# Efficient Broadcasting in Homogeneous and Heterogeneous Wireless Ad Hoc Network

**Kihwan Kim, Ying Cai, and Wallapak Tavanapong**

Department of Computer Science
Iowa State University,
Ames, IA 50011-1040, USA

**Summary**
Network-wide broadcasting is a fundamental and frequently invoked communication primitive in wireless ad hoc networks where there are no pre-existing communication infrastructures. Existing broadcasting techniques perform well with respect to two out of the three performance goals (low broadcast latency, low retransmission overhead, and high broadcast reachability), but require each host to track its neighbors within at least 2-hop distance away. This paper introduces a new broadcasting scheme called *Adaptive Scheduling with Adaptive assessment Periods (ASAP)* and its variants that achieve the three performance goals simultaneously while requiring each host to track only its one-hop neighbors. In particular, the ASAP schemes offer low broadcast latency, making them suitable for time-constrained applications such as broadcasts of emergency messages and multimedia applications.
***Key words:***
*Ad hoc networks, flooding, broadcast, radio transmission range.*

## 1. Introduction

In recent years, wireless ad hoc networks have received a great deal of research interest due to simplicity and low cost of deployment. No pre-existing communication infrastructure is required. A node can communicate directly to nodes within its transmission range or to those outside its transmission range via wireless packet relays. Network-wide broadcasting, simply referred to as “broadcasting” hereafter, is an operation for sending a packet to all the other nodes in the network. In wireless ad hoc networks, broadcasting is necessary for service and resource discovery and is a building block for many unicast and multicast routing protocols (e.g., DSR[1], AODV[2], ZRP[3], LAR 4], just to name a few).

The simplest broadcasting technique is *Simple Flooding* [5, 6]. In this scheme, when a host receives a broadcast packet, it retransmits the packet if it has not previously seen the same packet.[1] Otherwise, the host drops the packet. Simple Flooding guarantees that a broadcast packet reaches all other hosts that are reachable from its originator if no packet collision occurs. However, this scheme generates a large amount of network traffic because it requires every host to retransmit the same broadcast packet once. In the case that all hosts are within 1-hop distance of each other, a broadcast packet travels each pair of hosts twice. Such overwhelming amount of packet retransmissions, most of them unnecessary, can quickly exhaust the hosts' battery power. If the hosts have to compete for limited communication bandwidth, the excessive network traffic can also cause significant delays in packet transmissions and may hang up the entire network because of severe packet contention and collision [2].

As a communication primitive, broadcasting has a significant impact on the overall performance of wireless ad hoc networks. Numerous broadcasting schemes have been proposed along with a number of metrics to evaluate their performance and implementation cost. For ease of exposition, we assume a static network and no packet collision, under which these techniques generally work best. To measure the performance of a broadcasting technique, a commonly used metric is *broadcast reachability* defined as the percentage of reachable hosts that actually receive the broadcast packet. Reachable hosts are hosts in the network that can receive the broadcast packet through Simple Flooding. A broadcast reachability below 100% means that some host does not receive the broadcast packet because an intermediate host between itself and the source decides not to retransmit the packet. In the case of Simple Flooding, its broadcast reachability is 100%. Another metric is *retransmission overhead* defined as the percentage of receiving hosts (i.e., hosts receiving the broadcast packet) that actually retransmit the packet. The retransmission overhead of Simple Flooding is 100% since all hosts must transmit the packet once during a broadcast. Low retransmission overhead is desirable to reduce unnecessary network traffic. The third metric is *broadcast latency* defined as the time taken by a broadcast

[1] Each broadcast packet can be uniquely identified, for instance, via a unique host ID of the originator of the packet and the unique packet ID among packets sent by the originating host.

Manuscript received July 5, 2008.
Manuscript revised July 20, 2008.

packet from a source to reach the last reachable host in the network. This metric has recently gained more attention since it is critical for time-constrained applications such as broadcasts of emergency messages and real-time multimedia applications. While these three metrics measure the performance of a broadcasting technique, another metric that measure its implementation cost is the number of hops that it requires each host to track its neighbors. In general, the more hops a host needs to track, the more *network control overhead* incur.

An ideal broadcasting technique should simultaneously achieve three performance goals, i.e., 100% reachability, low transmission overhead, and low broadcast latency, with low network control overhead. Existing techniques, however, fall short in achieving these goals. While some of them make a tradeoff between broadcast reachability and transmission overhead, many others incur long broadcast latency, or require excessive network control overhead, or both. Note that minimizing transmission overhead and minimizing broadcast latency are two seemly conflicting goals. To keep retransmission overhead low, many techniques require each host to keep track of redundant packets received over a period of time (termed *assessment period* hereafter). The value of this period is either randomly chosen between zero and some pre-configured value, or based on factors such as a host's distance to where it receives the packet for the first time. At the end of its assessment period, a host checks all redundant packets it receives and then determines whether to retransmit the broadcast packet. A longer assessment period allows a host to collect more redundant packets, giving it a better chance to avoid retransmission of the packet. However, it tends to increase the broadcast latency.

