A Distributed Stable Backbone Maintenance Protocol for Ad Hoc Wireless Networks

Izhak Rubin
University of California, Los Angeles
Electrical Engineering Department
Los Angeles, CA 90095-1594
310-825-2326
rubin@ee.ucla.edu

Xiaolong Huang
University of California,
Los Angeles
todhuang@ee.ucla.edu

Y.-C. Liu
University of California,
Los Angeles
ychliu@ee.ucla.edu

Huei-jiun Ju
University of California,
Los Angeles
hju@ee.ucla.edu

Abstract—**We have recently introduced a hierarchical structure for ad hoc wireless networks that classifies nodes into Backbone Capable Nodes (BCNs) and Regular Nodes (RNs). Under our TBONE protocol, a backbone network (Bnet) is formed by dynamically electing Backbone Nodes (BNs) among BCNs. However the current TBONE protocol requires global topological information to elect and de-elect BNs, which can induce high control message overhead and slow down the Bnet layout adaptation process. In this paper, we resolve this problem by proposing a modified MBN Protocol (MBNP) for electing and de-electing BNs, which requires each candidate node to employ only local connectivity information (within two hops). While such schemes tend many times to be unstable, we prove and demonstrate that our process involving BN-BCN conversions is oscillation free. We show that the synthesized network configuration demonstrates desirable robustness and connectivity features, while demanding low control message overhead. Key performance characteristics of the modified protocol are exhibited by conducting simulation-based evaluations.**

1. INTRODUCTION

We have recently been investigating the operation of mobile wireless networks through the embedded establishment of a Mobile Backbone Network (MBN). In [1], we have defined the structure and elements of mobile backbone based ad hoc wireless networks and presented a protocol (TBONE) for the synthesis of MBNs. In [2]-[3], we have examined several approaches pertaining to the topological synthesis of mobile backbone networks. A mobile backbone network consists of a backbone network (Bnet), access nets (Anets), and regular (flat) ad hoc network(s). Its structure is illustrated in Figure 1. Thick solid lines connecting large solid circles represent the Bnet. Dashed ovals consisting of thin solid lines connecting small solid circles represent the Anets. The small solid circles and the thin dashed lines connecting them to each other represent the regular ad hoc network. In this paper, we concentrate on protocols governing the operation of the MBN's Bnet. The MBN is designed so that it involves a sufficient but not excessive number of backbone nodes, while providing high coverage, so that a high fraction of the low power nodes can access at least a single Backbone Node (BN) through 1 hop.

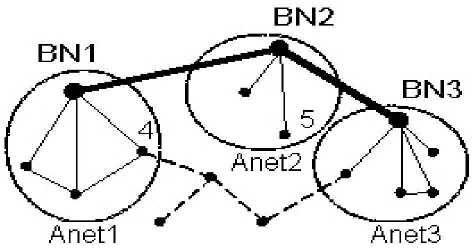


Fig. 1- The decomposition of a mobile backbone network into a Bnet , three Anets, and a regular ad hoc network with h=1 and k=1

We identify two key classes of elements in our MBN system: high capacity nodes and low capacity nodes. High capacity nodes can serve as either Backbone Nodes (BNs) or Backbone Capable Nodes (BCNs). They have high storage and processing resources, and include low power and high power radio modules that allow them to transmit/receive data and control messages in two different frequency bands simultaneously. The low power radio module is used to communicate to nodes that are in low-power communications range while the high power radio module is employed by a BN to communicate to its peer BNs (across the Bnet, spanning longer distances and operating at higher data rates). When a high capacity node is serving as a BCN, it will continuously and simultaneously listen for messages transmitted in its reception range across the two frequency bands. However, it *only* uses its high power radio module for receiving and transmitting control packets. A BCN has the capability to be converted into a BN, when selected to act as a BN by our protocol. Low capacity nodes act merely as Regular Nodes (RNs). RNs are limited in their power, storage, processing and communications assets and only employ a single low power radio module.

In this paper, we propose a modified MBN protocol (MBNP). This protocol includes the RN association algorithm, and the BN-to/from-BCN conversion algorithm. Through these three algorithms, we try to reduce the number of BNs to maintain Bnet connectivity under varying mobility conditions. The performance and stability of our protocol is shown by simulation.

