Aalborg Universitet

AALBORG UNIVERSITY
DENMARK

Framework for Optimizing Cluster Selection using Geo-assisted Movement Prediction

Kristensen, Thomas Sander; Madsen, Jacob Theilgaard; Pedersen, Michael Sølvkjær; Sørensen, Chres Wiant; Nielsen, Jimmy Jessen; Madsen, Tatiana Kozlova

Published in:
Proceedings of the 5th Symposium on Communications, Control and Signal Processing (ISCCSP), 2012

DOI (link to publication from Publisher):
10.1109/ISCCSP.2012.6217845

Publication date:
2012

Document Version
Accepted author manuscript, peer reviewed version

Framework for Optimizing Cluster Selection using Geo-assisted Movement Prediction

Thomas S. Kristensen, Jacob T. Madsen, Michael S. Pedersen, Chres W. Sørensen,
Jimmy J. Nielsen, Tatiana K. Madsen
{tskr08,jtma08,mp08,cwso}@student.aau.dk, {jjn,tatiana}@es.aau.dk
Section for Networking and Security, Department of Electronic Systems, Aalborg University, Denmark

***Abstract*—Due to the availability of sattelite- and radio-based location systems in most new devices, it is possible to use geographical location of a node for network management and communication protocol optimization. It is a common belief that usage of location information can bring performance benefits. However, inaccuracy and delay in obtaining such information, together with an associated overhead, can have a negative impact. In this paper we have considered a particular case of usage of location information, namely for cluster selection in mobile networks and have analyzed the impact of inaccurate movement prediction and inaccurate location estimation on its performance. The proposed algorithm is compared with two reference algorithms: when a considered node associates with either the first discovered cluster or the nearest cluster. Evaluation shows significant performance benefits in terms of average connectivity time to a cluster head and reduced overhead in case of exact future trajectory prediction. Under more realistic scenarios where location estimation or movement prediction are not perfect, performance benefits are reduced. This emphasizes the need for good movement prediction module if location-based schemes should be implemented in products.**

I. Introduction

Currently, location measurement technologies are becoming more and more widespread and following the noticeable advances in personal navigation aids and tracking services it is becoming realistic for internet service providers to offer location-based services and applications. What is more, the use of location information may help to advance mobile communication and mobile computing, e.g., by helping conducting efficient network resource management, and in this way by enhancing the quality of service of networks. Location information can potentially facilitate a node's interaction with other nodes in a network, e.g. for cooperative communication. In our work we study the feasibility of using location information for enhancing cluster management, which is an approach that organizes independent nodes into clusters.

In ad hoc and sensor networks clustering is used to create and maintain a reliable structure that can be realized in a distributed manner and, which is robust towards topological changes. Clustering techniques can potentially help to reduce scalability problems of routing protocols. Selected cluster heads (CH) will form a backbone along which information is forwarded, thus reducing route maintenance overhead. As the subject of clustering is not new, many algorithms for cluster creation and maintenance have been proposed. Depending on the scenarios and network applications, the optimization goal of clustering algorithms varies from being energy-efficiency that is important for battery-powered sensor networks, to increasing longelivity of clusters by grouping similarly moving nodes together.

Due to the availability of sattelite- and radio-based location systems in most new mobile devices, it is possible to include geographic location of the node as a parameter in the cluster formation algorithm. Here we give a brief overview of the most well-known geo-assisted clustering approaches. *Stable Cluster Protocol (SCP)* [1] predicts a suitable cluster head by comparing velocities of all nodes and selecting the one with the velocity closest to the average velocity. The *Mobility Based Clustering Algorithm (MBC)* [2] works in a similar manner, but instead of calculating the average velocity for a set of nodes, *MBC* calculates the relative mobility between each pair of nodes in a cluster. The *k-hop Compound Metric Based Clustering Algorithm* [3], [4] uses the velocity to calculate the link expiration time between two nodes, which, when used in a cluster, can help determine the most suitable cluster head.

