Formulas and Protocols for Broadcasting in Mobile Ad Hoc Networks

Chang Wu Yu and Cheng Yao Tseng
Department of Computer Science and Information Engineering
Chung Hua University, Hsin Chu, Taiwan, ROC
cwyu@chu.edu.tw

Abstract. An operation is called *broadcasting* if a node sends a packet to all other nodes in an ad hoc network. Broadcasting is an elementary operation to support many applications in ad hoc networks. Thus, many schemes have been proposed for reducing the number of re-broadcasting packets. However, to the best of our knowledge, few papers discuss the bound of necessary broadcasting packets. In the work, we derive formulas for estimating the number of required broadcasting packets by taking three different approaches. In addition, we also propose two protocols: the cluster-head-early method and the connected-dominating-set method to reduce the redundant rebroadcast packets without exploiting the mechanisms of hello messages or cluster formation.

1. Introduction

An ad hoc network consists of a collection of mobile nodes without centralized administration and infrastructure. Each node moves arbitrarily and therefore the network topology may change frequently. Nodes of these network act as routers that discover and maintain routes to other nodes. Applications of this type of network include emergency search-and-rescue operations, meetings in which persons want to share information, and data acquisition operations inhospitable terrain [11].

An operation is called *broadcasting* if a node sends a packet to all other nodes in a network. Broadcasting is an elementary operation to support many applications in ad hoc or sensor networks. For example, AODV uses broadcasting to find a main route [9]. Flooding is the most straightforward method for broadcasting. However, flooding results in two problems:

(1) Redundant rebroadcasts: When a node receives the broadcast packet at the first time, it will rebroadcast the packet to all its neighboring nodes immediately. Such blind flooding results in lots of redundant rebroadcasts. In Figure 1, for example, node *S* initiates a broadcast, and then the neighboring nodes {1, 2, 4, 6, 7} of *S* receives the packet and rebroadcast it in turn. The rebroadcasts of node 3, node 5, and node 8 are redundant, since their neighboring node sets {12, 13}, {15, 16}, and {19, 20} have received the same packet in the broadcasting.

(2) Heavy contention: When many neighboring nodes intend to broadcast messages at the same time, they will contend for network bandwidths with each other. Therefore, the time for the broadcasting will thus be seriously lengthened and the performance of the system will be degraded significantly.

A traditional graph technique can be used to model the desired problem. In a graph *G*, a set $S \subseteq V(G)$ is a *dominating set* if every vertex set not in *S* has a neighbor in S. A *connected dominating set* if S is a dominating set in G and the induced subgraph G_S is also connected. Theoretically, the problem to find the minimum number of rebroadcast is equivalent to finding a minimum connected dominating set containing a given vertex. However the above problem is NP-hard and seem exist no efficient algorithm so far [2].

Therefore, many heuristics have been proposed for reducing the number of re-broadcasting packets [6, 7, 10, 12, 14]. Some of them wait for a period of time to collect local connectivity information (and thus defer the completion time of the broadcasting) [6, 10, 12]. Other schemes pay high costs for maintaining cluster formation.

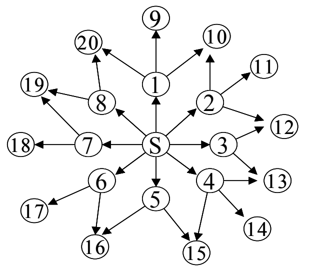

Figure 1. Flooding results in lots of redundant rebroadcasts

To the best of our knowledge, few papers discuss the bound of necessary broadcasting packets. The bound can be used to evaluate the performance of new broadcasting protocols. In the work, we attempt to derive formulas for estimating the number of required broadcasting packets by taking three different approaches. In addition, we also propose two protocols: the cluster-head-early method and the connected-dominating-set method to reduce the redundant rebroadcast packets without exploiting the mechanisms of hello messages or cluster formation.

The rest of the paper is organized as follows. We survey related work in the next section. Section 3 shows three formulas and their derivations. In Section 4, two proposed schemes are described. Finally, Section 5 concludes the paper.

