

AN OPTIMIZED MECHANISM FOR ADAPTIVE AND DYNAMIC POLICY BASED HANDOVER IN CLUSTERED MANET

¹Nidhi Parashar

Department of Information Technology, Meerut Institute of Technology, Meerut, U.P.,
India.

nidhi.csit@gmail.com

ABSTRACT

On-going revolution in ever-improving wireless communication enforces the necessity of a self configuring, rapidly deployable and infrastructure less network. MANET is such an autonomous wireless network that meets the requirements. At the same time MANET's random behavior and absence of any central intelligence to gather unambiguous knowledge about user contexts complexes QoS maintenance and hampers proper utilization of network resources resulting into unnecessary handovers. In past, few policy driven handover approaches have been proposed for MANET but none of them explores a comprehensive policy design. Therefore in this paper we propose an adaptive and optimized policy based handover mechanism which is based on explicitly designed policies like load balancing, service discovery and next hop selection .Efficient procedures for these policies are also explored .This work predicts the high time of handover need on the basis of application specific needs of individual freely roaming mobile nodes, avoiding unnecessary handovers and provides efficient handover procedure with optimized resource consumption, reduced latency and interruption time.

KEYWORDS

Mobile ad-hoc network (MANET), service discovery, Application specific threshold, Energy-Distance Factor, Caching, Load balancing, QoS, RSSI, Handover.

1. INTRODUCTION

With the recent advances in the development of wireless communication technologies an Infrastructure less network MANET is quickly growing for its ability to form the network of dynamic topology without any fixed or predefined infra-structure. The mobile nodes are constrained with high mobility and energy limitations therefore a hierarchical architecture is needed to achieve performance guarantee in a large scale MANET thus MANET nodes are grouped into clusters. Under a cluster structure resources can be spatially reused to increase the system capacity and two clusters may use the same frequency or code set if they are not neighboring clusters. Internal events in the cluster can be coordinated by its cluster head. For satisfactory performance of MANET an effective cluster formation, cluster head selection mechanism considering all the network dynamics is needed along with a handover mechanism that performs efficiently in various network conditions.

Policy based handover mechanisms have been done in past but no one specifies the design and mechanisms for policies to be considered while taking handover decision. Here an adaptive and optimized handover mechanism based on certain policies has been deduced that significantly avoids the overhead created due to unwanted handover initiation. At the same time detail mechanisms for different policies are optimized in terms of network throughput and QoS parameters resulting into an optimal handover decision reducing the handover latency and processing delay. Energy distance factor [7] facilitates the energy efficient decisions in different phases of the proposed mechanism.

Rest of the paper is organized as follows. Section-2 gives the literature review while proposed work is discussed in section-3. Section-4 concludes the contribution of this paper along with the future scope in Section-5.

2. BACKGROUND

Dynamic Cluster Protocol [1] having five states such as un-clustered state, orphan state, election state, cluster node state, and cluster head state are identified for a MANET. Detection and control mechanism for DDOS attacks over reputation and score based MANET [2] gives clustering technique using reputation and score value of nodes but the important parameters like distance from centroid, processing capability and mobility of a node in cluster head selection procedure have not been considered.

A distributed directory based service discovery mechanism [6] operates in a proactive mode and selects a provider based on distance and service capability of the provider. A probability based node selection method [7] identifies the inter-mediate node with optimum stored energy that could withstand through duration of connection. A handover scheme considering the current active application in use rather than a single fixed threshold value and dynamically changing the threshold value depending upon the currently active applications [8] have been used to avoid unnecessary handovers.

Policy Based Fast Handover Mechanism for MANET [4] provides proactive and reactive handover approaches based on policy for nodes of MANET but parameterized design for different policies in changing situations have not been explored. In addition to the more dynamic and adaptive mechanisms for cluster formation and cluster head selection, most of the policy mechanism to provide best handover decisions and reducing the overhead caused by unnecessary handovers.

3. PROPOSED WORK

This section presents the proposed Architecture and mechanism in the sub sections 3.1 and 3.2.

3.1 Architecture of Optimized Policy Based Handover

The proposed architecture in Fig. 1 includes the following modules.

