

Percolation Driven Flooding for Energy Efficient Routing in Dense Sensor Networks

Gergely Vakulya and Gyula Simon

Abstract—Simple flooding algorithms are widely used in ad hoc sensor networks either for information dissemination or as building blocks of more sophisticated routing protocols. In this paper a percolation driven probabilistic flooding algorithm is proposed, which provides large message delivery ratio with small number of sent messages, compared to traditional flooding. To control the number of sent messages the proposed algorithm uses locally available information only, thus induces negligible overhead on network traffic. The performance of the algorithm is analyzed and the theoretical results are verified through simulation examples.

Keywords— *flood routing, percolation, sensor network.*

1. Introduction

Energy efficiency is a key question in wireless sensor networks made of large number of inexpensive nodes with limited energy reserves and restrained processing and communication capabilities. Such networks have been used in a wide range of applications with various goals, using different platforms and technology [1]–[4]. Independently of the nature of the application, however, certain middleware services are always present, one of them being routing. Limited resources on the sensor nodes require that routing protocols be simple, energy-conserving and robust.

Depending on the actual deployment scenario and available hardware, several ad hoc routing algorithms have been proposed. Location aware routing schemes, e.g., greedy perimeter stateless routing [5], location aided routing [6], distance routing effect algorithm [7] use position information of the nodes obtained from global positioning system (GPS) or other localization services. Other data centric schemes do not use location information, e.g., directed diffusion [8], ad hoc on demand distance vector routing (AODV) [9], dynamic source routing [10], temporarily ordered routing [11], or zone routing [12], just to name a few. To allow sensor networks better cope with scaling, hierarchical protocols were designed, e.g., low energy adaptive clustering hierarchy (LEACH) [13] and threshold sensitive energy efficient sensor network protocol (TEEN) [14].

The simplest routing protocol is flooding [15], which is a useful simple (but inefficient) protocol in itself, and also is a building block for several more sophisticated algorithms, e.g., directed flood routing [16], controlled flood routing [17], tiny ad hoc routing [18], or AODV [9], which is used in Zigbee's routing protocol [19].

The basic flooding protocol is the following:

1. *Start*. When a source node intends to send a message to all other nodes in the network, it broadcasts the message to its neighbors.
2. *Relay*. When a node receives a message for the first time, it rebroadcasts the same message to its neighbors. All other copies of the same message, received later, will be discarded by the recipient node.

Flooding naturally can be used in its basic form for network-wide dissemination of information. In this context the important performance criteria of the algorithm are delivery ratio, latency, and the energy efficiency. Flooding is also used for route discovery [9]. Route discovery protocols find a route from a source node to a destination node in two phases. First the network is flooded by a discovery request message originated from the source, and then a reply message is sent back from the destination to the source node. In this way the network learns the path from the source to the destination and this route will be used in the subsequent communication sessions. In this context the quality – primarily the length – of the discovered route is an important performance criterion as well.

Apart from its simplicity the main advantage of the flood routing protocol lies in its robustness: the implicit redundancy built in the algorithm provides resistance against high degree of message loss and node failures as well. The drawback of the algorithm is the large number of packets transmitted in the network, referred to as broadcast storm: whether it is necessary or not, each node will retransmit each message. In a dense network the unnecessarily high number of messages can cause frequent collisions (and thus performance degradation) and it wastes network energy as well [20].

To reduce the number of routing messages, probabilistic variations of the flood routing were proposed. The main idea behind these algorithms is that a node randomly decides whether to forward a message or not. Several variants of random flood algorithms were proposed. According to the simplest protocol, a node will rebroadcast a message after its first reception with probability p and discards it with probability $1 - p$ (clearly, $p = 1$ results in basic flooding).

Test results verify what intuition suggests: higher p values provide higher network coverage than small p values. Moreover, in sufficiently large and sufficiently dense random networks there is an interesting bimodal behavior of

the algorithm: if $p > p_c$, where the critical probability p_c depends on the network topology, the message reaches practically all nodes in the network, otherwise only a small portion of the network receives the message. Thus the optimal choice clearly would be $p = p_c$. Modified algorithms try to further increase the performance by various modifications; e.g., premature message death can be avoided by varying p as a function of hop-distance from the source: nodes close to the source rebroadcast messages with higher probability, while distant nodes use smaller p . A comprehensive study on random flooding algorithms can be found in [15]. The common problem with these algorithms is that the design parameters (probabilities) depend on actual network layout, no automatic or adaptive solution is known. Especially in a network with varying node density the probabilistic algorithms tuned to work reliably will be suboptimal.

More sophisticated adaptive schemes can successfully handle networks with varying properties as well. Location based algorithms use node positions to optimize retransmissions [21], graph-based algorithms utilize connectivity information up to 2-hop neighbors to construct a dominating set in a distributed way [22]. Other heuristic rules, e.g., message counters and neighbor coverage schemes were also proposed [20]. These schemes can adapt automatically to changing topology at the price of higher complexity.

Sensor networks are often modeled by the Poisson-Boolean model. In this model nodes are scattered on the plane by a Poisson point process with constant density λ , and each node has a fixed communication radius r (disc model). Two nodes can communicate with each other if their distance is less than r . The Poisson point process applies reasonably well to controlled, but still random deployments. The simplistic disc model, however, is far from reality, where communication range is not circularly symmetric, but rather has an anisotropic shape. Speaking of connectivity, however, the disc model has an important property: it can be considered the worst case model, since other transmission models provide easier percolation under similar circumstances [23].

