MŰEGYETEM 1782

Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar

Hálózati Rendszerek és Szolgáltatások Tanszék

Iránymenti többes ugrásos adatterjesztés mobil önszerveződő hálózatokban

Location based direction broadcast for mobile self-organized networks

TDK dolgozat

Készítette
Kőkuti András

Konzulens
Dr. Simon Vilmos

2013. október 25.

Contents

Abstract in hungarian

A hagyományos távközlési rendszerek mellett egyre inkább felértékelődik az infrastruktúra nélküli, peer-to-peer kommunikációt lehetővé tevő elosztott hálózatok szerepe, közöttük jelentős szerepet tölt be a mobil önszerveződő hálózatok családja. Önszerveződő hálózati környezetben, ahol nem áll rendelkezésre központi infrastruktúra az átviendő csomagok tárolására és továbbküldésére, nagyon fontos szempont, hogy a többes ugrásos szórt adású (multi-hop broadcast az angol nyelvű szakirodalomban) adatterjesztő algoritmusok milyen hatékonysággal szórják szét az információt a hálózatban. Ezen adatterjesztő algoritmusok komoly felhasználást nyerhetnek olyan helyzetekben, mint például vészhelyzetek (pl.: tűz üt ki az épületben, földrengés ...), szenzorhálózatok nyomon követés céljából (pl.: rajkövetés), forgalom irányításban és még számos más alkalmazás említhető. Annak érdekében, hogy minimalizáljuk a csomópontok energiafelhasználását (amely nagyon szűkös erőforrás egy mobil önszerveződő hálózatban), olyan lokális információkra kell hagyatkoznunk, mint a szomszédos csomópontok aktuális helyzete. Az eszközök helyinformációi alapján kitüntetett irányok menti adást is végezhetünk (üzenet-továbbadási valószínűségekkel menedzselve), amely képes drasztikus mértékben visszaszorítani a hálózatban forgalmazott felesleges üzenetek számát.

Ezen megfontolások alapján egy olyan iránymenti adatterjesztő protokollt kínálok, amely a fentiek alapján irány menti adást tűz ki célul ebben a rendkívül dinamikus környezetben. A protokoll tesztelésre került az általam fejlesztett mobil önszerveződő hálózati szimulátorban. A szakirodalomban publikált három másik algoritmussal hasonlítottam (Distance Adaptive Dissemination, a Ni et al protokoll és a General Probabilistic Broadcast Algorithm) össze különböző paraméterbeállítások mellett. Az eredményekből látható, hogy számos olyan eset van, amely során nem csak teljesítmény hatékonyabban működik a protokoll, hanem még gyorsabb lefedettséget is biztosít.

A kutatás a TÁMOP 4.2.4.A/1-11-1-2012-0001 azonosító számú Nemzeti Kiválóság Program - Hazai hallgatói, illetve kutatói személyi támogatást biztosító rendszer kidolgozása és működtetése országos program című kiemelt projekt keretében zajlott. A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Abstract

In line with traditional communication systems, more and more attention is given to autonomous, self-organized networks with no central infrastructure, based on peer-to-peer communication. Designing multihop broadcast protocols for these networks is a complex problem as the task of these protocols is to disseminate messages in a network effectively while avoiding unnecessary use of resources. The vast majority of these protocols (as those used in the present day Internet) do not use spatial information of the nodes to optimize the bandwidth and channel usage. By increasing the awareness of the nodes and equipping them with their physical location, we can achieve a higher level of autonomous functioning, better performance, and higher level communication primitives, like transmitting in a given direction.

I have designed a novel communication protocol, based on the spatial properties of the system called the Direction Based Handshake Gossiping, which was implemented in my self-organizing network simulator. For performance comparison I have picked three other location based data dissemination protocols from the literature (Distance Adaptive Dissemination, the General Probabilistic Broadcast Algorithm, and Ni et al's location-based scheme). The simulation results show that my solution over-performs the other three protocols in terms of control overhead and number of duplications, which is crucial in self-organized mobile networks, where radio bandwith and energy are usually scarce resources.

This research was supported by the European Union and the State of Hungary, co-financed by the European Social Fund in the framework of TÁMOP 4.2.4. A/-11-1-2012-0001 'National Excellence Program'.

Introduction

Today there are more and more appreciated networks with no infrastructure, which are self-organized and where the communication is based on a peer-to-peer model. A major representative of these networks is the mobile ad hoc network (MANET [1]), which is a self-configuring network, consisting of mobile devices, that can communicate with each other via a radio interface. The distributed and self-configuring nature of these networks, combined with their ease and flexible deployment, make MANETs appealing for a wide range of application scenarios including, e.g., emergency situations, sensor networks for environmental monitoring [2], vehicular ad hoc networks [3], and many others [4, 5].

The common denominator behind all these application scenarios is the fully distributed nature of the network infrastructure supporting them, together with the support of nodes mobility. In particular, this last characteristic is reflected in a network topology that can change over time, depending on the density and mobility of nodes.

Many ad hoc applications rely on the existence of a broadcast medium for the dissemination of control information. The inexperienced first implementation of this was flooding: every node repeats the message after it is first received. However it was recognized soon after, that this is far from optimal, and collisions in the media can lead to serious congestion and loss of packets. To solve this problem many efficient broadcast techniques were designed, that take into account some information about their surroundings, instead of blindly repeating every packet.

These algorithms differ in their assumptions about the environment (like assumption of a connected or disconnected network) and in the information available for decision (availability of Global Positioning System (GPS) or other positioning techniques). The central problem of broadcast algorithms is to decide when and who should retransmit messages. Nodes have to forward packets so the message reaches every part of the network, however the performance relies heavily on the set of nodes that do this. When nodes decide whether to retransmit or not, they actually decide whether they are part of the forwarding set. Too many retransmissions cause collisions and waste the network bandwidth, but choosing the smallest forwarding set is not easy because a global view of the network is not available, and local information gets obsolete very quickly if the velocity of nodes is high. There is also a risk if the number of forwarding nodes is too small, as the message may not reach every node.

Only few of the existing multihop broadcast solutions utilize the up to date physical location and distance information of the nodes from the environment, which can provide a more reliable picture on the current state of the local topology (which changes dynamically). By obtaining an additional spatial information, the decisions of the given mobile nodes about when and who should broadcast messages will be further improved.

My goal is to give an overview of the existing location based multihop protocols, pointing out that novel location based communication protocols are needed to optimize the networking load if using them in real environment. And I show a solution (called Direction Based Handshake Gossiping), which is actually based on GPS coordinates, that could communicate more effectively than three others from the scientific literature, namely the Distance Adaptive Dissemination, the General Probabilistic Broadcast Algorithm, and Ni et al's location-based scheme when observing the number of the duplications and the coverage time.

1. Chapter

Multihop broadcast protocols

The inexperienced first implementation was the blind flooding[6] algorithm. The operation of BF algorithm is very simple, once a node receive a message, it immediately rebroadcasts it. We might think this is a good solution for data distribution, because it is very simplistic, and covers all the nodes (assuming that there is no separation), but it was realized very soon, that this is far from optimal. The effect of frequent packet repetition may result in collisions in the medium and can lead to serious congestions, such as broadcast storms[7].

The Counter Based Method, originally introduced in [8] is one of the first controlled broadcast methods . It is based on a simple observation, that if a duplicate of a packet is received, then the probability of reaching any new node is low. To exploit this idea, the nodes do not immediately transmit on the receipt of a packet, but instead they wait for a random time, which is called Random Assessment Delay (RAD). If a duplicate is received during the RAD a counter is increased. If the counter reaches a threshold before the RAD expires, the node cancels the transmission. The original method has different adaptive versions [9], which try to adapt the length of the RAD and the threshold of the duplicate counter to the current network conditions.

Another broadcast method is the Gossiping algorithm [8], where every node broadcasts the heard message with a predefined probability. The optimal probability can be calculated off-line, or can be learned adaptively. Some of these adaptive versions are covered in [10]. While the Counter Based method is a fine example of a simple heuristic based deterministic algorithm, Gossiping is an example of the simple heuristic based stochastic methods. Another problem can be, that while the optimal retransmission probability can be calculated off line, it heavily relies on the parameters of the environment. To overcome this limitation there are adaptive versions of the basic methods, like Hypergossiping.

Hypergossiping [11] is specifically designed for partitioned networks, where nodes are mobile, and partitions join and split from time to time. It is an advanced version of the Gossiping algorithm, extended by neighbor information and partition join detection. The algorithm uses a simple adaptive gossiping strategy for in-partition forwarding, but rebroadcasts some of the packets if it detects a join with another partition. The join detection is based on the simple heuristic, that the nodes in the same partition received the same messages recently. Every node maintains a list, called LBR (Last Broadcasts Received), of the recently broadcast messages. They send HELLO messages periodically, to indicate their presence. When a new node is detected, one of the nodes includes its LBR in the next HELLO message. When the other node receives this LBR, it compares with its own LBR. If the overlap between the LBR of two nodes is smaller than a threshold, then the node is considered coming from another partition, so a new message is sent, called BR (Broadcasts Received), which contains the list of messages that the node already received. From this the other node knows that a partition join happened, and rebroadcasts all the messages that were not inside the other nodes BR. After this rebroadcast, dissemination proceeds using adaptive gossiping.