3.1.1. Cluster formation module

It co-ordinates cluster formation in MANET. All mobile nodes in MANET are organized into clusters on the basis of Device type, Bandwidth, RSSI value, Services required, Distance between nodes.

3.1.2. Cluster Head Selection module

It facilitates the score based weighted selection of a node as Cluster head.

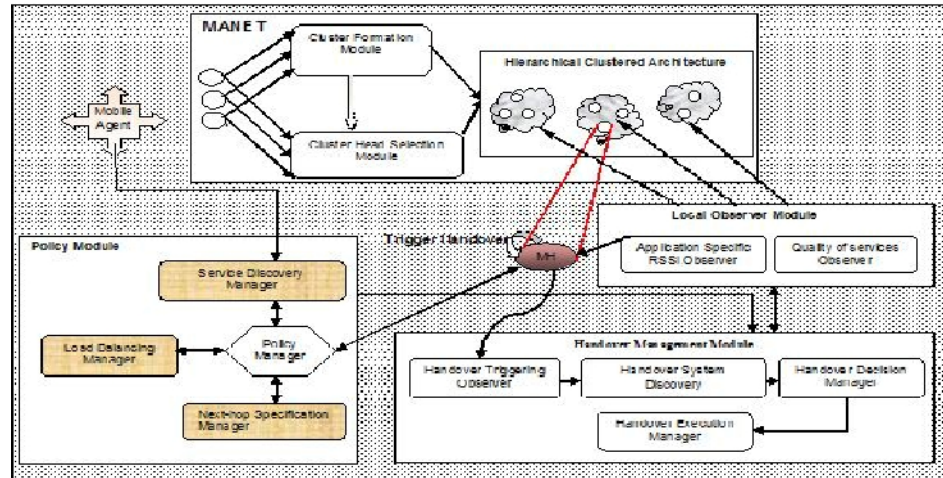


Figure 1. Adaptive and Optimized Policy Based Handover

3.1.3. Policy Module

It comprises different policy mechanisms that guarantee efficient resource utilization along with better QoS inside MANET[4].

Load Manager manages the load balancing task inside a cluster. On the basis of current load and remaining battery power of the cluster nodes the load among them is distributed for the smooth and efficient working of cluster.

Service Manager manages intra cluster and intra cluster service discovery to the MN while moving from one place to another.

Next-Hop Specification Manager selects optimal cluster for a MN when it receives equal RSSI value from two or more clusters.

Policy Manager Coordinates working of overall policy module .It collects the updated policy management in-formation from all the managers, compiles the gathered information and gives suggestions to MN regarding handover need and Quality of services management.

3.1.4. Mobile Agent

These freely moving ad hoc network nodes collect the location and description of services available in different clusters and make this global service information available to the CHs of all clusters.

3.1.5. Local Observer module

It observes the local functioning of a cluster and assists the CH in various cluster maintenance activities.

Application Specific RSSI Observer calculates the RSSI requirement of all the active mobile nodes within a cluster according to the currently running applications in each mobile node[8]. If more than one application is running simultaneously in a mobile node and all applications require different RSSI values then this module selects the maximum of them as $RSSI_{AST}$.

Quality of services Observer keeps track of all the network parameters like bandwidth, RSSI, load distribution on all the mobile nodes within the cluster.

3.1.6. Handover Management module

It provides optimal handover when MN detects the need to move seamlessly within MANET.

Handover Triggering Observer receives handover request from MN and invokes handover system discovery manager to find potential CHs.

Handover System Discovery Manager prepares list of target CHs on the basis of RSSI required, capability of target CHs and the context for which handover need is initiated and sends this list to Handover Decision Manager for target selection.

Handover Decision Manager selects the optimal target cluster.

Handover Execution Manager coordinates the actual handover including authentication and association to the target cluster.

3.2 Mechanism of Optimized Policy Based Handover

3.2.1. Optimized Cluster Creation and cluster Head Selection

In this primary phase details of cluster formation and cluster head selection mechanisms are specified.

