Proceedings of the 5th WSEAS International Conference on Applied Computer Science, Hangzhou, China, April 16-18, 2006 (pp1089-1094)

Scheduling Forwarding Nodes for Two-Hop Neighborhoods Broadcast in Wireless Ad Hoc Networks

CHENGUANG XU, MINHONG LIN, YINLONG XU
Department of Computer Science and Technology
University of Science and Technology of China
National High Performance Computing Center at Hefei
Hefei, Anhui, 230027, P. R. CHINA

Abstract: - This paper aims at scheduling the forwarding nodes in one-hop neighborhoods and making the two hop neighbors receive the message as soon as possible, which is modeled as the Minimum Time-Slot Forwarding (MTSF) problem. We prove that MTSF is NP-hard in general graph and propose two approximation algorithms. Theoretical analysis and simulation results show that the average time slot needed for broadcasting is a linear of ln|*P*|, where *P* is the set of two hop neighbors of source node *s*. We also compare the performance with that of *flooding*, and the simulation results show that both of them perform better than *flooding*.

Keywords: - **Minimum Delay Broadcast, Ad hoc, Minimum Time-Slot Forwarding**.

1 Introduction

Wireless ad hoc networks have received great attention in recent years, due to its applications in many areas such as battlefield communication, search and rescue, disaster relief and so on. Unlike wired networks or cellular networks, no wired backbone infrastructure is installed in wireless ad hoc networks. In wireless ad hoc networks, nodes are commonly powered by cells, and of limited transmission abilities, and a communication session is achieved through either direct communication when two nodes are close enough or forwarding of intermediate nodes otherwise.

Broadcasting is a popular application such as sending an alarm signal, network conferences, etc., and also a common used strategy to find a route in ad hoc networks. Broadcast is more difficult in wireless ad hoc networks than in traditional wired networks, due to the lack of infrastructures and the broadcast nature [2] of nodes in wireless networks. A simple broadcasting mechanism, known as *flooding*, is to let all the 1-hop neighbors forward, but it has a serious drawback, known as the broadcast storm [3] caused by redundant retransmission. One promising technique was exploited in [4, 5] to reduce redundant retransmissions: Each node maintains a local topology of its 2-hop neighborhood, and only a small subset of 1-hop neighbors forward the message, which can reduce the possibility of collision, decrease the delay of broadcast, and improve the throughput of the network. The subset of 1-hop neighbors selected should cover all the 2-hop nodes, and is referred to as forwarding set [5]. The related work about forwarding set in broadcast is the Minimum Forwarding Set problem [6], which is a special case of the Disk Cover problem [9].

In this paper, however, based on the same assumption that each node maintains a local topology of its 2-hop neighborhoods, we consider the broadcast problem in two-hop neighborhoods. But unlike the previous works [2, 10, 11] on broadcasting in ad hoc networks which mainly aim at providing energy efficient broadcast, we try to minimize the delay of broadcast. The work in this paper also reduces the possibility of collision and redundant retransmission, and improves the throughput of the network. This paper models the minimum delay broadcast in two-hop neighborhoods as the Minimum Time-Slot Forwarding (MTSF), and prove that MTSF is NP-hard in general graph, which means that there is no polynomial time algorithm for MTSF, unless NP⊆P. Two heuristics are presented to solve the MTSF problem. Theoretical analysis and simulation results shows that the average time slot needed for broadcasting is linear of ln|*P*|, where *P* is the number of two hop neighbor nodes of the source node *s*. At last we compare the performance of two algorithms with that of *flooding*, and the simulation results show that both of the algorithms perform better than *flooding*.

The rest of the paper is organized as follows. In Section 2, the network model and MTSF will be introduced. In Section 3, we will prove that MTSF is NP-hard. In Section 4, two algorithms for MTSF as well as the theoretical and simulation analysis are presented. In Section 5, we compare the performance of two algorithms with that of *flooding* by simulation experiments. Section 6 serves as conclusion.

2 Network Model

We assume that each node is of the same transmission radius, and two nodes *u* and *v* can communicate directly with each other if the distance between *u* and *v* is not more than the transmission radius. Then the networks can be modeled as a Unit-Disk Graph [8], in which there is an edge between two nodes if and only if their distance is not more than 1. We also assume that the transmission radius of all nodes equals to their interference radius, so that one node cannot correctly receive the message if two or more of its neighbors transmit at the same time. The main idea in this paper is to divide the broadcast process into several time slots, and in each time slot we choose a subset of 1-hop neighbors to forward. This scheme will avoid unnecessary collisions, and as a result the delay of broadcast can be reduced. At last, all the two-hop nodes should receive the message, and our objective is to minimizing the number of time slots. At first, we need to introduce the definition of Collision-Free Cover.

