Improving Inter-cluster Broadcasting in Ad Hoc Networks by Delayed Flooding

Adrian Andronache, Patricia Ruiz, and Steffen Rothkugel

FSTC, Campus Kirchberg, 6, rue Richard Coudenhove-Kalergi,
L-1359 Luxembourg, Luxembourg
{adrian.andronache,patricia.ruiz,steffen.rothkugel}@uni.lu

Abstract. Clustering is a well-known approach to cope with mobility in multi-hop ad-hoc networks. The aim of clustering hereby is to detect and maintain stable topologies within a set of mobile nodes. WCPD is an algorithm that allows to interconnect multiple stable clusters and to exchange data across single cluster boundaries. However, disseminating information beyond stable connected sets of nodes is required in some settings as well. The DWCF algorithm introduced in this paper aims at high coverage while keeping the overhead low. DWCF facilitates that by exploiting the cluster structures and selectively forwarding data to foreign neighbor clusters.

Key words: Networking, ad hoc, mobile, delay, tolerant, weighted, cluster, broadcast.

1 Introduction

An increasing number of present-day mobile devices like tablets, PDAs, smartphones and laptops provide adapters for both, ad hoc and backbone communication. In our hybrid network model mobile devices use Wi-Fi adapters to communicate in an ad hoc fashion and cellular adapters to access an infrastructured backbone. The backbone assists the mobile network by providing authentication and partition inter-connection mechanisms as well as information of interest. In both ad-hoc networks and sensor networks, clustering is one of the most popular techniques for locality-preserving network organization [1]. Cluster-based architectures effectively reduce energy consumption, and enable efficient realization of routing protocols [2], data aggregation [3, 4], and security mechanisms [5]. In our previous introduced backbone-assisted mobile ad hoc network applications [6, 7] a cluster is created by electing a so called clusterhead among one hop neighbor devices. The applications use the clusterheads to keep track of cluster interests and register them at the backbone by an uplink maintained for instance by a cellular connection. The backbone provides multimedia items of interest to the registered clusterheads which forward them to interested cluster slave devices in the ad hoc network. For the self-organization of the mobile devices in cluster structures we employ the Node and Link Weighted Clustering Algorithm – NLWCA [8] . NLWCA is designed to organize mobile ad hoc networks in one hop clusters. Each device elects exactly one device as its clusterhead, i.e. the neighbor with the highest weight. The weight of the own device is calculated by NLWCA based on device properties that are important for the application running on top of it. For instance good signal strength to the backbone network is important for a multimedia providing application since it increases the download bandwidth. Favorable is also a high network degree, which optimizes the multimedia items spreading process. These device properties along with the battery power are used to calculate the weight, which is sent by the beacon to the one-hop neighbors. An elected clusterhead also investigates its one-hop neighborhood, similarly electing the device with the highest weight as its clusterhead. This process terminates when a device elects itself as its own clusterhead, due to the fact of having the highest weight among all its neighbors. We call all intermediary devices along such clusterhead chains sub-heads. Each device on top of a chain is called a full clusterhead, or, in short, clusterhead. Hence, in each network partition, multiple clusterheads might coexist. The main goal of the algorithm is to avoid superfluous re-organization of the clusters, particularly when clusters cross each other. To achieve this, NLWCA assigns weights to the links between the own node and the network neighbor nodes. The link weight is used to keep track of the connection stability to the one-hop network neighbors. When a link weight reaches a given stability threshold the link is considered stable and the device is called stable neighbor device. The clusterhead is elected only from the set of stable neighbors which avoids the re-organization of the topology when two clusters are crossing for a short period of time. In previous work we introduced the Weighted Cluster-based Path Discovery protocol (WCPD) [9] , which is designed to take advantage of the cluster topology built by NLWCA in order to provide reliable path discovery and broadcast mechanisms in mobile ad hoc networks. In this work we present an approach that increases the ad hoc network inter-cluster broadcast performance of the WCPD protocol by using delayed flooding. The remaining of the paper is organized as follows: Section 2 describes the WCPD inter-cluster broadcasting mechanism. Section 3 introduces the *Delayed Weighted Cluster Flooding (DWCF)* algorithm. In Section 4 we describe the simulation settings and the results obtained. Related work is located in Section 5. Section 6 presents the conclusions and the future work.

