Efficient Flooding in Mobile Ad Hoc Networks
Hui Zeng\(^1\), Minming Li\(^2\), Hai Liu\(^2\), Xiaohua Jia\(^2\)
\(^1\)School of Computer Science, Wuhan University
\(^2\)Dept of Computer Science, City University of Hong Kong

Abstract:
Flooding is one of the most fundamental operations in mobile ad hoc networks. Different from broadcast that requires a broadcast routing for transmission of large amount data, flooding is the one-off operation which is usually used for dissemination of control packets. Most of the major routing protocols rely on flooding for disseminating route discovery, route maintenance, and topology update packets. However, pure flooding in which each node retransmits the packet once suffers from the problems of excessive redundancy of messages, resource contention, and signal collision. A lot of efficient flooding schemes have been proposed to avoid these problems. The chapter investigates existing solutions on efficient flooding schemes, and classifies these solutions into three categories based on the information each node keeps: 1) no need of neighbor information; 2) 1-hop neighbor information; 3) 2-hop or more neighbor information. We focus on the most representative works in each category, and discuss various algorithms, protocols, and techniques which make flooding more efficient. Challenges and future work are pointed out at the end.

Keywords: efficient flooding, broadcast, mobile ad hoc networks, wireless networks.

1. Introduction

1.1 Mobile Ad Hoc Networks
Mobile ad hoc networks (MANETs) have received significant attention in recent years. A mobile ad hoc network usually consists of wireless mobile nodes that communicate with each other, in the absence of a fixed infrastructure. Thus, it is suitable for use in situations where an infrastructure is unavailable or to deploy one is not cost effective. Current applications of MANETs include outdoor business meeting, disaster relief operations, battlefield, and so on.

“Mobile” and “wireless”, each of these two words enforces a list of requirements, and the daunting task is to fulfill them to their best. The *mobility* means that nodes in the networks moves at any time, which implies the short duration of neighbor-hood and topology information received at any moment. In order to handle mobility correctly and efficiently, the information need to be updated regularly. The wireless nature of the medium implies the limited bandwidth capacity, which is further reduced by the high bit error rate in radio transmissions. Being such a precious resource, the bandwidth in a wireless network usually requires prudent consumption. Therefore, to reduce the unnecessary use of the bandwidth is one of the main tasks in designing a protocol using wireless links. Moreover, unlike wired communication in
traditional networks, wireless communication in MANETs suffers from congestion issues. When an ad hoc node transmits a message, all neighbors within its transmission range receive or overhear the message. If there are concurrent transmissions in the network, it may result in collisions and loss of the messages. The requirements of mobile nature and wireless nature are often opposite to each other and the compromise is to manage the mobility of nodes while using minimum of the bandwidth resources.

The traffic generated to manage the mobility of nodes in a network is mostly the information that a node declares about its relative movements, its new position, or its new neighborhood, etc. A lot of message passing is required in the network to keep the information consistent at each node, due to regular announcement of the changes caused by mobility, or failure of links, etc.

Because routes between two nodes in MANETs may consist of hops through other nodes in the network, node mobility can cause frequent unpredictable topology changes and the task of finding and further maintaining routes in MANETs is nontrivial. [17]

1.2 Flooding scheme in the network

Flooding is one of the most fundamental operations in MANETs. Most of the major routing protocols, like DSR [13], AODV [27], LAR [17], ZRP [10], etc., rely on flooding for disseminating route discovery, route maintenance, or topology update packets. Flooding is a very frequently invoked function in MANETs. Therefore, an efficient implementation of the flooding scheme is crucial in reducing the overhead of routing protocols and improving the throughput of networks.

Efficient flooding schemes are different from the broadcast mechanisms discussed in [19][44]. The broadcast mechanism is used in transmission of a large amount of data or stream media data. These applications require an efficient broadcast route before the actual transmission of data, so that data can be transmitted efficiently along the pre-found route. In contrast, flooding is usually used in dissemination of control packets, which is a one-off operation and it does not need routing beforehand.

This report surveys the current works on efficient flooding schemes in mobile ad hoc networks. We classify them according to the information each node keeps when the flooding occurs: 1) no need of neighbor information; 2) 1-hop neighbor information; 3) 2-hop neighbor information. The rest of the report is organized as follows. Section 2 discusses the typical example in the first category: pure flooding. We show its serious problems and further introduce an ameliorated scheme called probabilistic-based scheme. Section 3 discusses some efficient flooding schemes based on 1-hop information. Flooding schemes based on 2-hop information are discussed in Section 4. The comparison among the performance of these existing flooding schemes is discussed in Section 5.
2. Simple Flooding

Flooding schemes in this category have the common characteristic that each node in the network re-transmits the message without the need of other nodes’ information.

2.1 Pure flooding

Pure flooding also called blind flooding, is the simplest flooding technique. The basic idea of this approach is every node in the network retransmits the flooding message when it is the first time to receive it [9] [12].

A node, on receiving a broadcast message for the first time, has the responsibility to rebroadcast the message. It costs $n$ transmissions in a network with $n$ nodes. This simple scheme guarantees that a flooding message can reach all nodes if the network is connected and there is no collision.

