



**Carleton**  
**UNIVERSITY**

*Computer Science COMP 4905 - Honours  
Project*

**Backbone Network Construction in Ad Hoc Wireless  
Networks**

Author: Dianshu Hu  
Student#: 100307865  
Advisor: Dr Michel Barbeau  
Faculty: Science  
Department: Computer Science  
Date: April 5, 2007

## Abstract

Ad hoc wireless network is a special type of wireless network. It consists of mobile hosts with wireless network interface. These hosts could form a network without central administration. In ad hoc network, one of the most important problems to solve is efficient routing. Hierarchical routing is usually used in such network. Connected-Dominating-Set-Based routing is a promising approach. In my honours project, I attempt to find the most efficient and scalable connected dominating set (CDS) construction algorithm proposed in the recent five years. In my study, six recently proposed algorithms for constructing CDS in ad hoc wireless network are reviewed. Two of them are selected based on the following criteria: (1) The CDS constructed converge in constant time, regardless of how many nodes are in the given topology or the network density. (2) The size of CDS generated is bounded by  $R|MCDS|$  where  $MCDS$  is the minimum connected dominating set in a given topology and its size is upper bounded by a constant  $O(A/r^2)$  where  $A$  is the area of the 2D space and  $r$  is the transmission range of mobile nodes, and  $R$  is the constant approximation ratio. If no constant bound on  $R$  has been proven, at least there exists a probabilistic bound on the average value of  $R$ . (3) Low message complexity, which is the size of hello message that each node broadcast during the formation of CDS should be small. Finally, the chosen two are implemented and their performances are evaluated via simulations in static network. The results of simulations show the superiority of ETSA [1] in ad hoc with very limited bandwidth and high node density. On the other hand, if the size of resultant is emphasized instead of low communication overhead, then algorithm Wu [8] is better.

## **Acknowledgment**

My sincere appreciation goes to Dr Michel Barbeau for inspiring me to find this interesting open topic and providing all necessary assistance.

I thank Jie Wu, a senior member of IEEE. In his research paper [8] I learn the most. I would also like to thank my intimate friends for giving me continuous supports.

# Table of Contents

Abstract.....	1
Acknowledgment .....	2
Table of Contents.....	3
List of Figures.....	4
List of Diagrams .....	8
List of Tables .....	10
Text.....	11
1 Introduction.....	11
2 CDS Based Routing.....	12
3 General knowledge about CDS.....	14
4 CDS construction algorithms.....	16
1) ETSA .....	19
2) d-SPR algorithm .....	24
3) Span .....	25
4) Wan’s algorithm .....	26
5) Wang’s algorithm .....	26
6) Wu’s algorithm.....	27
5 Simulation .....	33
6 Performance Evaluation.....	37
(1) Size of CDS.....	37
(2) Convergence Time .....	38
(3) Communication Overhead.....	38
7 Conclusion .....	39
Reference .....	40

## List of Figures

All the nodes coloured in green represent members in CDS (BNs), all the blue nodes represent BCNs.

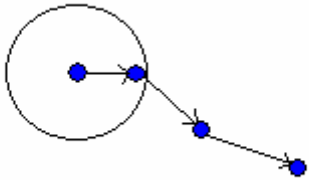


Figure 1

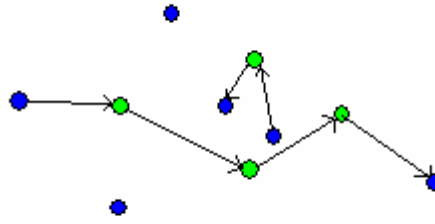


Figure 2

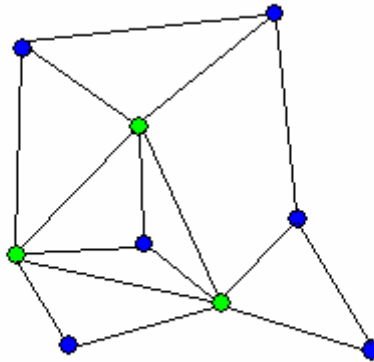


Figure 3

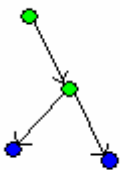


Figure 4a

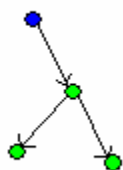


Figure 4b

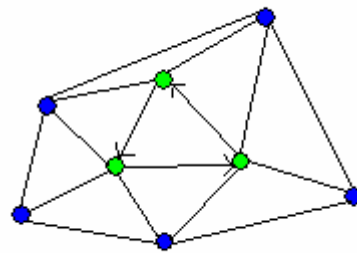


Figure 4c

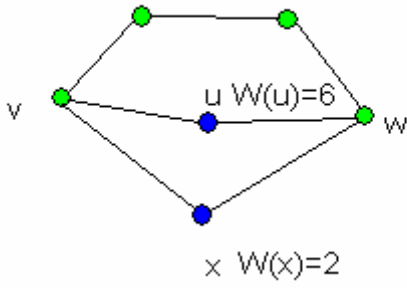


Figure 5a

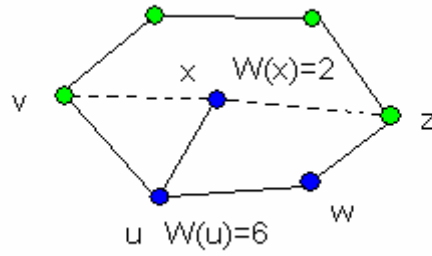


Figure 5b

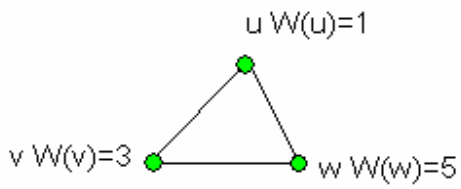


Figure 6a

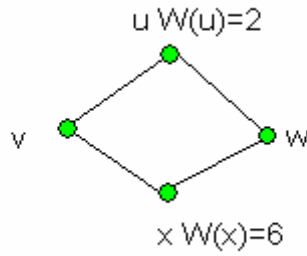


Figure 6b

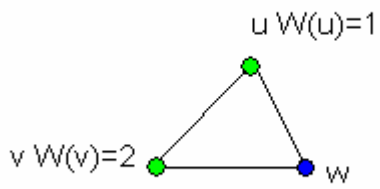


Figure 6c

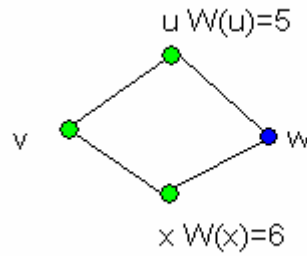


Figure 6d



Figure 7

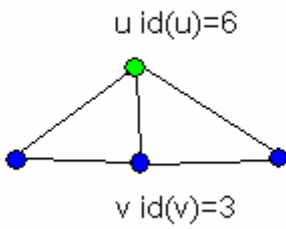


Figure 8a

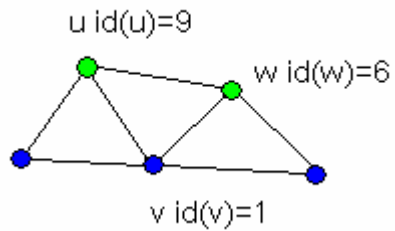


Figure 8b

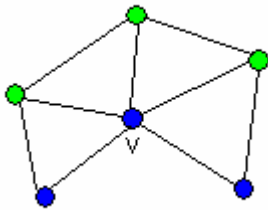


Figure 9

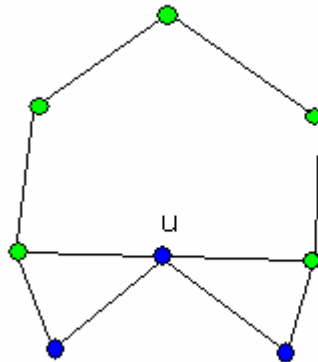


Figure 10

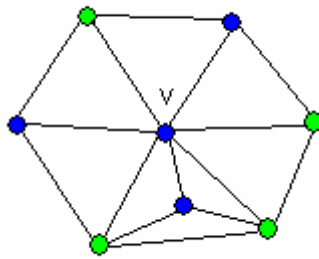


Figure 11

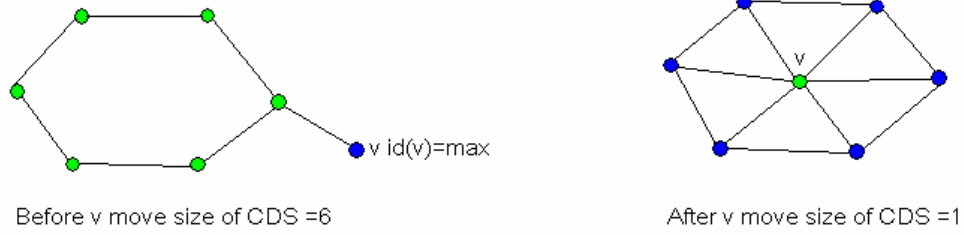


Figure 12 (Algorithm Wu applied)