2 WCPD inter-cluster broadcasting

In this section we describe the WCPD broadcasting protocol introduced in [8]. Since DWCF aims to improve the broadcast performance in mobile ad hoc networks, we present the WCPD protocol that works without the assistance of the backbone. WCPD runs on each network node and requires solely information available locally in the one hop neighborhood. The algorithm uses information provided by NLWCA: the set of stable connected network neighbor nodes and the ID of the own clusterhead. NLWCA also propagates by beacon the own weight and the ID of the current clusterhead. Besides the information provided by NLWCA, the WCPD protocol uses the beacon to disseminate the list of locally discovered nearby connected clusterheads. By doing so, every node has the following information about each stable one hop neighbor: its clusterhead ID and the ID set of discovered clusterheads and the respective path length. After the data of all stable one hop neighbors is checked, the set of discovered nearby clusterheads and the path length is inserted into the beacon in order to propagate it to the one hop neighborhood. Since WACA elects one clusterhead in each one-hop network neighborhood, the path length between two clusterheads of connected clusters is at least two hops and at most three hops. If the number of hops to a clusterhead is higher than three then its cluster is not directly connected to the local cluster and its ID is not added to the set of nearby clusterheads. In other words, WCPD keeps track only of stable connected clusterheads that are at most three hops afar. The WCPD broadcasting algorithm is simple and easy to deploy: the broadcast source node sends the message to the clusterhead, which stores the ID of the message and broadcasts it to the one hop neighborhood. After that, it sends it to all nearby clusterheads by multi-hop unicast 1. The inter-cluster destination-clusterheads repeat the procedure except that the message source clusters are omitted from further forwarding. Additionally the information about the ID of the broadcast messages and their sources is stored for a given period of time to avoid superfluous re-sending of the message. The protocol sends the broadcast message to nearby clusters connected by stable links in order to disseminate it to the network partition. Nevertheless the message also reaches crossing clusters since the broadcasts are received by all nodes in the one-hop neighborhood of local leaders. This increases the chance that the message reaches a high number of nodes in the mobile network partition.

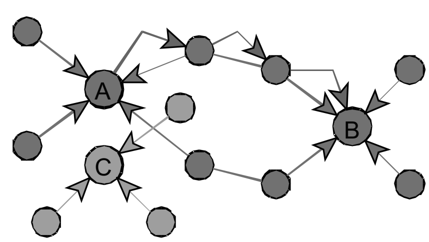

Fig. 1. The clusterheads A and B are stable connected in the topology built by NL-WCA. The main goal of the WCPD broadcasting protocol is to reach the nodes of stable connected clusters. To achieve this, when A broadcasts a message it also sends it by multi-hop unicast (illustrated by the arrows) to the stable connected cluster B. As side effect, the crossing clusterhead C receives the broadcast message from A like every other node in one-hop communication range. Clusterhead C re-broadcasts the message in order to reach the stable connected nodes, thus increasing the number of the broadcast receivers..