However, this algorithm will generate excessive amount of redundant network traffic when all nodes in the network are transmitting the flooding message. This will consume a lot of energy of the mobile nodes and also cause congestion in the network. Furthermore, due to the broadcast nature of radio transmissions, there is a very high probability of signal collision when all nodes flood the message in the network at the same time, which will cause more re-transmissions or some nodes to fail to receive the message. In other words, pure flooding can result in some drawbacks in MANETs as following:

- **Redundant rebroadcasts**: When a mobile node decides to rebroadcast a message to its neighbors, all its neighbors already have the message.
- **Contention**: After a node broadcasts a message, if many of its neighbors decide to rebroadcast the message, these transmissions (which are all from nearby nodes) may severely interfere with each other.
- **Collision**: Because of the deficiency of back off mechanisms and the absence of collision detections, collisions are more likely to occur and also cause more damage.

Collectively, the above phenomena are referred to as the broadcast storm problem [26]. Sinha et al. [36] claimed that “in moderately sparse graphs the expected number of nodes in the network that will receive a broadcast message was shown to be as low as 80%”.

2.2 Probabilistic flooding scheme

One approach to alleviate the broadcast storm problem is to inhibit some nodes from rebroadcast to reduce the redundancy, and thus contention and collision. Sze-Yao Ni et al. [26] presented a probabilistic scheme that use a probabilistic
rebroadcasting and differentiate timing of rebroadcasts to avoid redundancy and collisions.

The basic idea of probabilistic flooding schemes is that each node forwards a flooding message with probability $P$ upon receiving it for the first time. Clearly, when $P=1$, this scheme is equivalent to pure flooding.

The probabilistic schemes can be classified into four types: counter-based, distance-based, location-based and cluster-based. These schemes differ in how a node estimates redundancy and how it accumulates knowledge to assist its decision. Except the last scheme, which relies on some local connectivity information, all schemes operate in a fully distributed manner.

Counter based scheme: when a node tries to rebroadcast a message, the rebroadcast message may be blocked by busy medium, back off procedure, or other queued messages. There is a chance for the node to hear the same message again and again from other rebroadcasting nodes before the node actually starts transmitting the message.

Distance-based scheme: This scheme uses the relative distance between hosts to make the decision [26].

Location-based: This scheme uses the location information of the broadcasting nodes to make the decision. The location information may be supported by positioning devices such as GPS (Global Positioning System) receivers [38], it is also used to facilitate route discovery in a MANET [9].

Cluster-based: Yoav Sasson et al [31] further investigate the probabilistic scheme and show that the success rate curves for probabilistic flooding tend to become linear for the network with low average node degree, and resemble a bell curve for the network with high average node degree.

In these schemes, a non-redundant transmission might be dropped out, without being forwarded further. Therefore, it will make some nodes in the network fail to receive the flooding message (i.e., these nodes are not reached by the flooding). Besides this deliverability problem, another major concern of these techniques is the difficulty in setting the right threshold value in various network situations [32].

3. 1-Hop Neighbor Knowledge Methods

Schemes in this category assume that each node keeps the information of 1-hop neighbors. The 1-hop neighbor information can be obtained by exchanging the HELLO message in MAC layer protocols [11].
A major issue in the schemes that use 1-hop or 2-hop (discussed in the next section) information is the selection of a subset of neighbors for forwarding the flooding message. There are two strategies for choosing forwarding nodes: sender-based, where each sender nominates a subset of its neighbors to be the next hop forwarding nodes, and receiver-based, where each receiver of a flooding message makes its own decision on whether it should forward the message or not. The schemes proposed in [30] [24] [20] are sender-based, while the schemes proposed in [4] [37] [34] [42] [43] [21] [29] are receiver-based.

The purpose of studying flooding technique is to guarantee 100% deliverability of flooding messages and reduce redundant rebroadcasts or avoid collisions simultaneously. Flooding schemes in [3], [45], [22], and [25] are all 1-hop flooding schemes that guarantee 100% deliverability of the flooding message. Authors in [25] proved that a 1-hop flooding scheme achieves 100% deliverability if and only if for each node $s$ the neighbor’s area of $s$ can be reached by $F(s)$, where $F(s)$ is defined as a subset of $s$’s neighbors that are selected to forward the flooding message ($F(s)$ includes $s$ itself). The neighbor’s area of a node $s$ is the whole shadow area shown in Figure 1, where $a$, $b$, $c$ is the whole neighbors of $s$’s.

![Figure 1. Neighbor’s area of node s](image)

We cite the following definitions from [25] to better explain the idea:

**Def 1. Coverage disk of a node.** The coverage disk of node $s$, denoted by $d(s)$, is a disk that is centered at $s$ and whose radius is the transmission range of $s$.

**Def 2. Coverage area of a node-set.** The coverage area of a set of nodes $A$, denoted by $C(A)$, is the union of coverage disks of nodes in $A$.

**Def 3. Neighbor’s coverage area.** The neighbor’s coverage area of node $s$ is the union of coverage disks of all $s$’s neighbors plus $s$ itself, i.e., $C(N(s) \cup \{s\})$.

**Def 4. Boundary of neighbor’s area.** The boundary of neighbor’s area of node $s$ is the boundary of the area of $C(N(s) \cup \{s\})$. 
Def 5. **Minimum forwarding set** $F_{\text{min}}(s)$, is the smallest $F(s)$ such that $C(F(s))$ covers the neighbor’s area of $s$.

### 3.1 Flooding with Self Pruning (FSP)

The simplest flooding scheme based on 1-hop neighbor knowledge is flooding with self pruning (FSP) proposed by Lim and Kim [22].

