*International Journal of Parallel, Emergent and Distributed Systems*
Vol. 27, No. 3, June 2012, 225–234

# Broadcasting with seamless transition from static to highly mobile wireless ad hoc, sensor and vehicular networks

Ivan Stojmenovic*, Adnan Afsar Khan[1] and Nejib Zaguia[2]

*SEECS, University of Ottawa, Ontario, Canada*

(*Received 12 December 2011; final version received 12 December 2011*)

In a broadcasting task, source node wants to send the same message to all the other nodes in the network. Existing solutions range from connected dominating set (CDS) based on static networks to blind flooding for moderate mobility and hyper-flooding for highly mobile and frequently partitioned networks. Existing protocols for all scenarios are based on some threshold parameters (e.g. speed, which may be expensive to gather) to locally select between these three solution approaches. Here, we describe a new seamless broadcasting from static to mobile protocol, which adjusts itself to any mobility scenario without using any mobility or density-related parameter. Unlike existing methods for highly mobile scenarios, in the proposed method, two nodes do not transmit every time they discover each other as new neighbours. Each node maintains a list of two hop neighbours by periodically exchanging 'hello' messages, and decides whether or not it is in CDS. Upon receiving the first copy of message intended for broadcasting, it selects a waiting timeout and constructs two lists of neighbours: neighbours that received the same message and neighbours that did not receive it. Nodes not in CDS select longer timeouts than nodes in CDS. These lists are updated upon receiving further copies of the same packet. When timeout expires, node retransmits if the list of neighbours in need of message is non-empty. 'Hello' messages received while waiting, or after timeout expiration may revise all lists (and CDS status) and consequently the need to retransmit. This provides a seamless transition of protocol behaviour from static to highly mobile scenarios, which is applicable to a variety of multi-hop wireless networks.

**Keywords:** wireless networks; broadcasting; data communication

## 1. Introduction

We consider all network scenarios with respect to mobility or network density. Each node, or its neighbourhood, can be *static*, *moderately mobile* or *highly mobile*. The distinction between static, moderately and highly mobile nodes or networks, or between sparse, moderate and dense networks, is debatable. Contrary to existing solutions, we do not need any distinction between them for our new algorithm to run, and even outperform all existing ones.

The primary goal of a broadcasting task is to deliver the message to all nodes in a network (to achieve high delivery ratio) while minimising the total number of retransmissions. There exists a body of knowledge about centralised broadcasting, in which source node knows the whole network topology and can determine the whole broadcast process. However, collecting the required global knowledge demands

---

*Corresponding author. Email: ivan@site.uottawa.ca

ISSN 1744-5760 print/ISSN 1744-5779 online

http://dx.doi.org/10.1080/17445760.2011.649754
http://www.tandfonline.com

unacceptable communications overhead for dynamic networks. We consider only *localised* broadcasting protocols, in which nodes have only local knowledge about the network. One extreme is lack of any awareness of neighbours. That is, nodes do not send control 'hello' or beacon messages to inform neighbours about their presence. If each node periodically transmits 'hello' message, then one-hop knowledge can be gained. Our proposed algorithm is based on two-hop topological or one-hop positional knowledge, depending on whether or not nodes are aware of their own positions. Adding own position (in addition to node ID, if needed) to 'hello' message suffices for our protocol. Otherwise, a second round of 'hello' messages (each node transmits the list of its one-hop neighbours) to gather two-hop information generates alternative topological knowledge. We describe our new algorithm seamless broadcasting from static to mobile (SBSM) in terms two-hop topological knowledge. Modifications to work with one-hop positional knowledge are straightforward, and mainly in neighbour set definitions. This feature makes the same protocol appropriate as general solution to position-based networks such as sensor or vehicular, or topological-based networks such as ad hoc and actuator networks.