Geo-location based clustering approaches proposed so far in the literature are based on the assumption that accurate location information is available in the network. Under these idealized assumptions introducing location information as one of the parameters for clustering algorithm design is always beneficial. However, inaccuracy and delay in obtaining location information can have a negative impact on the algorithm performance. Additionally, in order to estimate relative mobility of devices, location information should be exchanged among nodes. This leads to extra resource consumption, which can again negatively influence the benefits from clustering. In our work we propose a framework for realistic assessment of geo-assisted clustering algorithms taking into account the above mentioned aspects and also verifying it under a non-linear mobility model.

We assume that the whole network is divided into several clusters; each cluster has a CH. The clustering and the selection of cluster heads can be done by using an existing protocol like LEACH [5], or any other suitable approach. We restrict our attention to the cluster selection problem: which is when a node is looking for a cluster to join. In order to minimize the communication overhead of multiple join and leave events, the criterium of cluster selection is chosen to be maximization of time spent in a cluster. Thus, only when a node is out of range of the current CH, it will start looking for a new cluster to join. This problem settings is also motivated by a scenario of heterogeneous network where only some nodes possess special capabilities and the other nodes can achieve performance benefits of communicating through those nodes, as for example with mobile relay stations.

II. Cluster selection algorithm using movement prediction

In the following we describe the proposed algorithm for cluster selection among available clusters. This approach is based on the availability of position information via satellite- or radio-based localization methods. We assume that each node can produce an estimation of its own position and it can keep in the memory a number of previous positioning samples. This information can be used by a node for its movement prediction. Nodes can request location information from neighboring CHs, either in the form of raw position estimates or in the form of a predicted future movement pattern. By comparing relative mobility, an individual node is able to make an educated guess of which cluster to be preferred for communication.

Generally, the framework for cluster selection consists of two modules: movement prediction module and cluster selection module. The output of the first module is used as an input parameter for the second. As the name indicates, the first module uses past positing information to make a prediction of future node movements. The algorithms used within this module can be chosen to be arbitrarily complex, e.g. based on learned typical mobility patterns of a node and a structure of an environment (walls and doors placement and floors layouts). In the second module based on the predicted movement of a node and surrounding cluster heads, a decision on cluster selection is made. Different optimization criteria can be used: e.g., to connect to a cluster where a distance to a CH is less than 10 meters for at least 3 minutes. In practice, algorithms used for both modules heavily depend on the application scenario, network configuration and properties of the environment.

As the purpose of our work is not to evaluate geo-assisted clustering under one specific scenario, but to investigate the feasibility of the framework and quantify the impact of location errors on the clustering, we have chosen simple algorithms for both modules. For the movement prediction averaging the change in position over the last 5 samples produces a velocity estimate. A linear model is used for future trajectory derivation. In the second module, the time to stay connected to each cluster head that are within communication range are estimated based on the predicted mobility and the cluster with the longest time-to-live connection is chosen. In the rest of the paper we will refer to this version of cluster selection algorithm as *Movement Prediction Algorithm*.

III. Evaluation

The proposed *Movement Prediction Algorithm* is evaluated using a custom matlab simulation that simulates the movements and connectivity of mobile nodes. The connectivity is realized using a binary disc connectivity model with a range of 60 m. That is, nodes that are within 60 m of each other are able to communicate, if the distance is larger they cannot. The simulation environment is a $500\ m \times 500\ m$ square area, which is artificially extended by letting the edges wrap around, effectively making the environment a borderless torus, as described in [6]. Two different mobility models are considered, namely a *Linear Mobility Model*, for which each node follows an initially chosen random direction with a common constant speed for the duration of the simulation, and a *Gauss-Markov Mobility Model*, as defined in [6], using a scaling factor $\alpha = 0.95$ and a 0.5% probability of turning either $90°$ left or right at each sample point.

The proposed *Movement Prediction Algorithm* is compared to two other clustering algorithms. These are the *Nearest Cluster Algorithm* and the *First Cluster Algorithm*. For the *Nearest Cluster Algorithm* we assume that the signal strength of the connection between the node and the clusters are directly proportional to the distance, and that the node by joining the cluster with the highest signal strength joins the cluster that is closest. The *First Cluster Algorithm* simply picks the first cluster it detects, and joins that cluster.