Node Admission Control manages the entry of a new node in MANET. While a node enters MANET it is in UN Clustered state. It sends a hello message to all neighbors. After a predefined number of tries If it does not get reply before expiry of timer (t_{uc}),it enters into Orphan state.

```

NodeAdmissionControl()
{
    If (t<=tuc)

        ApplyForCluster();

    Else

        {
            Ni turns Orphan.

            If(gets GetState from CH)

                ApplyForCluster ();

            Else

                CreateNewCluster();}
}

ApplyForCluster()
{For (i=1; i<=n; i++)

    If((clusterSize<MaxSize)&& (Ni=DType)&& (BWNi>=BWT)

        JoinCluster ();

    ElseIf(clusterSize>MaxSize)

        CreateNewCluster ();

    Else

        MergeCluster();//if no.ofnode<MinSize
}

JoinCluster()

{//Ni is in eligibility test and Nj is the existing node

    If((DNiNj<=DTHRESHOLD)&&(ClusterRangeNi<=RangeTH)&&(RSSINi>=RSSITH)&&
    (ServiceRequired∈ServiceOffered))

        Ni is accepted in the cluster;
}

CreateNewCluster()
{
    Appoint the initiating node as CH;
    If(CH does not get any message for a time
    period)
        CH changes its state from CH to

```

```

        Orphan State;
    Else
        ClusterHeadSelection();}

MergeCluster( )
{//NC is total no of user clusters.
    For (j=1; j<=NC-1; i++)
        If(SizeCi<MinSize||SizeCj<MinSize)&&(Ci and Cj are in range)
            If(SizeCi<=SizeCj)
                {//CP is the processing power of node.
                    If((CPCH>=CPTHRESHOLD)&&(CHnotoverloaded)&&(ServicesOfferedCi €
                    ServiceOfferedCj))
                        For all nodes in Ci
                            ApplyForCluste
                                r;
                            ClusterHeadSel
                                ection();
                    ElseIf((CPCH>=CPTHRESHOLD)&&(LoadCH<=CapacityCH)&&(ServicesOfferedCj € ServiceOffe-
                    redCi))
                        For all nodes in Cj
                            ApplyForCluste
                                r;
                            ClusterHeadSel
                                ection();}
                }
}

```

Cluster Head Selection process considers the following scores.

Reputation Score[2] returns trust value of the node depending upon its previous record of packet forwarding capability. Each node in cluster contains an entry {N_{ID},S_{TRUST},U_{TRUST}}.Let S_{TRUST} and U_{TRUST} are scores for correct and In-correct packet forwarding capability S_{TRUST} (i,j)=1, if i gets service from j and U_{TRUST} (i,j)=-1 otherwise. Trust value can be calculated as

$$tValue= S_{TRUST}(i,j)+ U_{TRUST}(i,j) \quad (1)$$

Mobility Score provides means to deduce stability factor of mobile node D_{i,j} is the distance between two nodes i and j and MD_i is Mean distance between N_i and all its neighbors ST_i is the stability of node i.

$$D_{i,j}=\sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2} \quad (2)$$

$$MD_i = \frac{1}{N} \sum_{i=1}^N D_{i,j} \quad (3)$$

$$ST_i = MD_t - MD_{t-1} \quad (4)$$

Energy Distance Score finds node that consume less energy to communicate with rest of cluster nodes resulting in less battery power consumption [7]. Every node calculates Energy-Distance factor(ED) with other node. Node having highest ED can be preferred as CH. Let E_i and D_i is residual Energy and distance of a node respectively.

$$ED = \frac{E_i \cdot D_i}{E_i + D_i} \quad //0 \leq ED \leq 1 \quad (5)$$

Degree_Score() gives the number of neighbor nodes. Node calculates no of neighbors on the basis of received Get State messages.

DFC() gives the distance from cluster centroid. Let (x_c, y_c) are co-ordinates of cluster centroid and $D_{i,c}$ is the distance of a node from centroid.

Residual Power() gives the approximation of total energy consumption to transmit a packet from sender to receiver node and calculates the residual energy[11].