The later sections are organized as follows: In Section 2, a new BCN-to-BN conversion and BN-to-BCN conversion protocol are designed. In section 3, we prove that our proposed protocols are oscillation free. In section 4,

---

[1] This work was supported by Office of Naval Research (ONR) Contract No. N00014-01-C-0016, by the Army Research Office (ARO) under Contract No. DAAG55-98-1-0338, and by the National Science Foundation (NSF) under Grant No. ANI-0087148.

simulation results are given to show the stability of the protocol. We finally conclude in Section5.

2. BCN-BN CONVERSION AND BN-BCN CONVERSION PROTOCOL

We are initially faced with a regular ad hoc network composed of Backbone Capable Nodes (BCNs) and Regular Nodes (RNs). Backbone Nodes (BNs) should be selected among Backbone Capable Nodes gradually to form a Mobile Backbone Network. Some of BNs should be deselected to be BCNs to reduce the network resource consumption, such as energy and control message overhead.

A BCN-to-BN (also denoted as BCN-BN) conversion algorithm is introduced in [1] to elect BNs among BCNs. In the proposed BCN-BN conversion algorithm, an unassociated BCN periodically broadcasts its node ID and dynamic weighted label to its one hop neighbors using its low power transmission module. An unassociated BCN will convert itself to a BN, if its dynamic weighted label is the highest among all its unassociated BCN neighbors. A BCN will also be converted to a BN, if there is no finite hop path between two of its neighbors. The purpose of the conversion is to connect disjoint backbone components. But unfortunately, to check whether there is a finite hop path between two nodes requires the node be aware of the global topology information, which induces significant control message overhead and makes the backbone network less adaptive to the topology change.

In this paper, both BN-to-BCN (also denoted as BN-BCN) as well as BCN-to-BN (also denoted as BCN-BN) conversion protocols are introduced, that only require a node be aware of local topology information within two hops for the decision of BCN-to-BN and BN-to-BCN conversions. These two protocols work concurrently to prevent the occurrence of frequent BN-to-BCN and BCN-to-BN oscillations and to reduce the number of required active BN nodes (thus contributing to energy conservation).

We start by defining the new BCN-to-BN conversion protocol.

BCN-to-BN Conversion Protocol

Definition 1: There is said to be an n hop path between bn_a and bn_b, if there exists a route $[bn_a, bn_0, bn_1, ..., bn_k, bn_b]$, where $k=n-2$. The path length is said to be n.

Under the BCN-to-BN conversion protocol, every BCN node periodically broadcasts *hello* messages to its one-hop neighbors. For instance, a BCN bcn_1 periodically reads its neighbor list ngl_1. This list is composed of the low power RN neighbors of bcn_1 and the high power BCN/BN neighbors of bcn_1. Node bcn_1 will convert itself to a BN if any of the following conditions is satisfied:

1. There exists an RN neighbor $rn_i \in ngl_1$ which has no BN neighbors.
2. All three conditions are satisfied:
    a) bcn_1 is unassociated.
    b) bcn_1 has at least one unassociated BCN neighbor.
    c) bcn_1 has the highest dynamic weighted label among all its BCN neighbors.
3. The following two conditions are satisfied:
    a) bcn_1 is associated.
    b) There exists a BCN neighbor $bcn_i \in ngl_1$, which has no BN neighbors.
4. There exist two BN neighbors, bn_a and bn_b, where there is NO path between bn_a and bn_b, whose path length m is smaller than or equal to $d1$, so that $m \leq d1$

Condition 1 forces a BCN to be converted to a BN when it has at least a single RN neighbor that cannot associate with any current BNs. This condition expands the backbone node population to admit (i.e., associate) more unassociated RNs into the MBN.

Condition 2 provides for the conversion of a BCN into a BN from a group of unassociated BCN neighbors. This condition is identical to a corresponding one specified for the previously proposed BCN-to-BN conversion algorithm.

Condition 3 forces an associated BCN to convert to a BN when at least one of its BCN neighbors cannot associate with any BN. This condition expands the backbone node population to connect (i.e., associate) more unassociated BCNs into MBN.

Condition 4 induces a BCN to convert to a BN, when the path length between at least two of its neighbors is greater than $d1$. As a special case, When the BCN has two BN neighbors that belong to separate network components (i.e., to disconnected sub-networks, so that there exists no path connecting these two BNs), the converted node will serve to connect these two components. This conversion from BCN to BN improves the MBN's topological connectivity by adding a new (shorter) path, as well as serves to potentially decrease the network's graph diameter.

Figure 2 shows the scenarios for the four conditions.