Definition 1. Collision-Free Cover: *Given* $G(V, E)$, *node* $v\in V$, *node set* $S\subseteq V$. *v is called to be collision-freely covered by S, if and only if there exists exactly one node* $u\in S$ *such that v is one of u's neighbors.*

Collision-Free Cover means `the effectively covered two-hop neighbor nodes in each time slot. In order to cover as many nodes as possible, we may choose two nodes *u* and *w* with the same neighbor *c* in the same time slot. However, we assume that there is a perfect MAC control scheme [12], which can handle this problem. However, we assume that there is a perfect MAC control scheme [12], which can handle this problem. And the 1-hop nodes may use the broadcast in IEEE 802.11, which has no collision detection or avoidance mechanism. In Fig.1 for example, nodes *u* and *w* may ignore the collision in node *c* and transmit simultaneously in the same time slot, for *c* can receive the message correctly in other time slot from another node. And the collision-free cover will guarantee that all the two-hop nodes will receive the message correctly at last. Now we will introduce the MTSF problem as follows.

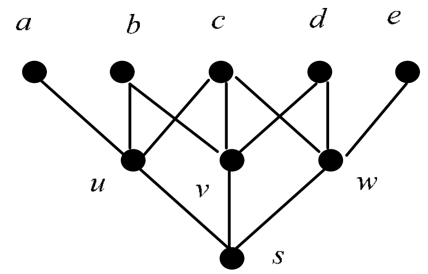

Fig.1 A typical 2-hop neighborhoods of *s*

Definition 2. Minimum Time-Slot Forwarding (MTSF): *Given a source node s, let D and P be the set of* 1- *and* 2-*hop neighbors of s respectively. It is to find a set* $F = \{D_1, D_2, ...D_n\}$ *to minimize the number of the time slot, n, subject to*

(1) $D_i\subseteq D$, $1 \le i \le n$,

(2) *For each node* $u\in P$, *there exists j*, $1 \le j \le n$, *such that u is collision-freely covered by* D_j.

If we find the set $F = \{D_1, D_2, \ldots D_n\}$, the two-hop broadcast can be performed in *n* time slots, in which for time slot *i*, we choose the forwarding node set D_i to send message. See in Fig.1. In time slot 1, we may let nodes in $D_1=\{u, w\}$ to forward, and nodes *a*, *b*, *d*, *e* will receive the message correctly, and in time slot 2, we let node in $D_2 =\{v\}$, and node *c* will be collision-freely covered. At last, all the two-hop nodes correctly receive the message in two time slots. Thus, the minimum delay broadcast in two-hop neighborhoods can be modeled as the MTSF problem in Unit-Disk Graph.

3 Complexity Analysis for MTSF

In this section, we will prove that MTSF is NP-hard in general graph, so there is no polynomial time algorithm for MTSF unless NP⊆P. Before the proof, we need to introduce the 3-Dimensional Matching problem (3-DM) [7], which has been proved to be NP-complete. Then we can prove that MTSF is NP-hard by reducing 3-DM to MTSF.

Definition 3. 3-DM[7]**:** *Given three disjoint sets* $X=\{x_1, ..., x_q\}$, $Y=\{y_1, ..., y_q\}$, $Z=\{z_1, ..., z_q\}$, *and a subset* $M\subseteq X\times Y\times Z$, *it is to find out if there exists a*

Proceedings of the 5th WSEAS International Conference on Applied Computer Science, Hangzhou, China, April 16-18, 2006 (pp1089-1094)

subset $M_0 \subseteq M$ such that each coordinate of any two triples are different and $|M_0|=q$, which also means that each element of X, Y, Z appears in M_0 exactly once.

