more than one cluster heads to exist as neighbors for a Cluster Contention Interval (CCI) period or till their weight difference exceeds a threshold value H. Higher cluster population of these two algorithms may increase the routing delay and affect the QoS of the network layer. But, when the stability of clusters is concerned, MOBIC and GDMAC provide the best result due to their lower reelection rate as shown in figure 2.26. So, for these two algorithms it may be concluded that they provide better cluster stability or the route stability (as the cluster heads are the routers of the dynamic network) at the cost of increasing routing hops or the delay in packet forwarding. MOBIC has a very high re-affiliation rate than GDMAC as shown in figure 2.27. So more control messages are exchanged for member updation

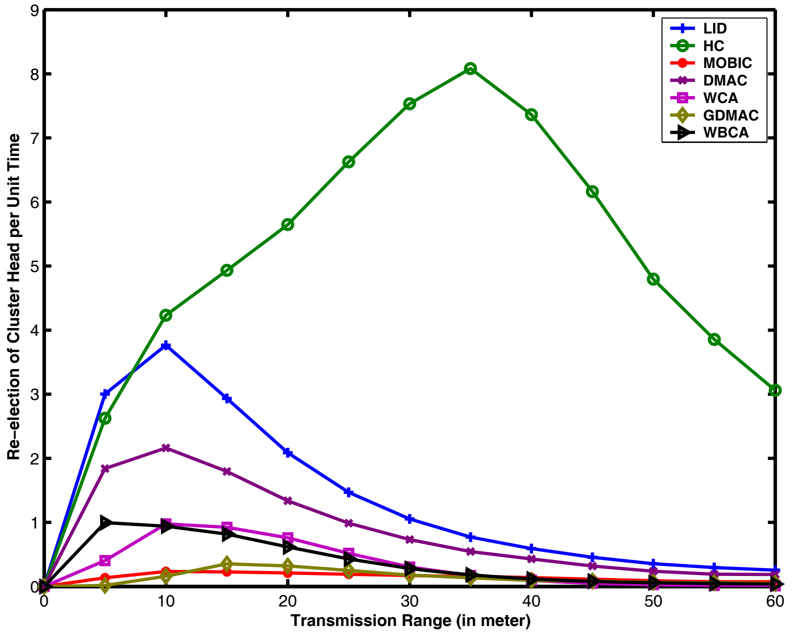


Figure 2.26: Comparison of algorithms for frequency of reelections of cluster heads

between the current head and the newly affiliated head reducing the efficiency of MOBIC.

WCA and WBCA provide better overall performance when all the performance metrics are considered. But they need multiple parameters to be considered for its weight calculation. Thus initial cluster setup is delayed due to the weight calculation and become the designer's choice to decide upon. Further, finding the lowest weight node among all the nodes in the network is not truly distributed in nature. However, the on demand reelection enhances the cluster stability by reducing the number of reelections in 2.26. They also provide lowest re-affiliation overhead in 2.27. In a nutshell, WCA and WBCA can be better option if cluster setup delay is compromised.

Finally, DMAC provides an average performance for all the parameters with a medium frequency of re-affiliations by the cluster nodes and reelection frequency. Its major strength is its mobility adaptive ness during cluster setup. That is node mobility is not freezed during cluster setup. Also, clustering is

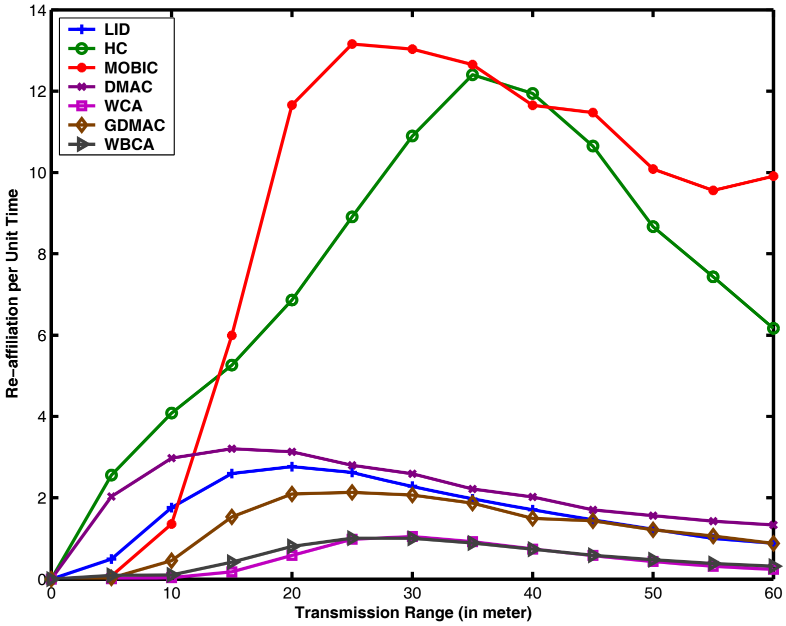


Figure 2.27: Comparison of algorithms for frequency of re-affiliation by the nodes

done with the knowledge of the local topology of the nodes in a distributive fashion. So in a mobile ad hoc network where topology change can not be avoided even during the cluster setup, this algorithm proves to be most applicable than any other existing one-hop clustering algorithms.

2.11 Summary

This chapter makes a thorough survey of clustering algorithms in MANET. In particular, the algorithms under consideration are single hop representatives of the energy constrained ad hoc network. The emphasis is given mostly on the cluster formation principles and the cluster maintenance parameters of the algorithms. Simulation results are discussed to describe the effect of transmission range and the size of the network on the parameters like cluster density, frequency of reelection, frequency of cluster changes in the dynamic network. Partitioning the mobile nodes is a NP-hard problem. So researchers have aimed to obtain either a minimum dominant set or maximum cluster stability by reducing the number of reelections and reaffiliations by the mobile nodes. The principle of partitioning the nodes varies in different algorithms by emphasizing various node parameters such as mobility, connectivity, identification, remaining battery power and sometimes the combination of multiple parameters. However, it could be seen that energy constraint which is a major challenge in this kind of network has not been emphasized properly in any of the algorithm. This provides a motivation for designing an energy efficient clustering algorithm that could reduce the maintenance overhead as well as could increase the cluster stability.

Chapter 3

Neighbour Detection Protocol: A CPN Approach for Validation

3.1 Introduction

With the advent in wireless communication and reduction in price of personal communication devices, the Internet services have reached every corner of the world. The demand of instant networking services has been tremendously increased in the areas of education, entertainment, business centres and emergency services. Ad hoc networks meet these demands quickly and cheaply as it does not need any time for its deployment. The ability of multi-hop network to do the data communication is possible by forwarding the packets through immediate neighbours. Thus, the connectivity among the nodes play a vital role in packet forwarding as well as resource sharing in the network. The node mobility changes their position in the network and so as the connectivity to their neighbors. In the MANET, the wireless nodes move freely while remaining reachable to each other. With fixed topology network, a connectivity matrix can be generated to identify the connections between the nodes. However, for MANET the generation of a connectivity matrix, where the frequent updation is required to reflect the changes in the network topology, is very difficult.

In this chapter, a neighbour detection protocol has been proposed that enables the nodes to find their one hop neighbours in the network. The protocol is modelled by using the well known Coloured Petri Net (CPN) tools and validated through simulation.

3.2 An Overview of Coloured Petri Nets

Coloured Petri Nets is a modelling tool that combines the strength of Petri-Nets with the strengths of formal modelling language. Petri Nets are also called as the Place-Transition Nets where a set of places and transitions provide the primitives for describing synchronisation of concurrent processes. The formal modelling language (ML) provides the primitives for definition of data types and manipulation of their data values.

From the system engineering point of view, it is wise to test and verify the properties of the system to eliminate any possible design errors before the actual implementation or deployment of the system takes place. Coloured Petri Nets have tools, known as the CPN tools that provide a graphical environment to model and simulate any proposed system to analyse it with respect to its required flow of operation. CPN is a formal method which is suitable for modelling analysing complex systems where hierarchy can be created, timing information can be included and complex information can be handled by using siple tokens [36]. Modelling a system using coloured petri nets is useful in which communication, synchronisation and resource sharing plays vital role. This is because, CPNs supports concurrency, non-determinism and distributive which are inherent to most of the system models.

3.2.1 Petri Nets

Petri nets are promising graphical and mathematical modelling tools for describing and studying information processing systems. The concepts of Petri

Nets came into existence in the year 1962 by C.A. Petri, where as from the year 1970 to 1975 the computation structure group at MIT conducted several petri-net related research and produced many reports and thesis on petri nets. Petri Nets could be used to model any system that can be described graphically like flow charts and needs some means of representing parallel or concurrent activities [37]. Few of the application areas include the communication protocols, discrete-event systems, fault-tolerant systems, distributed database systems and multiprocessor memory systems.

The concepts of Petri Nets could be easily understood by a resource allocation system as shown in figure 3.1. Let us assume that we have a set of processes which could be different computer programs and have a set of resources which could be the facilities shared by the programs. As seen in the above figure, we have two sets of processes, i.e., the P-processes and the Q-

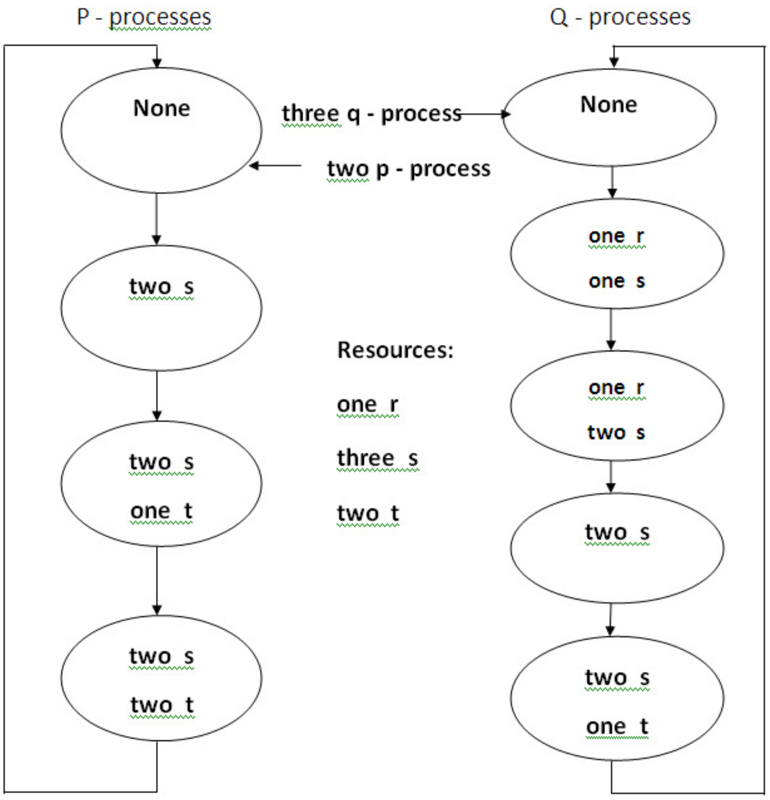


Figure 3.1: States of the processes in the resource allocation system [37]

processes. The processes are cyclic and during the individual parts of its cycle, the process needs to have exclusive access to a varying amount of the resources. As shown in the figure, the P-processes can be in four different states. Similarly, the Q-process have five different states and for each of these states, the required amount of resources is specified. Altogether, we have two P-processes and two Q-processes as shown by the arrows in the figure 3.1. Here it specifies the demands of the processes by describing the possible states. So, we say it is a state oriented system. However, the demands can also be explained as action oriented as shown in figure 3.2, where it gives the detailed description of the possible actions [37].

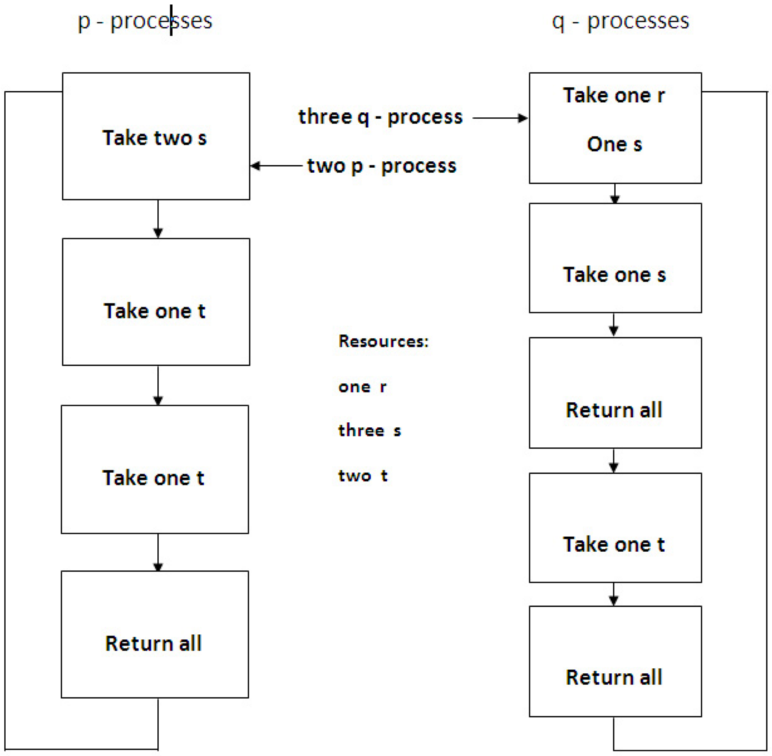


Figure 3.2: Actions of the processes in the resource allocation system [37]

In contrast to the above discussion, petri net specifications of the resource allocation problem is both state-oriented and action-oriented at the same time. The states of the resource allocation system are indicated by means of circles that are called *places* and the actions of the resource allocation system are indicated by means of rectangles, called the *transitions*.

A simple Petri Net model is shown in figure 3.3 [38], where places and transitions are connected with arrows or arcs. Arcs indicate flows from the place to transition or from transition to the place. Sometimes the arcs are labelled with a positive integer called the weight of that particular arc. For example, a k-weighted arc can be interpreted as the set of k parallel arcs. The initial state of a Petri Net graph is called the initial marking M_0. A marking assigns to each place a non-negative integer called the token or resource of the place. If t tokens are assigned to a place then it is marked with t number of black dots inside it in the graphical representation. In a Petri Net model, the transition that represents the action of a system can have certain number of input and output places representing the pre-condition and post-condition of the action. An action can occur or a transition can fire [38]:

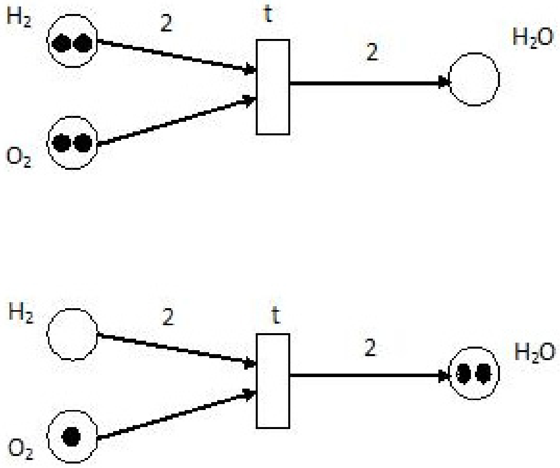


Figure 3.3: Sample petri net model [38]

- If each input place P of transition t is marked with at least $w(p,t)$ tokens, where $w(p,t)$ is the weight of the arc from P to t. A fired transition t

removes $w(p, t)$ tokens from each input place P of t, and adds $w(t, P)$ tokens to each output place P of t, where $w(t, P)$ is the weight of the arc from t to P.

- Depending on whether or not the event actually takes place, the transition may or may not fire.

A transition without any input place is called a source transition, and one without any output place is called a sink transition. A source transition is unconditionally enabled, and the firing of a sink transition consumes tokens but does not produce any. In figure 3.3, the model represents the well known chemical reaction: $2H_2 + O_2 \rightarrow 2H_2O$. Two tokens in each input place show that two units of H_2 and O_2 are available, and the transition t is enabled. After firing t, the state that occurs is shown in figure 3.3, where the transition t is no longer enabled. After taking 2 tokens from the H_2 place, it is left with no more tokens with in it. Where as the place O_2 is left with one token as the occurrence of transition removes only one token from it. The inscription on the arc indicates the number of tokens that can pass through it.

However, the limitations with the traditional petri nets are that, it does not support to design large complex models where modular structure is required. Similarly, the data types handled by the model are very limited for which the model becomes unnecessarily larger even while modelling small systems. Moreover, it is not possible to involve the time concept into it nor does it support to check for the availability of zero tokens in a place. All these limitations led to the design of high level Petri Nets called the Coloured Petri Nets (CPNs).

3.2.2 Coloured Petri Nets

Coloured Petri Nets [39] provide a frame work for the construction and analysis of distributed and concurrent systems [40]. A CPN model describes the states that the system may obtain and the possible transitions in between them. The strength of CPNs over traditional Petri Nets is that, it supports **hierarchy**, **colour**, and **time** in the model.

Hierarchy in the CPNs indicate that the models can be structured into number of related modules. This concept is based on the concept of hierarchical structuring of the programming language, that supports the bottom-up or top-down style. Modules created can be reused in several parts of the CPN model and further sub-modules can be created from it. The modules of the CPNs are called *pages*. A complex model can have as many as hundreds of pages similar to a lengthy and complex program, that is divided into several modules. In hierarchical CPNs, a transition and its associated components make a link to another CPNs providing a more precise and detailed description of the activity represented by the transition. Such transitions are called the *substitution transitions*. The hierarchy inscription in the substitution transition define the details of the substitution in separate modules called the *sub-pages*. The places in a sub-page are marked with an input tag *In*-tag, output tag *Out*-tag or input/output tag *I/O*-tag. These places are called the *port places*. They constitute the interface through which the sub page communicates with its surroundings. Figure 3.4 represents a port place, where the place is assigned with *I/O*-tag. The sub page receives tokens from its surroundings through the input port. It delivers tokens to its surroundings through the output ports and the I/O port communicates to its surroundings in both way. The places

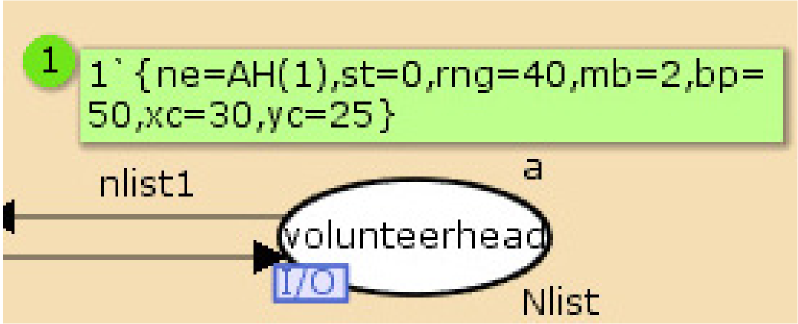


Figure 3.4: Sample place in CPN

associated with a substitution transition are called the *socket places*. The port places of the sub pages are related to the socket places of the substitution transition by providing the port assignments [practitioner's guide]. When a port place is assigned to a socket place, the two places become identical. The port place and the socket place are two different representations of a single conceptual place, i.e. the port and the socket places have always identical markings. When an input socket receives a token from the surroundings of the substitution transition, that token also becomes available at the input port of the sub-page, and hence the token can be used by the transitions on the sub-page. Similarly, the sub-page may produce tokens on an output port. Such tokens are also available at the corresponding output socket and hence they can be used by the surroundings of the substitution transition.

Another concept of hierarchical CPNs is the *fusion places* shown in figure 3.5. This indicates that a number of individual drawn places can be considered to be identical. i.e they all represent a single conceptual place. When a token is added or removed at one of the places, an identical token will be added or removed at all other places in the fusion set. So it is clear that the relationship between the members of a fusion set is similar to the relationship between

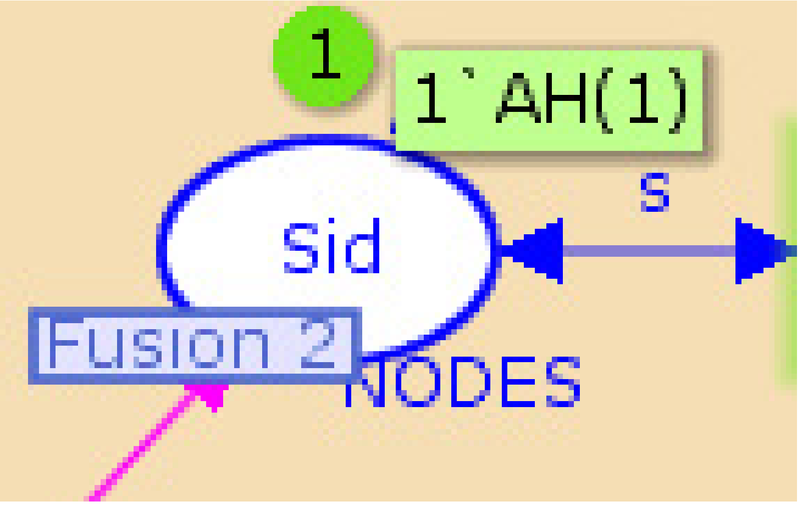


Figure 3.5: Fusion place in CPN

two places which are assigned to each other by a port assignment. When all members of a fusion set belong to a single page and that page has only one page instance, place fusion is nothing more than a drawing convenience to avoid too many crossing arcs in the model. But the situation is complex and interesting when the members of a fusion set belong to several difference subpages or to a page that has several page instances. The various kinds of fusion sets are the global fusion sets, page fusion sets and instance fusion sets. The global fusion set can have members from different pages where as the page fusion set and instance fusion sets only have members from a single page.

Colours associated with each place in the CPNs determine the type of data it may handle. The types of the places are similar to the types in programming language. It can be a complex type as the record which may contain heterogeneous data types. The colour set is usually defined as:

colset No= int;

Where “colset” is a keyword to declare the color set, “No” is the name of the colour set and “=int” indicates that this colset can have integral values as tokens. The state of a CPNs is called as its state that shows the number of tokens distributed on the individual places. Each token carries a value that belongs to the type of the place on which the token resides. The tokens present on a particular place denotes the *marking* of that place. The initial state of a place is denoted as the initial marking of it. It is usually written in the upper left or right of the place as shown in figure 3.4 and figure 3.5.

Time concept into the CPNs is redefined as timed CPNs. This introduces the concept of global clock. The clock value which is either discrete or continuous represent the model time. In the timed CPNs, each token carries a time value called the time stamp. The time stamp describes the earliest model time at which the token can be used, that is it can be removed by the occurrence of a binding element. In a timed CPNs a binding element is said to be color enabled when it satisfies the enabling rule for un-timed CPNs. However, to be enabled, the time stamps of the tokens to be removed must be less than or equal to the current model time [40].

The marking of a place where the tokens carry a time stamp becomes a timed multi-set specifying the elements in the multi set together with their number of appearances and their time stamps. The timed color sets are declared as :

colset No=int timed;

and the possible marking of a place with timed token is as:

2'(1,"colour")@[19,45]

This indicates the marking contains two tokens with value (1, "colour") and time stamps 19 and 45 respectively. The @ symbol can be read as "at" and the symbol [ ] is used to specify the time stamps.

3.2.3 State-of-Art on CPNs in Mobile Ad Hoc Networks

Coloured Petri Nets has been proved to be a powerful tool for simulating and analysing the non-determinism, concurrency and different level of abstraction of any communication protocol. Zhou et. al. have proved the strength of CPN for simulating and analysing TCP protocol [41]. Further to improve the TCP performance over MANET, Xiong et. al. [Xiong, Yim, Leigh and Murata] have proposed a reactive approach TCP-MEDX to detect the causes of packet loss. As mobility of nodes is the biggest challenge in MANET, the round trip time is replaced with an average propagation delay for indicating congestion. The authors claim that the TCP-MEDX mechanism is able to detect the packet loss much accurately.

Xiong et. al. have created a formal CPN model of the very well known routing protocol for MANET, the Ad hoc On-Demand Distance Vector (AODV) to analyse its correctness in service [42]. To meet the challenge of dynamism of the nodes, the authors have proposed a topology approximation (TA) mechanism. The TA mechanism works with certain assumptions. They are:

1. All the nodes in the MANET have equal transmission range

2. Every node in the MANET has the same number of neighbors which is equal to the average degree of the MANET graph.

Here, the second assumption is not a realistic approach. Because in MANET, the node movement frequently change the degree of connectivity among the nodes and also the network topology. In such a non-deterministic environment, the mechanism of topology approximation may not be possible.

Further, the dynamic operation of MANET using CPN are illustrated by Yuan et. al. in [43]. The authors have modelled the well known MANET routing protocol named as Destination Sequenced Distance Vector (DSDV), which is proactive by principle. The CPN tools are used by the authors for the elegant and simple modelling of the protocol without using any assumption or approximation. By using the formal specification and verification method of the modelling tool, the authors could be able to find the errors existing in the protocol and suggested the modification to eliminate those errors.

3.3 The Proposed Neighbour Detection Protocol

The mobile ad hoc network can be modelled as a unidirectional graph $G = (V, L)$ where V is a finite set of mobile nodes and L is a finite set of links that exist between the nodes. We assume that there exists a bidirectional link L_{ij} between the nodes i and j when the distance between the nodes $d_{ij} < t_{range}$ (transmission range) of the nodes. In the dynamic network the cardinality of the nodes $|V|$ remains constant, but the cardinality of links $|L|$ changes due to the mobility of the nodes. Each node $v \in V$ is uniquely identified by an integer identifier ID along with a wireless transmission range v_{trange}. When a node v_1 is within the transmission range of v_2, they are assumed to be connected by a unidirectional link $l_{12} \in L$, such that whenever v_1 broadcasts a message, it is received by v_2 via l_{12} . Similarly, when v_2 is within the transmission range of v_1 , they are assumed to be connected by an unidirectional link $l_{21} \in L$, such that whenever v_2 broadcasts a message, it is received by v_1 via l_{21}. Thus we say that there exists a bidirectional link between v_1 and v_2. The graph G representing the ad hoc network is assumed to be a simple graph. That is an un-weighted, un-directed graph containing no self-loops or no parallel links (when their end points are same).