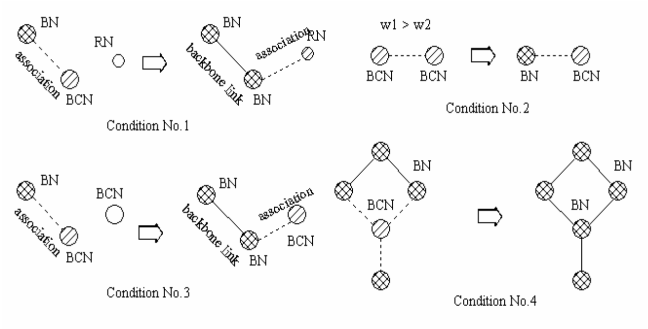


Fig.1 BN Election Conditions

When $d1$ is set to be 2, only local topology within 2 hops is required to check the condition. For the proposed Condition 4 to work properly, a BCN only needs to obtain the neighbor list of neighbors to test Condition 4. Consequently, neighbor list messages are only locally broadcasted, thus avoiding wider scope flooding of these messages (and consequently reducing the control overhead level).

In the following, we present the Local Connectivity Detection Algorithm that is used to test for the occurrence of events that induce Condition 4 to hold.

Local Connectivity Detection Algorithm (LCD)

```
Begin
Find all the neighbors of node n1;
    if ( no neighbor )
        return( Condition 4 false )
    else    for ( each of n1's neighbor ni )
                add ni and ni's neighbor to an array Ai
```

```
    if ( more than one neighbor of $n1$ )
        if ( each pair of arrays, $Ai$ and $Aj$, have at least one common member )
            return(Condition 4 true)
        else return(Condition 4 false)
    else return (Condition 4 true)
END
```

BN-to-BCN Conversion Protocol

A BN-to-BCN conversion protocol is designed in this paper to reduce the population of BNs in the MBN, while at the same time maintaining the connectivity of the backbone network. In general, a reduction in the BN population will lead to lower energy consumption. In the following, we present our modified BN-to-BCN conversion algorithm, as well as prove that it does not lead to an oscillatory BN-to/from-BCN conversion process.

BN-to-BCN Conversion Protocol

The following messages are transmitted by a BN under the BN-BCN conversion protocol.

Message *alert*: This message contains the ID of the BN. It is flooded to its 2-hop neighbors to alert other BNs that a BN-BCN conversion process is planned.

ID field *conversionID*: The ID field is to inform the BN whether it is legal to convert.

Timer *tmr*: The timer is used to delay the conversion of a BCN.

The algorithm is outlined below:

1. Begin
2. A BN bn_1 will check whether the following conditions are simultaneously satisfied.
    a) Condition 1: Each low power RN neighbor of bn_1 has at least one BN neighbor in addition to bn_1.
    b) Condition 2: One of the low power neighbors of bn_1 is a BN.
    c) Condition 3: Each low power BCN neighbor of bn_1 has at least one BN neighbor in addition to bn_1.
    d) Condition 4: For each pair of two high power BN neighbors of bn_1, say bn_a and bn_b , once this BN is converted, there is a path between bn_a and bn_b, whose length m is smaller than or equal to $d2$ hops, $m \le d2$. In our protocol, $d2$ is set to be 2.
3. If all the conditions stated above are satisfied, then:
    a) Set a timer to an expiration time of *tmr* to be large enough to permit sufficient time for the reception of *alert* messages from BNs that are two hops away.
    b) Set *conversionID* to be bn_1.
    c) Send an *alert* message.

    Otherwise, stop the conversion process.
4. While the time for *tmr* has not expired, set *conversionID* to be bn_2, upon receiving an alert message from bn_2, whose ID has a greater value than the ID value of bn_1.
5. If *conversionID* displays the ID of bn_1 at the end of the *tmr* period and Condition 3 holds, bn_1 converts itself to a BCN. Otherwise, the process is canceled.
6. End

In the following, we explain the above stated BN-to-BCN conversion algorithm. When bn_1 intends to convert itself to a BCN, an *alert* message with a hop count 2 is sent, it is marked with the ID bn_1. A timer is set to an expiration time *tmr* that is large enough to permit sufficient time for the reception of *alert* messages from BNs that are two hops away. If an *alert* message is received by bn_1, whose ID field value is greater than the ID field value of bn_1, the conversion process is canceled to prevent simultaneous conversions.

Condition 1 ensures that this BN will be potentially able to associate with another BN, if this BN is converted to a BCN.

Condition 2 and 3 ensures all the RN and BCN neighbors of this BN will be potentially able to associate with another BN, if this BN is converted to a BCN.