Percolation theory [24] has important impact on wireless communication networks. Results in continuum percolation theory show that if the density of transmitting stations (or alternatively, their communication power) in a Poisson-Boolean network is greater than a critical value λ_c then an unbounded connected component is formed with probability one (the network percolates) [23], [25]. On the other hand, if $\lambda < \lambda_c$ then all connected components are finite with probability one. Thus percolation is an important property of a network if long distance multi-hop communication is required.

In this paper a new percolation inspired solution will be proposed based on probabilistic flood routing algorithms. The new algorithm adaptively sets the rebroadcast probability based on the network topology, using locally available neighborhood information only. The proposed algorithm provides high performance with a low number of

messages and can adapt its behavior dynamically to the network properties. The proposed algorithm is only marginally more complex than the basic flood routing thus it can be successfully used in applications with limited resources as well.

The rest of the paper is organized as follows. In Section 2 related works will be summarized. In Section 3 the proposed algorithm will be introduced. An important property of the algorithm will be proven based on percolation theory: every message broadcast in the network will reach infinite number of nodes almost surely. In practice this property ensures that the message reaches almost all of the nodes in a finite-sized network. The performance evaluation of the algorithm through simulation can be found in Section 4. Conclusions are drawn in Section 5.

2. Related Work

Results of percolation theory have been applied to wireless networks, forming the theory of continuum percolation [26]. More realistic network models, i.e., unreliable communication links and anisotropic radiation patterns were studied in [23]. An important result of [23] shows that the debatable disc model actually provides a conservative estimate of the network connectivity: percolation in networks with disc communication shapes is more difficult than in networks with any other convex communication shape with the same surface ("discs are hardest" conjecture). This property justifies the usage of the disc model when connectivity issues are studied.

Results from percolation theory have been infiltrating various recent sensor networking algorithms. As a practical application of this phenomenon, the distributed minimum-link-degree rule was proposed to counterbalance local spatial inhomogeneities in ad hoc networks [25]. To provide network wide connectivity, the transmission powers of the nodes are tuned so that each node has at least a minimum number of neighbors.

In [27] a distributed energy saving mechanism is used by switching the nodes in the network on and off, while providing percolation in the network. The simple rule for the algorithm is to keep the active time ratio of the nodes larger than λ/λ_c . This algorithm does not scale well since each node in the network has to know the global parameter λ . The idea was further elaborated in [28], where a practically useable distributed algorithm was proposed for the control of the active time ratio. In this algorithm only the local density, namely the number of neighbors is used. For a node with degree k the active ratio η_k is defined as follows:

$$\eta_k = \begin{cases} \frac{\phi}{k} & k > \phi \\ 1 & k \leq \phi \end{cases},$$

where ϕ is a density independent design parameter. A similar idea will be used in the proposed flooding algorithm.

3. Percolation Driven Flooding

The idea behind the percolation driven flood routing is the following: use probabilistic flooding and set the retransmission probability adaptively at every node so that with high probability the message reaches almost all of the nodes in the network.

The algorithm is the following:

1. The source node broadcasts the message with probability one.
2. After the first reception of a message node n rebroadcasts it with probability p_n , and discards it with probability $1 - p_n$. Copies of the same message received multiple times are discarded with probability one. Probability p_n is defined as

$$p_n = \begin{cases} \frac{K_{\min}}{K_n} & K_n > K_{\min} \\ 1 & \text{otherwise} \end{cases}, \quad (1)$$

where: K_n is the degree of node n (i.e., the number of neighbors of node n), and the $K_{\min}$ design parameter is the required minimum number of neighbors. Two nodes are neighbors if they can hear each other (i.e., a symmetric link exists between them).

The algorithm requires neighbor discovery to determine the number of neighbors. Each node can gather this local information by transmitting and receiving *hello messages*. In practical situations parameter $K_{\min}$ is chosen around 7, as will be justified by the simulation examples.

Theorem 1: If in the infinite random network, generated by a Poisson-Boolean process with density λ and communication radius r , there exists design parameter $K_{\min} < \infty$ independent of λ , such that when node n with degree K_n transmits a message with probability according to Eq. (1), then each message reaches infinite number of nodes with probability one.

Proof: Let us denote the graph representing the network by $G(\lambda, r, 1)$ where λ is the density, r is the communication radius, and the third parameter is a probabilistic value used in the generator process (used and discussed later) and is simply set to one at the moment. Using scaling, we define another process with density $\lambda' = \lambda r^2$ and communication radius 1. This process yields the same graph $G(\lambda', 1, 1)$. Now let us use the following Theorem 2 [28].

Theorem 2: Given $G(\lambda', 1, 1)$ with $\lambda' > \lambda'_c$, there exists $2 < \varphi < \infty$, independent of λ , such that when each node with degree k is active with probability:

$$p(k) = \begin{cases} \frac{\varphi}{k} & k > \varphi \\ 1 & k \leq \varphi \end{cases},$$

then $G(\lambda', 1, p(\cdot))$ is percolated. The proof of Theorem 2 can be found in [28].

If in the network we decide *before* a particular message is broadcasted which nodes will retransmit the message (when

they receive it) by activating/deactivating nodes with probability (1) we get $G(\lambda', 1, p(\cdot))$. In this way we construct a virtual network for each message and we can apply Theorem 2 to it. Thus there exist $K_{\min} = \varphi < \infty$ so that G percolates. From this it follows that the message will reach infinite number of nodes [24]. ■

In real-world finite-size networks Theorem 1 means that with an appropriate choice of $K_{\min}$ each message reaches practically all nodes in the network. Further properties of the algorithm will be analyzed through simulation examples.