FSP is a receiver-based scheme which uses 1-hop information. A sender forwards a flooding message by attaching all of its 1-hop neighbors to the message. A receiver compares its own 1-hop neighbors with the node list in the message; it will not forward the message if all its 1-hop neighbors are already included in the list, otherwise it forwards the message as a sender.

The complexity of FSP is $O(\Delta)$, where $\Delta$ is the node’s maximum degree in the network. And the effect of self pruning is shown most significantly in the perimeter of the network. The nodes in the center are more likely to have nonoverlapping neighbor nodes. Self pruning requires extra transmission overhead of exchanging neighborhood information. To reduce the overhead, each node can store the received adjacent node list in their cache [42].

Therefore, Williams and Camp [41] showed that the improvement of FSP is very limited in most of network conditions although this technique can reduce the flooding cost and guarantee packet deliverability at the same time.

### 3.2 Edge forwarding

One notable work of efficient flooding that uses 1-hop neighbor information is Edge Forwarding [3]. It tries to minimize the flooding traffic by leveraging location information so that broadcast retransmission is limited only to the nodes near the perimeter of each broadcast coverage.

In the scheme, each node’s transmission coverage is partitioned into six equal size sectors. For example, as shown in Figure 2, the transmission coverage of the node $S$ is partitioned into six equal-size regions and denoted as $S_{P1}$, $S_{P2}$, $S_{P3}$, $S_{P4}$, $S_{P5}$, $S_{P6}$, respectively. We say a node is $S$‘s $Pi$ neighbor, if the node is currently inside partition $S_{Pi}$, where $1 \leq i \leq 6$. For example, $O$ is $S$‘s $P1$ neighbor as $O$ is located in $S_{P1}$.
Figure 2. Transmission Coverage Partitioning

Suppose $S$ is at $(x_s, y_s)$ and its 1-hop neighbor $O$ is at location $(x_o, y_o)$. We can decide which partition $O$ belongs to by some simple calculations as follows:

- If $x_s \leq x_o$ and $y_s \leq y_o$, then $O$ is in $S_{P1}$ if $\frac{x_o - x_s}{\text{dist}(S, O)} \geq \frac{1}{2}$; otherwise, $O$ is in $S_{P2}$;
- If $x_s \geq x_o$ and $y_s \leq y_o$, then $O$ is in $S_{P2}$ if $\frac{x_o - x_s}{\text{dist}(S, O)} \leq \frac{1}{2}$; otherwise, $O$ is in $S_{P3}$;
- If $x_s \geq x_o$ and $y_s \geq y_o$, then $O$ is in $S_{P4}$ if $\frac{x_o - x_s}{\text{dist}(S, O)} \geq \frac{1}{2}$; otherwise, $O$ is in $S_{P5}$;
- If $x_s \leq x_o$ and $y_s \geq y_o$, then $O$ is in $S_{P6}$ if $\frac{x_o - x_s}{\text{dist}(S, O)} \leq \frac{1}{2}$; otherwise, $O$ is in $S_{P6}$;

Then, the authors of [3] proposed two forwarding rules, basic forwarding and advanced forwarding, by which a node can determine whether or not it should forward a broadcast packet. When a node forwards a broadcast, it adds its ID to the packet header to inform each recipient of the message.

We use Figure 3 to explain the basic forwarding rule. Without loss of generality, assume $O$ is in $S_{P1}$. $O$’s partition lines divide $S_{P1}$ into 6 sub partitions: $S_{P11}, S_{P12}, S_{P13}, S_{P14}, S_{P15}$ and $S_{P16}$. When $O$ receives a broadcast from $S$, it first determines whether there is any of its $P_1$ neighbors inside $S_{P11}$. If no such node exists, $O$ forwards the broadcast. Otherwise, $O$ continues to check its 1-hop neighbor in $P_2$, $P_3$, $P_4$, $P_5$, and $P_6$, sequentially. If there is at least one node in each of $S_{P1i}$, where $1 \leq i \leq 6$, $O$ does not forward the broadcast. This strategy is quite energy efficient in practice.

Figure 3. Node $O$ divides $S_{P1}$ into 6 partitions
It is easy to show that the complexity of this procedure is $O(n)$, where $n$ is the number of $O$’s 1-hop neighbors. One can prove that there is at least one node in each of $S_{P1i}$, where $1 \leq i \leq 6$, that will forward the broadcast. Therefore, the broadcast can reach all of $O$’s 1-hop neighbors that are outside of $S$’s transmission coverage even if $O$ does not forward the packet [3].

Notice that a large portion of $O$’s $P_3$ partition has already been covered by $S$’s broadcast; and if there is any node inside the uncovered area, then very likely it is within 1-hop distance to all nodes inside $S_{P12}$. We do not need to consider the nodes inside $O_{P4}$ since this partition is completely covered by $S$’s broadcast. Based on this observation, an advanced forwarding rule is proposed—$O$ does not need to forward a broadcast from $S$ if the following three conditions are satisfied [3]:

1. there is at least one node in each of $S_{P11}$, $S_{P12}$, and $S_{P16}$.
2. all $O$’s $P_3$ neighbors beyond 1-hop distance to $S$ are within 1-hop distance to all hosts inside $S_{P12}$.
3. all $O$’s $P_5$ neighbors beyond 1-hop distance to $S$ are within 1-hop distance to all hosts inside $S_{P16}$.

We note that under the basic forwarding rule, a node does not need to forward a broadcast unless it is close to the edge of some broadcast partition. This characteristic eliminates a significant portion of unnecessary broadcast forwarding. The advanced forwarding rule enhances this by pushing the forwarding responsibility to only the nodes close to the transmission perimeter. This strategy, however, incurs more computation.