Most existing solutions address a single scenario. In *blind flooding*, each node, upon receiving the message for the first time, will retransmit it, and ignore further copies of the same message. This is a traditional broadcasting protocol that does not require neighbour knowledge. In dense networks, it can cause lot of redundancy, collisions and contentions and reduce rather than increase reliability [11]. Improved solutions aiming at full network coverage require two-hop neighbour topological knowledge and are based on connected dominating sets (CDSs) and neighbour elimination [20]. However, the mobility makes the maintenance of such knowledge expensive and, therefore, blind flooding is considered still as the favourite protocol for scenarios with moderate mobility. Blind flooding may not suffice in networks with temporary partitions and/or high mobility. *Hyper-flooding* was proposed for such scenarios [2,6,21], in which additional retransmissions occur whenever a node discovers a new neighbour. Reliability could be increased at the cost of high message overhead. However, reliability could even decrease if a node meets a dense network with all new neighbours informing it simultaneously, causing collisions and reception failure of all attempts.

We found only one protocol that attempts to describe a single broadcasting protocol suitable for all scenarios [21]. That protocol is based on applying high and low thresholds. Each node calculates its low and high threshold value based on past relative movements in its neighbourhood. It keeps track of relative movements in its neighbourhood, and compares it with low and high thresholds. Thus, different nodes can make different decisions and run different protocol modes. Each node decides to run scoped (restricted) flooding, blind flooding or hyper-flooding based on its own threshold values. Two threshold types were considered: mobility and traffic based. The protocol has a number of problems. First, the requested parameter value may be difficult or impossible to gather. For instance, the protocol uses speed and direction of movement, which adds some hardware to nodes and overhead to hello message exchanges. Traffic parameters are based on measuring collisions but they reduce reliability in high volume traffic and increase unnecessary overhead in low volume traffic.

Our objective is to describe a broadcasting protocol that will adapt itself to any mobility scenario automatically, without calculating, tracing and applying any type of thresholds, that is without using any parameter for distinguishing between mobility and density status of nodes, sub-networks or whole network. Therefore, our protocol will not measure speed or direction of movement, and will not monitor traffic for the purpose of deciding retransmission behaviour. Furthermore, we are looking for a protocol that will have fewer retransmissions and higher reliability. The proposed protocol is named 'seamless broadcasting from static to mobile networks' (SBSM). Each node maintains two lists, *R* (neighbours that received message) and *N* (neighbours that did not receive the same message). These lists are updated upon receiving each copy of the message, or a hello message. Nodes in CDS select shorter timeouts than nodes not in CDS. When waiting period expires, node retransmits if *N* is non-empty. 'Hello' messages may refresh *R* and *N* and subsequently cause further retransmissions. This adaptive protocol provides a seamless transition of protocol behaviour from static to highly mobile scenarios without monitoring or measuring any speeds or knowing what mobility scenario is.

This article is organised as follows. Literature review is discussed in Section 2. Our new protocol is described in Section 3. Conclusion and references complete this article.

Preliminary conference version of this paper appeared in Ref. [9]. In Ref. [9], the protocol was named 'Parameterless Broadcasting from Static to Mobile (PBSM)'. However, proportionality constant in waiting time delay was still a parameter in the protocol design. SBSM evolved from an idea described in 'Conclusion and future work' section of Ref. [15], which was credited to Stojmenovic ('Broadcasting in highly mobile networks, in preparation'), which eventually became Ref. [9].

## 2. Literature review

There exist a plethora of proposed broadcasting protocols. Their survey is given in Refs [16,18]. We describe here only protocols that are relevant to our proposed SBSM method and the adaptivity goal. Brief description of some competitive protocols is already given in the introduction; details can be found in Ref. [9] along with a number of other existing protocols that were demonstrated as being non-competitive.