4. Performance Evaluation

In this section the performance of the algorithm will be evaluated through simulation results. First the performance metrics and the network/communication models will be introduced, followed by the simulation environment and the simulation scenarios. Finally, the results of the simulations will be presented to illustrate the behavior of the proposed algorithm.

4.1. Performance Metrics

The main performance criteria of broadcast algorithms are the message delivery ratio and the number of sent messages. If the number of nodes in the network is N and the number of nodes receiving the message is N_{rec} then the coverage ratio is defined as $C_r = \frac{N_{rec}}{N}$.

Clearly, C_r close to 1 is required for a good quality of service.

Another quality metric is the total number of sent packets M while a message is propagated in the network. Trivially, for basic flood routing $M = N$, if $C_r = 1$. We expect lower number of messages and still high delivery ratio for a good performance algorithm. For easier comparison, we will use the normalized number of messages defined as

$$M_{norm} = \frac{M}{N}.$$

If flooding is used for route discovery, an important performance metric is the length L of the discovered route.

Other possible metrics are message delay/latency, length of flood period, and number of collisions. The actual low level details, e.g., implementation, hardware and media access control (MAC) layer properties are not investigated in detail in this paper, but their effect is examined through high level parameters they have impact on, i.e., C_r , M_{norm} , and L will be studied.

4.2. Network Model

To model random deployment of nodes or possibly mobile networks, in the simulations nodes are placed at random, according to a uniform distribution on a two-dimensional area. Finite communication distances are represented by the disc model, where a communication link is assumed to exist between two nodes if they are less than the communication radius apart. According to the results published

in [24], this – otherwise rather simple and idealistic – model can be considered a worst case model.

Our communication channel model does not deal with specific details of the physical layer or the MAC layer; rather we use a probabilistic model: a message can reach a neighbor within the communication radius with probability $p_{rec} < 1$. This high-level model represents message losses due to collision, fading, or other disturbances as well. The model does not distinguish between individual phenomena but rather incorporates the different sources in one parameter, thus some aspects of reality (e.g., inter-message dependencies) are neglected, but the model is faithful enough to provide useful and easy means for testing.

4.3. Simulation Environment

To perform high-speed simulation, a simulator was written in C to validate the efficiency of the proposed algorithm. The program places nodes randomly according to the different test scenarios considered. The number of nodes N , the communication radius r , and K_{min} are input parameters.

In each simulation, a new placement is generated, and a source node starts transmission. The simulator returns the size of connected component containing the source node, the number of nodes receiving the packet, and the total number of messages. Optionally, the software can visualize the topology of the network, the active data paths, and the nodes' reception status.

4.4. Test Scenarios

In the test we used 3 different scenarios to model sensor network setups. The first scenario is a random uniform distribution (with constant density), while in the second

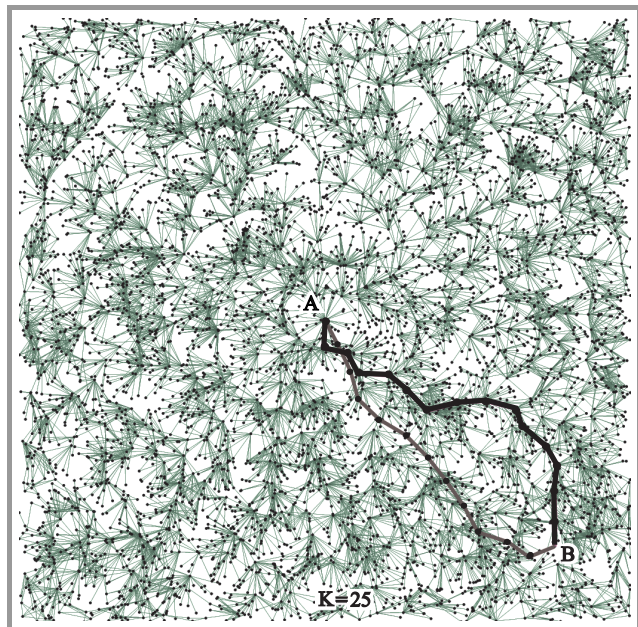


Fig. 1. Uniform random distribution scenario with 3000 nodes. The average number of neighbors is 25. Thin and thick lines show examples for the message paths between nodes A and B, discovered by flood and percolation driven flood, respectively.

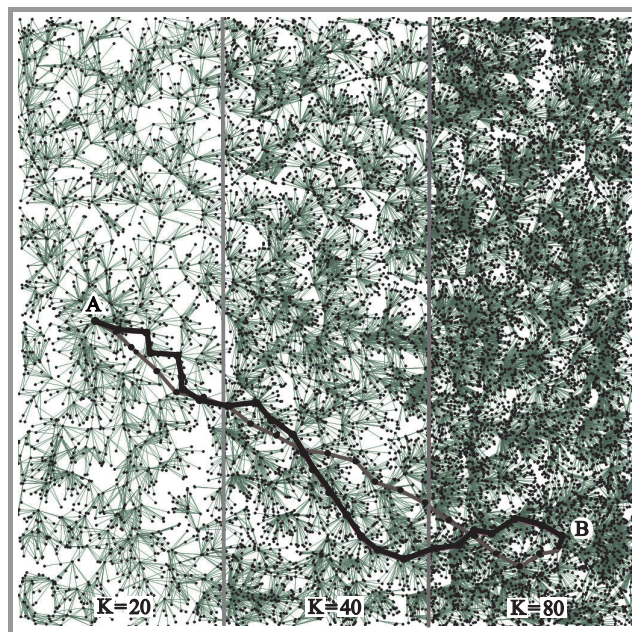


Fig. 2. The 3-density scenario. The average numbers of neighbors in the regions from left to right are 20, 40, and 80. Thin and thick lines show examples for the message paths between nodes A and B, discovered by flood and percolation driven flood, respectively.