For the considered algorithms the overhead experienced by the network per second at the IP layer is calculated as follows. We define 22 bytes as the minimum packet size based on the header of the IP, and 8 bytes as mobility data. We define the conversation between the node and the clusters, when the node wants to join a cluster, as:

- The node broadcasts its desire to join a cluster (22 bytes).
- All clusters in range respond with their mobility data (22 + 8 bytes per cluster).
- The node chooses which cluster to join and informs the chosen cluster (22 bytes).
- The chosen cluster acknowledges (22 bytes).

The total overhead for the *Movement Prediction Algorithm* is:

$$OH_{\text{MP}} = (N_{\text{CR}} \cdot (8 + 22) + 3 \cdot 22) \cdot N$$

and the total overhead for the two other algorithms is:

$$OH_{\text{NA}} = (N_{\text{CR}} \cdot 22 + 3 \cdot 22) \cdot N$$

where OH is the overhead experienced for the whole simulation, N_{CR} is the average number of clusters in range just before joining a cluster and N is the number of times the node joins a cluster.

IV. Results

A. Linear movement model

When nodes move along a linear trajectory, have a perfect location estimation and a prediction module is based on linear mobility model, there is no prediction error and we have a perfect knowledge about future movement trajectory for both a node that would like to join a cluster and all CHs. This is a somewhat idealizaed situation, and it can be expected that under these conditions using location infomration brings the largest benefits. Indeed, it can be seen in Fig. 1 that the

Movement Prediction Algorithm achieves a longer lifetime in clusters comparing with the reference algorithms. As the cluster density increases, the chances of a CH moving the same direction as a node in a vicinity of the node increases and it leads to increase in performance of *Movement Prediction Algorithm*, while the performance of two other algorithms is unchanged. Due to the same reason overhead of joining new clusters for *Movement Prediction Algorithm* stays low (see Fig. 2).

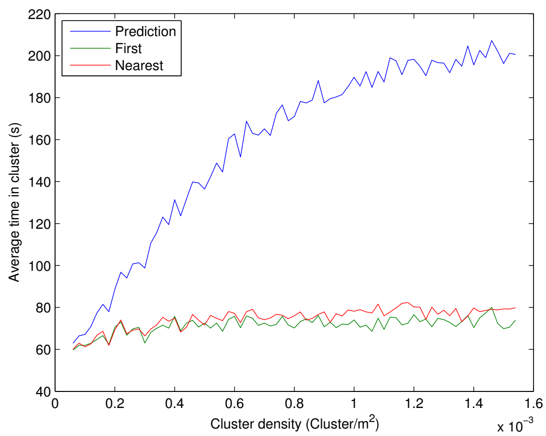

Fig. 1. Cluster density vs. average time in cluster for the *Linear Mobility Model*. Average speed is 1.3 m/s.

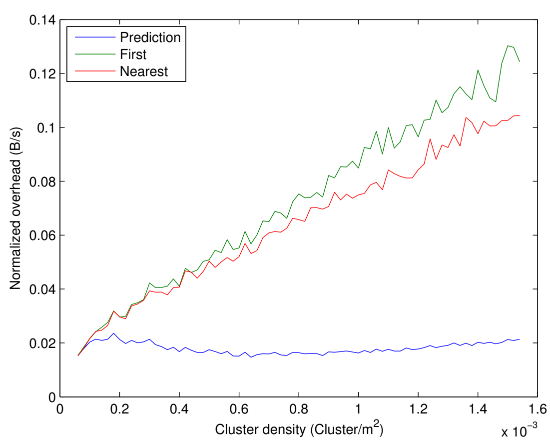

Fig. 2. Cluster density vs. overhead for the *Linear Mobility Model*. Average speed is 1.3 m/s.

Fig. 3 and 4 show that when increasing the speed, the average time in clusters drop and the overhead increases, since nodes move faster out of each other communication range. Still *Movement Prediction Algorithm* shows better performance.

