OPTIMIZING DISTRIBUTED COLLABORATIVE FILTERING IN MOBILE NETWORKS

Patrick Gratz and Adrian Andronache
University of Luxembourg
Faculty of Science, Technology and Communication (FSTC)
L-1359 Luxembourg
E-mail:{patrick.gratz, adrian.andronache }@uni.lu

KEYWORDS

Mobile networks, clustering, distributed, collaborative filtering.

ABSTRACT

Collaborative filtering mechanisms filter information by using collaboration among multiple data sources, typically involving large data sets. In this work we optimize the communication and computational load of a distributed collaborative filtering protocol designed to augment the information exchange in mobile networks.

INTRODUCTION

Recommender systems using collaborative filtering (CF) are a popular technique for information overload reduction and desired data discovery on the Internet. To achieve this, the user is provided by the collaborative filtering system with recommendations or predictions on data items based on the opinions of other *like-minded* users (Sarwar et al. 2001). The opinions on the data items can be obtained explicitly from users by rating or implicitly by counting clicks, viewing time, and alike.

Considering the increase in popularity of mobile devices like smart phones, PDAs and Tablet-PCs the problem of information overload also emerges in mobile networks. This devices are becoming faster in processing, gain more memory capacities, are able to execute more and more powerful applications and at the same time being equipped with various wireless and/or cellular communication capabilities.

In a previous work we introduced an ad hoc collaborative filtering mechanism designed to augment the information exchange in mobile hybrid networks (Gratz et al. 2008). The introduced application was designed for residents and tourists of big cities where many people—potential mobile device users—meet at train or metro stations, at universities and schools, in shopping centers, restaurants, and pubs. In this scenario the shopping centers of the city provide podcasts about offers of the day that contain multimedia files showing the products. The restaurants are podcasting the menu of the day also containing multimedia about the food, drinks and location. The podcasts of the theatres are containing information about tonight's shows and highlights and the podcasts of the discotheques are informing about events and party mottos presenting the DJ's and the music that will be played. The mobile device users in the city can subscribe to the podcasts of the favorite restaurants, theatres, shopping centers etc., thus being up to date about current offers and events. They can subscribe and download the podcasts on computers with broadband Internet connection and download them onto their mobile device, which is the current common procedure. In contrast to this, our application aims to augment the podcast providing mechanism by enabling the mobile devices to exchange episodes of subscribed podcasts, to recommend similar, well-rated podcasts and to subscribe to new ones in an ad-hoc mobile environment. Additionally, devices with a cellular network interface are able to pull new podcast episodes from the Internet servers and share them locally within ad-hoc networks.

Imagine a traveling user meeting other people in metros, trains, shopping centers, pubs or restaurants. Some of them might have recently downloaded new podcast episodes either via cellular uplinks or by copying them from desktop computers onto their mobile device. The client on the traveler's device will group up with the subscribers of similar podcasts and exchange new or missing episodes. Thus, the traveler will be able to receive podcasts on the way without having to use an Internet connection.

The application can be used for instance by a tourist which arrives in a big city and does not have knowledge about the local podcast providers. He will meet other tourists and residents with mobile devices at the airport, train or metro station, shopping centers, pubs and restaurants. The client on the tourist's mobile device will search the ad-hoc network for podcasts and provide the information to its user. Thus, the tourist is able to choose among the local podcasts about restaurants that offer the favorite food, locations and events he may like or shopping offers that can be interesting for him. By subscribing to the podcasts of interest found in the mobile ad-hoc network, the tourist will receive up to date information about offers and events in the city during the stay.

The collaborative filtering mechanism introduced in (Gratz et al. 2008) enables the user to get recommendations about the local offers based on the taste of other like-minded users in nearby mobile environments. The ability to get recommendations from the local ad-hoc network has the advantage that the

Proceedings of the 2008 High Performance
Computing & Simulation Conference ©ECMS
Waleed W. Smari (Ed.)
ISBN: 978-0-9553018-7-2 / ISBN: 978-0-9553018-6-5 (CD)

system can take into account new information without to require a connection to a central repository on the Internet. Thus, the mechanism provides updated recommendations even if the user has no backbone connection by a cellular network or access point. Furthermore, all ratings made by a user on the way, e.g. for the menu of the day in a restaurant will instantly have an impact on the calculated recommendations for like-minded users in the local vicinity.