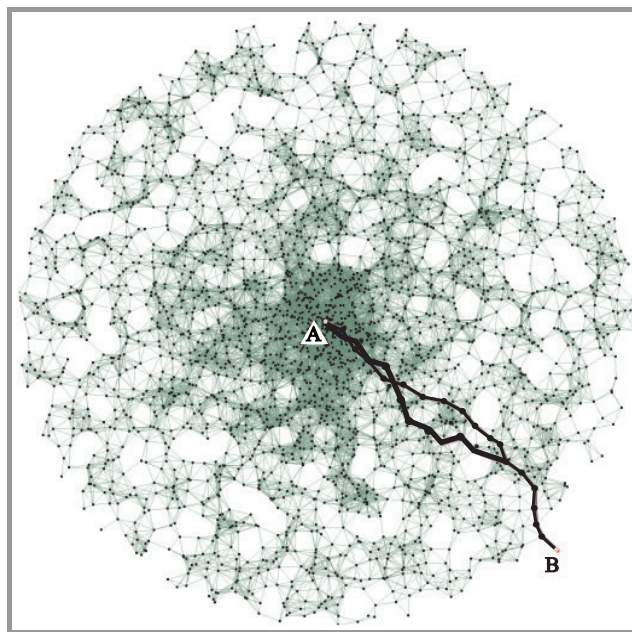


Fig. 3. The circular scenario with 3000 nodes. The average number of neighbors is 100 in the centre and 2 at the perimeter. Thin and thick lines show examples for the message paths between nodes A and B, discovered by flood and percolation driven flood, respectively.

scenario we used three regions; in each region nodes were placed with uniform random distribution, but with different densities. In the third scenario the density is increasing towards the centre, thus there is continuous change and extreme deviation in local density. Typical examples for the test networks can be seen in Figs. 1, 2 and 3.

4.5. Test Results

To test coverage ratio C_r and the number of messages M we placed 3000 nodes in a square-shaped area, as shown in Fig. 1.

We varied the communication radius to provide different network densities: the average number of neighbors K was set to 10, 30, and 100. The $K_{\min}$ value varied from 1 to 20 in 0.25 steps. We run the simulation 100 times for each communication radius, $K_{\min}$, and p_{rec} values, thus each point in the subsequent graphs is the average of 100 experiments.

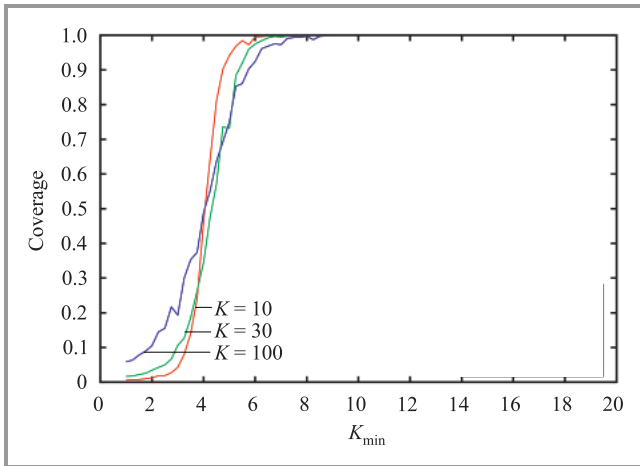


Fig. 4. Coverage versus $K_{\min}$ for different network densities in the uniform distribution scenario (constant network density), $p_{rec} = 1$.

Figure 4 presents coverage versus $K_{\min}$, while p_{rec} was set to constant 1 (error-free communication). The graphs for different network densities are similar: when $K_{\min}$ is low ($K_{\min} < 3$), the message delivery ratio is very low. If $K_{\min}$ is increased, the coverage is increasing very quickly, the critical value being around $K_{\min} \approx 4.5$. Coverage reaches 90% at $K_{\min} \approx 5$. If $K_{\min} > 7$, practically all nodes in the network receive the message. Figure 4 illustrates that percolation driven flood can indeed be used in networks with different densities: $K_{\min}$ is independent of the actual network density.

In Fig. 5 the normalized number of sent messages M_{norm} is shown versus $K_{\min}$, for the previous experiments. In case of the conventional flood routing $M_{norm} = 1$, because all nodes relay the received message. As Fig. 4 illustrates, $K_{\min} > 7$ gives almost perfect delivery ratio. Around this value $K_{\min} \approx 7$ the total number of messages is greatly decreased, as presented in Fig. 5. Depending on the node density, in the tests the total number of messages were reduced by 30..95%, higher reduction rates belonging to higher densities.

Clearly, in networks, where the average node degree K is only slightly higher than $K_{\min}$ the number of messages can be decreased only moderately, while in dense networks a much lower number of messages is enough to provide good delivery ratio, as shown in Fig. 5. According

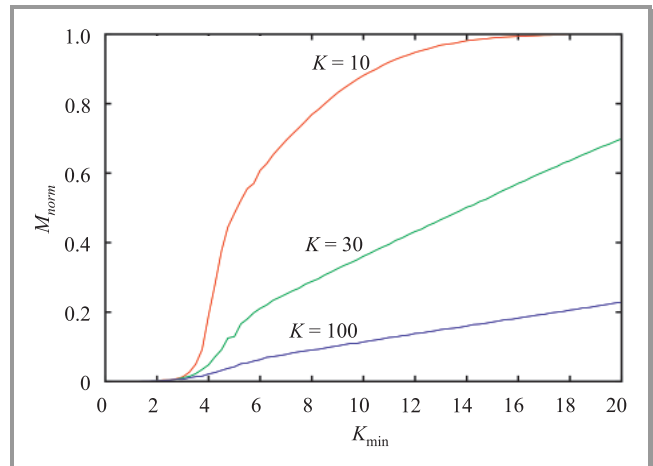


Fig. 5. Total number of messages versus $K_{\min}$ for different network densities in the uniform distribution scenario (constant network density), $p_{rec} = 1$.

to Figs. 4 and 5, in dense networks the percolation driven flood algorithm can effectively reduce the number of message while maintaining good coverage.