AvailableBandwidth() estimates the available bandwidth on a link[10].

```

ReputationScore( )
{ For(i=1;i<=n;i++)
    If(tValueNi>=0)
        Node is trustworthy;
    Else
        Node not trustworthy;
    Return(tValue);
}

Degree_Score ( )
{ DegreeNi =0
  For (i=1; i<=n; i++)
      //where j=1to n.
      DegreeNi++;
  Return (DegreeNi);
}

ResidualPower()
{
  If(Ni is Transmitter)
  {
    ET(n,d)=Eelec * n + Eamp * n * n2
    Eresidual = Eprev - ET(n, d)
  }
}

EnergyDistanceScore( )
{For(i=1;i<=n;i++)
  Calculate EDNi
  Return(EDNi);
}

Mobility_Score()
{For (i=1; i<=n; i++)
  Return (STi);}

```

```

Else
{   ER(n,d) = Eelec * n
    Eresidual = Eprev - ER(n ,d)
}   Return (Eresidual)
}

```

```

DFC()
{
For (i=1; i<=n; i++)
    Di,c = √((Xi - Xc)2 + (Yi - Yc)2)
Return (Di,c);
}

```

AvailableBandwidth()

//Let LC be the link capacity related with one hop neighbor. R_{agg} is the aggregate of the rates assigned all incoming and outgoing flows.

```

AB = MAX{0,LC - Ragg}
Return(AB); }

```

For each node all of these scores are assigned a weight so that summation of all weights is unity as.

$$\sum_{j=1}^n W_j = 1$$

Let W_T[Ni], W_S[Ni], W_E[Ni], W_D[Ni], W_{DFC}[Ni], W_{BP}[Ni] and W_B[Ni] are partial weights assigned for all scores. Global weight for each node is calculated as

$$\begin{aligned}
 W_G[Ni] \ W_G[Ni] = & (W_T[Ni] * t_{Value_{Ni}}) + (W_S[Ni] * ST_{Ni}) + (W_E[Ni] * Eff_{Ni}) + (W_D[Ni] * Degree_{Ni}) \\
 + & (W_{DFC}[Ni] * D_{i,c}) + (W_{BP}[Ni] * E_{residual}) + (W_B[Ni] * AB_{Ni});
 \end{aligned}$$

ClusterHeadSelection()

```

{
For (i=1; i<=n; i++) //for all nodes in the cluster
{   TVNi = ReputationScore(Ni);
    STNi = Mobility_Score(Ni);
    EffNi = Energy_Distance_Score(Ni);
    DegNi = Degree_Score(Ni);
    dfcNi = DFC();
    Battery_PowerNi = ResidualPower ();
    BandwidthNi = AvailableBandwidth();
}
}

```



```

}
For (i=1; i<=n;i++)
{
    calculate  $W_G[Ni]$ ;
    Min_Global_Weight  $W_{Gmin} = \text{MIN}(W_G[Ni] \dots W_G[Nn])$ ;
    Assign node having  $W_{Gmin}$  as CH; //Node having second lowest  $W_G$  is assigned as
    CHBackup
        If(CH fails || LoadCH >= CapacityCH)
        {
            CHBackup is selected as CH.
        }
        If(BatteryCH < BatteryTHRESHOLD)
        {
            CH sends power low signal to neighbors and turns orphan.
            Cluster Head Selection ();
        }
    }
}