A fine example of a self-pruning algorithms is the Scalable Broadcast Algorithm (SBA) [12]. It requires a 2-hop neighbor information and the last sender ID in the broadcast packet. When a node v receives a broadcast packet from a node u it excludes the neighbors of u, $N(u)$ from the set of its own neighbors $N(v)$. The resulting set $B = N(v) - N(u)$ is the set of the potentially interested nodes. If $|B| > 0$ then the node will start a Random Assessment Delay (RAD). The maximum RAD is calculated by the $\frac{d_v}{d_{max}} * T_{max}$ formula, where $d_v = |N(v)|$ and d_{max} is the degree of the node with the largest degree in $N(v)$, and T_{max} controls the length of the RAD. Nodes choose the time of transmission uniformly from this interval. This ensures that nodes with a higher degree often broadcast packets before nodes with fewer neighbors

A quite different approach from the algorithms discussed so far is the IOBIO algorithm[13]. It is a variation of the SPIN [14] dissemination protocol. It uses a simple 3-stage handshake to discover neighbors that are interested in one of the carried messages. The goal of the protocol is to reduce the unnecessary load of neighboring nodes by duplicate or unneeded data ("spamming"). There are three IOBIO message types that are used by the protocol. The ADV (Advertisement) messages are sent periodically, and they contain the list of messages that the sending node has. Neighbor nodes indicate their interest in the advertised messages by sending a REQ (Request) packet. In response to the REQ, the originator node sends the required DATA packets. The transmission of a REQ after an ADV is not done immediately, but after a random delay. During this delay, the nodes listen to each other, and they only request packets that were not requested before.

1.1. Location based protocols

Location based multihop protocols use spatial information to make their decision about message broadcasting, in additional to the classical neighborhood and topology information. In most of the cases this means that the device should have a Global Positioning System (GPS) or another positioning technique to acquire this information. These methods use HELLO messages, just like the neighbor information algorithms do, but they collect the location of the neighbors as well. There are also algorithms that need to know only the distance to their neighbors like the Distance Adaptive Dissemination (DAD)[15] algorithm, which may be measured by signal power. In this case, the use of HELLO messages may not be necessary, as it uses distance information instead of exact positions. The authors propose a scheme that chooses forward nodes according to their distance, using the signal strength as a measure for distance. The goal of the algorithm is to try to get the outermost neighbors of a node rebroadcast, thus minimizing overlap of transmission ranges. It uses 1-hop neighbor information and records signal levels from the neighbor nodes. The authors propose two variants called DAD-NUM and DAD-PER. DAD-NUM chooses a signal strength S_{thres} so that there are k number of neighbors that have transmitted with a signal strength lower than S_{thres}. On arrival of a new packet, the node checks if the signal strength is greater of S_{thresh} or not. If it is smaller then the node rebroadcasts. DAD-PER is very similar, but instead of finding the k farthest nodes it chooses p percent of them.

Another example for location based protocols is the Optimized Flooding Protocol (OFP) which is a deterministic dissemination algorithm, that uses a geometric approach instead of the usual graph theory solutions. The algorithm tries to cover the 2D space efficiently with R radius circles. I do not detail the algorithm here, mostly because I do not think circles are good approximations of transmission ranges in urban and in-building environments. Details can be found in [16].

Stojmenovic's method[17] also uses position information, it is a variant of Wu and Li's algorithm[18]. There are two important improvements over the original algorithm: it uses 1-hop information coupled with position information to implement the marking process and rules 1, 2. The other difference is that it also implements a random backoff scheme, similar to the Scalable Broadcast Algorithm (SBA). The nodes do not broadcast immediately, but rather wait for a random time. If a node v hears a transmission during this interval from a node u then he removes $N(u)$, the neighbors of u, from its own neighbor set $N(v)$.

Two others are introduced by Ni et al[8]. The first one tries to use the relative distance between hosts to make the decision. Suppose host H heard a broadcast message from S for the first time. If the distance, say d, between H and S is very small, there is little additional coverage H's rebroadcast can provide. If d is larger, the additional coverage will be larger. In the extreme case, if $d = 0$, the additional coverage is 0 too. If H has heard the same message several times, let d_{min} be the distance to the nearest host from which the same message is heard. If d_{min} is smaller than some distance threshold D, the rebroadcast transmission of H is canceled. The other one uses positioning devices such as GPS (Global Positioning System) receivers, to estimate the additional coverage more precisely. Let a host's location be $(0, 0)$, and suppose the host has received the same broadcast message from k hosts located at $(x_1, y_1), (x_2, y_2), ..., (x_k, y_k)$ positions. So the additional area that can be covered by the host's rebroadcast can be calculated. Let $AC((x_1, y_1), (x_2, y_2), ..., (x_k, y_k))$ denote the additional coverage divided by πr^2. This value should be compared to a predefined coverage threshold A $(0 < A < 0.61)$ to determine whether the receiving host should rebroadcast or not.

Min-Te Sun et al.[19] identified two primary design issues, namely the defer time generation and the redundant message classification, which result in a broadcast protocol that enjoys flooding's high reachability and the bandwidth efficiency of the non-flooding schemes'. Let us consider the issue of setting defer times. Nodes with a larger defer time are scheduled to retransmit a message later than those with a smaller defer time. Unless a node decides on redundant/non-redundant retransmission regardless of other nodes' retransmissions, the node with a larger defer time is more likely to find its retransmission redundant than the node with a smaller defer time. Since the purpose of a retransmission is to forward the message to more nodes, it seems plausible to let a node to cover new areas by retransmitting the message earlier than the node covering less of the still uncovered area. Thus, instead of randomly choosing a defer time, they have proposed the Distance-based Defer Time Scheme. When S receives a broadcast message from N, it sets the defer time to a value inversely proportional to a power of $||\overrightarrow{SN}||$. That is, $defer\,time = Max\_Defer\_Time * (R^{\epsilon} - ||\overrightarrow{SN}||^{\epsilon})/R^{\epsilon}$. The second issue is how to identify a redundant retransmission. Since broadcast service requires high reachability, a retransmission should not be easily discarded unless the coverage area is known to be completely covered. To achieve the maximal reachability and fast computation, they propose the Angle-based Scheme to identify redundant retransmissions . They calculate two values: α and β form the intersection of covered circles. So if during its defer period node S receives the retransmissions of a same message from a number of neighbors with cover ranges $[\alpha_1, \beta_1], ..., [\alpha_k, \beta_k]$ and if $\cup_i[\alpha_i, \beta_i] = [0, 360]$, then S will not further retransmit the message.

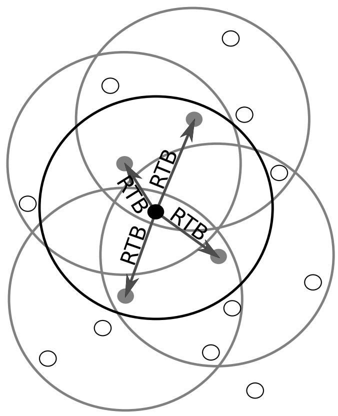
The Border Retransmission Based Probabilistic Flooding (BRBPF) was presented in [20]. They have observed that the distance between two nodes with a full duplex communication can be evaluated by comparing their neighbor lists. When two nodes *src* and *dist* can contact each other, the union of their communication areas (Z_{src} and Z_{dest}) can be partitioned in three zones:

- $Z_a = Z_{src} \cap \overline{Z_{dest}}$: the communication area covered only by *src*,
- $Z_b = \overline{Z_{src}} \cap Z_{dest}$: the communication area covered only by *dest*,
- $Z_c = Z_{src} \cap Z_{dest}$: the communication area covered by both *src* and *dest*.

They define a ratio: $\mu = \frac{N_b}{N_a+N_c}$, where N_i denotes the number of neighbors in zone Z_i. Based on this parameter the *dest* determinate its own probability of sending p whit the next formula: $p = \frac{A-\alpha}{M^{\alpha}}\mu^{\alpha} + \alpha$, where α, A and M are fine-tuning parameters.