### *2.1 Broadcasting in static networks*

When network is static and fixed, our SBSM reduces to the protocol described in [20]. We therefore present it in detail. Wu et al. described, in a series of articles (starting from [22]), a lightweight backbone construction scheme. We will use a modified definition from [20] of basic concept [22], because of its reduced (more precisely: eliminated) message overhead. A node is an *intermediate* node if it has two unconnected neighbours [22]. Node *A* is covered by neighbouring node *B* if each neighbour of *A* is also neighbour of *B*, and $key(A) < key(B)$. Nodes not covered by any neighbour are *inter-gateway* nodes. Node *A* is covered by two connected neighbouring nodes *B* and *C* if each neighbour of *A* is also a neighbour of either *B* or *C* (or both), $key(A) < key(B)$, and $key(A) < key(C)$. An intermediate node not covered by any neighbour becomes an *inter-gateway* node. An inter-gateway node not covered by any pair of connected neighbouring nodes becomes a *gateway* node. Dai and Wu [5] introduced a *generalised dominating set (DS)*, in which coverage can be provided by an arbitrary number of connected neighbours. The definition was modified in [18] to avoid similar message exchanges between neighbours, as follows. Node *A* is covered by its direct neighbours $B, C, D, \ldots$ if the neighbours $B, C, D, \ldots$ create connected sub-graph, any neighbour of *A* is a neighbour of at least one of nodes *B*, *C*, $D, \ldots$ and $key(A) < \min(key(B), key(C), key(D), \ldots)$. It is computationally simplified by Carle and Simplot-Ryl [3], as follows. First, each node checks if it is an intermediate node. Then each intermediate node *A* constructs a sub-graph *G* of its neighbours with higher *key* values. If *G* is empty or disconnected, then *A* is in CDS. If *G* is connected but there exists a neighbour of *A* which is not a neighbour of any node from *G*, then *A* is in CDS. Otherwise, *A* is covered and is not in CDS. Dijkstra's shortest path scheme can be used to test the connectivity. In *enhanced* CDS by Dai and Wu (elaborated in [8]), two-hop neighbours can be used to cover one-hop neighbours for smaller CDS. In this method, an intermediate node $u$ is not in CDS if there exists a connected set $A$ of its two-hop neighbours with higher priorities, such that each neighbour of $u$ either belongs to $A$ or is a neighbour of a node in $A$. Otherwise, $u$ is in CDS.

Wu's concepts require either one-hop knowledge of neighbours with their positions, or two-hop neighbour topology information. Experimental data from several sources confirm that Wu's concepts provide small size CDS on average. It was proven in [5] that generalised CDS concept has constant approximation ratio on average, and very low probability of having infinitely large approximation ratio. Each node makes decisions about CDS membership (in Wu's concept) without communications between nodes beyond the message exchanges that nodes use to discover each other and establish neighbourhood information.

In [20], the following framework and general algorithm were established for a reliable broadcasting. The algorithm is based on two concepts: CDS as the particular type of backbone that provides reliability, and neighbour elimination scheme (NES). In NES [12,20], a node does not need to rebroadcast a message if all its neighbours are believed to be covered by previous transmissions. After each received copy of the same message, a node eliminates, from its rebroadcast list, neighbours that are assumed to have received correctly the same message (based on the local knowledge). If the list becomes empty before the node decides to rebroadcast, the rebroadcasting is cancelled.

The general DS–NES (Dominating Sets NES) [20] for intelligent flooding proceeds as follows. The source node transmits the packet. Nodes not in CDS do not retransmit the packet. Upon receiving the first copy of the packet, node in the CDS will select a timeout period to wait. It will also eliminate from its forwarding list (originally containing all one-hop neighbours) all neighbours that received the same copy of the message. While waiting, more copies of the packet could be received. For each of them, all neighbours receiving it are eliminated from the forwarding list. When timeout expires, the node will retransmit if its forwarding list is non-empty, otherwise it will cancel retransmission. This framework was applied in [20] using clustering-based and Wu's concept-based backbones.

## 3. Broadcasting with seamless transition from static to mobile networks

### *3.1 Protocol overview and illustrations*

We propose an adaptive broadcasting protocol that does not require nodes to monitor and exchange their position, movement and/or traffic information, and yet performs better than existing threshold-based protocol [21]. Protocol is localised, and is based on applying CDS and neighbour elimination concepts on currently available neighbourhood information. Thus, two nodes do not transmit every time they discover each other as new neighbours. The proposed SBSM protocol does not rely on any threshold and provides smooth transition of protocol behaviour based on network dynamic.