```

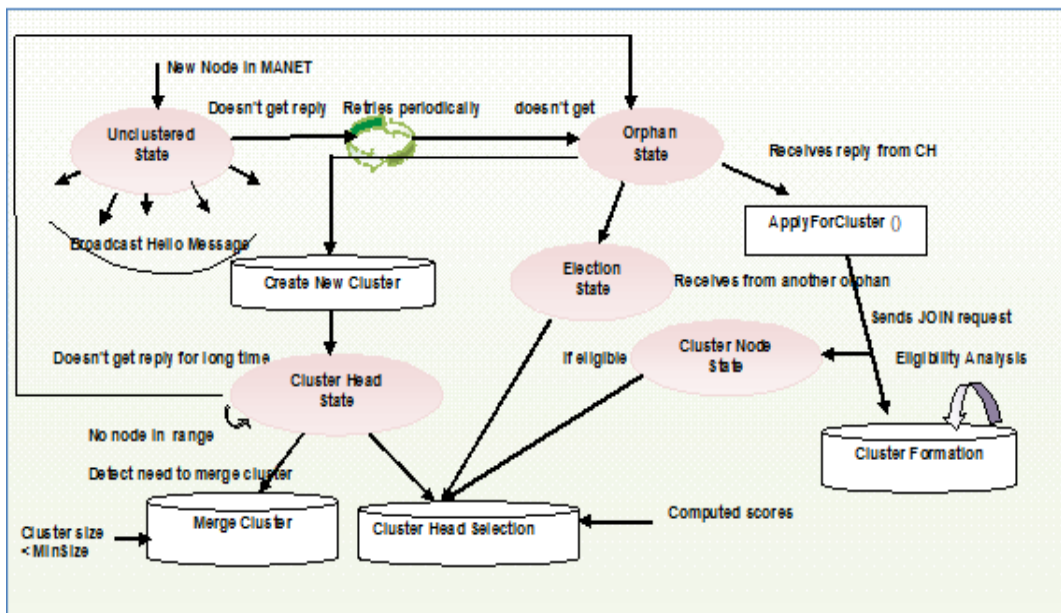


Figure 2. Cluster Creation and Cluster State Head Selection

3.2.2 Caching based on mobility prediction

After the weight based dynamic cluster head selection, the cluster head of every cluster maintains a local cache table (LCT) to store the details of the cached items in the same cluster and a global cache table (GCT) to store the cached items of adjacent clusters. CH predicts the mobility of cluster members using Hidden Markov Model (HMM)[12] to model various hidden parameters. If

the node is about to move, the information of that node is updated on both LCT and GCT. Caching activity at CH is depicted in Figure 3.

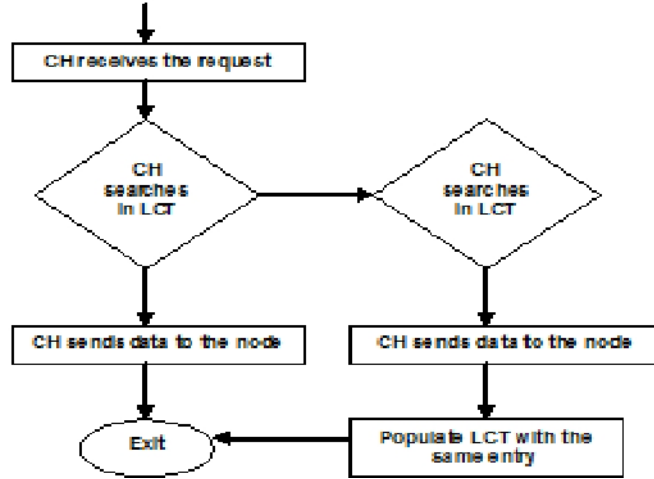


Figure 3. Caching

3.2.3 Optimal Policy Based Parameters

To discover accurate decision techniques for handover process, following policies are considered.

Load Balancing forces CH to periodically collect energy and load values from all cluster nodes [5]. Considering Capacity_{CH} is the max no of nodes CH can serve and Load_{CH} is the no of nodes it is serving at present, an overloaded node generates a Relaxation Request to CH. For load balancing among clusters we have used load balancing factor (LBF) [11] to quantitatively measure the load distribution among clusters. n is the number of CHs. xi is the cardinality of cluster i, μ is the average number of neighbors of a CH and N is total no of nodes.

$$\mu = (N-n)/n \quad (7)$$

LBF is inverse of the cardinality of the clusters.

$$LBF = n \div (xi - \mu)^2 \quad (8)$$

Higher value of LBF implies better load distribution. Load balancing ensures balanced clusters inside the MANET and raises the need for handover when load inside cluster is non manageable providing optimal handover triggering.

Application Requirement results into hybrid mode service discovery given in Figure.4 all SP nodes advertise their services to their CH proactively while SR nodes query their CH for service information whenever it is in need of any service reactively. CH is responsible for intra cluster service discovery and inter cluster service discovery is done with the help of mobile agents. Optimized service discovery mechanism reduces the risk of unnecessary handovers triggered due to requirement of services.