2. Related Work

In recent years, many approaches have been proposed to reduce the number of rebroadcast nodes [6, 7, 10, 12]. These can be classified into two groups: the *cluster schemes* and the *non-cluster schemes*. In the cluster scheme, the whole network consists of many clusters, and the cluster head manages the members of each cluster. Other nodes called *gateways* connect one cluster to another. In this scheme, only cluster heads and gateways can forward broadcasting packets. On the other hand, in the non-cluster scheme, every node in the network decides whether to rebroadcast a packet or not in a distributed mode.

Obviously, flooding is the most straightforward approach for this kind of broadcasting problem. Every node in flooding forwards the received packet straightly and immediately. As expected, flooding results in serious redundancy, contention, and collision.

In [6], Ni *et al.* proposed the counter-based scheme to alleviate the broadcast storm problem. When first receiving the broadcast packet, a node waits for a random number of time slots to hear the same packet from other re-broadcasting nodes. Each node is associated with a counter c to track the number of times of the same broadcast message received. Whenever $c \geq K$, the node is forbidden to rebroadcast, where K is a chosen counter threshold. They analyzed the counter threshold and the extra area that can be benefited from a rebroadcast node, and found that when $K \geq 4$ the expected additional coverage is below 0.05%.

The main idea of the proposed algorithm in [7] is that a node need not rebroadcast a message if all its neighbors have received the message. The algorithm consists of two phases: local neighbor discovery and data broadcasting. Each node in the algorithm waits for a period of time to collect information of neighboring nodes and rebroadcast nodes. As a result, the algorithm requires high overheads and defers the completion time of the broadcast significantly.

A technique restricts the number of rebroadcast as much as possible by selecting a subset of neighbors, called *multipoint relays* [10], which cover the same network region as the complete neighbors do. In this paper, Qayyum *et al.* also showed the problem of finding a minimal multipoint relay set is NP-complete by reducing from the dominating set problem. Therefore, they proposed a heuristic algorithm to compute a multipoint relay set of cardinality at most log n times the optimal multipoint relay number, where n is the number of nodes in the network. Since the information needed to compute the multipoint relays is the set of one-hop and two-hop neighbors, which are collected by sending hello messages periodically, multipoint relays requires high overheads.

3. Deriving bounds on the size of required re-broadcast nodes for broadcasting

In this section, the size of required re-broadcast nodes in mobile ad hoc networks is estimated by using different techniques. Throughout this article, we define n to be the number of nodes deployed randomly in the network, and r the communication radius of each mobile node.

3.1 The first formula

An ad hoc network can be modeled as a geometry graph. A *geometric graph* $G=(V, r)$ with nodes placed in 2-dimension space R^2 and edge set $E=\{(i, j) \mid d(i, j) \leq r$, where i, $j \in V$ and function $d(i, j)$ denotes the Euclidian distance between node i and node $j\}$. Theoretically, the required rebroadcast nodes can be represented by a minimum connected dominating set. Let γ_c be the size of the minimum connected dominating set, and γ the size of the minimum dominating set. Since a connected dominating set is also a dominating set, we have that $\gamma \leq \gamma_c$. A subset S of V is called an *independent set* of $G=(V, E)$ is no two vertices of S are adjacent. We also let α denote the maximum size of independent set in a graph.

It seems difficult to obtain the exact size of the minimum connected set even we collect the overall topology of interested ad hoc networks in a snapshot. However, the degree information of the corresponding graph would help to estimate the size of required broadcast nodes due to the following theorems.

Theorem 1[1]: For any connected graph, $\gamma_c \leq 3\gamma - 2$.

Theorem 2 [13]: Every n-vertex graph with minimum degree σ has a dominating set of size at most $n \times \frac{1+\ln(\sigma+1)}{\sigma+1}$; thus we have $\gamma \leq n \times \frac{1+\ln(\sigma+1)}{\sigma+1}$.

Combining Theorem 1 and 2, we have the following theorem.

Theorem 3: Every n-vertex graph with minimum degree σ, we have $\gamma_c \leq 3 \times (n \times \frac{1+\ln(\sigma+1)}{\sigma+1}) - 2$.

Sometime, the minimum degree σ is still difficult to know; however, when n mobile nodes are regularly distributed inside a square of length l units as shown in Figure 2, we may use the average degree of a vertex in the corresponding geometry graph to estimate the minimum degree σ.