Unlike other methods, in the proposed method, two nodes do not transmit every time they discover each other as new neighbours. This is the main novelty of the proposed SBSM protocol. The protocol should behave in a way such that success rate is nearly preserved while reducing flooding rate (overall number of messages sent) significantly. The other change is not to always rebroadcast the first time message is received, as in the blind flooding protocol, considered primarily for moderately mobile scenarios. This will create excessive messaging in dense networks with no or slow topology changes. In these networks, local knowledge information is nearly preserved while broadcasting is in progress, and therefore backbone nodes, having neighbours still in need of message, can be the first to retransmit.

Nodes periodically exchange hello messages to update local knowledge (one hop if position information is available, and up to two hops otherwise). CDS is calculated after each hello message round. Source node transmits the message. Upon receiving the message for the first time, each node initialises two lists: receiver list *R* containing all nodes (up to two-hop distance) believed to have received the packet, and list *N* containing neighbours in need of message. Node sets a timeout waiting period. If node is not in CDS, then it selects longer timeout than node from CDS so that nodes in CDS react first. For each further message copy received, and its own message sent, every node updates *R*, *N* and the timeout. At the end of timeout period, it transmits if *N* is non-empty. The message is memorised until *T* hello messages are received. For each hello message received, *N* is updated. Nodes that are no longer one-hop neighbours are eliminated from the list, while new neighbours, not present in *R*, are added. Regardless of previous decisions, all nodes that so far received broadcast packet check whether new *N* is non-empty. If so, they start fresh timeout. Nodes not in CDS also run timeouts whenever their *N* lists become non-empty.

We assume that there exists a timeout function that assigns waiting time to each node that decides to evaluate the need for retransmission. It can be selected in various ways. For example, it could be a random number in interval (0, *maxtime*] or it could be proportional to $1/|N|$ ($|N|$ is the number of nodes in *N*).

If the list *N* is empty, then *X* cancels the retransmission but retains and later possibly updates *R*. If *N* is non-empty, then *X* selects a waiting timeout. SMSM has two variants. One is that non-CDS nodes do not retransmit. In the unit disk graph model (adopted in [20]), this is not needed to guarantee reception by each node. In a model with realistic physical layer, retransmissions from CDS nodes may not be always received, and therefore backup retransmissions by non-CDS nodes could increase reliability. For this variant, non-CDS nodes also set a waiting timeout. If *X* is further not in CDS, then it adds maximal possible timeout value (e.g. *maxtime*), or a multiple of it, to the timeout. Therefore, it waits for nodes in CDS to decide first. In [13,14], waiting time for non-CDS nodes is proportional to $1 + 1|N|$. This choice could trigger some non-CDS nodes to make decisions before some neighbouring CDS nodes. However, other choices, e.g. $2 + 1/|N|$, reduce such chance but they may increase overall delay if some non-CDS nodes also eventually retransmit. Very large waiting time corresponds to cancellation of retransmission and version from [20] designed for static networks.

We illustrate protocol behaviour on examples. Consider first its behaviour on static networks. In Figure 1, *S* is the source node while *B*, *I*, *G* and *H* are backbone nodes (following generalised DS definition). *G* is the only backbone node to receive the transmission from *S* and thus it will retransmit first. Its backbone neighbours *B* and *H* choose timeout proportional to their number of uncovered neighbours: 1/3 (neighbours *E*, *I* and *J* of *B*) and 1/2 (neighbours *I* and *M* of *H*), respectively. Thus *B* retransmits next. At that moment *H* extends its timeout proportional to $1/1 - 1/3$ (one uncovered neighbour *K* left, 1/3 time elapsed), while *I* sets its timeout in proportion to 1/3 (neighbours *L*, *J*, *K*). Thus *I* retransmits next, and broadcasting completes successfully after retransmissions from three backbone nodes *G*, *B* and *I*. Since the network is static, SBSM behaves as the DS–NES protocol [20], and the first variant (in which non-CDS nodes do not retransmit) of SBSM behaves completely as the DS–NES protocol [20]. In the other variant, non-CDS nodes set a waiting time. In our example, non-CDS node *F* will then also retransmit at the