Figure 6 illustrates the effect of unreliable communication links. In the experiment constant density ($K = 30$) was used, while the p_{rec} reception probability was varied between 0.3 and 1. The figure shows that the algorithm

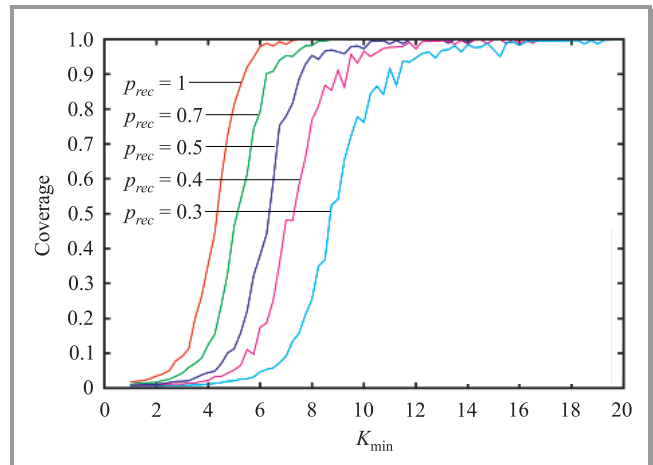


Fig. 6. Coverage versus $K_{\min}$ for different p_{rec} in the uniform distribution scenario (constant network density), $K = 30$.

can provide high coverage even in the presence of bad quality communication links, but as intuitively expected, higher $K_{\min}$ is necessary as the communication channel degrades.

Figure 7 presents the associated number of messages for the unreliable communication experiment.

Percolation driven flood routing algorithm is capable of handling varying network densities. To test this property we used the scenario illustrated in Fig. 2, where three areas with different densities are present. Clearly, a constant retransmission probability would either be suboptimal (set to provide sufficient coverage in the less dense area

as well) or would not provide good coverage in the sparse regions of the network.

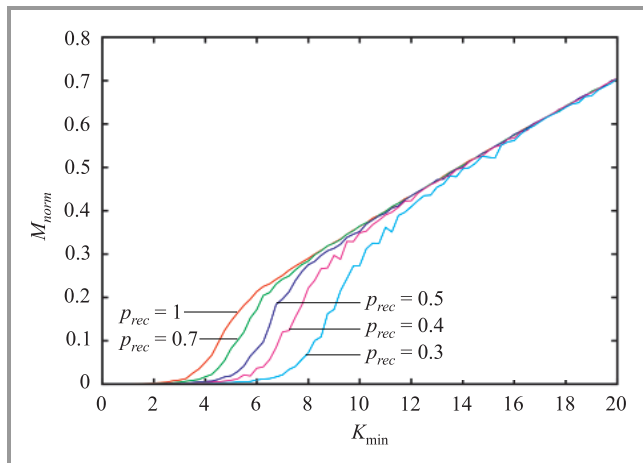


Fig. 7. Number of total messages versus $K_{\min}$ for different p_{rec} in the uniform distribution scenario (constant network density), $K = 30$.

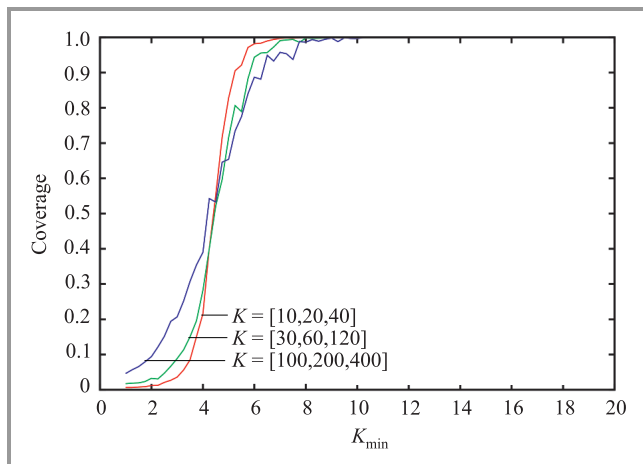


Fig. 8. Coverage versus $K_{\min}$ for different network densities in the varying network density scenario, $p_{\text{rec}} = 1$.

Figure 8 illustrates the effectiveness of the percolation driven flood algorithm, showing the coverage versus $K_{\min}$ when the densities of regions from left to right were set to [10, 20, 40], [30, 60, 120], and [100, 200, 400]; and $p_{\text{rec}} = 1$. The behavior is quite similar to that of the first scenario (constant density): percolation happens around the same $K_{\min} \approx 5$, and practically full coverage can be provided if $K_{\min} > 7$.

The associated total numbers of messages are shown in Fig. 9, where the gain with respect to the basic flooding algorithm is apparent.