Condition 4 is the main trigger of the BN-BCN conversion process. The objective of Condition 4 is to ensure that once this BN is converted, the distance between any two neighboring BNs of this BN will not degrade below its current level (which is equal to 2, as they are connected to each other through this BN). We prove in a later session that Condition 4 ensures that BCN-to-from-BN conversions do not exhibit oscillatory behavior. Condition 4 is tested by using the Local Connectivity Algorithm presented in the previous subsection.

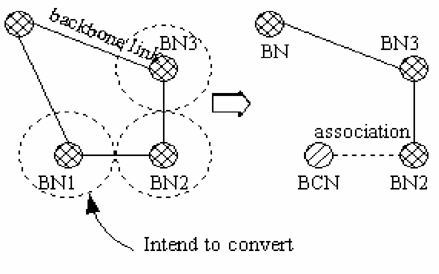


Fig.3 BN-BCN Conversion

The timer and the alert message avoid uncoordinated simultaneous BN-to-BCN conversions to occur in the following case. Figure 3 illustrates a case under which BN1, BN2, and BN3 consider at the same time to convert to BCNs. Through the coordinated use of *alert* messages, only one of the BNs will actually decide to convert to a BCN.

3. BN TO/FROM BCN OSCILLATIONS

Definition 2: We represent the backbone network topology as a graph $G(V, E)$, where V is the set of all the BNs in the MBN and E is the set of all the high power links between two neighbor BNs.

Definition 3: A node is said to experience BN to/from BCN oscillations if, over a finite period of time during, the node repeatedly converts between BN and BCN roles.

Definition.4: A protocol that results in no BN to/from BCN oscillations is said to be an oscillation free protocol.

Because of the distributed nature of the protocol, network BN and BCN nodes determine whether they should initiate a conversion process by using local status information, as they do not have access to global information and to performing network-wide cooperation. Consequently, a larger than desired number of BCNs (needed to sustain backbone network connectivity) may convert themselves to BNs. Subsequently, some of these BNs may be declared redundant (to reduce the number of BNs, improving energy conservation) and thus convert back to BCNs. This process can repeat, leading to oscillations. The BCN-to/from-BN conversion protocol should prevent such BN-to/from-BCN oscillations, leading to a backbone network virtual topology that is stable under static conditions (so that the conversion algorithm converges in a finite number of steps). In the following, we prove that our proposed BCN-to/from-BN conversion protocol with the preserved window phase is oscillation free.

Lemma 1: Given a stationary network topology, the BN-BCN conversion protocol presented above does not trigger the reverse conversion of this node to a BN, and it does not trigger the conversion of another BCN to a BN, when $d1 \geqslant d2=2$.

Proof: Assume node $n1$ converts from BN to BCN.

Because Condition 1 of BN-BCN conversion for $n1$ holds, Condition 1 of BCN-BN conversion for $n1$ does not hold. Because Condition 2 of BN-BCN conversion for $n1$ holds, Condition 2 of BCN-BN conversion for $n1$ does not hold. Because Condition 3 of BN-BCN conversion for $n1$ holds, Condition 3 of BCN-BN conversion for $n1$ does not hold.

Because Condition 4 of BN-BCN conversion for $n1$ holds, the distance of the path between any two BN neighbors of $n1$, say n_i, n_j, should be less than or equal to 2 ($d2$=2) without the aid of $n1$. Assume two BN neighbors of BCN node $n2$ are n_x, n_y. If the distance between n_x and n_y before $n1$'s conversion is $m1$, after $n1$'s conversion, for the distance between n_x and n_y, say $m2$, we have $m2 \leq m1$. It means, if $m1 \leq d1$, we have $m2 \leq d1$. Thus when $d1 \geqslant d2=2$, Condition 4 of BCN-BN conversion for $n1$ and $n2$ does not hold. ■

Theorem 1: Given a stationary network topology, the two-hop based BCN to/from BN conversion protocol is oscillation free.

Proof: Given a stationary network topology, consider every high capable node separately. For each high capable node, if it is a BN, we initialize a Sufficient Set (We simply call it 'set' in this proof.) which includes this BN and its two-hop BN neighbors. If it is a BCN, we initialize the same set when it firstly becomes a BN.

The set grows based on the following rule:

If there is BN to/from BCN conversion triggered by nodes, some of which are outside the set, those nodes will be added to the set. If there is one node in the set converted and triggered by some nodes outside the set, from the conversion of one node to the end of these new nodes added to the set, we call this period *update period*. For the conversion not triggered by the nodes outside the set, we call it *internal conversion*.