In this paper, we propose a novel broadcasting technique called *Adaptive Scheduling with Adaptive assessment Period (ASAP)* scheme. The crux of the proposed technique is the interplay of two features: *instant frontier forwarding* and *adaptive assessment period*. The former feature creates an initial schedule indicating the order that its neighboring hosts should forward the packet, based on their distance from the packet sender. The farthest neighbor will immediately forward the packet, while other closer neighbors wait for some assessment period set according to their order in the forwarding schedule. The instant forwarding of the farthest neighbor allows a broadcast packet to be propagated to the distant hosts *as soon as possible*, keeping broadcast latency low. The adaptive-assessment-period feature, on the other hand, allows a host to wait *as long as possible* to refrain from retransmission, minimizing retransmission overhead. Specifically, it lets a waiting host (closer neighbor) adjust its assessment period (i.e., adapt from the initial forwarding schedule) according to information in duplicate packets the host has seen so far. The excellent interplay of the two features enables our ASAP scheme to achieve the three performance goals, i.e., 100% reachability, low retransmission overhead and broadcast latency, simultaneously. As for its implementation cost, the new scheme requires each host to track only its neighbors within one hop. Thus, it incurs lower control overhead and is less sensitive to changing network topologies compared to many existing techniques that track neighbors within at least 2 or more hops away. In a dynamic network with host mobility, the performance of these schemes will downgrade much more significantly than those techniques tracking only 1-hop neighbors [12, 26].

The remainder of this paper is organized as follows. We summarize existing broadcasting techniques in Section 2, describe the concept and the details of the ASAP scheme in Section 3. In Section 4, we extend ASAP for heterogeneous networks, where hosts may have different transmission capability. In Section 5, we describe the simulation model and present the performance results. Finally, we provide our concluding remarks in Section 6.

## 2. Existing Broadcasting Schemes

### 2.1 Tables and Figures

Besides Simple Flooding, we classify existing broadcasting protocols into three broad categories and discuss their advantages and limitations in an ideal condition with no packet collisions as follows.

- *0-hop schemes*: These schemes do not require hosts to periodically track their neighbors. Many early broadcasting techniques belong to this category and can be further divided according to their dominant characteristics.
    - *Probabilistic-based schemes*: Each host retransmits a broadcast packet with some predetermined probability. The probabilistic-based scheme was first proposed in [7,8] and further investigated in [9]. Another scheme called the counter-based scheme[7] requires each host to count the number of redundant packets received during its assessment period. When the assessment period expires, the host retransmits the packet only if the number of redundant packets is less than a threshold value. Otherwise, the host drops the packet.
    - *Coverage-based schemes*: In these schemes, a host utilizes a coverage transmission area in its retransmission decision. Distances among hosts[7] or hosts' location (e.g., using a Global Position System) [7,10] were used in determining the coverage area. For the coverage-based scheme

using host distances, a host retransmits the packet after the expiration of its assessment period only if the host is sufficiently far (at least by a distance threshold) from the senders of the redundant packets. For a coverage-based scheme using hosts' location, a host retransmits the packet only if it covers sufficiently more area (by a pre-determined threshold). Another scheme utilizes angles computed using locations of the hosts sending the redundant packets to check the coverage area [10]. Note that the coverage-based schemes only consider the coverage area of the transmission regardless of whether there are hosts in the coverage area or not.

The $0$-hop schemes can effectively reduce retransmission overhead. Since they do not need to track any neighbors, they have similar implementation cost as Simple Flooding. Furthermore, their performance is not sensitive to host mobility and heterogeneous networks. However, retransmission overhead offered by these techniques is highly dependent on threshold values (e.g., retransmission probability for the probabilistic schemes, the distance threshold for a coverage-based scheme using host distances). Furthermore, the threshold values have a significant impact on broadcast reachability. Selecting appropriate thresholds to achieve both low retransmission overhead and high broadcast reachability is difficult in practice [10].

- *1-hop schemes*: Flooding with Self Pruning [11] is a 1-hop scheme since it requires each host to track neighboring hosts within its 1-hop distance via periodic broadcasts of beacons. A beacon contains the unique identity of the sending host. In this scheme, a host broadcasts a data packet that also includes all of its 1-hop neighbors in the packet header. Upon receiving a broadcast packet, a host retransmits the received broadcast packet except when all of its 1-hop neighbors have already been included in the broadcast packet header. In other words, no retransmission is needed when all of the host's neighbors are also the neighbors of the packet sender. No assessment periods are used in this scheme. Although this technique can both reduce the retransmission overhead and guarantee broadcast reachability, its performance improvement is very limited in most network conditions [12].

- *$2^+$-hop schemes*: Many recent broadcasting schemes are in this category. Neighbor information is exchanged through periodic broadcasts of beacons. For instance, a 2-hop scheme includes the host ID and IDs of this host's 1-hop neighbors in a beacon. The $2^+$-hop techniques can be further divided into *reactive* schemes and *proactive* schemes. In the proactive schemes [11] [13] [14] [15] [16] [17], a host selects some of its 1-hop neighbors as rebroadcasting hosts. When a host receives a broadcast packet, it drops the packet if it is not designated as a rebroadcasting host; otherwise, it recursively chooses some of its 1-hop neighbors as rebroadcasting hosts and then retransmits the broadcast packet. In the reactive schemes [18] [19] [20] [21] [22] [23] [24] [26], each host determines on its own whether or not to retransmit a broadcast packet. In general, with $2^+$-hop of neighborhood information, these techniques can offer very good performance in terms of broadcast reachability and retransmission overhead. However, the $2^+$-hop schemes are less attractive for dense networks with host mobility. This is because the overhead of discovering and maintaining a local network topology (within two or more hops) for each host may outweigh the reduction in the retransmission overhead [9, 10]. Furthermore, for the proactive schemes, selecting a suitable set of hosts to retransmit the broadcast packet is not trivial and requires significant computation on hosts. It was proven in [17] that finding an optimal set of rebroadcasting hosts is NP-hard.

## 3. Proposed Technique: ASAP

We first describe the assumptions used in this paper. Next, we discuss the two features of the ASAP scheme, i.e., *instant frontier forwarding* and *adaptive assessment period*. Finally, we present the algorithm of the scheme.