1.2. The 3-way handshaking

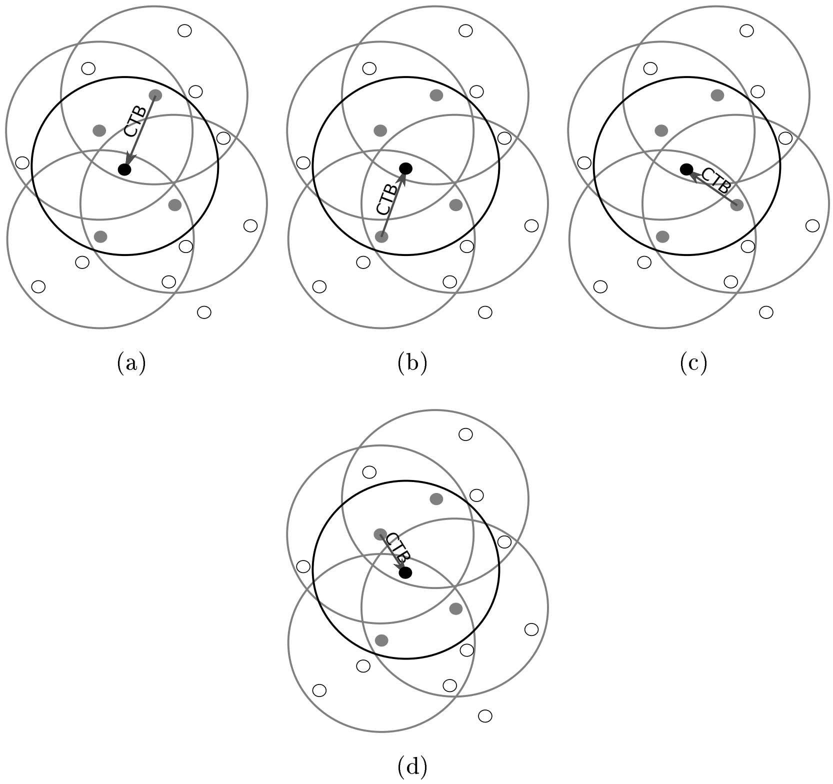
There is a group of adaptive multihop broadcast protocols: the Distance Based Handshake Gossiping, the Valency Based Handshake Gossiping, the Average Valency Based Handshake Gossiping and the Adaptive Handshake Gossiping [21], which all assign broadcasting probabilities to the mobile nodes, determining them from network parameters such as degrees of the nodes or distance of the nodes from each other, by sharing the current physical location of the nodes. The protocols differs from each other only in the methods of calculating the message forwarding probabilities. It is assumed that the devices have the knowledge of their actual geographical positions (with the help of GPS transceivers or through other positioning techniques, which is a realistic assumption with today's smart phones and notebooks). All three protocols consists from the same phases. The first phase(Figure 1.1) is called RTB [22] (Ready-To-Broadcast), which is similar to the RTS phase in CSMA/CA [23]. The device, which has got a data message to disseminate, broadcasts an RTB signal message in its radio range. It places in the packet its own unique identifier (for example MAC address) and in the case of multi-message communication (where mobile nodes disseminate different kind of data messages) the message type and identifier.



1.1. figure. *The first phase*

In Figure 1.1 the black point marks the sender node, the grays are the nodes in the senders range, while the whites depict the other devices in vicinity. The transmitting device, marked with black colour, is entering the first phase by disseminating an RTB packet. The neighboring nodes (gray colour) in its radio range receive the message due to the radio broadcast.

In the next phase all of the nodes, which received the RTB message, start a random timer from the [0;t_{CTB}] range. This timer ensures that the reply CTB (Clear-To-Broadcast) messages do not cause broadcast storms, and this way avoiding collisions. The message includes the CTB sender's and the "recipient"'s (which sent the RTB packet int the previous phase) unique identifier. With the returned identifiers, the central (marked with black colour in Figure 1.1) node can decide, if it is the recipient of the packet or another node at a 2-hop distance. The message also contains the geographical coordinates of the CTB sender (by using the GPS transceiver or other positioning technique), the number of its neighbors, namely its degree. To detect their neighbors, the nodes periodically distribute HELLO signal messages. All devices transmit a HELLO message at appropriate intervals for monitoring their own environment. If HELLO is received, the node updates the list of its neighbors. So the received HELLO messages help to determine the actual degree of the given node. In addition to these, CTB messages also include the identifiers of the messages (getting it from the RTB packet), which are needed by the CTB sender node.



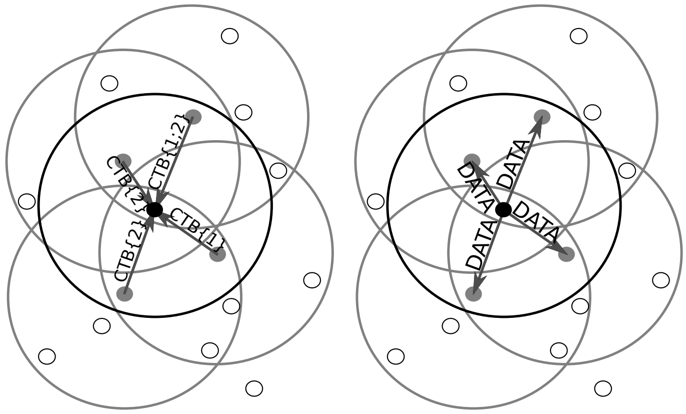
(a) (b) (c) (d)

1.2. figure. *The second phase*

First, as shown in Figure 1.2, the node in the (a) subfigure gets the opportunity to broadcast the CTB message, because its timer expired earlier than the other nodes which also started a timer upon receiving the RTB message. Then the nodes, shown in (b), (c) and (d) subfigure will transmit one after other, determined by their timers. Meanwhile the central node is waiting for a t_{DATA} time, which is calculated as the sum of the t_{CTB} and the maximum propagation time.

The last phase is responsible for the data message dissemination, the first two phases use only signal messages. The central node starts the data message transmission (the node which sent out previously the RTB packet) after its t_{DATA} timer expired. First, it summarizes based on its 1-hop neighbor list, which nodes are currently within its radio range, and to these IDs (which can be found in the reply CTB messages) assigns the degrees and the coordinates. After it calculates the retransmission probability for each adjacent device. From the CTB messages the central node can determine which data messages need to be disseminated, by calculating the union of the requested data messages.

Figure 1.3 shows an example of how is the central node determines which data messages



(a) An example for requesting the data messages (b) The DATA message dissemination

1.3. figure. *The third phase*

should be disseminated. The adjacent devices, marked by gray colour, put the requested data message IDs in the CTB packets (in braces). Based on these identifiers, the central node calculates which messages (here, the messages with ID 1 and 2), need to be broadcasted. Thus, all parameters are known (messages, probabilities) in order to complete the information dissemination process.

The DATA message's header contains the sender (same as RTB sender) unique identifier and the calculated message forwarding probabilities, the payload is comprised of the data message. The sender's identifier is necessary as there may be other 2-hop distance nodes, which are in the same transmission phase. Therefore the common neighboring nodes can distinguish which node's DATA packet have been received. From the rest of the header every adjacent node get its message forwarding probability, as it is assigned to the nodes ID. As I have mentioned, the protocols only differ in their methods of the probability calculation, so in the next subsections the concrete protocols will be described.

1.3. The Distance Based Handshake Gossiping (DBHG)

For the Distance Based Handshake Gossiping protocol the degree of the nodes is not needed, so that can be omitted from the CTB message. Moreover, the periodically sent topology discovery signal messages (HELLO) are unnecessary. For calculating the probabilities only the geographical coordinates are needed, which can be found in the CTB message. The calculation is based on the nodes distances, hence the name, which is defined using the following formula.

$$d_i = \sqrt{(\triangle x)^2 + (\triangle y)^2 + (\triangle z)^2} \tag{1.1}$$

$$d_i = \sqrt{(x_0 - x_i)^2 + (y_0 - y_i)^2 + (z_0 - z_i)^2} \tag{1.2}$$

The x_0, y_0, z_0 parameters are the central node's, while x_i, y_i, z_i are the i. adjacent device's coordinates. The central node selects the largest distance, ie the highest value, it will be d_{max}.

$$d_{max} = \max_{\forall i} \{d_i\} \tag{1.3}$$

The protocol, based on these parameters, assigns the following message forwarding probability to the i. neighboring node:

$$p_i = \frac{d_i}{d_{max}} \tag{1.4}$$

This means, being further from the central node, the message forwarding probability will get higher for the given adjacent node. This way the duplication caused by closer nodes can be avoided, moreover the area covered by more distant nodes is likelier to contain previously not covered nodes [24]. This means that the furthest node's (which is the furthest from the central node) associated probability variable will be 1, so it will initiate the retransmission with 100%. Thus in the set of the neighboring nodes there will be at least one node (the furthest), which will continue the information dissemination.