Before we proceed, we proof the lemma 2.

Lemma 2: Between any two update periods of the set, if a node in the set converts from BN to BCN, it will not convert back to BN.

Proof: Between any two update periods of the set, by lemma 1, any internal BN to BCN conversion will not trigger any internal BCN to BN conversion. If a node converts from BN to BCN, it must satisfy the criterion that all its neighbors are two-hop connected. Since there is no new node added to the set, the only reason that this node converts back to BN is there are two neighbors are not two-hop connected. This implies, there is some BN between these two neighbors converts to BCN. By the BN to BCN conversion criteria, this conversion is possible only when the two neighbors are two-hop connected. By contradiction, we proved the statement of lemma 2.

Now we are going to prove that between two update periods, the nodes inside the set will stay to be either BN or BCN in finite time. In each set, we create three subsets: old_BCN_set includes all the BCNs at the end of each update period; BN_set includes all the BNs; new_BCN_set includes those BCNs converted from BNs between the two periods. By lemma 2, between any two update periods, any BN converts to BCN will not convert back. Thus the size of old_BCN_set is non-increasing and the size of new_BCN_set is none-decreasing. In finite period of time, the sizes of old_BCN_set and new_BCN_set become constants, which mean the total number of BCN becomes a constant. Also the number of BN becomes a constant after finite time. Since BN to/from BCN conversion is a discrete event, after each finite time, each node in the set will stay to be BN or BCN.

Since the total number of high capable nodes is finite, thus after finite time, the set will not update any more. Also we proved that between after each update period, in finite time, the nodes in the set will be stabilized. Combine these two statements, we conclude that after finite time, the numbers of BCN and BN will become constants.

For each node, we can generate a set, which has the same property above. Thus after finite time, every node in the network stays as BN or BCN and will not change any more. So the protocol is oscillation free.

Corollary 1: Given that initially no BNs are present, and assuming the topology is static (i.e., no location changes), the final status of high capacity nodes is reached in a finite period of time.

Proof: Initially, the cardinality of the oldBCN is equal to C and the BN population size is equal to zero. Since Theorem 1 shows that for a static network topology, every node will reach after a finite period of time its final status as a BN or BCN regardless of the initial condition, the stated result clearly holds for the special initial state under which no BNs are present. ■

These results prove that the backbone maintenance protocol presented here induces a stable (BN-to/from-BCN conversion process based) operation.

4. SIMULATION

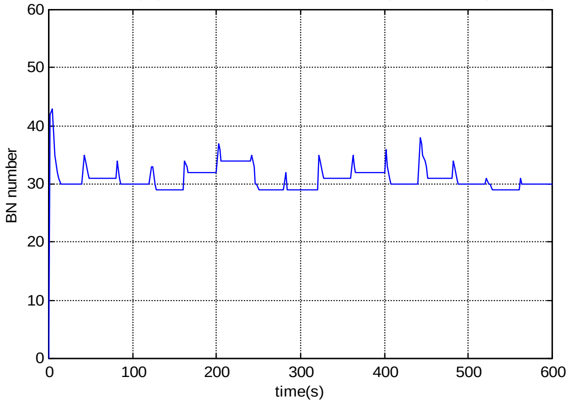
Our simulation is conducted for testing oscillation property of the proposed protocols and the protocol overhead. We set $d1=d2=2$, so that only local topology information is required for BCN-BN conversion and BN-BCN conversion. The simulation parameters are as follows.

Area	Low power range	High power range	Connectivity update time
1600m×1600m	250m	500m	1s

A scenario that contains 60 BCNs and no RNs is examined. The topology is changed every 40 seconds, while maintained to be static within the 40 seconds. The BN population under maintenance protocol is plotted in Chart 1.

It shows the BN population is always oscillation free for each static period.

Chart 1. BN population selected from 60 BCNs (maintenance protocol)



Under the scenario that contains 60 BCNs and no RNs, the conversion rate of BN to/from BCN per node versus the average mobility speed level is examined. The result is shown in Chart 2. It shows that under maintenance protocol, when the mobility level drops, the conversion rate per node is reduced approaching to zero.