In (Gratz et al. 2008) two algorithms to determine sets of similar neighbors were introduced. The first one called Hierarchical Cluster-based Neighborhood Resolution (HCNR) uses a weighted cluster topology generated by the Weighted Application Aware Clustering Algorithm (WACA) presented in (Andronache et al. 2006). The second algorithm—Weighted Neighborhood Resolution Algorithm (WNR)—is based on a simple peer-to-peer communication pattern. In order to compare the introduced algorithms for proper operation they have been implemented and tested on top of the JANE simulator by performing several experiments. As the results of these experiments show, HCNR provides a slightly better precision and scales distinctly better concerning the number of sent unicasts if we increase the network connectivity. However, WNR has the advantage that the used bandwidth is very small compared to HCNR.

In this paper we aim to optimize the communication and computational load of the distributed, cluster-based algorithm HCNR by using another, better suited clustering protocol.

The remainder of the paper is structured like follows: the next section presents related work. Section III briefly introduces the Hierarchical Cluster-based Neighborhood Resolution (HCNR) algorithm. In Section IV the clustering algorithm is described which is employed to optimize HCNR. Section V describes the simulation setups and Section VI presents the results. The paper concludes with future work in Section VII.

RELATED WORK

Recommender systems using collaborative filtering are one of the most successful recommendation techniques (Resnick et al. 1994; Shardanand and Maes 1995). In this section we introduce some existing research work about collaborative filtering in mobile environments.

In (Cöster and Svensson 2005) an incremental collaborative filtering algorithm for applications, where users are occasionally connected to a central server is introduced. The general idea is to store a subset of selected user profiles, together with a ranked list of predictions. When the user is in offline mode, a service on the local device can still recommend items based on the predictions made the last time the user was connected. Each time the user supplies new ratings, the list of predictions will be recomputed, even if the user is not connected to the server. In the case that a user encounters another user, the authors suggest that they exchange their profiles and recalculate their prediction lists. The past influence of the other user should be removed from all predictions and the new influences should be added. At last this case is not evaluated or considered any further in the paper and is a part of future work.

A further portable recommender system along with five peer-to-peer architectures for finding neighbors is presented in (Miller et al. 2004). The authors introduce a new collaborative filtering algorithm called PocketLens that can run on connected servers, on usually connected workstations or occasionally connected portable devices.

The presented algorithm is a variant of the item-item algorithm introduced in (Cöster and Svensson 2005) with modifications for a peer-to-peer environment. To reach the goal of portability a local similarity model is created for the user. Thereby, the algorithm only needs access to the ratings of the owner and one other user at a time. In this manner, the model is created incrementally in a distributed fashion.

In (Tveid 2001) an approach for making a scalable recommendation system for mobile commerce using P2P is considered. The main idea of the proposed approach is to transform the problem of finding recommendations using collaborative filtering, into a search problem in scalable P2P systems like Freenet or Gnutella. Thereby, a query (vector with votes on products) is broadcasted from the querying node to all neighbor peers. When a peer receives a query it calculates the proximity with other cached queries. If the proximity is higher than a threshold, the cached voting vector is sent back otherwise the query is broadcasted further. For sparse voting vectors the authors propose a binary interpolative compression algorithm. Furthermore, to improve the performance and quality of recommendations they propose an approach for clustering similar peers.

An approach to collaborative filtering in a mobile tourist information system for visitors of a festival based on spatio-temporal proximity in social contexts is proposed in (Spindler et al. 2006). This new approach is based on the idea that users who go to the same place at the same time tend to have similar tastes. In order to keep track about the visited places each user is equipped with a portable computer coupled with a GPS unit. Furthermore, a central server provides a database with information about all the events, restaurants, venues and bars at the festival (Belotti et al. 2005).