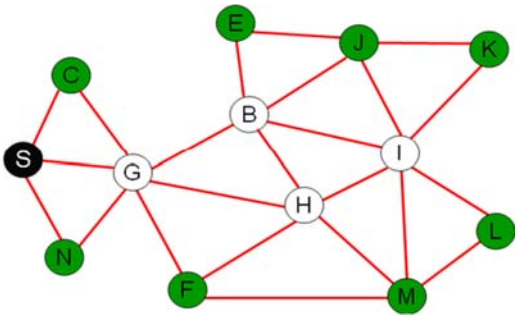

Figure 1. Source *S* and backbone nodes *B*, *I*, *G*, *H* in a static network; nodes *G*, *B* and *I* retransmit.

end of its waiting period, because it did not hear any retransmission covering its neighbour *M*.

Note that scoped flooding [21] (assuming the same waiting timeout function) would require retransmissions also from nodes *H*, *M* and *J* (all except those covered by a single neighbour). Other existing algorithms will perform blind flooding.

Consider example in Figure 2, in which network is initially partitioned into two components. Broadcasting starts from *S*, and nodes *N* and *C* receive the transmission. Other nodes are initially disconnected. After few rounds of 'hello' messages, node *G* was discovered (top of figure) as new neighbour of nodes *C* and *E* making network connected. *G* then retransmits and it serves as a new source of broadcasting for the rest of the network. Node *G* continues to move downward (all other nodes remain static all the time) and in the next hello message becomes neighbour of six nodes (*S*, *C* and *N* on the left and *B*, *H* and *F* on the right). Although all neighbours already got the message, it is not known to *G*. They are all new neighbours for *G*, and *G* will retransmit again (it will retransmit before *S*, *C* and *N* because it has denser neighbourhood). Other nodes will not retransmit again afterward, as they do not have other changes in their neighbourhood and do not find any neighbour in need of that message. On the other hand, in the same scenario, blind flooding would propagate in the left portion of the network, while the other partition never receives the message, since there is no action of 'refreshing' retransmissions after 'hello' message discovered new node. Thus when *G* appears as new neighbour of *C*, *C* does not retransmit in the blind flooding algorithm. Hyper-flooding would 'bridge' the message propagation

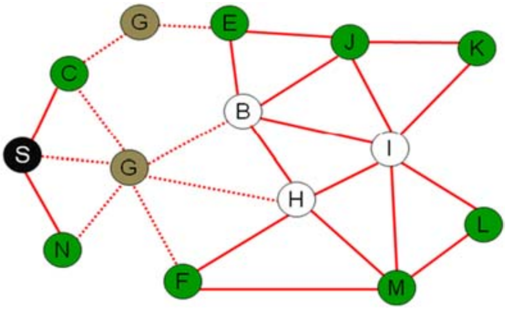

Figure 2. Node *G* merges two partitioned networks and moves inside the network, while other nodes are static. Message initiated by *S* is retransmitted by *G* upon discovery of new neghbours during movement.

but will cause additional retransmissions by all nodes that discover node *G*, from both network parts (thus six more retransmissions).

### 3.2 *Details of SBSM algorithm*

Before source node decides to initiate broadcasting of a message, nodes gain local knowledge and CDS status from the previous hello message round. Nodes react to two events: receiving message intended for broadcasting and sending/receiving hello messages at fixed intervals. These activities proceed until a fixed number *T* of hello messages occurred after the initial source transmission. *T* is a parameter that determines lifetime of message being broadcasted. The remaining lifetime can be carried with the message if synchronisation of lifetime is desirable. It may be beneficial that each node starts its own *T* at the moment of receiving the first copy of message, to reduce message size by not informing neighbours about the remaining lifetime.