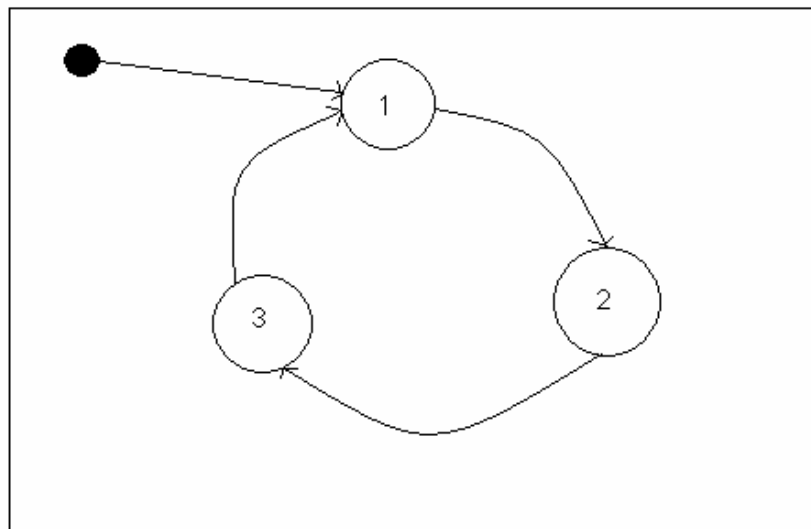
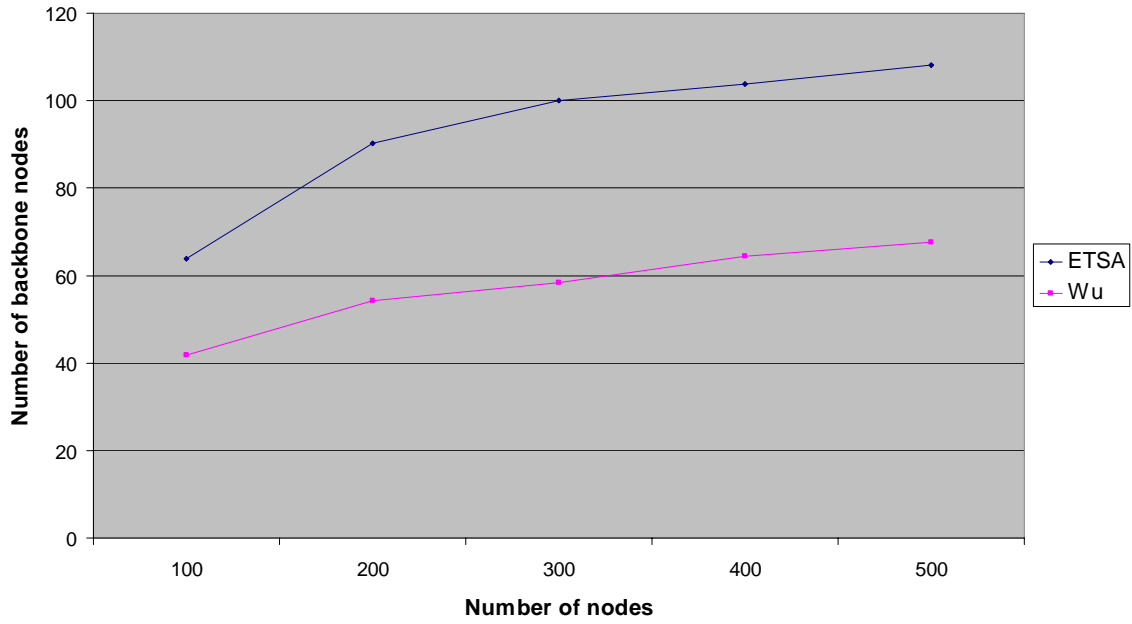