1.4. The Valency Based Handshake Gossiping (VBHG)

A shortcoming of the previous protocol is that it does not consider the degree of the nodes when assigning the message forwarding probabilities to the neighboring nodes. It may happen, that the furthest adjacent device has no other neighbors, than the initial transmitter, nevertheless it has got 100% forwarding probability because of its position. Furthermore the protocol still does not "penalize" the nearby nodes enough. In order to avoid unnecessary duplication of messages, the nearby nodes with a high degree need to get a low chance for the retransmission, while the further nodes should have a higher probability. The Valency Based Handshake Gossiping is trying to achieve this goal. Of course, in this case it is necessary to distribute periodically the topology discovery signal messages, and some additional parameters are needed. The d_i, d_{max} variables are identical to those used in the DBHG protocol. The $d_{average}$ indicates the average value of the distances, calculated as an arithmetic mean of the central node-neighbors distances

$$d_{average} = \frac{1}{n} * \sum_{i=1}^{n} d_i \tag{1.5}$$

where n is the number of adjacent nodes, while f_i is the i. neighboring node's degree. The neighbors can be distinguished by the unique identifier in the CTB message. Similar to d_{max} an f_{max} is needed for VBHG, to determinate the probabilities.

$$f_{max} = \max_{\forall i} \{f_i\} \tag{1.6}$$

The probability is calculated in the following way for the VBHG protocol:

$$p_i = \begin{cases} \frac{d_i}{d_{max}} - \frac{d_i}{d_{max}} * \frac{f_i}{f_{max}} \text{ , } if\; d_i < d_{average} \\ \\ \frac{d_i}{d_{max}} + \frac{d_{max}-d_i}{d_{max}} * \frac{f_i}{f_{max}} \text{ , } if\; d_i \geq d_{average} \end{cases} \tag{1.7}$$

As equation 1.7 shows the calculation is comprised from two disjoint cases. The upper equation is used, when the adjacent device is closer to the transmitter than the average distance. In this case, the goal is to assign a lower retransmission probability, because this node's radio range is already largely covered by the central node, so the rebroadcast would cause many unnecessary duplications. On the other hand, when the device is far form the transmitter, more likely it covers new, still not covered nodes, the lower equation is utilized for this case.

1.5. The Average Valency Based Handshake Gossiping (AVBHG)

The Average Valency Based Handshake Gossiping is based on the experience provided by the two previous versions. It uses one more parameter, the $f_{average}$ indicates the average valency of the network:

$$f_{average} = \frac{1}{n} * \sum_{i=1}^{n} f_i \tag{1.8}$$

where n is the number of the neighbors for the given node. Consequently, the probability is determined by:

$$p_i = \begin{cases} \frac{d_i}{d_{max}} + \frac{d_{max}-d_i}{d_{max}} * \frac{f_i}{f_{max}} & if\, f_i \geq f_{average} \\ & and\, d_i \geq d_{average} \\ \frac{d_i}{d_{max}} - \frac{d_{max}-d_i}{d_{max}} * \frac{f_i}{f_{max}} & if\, f_i \geq f_{average} \\ & and\, d_i < d_{average} \\ \frac{d_i}{d_{max}} & else \end{cases} \tag{1.9}$$

As it can be seen, the probability calculation is a combination of the previous two protocol probability assignments, therefore those nodes get lower probabilities for retransmission which are closer to the central node than the average distance, and at the same time they have a high valency value.

1.6. The Adaptive Handshake Gossiping (AHG)

All three versions of the protocols should be installed on the mobile device, as they differ only in the way of calculating the forwarding probability, therefore no additional memory or processing cost is introduced. Thus, the nodes can adaptively change the function of the probability assignment, depending on which parameter(s) needed to be optimized. If the system requires a rapid dissemination, the device would select the DBHG version. However, if it is of high cost to deliver a message, then they should switch to VBHG, while AVBHG beeing a trade-off between the two aspects.

Based on these an adaptive handshake gossiping protocol was designed, consisting of the DBHG and the VBHG protocols. If the device senses, that it is in a dense environment (this information can be obtained from the node's own local database, for example its valency value), it can change to the VBHG version, as this version causes the smallest number of duplications. Furthermore it performs nearly as fast as the DBHG in a dense environment In a sparse environment, the DBHG protocol should be used, because of its benefits. Thus after the second phase, when the CTB messages are received, the transmitting node should adapt the function of the probability assignment (utilizing DBHG or VBHG) depending on its own local parameters.

$$p_i = \begin{cases} Equation\ (1.4)\,,\ if\ f_{transmitter} < f_{average} \\ \\ Equation\ (1.7)\,,\ if\ f_{transmitter} \geq f_{average} \end{cases} \tag{1.10}$$

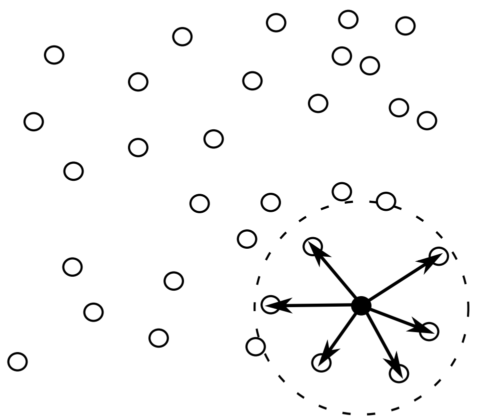
where $f_{transmitter}$ is the valency value of the currently transmitting node, while the $f_{average}$ is the same parameter, which was used in the previous protocols.

2. Chapter

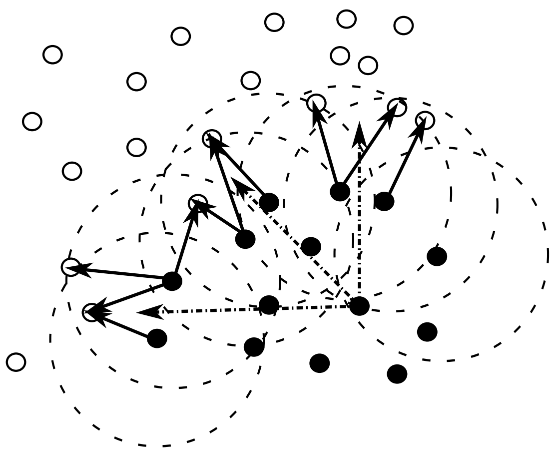
The novel protocol

2.1. Motivation

Huge number of multihop broadcast protocols (which are introduced in Chapter 1) can be found in scientific literature, which are all targeting the goal to communicate effectively in this specific environment. Almost all of them are using signal messages (since it is the only way to get some information about the neighbor nodes) to reduce the enormous number of unnecessarily sent messages, to avoid packet collisions or just to improve the quality of the service in the network. However it can be realized that the protocols performance highly relies on the gathered local information (which can be obtained from the signal messages, unfortunately producing more overhead).



(a) The first broadcast from the source node.



(b) After a few rebroadcasts a message propagation front line is formed (dashed arrows).

2.1. figure. *The data dissemination process*

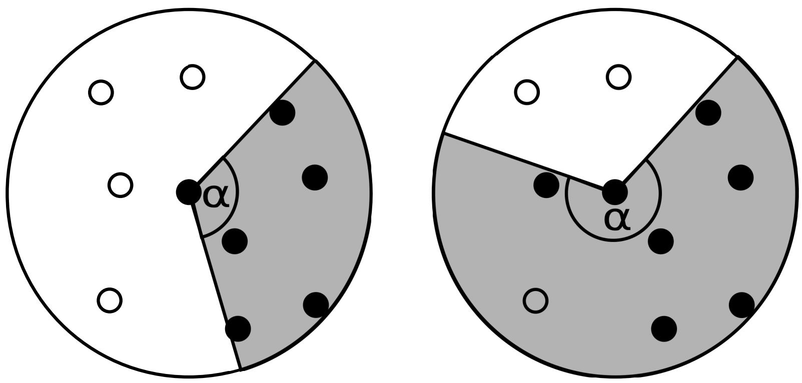
The current communication protocols for autonomous mobile networks do not enable the mobile nodes to choose their direction of message propagation by influencing the direction of the transmission using local interactions. The novel protocol presented by me is based on the simple assumption, that all the messages circulating among the mobile nodes are originated from a given source node. In this scenario it would be desirable, for the source node to be able to manipulate the direction of the message propagation in the network, if needed (shown in Figure 2.1(a)).

Thus my novel protocol's goal is to maintain the right direction of data disseminations, by using probability assignments to limit the message retransmissions. Obviously for this environment there is no solution which is the best in all performance indicators, as we need to have a trade off between efficiency (like number of the duplications) and simplicity (like spamming the messages all the time). This way, the simplier the protocol is, giving us less overhead, faster the coverage. But of course it produces the highest number of duplications and wastes too much energy at the mobile nodes.