In Fig. 10 the number of coverage versus $K_{\min}$ is illustrated in the circular scenario. In the experiments the number of neighbors varied between [120..4], [220..20] and [525..50], respectively. The figure clearly presents the excellent performance of the algorithm even if there are extreme differences in the number of neighbors of each node. The ideal value for $K_{\min}$ is around 7, in this case the algorithm gives

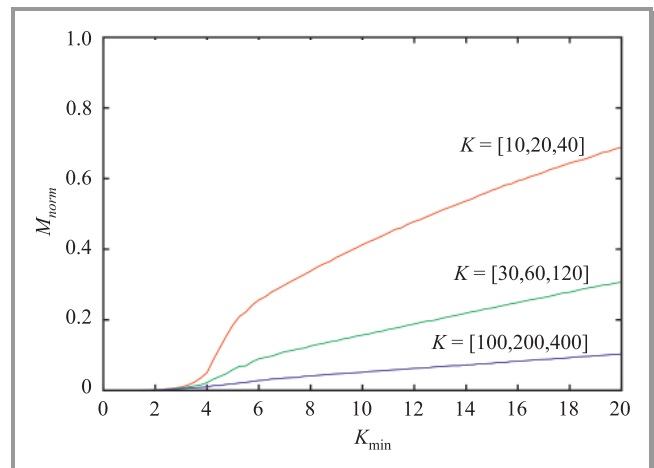


Fig. 9. Number of total messages versus $K_{\min}$ for different network densities in the varying network density scenario, $p_{\text{rec}} = 1$.

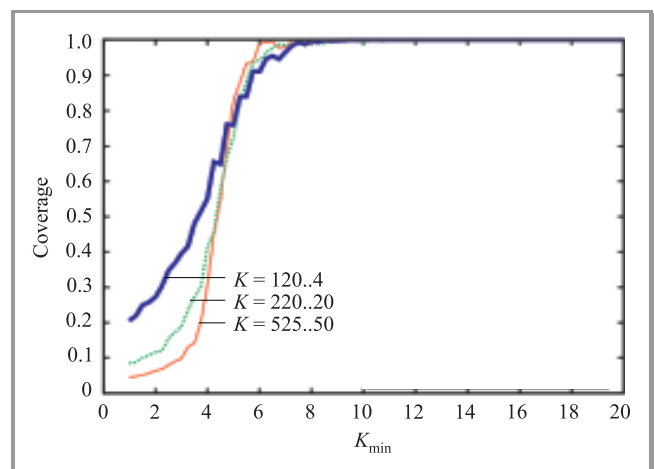


Fig. 10. Coverage versus $K_{\min}$ for different network densities in the circular scenario, $p_{\text{rec}} = 1$.

at least 95% delivery ratio. In case of $K_{\min} = 8$ practically full coverage is provided.

In Fig. 11 the normalized number of messages is shown for the three different density intervals. In case of $K_{\min} = 8$,

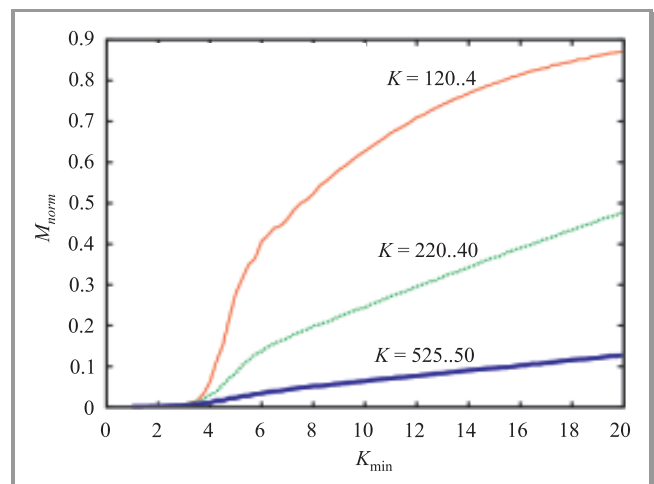


Fig. 11. Number of total messages versus $K_{\min}$ for different network densities in the circular scenario, $p_{\text{rec}} = 1$.

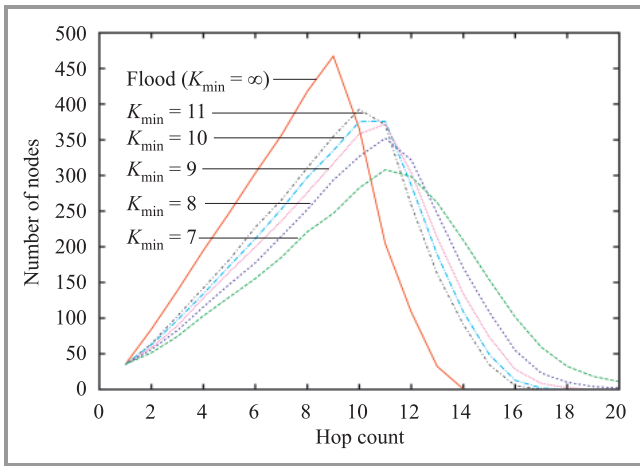


Fig. 12. Distribution of the hop counts in the uniform distribution network for different $K_{\min}$ values, $p_{rec} = 1$.

the algorithm gives a gain of 55–95%, with respect to the original flooding.

The length L of the discovered route is important when flooding is used for route discovery. To test the properties of percolation driven flooding in this respect we measured hop distances discovered by flooding and percolation driven flooding, for different $K_{\min}$ values. In the tests the uniform distribution (constant density) scenario was used with $K = 35$ and $p_{rec} = 1$. The source node was placed to the center of the field and hop distances to all other nodes were measured. The histogram created from averaging 100 independent experiments is illustrated in Fig. 12.