When a node *X* receives message for the first time, it initialises two lists, *R* and *N*. List *R* consists of nodes that are believed to have received the same message based on available and current local knowledge of *X*. This includes the sender and its known neighbours. The remaining one-hop neighbours of *X* are included in list *N* (nodes that would benefit from possible transmission of *X*).

Whenever a node *X* receives message for broadcasting, it updates its lists *R* and *N* by adding (eliminating, respectively) all known nodes believed to have received the same message. Note that *R* is being updated even if timeout at node *X* expired or has not even started. If the list *N* becomes empty, then *X* can terminate running timeout and decide not to retransmit. If node *X* is running a timeout that depends on $|N|$, then it updates it. If timeout at *X* expires while its list *N* remains non-empty, then *X* retransmits the message. It also updates its list *R* by moving nodes from *N* there, leaving *N* empty.

During each of *T* upcoming ‘hello’ message rounds, the following updates occur, which may cause additional retransmissions from otherwise terminated process. All nodes that have stored unexpired messages for broadcasting consider new or perhaps very first retransmission (including nodes that have active timeouts). Former one-hop neighbours that are not anymore listed as direct (one-hop) neighbours are eliminated from *N*, since *X* cannot cover them anymore. Note that this may cause a running timeout to terminate (if *N* becomes empty). Newly reported one-hop neighbours are verified if they are in accumulated list *R*. If so, no changes are made with respect to such neighbours. Otherwise they are added to list *N*. This may cause (re)activation of timeouts at *X* if *N* was empty before adding new node.

Hello messages may occur while some nodes were running timeouts already. These nodes behave as if hello message did not occur; timeout is impacted only by possible changes in *N* lists. That is, they continue running the remaining timeouts (more precisely, it is new timeout value reduced by already expired waiting time). Note that ‘refreshed’ or restarted timeouts can also be considered. However, we opted for described version to avoid problems with relatively short hello message intervals that may delay retransmissions for quite long time leading even to its expiration.

## 4. Conclusion and future work

SBSM was simulated in [9] for wireless ad hoc, sensor and actuator networks, and in [13,14] for vehicular networks. It can be extended to a variety of delay tolerant [4] and all-wireless [1] networks. It also works in conjunction with a variety of CDSs definitions [7,23].

SBSM protocol was implemented and compared with the only known competing protocol VO [21] addressing all mobility scenarios. To distinguish between gains made by threshold avoidance and gains made by replacing scoped flooding with CDS–NES approach, Khan et al. [9] added such protocol VO–CDS. It also added protocol CEM [2] in comparison as a benchmark for realistically achievable reliability and upper bound on message overhead. A simplified IEEE 802.11 MAC protocol has been implemented. It is based on *carrier sense multiple access with collision avoidance* scheme. After receiving message, node waits for $W$ message free slots. Afterwards, it retransmits for $p$ consecutive slots. In experiments, $W$ is random integer in [0, 25], while $p = 128$. Khan et al. [9] assumed the unit disk graph model. There is a single broadcasting task in the network. Nodes within fixed transmission radius receive message correctly unless it collides with message from other neighbours. There are no acknowledgements for any received message. Khan et al. [9] simulated two-hop variant of SBSM without using position information. However, it gave to VO and VO–CDS the privilege of using the position information to derive speeds and directions of movements.

It was noted that extremely high speeds may cause inaccurate CDS information. For this reason, Ros et al. [13,14] considered a variant in which nodes include the list of broadcasted messages they are aware of in their hello messages. This, however, causes increase in message length and reduced reception probability. SBSM with positional knowledge can address this issue by considering dynamic CDS (which can change between beacons). Geographic location, speed and direction of movements can be used to estimate how long each link will last [17,19]. When message arrives, node can then estimate current neighbourhood and CDS in a more precise way, and make better retransmission decisions.