3 The Delayed Weighted Cluster Flooding Protocol (DWCF)

The inter-cluster broadcast protocol of WCPD is designed to reach the members of stable-connected clusters in the network vicinity of the broadcast source node. The algorithm is easy to deploy and aims to avoid computational and communication load on the mobile nodes. To achieve this, the next hop to a cluster in the vicinity is picked up based on its weight, thus the device with the highest amount of energy left is elected as router. This method is simple but it does not take topological properties of the potential next-hop nodes into account. Since the protocol acts in mobile environments the elected router-node might lose its path to the destination cluster, thus dropping the message. Further, the mechanism is not well suited for applications that require network flooding since the main goal of the WCPD inter-cluster broadcast is to reach only the nearby stable-connected clusters. In order to improve the network flooding performance of the WCPD protocol we introduce the *Delayed Weighted Cluster Flooding (DWCF)* algorithm, which is inspired by the Delayed Flooding with Cumulative Neighborhood (DFCN) algorithm introduced in [10]. The DFCN algorithm aims to minimize the network load during flooding by taking into account the network density. When the network is sparse, it is quite difficult to spread a message, so in that conditions, a node should forward the message as soon as another device enters the communication range. This would lead to good results since every re-emission proves to be useful because of the reduced number of meeting points between nodes. However, in dense networks this strategy would lead to a high overload and broadcast storms. In order to avoid this, DFCN sets a random delay (RAD) when a node receives a new message. If the density is low then the RAD is immediately set to zero after a new neighbor is met, but this behavior is disable when the density is high. This perception of the density corresponds directly to its neighborhood and it is managed with a threshold called densityThreshold. DFCN attaches to the broadcast message a list T(m) containing the current neighbors of the node. This list is managed as follows: when a node s broadcasts a message m to its neighbors, it assumes that all of them will receive m. Therefore, T(m) is set with all the neighbors, N(s), of the sender-node s. Once the RAD is finished, DFCN uses this list to decide whether the received message will be forwarded or not, in terms of the neighbors that already received the message which are in the one hop neighborhood. For that, a threshold called minBenefit is set, which is formally defined on the benefit, computed as the ratio between the neighbors of s which do not belong to T(m), and the total neighbors of s, N(s). The higher the benefit, the higher the probability of (re)-emission of a message. The Delayed Weighted Cluster Flooding algorithm aims to be as simple as possible but at the same time to improve the flooding performance of the WCPD protocol. To achieve this, the protocol uses the cluster topology provided by NLWCA and the cluster information provided by the WCPD Nearby-Cluster Discovery protocol. The basic idea of the protocol is to use broadcast instead of unicasts to reach nearby clusters. This increases the chance that a sent message reaches the destination cluster even if the connecting cluster-border nodes are changing during transmission. Also, the broadcast might reach nodes belonging to different clusters, thus reducing the communication load. One of the main design goals is to keep the algorithm as simple as possible, which reduces computational load on the nodes, but at the same time to avoid communication overload. The main issue is to avoid that several nodes situated on the border of the cluster re-broadcast a message in order to reach the same neighbor clusters. Inspired by DFCN, the DWCF algorithm uses random delays before it forwards the broadcast messages. Further, the decision to re-send the message is based solely on the presence of foreign-cluster nodes in the one-hop neighborhood at the end of the delay. This keeps the algorithm simple and reduces the number of node IDs added to the broadcast message since generally the number of nearby clusterheads is much lower than the number of one-hop neighbors. The broadcast-handler method is called when a node receives a broadcast message. First of all DWCF checks if the message was already processed (delay = -1) and breaks the method if true. If the message was not processed but is already in the queue then the set of receiver cluster IDs is updated since the sender might reach new clusters. This avoids later that nodes re-send the message if some other node already sent it during the delay to all foreign clusters in one-hop neighborhood. In case the message is new, it is instantly re-broadcasted if the receiver is a clusterhead or a sub-head in order to reach all nodes in the cluster. If the receiver is a slave node then a random delay is set for the message, which is added to the queue and the delay timer is started. The delay timer fires a tick event every second. The tick handler method is in charge of decreasing the delay of every message in the queue that is not already processed (having a delay of -1). If the delay of a message reaches the value 0 then the DelayBroadcast method is called. If the queue contains no message with delay higher than 0 then the timer is stopped in order to save computational load. After the delay of a message, the DelayBroadcast method is called. DWCF searches the last received beacons of the one-hop neighbor nodes in order to discover foreign clusters. This information is provided by NLWCA, which uses the beacon to send the own weight node and the current clusterhead ID to the neighboring nodes. The set of discovered foreign cluster IDs is compared to the set of receiver-cluster IDs contained in the message. If at least one new cluster is reachable then DWCF replace the receiver-cluster IDs with the current neighboring cluster IDs and broadcasts the message. The message is also broadcasted if it was not already received from a member node of the own cluster. Thus, the messages received from foreign clusters reach the own clusterhead for further spreading. After the message is forwarded its payload is deleted but the ID is kept for a given time to avoid further processing. The pseudo code of the algorithm is like follows:

```
--- Required Data ---
d: The ID of the node.
C(d): The clusterhead ID of d.
N(d): The set of one-hop neighbor nodes of d.
m(d): A broadcast message received from d.
```

```
M(m): The ID set of cluster reached by broadcast of m.
f(m): True if m was received from the own cluster.
r(m): The re-send delay for m.
Q: The message queue.
T: Delay-timer, fires a tick event every second.

--- Handler method m(e) on node d ---
If( r(m) == -1 ) return; // already processed message
If( Q contains m ) do: // message already in queue
   Merge received M(m) with local M(m);
   return; // break
End do;
If( d is clusterhead or is sub-head ) do:
   r(m) = -1; // mark it as processed
   add (m,r(m)) to Q; // remember it
   Send_broadcast(m); // instantly re-send
   return;
End do;
If( C(e) equals C(d) ) f(m) = true;
r(m) = random number between 1 and n;
add (m,r(m)) to Q;
If( T is not running ) start T;

--- Handler method for the ticks of T ---
For each ( m(e) in Q ) do:
   run = false; // timer re-start flag
   If( r(m) > -1 ) r(m) = r(m) - 1;
   If( r(m) > 0 ) run = true;
   If( r(m) == 0 ) DelayBroadcast(m);
End do;
If( run = false ) stop T; // more messages to process

--- DelayBroadcast method for message m on node d ---
For each ( e in N(d) ) // find foreign clusters
   If( C(e) not equals C(d) ) add C(e) to K;
re-send = false;
If( K \ M(m) not null ) re-send = true;
If( f(m) == false ) re-send = true;
If( re-send == true ) do:
   M(m) = K;
   Send_broadcast(m);
   Delete payload(m); // keep only the ID
End do;
```