### 3.1 Preliminaries

We assume that broadcasting nodes assess a clear channel before broadcasting. Although an assessment of a clear channel cannot prevent packet collisions and the hidden node problem, it does not have the high overhead of the 802.11 MAC that utilizes the Request To Send/Clear To Send/Data/Acknowledgment procedure for unicasting packets. We assume that when a packet is given to the MAC layer, a very small random delay is added to prevent two neighboring hosts from simultaneously assessing a clear channel. This delay is different from the assessment period discussed earlier. Like other broadcasting schemes, we assume that hosts can uniquely identify a broadcast packet. These assumptions have been used in other previous work as well [12].

We assume that a host is able to determine its distance to a sender of a packet through radio signal strength [27]. We do not require each host to know its exact position. We later discuss the variant of the ASAP scheme if a host is equipped with an advance positioning technology such as Global Positioning Systems [28] and ultrasonic/infrared position sensing systems [29, 30, 31]. Each host tracks its 1-hop neighbors through periodic broadcasts of "Hello" messages or beacons. Each beacon contains only a host's ID. All hosts have the same transmission range.

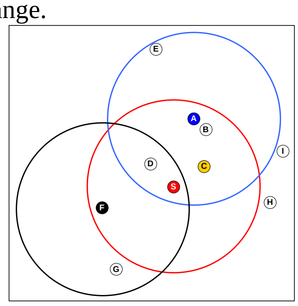



Fig. 1 Example ad hoc network.

Figure 1 shows an example ad hoc network of ten hosts with the same transmission range. Host *S* is the broadcast source. We only draw transmission coverages of three hosts, *S*, *A*, and *F* for ease of illustration. Most of 1-hop neighbors of each host can easily be seen from the figure except a few cases. That is, *H* is a 1-hop neighbor of *C* only. *A* is the only 1-hop neighbor of *E*, *F* is the only 1-hop neighbor of *G* and *I* is a 1-hop neighbor of *B* and *C*. To achieve the lowest broadcast latency, lowest retransmission overhead, and 100% broadcast reachability, only three hosts *A*, *C*, and *F* need to forward the packet immediately when they receive the broadcast packet for the first time. Note that the three hosts do not necessarily assess a clear channel simultaneously because of a small random delay introduced when a packet is passed to the MAC layer.

## 3.2 Instant Frontier Forwarding

In our scheme, a sending host creates a forwarding order based on its 1-hop neighbor knowledge and includes the forwarding order in the header of the broadcast packet. The forwarding order is a list of IDs of the neighboring hosts in the decreasing order of their distance from the sending host. In the example in Figure 1, when host *S* broadcasts its packet, it includes the forwarding order: *F*, *A*, *B*, *C* and *D* in the header of the packet. This forwarding order indicates that host *F* should forward the packet first followed by hosts *A*, *B*, *C*, and *D*, respectively since $F$ is the farthest neighbor and can cover the largest additional area among *S*'s neighbors. Note that with only 1-hop neighbor knowledge, host *S* cannot just choose some of its neighbors as forwarding hosts since it does not have the neighbor information of its neighbors. Furthermore, host *S* cannot determine the overlap in the transmission coverage areas of its neighboring hosts since host *S* does not know the exact locations of its neighbors.

In our scheme, the host receiving the broadcast packet for the first time (say host *i*) checks whether it has any neighbor that is not the neighbor of the packet sender. If all of its neighbors are also the sender's neighbors, the host does not need to retransmit the packet. Otherwise, the host checks whether it is the farthest neighbor (from the packet sender) compared to all of its neighbors that are also the sender's neighbor. If so, the host forwards the packet immediately; otherwise, it sets its assessment period *W* according to Equation 1:

$$W = Max\_Period \times \frac{R_s - dist(s,i)}{dist(s,i)} \quad (1)$$

where $R_s$ is the transmission radius of the sending host *s*, and *dist(s, i)* is the distance between the sending host and host *i*. Note that $dist(s, i) \leq R_S$ since host *i* must be in the transmission range of host *s* to be host *s*'s neighbor.

Given the example in Figure 1, using our scheme, hosts *A* and *F* retransmit the packet immediately when they receive the broadcast packet. Although host *A* is not the farthest neighbor of *S* (not the first host in the forwarding order), it is the farthest compared to the neighbors common to both *S* and *A* (i.e., *A*, *B*, *C*, and *D*). Host *F* also forwards the packet immediately for the same reason. In our scheme, host *C* still has to wait with its assessment period set to $W = Max\_Period \times \frac{R_S - dist(S,C)}{dist(S,C)}$.
The next feature of our scheme reduces this wait time.

## 3.3 Adaptive Assessment Period

Existing broadcasting techniques let each host set its assessment period for each broadcast packet only once when the host first receives a broadcast packet. This contributes to unnecessary increases in broadcast latency and retransmission overhead as illustrated in the following example. Figure 1 shows that hosts A and B are very close to each other and most of B's neighbors except I are A's neighbors. Once A retransmits the packet, B will soon retransmit the packet according to the forwarding order from S. This retransmission performed by B is unnecessary since most of B's neighbors are already covered by A and retransmission of C will cover both H and I (one of B's neighbor as well). Hosts C and D need to wait for some time. Although both C and D are at about the same distance from S, the waiting of D is good since it is not necessary for D to retransmit the packet. However, the waiting of C lengthens the broadcast latency since C is the only host that can deliver the packet to H.