2.2. The Direction Based Handshake Gossiping

The contribution is to enable the rebroadcasting nodes (which become the new source nodes) to propagate their messages in a chosen direction, by giving a higher retransmission probability to nodes in that direction (see Figure 2.1(b)). With this kind of assignment of retransmission probabilities, we are able to exclude many unnecessary duplications, and force the messages to follow a given direction from the sender nodes, which can be very useful in various application cases (like if we want to cover only portions of the network or to control group of nodes moving in a given direction). Also making nodes aware of their physical location is crucial to enable the network to implement more advanced communication primitives, such as transmitting a message in a given direction. This way the local interactions (probability assignments) will contribute to a global behavior of the network.

The Direction Based Handshake Gossiping protocol consists of two steps: the first one is the direction calculation, while the second one performs the probability assignment based on the first step's result. In the first step we need to determine whether the sender nodes define a clear transmission direction, or they just want to uniformly distribute the messages in the network. In the latter case, the noves solution will use the probability assignment of my earlier developed protocol, the Valency Based Handshake Gossiping, while in the first scenario the direction vector will need to be calculated. The initial sender node is given by its coordinates, together with its radio range, already covering a group of nodes in its 1-hop neighborhood. We want to calculate the smallest circular sector, containing all previously covered nodes (which do not need to receive the message again, as it will be a duplication). As the circular sector area is determined by its angle and the radius, and the latter is a constant value, only the angle will need to be calculated.



(a) The covered nodes define a propagation direction.

(b) Randomly distributed covered node is in the central node's radio range.

2.2. figure. *Node distributions taken into account by the adaptive mechanism in my solution.*

First we need to define a reference vector, with which we can calculate the angle to the covered nodes. Let this vector be the base vector $(1, 0)$, therefore the simplest way to calculate these angles will be the scalar product. In our case we need to modify the standard equations: let v_i be the vector to the i^{th} covered node, then

$$\alpha_i = \begin{cases} arccos(\frac{v_i * i}{|v_i| * |i|}) & , when\, y_i \geq 0 \\ 2\pi - arccos(\frac{v_i * i}{|v_i| * |i|}) & , else \end{cases} \tag{2.1}$$

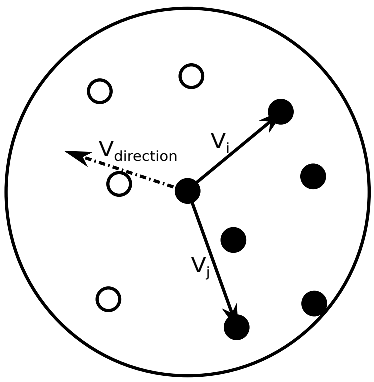
With this addition, the angle values will be in the $[0; 2\pi]$ domain, and not in $[0; \pi]$, as this is essential to a proper calculation. We can determine the required angle with the help of the following theorem.

Theorem: The solution will be given by adjacent vectors enclosing the smallest angle.

Proof: The theorem can be proved by the use of two lemmas. The first one states that the sought circle sector is determined by two adjacent vectors. This lemma can be trivially justified: Two vectors divide the whole circle in two circle sectors and these vectors are adjacent if and only if the two sectors contain only covered or uncovered nodes. The second lemma says that from the set of these adjacent pairs, the desired solution would be the one, where the angle of the circle sector, where the already covered nodes are, is the smallest. The angles can be calculated by the differences of the vector's angle (with some caution, because the sequence plays a major role in this operation), which we defined previously.

After the first step, α and the adjacent vectors are known, and based on these data we can proceed with the protocol's next step. My solution uses an adaptive mode of operation, therefore if $\alpha > \alpha_{threshold}$ then it uses the probability assignment of the Valency Based Handshake Gossiping protocol. In this case there is no distinctive transmission direction, therefore it is better to use an existing broadcast protocol.

If $\alpha \leq \alpha_{threshold}$, there is an emphasised direction of transmission. This means that by proper selection of $\alpha_{threshold}$, one can tune the message propagation directions in the network. The direction vector can be determined from the circle sector in the following way:



2.3. figure. *Calculation of the* $v_{direction}$ *vector.*

$$v'_{direction} = \begin{cases} \frac{-v_i}{|v_i|} & -\frac{v_j}{|v_j|},\ if\, \alpha \leq \pi \\ \frac{v_i}{|v_i|} & +\frac{v_j}{|v_j|},\ else \end{cases} \quad (2.2)$$

$$v_{direction} = \frac{v_{direction'}}{|v'_{direction}|} \quad (2.3)$$

By utilizing this direction vector, the probability calculation is simple, the retransmission probability (p_i) for a given uncovered mobile node will be the scalar product of $v_{direction}$ and the vector pointing to the uncovered node. Let v'_i point to node i, which is currently not covered, then

$$p_i = max(v'_i * v_{direction}, 0) \quad (2.4)$$

If this equation is examined more thoroughly, it can be noticed that the variable depends also on the length of the vectors:

$$v'_i * v_{direction} = |v'_i| * |v_{direction}| * cos(\theta) = |v'_i| * cos(\theta) \quad (2.5)$$

So the probability assignment of my protocol is based not only on the transmission direction, but also on the distance of the nodes (like in DBHG).

2.3. Performance analysis

2.3.1. The simulator

For measuring the performance of the protocols a self-organizing network simulator was implemented in C++, in which the protocols can operate under similar conditions as in a real MANET. The adjustable parameters of the simulator are introduced in Table 2.1

Parameters name	Description
N[pc]	Number of the nodes
R[e]	Transmission range
R'[e]	Movement range
A[e^2]	Simulation area
Pid	Simulated protocol's ID
t[pc]	Number of tests

2.1. table. *The parameter list*

A modified version of the random waypoint mobility model[25] was used, where the waypoint should be chosen in the area of node's movement range.

The simulator default settings could be seen below in Table 2.2.

Parameters name	Value
N[pc]	500,600,800
R[e]	0,2e
R'[e]	0,1e
A[e^2]	4e^2
Pid	DAD, GEN, Ni's, DiBHG
t[pc]	100

2.2. table. *Default settings of the simulator*

When there is only one type of data message to be disseminated in the system, we could assume that all of the network devices need to receive it (e.g. in an emergency case). It can be seen from this table that simulations were run for 500, 600 and 800 nodes. The results were calculated based on averaging 100 tests to deal with variations from test to test.

2.3.2. The performance indicators

One of the most important performance indicators of data dissemination in a mobile ad hoc network is the **number of duplications**, as it effects the resource usage and efficiency of the whole network (speed of the data dissemination, energy consumption, coverage). The number of duplications specify how many unnecessarily received messages exist in the system (the nodes already own them, so resources are consumed to "spam" other nodes). It can be calculated as a difference between the requested and the received messages in the system:

$$dupl = \sum_{i=1}^{N} i_{message\_received} - \sum_{i=1}^{N} i_{message\_requested} \tag{2.6}$$

A high number of duplications in the system would mean inefficient utilization of the resources, so optimally this metric would be 0, meaning that all devices have received only the messages requested by them.

Another important performance metric is **coverage**, the percentage of requested messages received by the nodes. Let $message\_received'$, be the number of messages received without duplications, then

$$coverage = \frac{\sum_{i=1}^{N} i_{message\_received'}}{\sum_{i=1}^{N} i_{message\_requested}} * 100\% \tag{2.7}$$

The third performance indicator is the **number of sent messages**. It shows how many broadcasts occurred in the network. It can be calculated as the sum of the number of broadcast transmissions of the nodes.

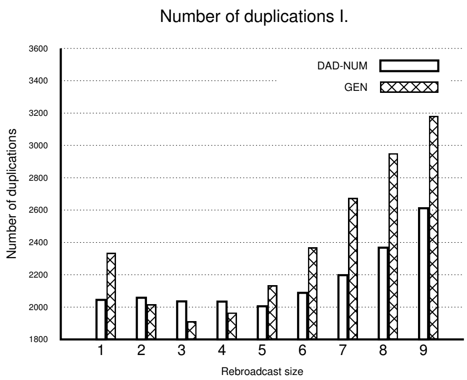
The fourth performance metric is **coverage time**, which is the time needed for a given protocol to achieve a fixed level of coverage of the nodes. An idealistic protocol would achieve a high coverage in the shortest time possible, however as it can be seen that the performance indicators are contradicting each other, so there will always be a tradeoff between them. The „goodness" of a data dissemination protocol is always dependent on the application. Different protocols prioritize the performance metrics differently. For example, the blind flood protocol delivers the best coverage in the shortest time, but it is an extremely wasteful solution regarding resource usage, collisions and number of duplications in the system.

The protocols performance was evaluated by the help of the above described parameters fixing the coverage level at 95%. Thus, the simulation will stop when the specified protocol reaches 95% coverage.