The neighbour detection protocol (NDP) has been designed to provide mechanism to detect the one-hop neighbours of the nodes in the ad hoc network. Here, the nodes use a special control packet called the *neighbour detection packet* (NDPAK) to realise their neighbours. The format of the packet is as:

As a node u wants to know about its existing neighbours, it broadcasts a neighbour detection packet (NDPAK). The sender specifies it as a neighbour request (NRQ) type packet and keeps the receiver identification field (RID) as X. The node(s) that is(are) within the transmission range of the sender u, receives the packet and send back acknowledgements to the sender. The packet being an acknowledgement packet has the type of packet as NAC and makes an unicast transmission to the source. Thus the RID field contains the identification number (NUM) of the source node that originally sent the NDPAK request. The node weight field WT is a non-negative number that is calculated by considering some node parameters to specify its capability to be selected as a cluster head. The STATUS field of the node indicates its current role in the network. In the current work, it is assumed that the nodes can have either the role of cluster head or cluster member or an undecided state. Initially, the status of every node is undecided with a status value of 0 before the cluster is actually formed. Subsequently, as the node is selected as cluster head or cluster member, this field is filled with appropriate values.

The mechanism of the neighbour detection protocol (NDP) is as follows:

Step 1: Node u broadcasts the Neighbour Detection Packet (NDPAK) to the network.

SID	RID	STR/RTR	NRQ/NAC	WT	STATUS

where the fields of the packet are specified as follows:

SID: Source Identification

RID: Receiver Identification

X: ALL nodes (Used in the broadcast message)

NUM: node with identifier NUM (used in the unicast message)

STR: Source Transmission Range

RTR: Receiver Transmission Range

NRQ/NAC:

NRQ: The neighbour request packet

NAC: Neighbour acknowledgement packet

WT: Weight of the packet sending node

STATUS: Status of the packet sending node

0: Uncovered

1: Cluster Member

2: Cluster Head

Step 2: Let the packet is received by node v which is within the transmission range of u. Node v sends back a Neighbour Acknowledgement (NAC) Packet to u along with its own information like ID, transmission range, weight and status enclosed in the packet.

Step 3: After receiving the acknowledgement NAC packet from v, node u updates its neighbour table (NTAB) by adding v as its immediate neighbour along with its information.

Step 4: Finally, u sends back a Neighbour Confirmation (NC) message so that v updates its own neighbour table and a bidirectional link is established between the two nodes.

The protocol can be explained with the help of figure 3.6 that consists of five nodes.

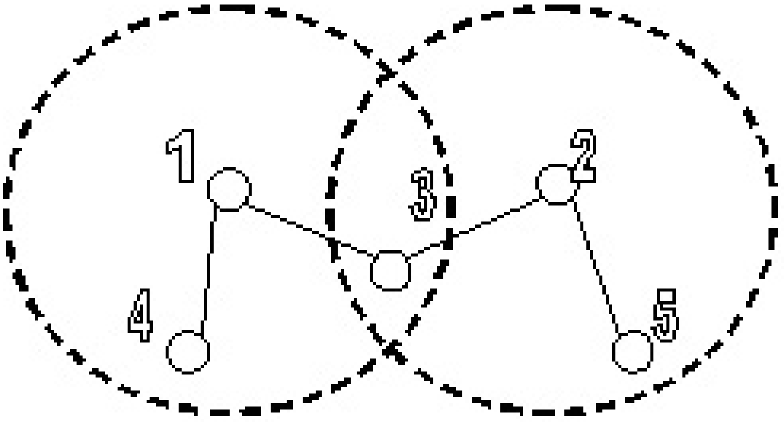


Figure 3.6: A sample network topology

Here, the nodes are identified with unique integer numbers from 1 to 5. The transmission zones for the nodes 1 and 2 are indicated by the dotted circles around it. It is clear from the figure that node 3 lies in the transmission range of both nodes 1 and 2 whereas node 4 and 5 are exclusively in the range of 1 and 2 respectively. The NDPAK sent by node 1 contains the following data:

1	X	1_{trange}	NRQ	WT(1)	0

Here, the STR field contains 1_{trange} indicating the transmission range of node 1 and the value 0 in the STATUS field indicates that the state of the node 1 is not yet decided. This packet will be received by both node 3 and 4, and they will send back the NAC packet to node 1 as an acknowledgement to the NDPAK. The contents of the acknowledgement packet (NAC) sent back by node 3 to node 1 may be written as:

3	1	3_{trange}	NAC	WT(3)	0

For the topology shown in figure 3.6, where node 3 receives the NRQ packet from both nodes 1 and 2, it sends back the NAC packets accordingly to both the senders and becomes the neighbours of both the nodes. The arrival of the NAC packets informs the sender about the presence of its one-hop neighbour and it updates its own neighbour table NTAB accordingly. The data structure of the neighbour table (NTAB) may be written as:

NID	DIST	NS	NTR	NWT

where the field specifications are as:

`NID:  Neighbour ID`

`DIST: Neighbour Distance`

`NS:   Neighbour Status`

`NTR:  Neighbour Transmission Range`

`NWT:  Neighbour Weight`

The neighbour distance DIST may be calculated from the signal strength of the received signal [44]. However, for the current work, the nodes are assumed to be equipped with the global positioning systems (GPS) that provides the location information of the nodes. So, here the Euclidean distance between two nodes is considered as the actual distance between the two nodes. When an acknowledgement packet is received by the source node, it updates its NTAB and sends on a NC message to confirm its connectivity with the later so that a bidirectional link is established between both of them.

3.4 Validation of NDP by Coloured Petri Nets

This section deals with the validation of the Neighbour Detection Protocol (NDP) using CPN tools. The validation through simulation is carried out with six nodes in the network. For the purpose of validation, the nodes are considered to be non-mobile during the execution of the protocol. The top level of the CPN model for the NDP is shown in figure 3.7. This model represents

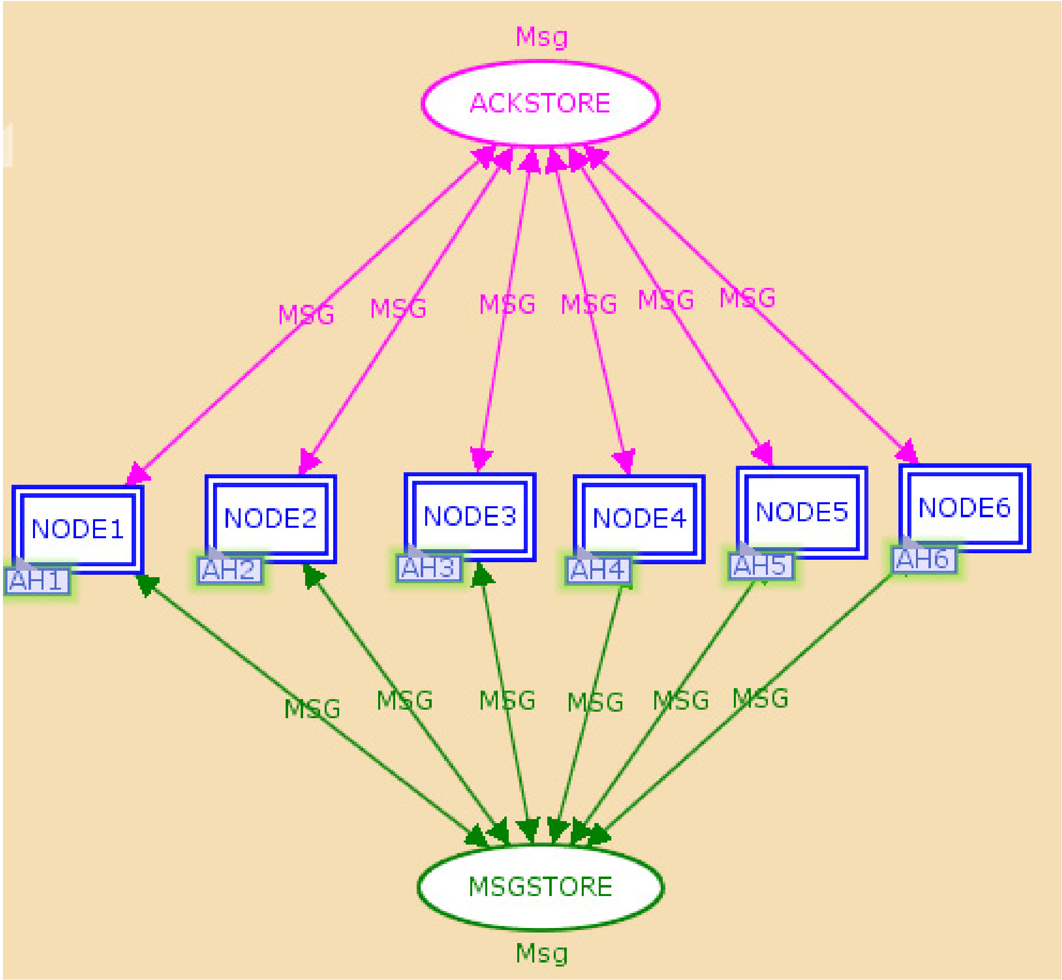


Figure 3.7: Top level of the model

an abstract view of the entire NDP. The substitution transitions *NODE*1 to *NODE*6 have their own subpages associated with it.

Two places are kept common for all the nodes in the model. They are the message store place *MSGSTORE* and the acknowledgement store place *ACKSTORE*. The place *MSGSTORE* contains the messages transmitted by all the nodes in it. Similarly, all the acknowledgements transmitted by the nodes are stored in the place *MSGSTORE*. The token types associated with both the places are denoted as Msg which is a record of different data types discussed later in this section. The nodes 1 to 6 are connected to the places through bidirectional arcs. This indicates that a node can store as well as retrieve its messages from the places.The inscriptions associated with the arcs indicate the kind of token it can pass through it. They are written in the CPN ML programming language.

Figure 3.8 represents the detail model of the ad hoc nodes in their current marking state M_0. The model consists of nine transitions and fifteen places.

The specifications of the places are:

- **coord** - This place specifies the current position of the node in terms of x and y coordinates. As the proposed protocol assumes the presence of GPS devices installed in the mobile devices to find their locations, so for the modelling purpose we have manually provided the location information of the nodes in terms of (x,y) coordinates. This place is associated with the colour set *POSITION* which is defined using the CPN ML keyword *colset* as:

  colset POSITION= product int * int ;

  This means that the token residing in the place is a two-tuple element, where both the components are of integer type. The number of tokens associated with the place is marked with an encircled 1. And the initial token value or the initial marking of the place is in scripted within the parenthesis as a pair of two elements. The no. of associated tokens along with the current marking of the place is indicated in a rectangular box at the upper right corner of the place.

- **STATUS** - This place specifies the current status of the node. In the shown model, the initial marking of the place is zero to indicate that its current status is yet to be decided. Usually, a node can have a status of a cluster head(2) or a cluster member(1) or undecided (0). The colour set associated with this place is INT type is defined as:

  colset INT= int;

  The number of tokens associated with the place is 1 as indicated in the model as an encircled 1.

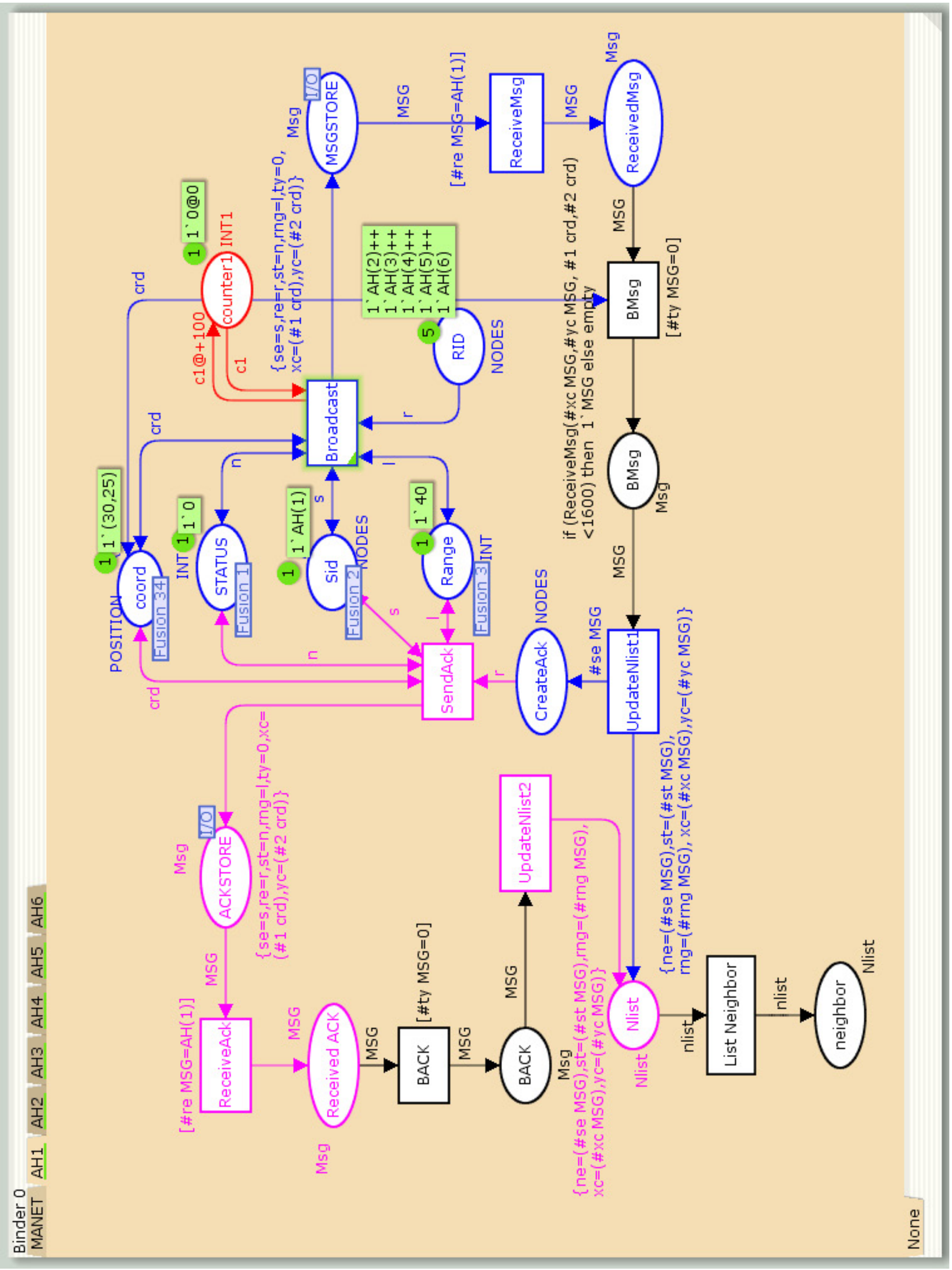


Figure 3.8: The model of NDP in initial marking M_0

- **Sid** - This place specifies the source identification number which is unique for the nodes in the network. The colour set NODES associated with this place is defined as:

  colset NODES=index AH with 1..NoOfNodes;

  where NoOfNodes is defined as:

  val NoOfNodes=6;

  This place holds a single token along with its current marking as the AH followed by a subscripted identification number. i.e. $1'$ AH(1).

- **Range** - This place specifies the transmission range of the node. In the current model it is kept a fixed value of 40 for all the nodes for the sake of simplicity. The colour set associated with this place is again INT type which is declared as:

  colset INT=int;

  The number of tokens that reside with this place is 1 and its current value is set to 40.

- **RID** -This place specifies the list of possible receivers to whom the source can send the messages. The maximum number of token that can be associated with the place is five. That is all the nodes except the sender itself are the possible receivers. The colour set associated with this place is same as that of *Sid*. The only difference is that it can hold upto 5 tokens at a time as shown within a circle in the model. The current marking of the place contains the receiving node IDs combined with an operator ++. This operator allows for the construction of a multi-set consisting of more than one token colours. The initial marking of the place is defined as:

  val A= $1'$AH(2)++ $1'$AH(3)++$1'$AH(4)++$1'$AH(5)++$1'$AH(6);

- **MSGSTORE** - As explained earlier, this place holds the messages broadcasted by all the nodes. The colour set associated with the place is Msg which is declared as :

colset Msg=record se:NODES*re:NODES*st:INT*rng:INT*ty:INT*xc:INT*yc:INT

where *se* denotes the sender node, *re* denotes the receiving node, *st* denotes the status of the node, *rng* denotes the range of the node, *ty* denotes the type of the message (in this case 0 for a message from an ordinary node and 1 for a message by a cluster head), xc and yc are the x coordinate and y coordinate respectively.

- **ReceivedMsg**- This place holds messages that are destined for a specified receiver. The token type associated with this place has the same colour set as that of *MSGSTORE* as both stores the messages in the same format.
- **BMsg** - This place holds the intermediate message for a node from its valid neighbor nodes, i.e. the nodes within the transmission range. The colour set associated with the place is Msg type which is discussed earlier.
- **Nlist and neighbour** - Both the places hold the list of the neighbours of the concerned node. For the figure 3.8, the concerned node is AH(1) as indicated in the *Sid* place. The colour set associated with this place is again a record that is described as:

  colset Nlist=record ne:NODES*st:INT*rng:INT*xc:INT*yc:INT;
- **CreateAck** - This place specifies the node to whom an acknowledgement has to be send. The token colour set for this place is NODES type which is described earlier.
- **ACKSTORE**- As discussed earlier, this place holds all the acknowledgement messages transmitted by the node. The bidirectional arcs from this place to the nodes help the nodes to access the place for both storing and retrieving. It holds the tokens described by the colour set Msg.

- **Received ACK** -The specification of the place is same as the place *ReceivedMsg*, except with the difference of the type of message it stores.
- **BACK** - Broadcasts the acknowledgement message to help the neighbor node to update its own neighbors.

The list of the transitions and their specifications are as:

- **Broadcast** - The occurrence of this transition carries the source node information from the respective input places in the form of a record and adds a token into the place *MSGSTORE*. The input arc expressions from the input places makes it an enable transition as shown by the green *auras* in the marking M_0. The inscription in the output arc of the transition creates the message to be stored in the output place *MSGSTORE*. The inscription of output arc is denoted as: *se=s,re=r,st=n,rng=t,ty=0,mb=m,bp=b,xc=(#1 crd),yc=(#2 crd)*
- **ReceiveMsg** - The occurrence of this transition is possible only when a valid token is present in its input place. That is when the *MSGSTORE* holds a specific message destined for the particular node, this transition is activated. For all other messages it remains inactive. In the current marking state M_0 of figure 3.8 it is not activated as the input place has not yet obtained any tokens. However, the situation is different for figure 3.9. We can see that the concerned node in this case is node 1 as indicated in the place *Sid*. The place *MSGSTORE* contains two messages with the *# re* field specified as AH(1). The messages are from nodes AH(3) and AH(4) respectively. The presence of these two valid tokens in the input place makes the transition *ReceiveMsg* enabled by putting green auras around it.
- **BMsg** - As this transition occurs, it checks for the presence of neighbor nodes of the concern node by finding the Euclidean distance between them. This transition has an additional input from the place *coord* to

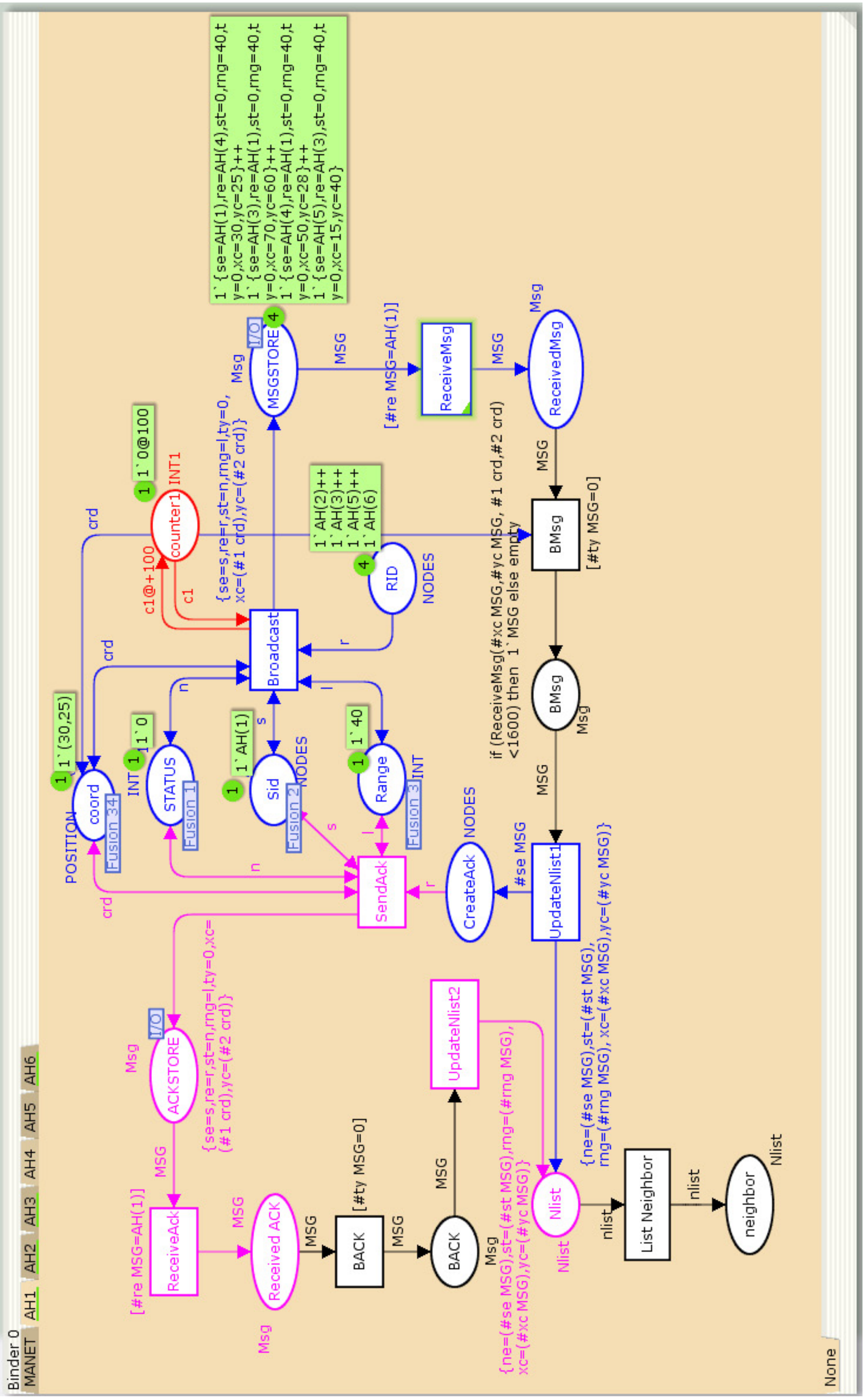


Figure 3.9: Marking M_1: After the occurrence of Broadcast in M_0

indicate the (x,y) coordinates of the node. The inscription MSG on other input arc carries the received message to the node. A function *ReceiveMsg* in the output arc is used to find the Euclidean distance between the two nodes by considering the (xc, yc) fields of the *MSG* and the coordinate fields of the arc inscription *crd*. The result of this transition is boolean type. If the distance between the two nodes are lesser than the transmission range, then that message is passed to the output place *BMsg* for further processing, else it is ignored. The output inscription of this transition is denoted as:

if(ReceiveMsg(#xc MSG, #yc MSG, #1 crd, #2 crd) ¡1600) then 1′MSG else empty

- **UpdateNlist1** - The occurrence of this transition gives two outcomes. First, it passes the message from the input place to the output place *Nlist* to store it as the neighbour details. Second, it passes the sender node ID to be stored in the output place. The output arc extracts the *#se* field from the *MSG* to get the node *ID* of the message sender.

- **List Neighbor** - The occurrence of this transition finally stores the neighbor details in the place *neighbour*. The output arc inscription of this transition nlist is defined as: var nlist:Nlist;

  where Nlist is described earlier.

- **SendAck** - This transition has multiple inputs. The function of this transition is to create an acknowledgement message. The input place *CreateAck* provides the node ID to which the acknowledgement is to be sent. The other input places like *coord, STATUS, Sid, Range* provides the relevant information to the transition so that the complete message is formed. The message gets stored in the out place *ACKSTORE.*

- **ReceiveAck** - The occurrence of this transition checks the input place *ACKSTORE* for the stored messages. As it finds any message whose #

re field is for the concerned node, then that message is removed from the input place and stored in the output place *Received ACK*. The inscription of the transition mentions the activation condition of the transition. For example *# re MSG=AH(1)* indicates that the *re* field of the message MSG has to be AH(1) to enable this transition.

- **BACK** - The occurrence of this transition is for broadcasting the acknowledgement to the sender.
- **UpdateNlist2** -The outcome of this transition is same as that of the transition *UpdateNlist1*. The occurrence of this transition removes one token from the input place BACK and stores it in the place *Nlist*.