A better approach will be for a waiting host to adapt the original forwarding order on its own. Recall that using our instant-frontier-forwarding feature, a waiting host, say host $i$, initially sets its assessment period to $W = Max\_Period \times \frac{R_S - dist(i,s)}{dist(i,s)}$ once it receives the broadcast packet for the first time from host $s$. Our adaptive-assessment-period feature lets each host adapt its assessment period as follows. If at time $T$ where $T \leq W$ time units later after the timer of the assessment period has started, host $i$ receives a redundant packet from host $j$, host $i$ adjusts its assessment period according to Equation 2.

$$W = Max\_Period \times \frac{R_j - dist(i,j)}{dist(i,j)} - T,$$

where $R_j$ is the transmission radius of host $i$. That is, host $i$ adjusts its waiting time based on the distance between itself and the sender of the latest redundant packet. For the example in Figure 1, after receiving a redundant broadcast packet from *A*, our scheme will make *B* wait longer because *B* is closer to *A* than *C*. *C* waits less since it is farther from *A* than from *S* and can quickly retransmit the packet to reach both *H* and *I*. Note that setting assessment periods to create a forwarding order that lets the farthest neighbor wait the least was first explored in [10], but their approach sets a host's assessment delay only once, i.e., at the time when the host first receives the packet. In their approach, the farthest neighbor still needs to wait for some small time period instead of making its forwarding decision immediately like in the ASAP scheme. Since the assessment period is set only once, selecting an appropriate value of the assessment period has a significant impact on their performance. In our approach, the assessment period is dynamically changed. Hence, imperfect setting of the initial assessment period due to some errors in distance estimation in practice does not have a severe impact on our scheme. It will be shown in our performance study shortly that such static setting of assessment periods retains a large amount of unnecessary retransmissions. Especially, when network density is not very high, its performance is almost the same as Simple Flooding.

### 3.4 Algorithm of ASAP

We describe the algorithm a host uses when receiving a broadcast packet in our scheme. Recall that when a host sends/retransmits a broadcast packet, it includes its 1-hop neighbors in the packet header. Each unique broadcast packet is associated with a life thread created when a host receives a broadcast packet for the first time. For this packet and all its duplicates received later, this thread executes the pseudo-code in Figure 2.

When the thread forwards the packet and terminates itself, it also destroys the timer associated with the packet. On the other hand, if the packet timer expires, the host forwards the packet if some neighbor has not been marked. The host then terminates the corresponding thread. This case happens when the host cannot collect enough duplicate packets during the assessment period. We note that in Step 3.1 of the algorithm in Figure 2, we exclude those neighbors that have seen this broadcast packet before and may have already forwarded the packet. By checking only the newly-marked neighbors, we can ensure that after a broadcast is sent, at least one more broadcast will follow immediately if there exists a neighbor that has not been covered. Thus, broadcast latency can be minimized.

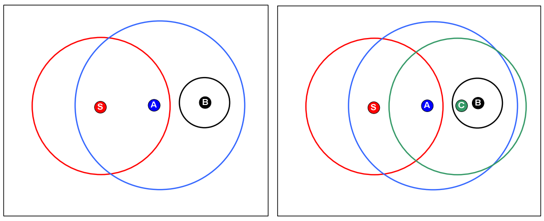
1. Check its 1-hop neighbors one by one and mark its neighbors that are also listed in the header of the received packet;
2. If all of the current host's neighbors have been marked, this thread terminates;
3. Otherwise, perform the following:
   - 3.1. For each of the newly-marked neighbors, calculate its distance from the packet sender;
   - 3.2. If the distance of the current host from the packet sender is the greatest among the computed distances, forward the packet and the thread terminates itself;
   - 3.3. Otherwise, set the assessment period as follows.
     - 3.3.1. If this packet is received for the first time, set the initial assessment period using Equation 1.
     - 3.3.2. Otherwise, adjust the assessment period using Equation 2.

Fig. 2 Algorithm invoked when a packet is received.

We can easily see that our ASAP scheme is simple to implement while it can achieve low broadcast latency, low retransmission overhead, and 100% broadcast reachability simultaneously in homogeneous and static networks. Another major advantage is low management overhead. ASAP requires each host to track only the hosts within its 1-hop distance. Most existing flooding techniques (e.g., Dominant Pruning [11], Multipoint Relaying [17], Scalable
Broadcast Algorithm [19], just to name a few) require each host to track its neighbors within 2-hop distance. Obviously, these schemes incur higher computation overhead and especially, significantly more control-related network traffic existing broadcasting techniques let each host set its assessment period for each broadcast packet only once when the host first receives a broadcast

## 4. Extending ASAP for Heterogeneous Networks

Existing broadcasting techniques were primarily proposed for homogeneous ad hoc networks, assuming that all hosts have the same transmission capability. In reality, however, different hosts may have different transmission coverage because of different wireless devices and/or varying rates of battery consumption. Such heterogeneity presents a new challenge for designing broadcasting techniques. Since a host may be unaware of another host with a smaller transmission range inside its transmission area, it could mistakenly drop the packet. In this section, we extend the ASAP scheme to handle host heterogeneity. We first present the effect of heterogeneity on broadcast reachability for some existing broadcasting techniques. Then, we describe the extended ASAP called ASAP-EXT to offer good performance under heterogeneous networks.




(a) Heterogeneous network I (b) Heterogeneous network II
Fig. 4 Host heterogeneity and broadcast reachability.