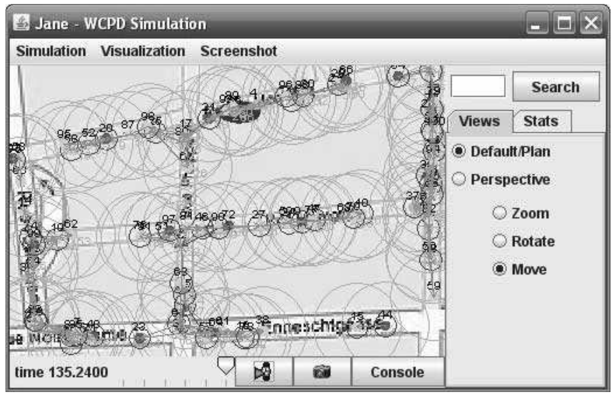

Fig. 2. JANE simulating the WCPD protocol on 100 devices. The mobile devices move on the streets of the Luxembourg City map. The devices move with a speed of 0.5 – 1.5 m/s and have a sending radius of 20 m.

4 Experiments and results

In order to compare the flooding performance of the two algorithms, we implemented them on the top of the JANE simulator [11]and performed several experiments. For the conducted experiments we used the Restricted Random Way Point mobility model [12], whereby the devices move along defined streets on the map of Luxembourg City 2. For each device the speed was randomly varied between 0.5 and 1.5 m/s (1.8 and 5.4 km/h), which are common human speeds. For this speed range the NLWCA link stability threshold is set on 2. At simulation startup, the devices are positioned at random selected crossroads and the movement to other crossroads is determined by the given random distribution seed. A number of ten different random distribution seeds were used in order to feature results from different topologies and movement setups. In order to monitor the information dissemination performance and network load of the broadcasting mechanisms, a node was chosen to broadcast a message every 10 seconds during different simulation runs using different distribution seeds. The number of sent messages (i.e. broadcasts and unicasts) during the dissemination and the number of reached network nodes were tracked for 300 seconds. The experiments were done in sparse networks with 100 mobile devices and in dense networks with 300 devices.

The tracking results regarding the message dissemination performance and network load of the inter-cluster broadcasting protocols are presented in Fig. 3 and Fig. 4. The overall results show that DWCF performs much better in terms of message dissemination than the WCPD inter-cluster broadcast protocol. The denser the network, the higher the difference between both the number of sent messages and the number of receiver nodes. The results for sparse networks with 100 devices are presented in Fig. 3. The WCPD uses a number of messages with a mean value around 100 to reach around 10% of the receivers. The low