The authors of [3] compare their scheme with FSP using simulations. They show that the performance of Edge Forwarding is better with the increase of network domain area and more suitable for flooding in large scale wireless ad hoc networks. In particular, edge forwarding can be easily incorporated into many existing routing protocols without any additional control overhead. It also discusses the situation when the nodes’ transmission radius is not stable (a constant) in the network.

### 3.3 Vertex Forwarding

Xinxin Liu et al. [20] proposes a flooding scheme called vertex forwarding which also tries to minimize the flooding traffic by leveraging location information of 1-hop neighbor nodes. It is a sender-based algorithm which assumes a hexagonal grid in the network field to guide the flooding procedure. When a node has a message to flood out, it assumes that it is located at a vertex of the virtual hexagonal grid, and the neighbors located at the adjacent vertices will be selected to forward the message. If there is no node located at these vertices, the nodes that are nearest to the vertices in terms of hops will be selected to forward the message once. The selection of the
forward candidate set is based on transmission coverage partition as the Edge Forwarding [3] (See also Figure 2).

It states that the efficient flooding can be considered as the covering problem which is described as follows: “What is the minimum number of circles required to completely cover a given 2-dimensional space?” We know that no arrangement of circles could cover the plane more efficiently than the hexagonal grid arrangement [16]. For this reason, the vertex forwarding scheme is efficient, and is able to reduce the number of forwarding nodes to be close to the lower bound (the number of MCDS—Minimum Connected Dominating Sets [43], [40] ). It is verified in the simulation.

The vertex forwarding scheme is compared with three existing flooding protocols: Pure flooding, edge forwarding and CDS-based flooding (it requires 2-hop neighbor knowledge) in the simulation. The flooding efficiency is evaluated by the Forwarding ratio (a ratio of forward nodes in a flooding operation over the total number of nodes in the network) and Delivery ratio (a ratio of the number of nodes that received packets to the number of the nodes in the network for one flooding operation). The results show that: 1) vertex forwarding can drastically reduce redundant retransmission of flooding messages and significantly outperform edge forwarding and CDS-based schemes. 2) Vertex forwarding and edge forwarding scheme are both highly scalable with respect to the network size, while CDS-based scheme is better in a smaller network but becomes worse when network size increases. 3) The delivery ratio of vertex forwarding is almost 100% and is not affected by node density, while performance of pure flooding and edge forwarding degrades as node density increases.

3.4 Efficient Flooding based on 1-hop Information (EF1)

Another flooding scheme is proposed by Hai Liu et al. [25]. It is based on 1-hop neighbor information and is easy to implement and light-weight in overhead.

In EF1, they assume the network is connected and represent it as a unit disk graph $G= (V, E)$ where all nodes in the network have the same transmission range $R$. Each node $v$ in $V$ has a unique $ID$, denoted by $id (v)$. Let $N (v)$ denote the set of neighbor nodes of $v$. In other words, nodes in $N (v)$ are within the transmission range of $v$ and can receive signals transmitted by $v$. Node $v$ needs to know the information of its neighbor, including their IDs and their geographic locations. This 1-hop neighbor information can be easily obtained from the HELLO messages periodically broadcasted by each node.

The basic idea of this flooding scheme is as follows. When a node (called the source) has a message to flood, it computes a subset of its neighbors as forwarding nodes and attaches the list of the forwarding nodes to the message. After that, it
transmits the message out. Then, every node in the network does the same as follows. Upon receiving a flooding message, if the message has been received before, it is discarded; otherwise the message is delivered to the application layer and the receiver checks whether it is in the forwarding list. If yes, it computes the next hop forwarding nodes among its neighbors and transmits the message in the same way as the source. The message will eventually reach all the nodes.

We can see that the forwarding node selection strategy in EF1 is sender-based. We introduce this algorithm in three parts: 1) forwarding node selection, where a node selects a subset of its 1-hop neighbors to forward the flooding message; 2) forwarding node optimization, which further reduces the number of forwarding nodes by removing the nodes that are already covered; and 3) mobility handling, where each node incrementally updates its forwarding set in response to the topology change.

1) Forwarding node selection
   The principle of selecting the forwarding node is to minimize $F(s)$ and achieve 100% deliverability at the same time.

   Suppose $s$ is a node that receives a flooding message for the first time and $s$ also appears in the forwarding list attached to the message ($s$ could be the original source of the message). The node $s$ then computes the next hop forwarding nodes from its neighbors. Since $s$ only has 1-hop neighbor information, it does not know who the 2-hop neighbors are. To achieve 100% deliverability, $F(s)$ must reach the entire neighbor’s area of $s$.

   To minimize $F(s)$, every node in $F(s)$ must contribute to the neighbor’s boundary of $s$; otherwise, this node can be removed from $F(s)$ without affecting the coverage area of $F(s)$. Therefore, computing the minimal $F(s)$ is to find a subset of $N(s)$ such that every node that contributes to the neighbor’s boundary of $s$ is in $N(s)$.