Figure 4(a) shows a network with three hosts *S*, *A*, and *B*, each having a different transmission range. Assume that *S* is the broadcast source. Using Simple Flooding, *B* with the smallest transmission range is able to receive the broadcast packet from S through *A*. Hence, Simple Flooding has 100% broadcast reachability. However, broadcasting schemes that employ neighbor knowledge (1-hop or $2^+$-hop schemes) in their retransmission decision have broadcast reachability below 100% because *A* does not know that *B* is within its transmission range. This results from the fact that B's beacons do not reach *A*. However, if the network is sufficiently dense like in Figure 4(b), B is able to receive the broadcast packet through C with the broadcasting schemes using neighbor knowledge.

---

/* To ensure that each quadrant is sufficiently dense */
1. Compute host density for each quadrant of the host's transmission area
2. If all of the density are greater than the density threshold, use ASAP-POS;
3. Otherwise, perform the following.
   3.1 Check the combined area covered by all its 1-hop neighbors
   3.2 If the combined coverage covers the entire transmission area of this host, drop the packet
   3.3 Otherwise, forward the packet

---

Fig. 5 ASAP-EXT.

We propose ASAP-EXT to handle heterogeneity. The main idea is for each host to first check its host density based on its 1-hop neighbors and transmission area. If the computed density is at least a density threshold, the host employs the ASAP scheme or its variant (i.e., using the distance-based forwarding order and adaptive assessment periods). Otherwise, the host uses some broadcasting scheme that does not rely on neighbor knowledge such as Simple Flooding or a coverage-based scheme using hosts' locations. ASAP-EXT can maintain high broadcast reachability as well as low retransmission overhead. Figure 5 illustrates one implementation of ASAP-EXT that checks the coverage area of neighboring hosts using their exact location, which requires each host to have the ability to obtain its exact location. Hosts exchange their location information and transmission range in the beacons. ASAP-POS is a suitable variant for this implementation.

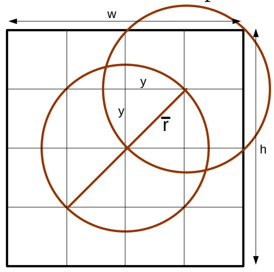



Fig. 6 Density threshold estimation.

To determine the appropriate density threshold, we estimate the minimum number of hosts required for a network in a rectangle of size $w \times h$ *meter*$^2$ to be connected. Assume a uniform distribution of hosts with different transmission radii. Let $\bar{r}$ be the expected transmission radius of all hosts. The density threshold (DT) is computed using Equation 4.

$$DT = 2 \times \frac{\left\lceil \frac{w}{y} \right\rceil \times \left\lceil \frac{h}{y} \right\rceil}{w \times h} \approx \frac{2}{y^2}, \quad (4)$$

Where $y = \bar{r}/\sqrt{2}$. Equation 4 is derived by overlaying a number of grids of size $y \times y$ *meter*$^2$ on a given network area. At each grid point, we place one host as illustrated in Figure 6. The minimum number of grids that can be placed in a transmission area of a host is 4. Based on a triangle geometry, we have $y = \bar{r}/\sqrt{2}$. The total number of grids is $\lceil w/y \rceil \times \lceil h/y \rceil$. We allocate the number of hosts twice the number of grids to account for the case that hosts may not be placed exactly on the grids.

## 5. Performance Study

In this section, we describe our simulators, simulation environments, and results under homogeneous and heterogeneous networks. These networks are static since host mobility is not the subject of our investigation in this paper.

### 5.1 Simulation Model

For homogeneous networks, we implemented detailed simulators for ASAP, Simple Flooding, Flooding with Self-Pruning [11] denoted as Self Pruning for short, and DBDT-ABS, which is a hybrid scheme that combines the two mechanisms proposed in [10]. The first mechanism called Distance-Based Defer Time (DBDT) requires a host to set its assessment period only once when it receives the broadcast packet for the first time. When the assessment period expires, the host employs the second mechanism called Angle-Based Scheme (ABS) using angles of locations of all the hosts sending the redundant packets to check whether the combined coverage covers this host's entire transmission area. If so, the host drops the packet; otherwise, it forwards the packet. The rationale for choosing these techniques are as follows. Simple Flooding is used as the upper bound to determine the savings in retransmission overhead provided by the other three techniques. We compare retransmission overhead of ASAP against that of Self Pruning since both require the same amount of neighbor knowledge but utilize them differently. We compare broadcast latency of ASAP against that of DBDTABS since both use assessment periods whereas the other two schemes do not. Since all the selected schemes can guarantee 100% reachability for homogeneous networks, we present only the results on retransmission overhead and broadcast latency.

For studies on heterogeneous networks, we simulated Self Pruning, ASAP, ASAP-EXT, and DBDT-AREA. DBDT-AREA is a hybrid scheme that uses the combined coverage area of hosts sending duplicate packets in their retransmission decision as ASAP-EXT does when host density is low. We use DBDT-AREA instead of DBDT-ABS since the angle-based mechanism as indicated in [10] is not applicable for heterogeneous networks. In addition to broadcast latency and retransmission overhead, we also evaluate broadcast reachability, since some broadcasting schemes may let hosts drop packets because they are not aware of neighboring hosts with a smaller transmission range.

All the simulators use a Null MAC. That is, when a host wishes to transmit a packet, it can acquire a clear communication channel and transmit the packet right away. Using a Null MAC provides a good theoretical view of the core of the broadcasting schemes. However, delays due to channel contention and packet collisions, which affect broadcast latency, cannot be measured. Nevertheless, our simulators are sufficient for our study since we are interested in relative performance rather than absolute performance of the simulated techniques. Note that using the Null MAC makes broadcast latency of Simple Flooding and Self Pruning much lower than that expected in practice. This is because no assessment periods are used in these schemes and delays due to channel competitions among a large amount of retransmitted packets are not accounted for. Due to this large discrepancy, we do not report the broadcast latency of both Simple Flooding and Self Pruning in this study.