B. Non-perfect location estimation

To model a non-perfect location estimation, we introduce an estimation error by adding a Gaussian zero-mean noise with a standard deviation σ_{loc} to the initially perfect location

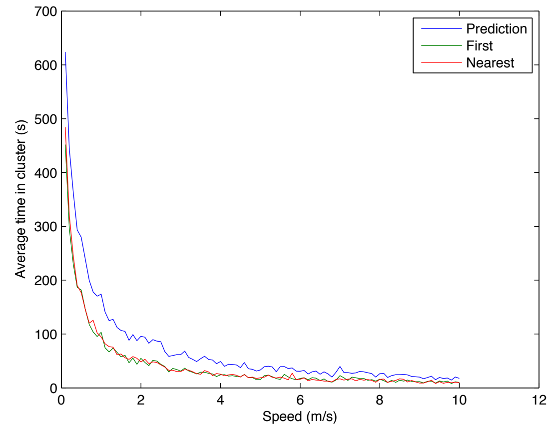

Fig. 3. Speed vs. average time in cluster for the *Linear Mobility Model*. Cluster density is 0.0004 $clusters/m^3$.

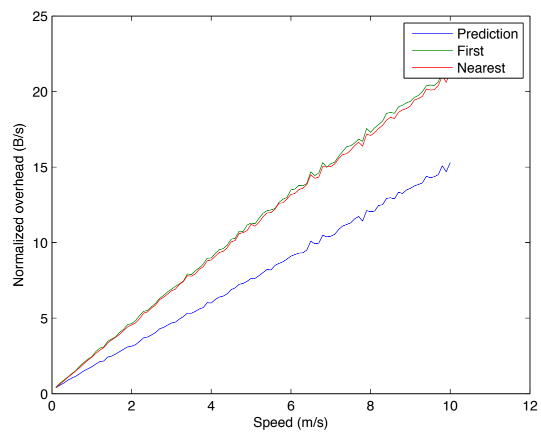

Fig. 4. Speed vs. overhead for the *Linear Mobility Model*. Cluster density is 0.0004 $clusters/m^3$.

estimate. This leads to non-perfect future trajectory prediction and possibly to sub-optimal choice of a cluster to join. This approach allows to quantify the impact of inaccurate location information on the proposed algorithm.

As shown in Fig. 5, the performance of the *Nearest Cluster Algorithm* and the *First Cluster Algorithm* do not change, as they do not rely on location information. The *Movement Prediction Algorithm* is affected by the noise, and the average time in cluster drastically decreases when the noise increases. It ends up fluctuating around the same values as the *Nearest Cluster Algorithm*. The prediction fit shown in the figure is an exponential function fit obtained using matlab.

C. Non-linear movement model

Using Gauss-Markov mobility model produces movement trajectories with many random turns that are smoothened. In this case linear prediction module can not produce a good estimate for futute trajectory and it will decrease the benefits that potentially can be achieved using location information.

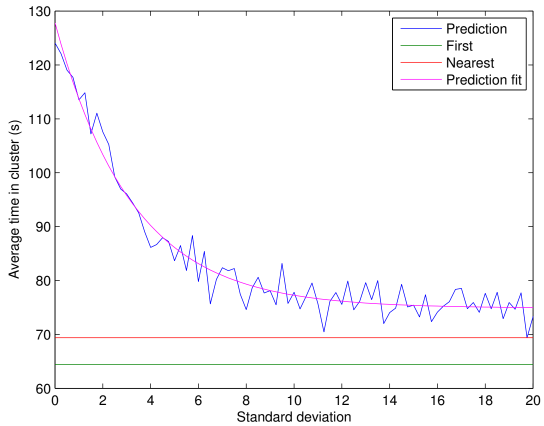

Fig. 5. Location error vs. average time in cluster for the *Linear Mobility Model*.

Fig. 6 shows that *Movement Prediction Algorithm* is still performing better than the two reference algorithms, but the difference is not as big as for the case of perfect prediction (see Fig. 7).

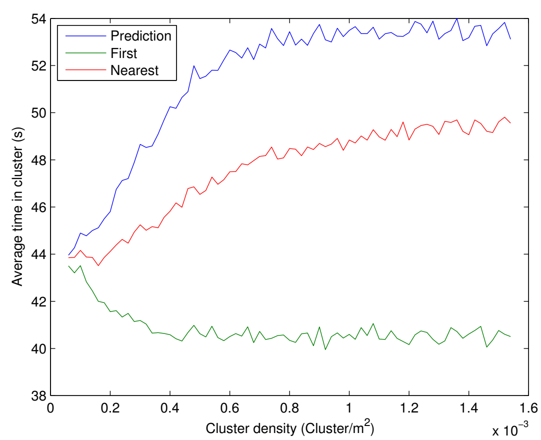

Fig. 6. Cluster density vs. average time in cluster for the *Gauss-Markov Mobility Model*.