Figure 13

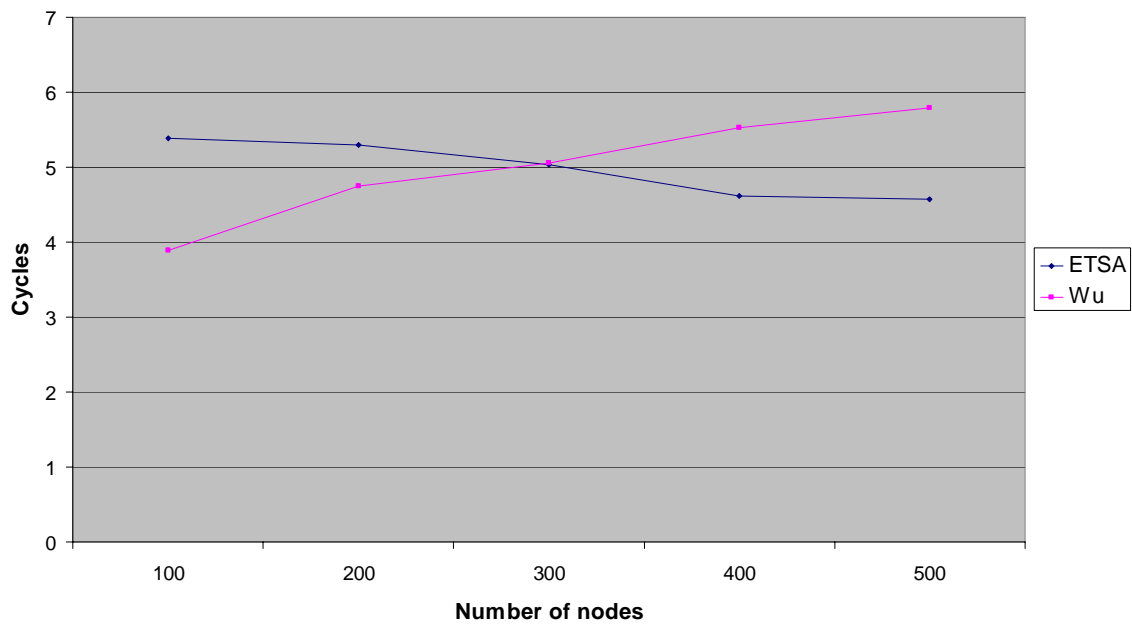
# List of Diagrams

## Size of CDS



## Diagram 1

## Convergence Time



## Diagram 2

**Total Communication Overhead**

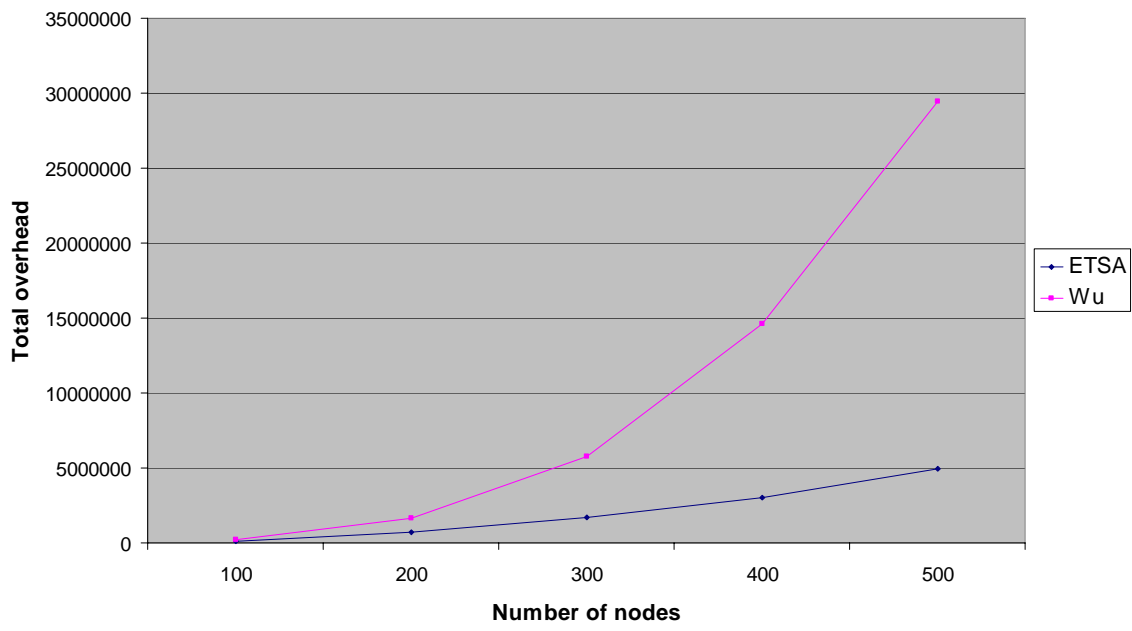


Diagram 3

## List of Tables

	100.00	200.00	300.00	400.00	500.00
ETSA_Size (Mean u1)	63.79	90.21	100.02	103.73	108.14
Wu_Size (Mean u2)	41.86	54.29	58.43	64.46	67.55
ETSA_Size (SD s1)	4.67	5.22	4.95	5.05	4.85
Wu_Size (SD s2)	2.70	2.53	3.33	4.03	3.14
ETSA_Size (90%CI)	(63.02, 64.56)	(89.35, 91.07)	(99.21, 100.83)	(102.90, 104.56)	(107.34, 108.94)
Wu_Size (90%CI)	(41.42, 42.30)	(53.87, 54.71)	(57.88, 58.98)	(63.80, 65.12)	(67.034, 68.07)
u1-u2	21.93	35.92	41.59	39.27	40.59
Lower bound of 90%CI of (u1 -u2)	19.13	32.91	38.50	35.92	37.60
Upper bound of 90%CI of (u1 -u2)	24.73	38.93	44.68	42.62	43.58

Table 1

	100.00	200.00	300.00	400.00	500.00
ETSA_Cycle (Mean u1)	5.38	5.30	5.03	4.61	4.57
Wu_Cycle (Mean u2)	3.89	4.75	5.05	5.53	5.79
ETSA_Cycle (SD s1)	0.91	0.92	0.80	0.71	0.82
Wu_Cycle (SD s2)	0.84	0.82	0.83	1.01	1.06
ETSA_Cycle (90%CI)	(5.23, 5.53)	(5.15, 5.45)	(4.90, 5.16)	(4.49, 4.73)	(4.44, 4.70)
Wu_Cycle (90%CI)	(3.75, 4.03)	(4.61, 4.89)	(4.91, 5.19)	(5.36, 5.70)	(5.62, 5.96)
u1-u2	1.49	0.55	-0.02	-0.92	-1.22
Lower bound of 90%CI of (u1 -u2)	0.85	-0.09	-0.62	-1.56	-1.91
Upper bound of 90%CI of (u1 -u2)	2.13	1.19	0.58	-0.28	-0.53

Table 2

	100	200	300	400	500
ETSA_OH (Mean u1)	137299	701161	1705124	3006411	4935170
Wu_OH (Mean u2)	195680	1642642	5764613	14590347	29450090
ETSA_OH (SD s1)	26977	135651	309733	495147	893900
Wu_OH (SD s2)	50599	270916	1020827	2347997	4953415
ETSA_OH (90%CI)	(132861, 141736)	(678849, 723474)	(1654178, 1756071)	(2924966, 3087855)	(4788137, 5082204)
Wu_OH (90%CI)	(187357, 204002)	(1598080, 1687203)	(5596702, 5932524)	(14204136, 14976559)	(28635326, 30264854)
u1-u2	-58381	-941481	-4059489	-11583936	-24514920
Lower bound of 90%CI of (u1 -u2)	-88118.9411	-1098610.352	-4612736.211	-12828420.53	-27125322.86
Upper bound of 90%CI of (u1 -u2)	-28643.0589	-784351.6484	-3506241.789	-10339451.47	-21904517.14

Table 3

# **Text**

## ***1 Introduction***

There is a special type of wireless network called ad hoc wireless network in which a collection of mobile hosts with wireless network interface can form a network without the aid of any central administration [6]. The main characteristics of such mobile host in ad hoc network are limited power supply, limited bandwidth, limited computing power and unrestricted mobility [10]. The applications of ad hoc wireless networks range from civilian usage, for examples, distributed computing and sensor networks, to disaster recovery like search-and-rescue, and military usage such as battlefield communication and satellite networks [7].

In ad hoc network, efficient routing among mobile hosts is one of the most challenging to solve [8] [6]. Imagine the situation that two hosts want to communicate but are outside of each other's transmission range. The communication will succeed only if they can find a route which consists of a connected set of mobile hosts that are able or willing to forward packets for them [6] (see Figure 1). Routing in such environment does not only require fast convergence but also inducing low communication overhead [6] [7]. Thus routing information has to be localized to adapt quickly to network topology changes, such as host movements [6] [7].

There are two kinds of routing protocol in ad hoc wireless networks: proactive and reactive. In proactive routing, routes to all destinations are computed prior to use. These

routes information is periodically updated by a process in the background to maintain the correctness. In reactive routing, route to a specific destination is computed only when it is needed (“on demand”). The control overhead induced by routing process, tends to degrade the performance of the network. This is called control broadcast storm problem which is described in [9]. To reduce overhead induced in controlling large complex dynamic networks hierarchical routing is used [7]. Thus it is desirable to construct a mobile backbone network (MBN) which consists of only a small fraction of nodes which are called backbone node (BN) to perform the routing process and related maintenance. On the other hand the backbone network (MBN in short) must include sufficient BNs to provide coverage for all the nodes in a ad hoc wireless network. The concept and characteristics of MBN were introduced in [3]. Many CDS construction algorithm has been proposed to form a small CDS in a network. The CDS generated can be used to form a virtual backbone of a wireless ad hoc network. Therefore, Dominating –Set-Based routing is a promising approach since the search space in the derived backbone network is much smaller than the original network [6][7][8].

## ***2 CDS Based Routing***

CDS generated by CDS construction algorithms can be used to implement CDS based routing algorithm. Assume that a CDS has been constructed for a given ad hoc network, dominating-set-based routing usually consists of three steps: 1) if the source is not a gateway host (CDS member), it sends the packets to a source gateway, which is one of the adjacent gateway hosts in its neighbor set. 2) This source gateway host acts as a new source to route the packet in the sub-network generated from the connected dominating

set. 3) Eventually, the packets reach a destination gateway which is either the destination host itself or a gateway host in the destination host's neighbor set. In the latter situation, the destination gateway host forwards the packets to the destination host directly [8] (see Figure 2).

CDS based routing has a significant advantage for ad hoc wireless network when comparing with other routing protocol. It simplifies the routing process to the smaller sub-network generated from the connected dominating set. Thus if a proactive approach is used, only the hosts in the CDS (gateway host) need to keep routing information. If a reactive approach is applied, the dynamic search process for a route is restricted to the dominating set only so the search space is reduced to the sub-network (backbone network) generated from the CDS. In general, by introducing CDS the searching space for routing is limited to the backbone network formed by members of CDS. In ad hoc wireless network, the topology is continuously changed due to node mobility, changes in links between nodes may invalidate the current backbone, thus the price paid to recalculate CDS should not be high [6][7][8].