To implement the schemes that use assessment periods, we keep a packet at the network layer until the corresponding assessment period expires. After which, the packet is either sent to the MAC layer or dropped. Such implementation of assessment periods and the Null MAC have also been employed in a recent study of performance comparisons among many broadcasting protocols [12]. To fairly study the relative performance among the schemes using assessment periods (i.e., ASAP, ASAP-EXT, DBDT-ABS, and DBDT-AREA) in terms of broadcast latency, we use Equation 1 to compute the initial assessment period and set Max Period to the same default value in all of these schemes. Note that the impact of the Null MAC on broadcast latency for these schemes is lower than that for Simple Flooding and Self Pruning because they have much less retransmission overhead as will be seen shortly. Hence, there are less network traffic and packet collisions. Our simulators compute the broadcast latency as the simulation time elapsed since the beginning of the broadcast until the last reachable host receives the packet.

For each simulation run, we generated a number of hosts and placed them randomly on a square domain. Because hosts stayed at the same location for the entire simulation run, we simulated beacon exchanges only once in the beginning of the run for Self-Pruning, ASAP, and ASAP-EXT to reduce the time to run experiments. Since the overhead of beacon exchanges and the 1-hop neighbor information in the packet header of the broadcast packet required in these techniques are about the same, we omit the overhead results in this paper.

Table 1: Simulation parameters.

| Parameter | default | variation | unit |
|---|---|---|---|
| Host density | 0.05 | 0.004 - 0.1 | *host/meter*$^2$ |
| Network area | 500x500 | 100x100 - 10,000x10,000 | *meter*$^2$ |
| Transmission radius | 10 | 5 - 15 meter | |
| Max Period | 10 | N/A | N/A |
| Number of runs | 30 | N/A | N/A |
| Confidence level | >= 95% | N/A | N/A |

In each simulation run, we chose the host in the center of the square domain as the broadcast source to ensure that broadcast latency is measured from the same originating location in all the simulations. The broadcast source broadcast one data packet using different broadcasting techniques. Only one broadcast occurred at any one time. For each broadcast, we recorded the performance and computed the corresponding average values. Each data point in our plots has a minimum confidence level of 95% and is the average of the results from 30 simulation runs generated as discussed above. Table 1 summarizes the parameter values used in the simulations. Roughly, we simulated a campus-size network, a domain region about 10 to 100 hops. In Section 5.2, we present the effects of host density, transmission radius, and network area on Simple Flooding, Self-Pruning, DBDT-ABS, and ASAP under homogeneous networks. In Section 5.3, we present the performance of Self-Pruning, DBDT-AREA, ASAP, and ASAPEXT under heterogeneous networks.

## 5.2 Results under Homogeneous Networks

### 5.2.1 Effect of Host Density

In this study, we fixed the transmission radius and the network area at the default values. We varied the number of hosts from 2500 to 25000. In other words, we increased host density from 0.01 to 0.1 *host/meters*$^2$. With host density of at least 0.04 *host/meter*$^2$, all hosts in each of the simulated networks were reachable from the broadcast source. When the host density is below 0.04 *host/meter*$^2$, networks were partitioned. That is, some hosts were not reachable from the broadcast source in the center of the domain region. In this case, broadcast reachability is still 100%, accounting for only hosts reachable from the broadcast source using Simple Flooding. Figure 7 shows the retransmission overhead under four techniques. Simple Flooding has 100% retransmission overhead as expected. Self Pruning provides the best savings in retransmission overhead (about 40% compared to Simple Flooding) in the most sparse network in our study. The retransmission overhead increases as the network becomes denser. This is because a host forwards the packet it receives unless all its 1-hop neighbors are included in the packet header. Hence, the scheme cannot reduce retransmission overhead by much when a host has more neighbors. When examining our results, we found that only the hosts close to the network borders can avoid retransmissions. DBDT-ABS performs only slightly better than Simple Flooding at the lowest density. However, as the density increases, DBDT-ABS improves and eventually outperforms Self Pruning. Let's examine the savings in the retransmission overhead of the three schemes compared to Simple Flooding at 0.08 host density. The results show that setting static assessment periods like DBDT-ABS saves retransmission overhead by about 25% and using 1-hop neighbor knowledge like Self Pruning saves retransmission overhead by about 8%. We can estimate that a hybrid scheme that tracks 1-hop neighbors and utilizes static assessment periods should offer the savings in retransmission overhead by about 33% for this host density. Our ASAP scheme has a savings in retransmission overhead of 82%. Therefore, the 49% additional savings in the retransmission overhead comes from the adjustment of the assessment period each host does to adapt the original forwarding order according to duplicate packets it has seen. Our ASAP scheme outperforms the other simulated schemes for all host density in this study, especially for dense networks. This is because, a waiting host can adapt more in a dense network than in a sparse network since it has more neighbors to overhear information from.

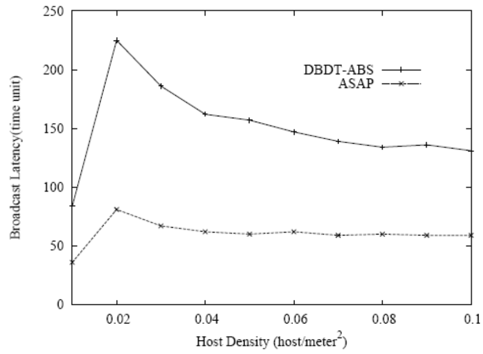



Fig. 7 Effect of host density on retransmission overhead.

