CHAPTER 5
ANALYSIS OF AHPC PROTOCOL

5.1 Introduction

In addition to the experimental analysis of AHPC protocol, an extensive mathematical analysis of AHPC protocol has been carried out to analyze the stability, control overhead, time complexity and message complexity. The sufficient lemmas have been provided and a theorem has been given to prove that the novel AHPC protocol forms a valid and stable cluster structure in mobile ad hoc networks. The analysis of control overheads have been carried out mathematically for Hello packets transmitted by the nodes, cluster formation phase and cluster maintenance phase of AHPC protocol. The time complexity and message complexity of AHPC protocol has been analyzed mathematically for the cluster formation phase. The section 5.2 deals with the stability analysis of AHPC protocol and section 5.3 show the analysis of control overheads of AHPC protocol. The section 5.4 shows the analysis of time and message complexity of AHPC protocol.

5.2 Stability Analysis of AHPC Protocol

The AHPC protocol can be stable if it adapts itself to the dynamic changes of the network topology through minimal cluster structure update. The stability of the cluster can be evaluated using significant metrics viz. Cluster head changes and Cluster head lifetime.

The frequent cluster head change leads to route invalidation, frequent re-affiliation, dominant set updates and re-clustering, resulting in rebuilding the whole network causing instability. The proposed AHPC protocol drastically reduces the
frequent cluster head change through computation of relative weights and utilization of CCI timer.

The increase of cluster head lifetime leads to the increase of stability of the network. Since more stable cluster heads are chosen using AHP methodology, the cluster head lifetime increases extensively leading to the stable network.

5.2.1 Proof of Correctness

The stability of AHPC protocol is proved with the formation of stable and valid cluster architecture guaranteed by the following ad hoc clustering properties [61, 41].

1. Every node has at least a cluster head as neighbor (dominance property)
2. Every ordinary node is at most one hop away from its cluster head
3. No two cluster heads can be neighbors (independence property)

Lemma 1: Every node has at least a cluster head as neighbor

Proof:

The Master node selects the cluster heads by traversing the Sort_List data structure. Each node in the network is assigned a role of cluster head or cluster member while traversing the Sort_List till the end. The highest weight node \( X \) is selected as a cluster head and its neighbors \( m(X) \) become its members to form a cluster \( \{X, m(X)\} \). The \( m(X) \) nodes are removed from the Sort_List to restrict their participation in further cluster head selection. Then, the next highest weight node \( Y \) is selected from the Sort_List and forms a cluster \( \{Y, m(Y)\} \) with its neighbors \( m(Y) \).

This procedure is repeated until all the nodes in the network are either a cluster head or a member. If a node is with undecided state, cluster formation phase is invoked to initiate cluster formation in the network. Therefore, every node in the network is
either a cluster head or neighbor to a cluster head. Thus, every node has at least a cluster head as neighbor to have a stable and valid cluster structure ■

**Lemma 2: Every cluster-member node is at most one hop away from its cluster head**

**Proof:**
Master node selects node $X$ as cluster head and forms cluster \( \{X, m(X)\} \) with nodes $m(X)$ that lie within the transmission range of node $X$. The member nodes, $m(X)$ that lies within the transmission range of node $X$ takes at most one hop to communicate directly with a cluster head. Thus, every member node is at most one hop away from its cluster head to provide fast intra and inter cluster communications. ■

**Lemma 3: No two cluster heads can be neighbors**

**Proof:**
If a cluster head $X$ receives *Hello* packets from another cluster head $Y$ that resides within its transmission range, it sets its CCI timer and check whether node $Y$ moves out of its range within the predefined timer interval. If $X$ realizes that cluster head $Y$ resides within its range even after the timer expires, it compares its weight, $X_w$ with that of cluster head $Y$’s weight, $Y_w$. If $X_w > Y_w$, then cluster head $X$ declares itself as a cluster head by broadcasting *Head-Declaration packet* to its neighbors. The cluster head $Y$ resigns its role and becomes a member of cluster head $X$. If $X_w < Y_w$, then cluster head $X$ broadcast *Cluster-resign packet* to resign its role and becomes a member of cluster head $Y$. Thus, no two cluster heads can be the neighbors. ■
Theorem:

A stable and valid clustered architecture is formed in mobile ad hoc networks using AHPC protocol.