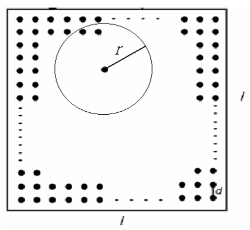

Figure 2. Arrange n nodes in $l \times l$ region

The average degree of a vertex in the geometry graph is the number of points covered by the node's communication range r (i.e., the number of points covered by a circle with radius r). To exactly compute the number of covered points, we need to tackle an unsolved problem in number theory [3]. Therefore, we adopt a different approach. The distance d between two points in Figure 2 is $d = \frac{l}{\sqrt{n}+1}$. Then the number of points in the dash-line triangle is $1+2+3+\left(\left\lfloor \frac{r}{d} \right\rfloor + 1\right) = \left(\left\lfloor \frac{r}{d} \right\rfloor + 1\right) \times \left(\left\lfloor \frac{r}{d} \right\rfloor + 2\right) / 2$. Consequently, the estimated number of points in the quarter of a circle (as shown in Figure 3) is $\left(\left\lfloor \frac{r}{d} \right\rfloor + 1\right)\left(\left\lfloor \frac{r}{d} \right\rfloor + 2\right) / 2 \times \frac{1/4\pi r^2}{1/2 r^2} = \left(\left\lfloor \frac{r}{d} \right\rfloor + 1\right) \times \left(\left\lfloor \frac{r}{d} \right\rfloor + 2\right) \times \frac{\pi}{4}$. Therefore, the estimated number of points in a circle is $\left(\left\lfloor \frac{r}{d} \right\rfloor + 1\right) \times \left(\left\lfloor \frac{r}{d} \right\rfloor + 2\right) \times \pi$.

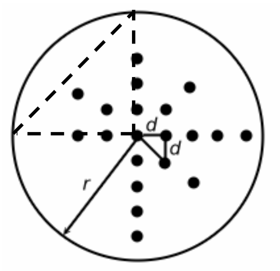

Figure 3. The expected number of nodes covered by a node

Finally, we obtain the first upper bound of the minimum number of required re-broadcast nodes by the help of Theorem 3.

Formula 1: We probably require at most $n\times\frac{1+\ln(T+1)}{T+1}$ nodes with communication range r to rebroadcast packets where $T=\left(\left\lfloor\frac{r}{d}\right\rfloor+1\right)\times\left(\left\lfloor\frac{r}{d}\right\rfloor+2\right)\times\pi$ and $d=\frac{l}{\sqrt{n}+1}$, when the $l\times l$ deployed area is fully covered by a connected mobile network.

Another precise formula, which is listed below, for estimating the average number of neighboring nodes in a mobile ad hoc network has been proposed by Yen and Yu [15]. We also can use it instead of T when applying Formula 1.

Theorem 4: The average (expected) neighbors of a node in an ad hoc network is $(n-1)\times\left(\frac{\frac{1}{2}r^4-\frac{4}{3}lr^3-\frac{4}{3}mr^3+\pi r^2 ml}{m^2l^2}\right)$, here A is a $l\times m$ rectangle deployed space and r is the communication radius of each mobile node.

3.2 The second formula

In this subsection, we try to derive another formula by estimating an upper bound of the maximum size of independent set. When the $l\times l$ deployed area is fully covered by a connected mobile network, the maximum size of independent set (α) is equivalent to the maximum number of circles each of which does not cover the center of any other circle. First, we deploy k circles in each of k rows where $k=\lfloor l/2r\rfloor$ (totaling k^2 circles). Moreover, we can also add a circle in the middle of every two neighboring circles (Figure 4) (totaling $2\times k(k-1)$ circles) and one circle in the center of four neighboring circles (totaling $(k-1)^2$ circles). Thus we can easily obtain that $\alpha\leq k^2+2\times k(k-1)+(k-1)^2=(2k-1)^2=(2\times\lfloor l/2r\rfloor-1)^2$.

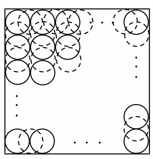

Figure 4. The maximum size of independent set

By Theorem 5 and above discussion, we can obtain the second upper bound.

Theorem 5[1]: For any connected graph, $\gamma_c \leq 2\alpha$-1.

Formula 2: We require at most $2\times(2\times\lfloor l/2r \rfloor-1)^2$ nodes with communication range r to rebroadcast packets when the $l\times l$ deployed area is fully covered by a connected mobile network.