Obviously, the overhead induced by the process of finding and maintaining a connected dominating set and the size of the sub-network formed by CDS members determines the efficiency of CDS based routing. It is desirable to find a small CDS without compromising the functionality, reliability and efficiency of an ad hoc network. In addition, the CDS construction algorithm should be localized (based on local information)

to satisfy two essential requirements for a routing protocol in ad hoc networks, that is low overhead and fast convergence [6][7][8].

### **3 General knowledge about CDS**

When discussing mobile nodes with the same transmission range in ad hoc wireless network, simple unit disc graphs are usually used to represent the static snapshot of topology at a certain moment. For a given topology, a unit disk graph  $G = (V, E)$  can be constructed in the following way. Vertices in set  $V$  represent the mobile nodes. If two vertices  $u$  and  $v$  are in the transmission range of each other, then a wireless link is established and the edge  $(u, v)$  is added to set  $E$ . Since all the nodes have the same transmission range, all the wireless connections are bidirectional [6].

In a given simple unit disk graph, a dominating set (DS)  $V'$  is a subset of  $V$  such that every vertex in  $V$  is either in the DS or adjacent to at least one member of DS. The resulted dominating set should be connected for ease of the routing process within the induced sub-network formed by elements in  $V'$ . Thus the goal is revised to find a connected dominating set (CDS) which is a special type of DS such that the set of elements in CDS is also a valid DS and the subgraph formed by CDS and related edges is connected [6] (see Figure 3).

In a real ad hoc wireless network, some links can be unidirectional due to either the disparity of transmission range of hosts or the hidden terminal problem [7]. In a network with unidirectional links, the concept of dominating has to be refined [7]. Similarly, a

simple directed graph without self-loop or multiple edges is constructed to represent a static snapshot of such ad hoc wireless network, that is  $D = (V, A)$  where  $V$  is a finite set of vertices and  $A$  is a set of directed edges. A host  $v$  in set  $V$  is called a dominating neighbor (absorbent neighbor respectively) of another host  $u$  in set  $V$ , if there is a directed edge  $(v, u)$  (edge  $(u, v)$  respectively). A subset  $V'$  of  $V$  is a dominating set of  $D$  if every vertex  $v \in V - V'$  is dominated by at least vertex in  $V'$  (see Figure 4a). Similarly, a subset  $V'$  of  $V$  is an absorbant set if for every vertex  $u \in V - V'$  there exists a vertex  $v$  of  $V'$  which is an absorbant of  $u$  (see Figure 4b). Thus the concept of CDS in network with unidirectional link is revised to be a subset of  $V$  that is both dominating and absorbant in a given directed graph  $D$  (see Figure 4c). In an undirected graph, a dominating set is also an absorbant set since all the links are bidirectional [7] [8].

As it is described in section 2 that a small CDS is desirable, so the ultimate goal is to find the minimum connected dominating set (MCDS) in a given graph  $G$  to implement routing protocol. In addition the unrestricted mobility and limited bandwidth of nodes determines that such finding process should have short convergence time and induce low communication overhead to make periodically executing such finding process feasible [6]. Unfortunately, finding a minimum dominating set (MDS) in a graph  $G = (V, E)$  is NP complete in general even if the topology information of the whole network is available [6]. So there is no algorithms have been found which can find the MDS for a graph  $G$  in polynomial time [4]. The connectivity requirement for resulted DS adds additional dimension of difficulty [7] [4]. Thus the problem of finding a MCDS in a graph  $G$  is also NP complete in general [4]. However, many approximation algorithms which run in

polynomial time have been proposed to find small CDS in a simple unit disk graph. Assume a CDS is generated by applying a CDS construction algorithm to a graph  $G$ , MCDS is the actual minimum connected dominating set for the graph. The approximation ratio of the algorithm  $R$  is equal to  $|CDS|/|MCDS|$  [14]. The approximation ratio is one of the most important metrics to evaluation the performance of such CDS construction algorithm. In the next section 6 algorithms are reviewed and compared theoretically.

#### ***4 CDS construction algorithms***

In real ad hoc wireless network. Not all the nodes are eligible to take the responsibility of a BN. As it is stated in [3], under MBN protocol the nodes are classified into two groups based on their respective computation resources, processing power, and transmission capabilities: regular nodes (RN) and backbone capable node (BCN). However in most researches that are aiming to compose an efficient CDS algorithm, the assumption that all the nodes in a given network are BCNs is made to simplify the problem. Most algorithms reviewed in this section have this assumption as a base [1] [2] [5] [6] [11] [12] [13]. The work for extending algorithm that can be applied to more general network is left to future work.

Algorithms for CDS construction in ad hoc networks can be divided into two categories:

- 1) Centralized algorithms that depend on global topology information (network-wide).
- 2) Decentralized algorithms that depend on local topology information only.

algorithms can be divided further into cluster-based algorithms and pure localized algorithms [8].

By applying centralized algorithms, smaller CDSs than those constructed by distributed algorithms is yielded usually. However application of centralized algorithms has limited usage in ad hoc network since they require global topology information thus have very slow convergence time ( $O(n)$  in worst case). Cluster-based algorithms usually consist of two phases. In the first phase, the host in network is partitioned into clusters and a cluster-head is elected for each cluster. Then elected cluster-heads are interconnected (through other hosts if necessary) to form a CDS in the second phase. The cluster-based algorithms has a constant approximation ratio but slow convergence time ( $O(n)$  in the worst case). As another type of distributed algorithms, pure localized algorithms has constant convergence time, produce small CDS in most cases, but the approximation ratio is not bounded by constant [8].

For an algorithm to be both efficient and scalable it should meet three requirement describe as following. In ad hoc network, fast convergence and inducing low communication overhead are two major requirements for routing [6]. (1) So as the base to implement routing protocol on, CDS should also converge in constant time ( $O(1)$ ), so its formation will not be affected by other variables such as number of nodes in topology or maximum nodes degree etc. The complexity of message that nodes need to send in order to exchange the information about their neighbor should also be low to achieve low overhead. Many distributed algorithms require 2-hop neighbor information by applying

such algorithm nodes need to exchange their 1-hop neighbor information [1]. (2) Thus the message complexity of chosen algorithms should be upper bounded by  $O(\Delta)$  where  $\Delta$  is the maximum node degree in the given network topology. Moreover the size of resultant CDS should be close to minimum because smaller CDS also makes the searching space for routing process smaller. (3) Ideally, the size of generated CDS has constant ratio to the size of MCDS. If no constant approximation ratio has been proven, then there should exist a probabilistic bound on the average size of CDS generated at least. Based on the requirements described above, I establish the following three criteria to select candidate algorithms: (1) The resultant CDS should converge in constant time. (2) The approximation ratio is a constant or bound by a constant in average cases. (3) The hello message complexity should not be greater than  $O(\Delta)$ .

Note that many terms for expressing the similar idea, to acknowledge the algorithm authors' work, when I describe an algorithm in detail, I use the original terms used in the references. In general, "backbone capable node (BCN)", "mobile host", "non-coordinator" can be used interchangeably. "backbone node (BN)", "gateway host", "coordinator" and "dominator" are used to denote CDS member. "hello message" and "control message" are used to denote the message that nodes need to broadcast in the formation of CDS.

## 1) ETSA

Wireless local area network systems are widely implemented today to provide hot spot coverage. They are extended by implementing a reliable meshed network which serves to interconnect the access points. The network formed by mesh access point and mesh link is a special type of ad hoc network. Such network usually has high nodal density, unstable communication link quality and less mobility. This algorithm is proposed to efficiently construct CDS in meshed network.

The predecessor of ETSA is MBN topology synthesis algorithm (TSA) which was proposed in [2]. It assumes that the networks discussed are all connected, and all the nodes are backbone capable and have the same transmission range in both [1] and [2].