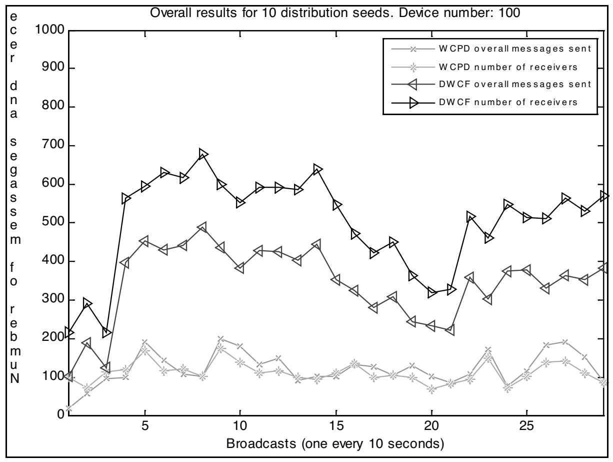

Fig. 3. Simulation results for sparse networks with 100 devices. The number of sent messages is almost the same as the number of received when using WCPD inter-cluster broadcasting. The flooding performance of DWCF is much higher but at the cost of a higher network overload.

reachability can be explained by the fact that WCPD aims to reach with the broadcast only the stable connected clusters. The number of sent messages is high in relation to the number of reached devices and it emerges due to the multi-hop unicasts used to forward the message to connected clusters. The DWCF protocol performs much better in terms of reached devices whose ratio has a mean value around 50%. The increased dissemination performance requires a higher number of sent messages, which has a mean value around 400.

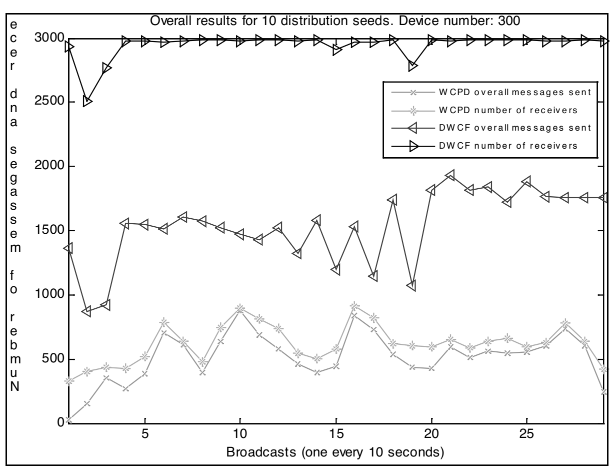

Fig. 4. In dense networks with 300 devices the performance of DWCF is even better. The difference between the number of sent messages and the number of receivers is much higher than in sparse networks.

The results in Fig. 4 show that the DWCF performs even better in denser networks with 300 devices, where only 1500 messages were sent. The most of them reached 100% of the receivers. Since DWCF uses broadcasts to forward the messages to neighbor clusters, the probability to reach them is higher than when using multi-hop unicasts. Even if the connecting nodes change during the forward process, the new ones receive and re-send the broadcasted message. The broadcasted messages also reach crossing foreign clusters in communication range of the forwarding border nodes. These clusters are missed by the WCPD unicasts.

5 Related work

The Delayed Flooding with Cumulative Neighborhood (DFCN) [10] is a broadcasting protocol designed for mobile ad hoc networks. The protocol uses the information about the one hop neighbors provided by the ad hoc network beacons. When a node receives a broadcast message a random delay is set for the re-sending of the message. The protocol aims to improve the flooding performance by forwarding the message instantly when a new node is detected in sparse networks and to wait longer in dense networks in order to avoid communication overload. DFCN is optimized in [13] using cMOGA, a new cellular multi-objective genetic algorithm. Main goals of the work are minimizing the duration of the broadcasting process, maximizing network coverage, and minimizing the network usage. Three different realistic scenarios were used corresponding to a shopping center, the streets in a city and a wide non-metropolitan area wherein several roads exist. In [14] a delay-tolerant broadcasting algorithm is proposed for wide and public wireless dissemination of data. The approach is not using flooding or routing, it uses only the mobility of the nodes in order to spread the data. The hybrid setting of Sun et al [15] consists of base stations that are inter-connected and mobile devices that can connect locally via an ad-hoc mode or to a base station if near enough to it. Two routing schemas are introduced to deal with different application requirements. Sun et al research points out that the efficiency of the chosen communication mode strongly depends on the applications running in the overall network.