3.3 The third formula

Another different approach is introduced in the section. Let $X_n=\{x_1, x_2, \ldots, x_n\}$ be a set of independently and uniformly distributed random points. We use $\Psi(X_n, r, A)$ to denote the *random geometric graph* (RGG) of n nodes on X_n with radius r and placed in area A. RGGs consider geometric graphs on random point configurations. Applications of RGGs include communications networks, classification, spatial statistics, epidemiology, astrophysics and neural networks [8].

A RGG $\Psi(X_n, r, A)$ is suitable to model an ad hoc network $N=(n, r, A)$ consisting of n mobile devices with a transmission radius of r unit length that are independently and uniformly distributed at random in an area A. When each vertex in $\Psi(X_n, r, A)$ represents a mobile device, each edge connecting two vertices represents a possible communication link because they are within the transmission range of each other. A geometric graph and its representing network are shown in Figure 5. In the example, area A is a rectangle that is used to model the deployed area such as a meeting room. Area A, however, can be a circle, other different shapes, or even infinite space.

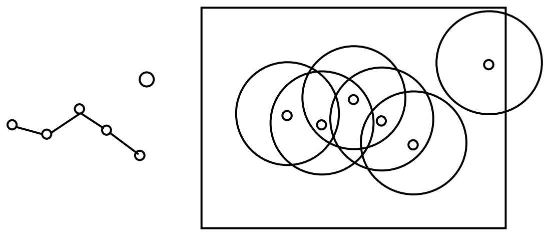

Figure 5. A random geometric graph G with its representing network N=(6, r, A), where A is a rectangle

Computing the probability of occurrence of specific subgraphs of a given RGG becomes important issues for modeling and evaluating the fundamental properties of wireless ad hoc networks. A simple algorithm for selecting a dominating set has been proposed Wu and Li [14]. Their algorithm selects the center of the induced path with length 2 (that is p_2) as a member of the desired set.

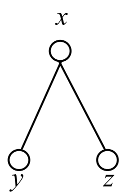

Figure 6. The subgraphs of p_2

We can estimate the number of dominating set produced in their algorithm.

Theorem 6[16]: For arbitrary three distinct edges $e_i=(u, v)$, $e_j=(u, w)$, and $e_k=(v, w)$ in a $\Psi(X_n, r, A)$, the probability of these three edges form p_2 is $\left(\frac{3\sqrt{3}}{4}\right)\pi r^4/|A|^2$, where $u\neq v\neq w$.

With the help of Theorem 6, we can obtain the number of p_2, denoted by $N(p_2)$.

Theorem 7: The $N(p_2)$ in a $\Psi(X_n, r, A)$ is $3\times\binom{n}{3}\left(\frac{3\sqrt{3}}{4}\right)\pi r^4/|A^2|$.

Proof: First, we compute the number of hidden-terminal pairs in any RGG. Since each hidden-terminal pair consists of three distinct labeled vertices, we set S to be the selected three-vertex set. Since there are $\binom{n}{3}$ different combinations for selecting three from n vertices and three different settings for labeling one from three as the center of the hidden-terminal pairs (*i.e.* the internal node of the induced path with length 2), we have the number of hidden-terminal pairs is $3\times\binom{n}{3}\times\Pr(G_S=p_2)=3\times\binom{n}{3}\left(\frac{3\sqrt{3}}{4}\right)\pi r^4/|A|^2$ by Theorem 6. ■

Then we can obtain the third formula listed as follows.

Formula 3: The expected number of dominating set selected by Wu and Li's algorithm in a $\Psi(X_n, r, A)$ is $n\times\binom{n-1}{2}\times\left(\frac{3\sqrt{3}}{4}\right)\pi r^4/|A|^2$.

Proof: Let S be the selected three-vertex set. Since there are n possible centers in the system and $\binom{n-1}{2}$ combinations for other two vertices, the number of the desired dominating set is $n\times\binom{n-1}{2}\times\Pr(G_S=p_2)=n\times\binom{n-1}{2}\times\left(\frac{3\sqrt{3}}{4}\right)\pi r^4/|A|^2$ by Theorem 7. ■

4. Two broadcasting protocols

In the section, we design two protocols: the cluster-head-early method and the connected-dominating-set method for broadcasting in ad hoc networks.