In TSA every node has two timers: Short\_Timer and Long\_Timer. Long\_Timer is three times long as Short\_Timer. Whenever the Short\_Timer expires, the node sends a hello message to its neighbors through broadcast. The hello message contains the “node ID”, “node status” (BCN or BN), “nodal weight”, the “associated BN ID” and its “BN neighbor list”. Whenever the Long\_Timer expires, the node runs Association algorithm and BCN to BN conversion algorithm (BN to BCN conversion algorithm respectively) if its status is BCN (BN respectively). Note that, there is no time synchronization between nodes. Every node maintains its own time and behaves in asynchronous fashion.

**Association algorithm:** BCNs will try to find a BN with highest weight among its 1-hop BN neighbors. The weight of a node can be assigned based on its ID, nodal degree,

capability, or on some other measures. This weight function provide network designer with the flexibility of choosing certain criteria which is used to determine the importance and preference of nodes. If no BN neighbor is found, it will select a BCN neighbors including itself which has highest weight (lowest ID is preferred in case of a tie) to associate with. Then it inserts the ID of the selected BCN into the hello message as its “associated BN ID”.

**BCN to BN conversion algorithm:** A BCN u will convert to a BN if any of the following three conditions is satisfied: (1) Connectivity condition<1>: If any two of its BN neighbors do not connect to each other within 2 hops that is directly connect or connect through a common BN, and u has the highest weight among all its BCN neighbors which can also provide such connection.. (2) Connectivity condition<2>: If any one of its BN neighbors and any one of its BCN neighbors do not connect to each other in 2 hops and u has the highest weight among all its BCN neighbors which can also provide such connection [2].

**BN to BCN conversion algorithm:** This algorithm is for redundant BNs to change their states back to BCN. A BN is redundant if the backbone network without it still maintains the connectivity and can provide network-wide coverage. A BN will covert to a BCN if all the following conditions are satisfied: (1) Connectivity condition: All its BN neighbors are connected in 2 hops in case of connect through another common BN, the BN should have higher weight. (2) Client coverage condition: If all its BCN neighbors have more than one BN neighbors [2].

TSA exhibits all the desirable features required by constructing CDS efficiently: (1) The CDS generated converge in constant time ( $O(1)$ ). (2) The size of CDS generated is independent of number of nodes and node degree. (3) Constant message complexity ( $O(1)$ ) [2]. However, this algorithm's performance (CDS generated has large size) is seriously degraded when employed in a network environment with high control message loss and asynchronous nodal behaviour [1].

In [1] an enhanced MBN topology synthesis algorithm (ETSA) is proposed based on TSA. In ETSA, a new field "BN-to-BCN indicator" is introduced for BN to maintain and send to its BN neighbor to inform its eligibility of changing back to BCN. Also, the contents of hello messages broadcast by BCN nodes and BN node are no longer the same. The BCN to BN conversion algorithm and the BN to BCN conversion algorithm are also slightly modified. Moreover ETSA introduce two BCN to BN conversion restricting rules (Rule 1 and Rule 2) to regulate excessive BCN to BN conversions induced by imperfect neighbor information which is caused by control message losses and asynchronous behaviour of nodes.

When broadcast a node insert the "node ID", "node status", "nodal weight", and its "BN neighbor list" in the hello message. In addition a BN node put the "BN-to-BCN indicator" into the hello message. A BCN node inserts the "associated BN ID" in the hello message respectively.

**The Modified BCN to BN Conversion Algorithm:** a BCN  $u$  will convert to a BN if any of the following three conditions is satisfied: (1) Client coverage condition:  $u$  has received association request at least once in the previous cycle or  $u$  has the highest weight among its unassociated BCN neighbors. (2) Connectivity condition <1>: At least one pair of its BN neighbor (e.g. BN  $v$  and BN  $w$ ) do not connect to each other directly or connect through another common BN neighbor (e.g. BN  $y$  see Figure 5a) in the backbone network and  $u$  has the highest weight (low ID is preferred in case of tie) among all its BCN neighbors (e.g. BCN  $x$ ) that can provide such connection. (3) Connectivity condition <2>: At least one of its BN neighbors (e.g. BN  $v$ ) and one of its BCN neighbors (e.g. BCN  $w$ ) do not connect to each other directly or through one common BN neighbor and (i)  $u$  has the highest weight among all its BCN neighbors that can provide such connection and (ii) none of its BCN neighbor (e.g. BCN  $x$ ) can directly connect to BN  $v$  as well as to at least one BN neighbor (BN  $z$ ) of BCN  $w$  (see Figure 5b) [1].

**The Modified BN to BCN Conversion Algorithm:** a BN  $u$  will convert to a BCN only if all the following conditions are satisfied: (1)  $u$ 's BN-to-BCN indicator have value 1. (2) Client coverage condition: every one of  $u$ 's BCN neighbor which has associated BN id equal to  $u$ 's id has at least one more BN neighbor other than  $u$ . (3) Connectivity condition <1>: Any two of  $u$ 's BN neighbors (e.g. BN  $v$  and BN  $w$ ) either (i) are directly connected to each other, and: node  $u$  does not have the highest weight among nodes  $u$ ,  $v$  and  $w$ ; or either  $v$  or  $w$  indicate that they cannot convert to a BCN (see Figure 6a), or, (ii) have at least one other common BN neighbor (e.g. BN  $x$ ), and BN  $x$ 's BN-to-BCN indicator has value 0 or has a higher weight than  $u$  (see Figure 6b). If condition (3) is not satisfied

because the common BN neighbor  $x$  has smaller weight,  $u$  set its BN-to-BCN indicator to 1. If condition (3) is not hold because no common neighbor  $x$  is found which means the network connectivity will be broken if  $u$  change back to BCN, so  $u$  sets its BN-to-BCN indicator to 0. (4) Connectivity condition <2>: Any one of  $u$ 's BN neighbors (e.g. BN  $v$ ) and any one of node  $u$ 's BCN neighbors (e.g. BCN  $w$ ) either (i) are directly connected to each other, and: BN  $v$ 's indicator has value 0 or has a higher weight than  $u$  (see Figure 6c), or, (ii) have at least one other common BN neighbor (e.g. BN  $x$ ), and:  $x$ 's indicator has value 0 or has a higher weight than node  $u$  (see Figure 6d). If condition (4) is not satisfied because the common BN neighbor  $x$  found has smaller weight,  $u$  set its indicator to 1. If condition (4) is violated because of no common BN neighbors are found,  $u$  sets its indicator to 0 [1].

**Rule 1:** If the number of a BCN's BN neighbors is higher than 10, the BCN should not convert to a BN [1].

**Rule 2:** If the number of a BCN's BN neighbors increase by at least one within the previous Short\_Timer period, it should not convert to a BN [1].

It had been verified through simulations that the two rules effectively reduce excessive BCN-to-BN conversion while maintaining all the desirable features of TSA that are: (1) Constant convergence time. (2) The size of backbone network generated has constant approximation ratio. (3) The complexity of hello message is of the order of  $O(1)$  [1].

Therefore, algorithm ETSA is chosen to be a candidate algorithm.

## 2) d-SPR algorithm

A d-SPR set is a subset of vertices set  $V$  in a given simple undirected graph  $G$  that is “d-hop connected”, “d-dominating” and has “d-shortest path property”[5]. “d-hop connected” means choosing any two nodes in the d-SPR set  $u, v$ , there always exists a path between  $u$  and  $v$  such that the hop count between consecutive nodes along the path that belong to the set never exceeds  $d$ . “d-dominating” means that every node in the network that is not in the d-SPR set is connected to at least one node in d-SPR set in  $d$  hops [5]. The d-SPR set has the “d-shortest path property” because, for any two nodes  $u$  and  $v$  in the given graph, there exists a shortest path connecting  $u$  and  $v$  such that the nodes on this path that are also in the d-SPR set, together with  $u$  and  $v$ , form a d-hop connected set [5].