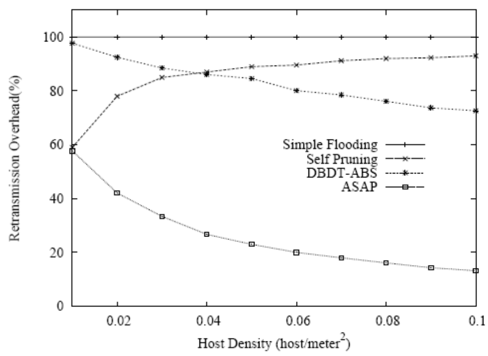


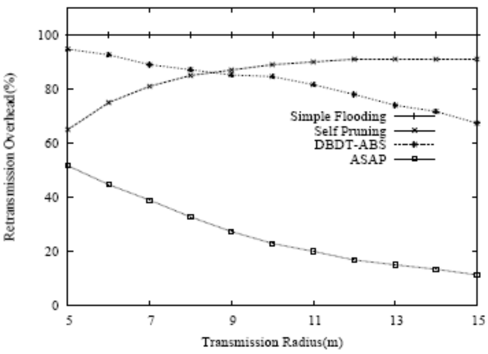

Fig. 8 Effect of host density on broadcast latency.

Let's turn our attention to broadcast latency. Recall that we only compare ASAP against DBDT-ABS since the other two schemes do not use any assessment periods. Figure 8 shows that DBDT-ABS incurs significantly longer broadcast latency than ASAP does. For the lowest host density, broadcast latency is unusually low. This is because a small number of hosts were reachable from the broadcast source due to network partitioning. As density increases to 0.02 *host/meter*$^2$, many more hosts were reachable, resulting in higher broadcast latency in both schemes. ASAP outperforms DBDT-ABS and has consistently low broadcast latency across all the studied host density. This is because ASAP enables the frontier hosts closest to the perimeter of a broadcast to retransmit the packet without any delays, allowing a broadcast packet to reach the distant hosts as quickly as possible. The broadcast latency under DBDT-ABS decreases as the network becomes denser. This can be explained as follows. In a dense network, each host has several neighbors within its transmission range. Hence, there is a higher probability that a host closest to the broadcast perimeter is very close to the perimeter. This host only needs to wait a small amount of time before retransmitting the packet, which results in low broadcast latency. As host density increases, ASAP has information from more neighbors to adjust its assessment period. However, the adjustment does not reduce broadcast latency by much because of the already good performance offered by the instant frontier forwarding feature.

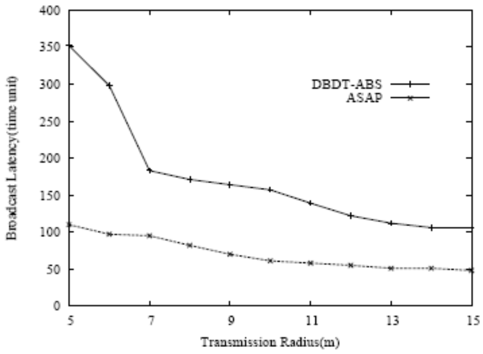
### 5.2.2 Effect of Transmission Radius

In this study, we varied the transmission radius from 5 to 15 meters. The network area and host density were fixed at the default values. Figure 9 depicts the results.

Our ASAP scheme outperforms the other schemes across all the studied transmission ranges with respect to retransmission overhead and broadcast latency. The plots in this figure exhibit similar trends to those in the study of the effect of host density. The rationale is as follows. Given a fixed number of hosts in the studied domain, the increasing transmission radius results in each host having more neighbors. This effect is close to the effect of increasing host density but keeping the same transmission radius like in the previous study.




(a) Retransmission overhead




(b) Broadcast latency

Fig. 9 Effect of transmission radius.

## 6. Conclusion

We have presented a highly efficient, yet very simple, broadcasting strategy called Adaptive Scheduling with Adaptive assessment Period. The new scheme is very effective in reducing broadcast latency and eliminating unnecessary broadcast retransmissions due to an excellent interplay of its two important features: instant frontier forwarding and adaptive assessment periods. The former feature lets the farthest frontier host forward the packet without any delays, making the broadcast packet reach many distant hosts as quickly as possible. With the latter feature, waiting hosts either (i) lengthen their assessment period as long as possible to avoid unnecessary retransmission or (ii) shorten their assessment period to make the necessary retransmission as quickly as possible. We have also extended the proposed scheme to handle host heterogeneity, where different hosts can have different transmission radius. Unlike many existing broadcasting techniques that require hosts to keep track of their neighboring hosts within 2-hop distance, our technique requires each host to track only its 1-hop neighbors. Therefore, it is more adaptive to host mobility and incurs much less overhead in terms of control-related network traffic and computation. Our study shows that under most of the simulated networks, our new technique significantly outperforms other existing schemes we simulated.

## References

[1] D. Johnson and D. A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In T. Imielinski and H. F. Korth, editors, *Mobile computing*, pages 153– 181. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.

[2] C. E. Perkins. Ad Hoc On-Demand Distance Vector (AODV) Routing. INTERNET DRAFT - Mobile Ad hoc NETworking (MONET) Working group of the Internet Engineering Task Force (IETF), November1997.

[3] Z. J. Haas and M. R. Pearlman. The Zone Routing Protocol (ZRP) for Ad Hoc Networks. INTERNET DRAFT - Mobile Ad hoc ETworking (MONET) Working group of the Internet Engineering Task Force (IETF), November 1997.