   ![Figure 4. Example of arcs](image)

   In order to present the algorithm of computing the minimal $F(s)$, they first introduce a data structures to represent arcs and boundaries. A boundary consists of a sequence of arcs. If we use the location of $s$ as the reference point, any arc in the neighbor’s boundary of $s$ can be uniquely defined by a 3-tuple $(\theta', o, \theta')$, where $\theta'$, $o$,
and \( \theta \) are the starting angle, the centre and the ending angle of the arc, respectively. \( \theta' \) and \( \theta'' \) are relative to the horizontal line going through \( s \) counting in anti-clock direction. For example in Figure 4, line \( xs \) is the horizontal line from \( s \), which is used as the reference line in counting starting and ending angles of arcs. Arc \( cb \) of disk \( u \) is represented by \( (\angle xsb, v, \angle xsc) \). where \( \angle xsb \) is the starting angle and \( \angle xsc \) the ending angle of \( cb \). According to this data structures, the neighbor’s boundary of \( s \) in Figure 4 can be represented as \( B [] = \{ xa, ab, bc, cd, dx \} = \{ (0^0, u, \angle xsa), (\angle xsa, s, \angle xsb), (\angle xsb, v, \angle xsc), (\angle xsc, s, \angle xsd), (\angle xsd, u, 360^0) \} \)

Using this data structure to represent arcs and boundaries, Hai Liu et al. [25] presents an algorithm FwdNodes to compute the minimal \( F(s) \) with time complexity \( O(n \log n) \).

The strategy of this method is to compute the Neighbor’s boundary of \( s \), and thus the nodes that contribute to this boundary are the nodes in \( F(s) \). It uses the pair-wise boundary merging method [25] to compute the boundary efficiently. Initially, each node is arbitrarily paired with another node to merge their coverage boundaries. Then, the merged pair’s boundary is further merged with another pair’s boundary. This merge operation is repeated until eventually there is only one big merged boundary, which is the neighbor’s boundary of \( s \). The minimal \( F(s) \) consists of the nodes that contribute to this boundary.

Furthermore, it is also proved that FwdNodes algorithm achieves local optimality in terms of: 1) the number of forwarding nodes is the minimum, i.e. \( F(s) = F_{\text{min}}(s) \); 2) the time complexity is the lowest.

After computing \( F(s) \) by FwdNodes algorithm, the node \( s \) attaches IDs of nodes in \( F(s) \) to the flooding message and broadcasts it out. When receiving this message, a neighbor node of \( s \) checks whether its own ID is in the forwarding list attached to the message. If yes, it will call the FwdNodes algorithm and forward the message, the same as \( s \) does. In this way, the message is forwarded hop by hop until all the nodes receive it.

2) Forwarding node optimization

The \( F(s) \) computed above is only locally optimal based on the 1-hop information of \( s \). When a node \( u \) receives the flooding message from \( s \) (we call \( s \) the parent of \( u \)) and
u is a forwarding node nominated by s (i.e., \( u \in F(s) \)), the computing of \( F(u) \) can be further optimized based on the information of \( F(s) \), which is attached to the flooding message from s. This is because some nodes in \( F(u) \) may be already covered by node s or node-set \( F(s) \), and thus \( F(u) \) could be further reduced by removing those nodes.

![Figure 5 An example of optimizing F(u).](image)

For example, Figure 5 shows a network where nodes u and v are neighbors of s and \( F(s) = \{u, v\} \). The coverage area \( d(u) \) overlaps with \( d(s) \) and \( d(v) \) (notice that node v is also in \( F(s) \)). The nodes in the overlapped area of \( d(u) \) with \( d(s) \) were already considered by s when computing \( F(s) \). Thus, these nodes can be removed from \( F(u) \). For the overlapped area of \( d(u) \) with other nodes in \( F(s) \), for example node v in Figure 5, we use node ID as the priority for forwarding messages. That is, the node with the smaller ID has to forward the message if its coverage disk overlaps with another node. Suppose \( id(v) \leq id(u) \) and \( F(u) = \{a, b, c, e\} \). Since nodes a and b are in \( N(s) \), they are already covered by s and can be removed from \( F(u) \). Node c is covered by v, and v is also a forwarding node and \( id(v) \leq id(u) \). Thus, node c can also be removed from \( F(u) \). Finally, \( F(u) = \{e\} \). That is, node u only needs to nominate the nodes of \( F(u) \) in the clear area.

This improved algorithm is called OptFwdNodes. It guarantees that all nodes can receive flooding message with complexity of \( O(n \log n) \), where \( n = |N(s)| \).

3) Mobility handling,

In MANETs, nodes may be mobile, which causes dynamic changes of the network topology. For the flooding scheme, each node, say s, maintains its neighbor information and computes \( F(s) \). To cope with the dynamic topology changes, there are two strategies to maintain the flooding scheme: a) No update. Each node re-computes its forwarding node set for each flooding request; or b) Incremental update. Each node incrementally updates its forwarding node set upon each topology change. For strategy (a), we do not need to do anything; for (b), an efficient algorithm that can incrementally update the forwarding node set as the topology changes is proposed [25]. By using this method, nodes do not need to re-compute the forwarding node set when it needs to flood a message and it is very efficient compared to re-computing.
EF1 is compared with Pure flooding, Edge Forwarding, and CDS-based flooding against two parameters: number of nodes and transmission range. By running simulations under the ns-2 test bed with the CMU wireless extension, results show that the performance of EF1 is significantly better than the performance of Edge Forwarding and CDS-based schemes because it uses less forwarding nodes, incurs less collision and obtains high deliverability ratio.