Obviously the d-SPR is more superior to CDS as it includes all the intermediate nodes on the shortest paths of length which is depended on the  $d$  value. The value of  $d$  must be greater than 1 and is usually set between 3 and 5 inclusively. However this desirable attribute does not come for free. A distributed algorithm has been proposed in [5] to compute d-SPR set in a given ad hoc wireless network. For nodes to be able to make correct decision that whether they are in the d-SPR set, they need to learn their  $(d+1)$ -local view. A node's  $(u)$   $(d+1)$ -local view consists of all the nodes that are connected to  $u$  within  $d$  hops and all the links between these  $d$ -hop neighbors, except for the links that connect two  $(d+1)$ -hop neighbors that are both at a distance  $(d+1)$  from node  $u$ . So the message complexity is in the range  $(O(\Delta^{d-1}), O(\Delta^d))$ . Thus the formation process induce too much communication overhead (the size of control message is at least of  $O(\Delta^2)$ )

which violate the third criteria I established. In addition, in [5] it has been verified that the d-SPR set has greater size than the size of CDS generated by Wu's algorithm [7].

Therefore this algorithm is not selected.

### 3) Span

Span is a pure localized algorithm for forming CDS in ad hoc wireless network. This algorithm is similar to the marking process proposed in [6]. By applying this algorithm, a set of special hosts called "coordinators" are elected. Then the coordinators form a CDS, so other hosts can switch in energy saving mode while maintaining the routing capability for the whole network. A host  $v$  becomes a coordinator if it has two neighbors that are not connected within 3 hops (directly connect to each other, connected through one common coordinator neighbor, or indirectly connected via two intermediate coordinators.) [11].

When a host finds it is eligible to become a coordinator, it needs to wait for a backoff delay which is determined by its energy level and its 2-hop neighborhood topology information. The backoff delay is used as priority value in this algorithm. That is the nodes that can provide more coverage for its unconnected neighbors and has higher energy level ( $E_r/E_m$  where  $E_r$  is the amount of energy a node still remains,  $E_m$  is the maximum energy the node can have) has shorter backoff delay, thus have higher chance to become a coordinator. A coordinator withdraw algorithm is proposed to reduce the size of the CDS generated. This algorithm is very similar to the pruning rule proposed in [7]. By applying this algorithm, a coordinator change back to non-coordinator if all its neighbors are covered by its coordinator neighbors with higher priority (i.e. id).

It has been proven that the performance of span is slightly better than restricted implementation of Rule K which is proposed in [8] in terms of the size of CDS generated, when node degree serve as priority value instead of backoff delay. However span require 3-hop neighbor topology information which make the hello message has complexity of order of  $O(\Delta^2)$ , thus algorithm span is not selected.

#### **4) Wan's algorithm**

Wan, Alzoubi and Frieder proposed a centralized CDS construction algorithm in [12]. In this scheme, a spanning tree is constructed for a given wireless network firstly. Then a maximal independent set (MIS) is elected such that each vertex in the MIS can be connected to the spanning tree through an extra vertices. The size of formed CDS has a constant approximation ratio [12]. However CDS does not converge in constant time under this scheme since it requires global topology information. So this algorithm is not selected.

#### **5) Wang's algorithm**

A cluster-based algorithm which is aiming to minimize the total cost of CDS formation is proposed by Wang et al. [13]. It consists of two sub-algorithms: algorithm 1 and algorithm 2. By applying algorithm 1, a cluster head (dominator) is elected in each cluster. When it finishes, a dominating set for the given network is formed. Under algorithm 2, a set of connectors is elected to connect the dominators elected in algorithm 1. The

connectors and dominators together form a CDS for a given network. It has been proven that the total cost induced in the formation of backbone is within a small constant factor of the optimum for homogeneous networks when either the nodes' costs are smooth or the network density is bounded [13]. However the resultant CDS does not necessarily converge in constant time, so this algorithm is not selected.

## 6) Wu's algorithm

This algorithm is a pure local algorithm, and it was initially proposed by Jie Wu and Hailan Li in 1999. It was composed of a marking process and 2 pruning rules. At that time it only covered simple unit disk graph (ad hoc network in which nodes have the same transmission range).

Assume a connected simple unit disk graph  $G = (V, E)$  is given. Every vertex in  $V$  has a marker which can have value either T (marked) or F (unmarked). The marker is represented by  $m(v)$  where  $v$  is a vertex in  $V$ . The open neighbor set of vertex  $v$  is represented by  $N(v) = \{u \mid \{v,u\} \in E\}$ .  $N[v] = \{u \mid \{v,u\} \in E\} \cup \{v\}$  represents the close neighbor set of vertex  $v$ . Every vertex in  $G$  has a distinct ID which is represented by  $id(v)$ . Assuming that initially all the vertices are unmarked and they know their open neighbor set already [6].

**Marking process:** Every  $v \in V$  exchanges its open neighbor set  $N(v)$  with all its neighbors by broadcasting hello message. Thus all the vertices learn their 2-hop neighbor information. If a vertex has two unconnected neighbors (i.e.  $u \in N(v)$ ,  $w \in N(v)$  but  $u$

$\notin N(w)$ ) then it changes its marker to T [6] (see Figure 7). Let  $V'$  represent the set of all the vertex with marker T in  $V$ .  $V' = \{v \mid v \in V, m(v) = T\}$ .  $G'$  is the subgraph of  $G$  induced by  $V'$ . The following three theorems hold for  $V'$ .

Theorem 1: Given a  $G = (V, E)$  that is connected, but not completely connected, the vertex subset  $V'$ , derived from the marking process, forms a dominating set of  $G$  [6].

Theorem 2: The reduced graph  $G' = G[V']$  is a connected graph.

Theorem 3: The shortest path between any two vertices does not include any non-gateway nodes (vertex not in  $V'$ ) as an intermediate node [6].

Therefore,  $V'$  derived from marking process is a connected dominating set and it contains all the intermediate nodes in shortest path for every pair of  $u$  and  $v$ . For detailed proves of the three theorems, see [6] on page 8.

At the same Wu and Li proposed two pruning rules to apply on  $V'$  to reduce the size of CDS generated [6].

**Rule 1:** Assume  $u$  and  $v$  are two vertices in  $G'$ , if  $N[v] \subseteq N[u]$ , and  $id(v) < id(u)$ , change the marker of  $v$  to F. So the  $G' = G' - \{v\}$  (1). The role of  $id$  is very important to avoid illegal simultaneous removal of vertices in  $V'$ . Note that other metrics can be used to break a tie (see Figure 8a) [6].

**Rule 2:** Assume vertices  $u$  and  $w$  are two marked neighbors of  $v$  in  $G'$ . If  $N(v) \subseteq N(u) \cup N(w)$  and  $\text{id}(v) = \min \{\text{id}(v), \text{id}(u), \text{id}(w)\}$ , then change the marker of  $v$  to F [6]. The condition  $N(v) \subseteq N(u) \cup N(w)$  implies  $u$  and  $w$  are pair-wise connected (see Figure 8b) [6].

Note that by applying rule 1 and rule 2, the theorems 1 and 2 are still hold for the reduced  $V'$ . However, Wu and Li did not mention that theorem 3 is no longer true in [6]. For proof see Figure 8b.

Wu and Li thought in ad hoc network, such a highly dynamic network environment (all the nodes move frequently), CDS needs to be periodically recalculated from scratch. The CDS recalculation process is described as following. Every node periodically broadcast its  $N(v)$  with  $\text{id}(v)$  and run making process. If its marker have value T, then it applies rule 1 and rule 2 to itself. This algorithm form a CDS in three rounds (two rounds of message exchanging for marking process, one round for rule 1 and rule 2). In this algorithm all the nodes behave synchronously, thus time synchronization is required.

At that time, Dominant pruning rules with more than two covering hosts are not considered based on the following 2 assumption: (1) Testing the coverage of multiple hosts can be costly; (2) only a few hosts' neighbor sets need to be covered by 3 or more other hosts. However the findings of later researches in [8] show either of the two assumptions is true in general.