Figure 3.9 shows the marking of the model as the transition *Broadcast* is occurred in marking M_0. The current marking of the model is represented as M_1. As seen in figure 3.8, *Broadcast* is the only enabled transition exists with green *auras* around it in the marking M_0. When this transition is occurred, it removes a token from each of its input places *coord, STATUS, Sid, Range and RID* forming a record of type Msg. The output arc of the transition passes the token and stores it in the place *MSGSTORE*. The bi-directional arc to some of its input places indicate that the occurrence of the transition replaces back the token values to the corresponding places for future use. The only place to which the token is not returned back is *RID*. The marking M_1 shows the messages stored in *MSGSTORE* that has been broadcasted from different nodes. In the marking M_1 the total number of tokens currently stored in **MSGSTORE** is 4. The inscription *# re MSG=AH(1)* of the transition *ReceiveMsg* explains that, the availability of a token in the place **MSGSTORE** with the *re* field intended for node AH(1) can make the transition enabled. As shown in figure 3.9, two messages from the senders AH(3) and AH(4) have *re* field destined for AH(1) stored in *MSGSTORE*. The green auras around the transition *ReceiveMsg* indicates that, the availability of certain tokens in the *MSGSTORE* has made it enabled, while rest of the transitions in the model remain unchanged.

The state of the model in figure 3.10 is marked with marking M_2.

The state has occurred with the occurrence of the transition *ReceiveMsg* after the availability of the token in *MSGSTORE* with *re*=AH(1). The state M_1 shows that two messages in the place *MSGSTORE* have the receiver field *re* as AH(1). In marking M_2 the occurrence of *ReceiveMsg* has removed a token from *MSGSTORE* with sender *se* = AH(3). The same token has been added to the place *ReceivedMsg* as its current marking. The availability of the token in the place has made an enabled binding for the transition *BMsg*. In marking M_2 the place *MSGSTORE* is left with 3 messages for which the transition *ReceiveMsg* still remains active.

The state M_3 of figure 3.11 shows the possible outcome of transition *BMsg*.

The output arc of the transition is inscripted with a receiving function *ReceiveMsg* that takes the positional coordinates of the sending and receiving nodes as the parameter. The function finds the Euclidean distance between two nodes by extracting the *xc* and *yc* coordinates of the receiving message MSG and current markings of the place *coord* that gives the positional coordinates of the concerned node. The output of the receiving function is the distance that is compared with the transmission range of the sender. If the distance is less than the range, then the message is received by the receiving node. This is indicated by passing the message MSG from the input place of the transition *BMsg* to the output place of the transition. Marking M_3 shows that the token that was present in the place *ReceivedMsg* with *se*= AH(3) and *re*=AH(1) could not reach the place *BMsg* after the occurrence of the transition *BMsg*. It does not satisfy the condition of the receiving function. However, the same marking M_3 shows that another token with *se*=AH(6) and *re*=AH(1) has arrived to the place *ReceivedMsg* by the occurrence of the transition *ReceiveMsg* keeping the transition *BMsg* still active. The marking also shows the accumulation of some more message tokens in the place *MSGSTORE*. However, the status of rest of the transitions remain unchanged.

The situation is slightly different in figure 3.12, where there is a successful

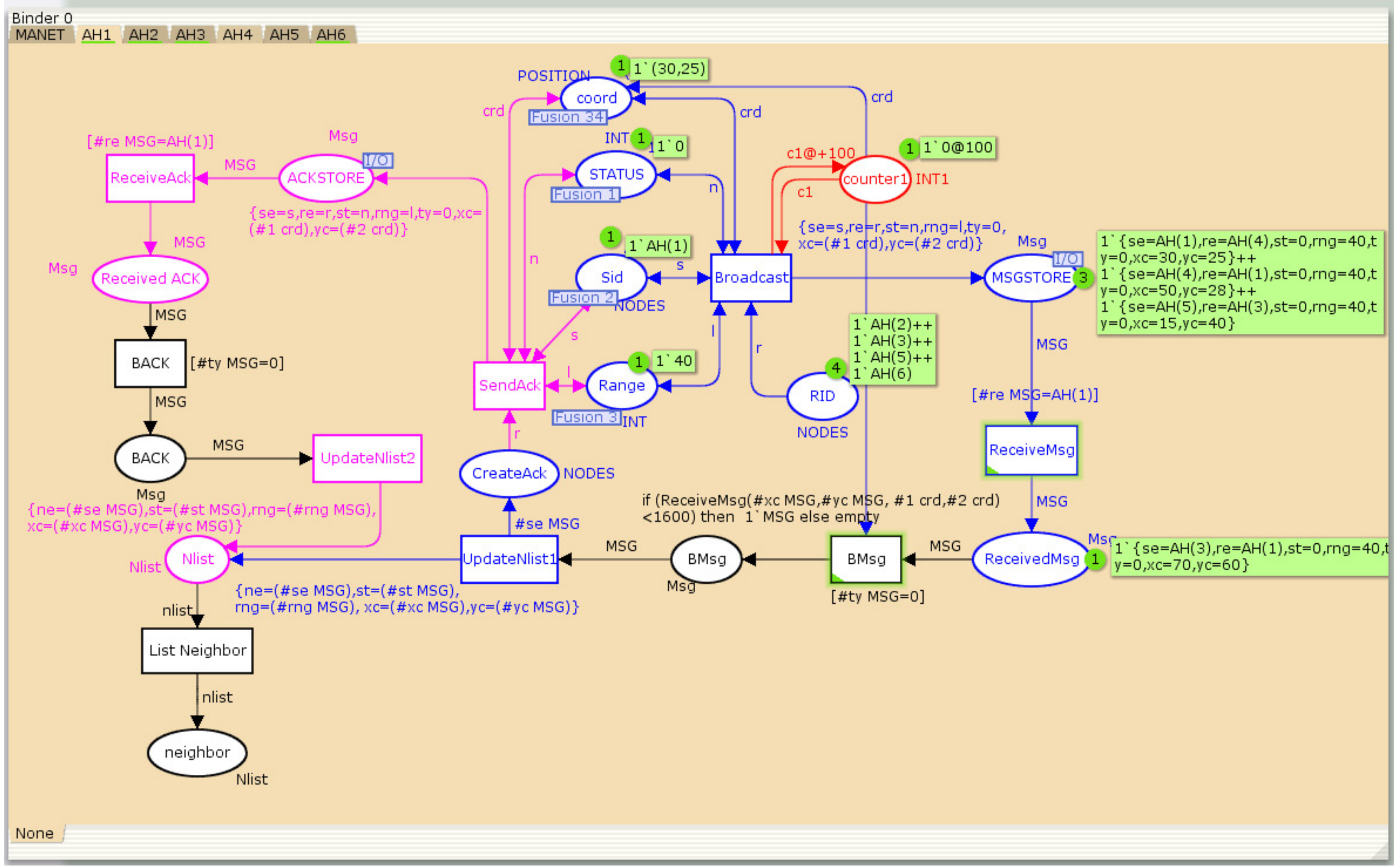


Figure 3.10: Marking M_2 : After the occurance of ReceivedMsg in M_1

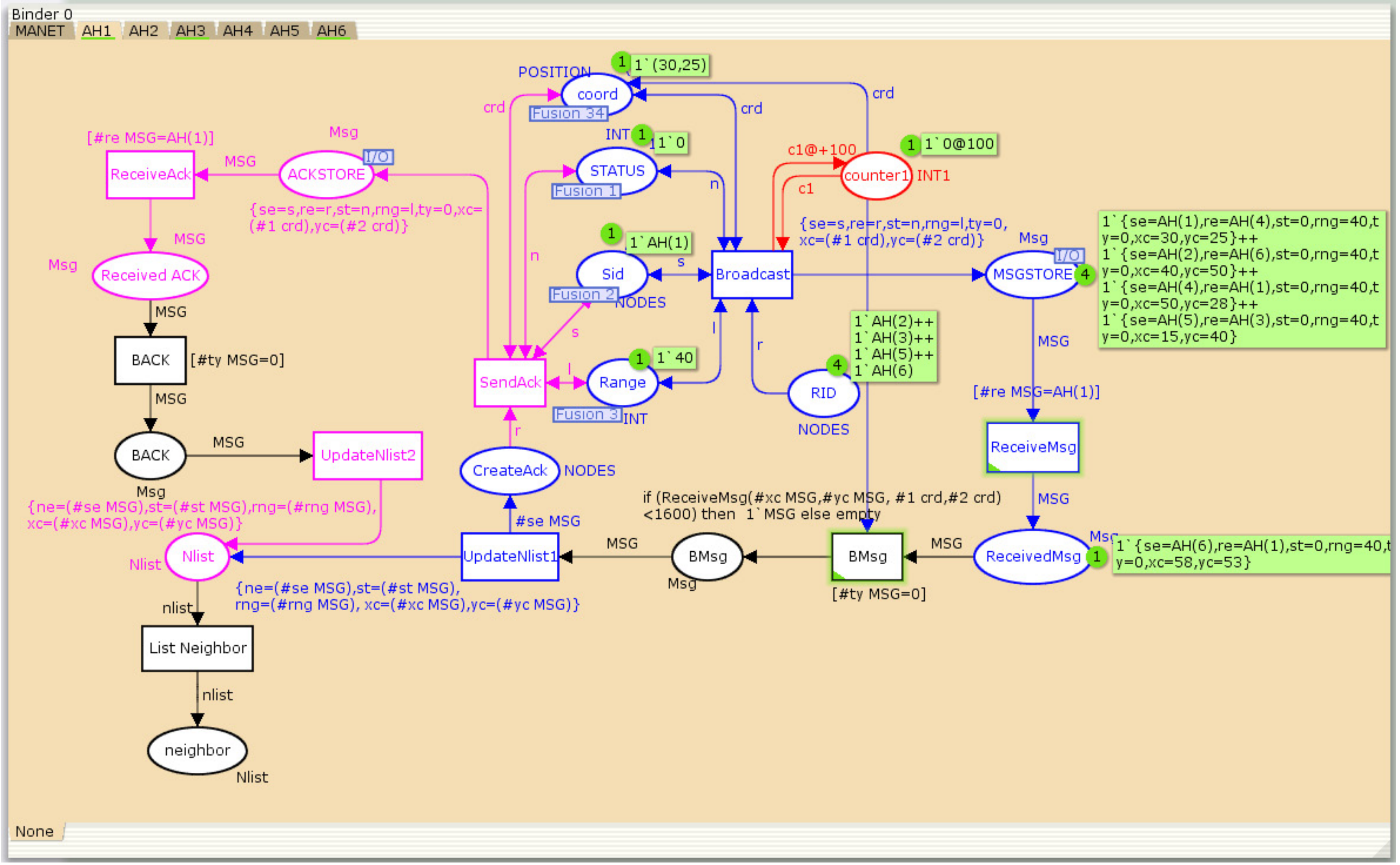


Figure 3.11: Marking M_3 : When the occurrence of BMsg fails to satisfy the condition

transmission of message from the input place to the output place of transition *BMsg*.

The current state M_4 has a successful occurrence of *BMsg*. That is the distance between both the sending and receiving node is found to be less than the transmission range. As a result, the message from the sender *se*=AH(6) to the receiver *re*=AH(1) has been removed from the input place *ReceivedMsg* and added as a new token to the place *BMsg*. The availability of this token in the place makes an enabled binding for the transition *UpdateNlist1* which is seen in marking M_4 of figure 3.12.

The next marking M_5 indicates the successful occurrence of *UpdateNlist1*. It results in two concurrent events as its two outcomes. One is the arrival of the token to the output place *Nlist* as in figure 3.13.

The colour set associated with the place *Nlist* is a record that contains the information like neighbor ID *ne*, its status *st* , range *rng* and its current position *xc* and *yc*. The inscription associated with the out arc of transition *UpdateNlist1* extracts the neighbor information from the message MSG and stores as the token to the output place as:

ne=AH(6),st= 0,rng=40, yc=58,xc=53.

Another result from the occurrence of the transition *UpdateNlist1* is the extraction of the sender field *se* from the message MSG. This is denoted in the inscription of the out arc of the transition for creating the acknowledgement. In the marking M_5 the sender *se* of the message is node AH(6) which has been filtered from the *MSG* and is stored as the only token in the output place *CreateAck*. The availability of the token in the place has made an enabled binding for the transition *SendAck* by making green auras around it. In this marking we can see that three transitions of the model are in the enabled state because of the availability of the tokens in their input places. The transitions are *ReceiveMsg*, *SendAck* and *List Neighbor* respectively.

The occurrence of transition *ReceiveMsg* in M_5 has resulted in the marking M_6 as shown in figure 3.14. A message stored in the *MSGSTORE* which was

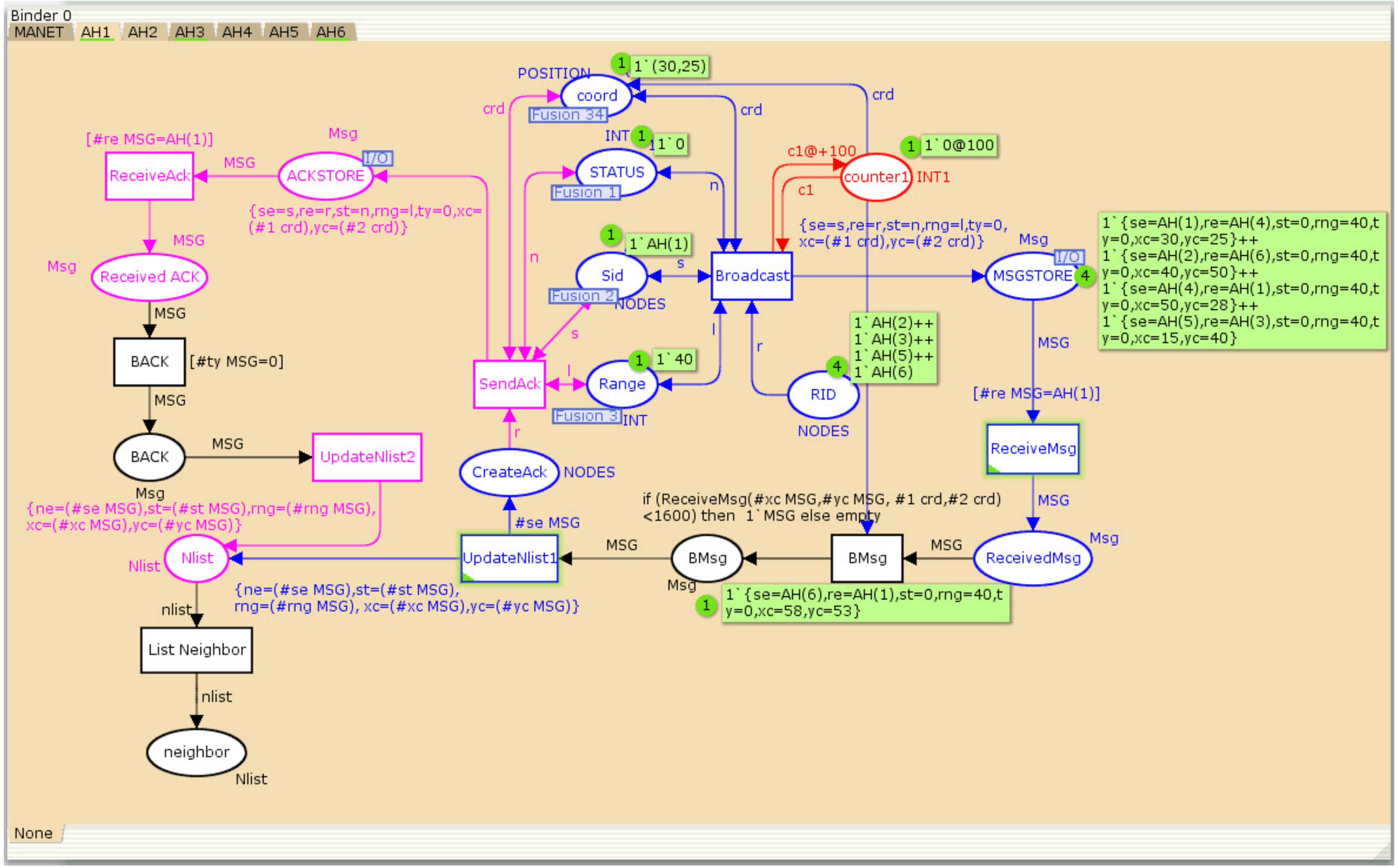


Figure 3.12: M_4 : Successful occurrence of transition BMsg

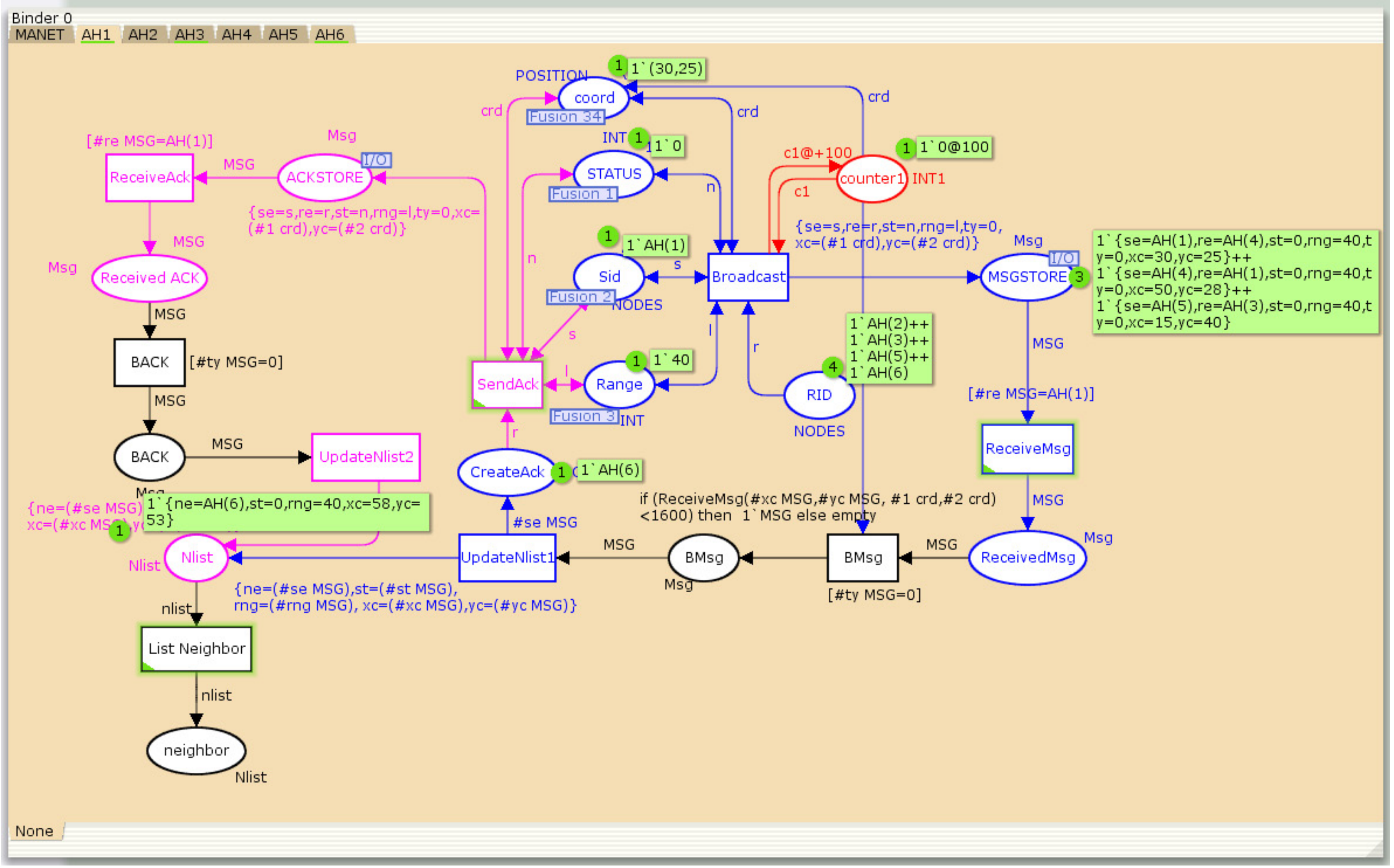


Figure 3.13: Marking M_5 : State with the occurrence of UpdateNlist1 in M_4

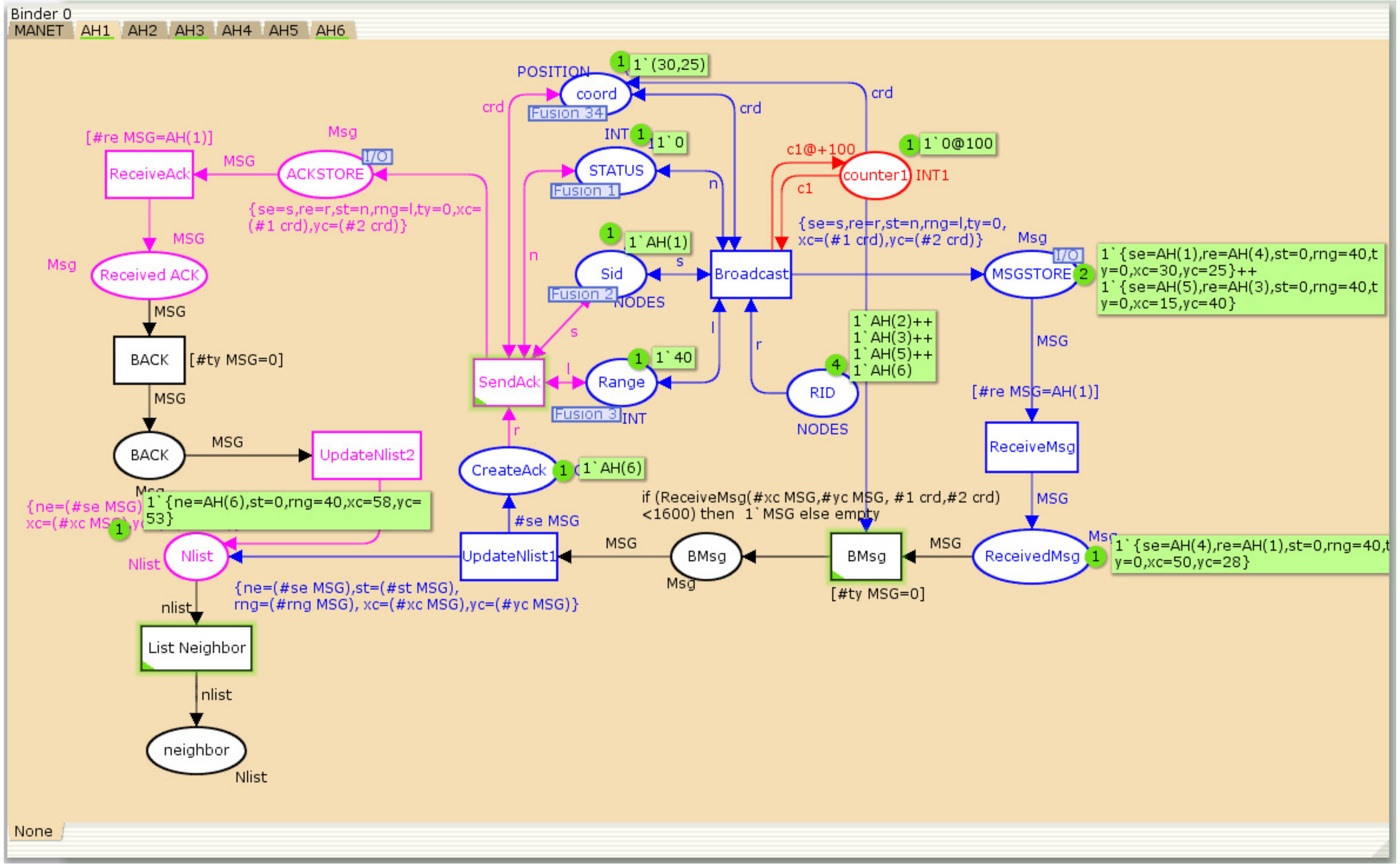


Figure 3.14: Marking M_6 : Occurrence of transition ReceiveMsg in M_5

sent by the sender AH(4) for node AH(1) made an enabled binding for the transition as in marking M_5. The successful occurrence of the transition removes the message from the place *MSGSTORE* and stores in the place *ReceivedMsg* as a single available token. The input place *MSGSTORE* still remains with two more tokens. But this does not make an enabled binding for the output transition *ReceiveMsg* as none of the tokens have re = AH(1).

Marking M_7 in 3.15 is the result with the occurrence of the transition *List Neighbor* in marking M_6.

It removes the token from the input place *Nlist* and adds a token to the output place *neighbor*. The inscription on the arc shows the type of the message that could be passed through. The token with all the neighbor information of node AH(1) has thus arrived to the place *neighbor* for future processing. Now the node AH(1) stores the details of its neighbor AH(6)in the neighbor table NTAB. So it is concluded that node 6 is now within the range of node 1. For setting up a bidirectional link between the both, another link is required to be established between node 6 and 1. As per the mechanism of the neighbor detection protocol, node 1 should send an acknowledgement to node 6. The rest of the article describes it.

The next occurrence of the transition that has taken place is *BMsg* . This has resulted to the marking M_8 in 3.16.

The rest of the transitions in the model remain unaltered. Similar to the state in M_4, the receiving function in the out arc of the transition satisfies the condition to result in the successful occurrence of *BMsg*. Thus the token is removed from the input place of *BMsg* and is stored in the output place *BMsg*. The sender *se* of the message is the node AH(4) as shown in the figure.

The successful occurrence of the transition *UpdateNlist1* in M_8 results into the state M_9 in 3.17.