### 3.5 Comprehensive Efficient Flooding (CEF)

In the simulation experiments [25], the EF1 is compared with three deliverability guaranteed schemes: Pure flooding, Edge forwarding and CDS-based flooding. We can see that the number of collisions is not satisfied. Therefore, Xianlong Jiao et al. in [15] proposed a new flooding scheme called CEF to further optimize this performance by considering utilizing the directional antenna. They first assume that every node is provided with a single-beam directional antenna in EFDA [14], and then propose CEF which is an algorithm improved upon EFDA, and aims to achieve better performance compared with EF1 and EFDA in the number of collisions, delivery rate and so on.

Directional antennas have been exploited in many papers for optimizing the performance recently. In [35], Akis spyropoulos et al. demonstrated that using directional antennas in ad hoc networks is beneficial for optimizing some performance in energy-efficient routing and scheduling etc. In CEF, the authors use directional antennas for optimizing the signal collisions. There are three parameters for a directional antenna: beam-radius, beam-width, and beam-orientation. Each node can adjust the beam-width and beam-orientation of the directional antenna, and the identical beam radius is fixed as the transmission range of the node. The coverage area of the transmission signal can be different by adjusting beam-width or beam orientation. For example, in Figure 6 (Ⅰ), node S can cover all the nodes when the beam-width is large enough. In Figure 6 (Ⅱ), S decreases the beam-width, and only covers b, c, and d. The coverage area will be similar if S changes the beam orientation, as in Figure 6 (Ⅲ).

![Figure 6 Example of beam-widths and beam orientations](image)

Figure 6 Example of beam-widths and beam orientations
CEF improves upon EF1 in that it modified optimized forwarding node selection algorithm in EF1, and held boundary merge algorithm and forwarding node selection algorithm which is used in EF1 unchanged.

In simulations, the authors in [15] compare CEF to Pure Flooding, EF1, and EFDA with the number of collisions and deliverability ratio. The performance of Pure Flooding significantly decreases with the number of nodes. EF1, EFDA, and CEF maintain good performance, while CEF achieves the best. When transmission range increases, the more signal collision will come forth in Pure Flooding. This change does not affect the number of collisions for EF1, EFDA, and CEF much, where CEF performs the best again. The deliverability ratio increases with the transmission range’s increase in CEF, EF1. CEF maintains the best performance due to the smallest number of collisions while EF1’s deliverability falls quickly after the pause time increases. These results show that signal collisions can be reduced by using the directional antenna together with neighbor information.

3.6 FONIAH

There is also a hybrid flooding approach which combines Neighbor knowledge based flooding and Area based flooding called FONIAH (Flooding based on One-hop Neighbor Information and Adaptive Holding) [21]. It utilizes knowledge of 1-hop neighbors and the concept of “adaptive holding time”. The “holding time” means “Packet holding time”, where a node receiving a packet is allowed to hold it for some amount of time before rebroadcasting [1]. This holding time increases as the distance from the sender to the receiver decreases, this concept may cause a high latency problem in a sparse network although it can be useful to rapidly enlarge a coverage area. The authors claim that FONIAH provides better efficiency without harming delivery compared to existing schemes of either purely Neighbor knowledge based or Area based approaches. However, it does not take into account minimizing the number of forwarding nodes in each step.

The scheme assumes each node knows geographical locations of all neighbors within one hop transmission radius. First, the broadcasting initiator node calculates the Maximum Distance ($d_{\text{max}}$)—distance from itself to the farthest 1-hop neighbor, and utilizes it to calculate a packet holding time at the receiver node. The value of $d_{\text{max}}$ is embedded in the broadcast packet with the location of the sender node and updated by the successive next hop nodes. A current holding time is inversely proportional to the distance between the sender and receiver node:

$$HT_{\text{cur}} = (1 - \frac{d_{\text{cur}}}{d_{\text{max}}}) \cdot HT_{\text{max}}$$
Here, $HT_{\text{max}}$ is the predetermined, constant maximum holding time, $d_{\text{cur}}$ is the distance from the previous hop node to the current node. In the term *Adaptive Holding Time*, "adaptive" means that the value of the maximum distance in this scheme is not fixed but is adjusted adaptively by the previous hop node via the packet.

When a node receives another copy of broadcasting packet when waiting for the holding time to expire, it estimates which of its neighboring nodes have also received the packet. A receiver node compares the distance from the packet sender to each of its neighbors with $d_{\text{max}}$ embedded in the packet. If it is not farther than $d_{\text{max}}$, then this neighbor is supposed to have received that packet. Before a holding time expires, if all neighbors have already received the packet, the deferred packet is dropped. Otherwise, the packet is rebroadcast.

The simulation results show that the total packet overhead (total size of all transmitted data and control packets) and the average latency in FONIAH are both smaller than Pure Flooding and GeoFlood (one of the area--based flooding scheme proposed in [1]).

### 4. 2-Hop or More Neighbor Knowledge Methods

Schemes in this category assume that each node keeps the information of 2-hop neighbors, i.e. the neighbors of the 1-hop neighbors. The 1-hop neighbor information can be obtained by exchanging the HELLO message in MAC layer protocols that are sent locally by each node to declare its presence. In a mobile environment, these messages are sent periodically to refresh the information.

To obtain the information of 2-hop neighbors, one solution may be that each node attaches the list of its own neighbors, while sending its HELLO messages. With this information, each node can independently calculate its 1-hop and 2-hop neighbor set. Once a node has its 1- and 2-hop neighbor sets, it can select a minimum number of 1-hop neighbors which covers all its 2-hop neighbors.