Lemma 1. *MTSF is NP-hard in general graph, and there is no polynomial time algorithm for MTSF unless $NP \subseteq P$.*

Proof: We prove Lemma 1 by reducing 3-DM to MTSF. Assume that the input of 3-DM is (q, X, Y, Z, M). We transform it into the input of MTSF: $G=(V_1 \cup V_2 \cup \{s\}, E)$, where $s \notin X \cup Y \cup Z \cup M$ is the source node, $V_1=M$, $V_2=X \cup Y \cup Z$ are the sets of 1-hop and 2-hop neighbors of s respectively, and

$E=\{(s,v) | v \in V_1\}$
$\cup \{(m_i, x_j) | x_j$ equals to the X-coordinate of $m_i\}$
$\cup \{(m_i, y_k) | y_k$ equals to the Y-coordinate of $m_i\}$
$\cup \{(m_i, z_l) | z_l$ equals to the Z-coordinate of $m_i\}$.

It is obvious that we can finish this transformation in polynomial-time.

Assume that the optimum result of the MTSF problem is F. (1) If $|F|=1$, without loss of generality, we let $F=|V_0|$. According to the definition of MTSF, all the 2-hop nodes will be collision-freely covered by V_0. And the corresponding 3-DM will come out true, for we may let $M_0=V_0$. (2) If $|F|>1$, then the corresponding 3-DM instant will come out false. So 3-DM $\propto$ MTSF. Since 3-DM is NP-complete, MTSF is NP-hard in general graph. □

4 Approximation Algorithms

The complexity of MTSF in Unit-Disk Graph may be seen from the NP-hardness proof of MTSF in general graph, though its theoretical analysis is still unavailable presently. In this section we present two approximation algorithms for the MTSF problem in Unit-Disk Graph. Both of them can easily be implemented by means of that each node maintains a local topology of its 2-hop neighborhoods.

4.1 Algorithm1

The main idea of Algorithm1 is that we try to collision-freely cover as many nodes as possible in each time slot. And the scheme is that once there are some nodes such that the set of collision-freely covered nodes will be enlarged by adding one of them to the forwarding node set, we arbitrarily choose one of them and add it to the forwarding node set.

4.1.1 The Description of Algorithm1

Algorithm1

Input: source node s, the sets D, P of 1- and 2-hop neighbors of s respectively.
Output: a subset $F = \{D_1, D_2, \ldots D_n\}$
Begin
Initially, $N_1=D$, $N_2=P$
F=nil, $S=N_1$, j=0
While($|N_2|$!=0)
{
  T=nil, j++, i=0;
  Repeat
  {
   (a) i++;
   (b) Randomly choose one of the nodes, say v, in S, such that $|N(T \cup \{v\})|>|N(T)|$, where $N(T)$ = {nodes of N_2 collision-freely covered by T };
   (c) $S=S-v$, $T=T \cup \{v\}$;
  }
  Until there is no node in S, such that $|N(T \cup \{v\})|>|N(T)|$.
  $F=F \cup \{T\}$, $N_2=N_2-N(T)$;
}
Return(F);
End

4.1.2 The Analysis of Algorithm1

We assume that all nodes are uniformly distributed in the analysis of Algorithm 1. When Algorithm 1 runs to Step (c) in each iteration, node v is being moved from S to T, $i=|T|$. Let

$OK_i = |N(T)|$,
$OK_{new} = |N(T \cup \{v\})| - |N(T)|$,
$ERR_i = |$ {nodes of N_2 covered by T with collision}$|$,
$NIL_i = |$ {nodes of N_2 not in the coverage of T} $|$,
$ERR_{new} = |$ {new nodes covered with collision by v}$|$,
$OK_{new} = |N(T \cup \{p\})| - |N(T)|$,

then

$$\begin{aligned} E(OK_{new}) &= E(E(OK_{new} \mid d)) \\ &= E(d \times E(NIL_i) / |N_2|) \\ &= E(d) \times E(NIL_i) / |N_2| \end{aligned} \quad (1)$$

$$E(ERR_{new}) = E(d) \times E(OK_i) / |N_2| \quad (2)$$

We also have

$$E(OK_{i+1}) = E(OK_i) + E(OK_{new}) - E(ERR_{new}) \quad (3)$$

$$E(NIL_{i+1}) = E(NIL_i) - E(OK_{new}) \quad (4)$$

$$E(ERR_{i+1}) = E(ERR_i) + E(ERR_{new}) \quad (5)$$

From (1) to (5), we have

$$E(OK_i) = E(d) \times i \times (1 - E(d)/|N_2|)^{i-1} \tag{6}$$

Assign $y = \ln(E(OK_i))$. So to find maximum $E(OK_i)$, we only need to find the maximum y.