The two outcomes of the successful occurrence of the transition are:

- 1. The token is removed from the input place *BMsg* and added to the output place *Nlist*. In the current marking, the place *Nlist* holds the

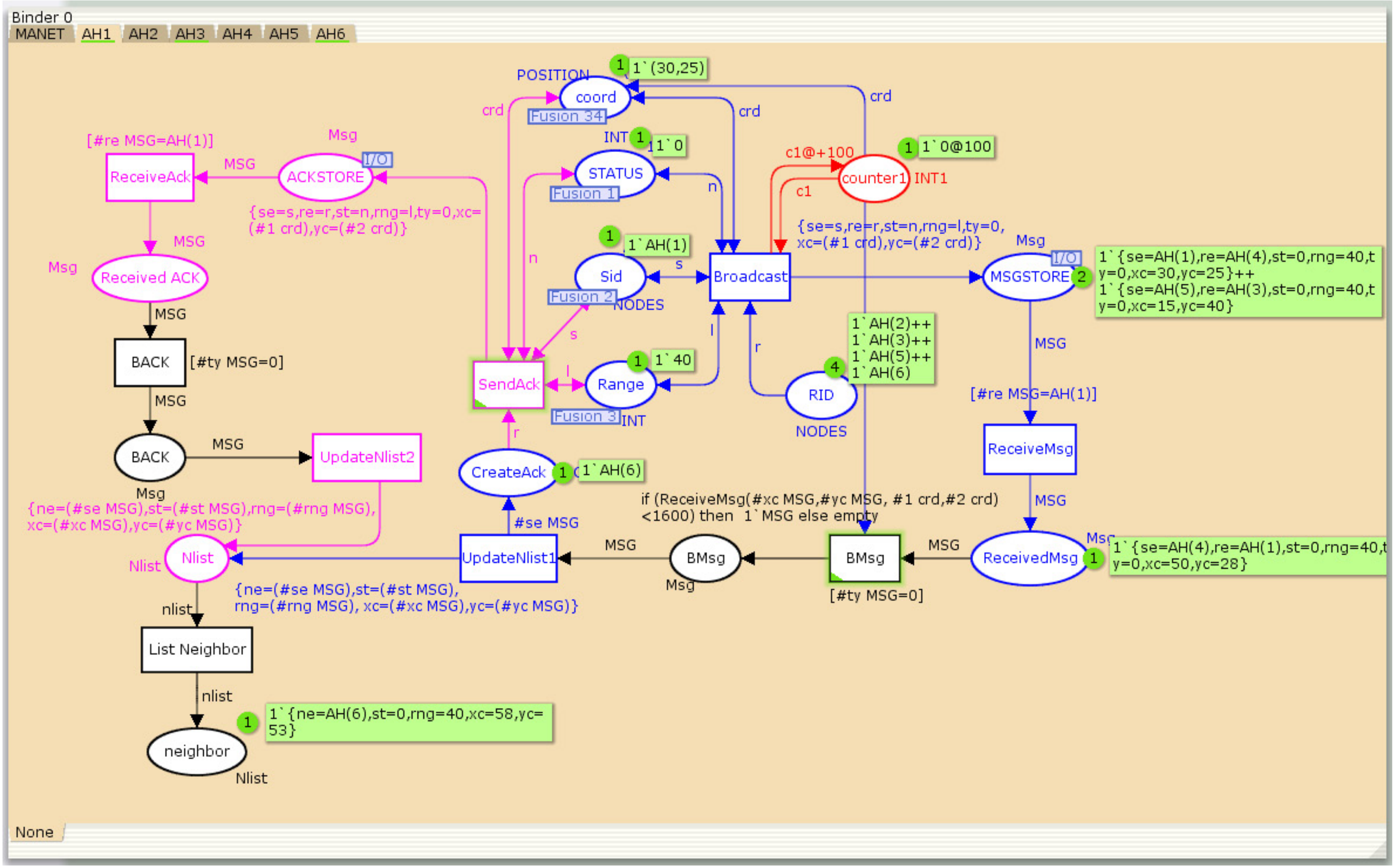


Figure 3.15: Marking M_7 : Node 1 receives the neighbor details with the occurrence of List Neighbor in M_6

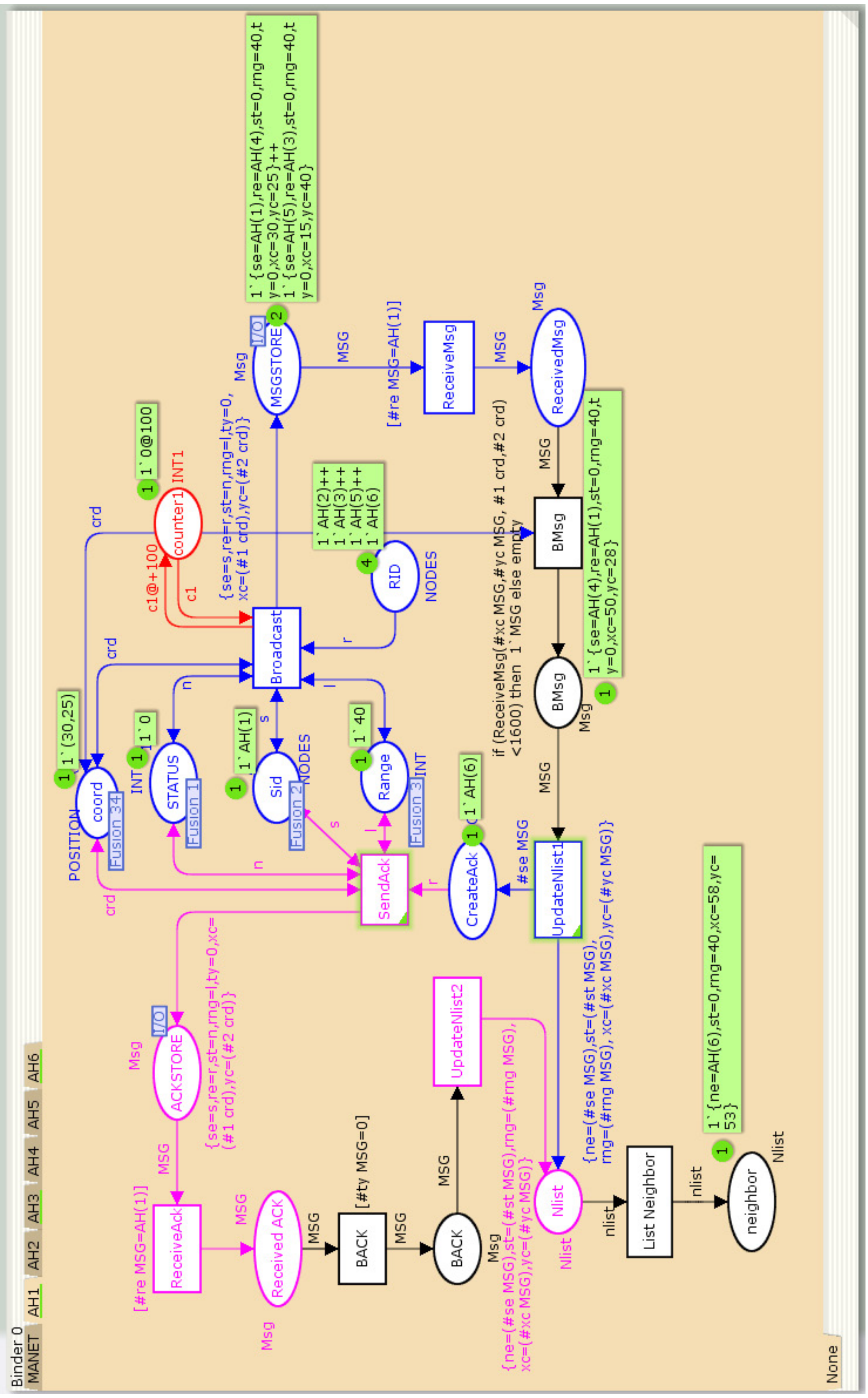


Figure 3.16: Marking M_8 : With the successful occurrence of BMsg in M_7

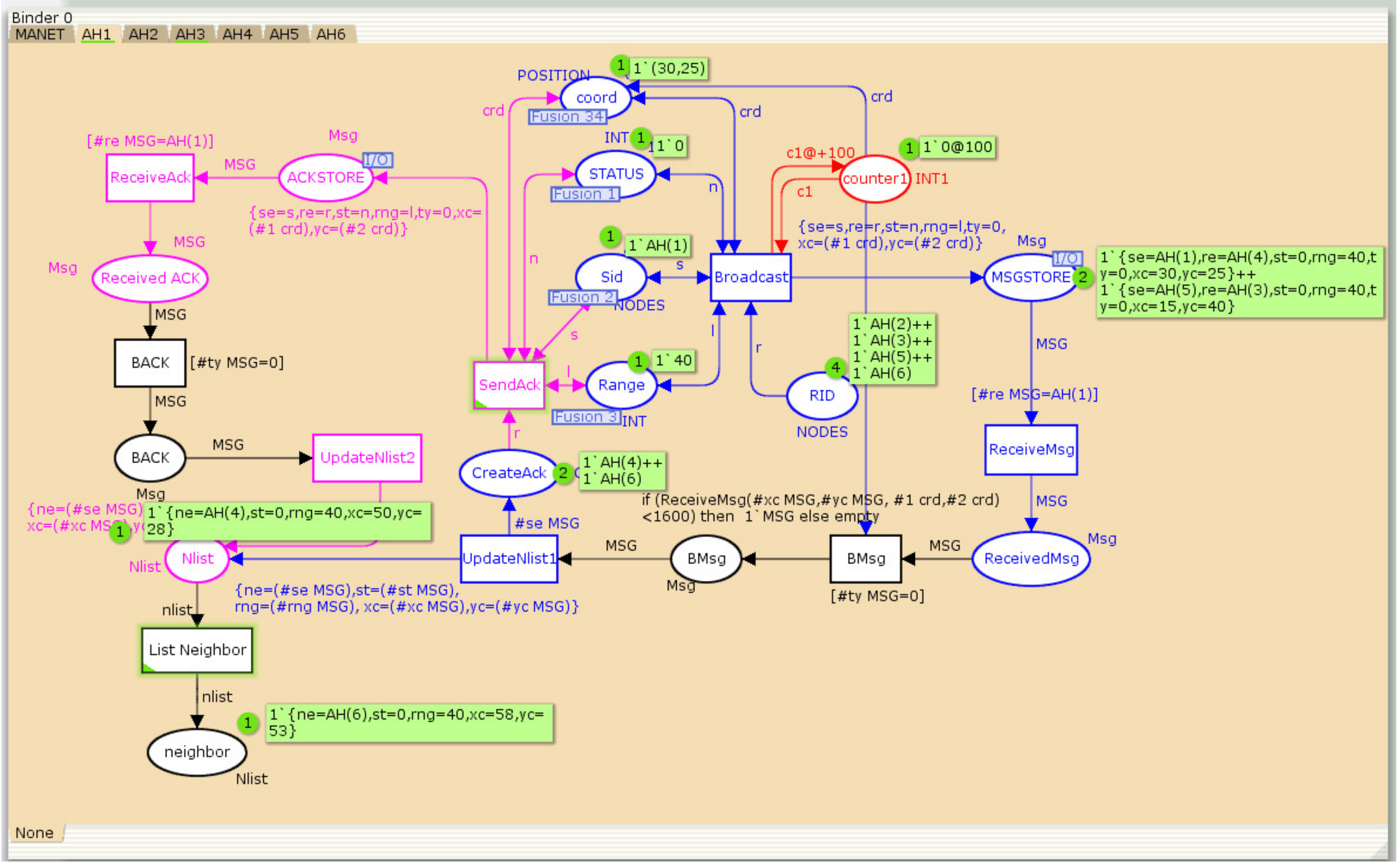


Figure 3.17: M_9 : The state with the occurrence of UpdateNlist1 in M_8

details like the node ID *ne*, status *st*, range *rng*, and the (x,y) coordinate of the sender node. The availability of token in the place *Nlist* makes the transition *List Neighbor* enabled by creating green auras around it as shown in marking M_9.

- 2. The sender field *se* has been extracted from the message *MSG* by the inscription on the out arc of the transition and placed in the output place *CreateAck*. As a result the place *CreateAck* now holds two tokens AH(4) and AH(6)as its current marking. This implies the node AH(1) that has received the messages from the said nodes has to send acknowledgements to both of them so as to enable them to update their neighbor table *NTAB*.

The occurrence of the transition *List Neighbor* has placed another token in the place *neighbor*, so that the NTAB of AH(1) is updated with one more neighbor. This is shown in the current state M_{10} in 3.18.

It is clearly understood from the figure that node AH(1) has two neighbor nodes AH(6) and AH(4). In the current marking none of the transitions have an enabled binding except the transition *SendAck*.

In marking M_{11}, the tokens in the place *CreatAck* indicates that two acknowledgements are to be sent to the detected neighbors of node AH(1). The result with the occurrence of the transition *SendAck* is shown in 3.19.

Here one token is removed from the input place *CreateAck* and is added to the place *ACKSTORE*. As discussed earlier, the enabled transition *SendAck* gets input from the places *Sid*, *STATUS*, *coord* and *range* so that a complete record of message of type Msg is created. The format of the acknowledgement message is as:

se=AH(1), re=AH(6), st=0, rng=40, ty=0, xc=30, yc=25.

The place *ACKSTORE* is commonly accessible to all the nodes in the network. The availability of a token for node AH(6) as indicated in the message field *re* MSG=AH(6) provides an enabled binding for the transition *ReceiveAck*

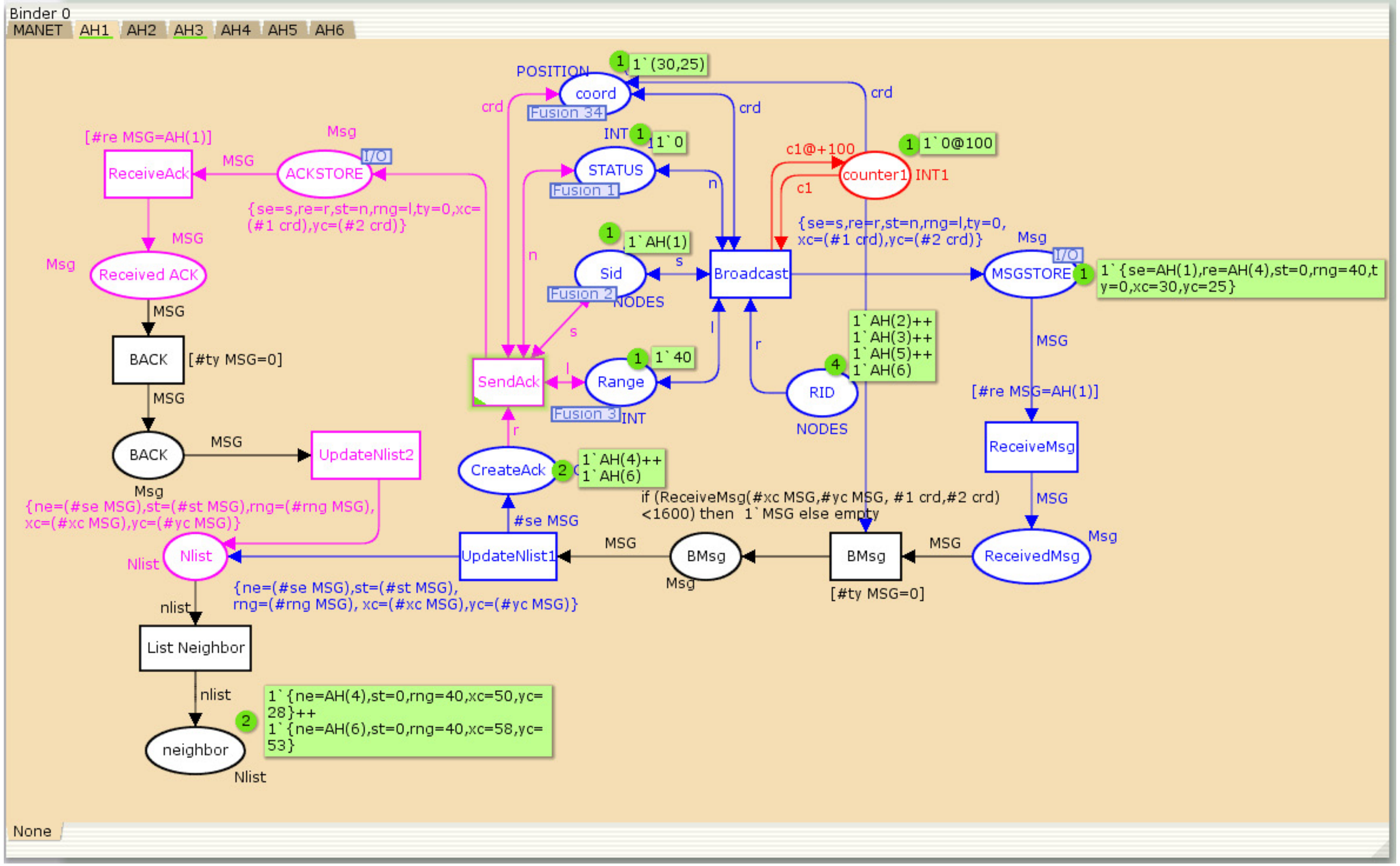


Figure 3.18: Marking M_{10} : The state with the successful occurrence *List Neighbor*

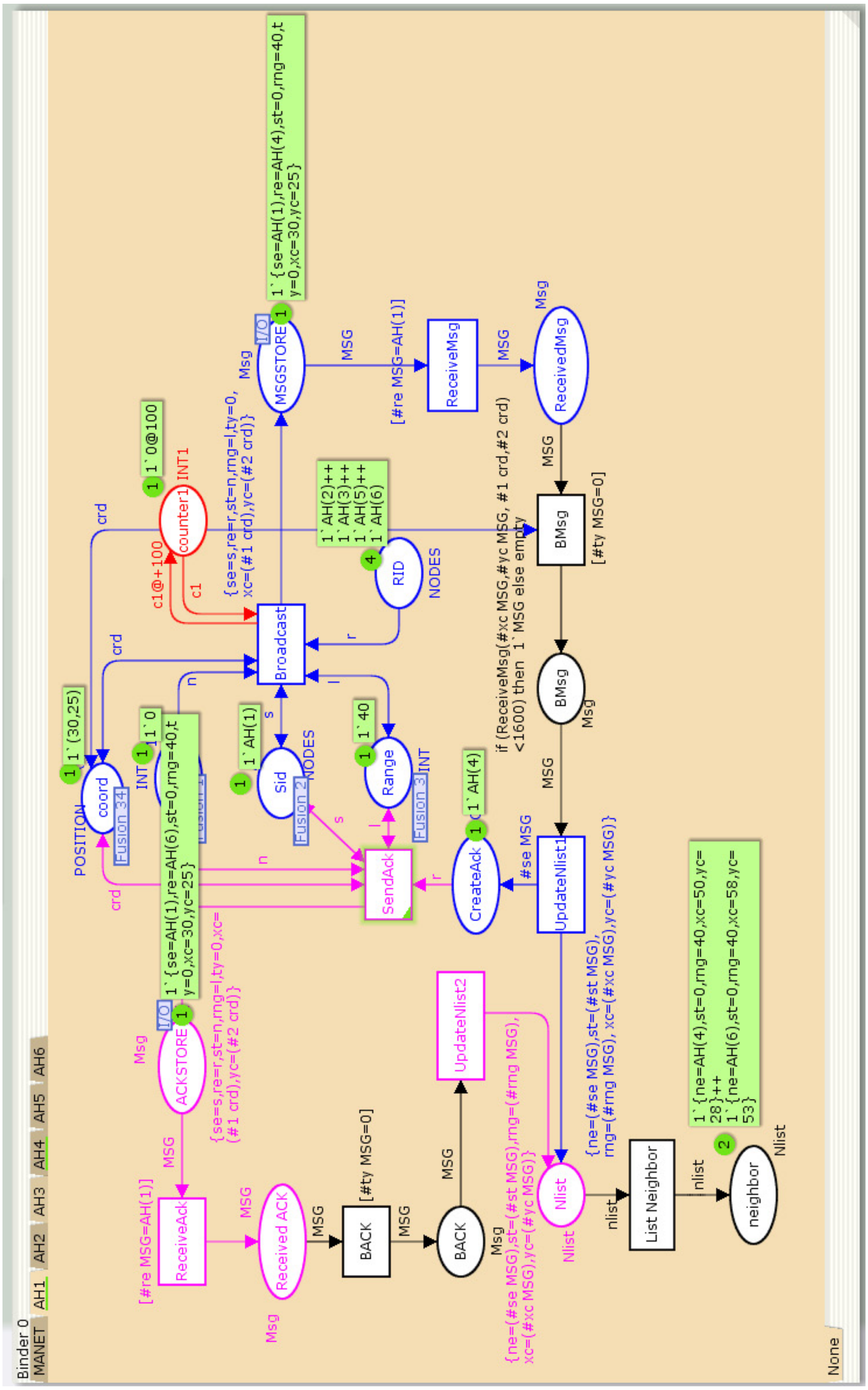


Figure 3.19: Marking M_{11} : With the successful occurrence of *SendAck*

of node AH(6). The successful occurrence of this transition removes the token from $ACKSTORE$ and stores in the place *Received Ack*. Further a consecutive occurrence of the transition *SendAck* removes the available token from the place *CreateAck* and stores in the common place $ACKSTORE$. The current marking M_{12} in figure 3.20 shows the outcome after the successive occurrence of two transitions.

As seen in the figure 3.21, one of the acknowledgement receiver (in this case node AH(4)) has received the acknowledgement. This state is marked as M_{13} in figure 3.21. This shows the availability of the token in the place *Nlist* of node AH(4). Thus the node AH(4) now records node AH(1) as its neighbor node along with its other information.

The same sequence of occurrence of transitions take place for every page corresponding to the mobile nodes. As a result of this simulation, every node gets updated with their neighbor nodes in the table NTAB.

3.5 Summary

CPN model of NDP provides new insights and thoughts into the design of the system. The successful simulation of the model results in further simpler and more streamlined design. Moreover, many design problems and errors can be discovered and resolved in the analysis and design phase rather than in the implementation or deployment phase. Simulation results of CPN model can validate a system to justify that it has been designed properly to meet the desired properties and help in better understanding of the required system. The study of the existing work on MANET using CPN tools has provided the motivation to propose the required protocol for the current work and verify its operation using the CPN tools to ensure its correctness before implementation.