#### 4.1 Selecting the Multipoint Relaying (MPRs)

In this category, since each node knows the network topology within a 2-hop radius, we can try to select a minimum number of 1-hop neighbors whose coverage areas cover all its 2-hop neighbors. Unfortunately, finding such a set with minimum size is NP-hard. Therefore, Amir Qayyum et al. [30] proposes a heuristic algorithm to compute the so called Multipoint Relays (MPR) as follow:

1. Find all 2-hop neighbors that can only be reached by one 1-hop neighbor. Assign those 1-hop neighbors as MPRs.
2. Determine the resultant cover set (i.e., the set of 2-hop neighbors that will receive the packet from the current MPR set).

3. From the remaining 1-hop neighbors not yet in the MPR set, find the one that would cover the most 2-hop neighbors not in the cover set.

4. Repeat from step 2 until all 2-hop neighbors are covered.

Lim and Kim [22] proposed another heuristic algorithm named Dominant Pruning to compute the MPR sets.

Dominant pruning is based on greedy set cover [18]. It iteratively chooses 1-hop neighbors which cover the most uncovered 2-hop neighbors until all 2-hop neighbors are covered.

Figure 7 Dominant pruning

Suppose node $v_j$ receives a packet from $v_i$ and $v_j$ is in the forward list, as Figure 7 shows, node $v_j$ should determine its own forward list so that all nodes within 2-hop distance from $v_j$ receive the packet. The forward list should be minimized to decrease the number of transmissions. Among nodes in $N(N(v_j))$, the nodes $v_i$, $v_j$, and $N(v_j)$ have already received the packet, and $N(v_j)$ will receive the packet when $v_j$ forwards the packet. Therefore a node $v_j$ determines its forward list so that all nodes in $U = N(N(v_j)) - N(v_i) - N(v_j)$ receive the packet. Let $B(v_i, v_j) = N(v_j) - N(v_j)$. Then we select a set of nodes $F = \{f_1, f_2, \cdots, f_m\} \subseteq B(v_i, v_j)$ such that $\bigcup_{f \in F} (N(f) \cap U) = U$.

Since finding a minimum $F$ is the set cover problem which is NP-complete [7], it use greedy set cover algorithm to find $F$ as follows:

1. Let $F = \emptyset$, $K = \{S_1, S_2, \cdots, S_n\}$ where $S_k = N(v_k) \cap U (1 \leq k \leq n)$ and $Z = \emptyset$. 
2. Find the set $S_k$ whose size is the maximum in a set $K$.
3. $F = F \cup \{v_k\}$, $Z = Z \cup S_k$, $K = K - \{S_k\}$, $S_i = S_i - S_k$ for all $S_i \in K$.
4. If $Z = U$, complete the algorithm.
5. Otherwise, repeat from 2 again.

This algorithm repeats selecting $v_k$ of which the number of neighbor nodes that are not covered yet is maximum. It has been proved that this approximation algorithm has the approximation ratio of $(\ln |U| + 1)$ [18]. Furthermore, T.H. Cormen et al in [5] proved that the algorithm can be implemented to run in $O(|U||K| \min(|U|,|K|))$ time.

The experiments in [22] compare the self pruning method with dominant pruning. Because dominant pruning uses 2-hop neighborhood information while self pruning uses 1-hop neighbor information only, the performance gain of dominant pruning is greater than that of self pruning. On the other hand, dominant pruning has larger overhead than self pruning and the overhead increases as the node mobility increases. Therefore, dominant pruning is more appropriate in a network with moderate size and little mobility.

4.2 Using Connected Dominating Set (CDS)

Another important flooding technique is the use of connected dominating sets [43], [6], [37]. A dominating set (DS) is a subset where that all the nodes in the system are either in the set or has at least one neighbor in the set. A CDS is then defined to be a connected DS. Routing in MANETs can be done efficiently via CDS.

The main advantage of routing based on connected dominating set is that it centralizes the whole network into the small connected dominating set sub network, which means only nodes in CDS keep routing information, so that as long as network topological changes do not affect this sub network, there is no need to recalculate routing tables.

B. Clack et al. [2] proves that finding a minimum connected dominating set (MCDS) is NP-complete even in the unit disk graph. Then, the approximation or heuristic algorithm which can determine (connected) dominating set in a given (connected) graph becomes the main and important research work [8]. There are some distributed algorithms for computing MCDS with guaranteed approximation ratios proposed in [6], [40]. Furthermore, the authors in [43] also discuss ways to update and recalculate the dominating set when the underlying graph changes with the movement of mobile nodes and describe some efficient routing using CDS.

In [28], the author present two algorithms for computing CDS, which require $O(\Delta \log^2 n)$ and $O(\log^2 n)$ running time respectively, where $\Delta$ is the maximum node degree and $n$ is the size of the network.
B. Williams and T. Camp [41] points out that it is impossible for any algorithm to perform better than the MCDS. However, maintaining a CDS in the network is costly; it is not suitable for flooding operations in highly mobile situations.

We know that the MCDS problem in UDG is shown to be NP-hard [2]. Therefore, Wan et al. [40] proposed the first distributed algorithm with the performance ratio of 8. Then, Li et al. [23] proposed a better algorithm with the performance ratio of (4.8+ \ln 5).

However, the transmission ranges of all nodes are not necessary equal in practice. My T. Thai et al. [39] presents three constant approximation algorithms to obtain a MCDS in DGB (Disk Graphs with only Bidirectional links) networks where nodes have different transmission ranges.