The proposed approach uses a user-based CF technique and calculates similar users via a spatio-temporal proximity measure, i.e. two users are considered as similar if they consume the same items simultaneously. The following exchange of rating information between such similar users is done via an ad-hoc peer-to-peer interaction. However, the defined similarity measure has one drawback. Users consuming the same periodic event at different times still share interests, but are not considered as similar. In a future work, the authors intend to investigate how their CF approach can be extended in order to exchange ratings between users in

spatial but not temporal proximity. Furthermore, they want to evaluate the introduced CF system at the Edinburgh Fringe festival.
(Jacobsson et al. 2006) introduce an approach for a mobile recommender system where media can find people rather than the other way around. Whereas, media files are autonomous, rule-following agents capable of building their own identities from interactions with other agents and users. The general idea is that the interaction of large ensembles of those interacting agents, distributed over mobile devices in social networks can emerge a collaborative filtering-like behavior.

HIERARCHICAL CLUSTER-BASED NEIGHBORHOOD RESOLUTION (HCNR)

The collaborative filtering recommender process can be roughly divided into three phases: determine similar neighborhood, update the recommender model, and calculate a prediction.
In order to deliver good recommendations a typical collaborative filtering system depends on a critical mass of users with commonly rated items. However, in our application scenario it is very likely that a tourist who visits a certain city for the first time has no commonly rated podcasts with users in his nearby environment. Nevertheless, this fact does not except that the tourist has no similar taste with other users in the local neighborhood. The tourist can have rated different podcasts that are similar concerning the content of those in the nearby neighborhood. For this purpose we calculated the similarity between two users based on an approach proposed by (Pazzani 1999) called collaboration via content. The idea behind this approach is to exploit a content-based profile for each user in order to calculate the similarity between two users via their content-based profiles instead of their commonly rated items. In the context of our application scenario the content-based profile is represented by a list of weighted keywords. For this purpose we presume that each podcast feed contains a set of keywords describing its content. Given the corresponding rating values $r(p)$ for all podcasts P_i rated by a specified user u_i together with the appropriate set of keywords K_i describing these podcasts, a weight for each keyword $k_j \in K_i$ can be calculated as follows:

$$w_j = \frac{\sum_p r(p)}{occur(k_j)}, \quad p \in P_i \wedge p \ni k_j$$

The function $occur(k_j)$ returns the number of podcasts containing keyword k_j. Thus, for each user u_i a vector of weighted keywords $wk_i = \langle (k_{i1}, w_{i1}), \ldots, (k_{in}, w_{in}) \rangle$ can be calculated, that represent his preferences. Given these content based profiles we can define the similarity between two users u_i, u_j via the cosine between the corresponding weighted keyword vectors:

$$sim(u_i, u_j) = \frac{\sum_{k \in K_{ij}} w_{ik} w_{jk}}{\sqrt{\sum_{k \in K_{ij}} w_{ik}^2} \sqrt{\sum_{k \in K_{ij}} w_{jk}^2}}, \; K_{ij} = K_i \cup K_j$$

The HCNR algorithm uses a weighted cluster topology generated by Weighted Application Aware Clustering Algorithm (WACA) (Andronache et al. 2006). WACA creates clusters in a hierarchical fashion. Each device elects exactly one device as its clusterhead, i.e. the neighbor with the highest weight. This clusterhead also investigates its one-hop neighborhood, similarly electing the device with the highest weight as its clusterhead. This process terminates in case of a device electing 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 (Figure 1).

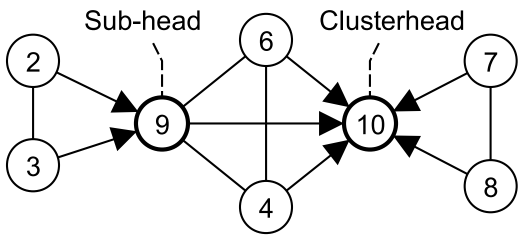

Figure 1: Topology built by WACA

Based on the WACA topology, the algorithm works as follows. At first, in order to determine a set of similar neighbors, each slave sends its own profile to the currently elected clusterhead. To keep the message complexity low, each device maintains a list of cluster-heads that have already received the current profile. As soon as the own profile changes this list will be cleared.
After receiving the profiles from its slaves, the clusterhead and all sub-heads calculate a similarity matrix via the received profiles. Subsequent to this calculation, the sub-head sends the calculated similarity matrix and the list of received profiles to the cluster-head. The cluster-head stores this profile list together with the corresponding similarity values and calculates the similarity values for all missing pairs in order to complete the similarity matrix. Figure 2 shows how the algorithm is using the cluster topology to exchange the information.