6 Conclusion and future work

In this paper we introduced a new flooding algorithm employed on top of the cluster topology built by NLWCA. The protocol is simple and easy to deploy and aims to avoid high computation and communication load on the network nodes. The simulation results show that DWCF highly outperforms the inter-cluster broadcasting protocol of WCPD in terms of message spreading. On the other side, WCPD focus on the communication with nearby clusters considered to be stable. This is beneficial for applications which require that a multi-hop path between two communicating nodes has certain stability. In future work we aim to improve the DWCF by decreasing the delays on high weighted nodes in order reduce the load on nodes with low battery power. Besides this, nodes with a higher number of neighboring foreign clusters should forward the message faster in order to decrease the network communication load.

References

1. Peleg, D.: Distributed computing: a locality-sensitive approach. Society for Industrial and Applied Mathematics. Philadelphia, PA, USA, (2000)
2. Heinzelman, W.B., Chandrakasan, A.P., Balakrishnan, H. and Mit, C.: An application-specific protocol architecture for wireless microsensor networks. In: IEEE Transactions on Wireless Communications, 1(4), 660–670. IEEE Press, (2003)
3. Dasgupta, K., Kalpakis, K. and Namjoshi P.: An efficient clustering-based heuristic for data gathering and aggregation in sensor networks. In: Wireless Communications and Networking, 3. 1948–1953. IEEE Press, (2003)
4. Luo, H.G., Ye, F.G., Cheng, J.G., Lu, S.G. and Zhang, L.G.: TTDD: Two-Tier Data Dissemination in Large-Scale Wireless Sensor Networks. In: Wireless Networks, 11 (1), 161–175. Springer, (2003)
5. Zhu, S., Setia, S. and Jajodia, S.: LEAP: Efficient security mechanisms for large-scale distributed sensor networks. In: Proceedings of the 10th ACM conference on Computer and communication security, 62–72. Washington, DC, USA, (2003)
6. Andronache, A., Brust, M.R. and Rothkugel, S.: Multimedia Content Distribution in Hybrid Wireless using Weighted Clustering. In: The 2nd ACM Workshop on Wireless Multimedia Networking and Performance Modeling, ACM Press, (2006)
7. Andronache, A., Brust, M.R. and Rothkugel, S.: HyCast-Podcast Discovery in Mobile Networks. In: The Third ACM International Workshop on Wireless Multimedia Networking and Performance Modeling. ACM Press, (2007)
8. Andronache, A. and Rothkugel, S.: NLWCA-Node and Link Weighted Clustering Algorithm for Backbone-assisted Mobile Ad Hoc Networks. In: The Seventh International Conference on Networking. Cancun, Mexico, (2008)
9. A. Andronache and S. Rothkugel: HyTrace-Backbone-assisted Path Discovery in Hybrid Networks. In: International Conference on Communication Theory, Reliability, and Quality of Service. Bucharest, Romania (2008)
10. Hogie, L., Guinand, F. and Bouvry, P.: A heuristic for efficient broadcasting in the metropolitan ad hoc network. In: 8th Int. Conf. on Knowledge-Based Intelligent Information and Engineering Systems, 727–733. Wellington, New Zealand, (2004)
11. Gorgen D., Frey, H. and Hiedels, C.: JANE-The Java Ad Hoc Network Development Environment. In: 40th Annual Simulation Symposium (ANSS07), 163–176. Norfolk, VA, (2007)
12. Ljubica B., Silvia G., Jean-Yves L. B.: Self Organized Terminode Routing. In: Cluster Computing, 5(2). Springer, (2002)
13. Alba E., Dorronsoro B., Luna F., Nebro A.J., Bouvry P. and Hogie L.: A cellular multi-objective genetic algorithm for optimal broadcasting strategy in metropolitan manets. Computer Communication journal, 30(4), 685–697. Elsevier, (2007)
14. Karlsson, G., Lenders, V., May, M.: Delay-Tolerant Broadcasting. In: SIGCOMM 2006 Workshops, ACM Press, (2006)
15. Sun, Y. and Belding-Royer, E. M.: Application-oriented routing in hybrid wireless networks. In: IEEE International Conference on Communications, ICC'03, 1, 502–506. Anchorage, Alaska, (2003)