### 4.3 Three-hop Horizon Pruning (THP)

There are few flooding or broadcast algorithms where the nodes keep more than 2-hop neighbor information. The three-hop horizon pruning (THP) algorithm is the first heuristic to take into account 3-hop information in the selection of relay nodes for broadcast packets [33]. THP is also the first neighbor-designated algorithm for computing TCDS—2-hop connected dominating set, which is a set of nodes such that every node in the network is within 2 hops from some node in the dominating set. In this algorithm, maintaining fresh routes to all nodes within 2 hops is possible for every node that has the 2-hop neighborhood information. In [33], it is shown that THP outperforms the existing best-performing self-pruning and neighbor-designated algorithms.

### 5. Conclusion and future work

In the report we present an overview of current works on efficient flooding scheme in mobile ad hoc networks. We classify the existing flooding methods into three categories according to the information each node keeps. We discuss various algorithms, protocols and techniques that guarantee the schemes with deliverability of flooding messages and alleviation of the broadcast storm problem. In fact, it is impossible for any algorithm to perform better than the MCDS and unlikely to perform worse than pure flooding. The algorithms are highly dependent on the density of the network. In sparse networks, the protocols are expected to perform similar to pure flooding, as each node may have to rebroadcast to reach isolated neighbors. As density increases, proportionally fewer nodes should rebroadcast.

In following table, representative flooding schemes in each category are illustrated and compared with their basic ideas, time complexity, advantage, and disadvantage, etc.
Table 1. Comparison of flooding algorithms.

<table>
<thead>
<tr>
<th>Neighbor Information</th>
<th>Classification</th>
<th>Basic Idea</th>
<th>Time Complexity</th>
<th>Advantage</th>
<th>Disadvantage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pure Flooding</td>
<td>Receiver-based</td>
<td>Each node transmits the flooding message once.</td>
<td>$O(N)$, $N$ is the number of nodes in the network.</td>
<td>Simple; Low complexity</td>
<td>Broadcast storm</td>
</tr>
<tr>
<td>Probabilistic Flooding</td>
<td>Receiver-based</td>
<td>Nodes forward a flooding message with probability $P$.</td>
<td>$O(N)$, $N$ is the number of nodes in the network.</td>
<td>Alleviate the broadcast storm problem compared Pure Flooding.</td>
<td>Some nodes may miss the flooding messages and difficult to set the right value of $P$ in various network situations.</td>
</tr>
<tr>
<td>FSP</td>
<td>Receiver-based</td>
<td>A receiver forwards the message if its 1-hop neighbors are not fully covered by the sender.</td>
<td>$O(\triangle)$, $\triangle$ is the maximum node degree.</td>
<td>Low complexity; reduce flooding cost.</td>
<td>Improvement is limited in most networks.</td>
</tr>
</tbody>
</table>
Transmission coverage is partitioned into six sectors.

**Edge Forwarding**
- Receiver-based forwarding decisions are based on the availability of other forwarding nodes in the overlapping areas.
- Scalable and suitable for dense networks.
- Can not avoid redundant transmissions.

**Vertex Forwarding**
- Sender-based forwarding decisions are based on the availability of vertices in the grids to forward messages.
- Simple and efficient.
- Some nodes may miss the flooding messages.

$$O(N), N$$ is the number of nodes.

**EF1**
- The number of forwarding nodes is $$O(N \log N)$$.
- Select the minimal forwarding set by computing the boundary of each 1-hop neighbors area.
- The number of forwarding nodes is locally minimized and the time complexity is the lowest.

$$O(N \log N), N$$ is the number of 1-hop neighbors.

**1-hop information**
- No theoretical guarantee on global performance.
<table>
<thead>
<tr>
<th>Method</th>
<th>Node Information</th>
<th>Knowledge Type</th>
<th>Complexity</th>
<th>Theoretical Basis</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>FONIAH</td>
<td>More than 1-hop</td>
<td>Receiver-based</td>
<td>N/A</td>
<td>Low packet overhead, reasonable latency, and high reliability compared with Pure Flooding and GeoFlood.</td>
<td>Every node needs geographical locations of all its 1-hop neighbors.</td>
</tr>
<tr>
<td>Selecting the Multipoint Relaying (heuristic in [30])</td>
<td>2-hop information</td>
<td>Sender-based</td>
<td>$\log N$ (for each stop), $N$ is the number of 2-hop neighbors</td>
<td>Theoretically based on MPRs.</td>
<td>N/A</td>
</tr>
<tr>
<td>CDS Based</td>
<td>Receiver-based</td>
<td>Receiver-based</td>
<td>N/A</td>
<td>Theoretically based on MCDS.</td>
<td>Maintaining a CDS is costly and is not suitable for highly mobile situations.</td>
</tr>
</tbody>
</table>
Almost each of the existing flooding schemes relies on the assumption that the network can be represented by a unit disk graph (UDG) and all nodes have the same transmission ranges. One of the future works is to extend the access condition of these techniques. We can allow each mobile node in the network to have different transmission radii or it can change its radius based on surroundings, such as calculating MCDS in DGB discussed in [39], etc. Authors also notice that most of existing flooding schemes are either sender-based or receiver-based. Work in [25] showed the potential benefit which can be achieved by hybrid methods. Hence, more hybrid flooding schemes need to be studied in future. Moreover, none of existing works considers both forwarding node selection and transmission schedule. Improper schedule on transmissions of forwarding nodes may result in congestion and loss of packets. Once forwarding node set is determined, the problem of how to schedule those forwarding nodes to efficiently rebroadcast flooding messages requires further study.

References