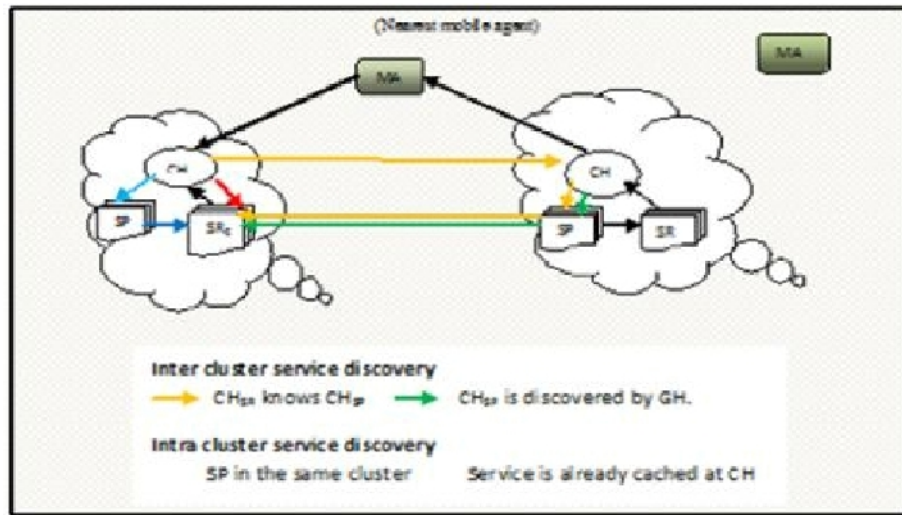


Figure 4. Hybrid Service Discovery

Next-hop Specification helps the cluster nodes who may reach to more than one CH. Let CH_{cur} current CH and CH_{alter} is alternative CH. Such nodes may obtain equal RSSI value from two different CHs. Taking the current load and battery power intermediate hop count of each CH into consideration the better cluster is decided.

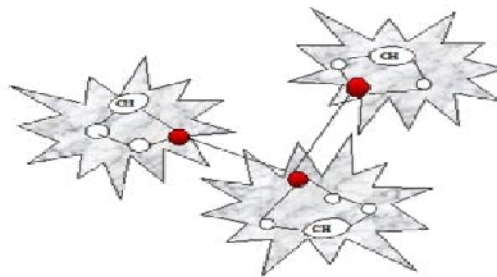


Figure 5. Next Hop Selection

```

IntraClusterLoadBalance( )
{
    For(i=1; i<=n; i++)//for all nodes in cluster

        If(((LoadNi>=MaxLoad)&&(ENi<=MinE))
            ||((LoadNi>=MaxLoad)||(ENi<=MinE))

            Ni sends RelaxReq to CH and CH sends address of capable nodes in
            RelaxRep.
        If(Ni receives more than one address in RelaxRep)
        {
            For(j=1; j<=k; j++)//k possible nodes
                Compute Dij.
        }
    }

```

MinDistance= MIN ($D_{i1}, D_{i2}..D_{ik}$);

Nj at Minimum distance is selected

```

    }
}

```

InterClusterLoadBalance()

```

{
    Every CH periodically calculates LBF If( $LBF_t < LBF_{min}$ )
        Initiate load balancing among clusters.
    Else
        Continue;
}

```

Service_discovery()

```

{
    If (SP exists in same cluster)
        S_req is forwarded to SP that replies to SR .
        Elseif(same S_req has been cached in  $CH_{SR}$ )
             $CH_{SR}$  replies to SR.
        Else if(SP doesn't exist in cluster)
            If( $CH_{SR}$  knows the SP)
                 $CH_{SR}$  forwards S_req to  $CH_{SP}$  which replies.
            Else
                 $CH_{SR}$  sends S_req to nearest MA who forwards S_req to  $CH_{SP}$ 
}

```

NextHopSelection()

```

{
    If(MN directly connects  $CH_{cur}$  || MN directly connects  $CH_{alter}$ )
        CH at zero hop distance is selected as  $CH_{target}$ 
    Else //Let k intermediate nodes to  $CH_{cur}$  and l nodes to  $CH_{alter}$ 
         $ED(CH_{alter}) = \frac{1}{l} \sum_{i=1}^l ED_i$ 
         $ED(CH_{cur}) = \frac{1}{k} \sum_{i=1}^k ED_i$ 
        If( $ED(CH_{alter}) == \text{Max}(ED(CH_{alter}), ED(CH_{cur}))$  && ( $\text{Load}_{CH_{alter}} < \text{Load}_{CH_{cur}}$ )).
             $CH_{alter}$  is selected as  $CH_{target}$ 
}

```

3.4.4. Optimized Handover

Handover can be performed proactively or reactively according to the following stages:

Handover Triggering predicts the need for handover depending upon $Load_{CH}$, available services and $RSSI_{AST}$

Handover System discovery searches for better RSSI value proactively or reactively whenever mobile node experience ($RSSI < RSSI_{AST}$).