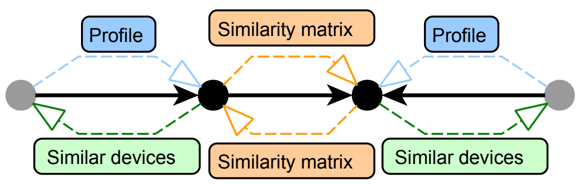

Figure 2: HCNR similarity matrix calculation and similar neighbor discovery

By doing so, the matrix on the clusterhead stores the similarities between all users connected to the current cluster. Finally, in order to determine the similar neighborhood for all slaves that are not directly connected with the clusterhead a copy of this similarity matrix is replicated to each sub-head in the cluster. The cluster-heads and the sub-heads provide a list of similar neighbors to each slave in the current cluster.

HCNR OPTIMIZATION

A drawback of WACA is the fact that the algorithm provides no mechanism to avoid the re-organization of two crossing clusters. One can imagine a group of mobile device users traveling by bus. The applications on the mobile devices use WACA to build clusters. On the way, the clusters may meet other clusters build for instance by devices in another bus waiting at the same traffic light. In this scenario WACA re-organizes the crossing clusters, which is needless since the devices build the initial clusters after the busses pass each other.
The HCNR algorithm works on top of a clustered mobile network, thus the robustness of the topology has a crucial impact on the communication and computational complexity of the collaborative filtering protocol.
In (Andronache et al. 2008) the Node and Link Weighted Clustering Algorithm—NLWCA—was introduced. The algorithm is designed to protect stable clusters from re-organization in order to avoid needless network communication.
Unlike WACA, which uses only the weight of the nodes to elect a local clusterheads, NLWCA also assigns weights to the links between the own node and the network neighbor nodes. This weight is used to keep track of the connection stability to the one-hop network neighbors. The algorithm increases the weight of the links to neighbors that are for a longer time in communication range. 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 (Figure 3).

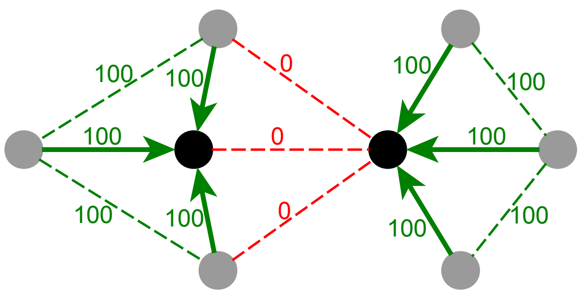

Figure 3. The low weight of the links avoids superfluous re-organization of the topology when for instance two clusters are cross in mobile networks.

The simulation results in Figure 4 and 5 show that NLWCA outperforms WACA in terms of number of re-elections independent of the used speed and node number. More simulation results and detailed results can be found in (Andronache et al. 2008).
Motivated by the good results in terms of topology stability obtained by the NLWCA, the algorithm was employed to optimize the communication of the HCNR protocol.

SIMULATIONS SETUP

In order to observe if the network and computational load of the HCNR protocol is improved on top of the cluster topology build by NLWCA, we performed several simulation runs using the JANE simulator (Görgen et al. 2007).
For our experiments, we used the MovieLens Data Set (available at: http://www.grouplens.org) that consists of 100,000 ratings for 1682 movies by 943 users, where each user has at least rated 20 movies.

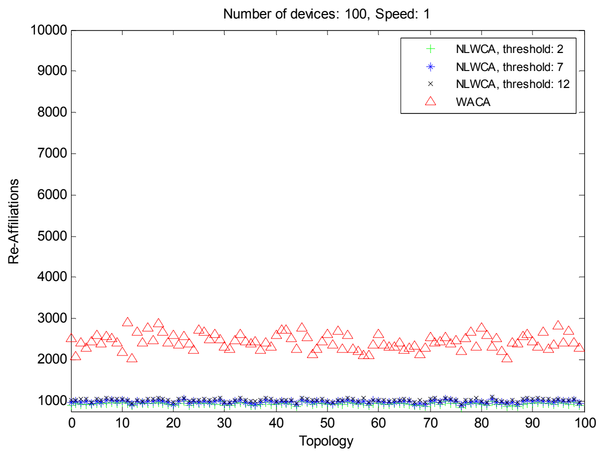

Figure 4: Results for 100 mobile device moving with 1m/s during 5 minutes on an area of 300x300m with sending range of 40m. NLWCA outperforms WACA in terms of re-elections number independent from the used stability threshold.