From $\frac{dy}{di} = 0$,

$$i = \frac{-1}{\ln(1 - E(d)/|N_2|)}.$$

So

$$Max\{E(OK_i)\} = \frac{-E(d)}{e \times (1 - E(d)/|N_2|) \times \ln(1 - E(d)/|N_2|)},$$

and

$$\frac{Max\{E(OK_i)\}}{|N_2|} = -\frac{E(d)}{|N_2|} \times \frac{1}{e \times (1 - E(d)/|N_2|) \times \ln(1 - E(d)/|N_2|)} \tag{7}$$

Equation(7) indicates that the average number of collision-freely covered nodes in one time slot is a function of $E(d)/|N_2|$, and the following lemma will help us to estimate $E(d)/|N_2|$.

Lemma 2. *$E(d)/|N_2|$ is a constant in Unit Disk Graph if nodes are uniformly distributed.*

Proof: See in Fig.2 as shown below. S_1 is the area covered by s, which is also the locating area of 1-hop neighbors of s, and S_2 is the maximum possible locating area of 2-hop neighbors of s. Then $S_2 = 3S_1$. For any 1-hop node v of s, denote its coverage area as S_v. Let $S_{in} = S_v \cap S_1$, $S_{out} = S_v \cap S_2$.

Assume that the degree of node v is d, and the distance between v and the source node s is x. Under the assumption of uniform distribution, we have

$$\begin{aligned} E(d \mid x) &= |N_2| \times S_{out}(x)/S_2 \\ &= |N_2| \times (S_1 - S_{in}(x))/S_2 \\ &= (|N_2|/3) \times (1 - S_{in}(x)/S_1) \end{aligned}$$

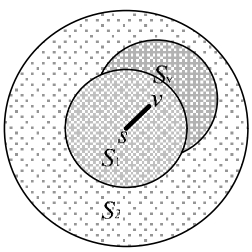

Fig.2 Coverage of one hop node

It can be worked out easily that

$$S_{in}(x) = 2 \times (\arccos(x/2) - (x \times \sqrt{4 - x^2}/4)), \quad S_1 = p.$$

$$\begin{aligned} E(d) &= E(E(d \mid x)) \\ &= \int_0^1 2x \times (|N_2|/3) \times (1 - S_{in}(x)/S_1) dx \\ &= |N_2|/3 - (|N_2|/3p) \times \int_0^1 2x \times 2 \times (\arccos\frac{x}{2} - \frac{x \times \sqrt{4 - x^2}}{4}) dx \\ &= 0.1378 \times N_2 \end{aligned}$$

So $E(d)/|N_2| = 0.1378$ is a constant in Unit Disk Graph. □

With $E(d)/|N_2| = 0.1378$, Equation (7) becomes

$$\frac{Max\{E(OK_i)\}}{|N_2|} = 0.3966,$$

which indicates that averagely about 40% of the nodes of N_2 are collision-freely covered at each time slot. Assume that δ nodes are covered without collision in the last time slot and t is the number of time slot needed by Algorithm 1, then

$$|P| \times (1 - 0.3966)^{t-1} = d,$$

where P is the set of two-hop neighbors of s. So

$$\begin{aligned} t &= (\ln d - \ln|P|)/\ln(1 - 0.3966) + 1 \\ &= 1.9795 \ln|P| - 1.9795 \ln d + 1 \\ &= 1.9795|P| + b \end{aligned}$$

It indicates that the average time slot needed for broadcast of Algorithm1 is a linear of $\ln|P|$, and the simulation result confirms it too. See in Fig.3. In Fig.3, the number of two-hop neighbors varies from 48 to 1536, set $b = -3.0939$, and randomly run 500 times. As in Fig.3, the theoretical result and the experimental result match very well. The number of two-hop neighbors is not very large practically, and the reason that we choose large number of two-hop nodes is for the comparison between theoretical result and the experimental result. As we will see in Section 5, the number of two-hop neighbors varies from 5 to 50, in the comparison between the two algorithms proposed and flooding, which is more practical.