One of protocols from Ref. [6] considers short advertisements, sent whenever new node is found, followed by full message if any neighbour responded requesting it. The approach is justified when full message is much longer than short one. The protocol for disseminating short messages alone can be considered as a broadcasting task in itself. We propose two modifications from the protocol [6]. The first one is to introduce waiting periods before sending advertisements. The second is that it is not necessary for all nodes to send advertisements. Only those that are in CDS, and with non-empty $N$ lists, can do so, as in the protocol we proposed. Thus, we can apply our proposed protocol to disseminate short messages. Because of protocol similarity, there was no need to simulate this proposed variant and compare it with the one in Ref. [6]. The cost of sending full messages appears similar in both protocols, because each node will request it at most once, so it is naturally quite restricted and efficient.

Ros et al. [13,14] proposed a broadcast algorithm suitable for a wide range of vehicular scenarios, which only employs local information acquired via periodic beacon messages, containing acknowledgements of the circulated broadcast messages. Each vehicle decides whether it belongs to a CDS. Vehicles in the CDS use a shorter waiting period before possible retransmission. At timeout expiration, a vehicle retransmits if it is aware of at least one neighbour in need of the message. To address intermittent connectivity and appearance of new neighbours, the evaluation timer can be restarted. This AckBSM algorithm resolves propagation at road intersections without any need to even recognise intersections. It is inherently adaptable to different mobility regimes, without the need to classify network or vehicle speeds. In a thorough simulation-based performance evaluation, AckBSM algorithm is shown to provide higher reliability and message efficiency than existing approaches for non-safety applications.

Warning delivery in safety applications was addressed in Ref. [10]. Motivated by the dilemma between solving broadcast storm and providing rapid message delivery, Liu et al. [10] proposed ReC, which exploits 2D geographical information to help nodes autonomously achieve agreement on forwarding strategies. Each forwarding candidate ranks itself and its neighbouring candidates by the distance to the ideal forwarding location, which is the centroid of neighbouring vehicles in need of message. They use ideal location ranking to assign different priority in forwarding among neighbouring nodes and greatly suppress unnecessary retransmission, while enabling best nodes to transmit the packet without waiting.

## Acknowledgements

This work was partially supported by NSERC CRDPJ 386874 - 09 (Reliable and secure QoS routing and transport protocols for mobile ad hoc networks) and NSERC Discovery grant of the first author.

## Notes

1. Email: akhan083@site.uottawa.ca
2. Email: zaguia@site.uottawa.ca

## References

[1] D. Bein and S.Q. Zheng, *An effective algorithm for computing energy efficient broadcasting trees in all-wireless networks*, Ad Hoc Sens. Wirel. Netw. 10(4) (2010), pp. 253–265.

[2] D.E. Cooper, P. Ezhilchelvan, and I. Mitrani, *High coverage broadcasting for mobile ad hoc networks*, IFIP-TC6 Networking Conference, Athens, Greece, May 2004 Lecture Notes in Computer Science 3042 (2004), pp. 100–111.

[3] J. Carle and D. Simplot-Ryl, *Energy efficient area monitoring for sensor networks*, IEEE Comput. Magaz. 37(2) (2004), pp. 40–46.

[4] X. Chen, J. Shen, and J. Wu, *Enhanced delegation forwarding in delay tolerant networks*, Int. J. Parallel Emergent Distrib. Syst. 26(5) (2011), pp. 331–345.

[5] F. Dai and J. Wu, *Distributed dominant pruning in ad hoc networks*, IEEE International Conferance on Communications ICC, Anchorage, Alaska, 2003.

[6] J. Hahner, C. Becker, and K. Rothermel, *A protocol for data dissemination in frequently partitioned mobile ad hoc networks*, IEEE International Symposium on Computers and Communications ISCC, Turkey (July 2003), pp. 633–640.

[7] K. Islam and S.G. Akl, *A local algorithm to compute multiple connected dominating sets in wireless sensor networks*, Int. J. Parallel Emergent Distrib. Syst. 26(5) (2011), pp. 369–380.