Jie Wu extend the algorithm proposed in [6] to networks with unidirectional links in [7]. He proposed the concept of dominating set and absorbant set in ad hoc wireless network with unidirectional links. For each node, determining dominating set is simple. When a vertex  $u$  receives the message from its neighbor  $v$ , it knows  $v$  is its dominating neighbor. But it is difficult to find its absorbant in the situation that an absorbant neighbor  $w$  has smaller transmission range so it can not send confirmation message back to  $u$  directly. It requires other host to forward the  $w$ 's confirmation signal back to the sender. This will cause high consumption of bandwidth. One solution is to assign time-to-live (TTL) value to the message in order to limit the number of hops allowed (propagation range). Intelligent decision on TTL is needed to balance the bandwidth consumption and the drop rate of unidirectional links [7]. He also made the refinement of the concept of CDS in such network that is a subset of  $V'$  which is both dominating set and an absorbant set of the corresponding directed graph  $D = (V, A)$ , and  $V'$  is connected [7]. The marking process is extended to find a CDS in network with unidirectional links. It is described as following:

**Extended marking process:** Initially every vertex set their marker to F. Each  $u$  exchanges its dominating neighbor set ( $N_d(u)$ ) and absorbant neighbor set ( $N_a(u)$ ) with all its neighbors. A vertex  $u$  change its marker  $m(u)$  to T if there exist vertices  $v$  and  $w$  such that  $(w, u) \in A$ , and  $(u, v) \in A$  but  $(w, v) \notin A$  [7].

The two rules proposed in [6] are also extended to network with unidirectional link. A CDS member  $u$  will assign its marker to F if its dominating neighbors and absorbant neighbors are covered by no more than two connected vertices in  $V'$  which have higher ID than  $\text{id}(u)$  [7].

Finally in [8] Jie Wu and Fei Dai proposed a generalized version of pruning rule (Rule K) to replace the Rule 1 and Rule 2, which are two special cases of generalized Rule k.

Unmarked nodes by using rule 1 and rule 2 can be also unmarked under Rule K. On the other hand, unmarked nodes (e.g.  $v$ ) by using Rule k are not necessarily unmarked under the combination of Rule 1 and Rule 2 (see Figure 9). CDS pruning rules with more than two covering hosts were not considered in early studies due to the following two assumptions: 1) testing the coverage of multiple hosts could be costly and 2) only a few hosts' neighbor set need to be covered by three or more other hosts. In (3), it has been shown that the two assumption made in [6] based on which generalized dominant pruning rule is not considered, are not true in most situations.

**Rule K:** Assume  $G' = (V' E')$  is the induced subgraph of a given directed graph  $G = (V, E)$  from marked vertex set  $V'$ .  $V_k' = \{v_1, v_2, \dots, v_k\}$  is the vertex set of a strongly connected subgraph in  $G'$ . If  $N_d(u) - V_k' \subseteq N_d(V_k')$  and  $N_a(u) - V_k' \subseteq N_a(V_k')$  in  $G$  and  $\text{id}(u) < \min\{\text{id}(v_1), \text{id}(v_2), \dots, \text{id}(v_k)\}$  where  $N(V_k') = \bigcup_{v_i \in V_k'} N(v_i)$ , then change the marker of  $u$  to F [6].

Rule K can be implemented in either restricted approach or unrestricted approach. In the restricted implementation, all the covering vertices in  $V_k'$  are neighbors of vertex  $u$ .

They are self-connected to each other, not connected via other intermediate gateway host (see Figure 8b). Surprisingly, the complexity of Rule k in restrict implementation is less than rule 2 [8]. In the unrestricted implementation, the covering vertices could be connected via other intermediate gateway hosts which are not neighbors of  $u$  (see Figure 10). Applying unrestricted rule requires global information, which is unrealistic to implement. In addition it has been proven by simulation in [8] that the numbers of hosts unmarked by restricted and unrestricted rule K are very close.

The theorem 5 in [8] shows that the combination of the marking process and restricted Rule K takes 3 rounds to complete. Marking process uses 2-hop neighbor information which can be collected through 2 rounds of information exchanges. Each marked host notifies its neighbors the change of its marker in round 3. So it takes constant time to determine a CDS in a given network topology in which all the nodes operates in a synchronous fashion. The message complexity is bounded by  $O(\Delta)$ , since each node need to broadcast its 1-hop neighbor information. Moreover the theorem 2 in [8] shows that the probability of the approximation ratio  $R$  being infinitely large is very small and the average value of  $R$  is bounded by a constant which is independent of the number of nodes and network density. Therefore algorithm Wu is chosen to be a candidate algorithm.

## **5 Simulation**

In section 4, six CDS construction algorithms are reviewed and assessed . Wu's algorithms and ETSA are chosen to be the candidates since they both meet all the criteria that I have established for being an efficient and scalable backbone network synthesis algorithm.

I made the following assumptions for simulation, since the two candidates are proposed based on assumptions that contradict reality in general: (1) All the nodes have the same maximum transmission range. Although Wu's algorithm has been extended to cover ad hoc network with unidirectional wireless link, it does not guarantee to yield a CDS that covers all the nodes with unidirectional link. The reason is that the TTL for confirmation signal can not be big otherwise the induced overhead will degrade the performance of the network dramatically, and it needs to be predetermined for all the nodes. If a confirmation signal takes more hops than specified by TTL, then the link to the node will be dropped. ETSA was proposed under the assumption that all the nodes have the same transmission range. It has not been extended to cover unidirectional link. Therefore, all the nodes are set to have equal transmission range in my simulation. (2) Perfect MAC layer operation that does not induce control packet losses. In [1], ETSA is proven that it still works without dramatic performance degradation, even if control message losses cause nodes having imperfect neighbor information. But Wu's algorithm is not proven to have consistent performance under such condition [8]. In order to compare them equally, I assume perfect MAC layer operation in network. (3) All the nodes have adequate

computing power and power supply to become BNs. This assumption is made since both candidate algorithms were proposed for network that consist of BCNs only.

In [8], algorithm SC-DFS was proposed for a node to find its strongly connected component set which consists of its eligible BN neighbors. In my simulation there is no unidirectional links, so I implement a simpler algorithm which is based on spanning tree algorithm to compute node's connected components. This algorithm is described as following: (1) Build a subgraph which consists of its BN neighbors (neighbor with marker T) with higher ID. (2) Compute the set of connected component in the subgraph by growing a spanning tree in the subgraph. When spanning tree algorithm finishes, put the tree into the connected component set. If there are still nodes left repeat spanning tree algorithm until no nodes left. (3) Change the node's marker to F, if there exists a connected component in the set that covers all its neighbors (see Figure 11).

Wu's algorithm was proposed and only tested in the situation that nodes operate synchronously. In a simulation, the scenario is described as following: In the first round every nodes broadcasts it own information and builds its 1 hop neighbor set based on messages received. In the second round each nodes broadcast the message of information about its 1-hop neighbor and itself, and identifies the links among its 1-hop neighbor based on information received (2-hop neighbor information is received in this round). Then every node runs the marking process. When all the nodes complete marking process, every node advertises its new marker value by broadcasting in round 3. Then the restricted rule K is applied to remove redundant CDS members. On the other hand, ETSA

is proposed for nodes that operate in asynchronous fashion. In order to fairly compare the two, I implement Wu's algorithm in a similar way as ETSA to cover asynchronous manner of operation. In my design for asynchronous implementation of algorithm Wu, every node has a Short\_Timer and a Long\_Timer. The Long\_Timer is three times as long as the Short\_Timer. Whenever the Short\_Timer expires, the node broadcasts a hello message which contains its ID, marker, and its 1-hop neighbor information. Whenever the Long\_Timer expires, the nodes with marker F run marking process if it finds two unconnected neighbors such that none of its connected components which consists of its BN neighbor can provide such connection (Rule K) for them, then the node changes its marker to T. I call this algorithm which is a combination of marking process and Rule K "BCN-to-BN-Wu". Nodes with marker T apply Rule K whenever the Long\_Timer expires. If it finds that one of its connected components covers all its neighbors, then the node change its marker to F.

The way I determine whether a CDS has converged in a topology is by running the simulation one more cycle. If the new CDS generated in the additional cycle is the same as the one generated in the previous cycle, then I declare the CDS converge in the previous cycle. The focus of my research is on static network in which nodes are not allowed to move. The main reason is that the topology is continuously changing because of nodes' movement. So the CDS generated never become steady in a mobile network (see Figure 12).