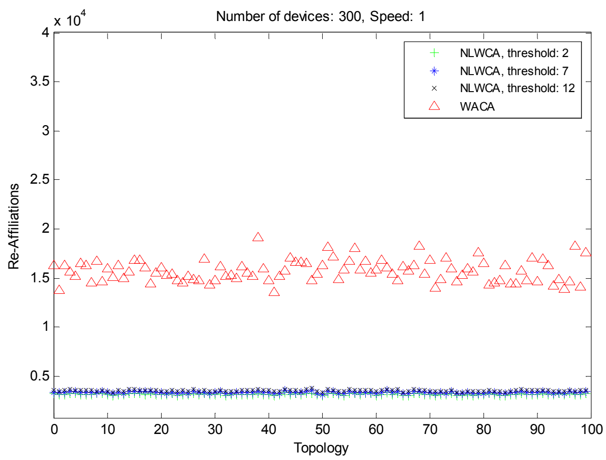

Figure 5: Results for 300 mobile devices using the same settings as above. Also in the dense networks, NLWCA outperforms WACA in terms of re-affiliation of nodes to new clusterheads.

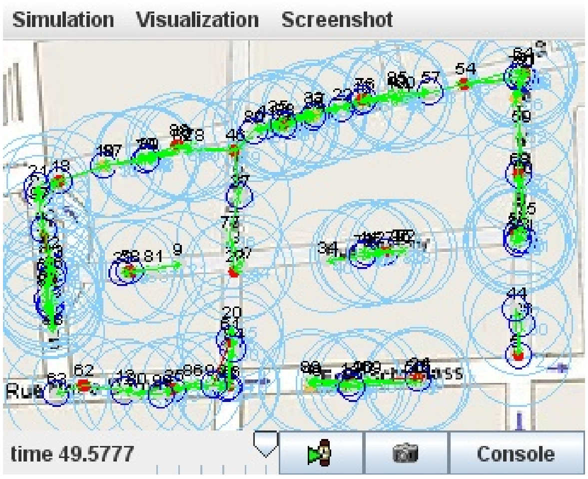

Figure 6: JANE simulating HCNR on top of NLWCA running on 100 devices. The mobile devices move on the streets of the Luxembourg City map. The devices move with a speed of 0.5 – 1.5 m/s (1,8 – 5,4 km/h)

We generated 5 different training sets containing 15 votes that are considered as the observed votes for each user and 5 different test sets containing the remaining votes. After each simulation run we compare the predicted votes with the corresponding votes in the test set and calculated the Mean Absolute Error (MAE), which has been used to measure prediction performance in several cases (Shardandand and Maes 1995; Breese et al. 1998; Herlocker et al. 1999). If a predicted item did not have an adequate entry in the test set it was eliminated from the evaluation. Note that we used the MAE only to compare how accurately our algorithms predict a randomly selected item rather than evaluating the user experience of generated recommendations.

For each experiment we used the Restricted Random Way Point mobility model (Blazevic et al. 2002), whereby the devices move along defined streets on the map of Luxembourg city for 5 minutes and 30 seconds (Figure 6). For each device the speed was randomly varied between [0.5;1.5] units/s in a first run and [11;16] units/s in a second run. While, every time a device reaches a crossroad, it randomly selects a street to turn in at next.

At startup, the devices are positioned at random selected crossroads and initialized with 15 selected votes in order to calculate an initial user profile. In order to avoid a data exchange at this point, where the devices are already strongly clustered at the crossroads, we delay the startup of our HCNR algorithm via a timeout of 30 seconds. After this timeout the devices begin to exchange their profiles in order to determine the k-most similar neighbors. For all experiments we simulated with 5 different training sets and 5 different topologies per training set. Furthermore, we selected for NLWCA a stability threshold of 2 for speed [0.5;1.5] and 12 for speed [11;16]. All results are shown with 95% confidence intervals.