The purpose of designing NDP is to enable every node to identify its one hop neighbour node. The protocol creates a neighbour table for every node in the network and keeps track of the neighbours. The validation of NDP

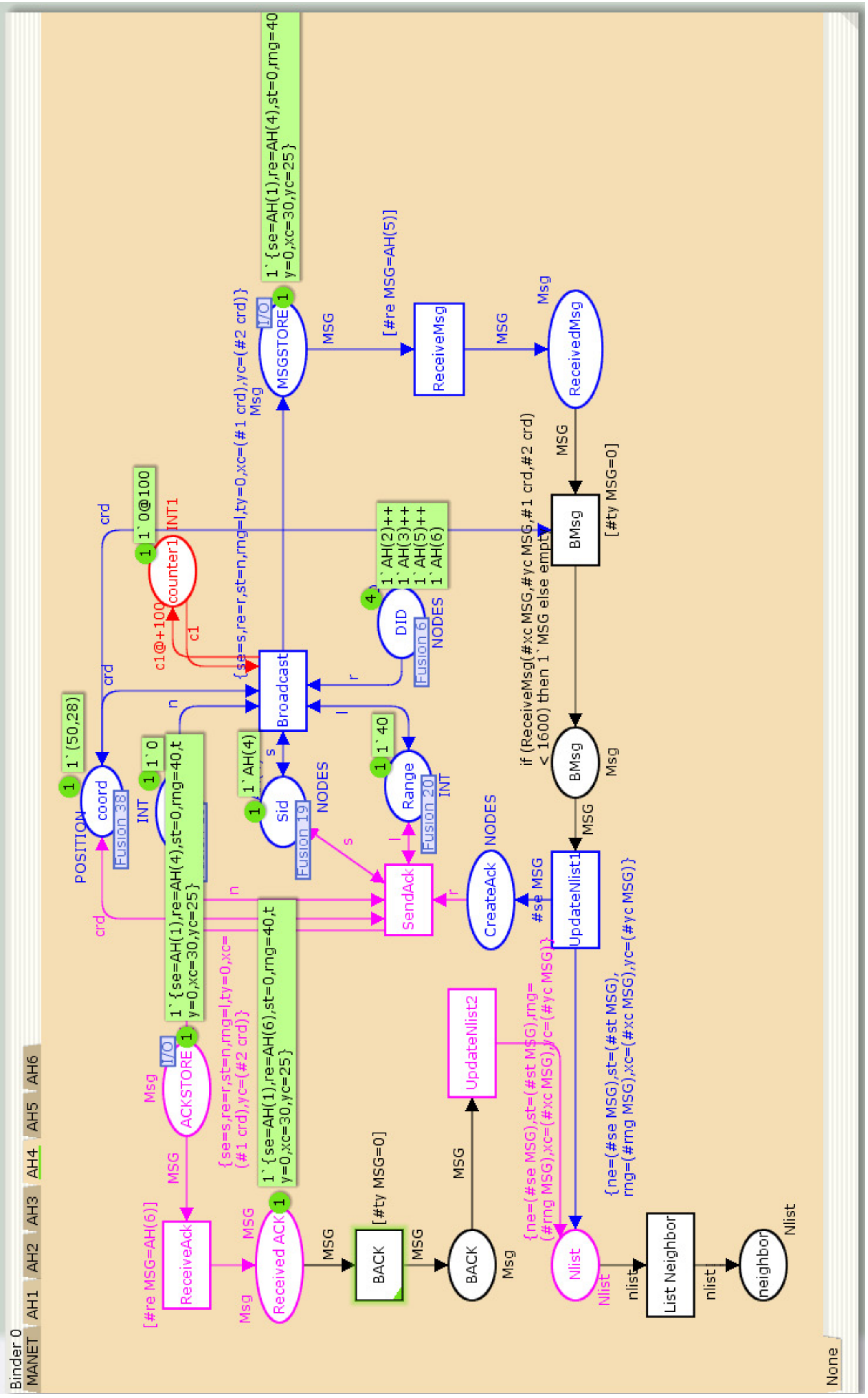


Figure 3.20: Marking M_{12}: With the consecutive occurrence of *ReceiveAck* and *SendAck*

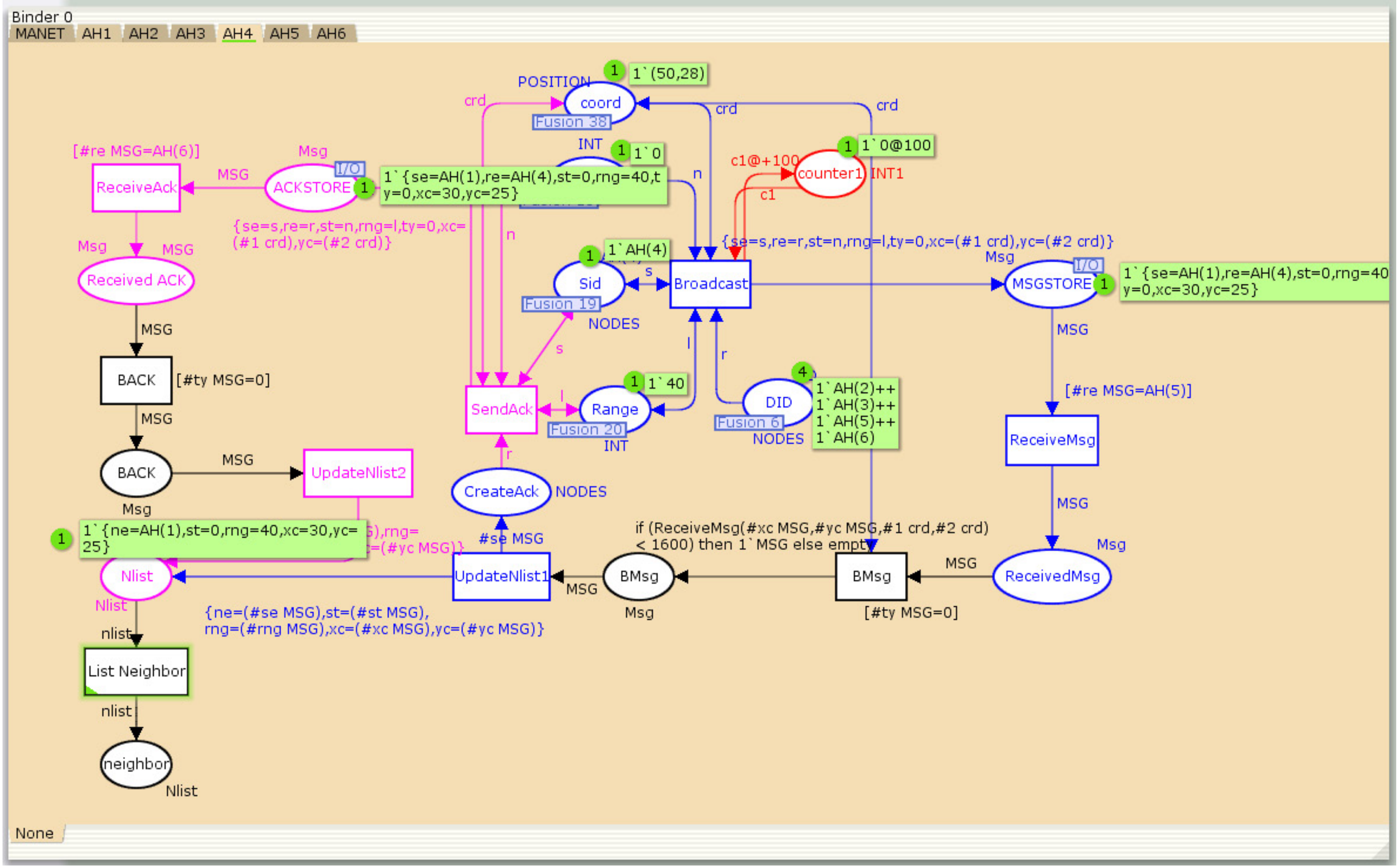


Figure 3.21: M_{13}: The state after the successful occurrence of transition *Back*

starts from the initial marking of the model M_0 to its final marking where every node is updated with its one-hop neighbours in the NTAB. Every state of the simulation is marked with a marking M_x indicating the flow of control through the different CPN components. The results of the simulation ensures that a node in the network is able to detect its one hop neighbours and establishes a bidirectional link with them by the exchange of neighbour request and acknowledgement messages.

However, it can be observed that the neighbours of a node are detected for a instance when the node is static. As the nodes move in the network, the connectivity among them changes requiring the NTAB to get updated. Thus, NDP can be called upon periodically or on-demand basis whenever the NTAB has to be updated.

Chapter 4

Proposed Clustering Algorithm

4.1 Introduction

Clustering in mobile ad hoc networks is the process of partitioning the nodes into logical groups. This can be mapped to the process of graph partitioning with some added constraints. Clustering results in selecting a set of vertices $H \subseteq V(G)$ called as the dominant set where every member of the dominant set H holds the responsibility of cluster head for the cluster. If $\Delta(h)$ is the degree of links associated with the head h, and $|h|$ is the cardinality of one-hop members of h, then $|h| \leq \Delta(h)$. This indicates that a node that is physically within the transmission range and has a link to the cluster head may not be a member of the cluster head. This gives an example of non-overlapping logical clusters. However, the union of the members of all the clusters results in the whole network [45]. Mathematically, if $h \epsilon H$ and $\Gamma(h)$ is the set of one-hop neighbors of h then $H \cup \Gamma(h) = G$.

This chapter proposes a topology adaptive distributed clustering algorithm, that targets to select minimum number of cluster heads to minimise the number of nodes in the virtual back bone. The algorithm uses the Neighbor Detection Protocol to learn its one-hop neighbors. This chapter also deals with the energy consumption of the different mobile nodes in various operating modes. The cluster maintenance strategies to support the node mobility as well as to retain the cluster control structure are also included in this chapter.

4.2 Basis of the Proposed Algorithm

The proposed topology adaptive clustering algorithm TACA is designed with the following features:

- The nodes in the ad hoc network are capable to increase or decrease their transmission range. However, the maximum permissible transmission range a node can posses is denoted as $Range_{MAX}$. The purpose of enabling the nodes to adjust their transmission range is to control the network topology even when the nodes move freely within it. However, during the initial phase of the topology adaptive clustering algorithm TACA, the range of all the nodes are kept equal.
- Out of several parameters of the nodes in the MANET, the node mobility is considered to be a major challenge. It changes the node connectivity very often. So a frequent topology change occurs in the network. The higher the rate of node movement, the greater is the frequency of topology changes. Similarly, the battery power of the light weight nodes are another major constraint. The development of techniques for energy resources are much slower than the network devices counterpart. Both of these parameters decide the stability of the cluster as well as the network. Hence, in the proposed algorithm these two factors are chosen as the weight deciding factors for the nodes.
- A cluster head selection procedure takes place when the network is first activated. The set of selected cluster heads are called the volunteer cluster heads.
- A volunteer cluster head serves its one-hop members till it exhausts its battery power beyond a threshold value. Then the head selects another

node within its cluster zone to act as a new head. The newly selected cluster head by the volunteer cluster head is called the non-volunteer cluster head.

- As a node drains its battery power completely, it becomes dead and is removed from the network. As a result, the topology of the network is disturbed. Hence, in order to use the node battery power efficiently, the nodes get almost fair chances of serving as cluster heads, so that load on individual nodes could be avoided.

4.3 Calculation of the Node Weight

The mobility of nodes changes the network topology frequently which in turn hampers the network stability. So, choosing the less mobile nodes for the formation of the virtual back bone is preferred. This ensures better backbone stability. Similarly the limited battery power devices consume their energy and become dead while routing the packets through them. This de-links the path for packet routing and demands for further establishment of routing back bone. In order to ensure the availability of routers in the routing backbone, nodes with more available battery power are chosen as the back bone forming nodes.

Keeping these factors in mind, the node weights are calculated by considering the node mobility and its available battery power as the key values. Here δ is assumed to be the maximum permissible speed of a node in the network. Thus the mobility factor of every node is calculated by computing the difference of δ and its average speed during a certain time interval. A larger mobility factor indicates a node with less mobility and vice versa. The available battery power is the energy associated with the node at the instant of weight calculation. These two parameters are added with different weight factors to find the individual node weights.

The steps for calculating the weights are described below:

Step 1: Let the total distance covered by a node v during last n time units is

$D_v = \sum_{i=t-n}^{i=t} dist_v$, where t is the current time. So, average speed of a node is computed as

$S_v = D_v/n$.

Step 2: Compute Mobility factor $\Delta M = \delta - S_v$. This indicates the difference of the average speed of the node from maximum permissible network speed δ.

Step 3: Compute available battery power as $P_{av} = P_{av} - P_{cons}$

where, P_{av} = Available battery power of the node.

P_{cons} = Battery power consumed by the node.

Step 4: Compute the weight of the node as

$$WT(v) = x_1 \Delta M + x_2 P_{av} \tag{4.1}$$

where x_1 and x_2 are the weight factors that are normalised so that $x_1 + x_2 = 1$. The weight factors indicate the major constraints of a network. For a highly mobile network, x_1 may be given a higher value where as for an energy constrained network x_2 may be given a higher value.

4.4 Selection of Volunteer Cluster Head

After the weight calculation of the nodes, the initial clustering algorithm is called upon to select the set of volunteer cluster heads. A pseudo-code segment of the algorithm is presented below.

```
Begin
...
for $(every\ v \in V)$
{
if $W_v > W_i$ where $i \in \Gamma(v)$
```

```
Then Set head= $v$
for ($every\ x \in \Gamma(v)$)
{
if STATUS(x) = 0 then
Set HEAD(x)= head
}
}
...
End
```

The algorithm indicates that a node having maximum weight among its 1-hop neighbors declares itself as the volunteer cluster head. And its 1-hop uncovered neighbors, (i.e., whose role is not yet decided) become the members of the volunteer head. The set of covered nodes are exempted from taking part in subsequent selection procedure and this process is repeated till all the nodes are assigned with their role either as a cluster head or a cluster member.

During the cluster head selection phase, every node broadcasts it's ID along with its weight WT_i to the neighbors and stores the weights WT_j that it hears from other nodes. It is understood that, every node has a neighbor table NTAB that stores the list of neighbors of the node which has been learnt by the execution of the NDP discussed in the last chapter. If a node does not hear another node with weight higher than its own weight, then it declares itself as a volunteer cluster head and its one-hop uncovered neighbor nodes become its members. In case of a tie in the node weights the lower ID node is preferred for the role of cluster head. Unlike the re-affiliation issues of DMAC algorithm discussed by the authors of [30], where a node resigns its current head after finding any higher weighted node within its proximity, in the current algorithm a member node remains affiliated with a cluster head, unless either of them go out of the range of each other or the head drains out its battery power. This reduces the number of re-affiliations lowering the cluster maintenance overhead.

The example of volunteer cluster head setup phase of the proposed algorithm is demonstrated with the help of the figures. In figure 4.1 every node is identified with a unique ID and its associated weight in parenthesis.

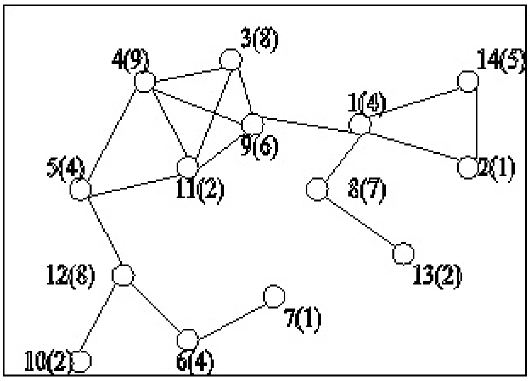


Figure 4.1: Initial topology of nodes after weight calculation

It is assumed that the weights are already computed for every node. The link between every pair of nodes denotes that they are within the transmission range of each other and establish a bidirectional link among them (i.e. one-hop neighbors).

Figure 4.2 shows the network as the volunteer cluster heads are identified as the solid circles after the exchange of their weights within the local topology.

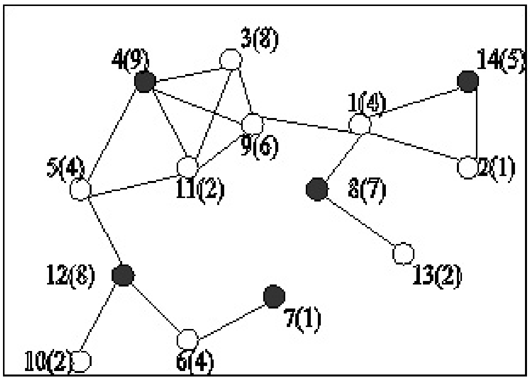


Figure 4.2: Initial cluster heads are identified as solid circles

A node having the highest weight among its 1-hop neighbors become the head and its immediate uncovered neighbors become its members. Figure 4.3 shows the network after the initial clusters are formed.

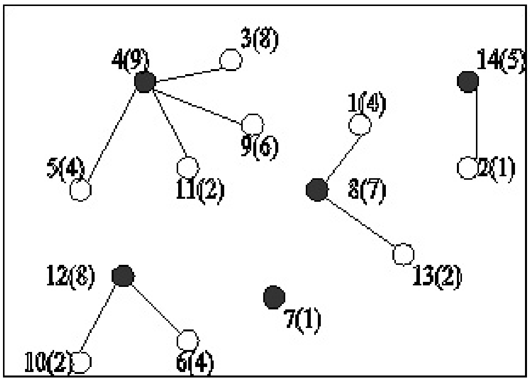


Figure 4.3: Clusters are formed

Every node in the network maintains its own cluster table. The data structure of the cluster table (CTAB) may be written as:

MID	MWT	MDIST	MT

MID : Member ID

MWT : Member Weight

MDIST : Member Distance

MT : Member Transmission Range

The cluster table is updated with the occurrence of node re-affiliation and reelection. It implies that, when a member of the current head goes out of its transmission range, it is deleted from the CTAB. And at the same time new row is added when a new node becomes the member of this cluster. It is shown in figure 4.3 that some cluster heads have as many as four members as in the case of head 4. Where as a head like 7 has zero members affiliated to it. In figure 4.2, it can be seen that node 6 has connectivity to both the nodes 12 and 7. But as it finds the head 12 within its range having higher weight than the head 7, it affiliates as a member to 12. This leaves node 7 as an isolated node without having any cluster member. Similar situation occurs with node 5 and

1, where they have affiliated to node 4 and 8, in spite of having connectivity to cluster heads 12 and 14 respectively. Hence, it is understood that a node always prefers to join a higher weighted node if it finds more than one head in its proximity.

4.5 Proposed Energy Consumption Model for the Ad Hoc Nodes

According to IEEE 802.11, the wireless network interface can operate in either base station mode (BS) or ad hoc mode [46]. In BS mode, every mobile host remains in the transmission range of one or more base station, which are responsible for buffering and forwarding traffic between hosts. Nodes who want to transmit can send outgoing traffic to the base station anytime and can receive the incoming traffic from the base station by polling it periodically. The rest of the time the node can enter into a non-operating sleep state, from which the interface must wake up in order to send or receive traffic. The guaranteed availability of fixed infrastructure like base station for buffering and traffic management supports energy conserving functionality by allowing some nodes to enter into the sleep state.

In contrary, ad hoc mode of operation does not use any base station. So a node communicates directly with one-hop reachable nodes and indirectly with unreachable nodes using dynamically computed routes. This demands the nodes to remain awake all the time to receive traffic from their neighbors and does not allow them to enter into sleep state. However, a node can enter into idle mode when it neither transmits nor receives any traffic. But it constantly listens to the wireless media and consumes energy which is almost same as the energy consumption in receiving traffic. Thus, energy consumption of the mobile devices depends on the operating mode of its wireless network interfaces. In the thesis the energy is assumed to be the battery power of the mobile node. So energy consumption of the node means its battery power consumption.

Thus, the wireless interfaces of the ad hoc network can be in any of the following operating modes:

- **Transmit:** for transmitting data
- **Receive:** for receiving data
- **Idle:** a default mode when the node is ready to transmit or receive.

A simple linear model can be considered for the energy consumption cost of mobile nodes for sending or receiving packets. The authors of [47] have presented a linear model for the per-packet energy cost that consists of an incremental cost m associated with the size of the packet and a fixed cost c that is associated with the channel acquisition to represent broadcast communication as

$$Energy = m_{send/receive} * size_{packet} + c_{broadcast} \tag{4.2}$$

In a broadcast traffic, the sender listens briefly to the channel. If the channel is found to be free then the packet is sent and is received by all nodes in wireless range. The destination node processes the packet where as the non-destination nodes ignore it. If the channel is found busy, the sender has no choice but to back off and retry later. In the current clustering algorithm, the cluster members send or receive the packets directly to/from their cluster head. Due to the broadcasting environment, when a cluster member transmits, its neighbor non-destination nodes (called the exposed terminals) over hear the CTS message and data packets. Being the non-destination nodes they consume some amount of energy in discarding these overhear packets. Similarly, the nodes in the range of the destination node (called the hidden terminals) overhear the RTS message and the data packets. These nodes also consume some energy in discarding the overhear packets. Thus, it can be concluded that, the total energy consumption for the cluster members include the energy consumption for packet communication and discarding the overhear packets for the nodes. The total energy consumption by the members for discarding the overhear packets is represented in equation (4.3) as [47]:

$$\begin{aligned} Energy_{discard} = &\sum_{n\in sender} c_{discardRTS} + \sum_{n\in dest} c_{discardCTS} + \\ &\sum_{n\in sender} (m_{discard} * size_{packet} + c_{discard}) + \sum_{n\in dest} c_{discardACK} \end{aligned} \tag{4.3}$$

In order to avoid the complexity of the model, the energy consumption for discarding the overhear control packets like CTS/RTS or ACK may be ignored. Further, in the worst case:

$$m_{discard} = m_{recv}, c_{discard} = c_{recv} \tag{4.4}$$

That is the non-destination nodes just receive the packets and ignore them.Now, the equation (4.3) can be re-written as:

$$Energy_{discard} = \sum_{n\in sender} (m_{recv} * size_{packet} + c_{recv}) \tag{4.5}$$

Combining the equation (4.2) and equation (4.5), we propose the final energy consumption model for the cluster members as:

$$Energy_{member} = m_{send/recv} * size_{packet} + c_{send} + \sum_{n\in\Gamma(neighbors)} (m_{recv} * size + c_{recv}) \tag{4.6}$$

where the first part of equation (4.6) explains the energy consumption in the actual transmission of the data and the second part explains the energy consumption in listening to the overhear packets.

The situation is little bit different for the cluster heads. In addition to the job of forwarding the packets to and from the cluster, it has an additional job of resource management for its members. Thus, the cost of consumption of energy is proportional to the number of member nodes served by the cluster head. Moreover, the radio range coverage by the head node has a considerable effect on its energy consumption. Depending on the RF environment the energy consumption can vary from P_v^2 to P_v^4, where p_v is the transmission power utilised by the head node in communicating a one-hop neighbor within its cluster [48]. As the distance between the nodes in a cluster is considered to be very small, so a linear relation between the transmission power and the energy consumption of the head node is assumed in this work. As a whole, the energy consumption of a cluster head may be considered to depend on the following parameters:

- No. of members served by the head
- The traffic forwarded by the head
- Total transmission power utilised by the head in serving the members

Thus considering the above three parameter,an energy consumption model for the cluster heads can be proposed as:

$$energy_{head} = f(energy_{|n_i|}, energy_{traffic}, energy_{transpower}) \tag{4.7}$$

Let the number of members served by the cluster head n_i is defined as its cardinality $|n_i|$. As the cardinality of a cluster head increases, its power consumption also increases. If the cluster head consumes one unit of battery power per member, then the total energy consumed by the cluster head for serving $|n_i|$ members is:

$$energy_{|n_i|} = |n_i| \tag{4.8}$$

The traffic handled by the cluster head indicates the volume of data that is transmitted or received by the head for its members. Thus, the energy consumed by the cluster head for forwarding the traffic can be proposed as:

$$Energy_{traffic} = m_{send} * \sum traffic_{send} + c_{send} + m_{recv} * \sum traffic_{recv} + c_{recv} \tag{4.9}$$

As discussed earlier, the energy consumption by the cluster head is function of the transmission power utilised by the head in communicating its one hop members. So the energy consumption due to the transmission power utilisation can be proposed as:

$$energy_{transpower} = energy_{unitdistance} * \sum_{i' \in C_i} Dist(i, i') \tag{4.10}$$

Thus, the equation for the energy consumption of the cluster head in equation (4.7) can be rewritten as:

$$energy_{head} = \alpha * energy_{|n_i|} + \beta * energy_{traffic} + \gamma * energy_{transpower} \tag{4.11}$$

Where α, β and γ are the weighing factors for the corresponding network parameters. These values are kept flexible so that they can be changed as per the network scenario. For example, when the network traffic is very high, β can be given more weightage than the other two. Similarly, in a dense network, where the cardinality of clusters are more, the weightage of α dominates the other factors. All three parameters are chosen so that $\alpha + \beta + \gamma = 1$.

4.6 Selection of Non-Volunteer Cluster Head

From the energy consumption model of the ad hoc nodes, it is understood that the cluster heads drain their battery power faster than the cluster members. Thus in order to have fair distribution of energy drainage among the nodes in the network, local selection for non-volunteer cluster heads takes place. When the current head (either the volunteer head or non-volunteer head) drains its battery power above a threshold value, it selects one of its own cluster members having the highest weight among others and sends an invitation for the role of cluster head. The selected node has the choice to accept or reject the invitation depending on its available resources. After accepting the invitation of the current head, the selected node becomes the new cluster head for that cluster. The existing head hands off its members to the new head that lie within the range of the later. The hand off that takes place is soft hand off (i.e. the resources allocated to the members remain unaltered). The nodes that do not remain within the range of the new head, try to join any other cluster head within their proximity. If the node does not find any head to do so, then it declares itself as an isolated volunteer cluster head. Finally, the current head affiliates as a member to the newly selected non-volunteer cluster head. Here, the selection process takes place locally within a cluster reducing the computation and communication overhead that would have yield in the global cluster head selection procedure. The pseudo-code segment of the algorithm for finding the non-volunteer head may be written as:

```
Begin
...
Set $i$ = current-head //volunteer or non-volunteer
Set max-wt= maximum ($WT_v$) where $v \in cluster_i$
Set next-head= $v_{max-wt}$
Head ($i$ ) = next-head
for (every $member \in cluster_i$ other than next-head )
{
if dist $(next - head, member) \leq next - head_{trange}$ then
{
Head $(member)$ = next-head //hand off
else if
Reaffiliate $member$ to other head within range
// Reaffiliation
else
Select $member$ as volunteer head //Reelection
}

   }
...
End
```

The sample of a non-volunteer cluster head selection is shown in figure 4.4. Initially, node 4 was selected as the cluster head with node 3,9,11 and 5 as its cluster members. Subsequently, the cluster head 4 selects node 3 as the non-volunteer cluster head as it has the highest weight among other cluster members. The existing members of 4, i.e. the node 9 and 11 are within the reach of the new non-volunteer cluster head. So these two nodes re-affiliate to head 3 as its cluster members. But the node 5 that was earlier a member of head 4, is not within the range of the new head 3. Assuming that node 5 is not even within the range of the head 12, becomes an isolated cluster head without any members. Finally, the former cluster head becomes a member to the new head 3.

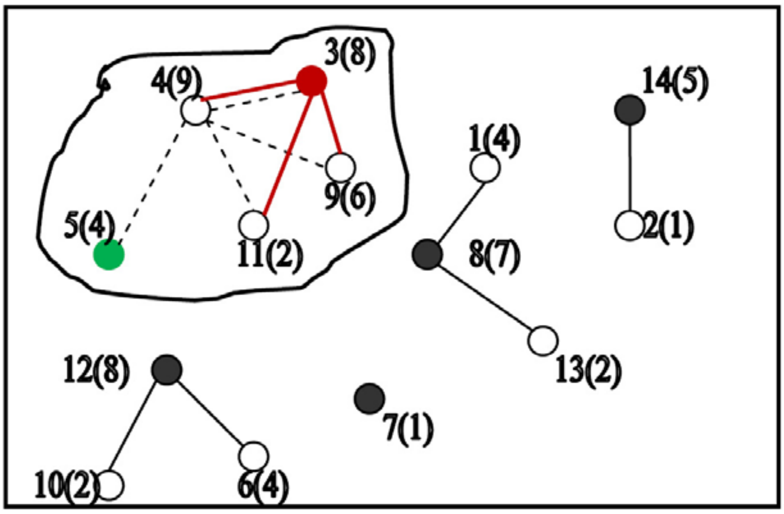


Figure 4.4: Non-volunteer cluster head selection

4.7 Cluster Maintenance

The process of clustering consists of two stages, i.e., cluster formation and cluster maintenance. The objective of clustering phase is to obtain minimum number of cluster heads with maximum cluster stability, where as the objective of cluster maintenance is to preserve as much as of the existing clustering structure as possible. In one-hop cluster, there exists bidirectional link between the cluster head and its members till both are within the transmission range of each other. As one of the mobile node moves away from the other, there occurs link failure between them. If a cluster member gets disconnected from its cluster head, then it searches for another new head within the close proximity to get affiliated. Such a changing situation of a member node from its current head to another head is called re-affiliation. For every single re-affiliation of a member node, both its previous and current cluster head update their member lists by deleting or inserting the details of this cluster member. This results in multiple messages to flow between the nodes and increase the message complexity as well as congestion of the network. Thus, minimising the frequency of re-affiliation becomes the objective of any efficient clustering algorithm.