[4] Y. Ko and N. Yaidya. Location-Aided Routing (LAR) in Mobile Ad Hoc Networks. In *Proc. Of MOBICOM'98*, pages 66– 75, 1998.

[5] C. Ho, K. Obraczka, G. Tsudik, and K. Viswanath. Flooding for Reliable Multicast in Multi-hop Ad Hoc Networks. In *Proc. of the Int'l Workshop on Discrete Algorithms and Methods for Mobile Computing and Communication*, pages 64– 71, 1999.

[6] J. Jetcheva, Y. Hu, D. Maltz, and D. Johnson. A Simple Protocol for Multicast and Broadcast in Mobile Ad Hoc Networks. Internet Draft: draft-ietf-manet-simple-mbcast-01.txt, July 2001.

[7] S. Ni, Y. Tseng, Y. Chen, and J. Sheu. The Broadcast Storm Problem in a Mobile Ad Hoc Network. In *Proc. of MOBICOM'99*, pages 151– 162, 1999.

[8] Y. Tseng, S. Ni, and E. Y. Shih. Adaptive Approaches to Relieving Broadcast Storms in a Wireless Multihop Mobile Ad Hoc Networks. In *Proc. of ICDCS'01*, pages 481– 488, 2001.

[9] Y. Sasson, D. Cavin, and A. Schiper. Probabilistic broadcast for flooding in wireless mobile ad hoc networks. In *Swiss Federal Institute of Technology, Technical Report IC/2002/54*, 2002.

[10] M. T. Sun, W. C. Feng, and T. H. Lai. Location Aided broadcast in wireless ad hoc networks. In *Proc. of GLOBECOM'01*, 2001.

[11] H. Lim and C. Kim. Multicast Tree Construction and Flooding in Wireless Ad Hoc Networks. In *Proc. of the ACM Int'l Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM)*, pages 61– 68, August 2000.

[12] Brad Williams and Tracy Camp. Comparison of broadcasting techniques for mobile ad hoc networks. In *MobiHoc '02: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, pages 194– 205. ACM Press, 2002.

[13] T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, and a. Qayyum et L. Viennot. Optimized Link State Routing Protocol. In *Proc. of IEEE INMIC '01*, 2001.

[14] M. Benzaid, P. Minet, and K. Al Agha. Integrating fast mobility in the OLSR routing protocol. In *Proc. of IEEE Conference on Mobile and Wireless Communications Networks (MWCN)*, 2002.

[15] W. Lou and J. Wu. On Reducing Broadcast Redundancy in Ad Hoc Wireless Networks. *IEEE Transactions on Mobile Computing*, 1(2):111– 123, 2002.

[16] W. Peng and X. Lu. AHBP: An Efficient Broadcast Protocol for Mobile Ad Hoc Networks. *Journal of Science and Technology, Beijing, China*, 2002.

[17] A. Laouiti, A. Qayyum, and L. Viennot. Multipoint relaying: An efficient technique for flooding in mobile wireless networks. In *35th Annual Hawaii International Conference on System Sciences (HICSS'2001)*. IEEE Computer Society, 2001.

[18] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks. In *Proc. of MOBICOM'01*, pages 85– 96, July 2001.

[19] W. Peng and X. Lu. On the Reduction of Broadcast Redundancy in Mobile Ad Hoc Networks. In *Proc. of MOBIHOC'00*, 2000.

[20] I. Stojmenovic, M. Seddigh, and J. Zunic. Dominating Sets and Neighbor Elimination Based Broadcasting Algorithms in Wireless Networks. *IEEE Transactions on Parallel and Distributed Systems*, 13(1):14– 25, January 2002.

[21] J. Sucec and I. Marsic. An Efficient Distributed Network-Wide Broadcast Algorithm for Mobile Ad Hoc Networks. In *Rutgers University, CAIP Technical Report 248*, September 2000.

[22] J. Wu and H. Li. On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks. In *Proc. of the 3rd Int'l Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DiaLM)*, pages 7– 14, 1999.

[23] J. Wu and F. Dai. Broadcasting in Ad Hoc Networks Based on Self-Pruning. In *Proc. of INFOCOM'03*, March 2003.

[24] J. Wu and F. Dai. A Generic Distributed Broadcast Scheme in Ad Hoc Wireless Networks. In *Proc. Of the 23rd IEEE International Conference on Distributed Computing Systems (ICDCS)*, pages 460– 467, May 2003.

[25] Weifa Liang. Constructing minimum-energy broadcast trees in wireless ad hoc networks. In *Proc. Of ACM MobiHoc 2002*, pages 112– 122, Lausanne, Switzerland, June 2002.

[26] Rajiv Gandhi, Srinivasan Parthasarathy, and Arunesh Mishra. Minimizing broadcast latency and redundancy in ad hoc networks. In *MobiHoc '03: Proceedings of ACM international symposium on Mobile ad hoc networking & computing*, pages 222– 231. ACM Press, 2003.

[27] C. Randell and H. Muller. Low cost indoor positioning system. In *Ubicomp2001*, pages 42– 48, 2001.

[28] E. D. Kaplan. *Understanding GPS: principles and applications*. Artech House, Boston, MA, 1996.

[29] R.Want and A. Hopper. The active badge location system. *ACM Transactions on Information Systems*, 10(1):91– 102, 1992.

[30] A. Ward, A. Jones, and A. Hopper. A new location technique for the active office. *IEEE Personal Communications Magazine*, 4(5):42– 47, 1997.

[31] P. Castro, P. Chiu, T. Kremenek, and R. Muntz. A Probabilistic Room Location Service for Wireless Networked Environments. In *Ubicomp2001*, pages 18– 34, 2001.