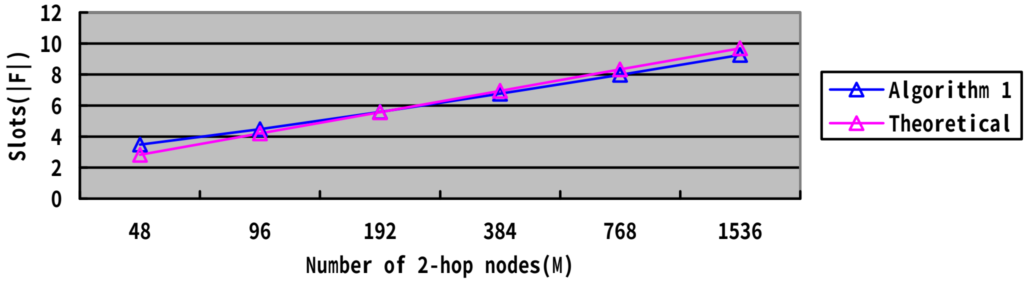

Fig.3. Simulation results for algorihm1

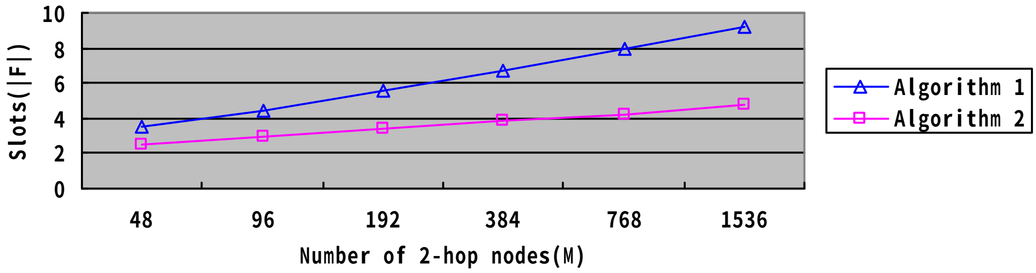

Fig.4. Comparison between agorithm1 and algorithm2

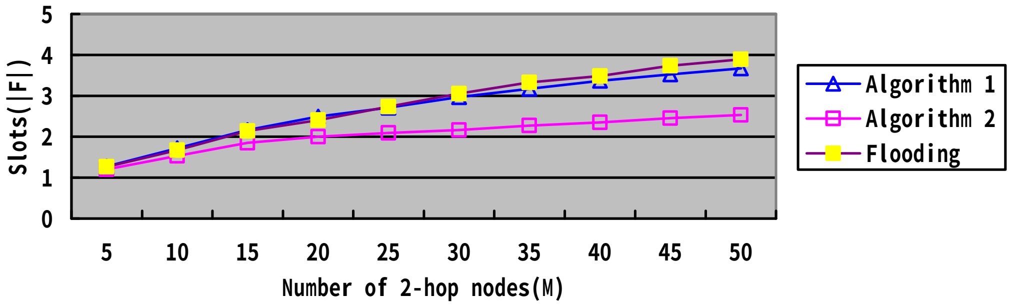

Fig.5. Performance comparison with *flooding*

4.2 Improvement of Algorithm

In Algorithm 1, for each time slot, we choose the new forwarding nodes randomly. However, we can design some positive schemes to choose the forwarding nodes, which may decrease the time slots needed in Algorithm1.

Algorithm2

Input: source node s, the sets D, P of 1- and 2-hop neighbors of s respectively.

Output: a subset $F = \{D_1, D_2,\ldots D_n\}$

Begin

Initially, $N_1=D$, $N_2=P$

F=nil, $S= N_1$, j=0

While($|N_2|$!=0)

{

 T=nil, j++, i=0;

 Repeat

 {

 (a) i++;

 (b) choose the node v in S, satisfying $|N(T\cup\{v\})|>|N(T)|$ and for all $u\in S$ $|N(T\cup\{v\})|\geqslant|N(T\cup\{u\})|$, where $N(T)$ = {nodes of N_2 collision-freely covered by T };

 (c) $S=S-v$, $T=T\cup\{v\}$;

 }

 Until there is no node in S, such that $|N(T\cup\{v\})|>|N(T)|$.

 $F=F\cup\{T\}$, $N_2= N_2-N(T)$;

}

Return(F);

End

As described in Algorithm2, the forwarding node, which can collision-freely cover the largest number of nodes of N_2, is added to T at Step (c). And the simulation results reveal that the performance of Algorihm2 is much better than Algorihm1. See in Fig.4. We guess that the average time slots needed in Algorihm2 is also a linear of ln|P|, but its theoretical analysis is unavailable presently.