Another overhead in maintaining the linked cluster architecture is the average number of cluster head updations per unit time. When any of the existing cluster heads become unable to serve the member nodes for any reason, an updation of cluster head set takes place. A lower cluster head updation frequency can lead to improved route stability.

After the initial cluster head election for the proposed algorithm a reelection of cluster head takes place when:

- A single node becomes orphan or isolated by moving away from other nodes and declares itself as a volunteer cluster head.
- An existing cluster head consumes its battery power beyond threshold value; so that it resigns from its current role and hands off its role to one of its member node with maximum available battery power. Such a forcibly elected node is called non-volunteer cluster head.
- the hand-off of cluster members takes place from the current head to the new non-volunteer cluster head.

Similarly, re-affiliation of cluster members occurs when:

- A member node leaves the transmission zone of its current head and enters into another cluster zone.
- A non-volunteer head is elected so that nodes of the existing head reaffiliates to the new head (if exists within the transmission range) or finds another head within its range.

4.8 Simulation Results and Discussion

The simulation of the proposed algorithm TACA is carried out in 100 X 100 grid area. The mobility model under consideration is the Random Walk mobility model. This mobility model represents the most erratic and unpredictable movement of a node [49,50]. Here, a mobile node (entity) moves from its current location by choosing a random speed between $(speed_{min}, speed_{max})$ and a random direction between $(0,\ 2\pi)$ respectively. In random walk model, when a mobile node reaches a simulation boundary it bounces back with an angle determined by the incoming direction. This is a memory-less mobility pattern as it retains no knowledge about its past direction and speed value [51]. This model can be used for vehicular and large scale environments. For the simulation of TACA, the speed of nodes is kept between 0 mt/sec to 5 mts/sec. During the node weight calculation, the total distance covered by the node for n time units are computed to find the average speed S_v. For the current simulation, n is taken as 5. Similarly, the maximum permissible speed δ for the network is considered to 5. The packet size is taken as 1024 bytes for the current work. Thus, the simulation parameters can be summarised as:

The battery power consumption by the mobile nodes for different operating modes are considered as per the IEEE 802.11 LUCENT WAVELAN cards as

Area	100 X 100 m
Mobility Model	Random Walk
Packet Size	1024 bytes
Max. Nodes	70
Trans. Range	5-40
$Speed_{min}$	0 m/sec
$Speed_{max}$	5 m/sec
δ	5
n	5

[46, 47]:

$$Broadcast_{send} = 1.9\mu W.s/byte * size_{packet} + 250\mu W.s \tag{4.12}$$

$$Broadcast_{recv} = 0.5\mu W.s/byte * size_{packet} + 56\mu W.s \tag{4.13}$$

$$idle = 808mW. \tag{4.14}$$

Figure 4.5 shows the comparison of results of TACA with that of LID and WBCA algorithms for the frequency of node re-affiliations.

The figure indicates that, the pattern of the results for all the algorithms are almost similar. That is the frequency of re-affiliation increases with respect to the transmission range for a certain period. Subsequently it decreases with the increase in the transmission range. The reason is that, higher transmission range of cluster heads enable them to accommodate the mobile members for a longer period of time. The results show that, the frequency of re-affiliation of node for TACA is lower than that of the other algorithms. As WBCA depends on the mean connectivity of the nodes for weight calculation, the change in the transmission range changes the connectivity of the nodes and so as the weights. This may lead to the frequent cluster head changes as well frequent re-affiliations to the heads. But TACA does not depend on the degree of node connectivity for selecting its cluster heads. So the re-affiliation overhead is lesser in comparison to the other algorithms.

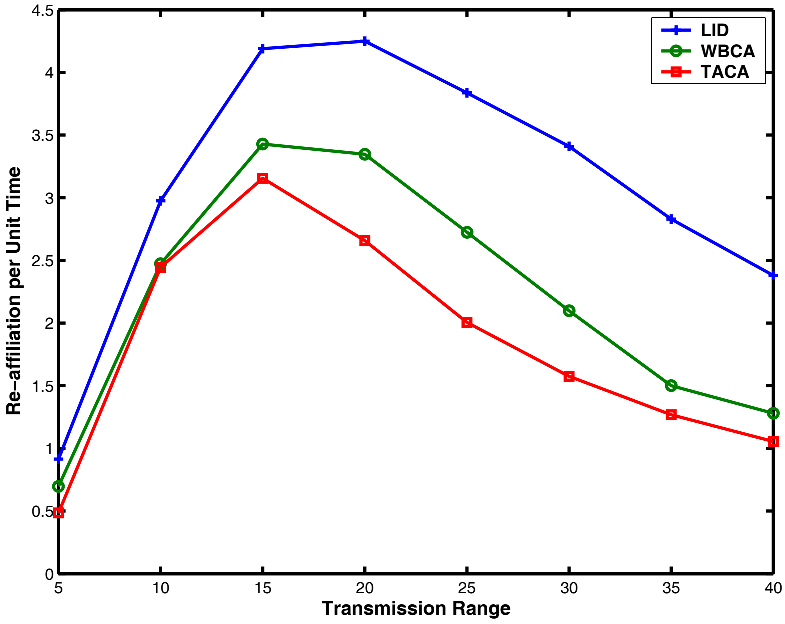


Figure 4.5: Comparison for Frequency of Re-affiliation

Figure 4.6 shows the comparison of the results of TACA with that of LID and WBCA for the frequency of cluster head re-election. The results for the same for both LID and WBCA algorithms have been discussed earlier in chapter 2. Here, it is seen that the frequency of re-election of cluster heads for TACA is much lesser than that of LID, but it remains almost similar to that of WBCA. Figure 4.7 shows the results for the average number of cluster heads for the three algorithms.

It is seen that the number of cluster heads are slightly higher than that of the other algorithms. The reason is the formation of extra isolated cluster heads during the hand-off of existing members of the current head to the new non-volunteer cluster head in TACA. However, the major strength of TACA is its ability to increase the network life time. In the current work the network life time is defined as the period that elapses between the network initiation time till the first node drains its energy completely and becomes dead. Selecting the non-volunteer cluster heads by the existing heads that drains its battery power beyond a threshold value, it saves its energy for its actual transmission.

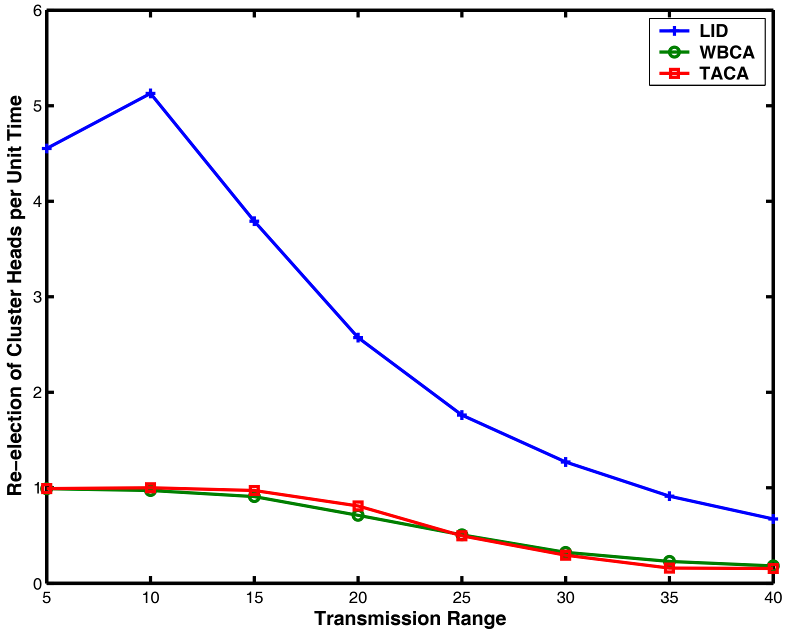


Figure 4.6: Comparison for the Frequency of Cluster Head Re-election

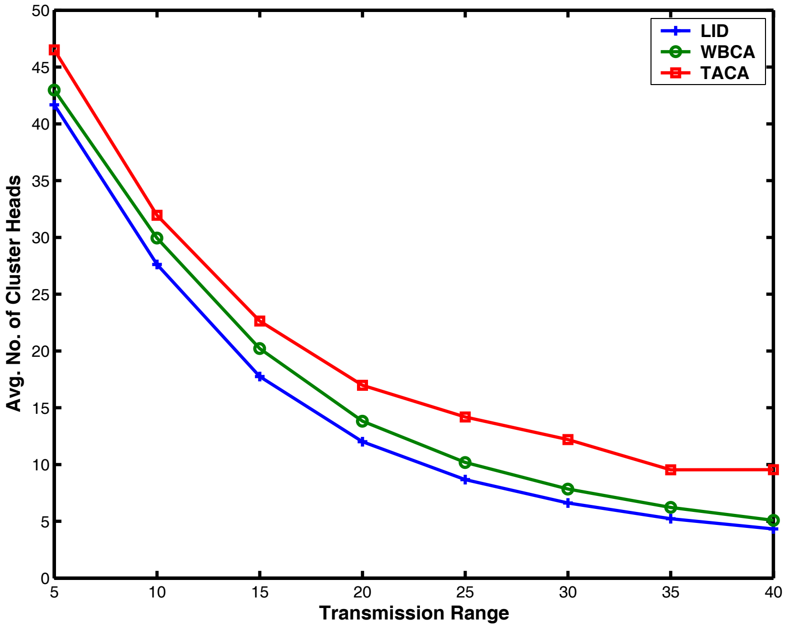


Figure 4.7: Avg. No. of Cluster Heads

In result, its life time is increased and ultimately the total network life time is also increased. This can be concluded that the consumption of individual node battery power is fairly distributed among the nodes so that none of the node is overloaded while acting as the cluster head. Most of the nodes are given oppertunity to act as the cluster head either as a volunteer one or as the non-volunteer one. This mechanism cumulatively increases the total network life time to a great extend which is clearly visible in the figure 4.8. The value is multifold in comparison to that of LID and WBCA.

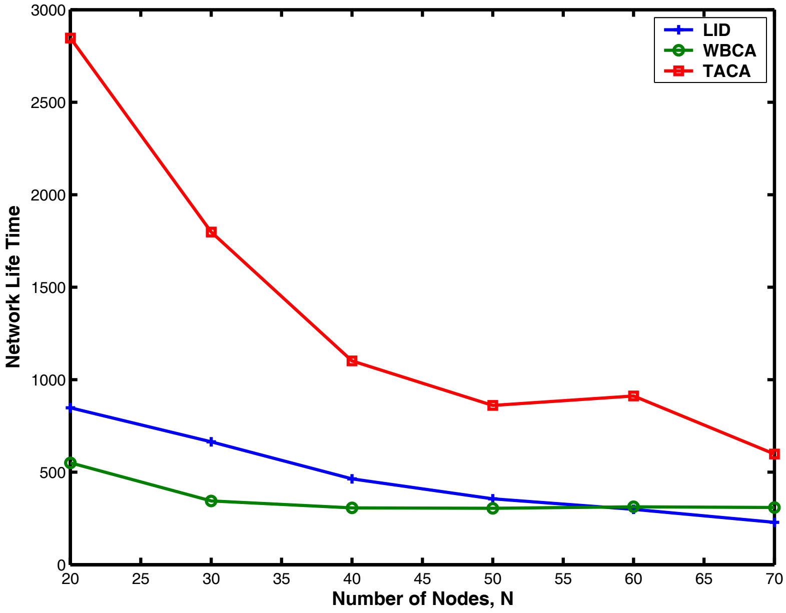


Figure 4.8: Network Life Time (in ms)

4.9 Validation of TACA through CPN

In this section, the validation for the cluster head selection procedure has been made by the well known CPN tools. Figure 4.9 gives the complete model of the TACA for cluster selection among the one-hop neighbors. The model is marked with the current markings on some of its places as indicated in the figure. This is an extension model of the NDP that was discussed in the last chapter. It can be seen in the figure 4.9 that, the upper part of the model remains unaltered as that of NDP. In the lower part of the model, two substitution transitions *select* *head* and *sendCACK* have been added. These two substitution transitions are linked to their respective subpages shown in figure 4.10 and 4.11. The detailed descriptions of the substitution transitions are as follows.

select head: The purpose of this transition is to select the cluster head by choosing the node having maximum weight among its one-hop neighbors. This transition has two input places *Nlist* and *volunteerhead* and two outputs *non-head* and *volunteerhead*. In the subpage of the substitution transition as shown in the figure 4.10, the input place *Nlist* provides the list of neighbors of a node. So this place is assigned with an *in-port* in the subpage. The place *volunteerhead* is initialised with a node selected as the cluster head. It has both input as well as output arc to the transition. The input arc carries the data from the place and compares the weight of this assumed cluster head to that of the node received from the *Nlist* place. If the weight of the neighbor node is higher than the currently assumed head, then the data is updated as indicated by the arc inscription from the transition to the place *volunteerhead*. The place *volunteerhead* is assigned with the *I/O* port as it has both input arc and output arc to the transition. The node having the lower weight than the other go to the *non-head* place as indicated by the arc inscription from transition to this *non-head* place. This place is assigned with the *out-port* in the subpage. Thus with the occurrence of the transition, place *volunteerhead* stores the details of the cluster head and the place *non-head* stores the details of the ordinary nodes. The tokens stored in both the places makes an enabled state for the transition *send to Msgstore* in the main page of the model. The output arc inscription of the transition *send to Msgstore* indicates the frame of the token to be passed to the place *MSGSTORE*. It can be seen that, the $\#ty$ field of the message has been set to 1. This implies that the cluster head has been selected. This message is now stored in the common place *MSGSTORE*. The occurrence of transition *ReceiveMsg* and the subsequent occurrence of transition *ClstMsg* adds a token having the $\#re = AH(1)$ and $\#ty = 1$ in the place *ClstMsg* as indicated by the occurrence conditions of the respective

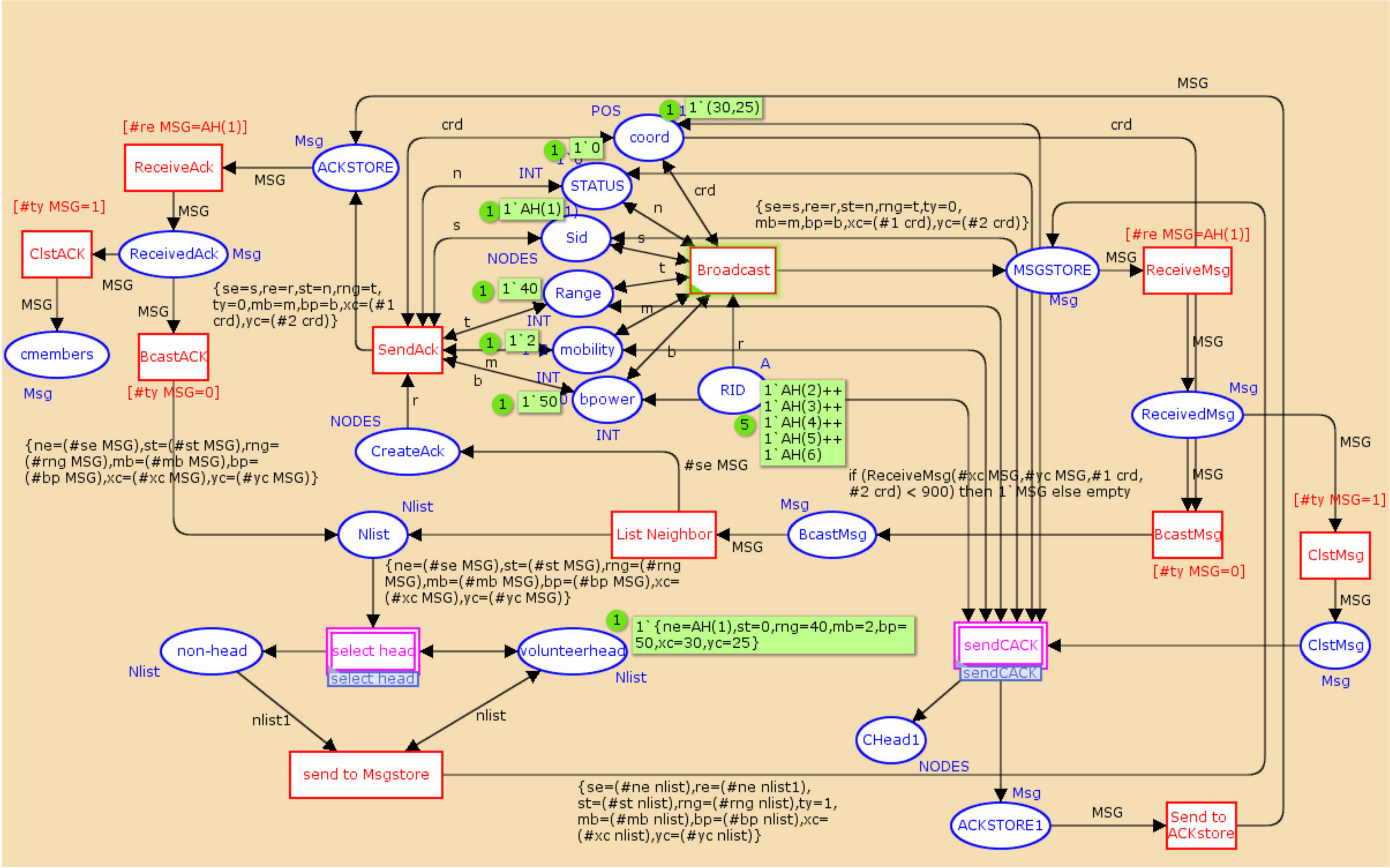
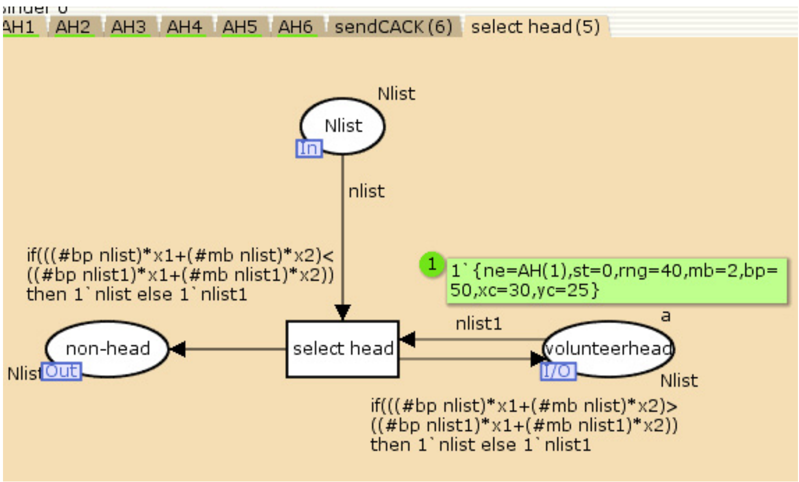


Figure 4.9: Model for validation of TACA



```
if(((#bp nlist)*x1+(#mb nlist)*x2)<
((#bp nlist1)*x1+(#mb nlist1)*x2))
then 1`nlist else 1`nlist1
```

```
1`{ne=AH(1),st=0,rng=40,mb=2,bp=
50,xc=30,yc=25}
```

```
if(((#bp nlist)*x1+(#mb nlist)*x2)>
((#bp nlist1)*x1+(#mb nlist1)*x2))
then 1`nlist else 1`nlist1
```

Figure 4.10: Subpage that finds the volunteer cluster head

transitions. Thus, now the place *ClstMsg* contains a message from the cluster head.

sendCACK: This substitution transition has a input place *ClstMsg* and two output places *CHead1* and *ACKSTORE1* as shown in figure 4.11. These places are assigned with *In-port* and *out-port* in the subpage as shown in the figure. In addition to this, it has multiple inputs from the places *coord, STATUS, Range, Sid, mobility* and *bpower* to enable the transition to occur. These places are assigned with the *I/O ports* in the subpage. The output arc inscription *#seMSG* from the transition *sendCACK* to the place *CHead1* indicates that the cluster head ID is stored in the place *CHead1*. The output arc inscription from transition *sendCACK* to the place *ACKSTORE1* indicates that an acknowledgement is created to be stored in the place. In the main page the availability of tokens in the place *ACKSTORE1* makes an enabled state for the transition *Send to ACKstore*. The occurrence of this transition stores the acknowledgement in the common place *ACKSTORE* as indicated by the output arc. The occurrence of transition *ReceiveAck* and the subsequent occurrence of transition *ClstACK* adds a token having the $\#re = AH(1)$ and $\#ty = 1$ in the place *cmembers* as indicated by the occurrence conditions of the respective transitions.

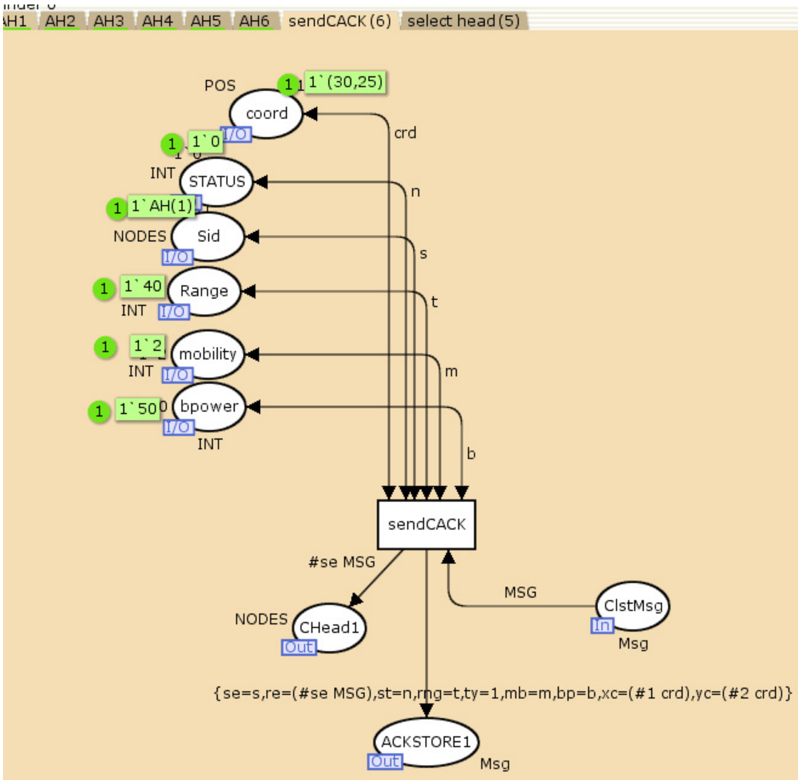


Figure 4.11: Subpage that send the acknowledgement

4.10 Summary

A topology adaptive clustering algorithm has been proposed in this chapter. The algorithm is distributed in nature that considers only the local topology to select a cluster head. That is a node having highest weight among its one-hop neighbor is selected as the cluster head. The weight of a node is calculated by considering its average mobility and its available battery power. This enables a less mobile node with more available battery power to get the chance to become a cluster head resulting in better cluster stability. Further, to reduce the load of cluster head on a single node, non-volunteer cluster heads are selected locally by the existing heads so that the node battery power is saved to some extend. The energy consumption by the nodes are considered for their different operating modes and accordingly, the total consumption by a cluster head or by a cluster member is calculated. It has been studied in the result that, during the hand off process of cluster members, some more isolated cluster heads are formed increasing the number of members in the virtual back bone. However, the network life time has been improved to a considerable value in comparison to the existing algorithms [20, 35] which is the major objective of any clustering algorithm. The isolated cluster heads that have been generated by this algorithm can be reduced by allowing those heads to affiliate to any nearby cluster head. This could be made possible by the design of any topology control algorithm. Finally, the model of the cluster head selection process of TACA has been validated by using the CPN tools, to ensure that the flow of the data and control among the nodes are in right sequence and meet the desired objective of the system.

Chapter 5

Topology Control Protocol for Clustered Ad Hoc Networks

5.1 Introduction

Design of topology control protocols deal with the algorithms, where the mobile radios are allowed to adjust their transmission ranges so as to retain the desired topological property of the network. That is the connectedness of the nodes are maintained with optimum energy consumption. Usually the transmission range of the nodes are much smaller than the span of such networks. For the purpose of packet forwarding, a multi hop network is created where every node performs the role of router for other nodes. But due to the node mobility, the wireless link between the nodes are frequently disrupted. The objective of topology control is to allow the wireless transceivers to increase or decrease their transmission ranges to an optimal value so that the network remains connected. Such minimum value of the transceiver's transmission range that enable to maintain network connectivity is called the ***critical transmission range*** (CTR). A perfect topology control enables the major number of nodes to stay connected in the network so that the end to end delay in packet routing is reduced. Hence a more specific definition of topology control is given by the author of [52] as:

"*Topology control is defined as the art of coordinating node's decisions regarding their transmitting ranges, in order to generate a network with the desired properties of connectivity along with reduced energy consumption and/or network capacity.*" The diverse approaches to the topology control problem may be organised into a coherent taxonomy as in figure 5.1.