V. Conclusion

In this paper we have considered location based cluster selection and analyzed the impact of inaccurate movement prediction and inaccurate location estimation. We implemented a simulation of a simple location based cluster selection protocol in matlab for evaluation. This protocol was compared to two reference schemes: when a considered node associates with either the first discovered or the nearest cluster. Further, we considered both a linear mobility model and a non-linear Gauss Markov mobility model.

Our results have shown that the location based scheme is superior to the simple schemes in terms of providing the longest average time in cluster for the least amount of communication overhead. This is especially true for the ideal situation

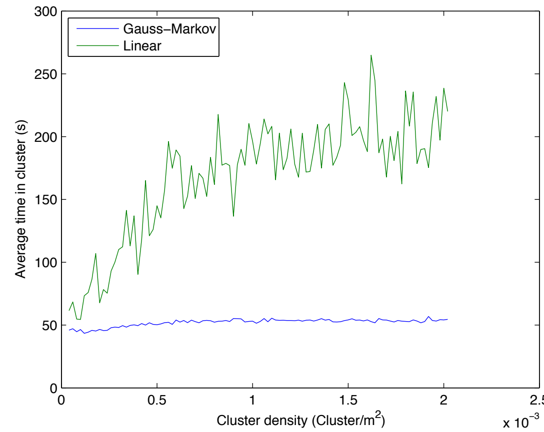

Fig. 7. The average time in clusters for the *Movement Prediction Model*, using both the linear and the *Gauss-Markov Mobility Model*.

of perfect movement prediction and no location estimation errors. For increasing location errors the performance of the location based scheme dropped, converging to the level of the simple schemes for a Gaussian zero-mean location error with std. dev. of around 8 m. For the non-linear mobility model, the performance of the location based algorithm was only slightly better than the two other algorithms, due to the inability of the linear movement prediction to cope with the non-linear movement patterns. This emphasizes the need for a good movement prediction algorithm. Generally, a location-based scheme will require more communication messages to be sent when a node joins a cluster. However, if knowing location information can help to minimize the need for cluster re-selection, overall communication overhead is kept low. This approach therefore presents an interesting trade-off and is beneficial when future trajectories can be well predicted.

References

[1] K. Liu, J. Su, J. Zhang, F. Liu, and C. Gong, "A novel stable cluster protocol for mobile ad hoc networks," in *Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications, 2005. MAPE 2005. IEEE International Symposium on*, vol. 2, aug. 2005, pp. 1328 –1332 Vol. 2.

[2] B. An and S. Papavassiliou, "A mobility-based clustering approach to support mobility management and multicast routing in mobile ad-hoc wireless networks," *International Journal of Network Management*, vol. 11, no. 6, pp. 387–395, 2001.

[3] S. Leng, L. Zhang, and Y. Chen, "k-hop compound metric based clustering scheme for ad hoc networks," in *Communications, 2005. ICC 2005. 2005 IEEE International Conference on*, vol. 5, may 2005, pp. 3396 – 3400 Vol. 5.

[4] S. Leng, Y. Zhang, H.-H. Chen, L. Zhang, and K. Liu, "A novel k-hop compound metric based clustering scheme for ad hoc wireless networks," *Wireless Communications, IEEE Transactions on*, vol. 8, no. 1, pp. 367 –375, jan. 2009.

[5] M. Handy, M. Haase, and D. Timmermann, "Low energy adaptive clustering hierarchy with deterministic cluster-head selection," in *Mobile and Wireless Communications Network, 2002. 4th International Workshop on*. IEEE, 2002, pp. 368–372.

[6] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications & Mobile Computing (WCMC): Special issue on mobile ad-hoc networking: Research, Trends and Applications*, vol. 2, pp. 483–502, 2002.