4.1 The cluster-head-early method

The first approach is called the *cluster-head-early method* (CHEM) because it tries to make cluster-head nodes carry rebroadcasts early. It is helpful to first introduce a technique called the *simple clustering method* (SCM) that has been used for designing algorithms for the maximal independent set problem [4] and for devising the adaptive clustering for mobile wireless networks [5]. SCM associates each node with a distinct integer randomly selected from the set {1, 2, 3, …, n}, where n is a fairly large integer

(See Figure 7). For convenience, we name each node the assigned integer from now on.

According to their assigned numbers, SCM partition nodes of the network into three kinds of sets. A *cluster head* is a node whose assigned integer is the largest as comparing to its neighboring nodes. Note that a cluster head cannot be a neighbor of another cluster head. A *cluster member* is a node that at least one of its neighbors (not itself) is a cluster head. An independent node is a node that is neither a cluster head nor a cluster member. Figure 7 shows an example. The cluster head set, the cluster member set, and the independent node set are {48, 65, 84, 96}, {2, 20, 35, 40, 54, 87}, and {44} respectively.

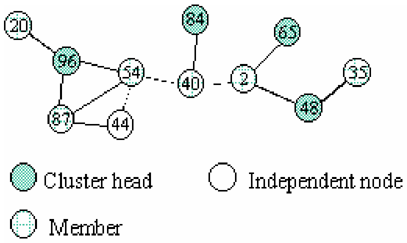

Figure 7. Partition network nodes into three sets

However, CHEM is different from SCM. The intuitive idea of CHEM is described as follows. The rebroadcasts of cluster members would be redundant, provided cluster heads have already re-broadcasted the same packet. Cluster heads in our protocol are therefore planned to receive and rebroadcast the packet early. On the contrary, cluster members and independent nodes are required to wait for a period of time after receiving the first broadcast packet.

Note that SCM will select nodes with large weights as cluster heads. Suppose node *B* (with assigned integer *IntB*) receives the first packet from node *A* (with assigned integer *IntA*). If $IntB > IntA$ then node *B* in CHEM forwards the packet to its neighbors without hesitation. Otherwise, node *B* will wait for $t=(1-(IntA/IntA+IntB))\times T$ time to receive the same packet from other nodes, where T is a predefined constant. Note that the value of t is designed intentionally to inversely proportional to the value of *IntA*.

We also associate each node with a counter c (this concept is borrowed from the counter-based scheme [6, 12]) to record the number of packets received. A counter threshold C is also chosen. Whenever $c \geq C$ in the period of t time, the node is forbidden to rebroadcast. An example is shown in the following figure.

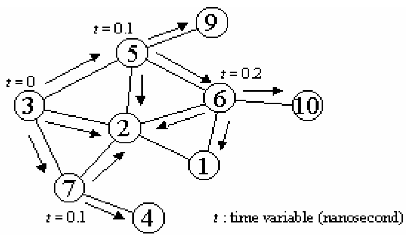

Figure 8. An example for CHEM

Suppose that each single hop communication takes 0.1 seconds, and the count threshold *C*=3 and the selected constant *T*=1 second. In Figure 8, node 3 (the initial node) starts to broadcast a packet at *t*=0. Node 5, node 2, and node 7 receives the packet from node 3 at *t*=0.1. Since we have 5>3 (and 7>3), node 5 (and node 7) rebroadcasts the packet immediately. On the other hand, node 2 is scheduled to wait for 0.4 (=1-(3/(3+2))=2/5) seconds due to 2<3. Similarly, node 6 (at *t*=0.2) receives the packet and rebroadcast the packet immediately because 6>5. When *t*=0.3, node 2 receives four packets from its neighbors. Since *c*=4>3=*C* (the count threshold), node 2 is forbidden to rebroadcast the packet.

4.2 The connected-dominating-set method

The *connected-dominated-set method* (CDSM) consists of two phrases: the gathering phrase and the selection phrase. In the gathering phrase, every node in CDSM tries to collect neighboring node information through one-time flooding. So that the desired rebroadcast nodes can be determined and its size can also be reduced in the selection phrase. In a word, CDSM is devised for finding a small set of nodes as a connected dominating (CD) set (in graph term) which carry broadcast. We name these nodes *dominating nodes*.