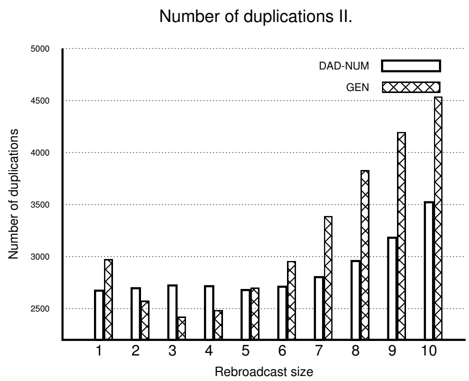
2.3.3. Results

My novel protocol, the Direction Based Handshake Gossiping (DiBHG) was compared to the Distance Adaptive Dissemination (DAD) [15], the General Probabilistic Broadcast algorithm [8], and to Ni et al.'s algorithm, which were all described in Section 1.1. (The previous 4 protocols, the DBHG, VBHG, AVBHG and the AHG, have been already presented in [21]) These protocols try to optimize the resource usage in the system not only by using only locally available information about their neighbors, but also by utilizing available spatial information. Performance measurements of the described protocols were published in several papers, proving that they disseminate information in self-organized mobile networks very efficiently, over-performing most of the available data dissemination algorithms, that can be found in literature.

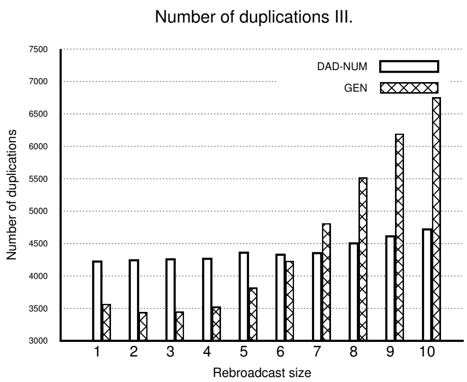
I have implemented the three reference protocols in my mobile self-organized network simulator, and determined the value of the k parameter (the target rebroadcast size), for which the Distance Adaptive Dissemination (DAD-NUM) and the General Probabilistic Broadcast Protocol (GEN) performs the best, regarding the number of duplications, by simulations. Choosing the optimal k value for further performance evaluations ensures a fair comparison between the tested protocols. The number of duplications for the best k values were measured for all three mobile node densities and summarized in Table 2.3.



(a) In a network with 500 nodes



(b) In a network with 600 nodes



(c) In a network with 800 nodes

2.4. figure. *Duplication overhead of the DAD-NUM and GEN protocols*

Figure 2.4 shows that for all three different node density scenarios, the GEN protocol over-performed the DAD-NUM solution regarding the number of duplications only for small k values. By increasing the k parameter, the DAD-NUM always outperforms the GEN protocol in terms of number of duplications. I have selected the best k parameter values for all three scenarios and summarized them in Table 2.3.

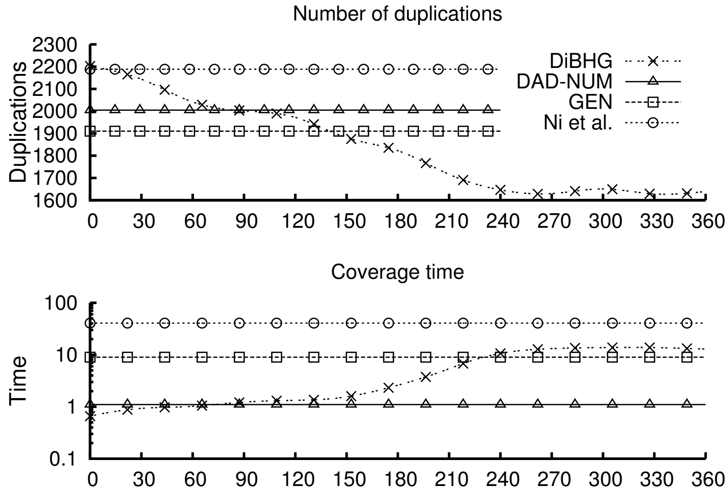
Number of nodes	Protocols	Rebroadcast sizes	Number of duplications
500	DAD-NUM	5	2004.71
	GEN	3	1910.29
600	DAD-NUM	1	2672.82
	GEN	3	2420.84
800	DAD-NUM	1	4222.48
	GEN	2	3438.40

2.3. table. *The selected best k values for DAD-NUM and GEN protocols.*

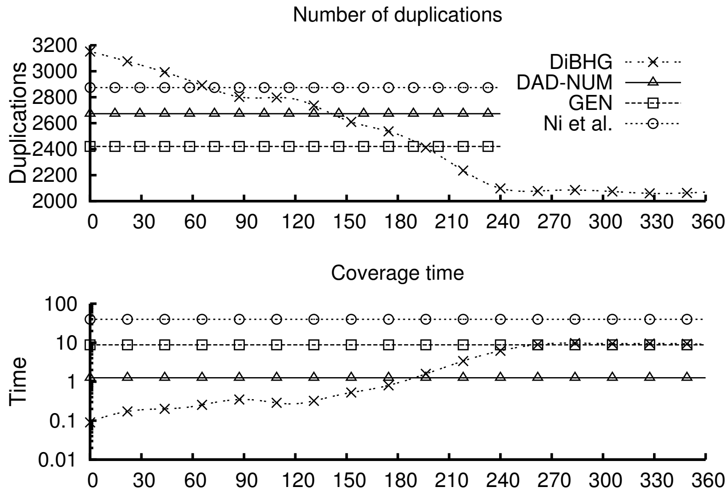
Ni et al's algorithm does not require any additional input parameter (like the k parameter for DAD-NUM and GEN), therefore it can be compared more easily with the Direction Based Handshake Gossiping.

In the DiBHG protocol, the input parameter is the angle, $\alpha_{threshold}$, with which we can adjust the propagation direction. As described earlier in Section 2.2, if this parameter is set to a smaller value, the probability assignment of the Valency Based Handshake Gossiping protocol will be used in the majority of the cases instead of the direction based mechanism. Otherwise, the direction based solution will be the dominant one.

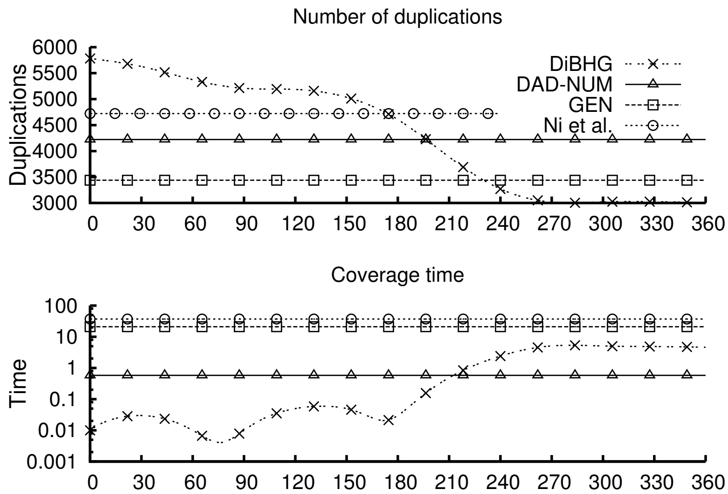
The two performance metrics I deemed most important are the number of duplications and coverage time. The two metrics contradict each other, as we can reduce the coverage time only by the cost of having more duplications, and vice versa. For example, the blind flood has the shortest coverage time, as every node rebroadcasts every received message without any backoff time, but this way the number of duplications will explode. Therefore a compromise and tradeoff should be found between the two.