Handover Decision State selects the optimal cluster. In Proactive mode to reduce the handover latency period, as soon as the mobile node gathers a number of probe advertisements it starts comparing the QoS offered in each cluster and based on various policies and user contexts selects the best cluster to get better QoS. A Prepare Handover request is sent to the target cluster so that target cluster may reserve the required resources in advance. Same policy is applied in reactive mode.

Handover Execution Phase starts with $Authentication_{Req}$ sent to target CH by MN, After receiving $Authentication_{Ack}$ MN sends the $Association_{Req}$ and waits for $Association_{Ack}$. At the end binding update informing about its new location is sent to the old CH which replies with Binding Ack.
Trigger_Handover()

```

{   For (i=1; i<=n; i++)

        If(( $Load_{CH} \geq Capacity_{CH}$ ) || (!ServicesPresent) || ( $RSSI_{Ni} \leq RSSI_{AST}$ ))

                Initiate System Discovery();

}

Handover Decision( )
{   Find CH with ( $RSSI \geq RSSI_{AST}$ ) && ( $Load_{CHi} \leq Load_{CHcur}$ )

        For(i=1; i<=P; i++) // P is total potential CHs

                If(( $ED_{CHi} == \text{Max}(ED_{CH1}..ED_{CHp})$ ) && ( $ServicesReq \in ServiceOff_{CHi}$ ) && ( $Battery = Battery_{TH}$ ) && ( $Stability_{CHi} = \text{Max}(Stability_{CH1}..CHp)$ ) && ( $Latency < Latency_{TH}$ ))

                        Select CHi as target CH for handover procedure
                Else
                        Repeat the process for the next CH.

}

```

Figure 6. Handover Decision Procedure

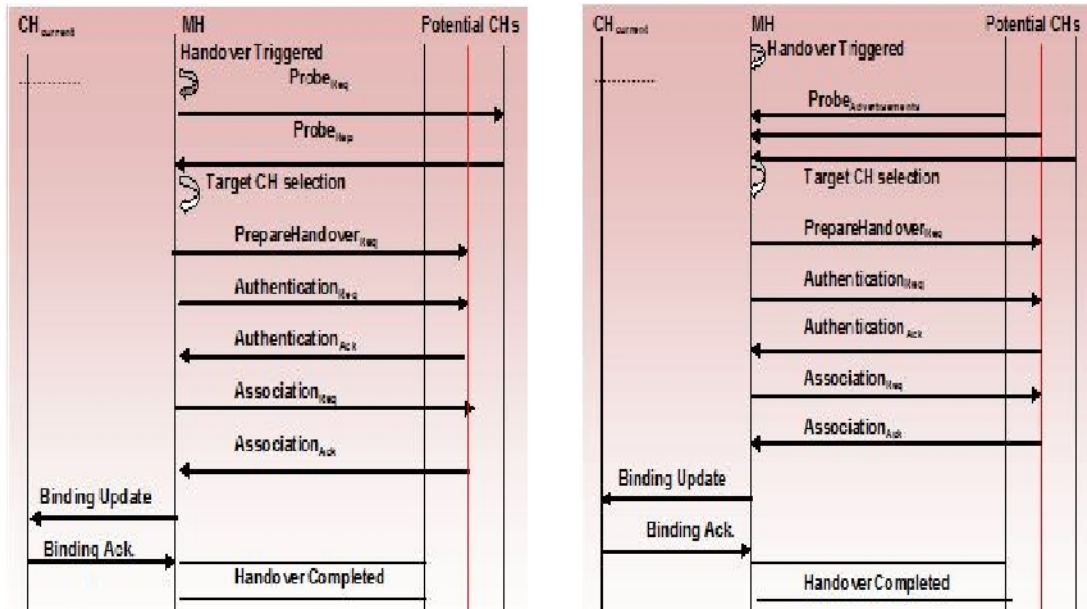


Figure 7. Reactive and Proactive Handover Procedure

4. CONCLUSION

MANET allows mobile nodes to spontaneously form a network and share their services but due to the network's dynamics and resource constraints of wireless nodes, It requires adequate support for service discovery. An effective cluster formation and power efficient cluster head selection mechanism based on reputation mobility, residual energy and link bandwidth have been proposed in this work. Certain policy framework has been explored to achieve balanced clusters, adaptive service discovery and best cluster selection for a freely roaming mobile node. These policies have been enforced on all handover phases resulting in optimized handover with reduced handover, optimal resource consumption, reduced latency and interruption time.