5 Comparisons with Flooding

In this section, we will present our performance evaluation to compare two proposed schemes, Algorithm1 and Algorithm2, with *flooding* with NS-2. We assume that nodes are uniformly distributed in the networks. The number of two-hop nodes varies from 5 to 50, and the number of one-hop nodes is about 1/3 of the number of two-hop nodes. The simulator is round based. At the beginning of every round, 1-hop nodes send RTS packets for transmission, and only those who receive the CTS packets can transmit. Other 1-hop nodes must wait until the next round. We calculate only the total rounds for all the 2-hop nodes to receive the data, and don't consider the actual transmission time. And the simulation result is shown in Fig.5.

As in Fig.5, both Algorithm1 and Algorithm2 perform better than *flooding*, especially Algorithm2. And with the increasing of the number of two-hop neighbors, the gap between the performance of Algorithm2 and that of *flooding* become larger. It is due to that more two-hop nodes induce to more possibility of collision in *flooding*, which has a negative impact on the performance of *flooding*.

6 Conclusion

This paper focuses on the minimum delay broadcast problem in two-hop neighborhoods, and aims at scheduling the forwarding nodes in one-hop neighborhoods in order to minimize the transmission delay of broadcast. We model it as the Minimum Time-Slot Forwarding problem, and prove that MTSF is NP-hard in general graph, but whether MTSF is NP-hard in Unit-Disk Graph is still open. Two heuristics are proposed to solve the MTSF problem. Theoretical analysis and simulation results show that the average time slot needed for broadcasting is linear of ln|P|, where P is the set of two hop neighbors of source node s. We also compare their performances with *flooding*, and the simulation results show that both Algorithm1 and Algorithm2 perform better than *flooding*, especially Algorithm2. And the future work is to design algorithms to scheduling forwarding nodes for multi-hop broadcast or multicast, which is more general.

References

[1] J. Broch, D. B. Johnson, and D. A. Maltz, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," *IETF Internet Draft*, draft-ietf-manet-dsr-05.txt, March 2001.

[2] Jeffrey E. Wieselthier, Gam D. Nguyen, and Anthony Ephremides, "On the construction of energy-efficient broadcast and multicast trees in wireless networks," in *Proceedings of IEEE INFOCOM*. 2000, vol. 2, pp. 585–594.

[3] S.-Y. Ni, Y.-C. Tseng, Yuh-Shyan Chen, and J.-P. Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network," Proceedings of the fifth annual *ACM/IEEE international conference on Mobile computing and networking* （*MOBICOM*）, 1999, Pages 151 - 162.

[4] P. Jacquet, P. Muhlethaler, A. Qayyum, A. Laouiti, L. Viennot, and T. Clausen, "Optimized Link State Routing Protocol," *IETF Internet Draft*, draft-ietf-manet-olsr-04.txt, March 2001.

[5] P. Sinha, R. Sivakumar, and B. Vaduvur, "Enhancing Ad Hoc Routing with Dynamic Virtual Infrastructures," *IEEE INFOCOM* 2001. pages 1763-1772.

[6] G. Calinescu, I. Mandoiu, P. J. Wan, and A.Zelikovsky, "Selecting forwarding neighbors in wireless ad hoc networks," Proc. *ACM DIALM*'2001.

[7] M. Garey and D. Johnson, "Computers and Intractability:A Guide to the Theory of NP-Completeness," New York: Freeman, 1979.

[8] B. N. Clark, C. J. Colbourn, and D. S. Johnson,"Unit Disk Graphs," *Discrete Mathematics*, 86:165-177, 1990.

[9] H. Bronnimann and M.T. Goodrich, "Almost Optimal Set Covers in Finite VC-Dimension," *Proc.10th ACM Symp on Computational Geometry* (*SCG*), 1994, pages 293-302.

[10] L. Deying, J. Xiaohua, L. Hai, "Energy Efficient Broadcast Routing in Static Ad Hoc Wireless Networks," *IEEE Trans on Mobile Computing*, Vol.3, No.2. April-June 2004.

[11] P.-J Wan, G. Calinescu, X.-Y. Li, and O. Frieder, "Minimum-energy broadcast routing in static ad hoc wireless networks," in *Proceedings of IEEE INFOCOM*. 2001, vol. 2, pp. 1162–1171.

[12] C.H.R. Lin and M. Gerla, "Asynchronous Multimedia Multihop Wireless Networks," In *Proc. Joint Conf. of the IEEE Computer and Communications Societies* (*INFOCOM*), April 1997.