Suppose that each node *x* is allocated with a variable *nblist* (initialized to be empty), which denote the set of collected neighboring nodes of *x*. In the beginning of the selection phrase, the *nblist* of a broadcasting node will be sent to all its neighbors accompanied with the subsequent broadcasting message. Then the following three rules are applied to select suitable nodes into the CD set, each of which takes responsibility for hereafter broadcasting. A dominating node must satisfy the following three rules:

Rule 1: A dominating node received the first *nblist* and its *nblist* is not covered by sender's *nblist*.

Rule 2: A dominating node must have two neighboring nodes *x* and *y* such that node *x* is not a neighbor of *y*.

Rule 3: If a dominating node receives another node's *nblist* again, the dominating node will check whether its *nblist* is included in the union of the two collected *nblists*. If the answer is not, it will still be the dominating node. Otherwise, it is not a dominating node anymore.

5. Conclusion

In this work, we have derived three formulas for estimating the number of required broadcasting packets by taking different approaches. We also proposed two broadcasting protocols without exploiting the mechanisms of hello messages or cluster formation. We have conducted extensive simulations by using ns-2 and demonstrated that our protocols outperform the counter-based scheme and flooding in rebroadcast node and latency. Owing to page limit, however, this work excludes simulation results intentionally.

References

1. P. Duchet and H. Meyniel, “On Hadwiger’s number and stability number,” *Ann. Discrete Math.*, vol. 13, pp. 71-74, 1982.
2. M. R. Garey and D. S. Johnson, *Computers and Intractability: A guide to the Theory of NP-completeness*, Freeman, San Francisco, 1978.
3. Peter Hall, *Introduction to the Theory of Coverage Process*, John Wiley and Sons, New York, 1988.
4. Michael Luby, “A simple parallel algorithm for the maximal independent set Problem,” *SIAM Journal on Computing*, vol.15, pp. 1036-1053, 1986.
5. C.R. Lin and M. Gerla, “Adaptive clustering for mobile wireless networks,” *IEEE Journal on Selected Areas in Communications*, vol.15, pp. 1265-1275, 1997.
6. Sze-Yao Ni, Yu-Chee Tseng, Yuh-Shyan Chen, and Jang-Ping Shen, “The broadcast storm problem in a mobile ad hoc network,” *Mobicom’99*, pp. 151-162.
7. Wei Peng and Xi-Cheng Lu, “On the reduction of broadcast redundancy in mobile ad hoc networks,” *MobiHoc*, 2000, pp. 129 -130.
8. Mathew D. Penrose, *Random Geometric Graphs*, Oxford University Press, 2003.
9. C.E. Perkins and E.M. Royer, “Ad-hoc On-demand Distance Vector routing, “ *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, 1999, pp. 99-100.
10. Amir Qayyum, Laurent Viennot, and Anis Laouiti, “Multipoint relaying: An efficient technique for flooding in mobile wireless networks,” *INRIA, Tech. Rep.*, RR-3898, 2000.
11. E. M. Royer and C.-K. Toh, "A review of current routing protocols for ad hoc mobile wireless networks," *IEEE Personal Communication*, pp. 46-55, 1999.
12. Yu-Chee Tseng, Sze-Yao Ni, and En-Yu Shih, “Adaptive approaches to relieving broadcast storms in a wireless multi-hop mobile ad hoc network,” *International Conference on Distributed Computing Systems*, 2001, pp. 207-216.
13. D. B. West, *Introduction to Graph Theory*, Prentice Hall, 2001.
14. Jie Wu and Hailan Li, “Domination and Its Applications in Ad Hoc Wireless Network with Unidirectional Links,” *International Conference on Parallel Processing*, 2000, pp. 189 –197.
15. L.-H. Yen and C. W. Yu, “Link Probability, Network Coverage, and Related Properties of Wireless Ad Hoc Networks,” *The 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, 2004, pp.525-527.
16. C. W. Yu and Li-Hsing Yen, “Computing subgraph probability of random geometric graphs: Quantitative analyses of wireless ad hoc networks,” *Springer-Verlag Lecture Note in Computer Science*, vol. 3731, pp. 458-472, 2005.