(a) In a network with 500 nodes



(b) In a network with 600 nodes

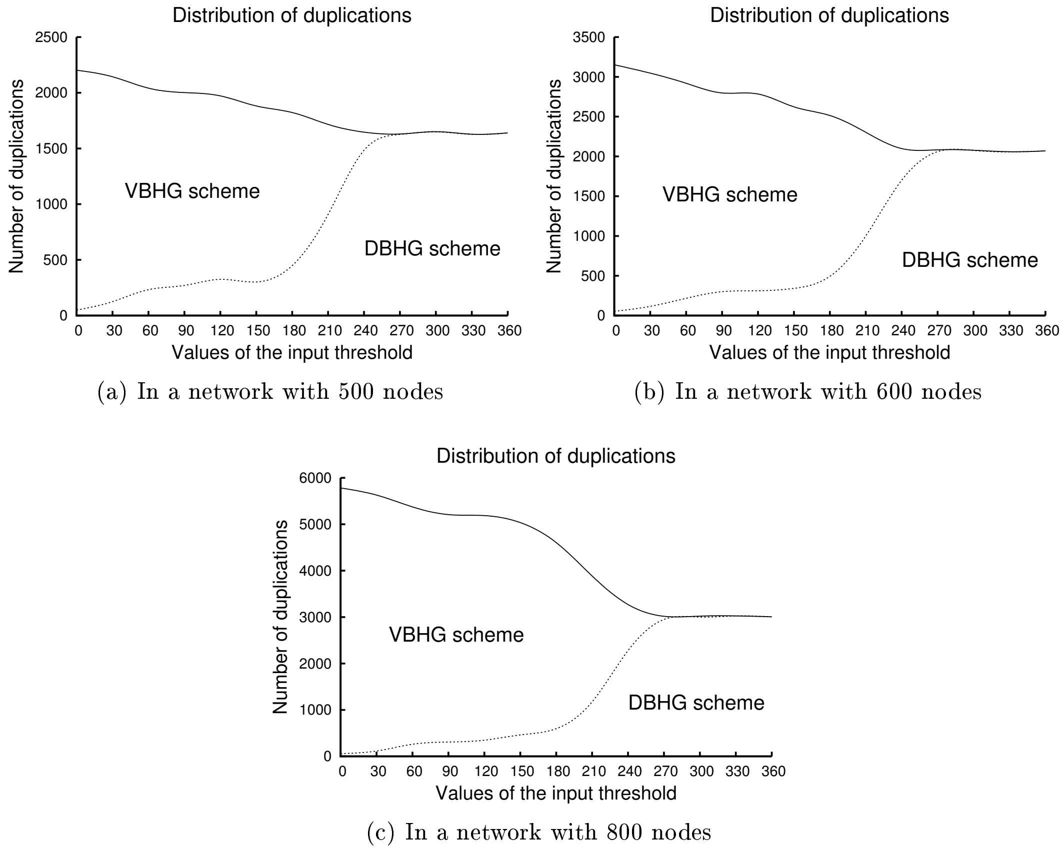


(c) In a network with 800 nodes

2.5. figure. *Results for the 4 protocols in different scenarios*

When testing my new solution, we could focus on these two metrics to verify if it can hold a reasonable tradeoff compared to the existing protocols. The results for number of duplicated messages and coverage times for 500 nodes can be seen in Figure 2.5(a) for the Direction Based Handshake Gossiping, in the function of the $\alpha_{threshold}$ input parameter. For easier comparison, I put the number of duplications and coverage times for the other three protocols in the same figure, measuring them with the previously determined k parameters (listed in Table 2.3). As expected, for small $\alpha_{threshold}$ values the number of duplications for the Direction Based Handshake Gossiping is high, as a small $\alpha_{threshold}$ value means that only a small fraction of the neighboring nodes are covered, which results in a broad angle of data dissemination. In the majority of these cases, (where the $\alpha_{threshold}$ is small), the retransmission probability assignment of the Valency Based Handshake Gossiping is utilized, without a dominant dissemination direction. As it can be seen, for even small values of the $\alpha_{threshold}$, when running simulations with 500 mobile nodes, the DiBHG overperforms the Ni et al. algorithm in terms of number of duplications, and it's performing better than the DAD-NUM protocol above $\alpha_{threshold} = 80$, and better than the GEN protocol starting from $\alpha_{threshold} = 140$. The figure shows that the Direction Based Handshake Gossiping can overperform the other three protocols in terms of number of duplications, if the input parameter is chosen wisely. However, coverage time should also be considered. It can be seen, that at the point it overperforms the DAD-NUM protocol in terms of duplications, it will be less effective in terms of coverage time. That means, that the DAD-NUM will cover the nodes faster, but will cause more duplicates. When compared to the GEN protocol, it is less effective in terms of coverage time from around $\alpha_{threshold} = 230$, while in terms of duplications it is already better from $\alpha_{threshold} = 140$, which means that there is a wide domain of the input parameter for which my protocol performs better than the GEN for both properties. When compared with the Ni et al. algorithm, it over-performs it for all of the values of the $\alpha_{threshold}$ parameter, either in number of duplications, or coverage time. The Direction Based Handshake Gossiping is even more convincing for scenarios where 600, or 800 mobile nodes are simulated. If the $\alpha_{threshold}$ is tuned well, it can significantly reduce the number of duplications, and in the meantime keep the coverage time at the same level.

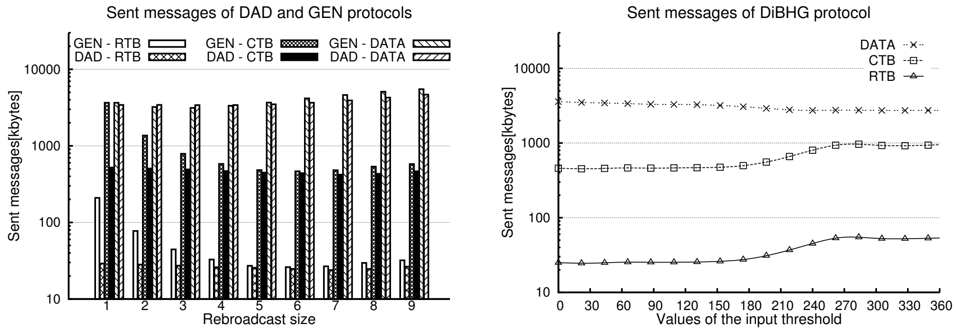
As stated before, my solution is an adaptive scheme, therefore if $\alpha > \alpha_{threshold}$, it uses the probability assignment determined by the Valency Based Handshake Gossiping protocol. Otherwise, if there is an emphasised direction of the transmission, the Direction Based Handshake Gossiping scheme is utilized. It is interesting to check, how the selection of the $\alpha_{threshold}$ input parameter affects the retransmission probability assignment. Figure 2.6 shows us the distribution of the two schemes, when observing the number of duplications. As it can be seen for all three mobile node densities, when setting small $\alpha_{threshold}$ values (under 150) the Valency Based Handshake Gossiping scheme prevails, which will cause plenty of duplications, as there is no distinguished direction of the transmission. This is the range of the α parameter, where my solution is not performing so well in comparison to the other three reference protocols (Figure 2.5). By increasing the value of the input parameter (which means we define a clear direction of the transmission), the probability



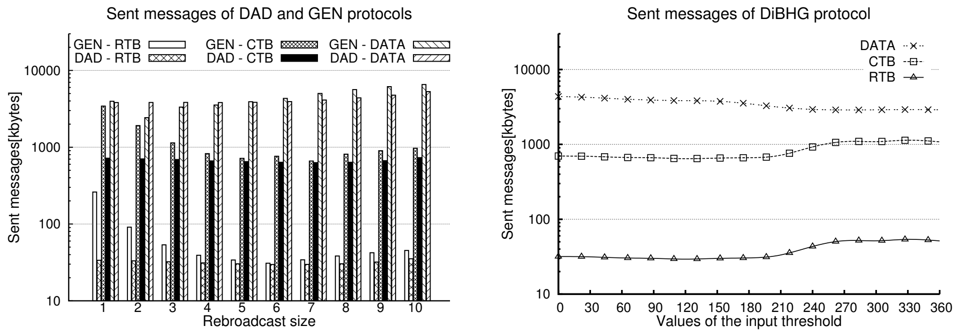
2.6. figure. *Duplication distribution of the VBHG and DiBHG probability assignments.*

assignment of the Direction Based Handshake Gossiping will be dominant, and after reaching $\alpha_{threshold} \approx 270$, the adaptive scheme utilizes solely the Direction Based Handshake Gossiping retransmission probability assignment. From this point on, the number of duplications drop drastically, this is the period when the new solution significantly outperforms the reference protocols.

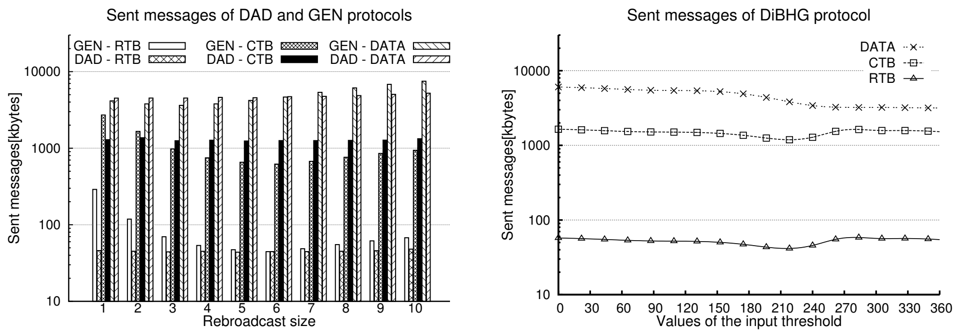
Another interesting (and often overlooked) performance characteristic to investigate is the amount of control overhead produced to achieve these results. It is important to note, that all of the measured protocols are equipped with the 3-way handshaking transmission mechanism, as without this mechanism the reference protocols would produce many more duplications, therefore a fair comparison would not be possible. The 3-way handshake uses RTB (Ready To Broadcast) and CTB (Clear to Broadcast) control messages to coordinate transmissions locally, and to avoid broadcast storms and unnecessary message transmissions. The concrete algorithm of the 3-way handshake along with measurement results are described in [21]. As depicted in Figure 2.7, all measured protocols produce around the same amount of RTB and CTB control messages, CTB being the dominant one, compared to the amount of RTBs produced (the source node sends only one RTB message, but all neighboring nodes send back CTB messages). More interesting is the ratio of the data messages and the control messages, also referred to as control overhead. When observing the DiBHG for the small values of $\alpha_{treshold}$ the overall amount of control messages



(a) In a network with 500 nodes



(b) In a network with 600 nodes



(c) In a network with 800 nodes

2.7. figure. *Distribution of data and control messages of the measured protocols.*

(RTB+CTB) is less than 5% compared to the data messages. As the $\alpha_{treshold}$ increases, the ratio of the control overhead and the data traffic grows (even up to 10% in some cases), however in these cases the amount of the data messages will decrease, which means we need less sent messages to obtain the same coverage. In these simulations the size of the data messages were fixed to 50 kbytes, while the control messages were less then 100 bytes. We are well aware that the 3-way handshake mechanism is only effective, when using it for the propagation of data messages, which are in average larger than 20-30 kbytes. If the average data message is less than this size, the 3-way handshake mechanism should not be used, because of the inefficient control overhead it would produce.