Proof:

The correctness of theorem is immediate from Lemmas 1-3. ■

5.3 Analysis of Control Overheads

The control overheads of AHPC protocol have been analyzed mathematically for Hello packets transmitted by the nodes, cluster formation phase and cluster maintenance phase of AHPC protocol.

5.3.1 Hello Overhead

Let \( N \) be the total number of nodes in the network and \( m_i \) be the number of neighbors of \( i^{th} \) mobile node. The Hello packets transmitted by \( N \) nodes are \( \sum_{i=1}^{N} m_i \).

Let \( m_{it} \) be the number of neighbors of \( i^{th} \) mobile node at \( t^{th} \) interval.

The Hello packets transmitted by \( N \) nodes for \( S \) simulation time (contains \( S/t \) intervals)

\[
= \sum_{t=1}^{S/t} \sum_{i=1}^{N} m_{it}
\]

Let Master node periodically sends Hello packets at \( t \) intervals after \( v \) seconds from the start of the simulation to all the \( H \) cluster heads in the network.

The number of intervals the Hello packets sent by the Master node \( = \frac{(S - v)}{t} \).

Total overhead due to Hello packets, \( O_h \) is derived as

\[
O_h = \sum_{t=1}^{S/t} \sum_{i=1}^{N} m_{it} + \sum_{i} H_i
\]
Where $O_h = \text{Total } Hello \text{ packets sent by the Master node and all the nodes during } S \text{ simulation time.}$

### 5.3.2 Cluster Formation Overhead

Let the Master node sends request messages to all $N-1$ nodes to provide information about their node parameters in the network. On receiving the request message from Master node, the $N-1$ mobile nodes respond back with their information. Therefore, $2(N-1)$ messages are required to obtain node’s parameters of all the nodes in the network. The Master node selects $H$ number of cluster heads and sends *Head-intimation* messages to all the cluster heads in the network. Each cluster head broadcast *Head-declaration* messages to its neighbors. Let $m_i$ be the neighbors of $i^{th}$ cluster head in the network. The cluster formation overhead, $O_f$ has been derived mathematically as

$$O_f = \sum_{i=1}^{H} m_i + [H + 2(N-1)]$$

### 5.3.3 Cluster Maintenance Overhead

The cluster maintenance overheads have been derived mathematically for all the four scenarios viz. Master signs off or disappears, Cluster head signs off or disappears, New node joins or leaves the network, Merging of cluster heads that occur during the cluster maintenance phase. The control overheads incurred during each scenario have been derived mathematically.

**AHPC-Scenario1: Master node signs off or disappears**

This scenario involves two events viz. Master node sign off and disappears that may occur during the operation of the network. The control overheads incurred during each event in this scenario have been derived mathematically.
a. Master node signs off:

The Master node nominates next highest weight node and sends a "Master-intimation" packet to that node. The nominated Master node sends "Master-declaration" packet to all the \(N - 2\) nodes in the network. The overhead due to Master node signs off, \(O_{ms}\) is found to be

\[ O_{ms} = N - 1 \]

b. Master node disappears:

Let cluster head \(C\) realizes the disappearance of Master node. It broadcasts "Master-disappear" packet to its \(C\_n\) neighborhood cluster heads. Let \(y\) be the number of "Disappear-ack" packets received by node \(C\). Let \(x\) "Disappear-ack" packets are the upper limit of \(C\) and \(O_{md}\) be the Master-disappear overhead. If the cluster head \(C\) receives "Disappear-ack" packet from more than \(x\) neighborhood cluster heads, it requires two messages for retrieving node ID of the highest weight cluster head from the Backup node. The "Master-intimation" packet is send to the highest weight cluster head to declare itself as a Master node and the Master node broadcasts "Master-declaration" message to all the \(N - 2\) nodes in the network. The overhead due to Master node disappears, \(O_{md}\) is found to be

\[
O_{md} = \begin{cases} 
(C_n + y) & \text{if } y < x \\
(C_n + y) + (N + 1) & \text{if } y \geq x
\end{cases}
\]

AHPC-Scenario2: Cluster head node signs off or disappears

This scenario involves two events viz. Cluster head node sign off and disappears that may occur during the operation of the network. The control overheads incurred during each event in this scenario have been derived mathematically.
a. Cluster head signs off:

The cluster head nominates next highest weight member node as cluster head and sends *Head-intimation* packet to that node. The nominated cluster head broadcast *Head-declaration* packet to its $m$ neighbors and sends *Update-intimation* packet to the Master node. The overhead incurred due to cluster head signs off, $O_{cs}$ is found to be as

$$O_{cs} = m + 2$$

b. Cluster head disappears:

If member L of a cluster head realizes the disappearance of its cluster head, it broadcast *Head-disappear* packets to its $m$ neighbors. Let $y$ be the number of *Disappear-ack* packets received by node L. Let $x$ *Disappear-ack* packets be the upper limit of L. If the *Disappear-ack* packets received by node L are more than the upper limit, node L informs the Master node with the *Head-disappear* packet. The Master node selects new cluster head $s$ and sends *Head-intimation* packet to that node. The new cluster head, $s$ that broadcasts *Head-disappear* packet to its $m_s$ neighbors and sends *Update-intimation* packet to the Master node to inform its neighbors. The overhead due to cluster head disappears, $O_{cd}$ is found to be

$$O_{cd} = \begin{cases} (m + y) & \text{if } y < x \\ (m + y) + (m_s + 3) & \text{if } y \geq x \end{cases}$$

AHPC-Scenario3: New node joins or leaves the network

When a new node joins the network, the cluster head that receives its *Hello* packets add it as its member and it has been observed that there is no overhead if a new node joins the network. But, if a node leaves the network, its cluster head communicates with the Master node with an *Update-intimation* packet. Since only
one message has been sent in this scenario, the overhead incurred due to node leaves, $O_i$ during this scenario has been derived as $O_i = 1$

**AHPC-Scenario4: Merging of cluster heads**

The cluster head that first realizes the incoming cluster head in its transmission range requires two messages to retrieve the weight of the incoming cluster head from *Backup* node. The cluster head whose weight is higher declares itself as a cluster head through *Head-declaration* packet to its neighbors and the other cluster head broadcast *Cluster-resign* packet to its neighborhood with an *Update-intimation packet* sent to the Master node. Let $m_j$ be the neighbors of $j^{th}$ cluster head. The number of cluster heads involved in merging is 2. The overhead incurred due to merging of clusters, $O_m$ has been derived as

$$O_m = \sum_{j=1}^{2} m_j + 3$$

**5.4 Analysis of Time and Message complexity**

The time and message complexity of AHPC protocol has been analyzed mathematically for the cluster formation phase.

**Time Complexity**

Each mobile node assumes $T_{static}$ time period until the cluster formation is initiated. Hence, number of time steps taken by each node before it could decide to be a cluster head or a member is at least $T_{static}$. When Master node selects the cluster heads in the network, it takes one time step to communicate with the nodes that are chosen as cluster heads. The node that receives *Head-intimation packet* from the Master node takes one time step to declare itself as a cluster head to its neighbors. Therefore, the time complexity, $T$ of AHPC has been derived as $T \leq T_{static} + 2$. 
Message Complexity

The Master node sends request to all the nodes in the network to provide information about their node’s parameters. As there are \( N \) nodes in the network, the Master node sends \( N-1 \) request messages to all the nodes and each node responds to its request by sending their node parameters to the Master node. Therefore, \( 2(N-1) \) messages are required for the computation of relative weights of the mobile nodes. After the computation of weights, the Master node selects \( H \) number of cluster heads in the network. The Master node sends \textit{Head-intimation packet} to the entire selected cluster heads to declare itself as a cluster head. Therefore, \( H \) cluster heads broadcast \textit{Head-declaration packet} to its neighbors. Therefore, the message complexity, \( M \) of AHPC has been derived as \( 2(N + H - 1) \).

5.5 Conclusion

Extensive mathematical analysis of AHPC protocol has been carried out for the cluster formation phase and cluster maintenance phase. The stability analysis of the AHPC protocol has been carried out. The Lemmas and theorem proves that novel AHPC protocol forms stable and valid clustered architecture in the network. The control overhead analysis has also been carried out to determine the control overheads incurred during the cluster formation phase and cluster maintenance phase of the AHPC protocol in the network. In addition, the time complexity and message complexity has also been evaluated for cluster formation phase of AHPC protocol. It has been observed that time complexity and message complexity is relatively lesser than other protocols viz. MOBIC, MobDHop, WCA, Max-Min D-Cluster clustering algorithms.