Chart 2. Conversion rate vs. Mobile Level

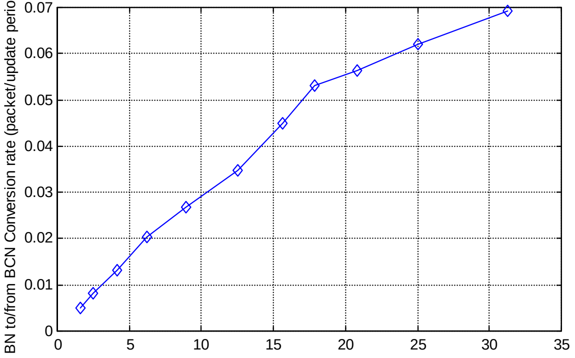
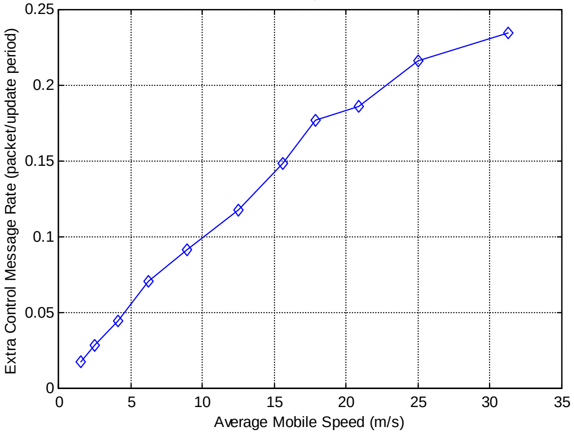


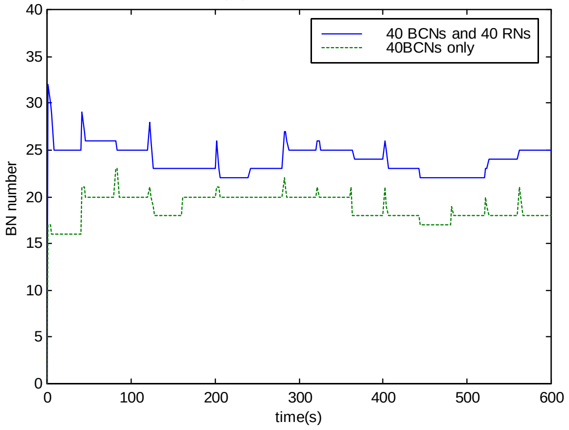
Chart 3. Extra Control Message Overhead vs. Mobile Level



The excess control overhead (defined as the protocol's control overhead rate excluding the periodically conducted connectivity updates at a rate of 1 'hello' message /sec) of the protocol per node versus the average mobility speed level is examined under a scenario that contains 60 BCNs and no RNs. The result is shown in Chart 3. It shows that when the mobility level drops, the excess nodal control overhead of the protocol drops approaching to zero.

Another scenario with 40 BCNs is examined to show the influence of bring RNs into the topology. The result is sown in Chart 4. When 40 RNs are added into the topology, more BCNs are required to convert themselves to BNs, thus RNs can associate to BNs to connect to the backbone network.

Chart 4. BN population selected from 40 BCNs



5. CONCLUSION

In this paper, a stable backbone maintenance protocol is proposed for mobile backbone network (MBN) based ad hoc wireless network architectures. This protocol serves to elect Backbone Nodes (BN) among Backbone Capable Nodes (BCN) and convert BNs back to BCNs to reduce the active BN population, leading to energy and BN resource conservation. The conversion from BCNs to BNs is used to construct backbone networks that have desired connectivity features. The conversion from BNs to BCNs is used to reduce the BN population, as noted above, while maintaining the desired connectivity level of the backbone network. Our new protocol is introduced and shown to yield an oscillation free conversion process, for a static topological configuration.

To implement a simpler and faster adaptive distributed operation for the conversion process, which also induces lower control overhead rates, the protocol presented in this paper requires each node to collect connectivity information from nodes that are only two hops away. For a greater reduction in the number of actively elected backbone nodes, the network designer may wish to allow weaker requirements for connectivity degradations that may occur under such conversions. Such protocols are similarly specified, and are currently under investigation.

REFERENCES

[1] I. Rubin, A. Behzad, R. Zhang, H. Luo, E. Caballero, "TBONE: A Mobile-Backbone Protocol for Ad Hoc Wireless Networks," *IEEE AEROSPACE 2002*

[2] I. Rubin and P. Vincent, "Topological Synthesis of Mobile Wireless Networks," *IEEE MILCOM Conference*, session on Digital Battlefield Communications,2000

[3] I. Rubin and P. Vincent, " Topological Synthesis of Mobile Wireless Networks for Managing Ad hoc Wireless Networks," *Proceedings IFIP/IEEE International Conference on Management of Multimedia Networks and Services (MMNS 2001),*