The distribution is linear for small hop distances. The explanation is shown in Fig. 13 for the ideal case: messages are propagated in belt-shaped increments. The areas of the subsequent belts increase linearly, thus the number of nodes in each belt is linearly increasing. If the node density is smaller, the number of nodes in each belt is smaller, thus the slope of the histogram will be smaller. According to Fig. 12, smaller $K_{\min}$ also causes decrease in the slope: e.g., for $K_{\min} = 9$, the slope is 40% smaller than in the basic flooding case. This means that the detected route is 40% longer than that of the basic flooding.

The declining part at higher hop distances is due to the finite size of the network: the messages reach the network

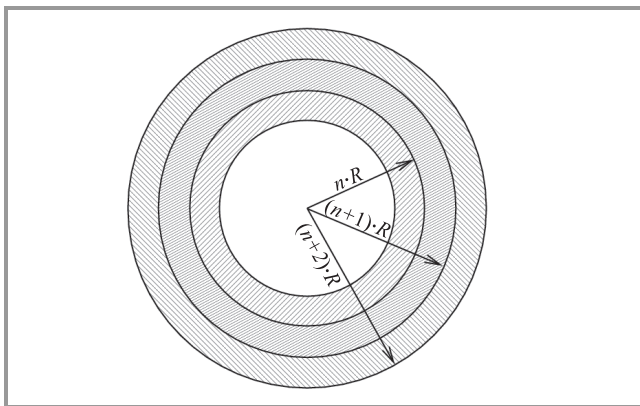


Fig. 13. Ideal propagation of a message in a dense network.

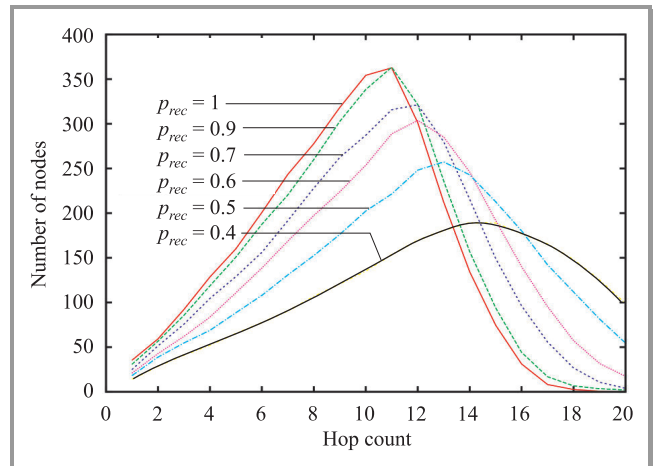


Fig. 14. Distribution of the hop counts in the uniform distribution network for different p_{rec} values, $K_{\min} = 9$.

perimeter and eventually die. The tail of the distribution presents the maximum hop distance from the source, which is in the square setup ideally $\sqrt{2}$ times the hop distance measured at the maximum of the histogram.

The effect of unreliable links is shown in Fig. 14. The histogram shows the average of 100 experiments conducted in the uniform test scenario with $K = 35$, and $p_{rec} = 0.4..1$. As intuitively expected, unreliable links cause smaller slope, i.e., higher hop distances. For example, link quality $p_{rec} = 0.7$ results in the increase of hop distance by 18%, while for the rather unreliable link quality $p_{rec} = 0.4$ the increase was as high as 63%.

As simulation examples show, percolation driven flooding results in considerably higher hop distances than basic flood routing when small $K_{\min}$ parameters are used. Naturally, low link quality also causes larger hop distances. This effect may be an undesired side effect in route discovery protocols, thus in these applications basic flooding may be more advantageous.

5. Conclusions

A percolation driven flood routing algorithm was proposed, which uses only locally available neighborhood information to reduce broadcast storm. The algorithm is able to massively reduce the number of messages in the network and maintaining high delivery ratio at the same time. Theoretical results prove the usefulness of the algorithm: it is able to provide high coverage, if the network density is high enough.

Simulation tests were performed to validate the performance of the algorithm. The proposed algorithm reduced the total number of messages in the network by 30–95%, depending on the network density, while the coverage was almost 100%, with appropriate choice of $K_{\min}$ ($K_{\min} > 7$). The percolation driven flood algorithm is adaptive to changing node density thus provides high coverage with low number of messages in all scenarios. The algorithm is robust in the presence of unreliable links as well.

According to test result, the percolation driven flood routing algorithm results considerably (even 50%) longer hop distances when used for route discovery purposes.

The overhead of the algorithm is very low since only the number of neighbors must be known by each node. This information can be gained by simple neighborhood discovery protocols, e.g., sending and receiving *hello messages*.

For route discovery purposes the application of the traditional flooding is more advantageous, if the side effect of the longer routes is undesirable. On the other hand, the proposed algorithm is a superior and robust alternative to traditional flooding algorithm in dense networks for message dissemination purposes.