3. Chapter

Conclusion

The clear majority of the data dissemination protocols for self-organized networks do not use spatial information of the network nodes to optimize the bandwidth and channel usage. Even the communication protocols which utilize spatial properties of the network are not capable of enabling the mobile nodes to contribute with their local message retransmissions to a global message propagation direction, this way empowering the mobile nodes to be aware of the global targets. We have designed a novel communication protocol for autonomous mobile systems, the Direction Based Handshake Gossiping, which was implemented in my self-organizing network simulator, together with three other location based data dissemination protocols from the literature (Distance Adaptive Dissemination, the General Probabilistic Broadcast Algorithm, and Ni et al's location-based scheme). The novel solution enables the nodes to propagate their messages in a chosen direction, by giving a higher retransmission probability to nodes in that direction. With this kind of retransmission probability assignment, it is possible to avoid many unnecessary duplications and attempt to follow a given direction from the sender nodes. This can be a very useful trait in various application cases, for example covering only portions of the network or controlling a group of nodes moving in a given direction.

The simulation results show that the Direction Based Handshake Gossiping can overperform the other three protocols in terms of the number of duplications, if the input parameter is chosen correctly. If this $\alpha_{threshold}$ is tuned well, it can significantly reduce the number of duplications, while keeping coverage time the same, and even improving it in some cases. Another important performance indicator we measured is the control overhead, the ratio of data and control messages. When observing the DiBHG for the small values of $\alpha_{threshold}$ the overall amount of control messages (RTB and CTB) is less than 5% of the data messages. As $\alpha_{threshold}$ increases, the control overhead also increases up to 10% in some cases, while the amount of data messages decrease, which means that our protocol can reach the same level of coverage with less data messages sent.

It can be concluded that the adaptive version of the DiBHG (combined with the probability assignment of the Valency Based Handshake Gossiping) can be an effective data dissemination scheme for mobile self-organized networks. Making the nodes aware of their physical locations I was able to implement a directed transmission communication primitive, which can be very useful in a variety of use cases.

Obviously many modifications or improvements (e.g. fine-tuning the probability assignment, or changing the cost function) are possible for the novel algorithm, but in my opinion placing it in a real application will be more interesting. For example it could be an interesting experiment, if we put this (or these) protocol(s) to an emergency simulator. As in this case the hazard is coming from a well-defined location, so it will be important to keep the communication in the right direction (from the fire node to the exits). But it could be interesting as well, if we examine the efficiency of the protocol(s) in an urban scenario, for example when our goal is to avoid traffic jams on the streets (in this case we should transmit the messages from the central of the jams to the cars along the queue).

List of figures

List of tables

References

[1] S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic. *Mobile ad hoc networking*. Wiley-IEEE press, 2004.

[2] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.S. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. In *ACM Sigplan Notices*, volume 37, pages 96–107. ACM, 2002.

[3] J. Burgess, B. Gallagher, D. Jensen, and B.N. Levine. Maxprop: Routing for vehicle-based disruption-tolerant networks. In *Proc. ieee infocom*, volume 6, pages 1–11. Barcelona, Spain, 2006.

[4] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss. Delay-tolerant networking: an approach to interplanetary internet. *Communications Magazine, IEEE*, 41(6):128–136, 2003.

[5] K. Fall. A delay-tolerant network architecture for challenged internets. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 27–34. ACM, 2003.

[6] A. Al Hanbali, M. Ibrahim, V. Simon, E. Varga, and I. Carreras. A survey of message diffusion protocols in mobile ad hoc networks. In *Proceedings of the 3rd International Conference on Performance Evaluation Methodologies and Tools*, page 82. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.

[7] O.K. Tonguz, N. Wisitpongphan, J.S. Parikh, F. Bai, P. Mudalige, and V.K. Sadekar. On the broadcast storm problem in ad hoc wireless networks. In *Broadband Communications, Networks and Systems, 2006. BROADNETS 2006. 3rd International Conference on*, pages 1–11. IEEE, 2006.

[8] S.Y. Ni, Y.C. Tseng, Y.S. Chen, and J.P. Sheu. The broadcast storm problem in a mobile ad hoc network. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 151–162. ACM, 1999.

[9] Y.C. Tseng, S.Y. Ni, and E.Y. Shih. Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network. *Computers, IEEE Transactions on*, 52(5):545–557, 2003.

[10] J. Cartigny, D. Simplot, and J. Carle. Stochastic flooding broadcast protocols in mobile wireless networks. *LIFL Univ. Lille*, 1, 2002.

[11] A. Khelil, P.J. Marrón, C. Becker, and K. Rothermel. Hypergossiping: A generalized broadcast strategy for mobile ad hoc networks. *Ad Hoc Networks*, 5(5):531–546, 2007.

[12] W. Peng and X.C. Lu. On the reduction of broadcast redundancy in mobile ad hoc networks. In *Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing*, pages 129–130. IEEE Press, 2000.

[13] E. Varga, T. Csvorics, L. Bacsardi, M. Berces, V. Simon, and S. Szabo. Novel information dissemination solutions in biologically inspired networks. In *Telecommunications, 2007. ConTel 2007. 9th International Conference on*, pages 279–284. IEEE, 2007.

[14] W.R. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 174–185. ACM, 1999.

[15] X. Chen, M. Faloutsos, and S. Krishnamurthy. Distance adaptive (dad) broadcasting for ad hoc networks. In *MILCOM 2002. Proceedings*, volume 2, pages 879–883. IEEE, 2002.

[16] V.K. Paruchuri, A. Durresi, D.S. Dash, and R. Jain. Optimal flooding protocol for routing in ad-hoc networks. In *IEEE Wireless Communications and Networking Conference*, 2003.

[17] I. Stojmenovic and X. Lin. Applications and algorithms-loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 12(10):1023–1032, 2001.

[18] F. Dai and J. Wu. Performance analysis of broadcast protocols in ad hoc networks based on self-pruning. *Parallel and Distributed Systems, IEEE Transactions on*, 15(11):1027–1040, 2004.

[19] M.T. Sun, W. Feng, and T.H. Lai. Location aided broadcast in wireless ad hoc networks. In *Global Telecommunications Conference, 2001. GLOBECOM'01. IEEE*, volume 5, pages 2842–2846. IEEE, 2001.

[20] J. Cartigny and D. Simplot. Border node retransmission based probabilistic broadcast protocols in ad-hoc networks. In *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on*, pages 10–pp. IEEE, 2003.

[21] A. Kokuti and V. Simon. Location based data dissemination in mobile ad hoc networks. In *Telecommunications and Signal Processing (TSP), 2012 35th International Conference on*, pages 57–61. IEEE, 2012.

[22] G. Korkmaz, E. Ekici, and F. Ozguner. Black-burst-based multihop broadcast protocols for vehicular networks. *Vehicular Technology, IEEE Transactions on*, 56(5):3159–3167, 2007.

[23] K. Xu, M. Gerla, and S. Bae. How effective is the ieee 802.11 rts/cts handshake in ad hoc networks. In *Global Telecommunications Conference, 2002. GLOBECOM'02. IEEE*, volume 1, pages 72–76. IEEE, 2002.

[24] J. Kim, Q. Zhang, and D.P. Agrawal. Probabilistic broadcasting based on coverage area and neighbor confirmation in mobile ad hoc networks. In *Global Telecommunications Conference Workshops, 2004. GlobeCom Workshops 2004. IEEE*, pages 96–101. IEEE, 2004.

[25] J. Yoon, M. Liu, and B. Noble. Random waypoint considered harmful. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 2, pages 1312–1321. IEEE, 2003.