[8] F. Ingelrest, D. Simplot-Ryl, H. Guo, and I. Stojmenovic, *Broadcasting in ad hoc and sensor networks*, in *The Handbook of Computer Networks*, 3 Volume Set (Hossein Bidgoli, EIC), Volume II, Chapter 127, Wiley, December 2007.

[9] A.A. Khan, I. Stojmenovic, and N. Zaguia, *Parameterless broadcasting in static to highly mobile wireless ad hoc, sensor and actuator networks*, 22nd IEEE International Conference on Advanced Information Networking and Applications AINA2008, Ginowan, Okinawa, Japan, March 25–28 (2008), pp. 620–627.

[10] J. Liu, Z. Yang, I. Stojmenovic, and Y. Liu, *Receiver consensus: rapid and reliable broadcasting for warning delivery*, Manuscript, 2011.

[11] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, *The broadcast storm problem in a mobile ad hoc network*, Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom), Seattle, USA, August 1999.

[12] W. Peng and X. Lu, *On the reduction of broadcast redundancy in mobile ad hoc networks*, ACM International Symposium on Mobile Ad Hoc Networking and Computing MobiHoc, Boston, USA, 2000.

[13] F.J. Ros, P.M. Ruiz, and I. Stojmenovic, *Reliable and efficient broadcasting invehicular ad hoc networks*, IEEE Vehicular Technology Conference VTC, Barcelona, April 2009, p. 6.

[14] F.J. Ros, P.M. Ruiz, and I. Stojmenovic, *Acknowledgment-based broadcast protocol for reliable and efficient data dissemination in vehicular ad-hoc networks*, IEEE Trans. Mobile Comput. 11(1) (2012), pp. 33–46.

[15] J.A. Shaikh, J. Solano-Gonzalez, I. Stojmenovic, and J. Wu, *New metrics for dominating set based energy efficient activity scheduling in ad hoc networks*, IEEE Conferance on Local Computer Networks/WLN, Bonn, Germany, October 20–24, 2003, pp. 726–735.

[16] D. Simplot-Ryl, I. Stojmenovic, and J. Wu, *Energy efficient backbone construction, broadcasting, and area coverage in sensor networks*, in *Handbook of Sensor Networks: Algorithms and Architectures*, I. Stojmenovic, ed., Wiley, Hoboken, NJ, 2005, pp. 343–379.

[17] I. Stojmenovic, M. Russell, and B. Vukojevic, *Depth first search and location based localized routing and QoS routing in wireless networks*, IEEE International Conference on Parallel Processing, Toronto, August 21–24, 2000, pp. 173–180.

[18] I. Stojmenovic and J. Wu, *Broadcasting and activity scheduling in ad hoc networks*, in *Mobile Ad Hoc Networking*, S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, eds., IEEE Press 2004, Piscataway, NJ, pp. 205–229.

[19] W. Su, S.J. Lee, and M. Gerla, *Mobility prediction and routing in ad hoc wireless networks*, Int. J. Netw. Manage. 6(4) (2001), pp. 351–360.

[20] I. Stojmenovic, M. Seddigh, and J. Zunic, *Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks*, IEEE Trans. Parallel Distr. Syst. 13(1) (2001), pp. 14–25, 15, 11, 2004, 1054-1055.

[21] K. Viswanath and K. Obraczka, *An adaptive approach to group communications in multi-hop* ad hoc *networks*, International Conference on Networking (ICN 2002), Atlanta, 2002; IEEE International Symposium on Computers and Communications ISCC, Taormina, Italy, 2002, pp. 559–566.

[22] J. Wu and H. Li, *A dominating set based routing scheme in ad hoc wireless networks*, Proceedings of the DIAL M, Seattle, August 1999, 7–14, Telecommunication Systems 18(1–2) (2001), pp. 13–36.

[23] N. Zhang, I. Shin, F. Zou, W. Wu, and M.T. Thai, *Construction of virtual backbone with multiple factors constraints in wireless ad-hoc networks*, Ad Hoc Sens. Wirel. Netw. 10(1) (2010), pp. 27–59.