References

- [1] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring", in *Proc. WSN Conf.*, Atlanta, USA, 2002, pp. 88–97.
- [2] A. Arora *et al.*, "A line in the sand: a wireless sensor network for target detection, classification, and tracking", *Comput. Netw.*, vol. 46, no. 5, pp. 605–634, 2004.
- [3] A. Ledeczi *et al.*, "Countersniper system for urban warfare", *ACM Trans. Sen. Netw.*, vol. 1, no. 2, pp. 153–177, 2005.
- [4] T. He *et al.*, "VigilNet: an integrated sensor network system for energy-efficient surveillance", *ACM Trans. Sen. Netw.*, vol. 2, no. 1, pp. 1–38, 2006.
- [5] B. Karp and H. T. Kung, "Greedy perimeter stateless routing (GPSR) for wireless networks", in *Proc. ACM MobiCom Conf.*, Boston, USA, 2000, pp. 243–254.
- [6] Y. B. Ko and N. H. Vaidya, "Location-aided routing (LAR) in mobile ad hoc networks", in *Proc. ACM MobiCom Conf.*, Dallas, USA, 1998, pp. 66–75.
- [7] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward, "A distance routing effect algorithm for mobility (DREAM)", in *Proc. ACM MobiCom Conf.*, Dallas, USA, 1998, pp. 76–84.
- [8] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking", *IEEE/ACM Trans. Netw.*, vol. 11, no. 1, pp. 2–16, 2003.
- [9] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing", in *Proc. 2nd IEEE Worksh. Mob. Comput. Syst. Appl.*, New Orleans, USA, 1999, pp. 90–100.
- [10] D. B. Johnson and D. A. Maltz, *Dynamic Source Routing in Ad Hoc Wireless Networks*. Boston: Kluwer, 1996.
- [11] V. Park and M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks", in *Proc. IEEE Infocom Conf.*, Kobe, Japan, 1997, pp. 1405–1413.
- [12] Z. Haas and M. Pearlman, "The performance of query control schemes for the zone routing protocol", in *Proc. ACM SIGCOMM Conf.*, Vancouver, Canada, 1998, pp. 167–177.
- [13] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless sensor networks", in *Proc. Hawaii Int. Conf. Syst. Sci.*, Hawaii, USA, 2000.
- [14] A. Manjeshwar and D. P. Agrawal, "TEEN: a protocol for enhanced efficiency in wireless sensor networks", in *Proc. 1st Int. Worksh. Paral. Distrib. Comput. Iss. Wirel. Netw. Mob. Comput.*, San Francisco, USA, 2001.
- [15] Z. J. Haas, J. Y. Halpern, and L. Li, "Gossip-based ad hoc routing", *IEEE/ACM Trans. Netw.*, vol. 14, no. 3, pp. 479–491, 2006.
- [16] M. Maróti, "Directed flood-routing framework for wireless sensor networks", in *Proc. 5th ACM/IFIP/USENIX Int. Conf. Middlew.*, Toronto, Canada, 2004, pp. 99–114.
- [17] L. H. Costa, M. D. De Amorim, and S. Fdida, "Reducing latency and overhead of route repair with controlled flooding", *Wirel. Netw.*, vol. 10, no. 4, pp. 347–358, 2004.
- [18] P. Gburzynski and W. Olesinski, "On a practical approach to low-cost ad hoc wireless networking", *J. Telecommun. Inform. Technol.*, no. 1, pp. 29–42, 2008.
- [19] Zigbee Alliance [Online]. Available: <http://www.zigbee.org>
- [20] Y. C. Tseng, S. Y. Ni, and E. Y. Shih, "Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network", in *Proc. 21st Int. Conf. Distrib. Comput. Syst.*, Phoenix, USA, 2001, pp. 481–488.
- [21] J. Yang, B. Kim, M.-T. Sun, and T.-H. Lai, "Location-aided broadcast in wireless ad hoc networks", *J. Inform. Sci. Eng.*, vol. 23, no. 3, pp. 869–884, 2007.
- [22] J. Wu, F. Dai, M. Gao, and I. Stojmenovic, "On calculating power aware connected dominating set for efficient routing in ad hoc wireless networks", *J. Commun. Netw.*, vol. 5, no. 2, pp. 169–178, 2002.
- [23] M. Franceschetti, L. Booth, M. Cook, R. Meester, and J. Bruck, "Percolation in multi-hop wireless networks", in *Caltech Paradise, ETR055*, 2003 [Online]. Available: <http://www.paradise.caltech.edu/papers/etr055.pdf>
- [24] G. Grimmet, *Percolation*. New York: Springer, 1989.
- [25] I. Glauche, W. Krause, R. Sollacher, and M. Greiner, "Continuum percolation of wireless ad hoc communication networks", *Physica A*, vol. 325, no. 3–4, pp. 577–600, 2003.
- [26] J. Jonasson, "Optimization of shape in continuum percolation", *Ann. Probab.*, vol. 29, no. 2, pp. 624–635, 2001.
- [27] O. Dousse, P. Mannersalo, and P. Thiran, "Latency of wireless sensor networks with uncoordinated power saving mechanisms", in *Proc. 5th ACM Int. Symp. MobiHoc'04*, Tokyo, Japan, 2004, pp. 109–120.
- [28] Z. Kong and E. M. Yeh, "Distributed energy management algorithm for large-scale wireless sensor networks", in *Proc. 8th ACM Int. Symp. MobiHoc'07*, Montreal, Canada, 2007, pp. 209–218.

Note: this text is reprinted from JTIT 2/2009



Gergely Vakulya received his M.Sc. in information technology from the University of Pannonia, Hungary, in 2008. Currently he is pursuing his Ph.D. in information science and technology at the University of Pannonia. His research interest is sensor networking.

e-mail: vakulya@dcs.uni-pannon.hu
University of Pannonia
Egyetem u. 10
8200 Veszprém, Hungary



Gyula Simon received his M.Sc. and Ph.D. in electrical engineering from the Budapest University of Technology, Hungary, in 1991 and 1998, respectively. Currently he is an Associate Professor at the University of Pannonia. His main research interest includes adaptive signal processing and sensor networks.

e-mail: simon@dcs.uni-pannon.hu
University of Pannonia
Egyetem u. 10
8200 Veszprém, Hungary