5. FUTURE SCOPE

This mechanism is proposed to achieve required level of flexibility and adaptability to combat the network dynamics. Further exploration is required for a formal validation, based upon prototype implementation and performance evaluation. There is a scope for additional policies which can be explored and implemented as per requirements to increase the effectiveness of proposed scheme. An effective mechanism for location prediction can be incorporated in this framework to assure an optimized handover.

REFERENCES

- [1] Shubha Mishra, Dr. Manish Shrivastava, (2011) "Efficient Secure Clustering Protocol For Mobile Ad-Hoc Network", Journal of Global Research in Computer Science, Vol. 2, No. 9, pp 39-45.
- [2] Rizwan Khan, A. K. Vatsa, (2011) "Detection and Control of DDOS Attacks over Reputation and Score Based MANET", Journal of Emerging Trends in Computing and Information Sciences, Vol. 2, No.11.
- [3] Pradeep Kumar, A. K. Vatsa, (2011) "Novel Security Architecture and Mechanism for Identity based Information Retrieval System in MANET", International Journal of Mobile Adhoc Network, Vol 1, No.3.
- [4] Ankur Gupta, A. K. Vatsa, (2011) "Policy Based Fast Handover Mechanism for MANET", World of Computer Science and Information Technology Journal, Vol.1, No. 8, pp 339-345.
- [5] Shruti Sangwan, Nitin Goel, Ajay Jangra, (2011) "AELB: Adaptive and Efficient Load Balancing Schemes to Achieve Fair Routing in Mobile Ad hoc Networks", IJCST Vol. 2, No.3, pp. 11-15.
- [6] Cynthia Jaypal, Sumathi Vembu, (2011). "Adaptive Service Discovery Protocol for Mobile Ad Hoc Networks", European Journal of Scientific Research, Vol.49, No.1, pp. 6-17.
- [7] Jailani Kadir, Osman Ghazali, Mohamed Firdhous, Suhaidi Hassan, (2011) "Node Selection Based on Energy Consumption in MANET".
- [8] V. Berlin Hency, S. Aravind Prasad, Y.RA. Kannan and D. Sridhar an, (2011) "A Novel Approach For Handover Avoidance And Qos Improvement In Wlan". International Journal of Distributed and Parallel Systems, Vol.2, No.5, pp. 77-86.
- [9] Mohammad Faisal, Muhammad Nawaz Khan, (2012) "An enhanced Scheme for Reducing Vertical handover latency", (International Journal of Advanced Computer Science and Applications, Vol. 3, No.1, pp. 100-105.
- [10] R. Venilla, V. Duraisamy, (2012) "Qos based clustering technique for multicast security in MANET", European Journal of Scientific Research, Vol. 81, No. 1, pp. 33-46.
- [11] R. Deepa, S. Swamynathan, (2012) "Load Balancer for Energy Efficient Clustering Protocol in MANETs", International Conference on Advances in Computing, Communication and Informatics, pp. 269-275.
- [12] P. Kuppasamy, K. Thirunavukkarasu, B. Kalaavathi, (2012) "Cluster Based Cooperative Caching Technique in Mobile Ad Hoc Networks", European Journal of Scientific Research, Vol. 69, No. 3, pp. 337-349.

AUTHOR

Nidhi Parashar has done Master in Technology in Computer Engineering. She is presently working as Senior Lecturer in Department of Information Technology at M.I.T., Meerut, U.P. During academic session she has coordinated and attended various National Conferences and Workshops at university level. Her research area includes MANET (Mobile Ad-hoc Network), Wireless Sensor Network (WSN) and Congestion control.