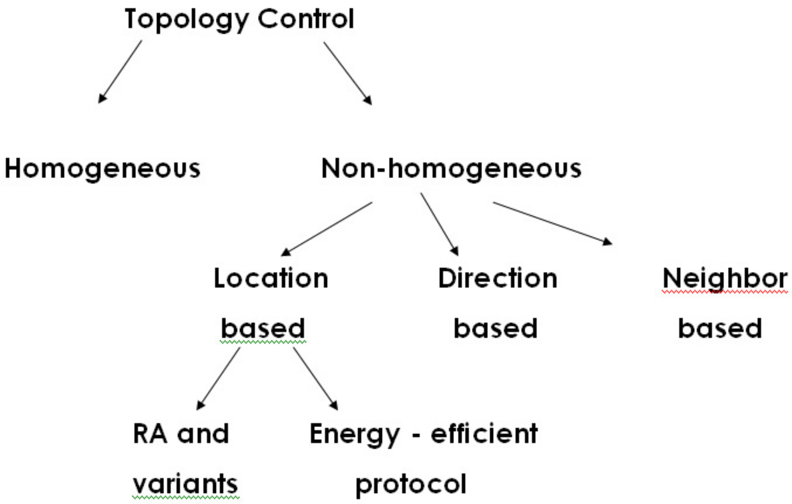


Figure 5.1: Taxonomy of Topology Control Protocols

The homogeneous topology control is the simplest kind, where every node in the network posses equal transmission ranges. Here the topology control problem can be visualised as the problem of obtaining the value of critical transmission range so that the network wide connectivity is maintained.

The non-homogeneous topology control allows the nodes to choose different transmission ranges within a maximum limit depending on the type of information they use to compute the topology. In this catagory, the *location based* approach assumes that most of the nodes are equipped with GPS receivers, so that their exact node locations can be easily obtained. This information is either used centrally or in a distributed manner. The use of a central authority is similar to the Range Assignment problem. Where as the distributed way of information exchange is used for building energy-efficient topology. The *direc-*

tion based approach helps the nodes to estimate the relative direction of their neighbors while they don't have the knowledge of their node positions. But it has been studied that the *neighbor based* approach is most suitable for mobile ad hoc networks, where the nodes have very little neighbor node information with them.

5.2 Essential features of Topology Control Protocol

Having no central infrastructure in an ad hoc networks, the design of fully distributed and asynchronous topology control protocol seems to be more realistic and adaptable. In a distributed topology control protocol, every node in the network exchanges its available information with other nodes. This information exchange could be made locally or globally. That is the information exchange by a node can be among its immediate neighbors(local exchange) or it could be a network wide exchange among every node in the network(global exchange). The global information exchange consumes large amount of time by the nodes to have the knowledge of the network wide topology. The condition is worse if the nodes are mobile. Because in a mobile ad hoc network, the frequent topology change needs a faster topology control protocol. Otherwise the stale route information available with the nodes may cause further link failure while routing the packets. Thus local information exchange may result faster topology reconfiguration with large networks even when the addition and deletion of nodes take place in the network.

Communication between two nodes is never possible without establishing a bidirectional link between them. This has to be considered while designing the topology control protocol. This is equally important for preserving the network connectivity during the reconfiguration of the topology. So, another essential feature of topology control protocol is that it should result in generating bidirectional links between the nodes while preserving the basic requirement of node connectivity in the network.The quality of information exchanged by the nodes plays a vital role in designing the topology control protocol. Some of the high quality information like the accurate node position needs additional hardware like GPS which may be expensive and in certain cases may not be applicable (indoor environment).So it is preferred to exchange the low quality information like node identity and its transmission power between the nodes.

The motivation behind the topology control is to reduce the transmission interference between the nodes. This can be achieved by reducing the degree of connectivity of the nodes. Thus in a nutshell, a topology control protocol should have the following features:

- Fully distributed and asynchronous
- Principle of locality
- Setting up of bidirectional link as well as preserving the network connectivity
- Trust on low-quality information
- lesser degree of connectivity

5.3 Effect of Mobility on Topology Control Protocol

Node mobility is a prominent feature in mobile ad hoc network that can not be ignored.

An ad hoc network can range from a highly mobile network(i.e. a vehicular network) to a very slow mobile network(i.e. sensors fitted to the body of pedestrians to study the traffic or sensors fitted to the body of insects to study their movements) . The design of a distributed topology control protocol with the local information is a vital issue during the node mobility. It is quite understood that the centralised approaches as well as the solutions, that require the exchange of global information are impractical when node mobility is moderate to high, unless otherwise the network is extremely small. As the propagation of network wide information needs a lot of time, so the construction of communication topology using global information may use only the stale information for building the network topology. This results in frequent link failure of the nodes as well as execution of route discovery procedures and the re-execution of topology control algorithms. Hence most of the bandwidth consumption is for transmission of control packets that are used to update the network topology and the routing information. In the worst scenario, the topology is never established. The authors of [52] indicate that the protocol used to build the network topology must be fast enough to adopt the node mobility. For this, the protocol should exchange less information with the neighbor nodes as the same with distance nodes may consume considerable amount of time. At the same time, a simple algorithm should be built up to compute the neighbor set with the information exchanged.

The issue of preserving the network connectivity and setting up a bidirectional link between the nodes for communication has drawn much attention by the researchers. In a mobile ad hoc network with frequent change in the topology, preserving the worst case connectivity is almost impossible. Even if a smarter and efficient protocol that could built a connected network topology at time t_0 may get disconnected at time $t_0 + \theta$ by the slightest movement of nodes with the effect of mobility.

Further the topology control protocol can not be executed very frequently to avoid the large control message overhead. So it is expected that the designed protocol should be able to handle the node mobility in such a way that a vast majority of the nodes remain connected during the operation time. The same concept is applicable for the degree of connectivity of mobile nodes. This parameter may change so fast during node mobility that, maintaining the lower degree of connectivity may be impossible in certain scenarios. However, setting up a bidirectional link between the mobile nodes is possible by exchange of few localised information.

How far the quality of information is concerned in the network topology control, the MANET should use the information that remain unaffected with the node mobility. For an example, let us consider two network nodes p and q that are moving with certain velocities $(v_p\ ,\ v_q) > 0$. When the various non-homogeneous topology control approaches are considered for these two nodes, it is clear that since the nodes are moving, their location information changes continuously over time. Thus the location based topology control approach is not suitable for controlling the network topology. However, if the nodes are moving in the same direction, their relative direction does not change. So the direction based topology control approach may be suitable for such network. Further, if the nodes are moving in such a way that their relative distance does not change too much, then the neighbor based information can be used for topology control. Thus it is concluded that using relatively less accurate information is preferable to build a topology control protocol when the mobility of the nodes can not be ignored.

Summarising the above, a topology control protocol for mobile networks should

- Be fully distributed and asynchronous.
- Be fast when the node mobility is high. That is the protocol should exchange few messages with neighbors.
- Generate a network topology in which most of the nodes are connected for most of the network operational time, and have a relatively small physical degree.
- Rely on information that is relatively resilient to node mobility.

5.4 State-of-art on Topology Control

There exists some topology control algorithms, some of which are suitable for static ad hoc nodes and some are suitable for mobile ad hoc nodes. Hou et. al. have tried to develop a model for analysing the throughput and forward progress where each station in the network has variable transmission ranges [53]. Here the authors have analysed three transmission strategies without considering the backward communication.

The three transmission strategies are **Most Forward with Fixed Radius R (MFR), Nearest with Forward Progress(NFP) and Most Forward with Variable Radius(MVR)**. MFR transmits to a neighbor with the largest forward progress by using a transmission radius **R** irrespective of the position of the receiver node. Here the main focus is to minimise the number of hops needed for a packet to reach its destination. However, the drawback is that the larger transmission radius **R** invites more transmission interference by degrading the throughput of transmission.

The NFP technique allows a node to transmit to the nearest neighbor so that a forward progress is obtained. Here the main target is to minimise the conflicts as much as possible. But it results with higher number of intermediate hops for end to end communication.

In the MVR strategy the transmission radius is adjusted to the value so that it is equal to the distance between the transmitter and receiver. This aims to obtain the largest possible progress with minimum conflicts. The authors of [53] have shown that NFP provides better throughput than the other two strategies. NFP achieves better stability with higher node density and is suitable for mobile radio network where the topology changes very frequently.

Rodoplu et. al. have proposed location based distributed topology control protocol [54] that aims to design an energy efficient network topology from the set of stationary nodes to a master node. The protocol has basically two phases. In the first phase, a local search is made by every node in which it broadcasts its position information locally to every node in its transmission zone. The process is repeated till every node builds up a graph of its strongly connected neighbors. The first phase constitutes a link setup and configuration phase. The basic assumption here is that, the nodes should be equipped with low power GPS receivers. This assumption makes the implementation expensive and sometimes even it is impossible to install the GPS receivers in all the nodes.

The second phase is a cost distribution phase, that finds the optimal link on the above graph. The Bellman-Ford Shortest Path algorithm is applied on the graph by considering the power consumption as the cost metric. This phase results in finding the minimum energy consumption path from a node to the master node. When the protocol is simulated over static as well as dynamic nodes, a significant energy saving is achieved. But considering the same protocol to a mobile ad hoc network with no fixed infrastructure to act as the master node is impossible. Further, the protocol aims to minimise the power consumption while transmitting to a single master node only. This may not be very effective in a multi-hop environment where an over all power saving is required while the nodes communicate among each other to self-configure the network.

Li et al. have proposed direction based topology control protocol named as the Cone Based Topology Control (CBTC) that was extended to support the node mobility [55]. The authors used a neighbor discovery beaconing protocol for reconfiguration of the network. The beaconing protocol was used by every node to tell the other nodes that it is still active. The beacon includes the sender's ID and transmit power. If beacons from a certain neighbor v are not received for a time period τ , then the node v is considered to be out of reach by the other nodes. The authors discuss that if the network topology ever stabilises, then the reconfiguration algorithm builds graph that preserves the connectivity of the final network, as long as the periodic beaconing is guaranteed.

The neighbor based KNEIGH protocol was proposed by Blough et al. for static ad hoc networks [56]. This simple protocol depends on relatively low quality information such as the distance between the nodes for the purpose of topology control. As per the principle of the protocol, when a node u receives a message from node v, u is able to estimate the distance to node v by considering the strength of the radio signal or by calculating the time of the arrival of the signals. Initially every node broadcasts its ID at maximum power. Upon receiving the same from other nodes, every node keeps track of its neighbors and computes their respective distances. Then, every node computes its K-closest neighbors list KN, and broadcasts this information at maximum power. By the exchange of the neighbor list KN, the nodes can include their symmetric neighbors (two nodes are symmetric neighbors if and only if they appear in each others KN list) in the final topology. Finally, the transmit power of a node u is set to the minimum value needed to reach the farthest node in its list KN(u). However, it has been studied that, KNIEGH does not preserve network connectivity in the worst case.

P. Santi et. al. have made a probabilistic analysis for the range assignment problem in ad hoc networks [57]. Assuming equal transmitting range for every node in the network, the authors have tried to find the minimum transmission range r , so that the network remains strongly connected. It was found that as compared to the deterministic case, the probabilistic solution to the range assignment problem achieved substantial energy savings. However, the authors have considered a static model of the nodes that does not consider the node mobility.

Similarly, Sanches et. al. have designed a centralised algorithm to find the critical transmission range so that the network partition is avoided and the nodes remain connected in the network [58]. However, implementation of any centralised algorithm enhances the message and computational complexity in mobile ad hoc networks.Hence, a distributed algorithm is always preferred.

Paul et. al. have proposed a transmission range control protocol (TRCP) for obtaining stable routing between the sender and the receiver in a mobile ad hoc networks [59]. The movement of intermediate nodes in a multi-hop communication reduces the network throughput. Thus protecting the neighborhood relation was made by adjusting the transmission range of the mobile nodes. TRCP does the job of neighborhood establishment, neighborhood retention and neighborhood denial. The nodes increase their transmission range so as to register maximum six nodes as its neighbors. When more than six nodes register to the concerned node, it de-registers the nodes those are far away to save the energy consumption. However, in a mobile ad hoc network where the node density in any location is highly non-predictive, we can not restrict the number of neighbors to a fixed number.

Ramanathan et. al. have proposed an energy efficient distributed heuristics called Local Information No Topology (LINT) protocol explicitly for the mobile networks [60]. Probably it is the simplest heuristic where the nodes adjust their transmission power to preserve the network connectivity even with the changing topology.The authors claim the heuristic to be zero overhead protocol for not using any special control messages for its operation.

LINT is a neighbor based protocol. It uses locally available neighbor information to limit the degree of connectivity of a node. The neighbor information is generally obtained by neighbor discovery phase of any routing algorithm.LINT basically considers three parameters of a node. They are

- Desired node degree d_d
- High threshold node degree d_h
- Low threshold node degree d_l

Every node makes a periodical check of the number of active nodes that is obtained from the routing technique. If the degree of connectivity is found to be more than d_h then, the node reduces its transmission power. Similarly, if the degree of connectivity is found to be less than d_l then, the node enhances its transmission power. And no change in power level occurs if the degree of connectivity remains within the limit. However, the maximum and minimum power level is set within its limit. The drawback associated with LINT protocol is deciding the number of neighbors to the network traffic. In case of a low traffic, the routing protocol might provide little or stale information about neighbors to LINT resulting in incorrect power settings. Thus LINT is more appropriate when regular exchange of messages take place.

In contrast to LINT , LILT exploits the global topology information obtained by some routing algorithms like Link-State algorithm. It comprises of two parts named as **The neighbor reduction protocol** and **The neighbor addition protocol**. This does the same work as that of LINT protocol. However, the later has a very special function like it may override the high threshold bound and may increase the transmission power of a node when the link state routing update results in an undesirable network topology. The authors of both the protocols have noticed that LINT is more effective than LILT in increasing the throughput. With high node densities, the link state data base used by LILT to obtain network connectivity information is often stale causing unnecessary transmission power increases. This confirms that using global information to set up the topology in mobile networks is impracticable.

Some of the existing protocols on topology control for mobile ad hoc networks has been discussed here. Most of the protocols are designed for the static nodes, that does not support mobility. Some topology control protocols, exchange the node information locally or globally among the nodes to support the mobility. However, it has been studied that, local information exchange eliminates the problem of stale routing route that usually occurs in the protocols using global information exchange. Moreover, the information that are used by the nodes are less accurate and of low quality. Because obtaining more accurate and high quality node information adds up extra communication and computation over head into the protocol. Some protocols target to achieve strongly connected network, while some target to design an energy efficient protocol. All these factors have provided a motivation for the design of a transmission range adjustment protocol that can be used to enhance the performance of Topology Adaptive Clustering Algorithm in the MANET.

5.5 Proposed Range Adjustment Protocol

In this section, a transmission range adjustment protocol (TRAP) has been proposed which provides the mechanism to increase or decrease the transmission range of the mobile nodes enabling them to remain connected with others as and when required. The proposed distributed clustering algorithm TACA selects non-volunteer cluster heads that leads to the creation of isolated cluster heads and/or orphan nodes during the hand off process. The excess number of cluster heads increase the number of nodes in the virtual back bone of the communication network. Thus the objective of the proposed algorithm is to reduce the number of nodes in the virtual back bone. This is made possible by allowing the isolated cluster heads formed during the execution of TACA to adjust their transmission ranges. It helps them to get affiliated to other nearby cluster heads in stead of becoming isolated heads without having any cluster members. In the figure 4.3, node 7 is an isolated node without having any cluster member. The TRAP enables such nodes to adjust their transmission ranges so that they become the cluster members of other heads reducing the number of nodes in the virtual back bone.

The TRAP is called upon when the isolated cluster heads are formed during

- The initial volunteer cluster head selection
- The hand off of cluster members by the existing volunteer / non-volunteer heads to the newly selected cluster heads.
- The node movement to an area which is out of the transmission range of the existing heads.

To understand the Transmission Range Adjustment Protocol (TRAP), let us assume that node u is an isolated node which is unreachable to all the cluster heads. To get connected to any nearby cluster head, let it increases its original transmission range R by ΔR. Thus its new transmission range is now $R + \Delta R$. With this new transmission range of u, the following steps are implemented for the TRAP.

1. Node u broadcasts a neighbor detection packet (NDPAK) to the network mentioning its new range.

2. Let the packet is received by another node(s). Below are mentioned the different scenarios:

   Case 1: A single node v which is a cluster member receives the NDPAK sent by u.

   Result: Node v understands the increased transmission range of the sender u. It does not react to the NDPAK and thus does not send back the neighbor acknowledgement NAC packet. Because it is a cluster member only, it does not have the privilege to affiliate a node as its member.

   Case 2: A single node v which is a cluster head receives the NDPAK sent by u.

   Result:

   - Node v checks for the possibility to increase the transmission range. If it finds the possibility to do so, by checking its current value of the range and the number of its current cluster members, then it increases its range from its current value R to $R + \Delta R$. Then after, it sends a NAC to u so that the connectivity with u is established.

   - Node u includes v as a neighbor in its neighbor table NTAB and sends a neighbor confirmation NC packet to v.

   - Node v sets its transmission range to $R + \Delta R$.

- Node u joins as a cluster member to v and v updates its Cluster Table ($CTAB$) accordingly.
- Node v continues with this updated transmission range till it serves as the cluster head after which it again resets its range to the original value of R.

Case 3: Two cluster heads v_1 and v_2 receive the NDPAK sent by node u.
Result:

- Both v_1 and v_2 increases their transmission ranges from R to $R+\Delta R$ and send back the NAC to u.
- Let Node u finds the cluster head v_1 with higher weight than the other. So it chooses v_1 as the neighbor and sends back the neighbor confirmation NC packet to it. u ignores the acknowledgement of the other head v_2.
- The head that receives the NC packet (i.e. v_1) sets its transmission range with the new value $R + \Delta R$ and the cluster table CTAB accordingly. Head v_1 continues with the updated transmission range till it serves as the cluster head after which it again it resets range to the value of R.
- The head that does not receive the NC packet (i.e. v_2) within the time-out period, continues with its original range of value R.

Case 4: None of the nodes receive the NDPAK packet.
Result:

- Node u waits for time out period to get NAC from one or more cluster head. As it fails to get any, it further increases the range by ΔR. Thus the new range is now $R + 2\Delta R$.

- Step 1 and 2 are repeated till the transmission range of u reaches its maximum value $Range_{MAX}$ or node u gets affiliated with an existing cluster head, whichever occurs earlier.

3. Node u declares itself as an isolated head.

5.6 Result Discussion

The simulation for the TRAP is carried out on the same simulation bed as that of for the clustering algorithm TACA. The simulation parameters remain unchanged except in the variation of the transmission range of the nodes. The simulation results for this protocol indicates the improvement over the proposed TACA. Figure 5.2 shows the results for the frequency of node re-affiliations when the TRAP is implemented on the clustering algorithm TACA. The reduction in frequency of re-affiliation is possible because of the range enhancement done by some nodes. The enhanced transmission range of cluster heads enable the corresponding cluster members to stay connected for more

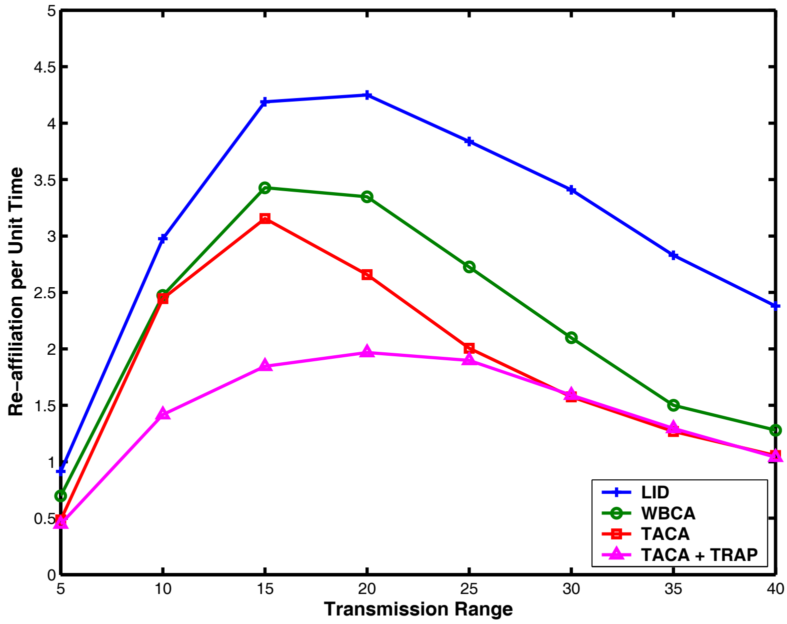


Figure 5.2: Frequency of re-affiliation when TRAP is executed

time even though they move in the network. It is clear from the figure that, the proposed topology control algorithm reduces the re-affiliation overhead even for small transmission ranges of the nodes meeting the requirements of an efficient clustering algorithm.

Figure 5.3 shows the results for the frequency of cluster head re-elections in the network. It is understood that, when TRAP is executed on TACA, the

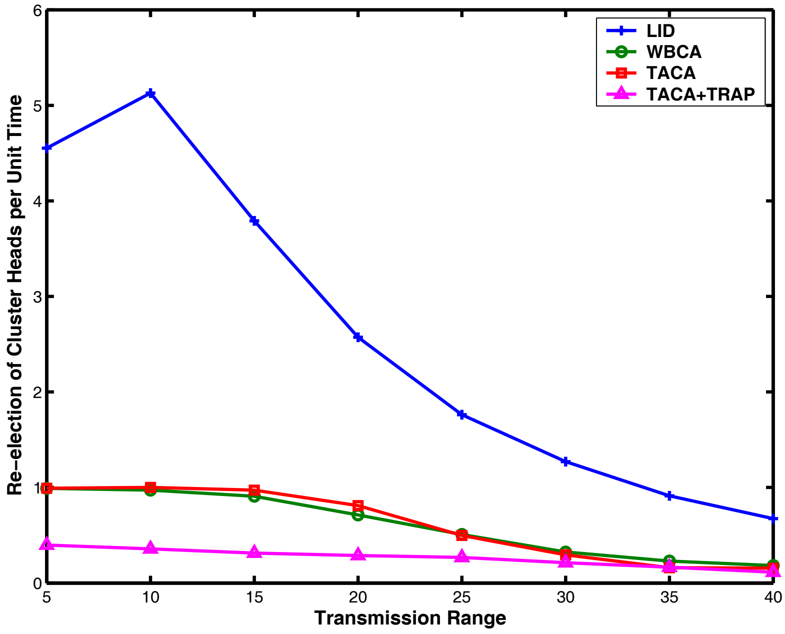


Figure 5.3: Frequency of cluster head re-election when TRAP is executed

transmission range of some of the nodes are increased. This allows most of the nodes to stay connected within the wider range and do not move out of the range of each other. Thus the formation of isolated cluster heads are reduced as happened in TACA. So finally, the re-election overhead is also reduced in the network.

The objective of any communication protocol is to reduce the end-to-end delay in communication. In a clustered mobile ad hoc network, the communication takes place by the virtual back bone that is formed by the selected cluster heads. Thus reduction in the number of hops in the virtual back bone may reduce the end-to-end delay in communication. Figure 5.4 shows that the

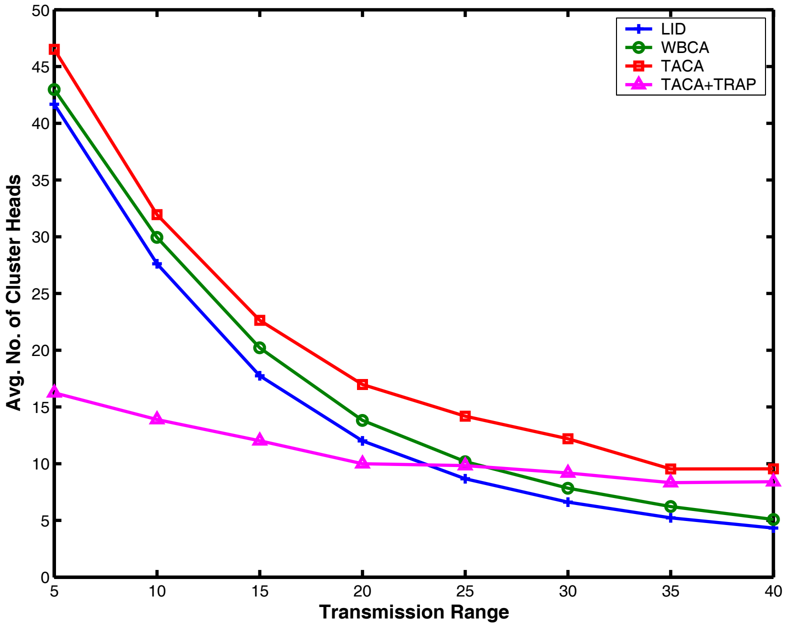


Figure 5.4: Avg. No. of Cluster Heads when TRAP is executed

proposed clustering algorithm TACA results in more number of cluster heads in comparison to the existing clustering algorithms. This is true because during the non-volunteer cluster head selection procedure more number of isolated cluster heads are formed due to the hand off procedure. These cluster heads being unable to re-affiliate with any cluster head, remain as isolated cluster heads. When TRAP is executed on the TACA, the range of some nodes are increased. Thus the isolated cluster heads become able to affiliate to nearby cluster heads. As a result, the total number of cluster heads in the network are reduced which is shown in figure 5.4.