In my implementation, nodes have three states instead of having two timers. In the network in which ETSA is employed, a node will broadcast hello message when in state 1, broadcast hello message and record the size of its BN neighbor set when in state 2. When in state 3 the node will broadcast hello message, record the size of its BN neighbor set again and determine whether a new BN neighbor occur in the previous Short\_Timer period, then it runs association algorithm and BCN to BN algorithm, or BN to BCN algorithm depend on its status. When a node finishes the tasks specified on its current state, it changes to the next state (see Figure 13). In networks where Wu's algorithm is employed, a node simple broadcast when in state either 1 or 2, or run algorithm "BCN-to-BN-Wu" if marker is F and in state 3. For nodes with marker T, they run Rule K in state 3.

A Simulation is described as the following. The maximum transmission range is set to 100 units. In  $500 * 500$  square units of a 2-D area, randomly distribute predetermined number of nodes into this area. The formed network must be connected; otherwise it is discarded. The process is repeated until a connected network is generated. The reason is that if a generated network is not connected then the network can be partitioned into connected sub-networks and each with fewer nodes. When a network is instantiated, all the nodes are created in random order with randomly assigned state value. The convergence time is measured in cycles. All the nodes go through its 3 states once in a cycle.

## **6 Performance Evaluation**

To compare algorithm Wu and ETSA fairly, I first created 5 sets of network topologies. Networks in the 5 sets have 100, 200, 300, 400, or 500 nodes respectively. Each set contains 10 topology files. Through testing I found that for a given topology file two simulation which use the same algorithm are very unlikely to generate exactly the same result (in terms of size of CDS yielded and convergence time) but the results are very closed. This difference can be easily verified by using my simulator run on a topology file more than once. The variations in results are caused by the random order for creating nodes and the randomly assigned state values. Therefore I run the simulation 10 times for each topology file under each algorithm. That is, 100 simulations for each set of topology under an algorithm. The reason for running simulations in a scenario so many times is to be able to calculate a more accurate mean value and a narrow 90% confidence interval. When all the simulations finish, the resultant data (convergence time, size of CDS and total overhead) are entered into Minitab as input to calculate the means, the standard deviations and 90% confidence interval for each set under a candidate algorithm. Finally all the statistic data are summarized in Table 1 2 and 3 and corresponding diagrams are created.

### **(1) Size of CDS**

It seems that there always exists a trade off between the neighborhood information and the size of CDS generated, that is the more neighborhood information a node can learn, the better decision it can make, thus a smaller CDS is yielded. The diagram 1 shows that in terms of CDS size, algorithm Wu outperform algorithm ETSA in all the sets of

simulations. The 90% confidence intervals in Table 1 confirm that the mean of CDS sizes of algorithm ETSA are greater than the mean of CDS sizes generated by employing algorithm Wu in all the cases. Therefore, algorithm Wu is more efficient in terms of CDS size.

## **(2) Convergence Time**

The convergence time in terms of total cycles each algorithm take to construct a CDS in different cases is show in diagram 2. In general, the difference is very small (at most 1 cycle) in all cases. Note that when the node degree is relative sparse (100 or 200 nodes in networks), algorithm Wu takes fewer cycles than algorithm ETSA to converge CDS.

When the networks are relatively dense (400 or 500 nodes), ETSA take fewer cycles than algorithm Wu. The 90% confidence intervals in Table 2 confirm that when there are 300 or more nodes in network topology the mean of convergence times of ETSA is shorter than the mean of those of algorithm ETSA. In general, algorithm Wu forms a CDS a litter bit faster in sparse network. In dense network, CDS construct by ETSA converge in slightly shorter period of time.

## **(3) Communication Overhead**

As shown in diagram 3, algorithm Wu induce much more total communication overhead than ETSA in dense networks. This may look like a surprise initially as two algorithms take similar number of cycles to converge CDS in general, but it is not since the hello message of ETSA only contains information of 1-hop BN neighbors which are bound by

22, thus the size of the hello message is of the order of  $O(1)$  [1]. On the other hand, the hello message of algorithm Wu includes full 1-hop neighbors' information, so the size of hello message increases proportionally to the increase in network density [8]. The 90% confidence intervals for the differences between the means of overhead induced by two algorithms in different cases are calculated and presented in Table 3. These CIs confirm the difference shown in diagram 3.

## **7 Conclusion**

In this paper, six CDS construction algorithms are reviewed. Two of them are implemented and tested in static network simulations. Their performance are evaluated and compared in terms of size of CDS, convergence time, and communication overhead respectively. The statistic data calculate based on the data gathered from simulations confirms the greatest advantage of ETSA, the constant hello message size. This makes ETSA much more superior if the bandwidth of wireless links is very limited in an ad hoc wireless network. On the other hand algorithm Wu is more efficient if the size of backbone network instead of low communication overhead is emphasized.

## Reference

- [1] H.-J. Ju, Izhak Rubin, “Mesh Topology Construction for Interconnected Wireless LANs”, Proceedings of IEEE SECON, pages: 284-294, 2005
- [2] H. Ju, I. Rubin, K. Ni, C. Wu, “A Distributed Mobile Backbone Formation Algorithm for Wireless Ad Hoc Networks”, Proceedings of IEEE International Conference on Broadband Networks (BroadNets), pages: 661-670, 2004
- [3] I. Rubin, A. Behzad, H. Ju, R. Zhang, X. Huang, Y.-C. Liu, R. Khalaf, “Ad Hoc Wireless Networks with Mobile Backbones”, Proceedings of IEEE International Symposium on Personal, Indoor and Radio Communications (PIMRC), vol. 1, pages: 566-573, 2004
- [4] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Cliff Stein, (2001), “Chapter 34 NP-Completeness”, Introduction to Algorithms (2<sup>nd</sup> ed., PP. 966-1021), Cambridge: MIT Press and McGraw-Hill.
- [5] S. Dhar, M. Q. Rieck, S. Pai, and E. J. Kim, “Distributed Routing Schemes for Ad Hoc Networks Using d-SPR Sets”, Journal of Microprocessors and Microsystems, Special Issue on Resource Management in Wireless and Ad Hoc Mobile Networks, Oct. 2004.
- [6] J. Wu and H. Li, “On Calculating Connected Dominating Set for

Efficient Routing in Ad Hoc Wireless Networks”, Proc. of the Third International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, 1999

[7] J. Wu, “Extended Dominating-Set-Based Routing in Ad Hoc Wireless Networks with Unidirectional Links”, IEEE Trans. Parallel and Distributed Systems, vol. 9, no. 3, pp. 189-200, Sept. 2002.

[8] F. Dai and J. Wu, “An Extended Localized Algorithm for Connected Dominating Set Formation in Ad Hoc Wireless Networks”, IEEE Transactions on Parallel and Distributed Systems, vol. 15, pages: 908 – 920, Oct. 2004

[9] Y. Tseng, S. Ni, Y. Chen, and J. Sheu, “The Broadcast Storm Problem in a Mobile Ad hoc Network”, Wireless Networks, Vol. 8, Mar. –May 2002

[10] Larry L. Peterson and Bruce S. Davie. (2003). “Wireless”. Computer Networks a Systems Approach (3rd ed., PP. 131-136). San Francisco: Morgan Kaufmann Publishers.

[11] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, “Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks”, ACM Wireless Networks J., vol. 8, no. 5, pp. 481-494, Sept. 2002.

- [12] P.-J. Wan, K. Alzoubi, and O. Freder, “Distributed Construction of Connected Dominating Set in Wireless Ad Hoc Networks”, Proc. IEEE INFOCOM, vol. 3, pp. 1597-1604, June 2002.
- [13] Y. Wang, W.Z. Wang, and X.-Y. Li “Distributed Low-Cost Backbone Formation for wireless Ad Hoc Networks”, Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing MobiHoc, pp. 2-13 May 2005.
- [14] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Cliff Stein, (2001), “Chapter 35 APPROXIMATION ALGORITHMS”, Introduction to Algorithms (2<sup>nd</sup> ed., PP. 1022-1054), Cambridge: MIT Press and McGraw-Hill.