From the above results it is clear that, the increase in transmission range for some nodes improves the performance of the clustering algorithm in reducing its maintenance overhead. But, it is learnt that the energy consumption of the node is proportional to its radio range. So the increase in transmission range of a node, increases its the battery consumption. This can be understood from the results of figure 5.5, where it is shown that the network life time reduces with the increase in the transmission range. The results help in choosing the

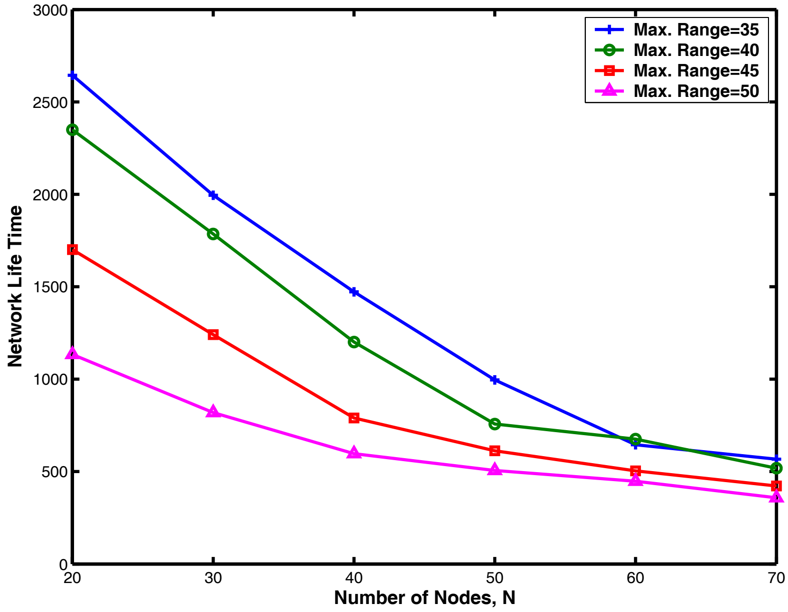


Figure 5.5: Network life time (in ms) for various $Range_{Max}$

$Range_{Max}$ for the topology control protocol.

Finally, the results for the network life time is indicated in figure 5.6.

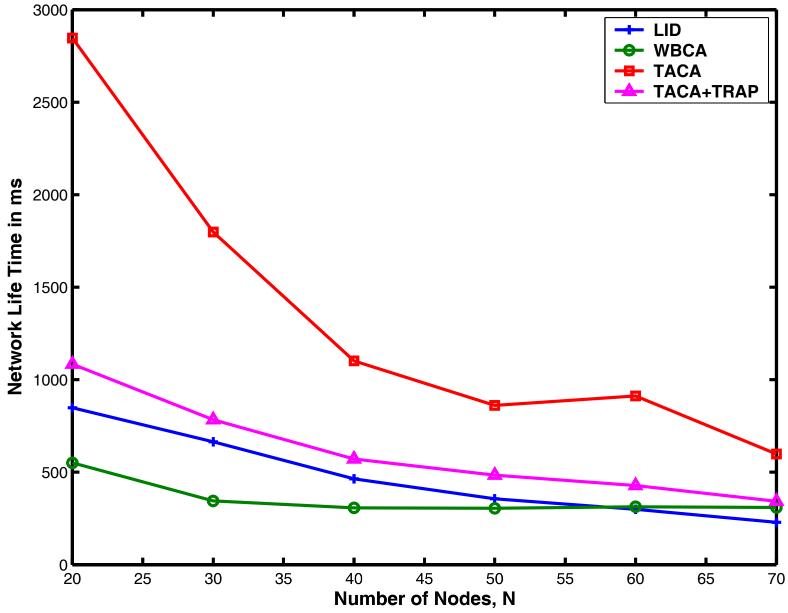


Figure 5.6: Network Life Time (in ms) when TRAP is executed

As discussed earlier, the network life time may be defined as the time elapsed since the network is first activated till the first node in the network goes dead. The longer the network life time, the better the communication stability. Thus, every communication network targets to achieve longer network life time. The results in the figure indicate that when TRAP is executed on TACA, the network life time is reduced. Because higher energy consumption of some nodes causes early death of those nodes resulting in reduced network life time. However, it is also seen in the figure that the network life time still remains higher in comparison to the existing algorithms, when $Range_max$ is choosen as 50mts.

5.7 Summary

The objective of topology control is to enable most of the nodes to remain connected in the network while consuming optimum energy. In this chapter, a transmission range adjustment protocol is proposed to allow the isolated cluster heads to increase their transmission ranges, so that they could affiliate to the existing cluster heads. This reduces the number of hops in the virtual backbone reducing the delay in communication. Further, the increase in transmission range of some nodes enables the other nodes to remain connected even on the move resulting in reduced frequency of re-affiliation and cluster head re-election. Thus, it can be concluded that, the proposed protocol reduces the maintenance overhead as well as the delay in communication meeting the requirements of an efficient cluster design. However, the maximum range that a node can posses has to be decided judiciously, so that the network life time is not compromised.

Chapter 6

Conclusion

In cellular networks, base stations are connected with wired backbone, which support the mobility of client nodes for hand-off protocols, paging and user tracking in the network. In such networks, the mobile nodes directly communicate with the fixed base station, reducing the wireless part of communication to the single hop problem. This concept of cellular networks can be mapped into the infrastructure-less peer-to-peer network, so that selected number of nodes perform the job of base stations and form the virtual backbone of communication. This process of selecting few nodes as the virtual base stations, where their one hop neighbors directly communicate with them, can be visualised as the formation of logical clusters in the network. Thus, every cluster consists of a cluster head representing the virtual base station and its one hop members within it.

In this thesis, protocols and algorithms are proposed for the efficient design of clustering in MANET. The existing clustering schemes are throughly invesigated by simulation in chapter 2. The comparison of the simulation survey highlights the flaws and strengths associated with the existing one hop clustering algorithms. Analysis of results have provided a direction to move with proposing new protocols and algorithms for clustering.

To start with a clustering algorithm, it is required that the nodes in the

MANET must be aware of the network topology. Aquiring the knowledge of the global network topology is more time consuming than that of its local counterpart. So to enable the nodes to probe their one-hop neighbours in the network,a neighbour detection protocol is proposed in chapter 3. The protocol uses neighbour request packets by the sender nodes and corresponding acknowledgements by the receivers. This protocol also enables the nodes to receive the detail information about the neighbours. The working principle of the protocol is analysed through simulation by using the CPN tools. The validation process ensures that the flow of control and data in the system takes place as per the requirements.

A distributed topology adaptive clustering algorithm TACA is proposed in chapter 4, that considers the node mobility and its battery power for calculating its node weight. Initially, a node is selected as the volunteer cluster head, if it posseses the highest weight among its one-hop neighbors. Selecting the head with the knowledge of local topology, makes the selection process faster reducing the cluster set up time. This also reduces the number of message exchanges by the nodes to find the higher weighted node. Moreover, the change in topology outside the range of the concerned node hardly affects the cluster head selection procedure. After the cluster heads are selected, its one-hop uncovered nodes become its members forming the one-hop logical clusters. Subsequently, non-volunteer cluster heads are also selected, as the current head consumes its battery power beyond a threshold value. This selection process also takes place locally within a cluster zone, so that rest of the nodes outside the cluster remain undisturbed. In this process, the current cluster head slects one of its members with the highest current weight to become the new non-volunteer head. And the members of the current head those lie within the range of the newly selected head are handed over to it by the soft hand-off procedure. The member which does not lie within the range of the non-volunteer head are allowed to choose either any nearby head to get affiliated or to become an isolated head without any member. The selection of non-volunteer head by the current head enables it to save its remaining battery power and to increases its life span. In the current work remaining of 50 percent of its total battery power is considered as the threshold amount, after which the existing head can resign from its current status and allow other node with more available battery power to take up the role of cluster head. The distribution of energy consumption by the nodes in the network increases the network life time, which is a prime goal of communication. The proposed clustering algorithm has been validated through simulation by CPN to ensure its functionality.

As discussed,the proposed clustering algorithm results in the formation of some isolated cluster heads during the initial clustering phase and during the subsequent non-volunteer cluster head selection phase. The increase in the number of cluster heads increase the length of the communication backbone in terms of number of hops. This may increase the end-to-end delay in communication for the packets. Hence, a topology control protocol is proposed in chapter 5, that allows the nodes to adjust their transmission ranges to remain connected with other nodes as and when required. The proposed transmission range adjustment protocol helps the isolated nodes to get affiliated with existing cluster heads instead of becoming new heads. The proposed protocol reduces the end-to-end delay by reducing the number of cluster heads in the network. Simultaneously, the cluster maintenance overhead is also reduced as discussed in the chapter5.

Scope for Further Research

The research findings made out of this thesis has opened several auxiliary research directions, which can be further investigated. The proposed protocols that mostly deal with the cluster formation, cluster maintenance and energy consumption, can be extended to some other areas of clustering like load balancing among the cluster head, fault tolerant clustering or privacy and security in clustered MANET. Forming a connected dominating set with the topology control could be an interesting topic to proceed with. By implementing the range adjustment protocol for the nodes, their transmission ranges become variable. So at this juncture, how the nodes will detect their one-hop neighbors with NDP could be analysed carefully and further research could be made in this direction. At times, the GPS installed to the nodes may not provide the accurate position of the nodes. Thus deriving the actual speed of the node with respect to its node position is a challanging task. A further study will be helpful to find the accurate mobility of the node and its subsequent analysis. Another direction is to optimise the value of maximum transmission range for which the cluster maintenance can be reduced without compromising with the network life time.

This thesis deals with the CPN tools for validating the protocols, when the nodes are assumed to be static, that is at an instance when the nodes do not move. Such an assumption of non-mobility is not a realistic assumption when the nodes movement are quite unpredictable. Some authors assume a topology approximation method to predict the mobility, which is again not a viable solution. Thus, another challenging research direction could be to persue the validation of the protocols in MANET with its frequently changing topology by using the timed petrinets where the mobility of nodes could be studied with respect to its changing positions from time to time.

Bibliography

[1] C. E. Perkins. *Ad Hoc Networking.* Addison Wesley, 2000.

[2] Luc Hogie. *Mobile Ad Hoc Networks: Modelling, Simulation and Broadcast-based Applications.* PhD thesis, University of Luxembourg, April 2007.

[3] C. R. Lin and M. Gerla. Adaptive clustering for mobile wireless networks. *IEEE Journal on Selected Areas in Communications*, 15(7):1265–1275, sept 1997.

[4] http://trends-in-telecoms.blogspot.com/2011/05/4g-wireless-architecture-and.html.

[5] George Aggelou. *Mobile Ad Hoc Networks: From Wireless LANs to 4G Networks.* Tata McGraw-Hill, 2009.

[6] C Siva Ram Murthy and B S Manoj. *Ad Hoc Wireless Networks.* Pearson Education, 2005.

[7] C K Toh. *Ad Hoc Mobile Wireless Networks: Protocols and Systems.* Prentice Hall PTR, 2002.

[8] C. C. Chiang, H.K. Wu, W.Liu, and M. Gerla. Routing in clustered multihop, mobile wireless networks with fading channel. In *Proceedings of IEEE Singapore International Conference on Networks SICON' 97*, pages 197–212, 1997.

[9] I. Chlamtac, M.Conti, and J.-N. Liu. Ad hoc networking: imperatives and challanges. *Ad Hoc Networks*, 1:13–64, 2003.

[10] I. Chlamtac and A. Farago. A new approach to the design and analysis of peer-to-peer mobile networks. *Wireless Networks*, 5:149–156, 1999.

[11] T. Ohta aand S. Inoue and Y. Kakuda. An adaptive multi-hop clustering scheme for highly mobile ad hoc networks. In *proceedings of sixth international symposium on autonomous decentralized systems(ISADS'03)*, Pisa, Italy, April 2003.

[12] C. R. Lin and M. Gerla. A distributed control scheme in multi-hop packet radio networks for voice/data traffic support. In *Proceedings of IEEE GLOBECOM*, pages 1238–1242, 1995.

[13] S Bandyopadhyay and E J Coyle. An energy efficient hierarchical clustering algorithm for wireless sensor networks. In *Proceedings of INFOCOM 03*. IEEE, April 2003.

[14] S. Basagni, I. Chlamtac, and A. Farago. A generalized clustering algorithm for peer-to-peer networks. In *Proceedings of Workshop on Algorithmic aspect of communication (satellite workshop of ICALP)*, July 1997.

[15] A Ephremides. Energy concerns in wireless networks. *IEEE Wireless Communications*, pages 48–59, 2002.

[16] J. H. Chang and L. Tassiulas. Energy conserving routing in wireless ad hoc networks. In *Proceedings of Workshop on Algorithmic aspect of communication (satellite workshop of ICALP)*, pages 22–31, Tel-Aviv, Israel, March 2000.

[17] M. R. Garey and D. S. Johnson. *Computers and Intractability: A guide to the theory of NP-completeness*. Freeman, San Francisco, CA, 1979.

[18] S Basagni, M Mastrogiovanni, and A Panconesi. Localized protocols for ad hoc clustering and backbone formation: A performance comparison. *IEEE Transactions on Parallel and Distributed System*, 17(4):292–306, 2006.

[19] B. Bollbas. *Random Graphs.* Academic Press, 1985.

[20] D. J. Baker and A. Ephremides. The architectural organisation of a mobile radio network via a distributed algorithm. *IEEE Transactions on Communications*, 29(11):1694–1701, November 1981.

[21] F.A Tobagi and L. Kleinrock. Packet switching in radio channels: part ii- the hidden terminal problem in carrier sense multiple access and the busy tone solution. *IEEE transaction on Communication*, 23:1417–1433, December 1975.

[22] D. J. Baker, A. Ephremides, and J. A. Flynn. The design and simulation of a mobile radio network with distributed control. *IEEE Journal on selected areas in communication*, SAC-2(1):226–237, January 1984.

[23] D. Turgut S. Das, M. Chatterjee. Wca: A weighted clustering algorithm for mobile ad hoc networks. *Journal of Cluster computing (special issue on mobile ad hoc networks)*, 5(2,4):193–204, 2002.

[24] A. A. Siddiqui and Ravi Prakash. Effect of availability factor threshold and clustering gap on performance of clustering mechanisms for multi-cluster mobile ad hoc networks. In *Proceedings of IEEE Conference on Communications (ICC'02)*, volume 5, pages 3296–3300, April 2002.

[25] J.E. Wieselthier A. Ephremides and D.J. Baker. A design concept for reliable mobile radio networks with frequency hopping signalling. *Proceedings of IEEE*, 75(1):56–73, 1987.

[26] M. Gerla and J. T. C. Tsai. Multicluster, mobile,multimedia radio network. *Wireless Networks*, 1:255–265, 1995.

[27] A.K. Parekh. Selecting routers in ad hoc wireless network. In *Proceedings of the SBT/IEEE International Telecommunication Symposium ITS*, August 1994.

[28] N. Khan P.Basu and T.D.C. Little. A mobility based metric for clustering in mobile ad hoc networks. In *Proceedings of IEEE ICDCS 2001 workshop on wireless networks and mobile computing*, Phoenix, AZ, 2001.

[29] S.Basagni. Distributed clustering for ad hoc networks. In *Proceedings of International symposium on parallel architectures, algorithms and networks*, pages 310–315, June 1999.

[30] S.Basagni. Distributed and mobility adaptive clustering for multimedia support in multi-hop wireless networks. In *Proceedings of vehicular technology conference VTC*, volume 2, pages 889–893, 1999.

[31] R Vilzmann. Analysis of different clustering techniques for mobile ad hoc networks. Wireless Communications, March 2002.

[32] M. Chatterjee S. Das and D. Turgut. An on demand weighted clustering algorithm (wca) for ad hoc networks. In *Proceedings of IEEE GLOBECOM 2000*, pages 1697–1701, San Francisco, November 2000.

[33] M. Chatterjee, S.K. Das, and D. Turgut. A weight based distributed clustering algorithm for mobile ad hoc networks. In *Proceedings of 7th International Conference on High Performance Computing (HiPC 2000)*, pages 511– 521, Bangalore, India, December 2000.

[34] R. Ghosh and S. Basagni. Limiting the impact of mobility on ad hoc clustering. In *proceedings of the 2nd ACM Internaional Workshop PE-WASUN'05*, pages 197–204, Montreal, CA, 2005.

[35] W. D. Yang and G. Z. Zhang. A weight- based clustering algorithm for mobile ad hoc network. In *Proceedings of 3rd International Conference on*

wireless and mobile communications (ICWMC '07), page 3, Guadeloupe, March 2007.

[36] Lisa Wells. Performance analysis using cpn tools. In *Proceedings of the 1st international conference on Performance evaluation methodolgies and tools (VALUETOOLS 06)*, page 59. ACM, Oct 2006.

[37] Kurt Jensen. *Coloured Petri Nets.* Springer Verlag, 1992.

[38] T. Murata. Petri nets: Properties, analysis and applications. In *Proceedings of IEEE*, volume 77(4), pages 541–580, 1989.

[39] http://daimi.au.dk/CPnets/proxy.php?url=/CPnets/index.

[40] S. Christensen L. M. Kristensen and K. Jensen. The practitioner's guide to coloured petri nets. *International Journal on Software Tools and Technology Transfer*, 2:98–132, 1998.

[41] T. Murata Y. Zhou and T. Defanti. Modelling and performance analysis using extended fuzzy-timing petri nets for networked virtual environments. *IEEE Transactions on Systems, Man and Cybernetics- Part B: Cybernetics*, 30(5):737–756, 2000.

[42] T. Murata C. Xiong and J. Tsai. Modelling and simulation of routing protocol for mobile ad hoc wireless networks using coloured petri nets. In *proceedings of Workshop on Formal Methods Applied to Defence Systems in Formal Methods in Software Engineering and Defence Systems*, volume 12, pages 145–153, 2002.

[43] J. Billington C. Yuan and J. Freiheit. An abstract model of routing in mobile ad hoc networks. In *proceedings of Workshop on Formal Methods Applied to Defence Systems in Formal Methods in Software Engineering and Defence Systems*, volume 12, pages 145–153, 2002.

[44] R Dube, C D Rais, K Y Wang, and S K Tripathi. Signal stability based adaptive routing(ssa) for ad hoc mobile networks. *IEEE Personal Communications*, 4(1):36–45, 1997.

[45] M A Rajan, M G Chandra, L C Reddy, and P Hiremath. Concepts of graph theory relevant to ad hoc networks. *Journal of Computers Communications and Control*, III(2008):465–469, 2008.

[46] L. M. Feeney and M. Nilsson. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *proceedings of IEEE INFOCOM 01*, pages 1548–1557, 2001.

[47] L. M. Feeney. An energy consumption model for performance analysis of routing protocols for mobile ad hoc networks. *Journal of Mobile Networks and Applications*, 6(3):239–249, 2001.

[48] Q. Gao, K. J. Blow, D. J. Holding, I. W. Marshall, and X. H. Peng. Radio range adjustment for energy efficient wireless sensor networks. *Ad Hoc Networks*, 4(3):75–82, 2006.

[49] G. Noubir G. Lin and R. Rajaraman. Mobility models for ad hoc network simulation. In *proceedings of IEEE INFOCOM 04*, 2004.

[50] M. Zonoozi and P. Dassanayake. User mobility modeling and characterization of mobility pattern. *IEEE Journal on selected areas in communications*, 15(7):1239–1252, 1997.

[51] J. Boleng T. Camp and V. Davies. A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing (WCMC): Special Issue on Mobile Ad Hoc Networking: Research, Trends and Applications.*, 2(5):483–502, 2002.

[52] Paolo Santi. *Topology Control in Wireless Ad Hoc and Sensor Networks.* John Wiley & Sons Ltd., 2005.

[53] T. Hou and V. Li. Transmission range control in multi-hop packet radio networks. *IEEE Transactions on Communications*, COM-34:38–44, 1986.

[54] V Rodopu and T Meng. Minimum energy mobile wireless networks. *IEEE Journal in Selected Areas in Communication*, 17(8):1333–1344, 1999.

[55] L. Li, J Halpern, P. Bahl, Y. Wang, and R Wattenhofer. Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks. In *proceedings of ACM PODC 01*, pages 264–273, Newport, RI, 2001.

[56] G. Resta D. Blough, M. Leoncini and P. Santi. The k-neighbors protocol for symmetric topology control in ad hoc networks. In *proceedings of ACM MobiHoc 03*, pages 141–152, Annapolis, MD, 2003.

[57] P Santi. The critical transmitting range for connectivity in mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 4(3):310–317, 2005.

[58] Pietro Manzoni Departamento, Miguel Snchez, Pietro Manzoni, and Zygmunt J. Haas. Determination of critical transmission range in ad-hoc networks. In *in Proceedings of Multiaccess Mobility and Teletraffic for Wireless Communications 1999 Workshop (MMT99)*, 1999.

[59] K Paul and S Bandyopadhyay. Self adjusting transmission range control of mobile hosts in ad hoc wireless networks for stable communication. In *Proceedings of 6th International Conference on High Performance Computing (HiPC 99)*, Kolkata, 1999.

[60] R. Ramanathan and R. Rosales-Hain. Topology control of multi-hop wireless networks using transmit power adjustment. In *proceedings of IEEE Infocom 00*, pages 404–413, Tel Aviv, 2000.

[61] S. Basagni, M.Conti, S.Giordano, and I.Stojmenovic. *Mobile Ad Hoc Networking*. John Wiley & Sons Inc., 2004.

[62] E. M. Royer and C. K. Toh. A review of current routing protocols for ad hoc mobile wireless networks. *IEEE Personal Communication*, pages 46–55, 1999.

[63] K. Erciyes, O. Dagdeviren, D. Cokuslu, and D. Ozsoyeller. Graph theoretic clustering algorithms in mobile ad hoc networks and wireless sensor networks-survey. *Appl. Comput. Math*, 6(2):162–180, 2007.

[64] S. Bandyopadhyay and E. J. Coyle. An energy efficient hierarchical clustering algorithm for wireless sensor networks. In *Proceedings of IEEE INFOCOM '03*, April 2003.

[65] J. Y. Moulin E. Lochin, A. Fladenmuller and S. Fdida. Energy consumption models for ad-hoc mobile terminals. In *Mobile Ad Hoc Network*, Paris Cedex 05, France, 2005.

[66] G. Noubir G. Lin and R. Rajaraman. Mobility models for ad hoc network simulation. In *Proceedings of IEEE INFOCOM'04*, Hong Kong, March 2004.

[67] V. Davies T. Camp, J. Boleng. A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing (WCMC)*, 2(5):483–502, 2002. Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications.

[68] L. M. Kristensen K. Jensen and L. Wells. Coloured petri nets and cpn tools for modelling and validation of concurrent systems. *International Journal on Software Tools and Technology Transfer*, 9:213–254, 2007.

[69] Bulent Tavli and Wendi Heinzelman. *Mobile Ad Hoc Networks: Energy Efficient Real-Time Data Communications.* Springer, 2009.

[70] I Chlamtac, M Conti, and J N Liu. Mobile ad hoc networking: Imperatives and challenges. *Ad Hoc Networks*, 1:13–64, 2003.

[71] Y J You and J H P Chong. A survey of clustering schemes for mobile ad hoc networks. *IEEE Commun. Surv Tutor*, 7(1):32–47, 2005.

[72] J B D Cabrera, R Ramanathan, C Gutierrez, and R K Mehera. Stable topology control for mobile ad hoc networks. *IEEE Communications Letters*, 11(7):574–576, 2007.

[73] Andy An-Kai Jeng and Rong-Hong Jan. Adaptive topology control for mobile ad hoc networks. *IEEE Transactions on Parallel and Distributed Systems*, 99(preprints), 2011.

[74] C Constantopoulos, D Gavalas, and G Pantziou. Clustering in mobile ad hoc networks through neighborhood stability. *Comput. Netw.*, 52(9):1797–1824, 2008.

Dissemination

1. Suchismita Chinara and Santanu Kumar Rath. A survey on one-hop clustering algorithm in mobile ad hoc networks. *Journal of Networks and System Management*, 17:183–207, May 2009.
2. Suchismita Chinara and Santanu Kumar Rath. Topology control by transmission range adjustment protocol for clustered mobile ad hoc networks. *ISRN Communications and Networking*, 2011. (*Accepted*).
3. Suchismita Chinara and Santanu Kumar Rath. Validation of Neighbour Detection Protocol for MANET using Coloured Petri Nets. *International Journal of Computer Science*, (*Communicated*).
4. Suchismita Chinara and Santanu Kumar Rath. Modeling of a topology adaptive clustering algorithm for mobile ad hoc networks using coloured petri nets. In *Proc. of International Conference on Software and Computer Engineering (ICSCE '10)*, pages 1191–1199, Singapore, Aug. 2010. WASET.
5. Suchismita Chinara and Santanu Kumar Rath. Taca: A topology adaptive clustering algorithm for mobile ad hoc networks. In *Proc. of International Conference on Wireless Networks (ICWN '09)*, pages 391–397, Las Vegas,USA, July 2009. WORLDCOMP.
6. Suchismita Chinara and Santanu Kumar Rath. Energy efficient mobility adaptive distributeed clustering algorithm for mobile ad hoc networks. In *Proc. of Internaional Conference on Advanced Computing and Communications (ADCOM 2008)*, pages 265–272, Chennai, Dec. 2008. IEEE.
7. Suchismita Chinara and Santanu Kumar Rath. Mobility based clustering algorithm and the energy consumption model of dynamic nodes in mobile ad hoc networks. In *Proc. of IEEE International Conference on Information Technology (ICIT'08)*, Bhubaneswar, Dec. 2008. IEEE.
8. Suchismita Chinara and Santanu Kumar Rath. Energy efficient intra cluster hand-off algorithm for mobile ad hoc networks. In *Proc. of International Conference on Information Systems and Technology (ICIST '07)*, pages 23–27, Kerala, Dec. 2007.

Resume

Suchismita Chinara

Assistant Professor
Dept. of Computer Science
NIT Rourkela,
Rourkela - 769008, Odisha, India

Ph: `+91-661-2462361 (O), +91-9437116795 (M)`

e-mail: `suchismita@nitrkl.ac.in`

Date of Birth

June 28, 1972

Permanent Address

S-28, Maitrivihar Ph-I, Chandrasekharpur,
Bhubaneswar – 751 023, Odisha, India.

Qualification

- Ph.D. (Continuing)
  NIT Rourkela
- M.Tech. (CS)
  Regional Engineering College, Odisha [First Division with Honours]
- AMIE (CSE)
  The Institution of Engineers (India), Kolkata, [First division]

Publications

- 02 Journal Articles
- 01 Journal Communicated
- 08 Conference Papers