RESULTS

Figure 7 shows the overall computed similarities after 5 minutes of simulation. As the figure shows, using NLWCA distinctly reduces the number of computed similarities in dense network settings. In particular, if the devices move with a speed between 11 and 16 units/s WACA produces nearly 3 times more similarity calculations as NLWCA. Due to the fact, that NLWCA produces provable lesser re-elections than WACA this result was expected. Since, each time a new cluster-head is elected this cluster-head potentially has to calculate similarities again.

In Figure 8, we illustrate the number of sent unicasts during the simulation. Again the use of NLWCA reduces overhead. Thus, for each setting HCNR using NLWCA needs constantly lesser unicasts than using WACA. However, more considerable is the diversity of the used bandwidth. As shown in Figure 9, in dense network settings WACA used nearly up to 2 times more bandwidth at a device speed between 0.5 – 1.5 units/s and about 7 times more bandwidth if the device move with a speed between 11 – 16 units/s.

However, all these communication and computational optimization comes at a cost.

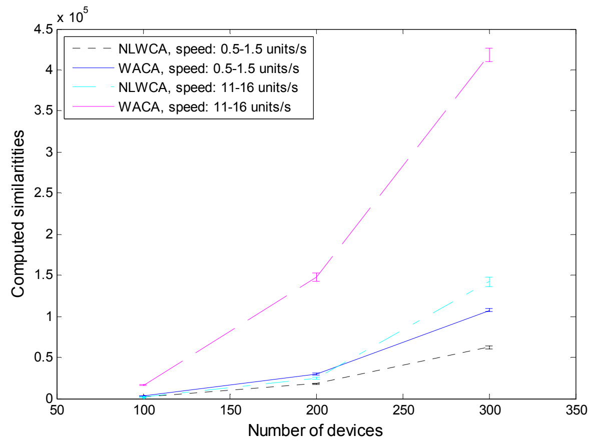

Figure 7: Overall computed similarities

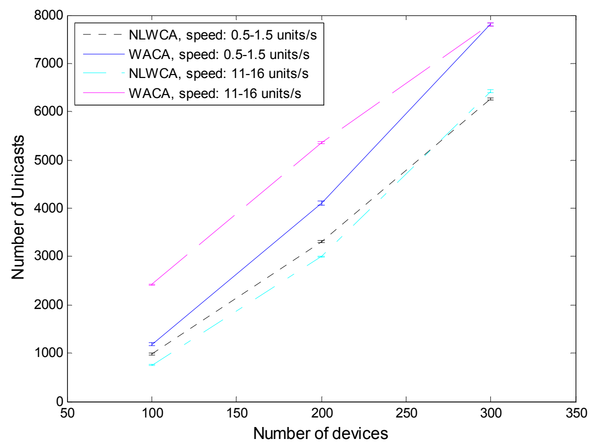

Figure 8: Number of sent messages

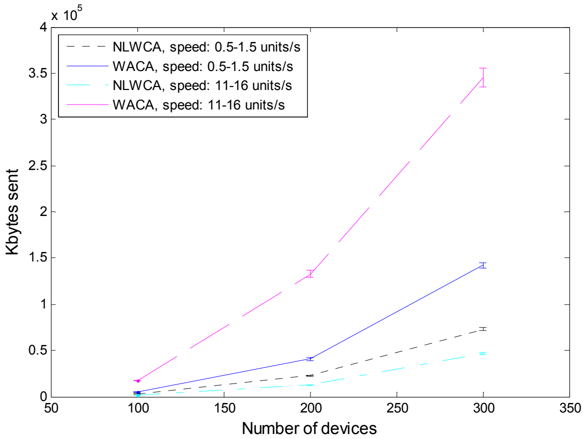

Figure 9: Bandwidth usage

As Figure 10 shows, the MAE of the calculated predictions, with 300 mobile devices is about 1.8% respectively 6.4 % better when using WACA concerning the two different speed intervals. Again, this result was expected due to the fact, that NLWCA allows only communication between devices building stable clusters based on a defined stability threshold.

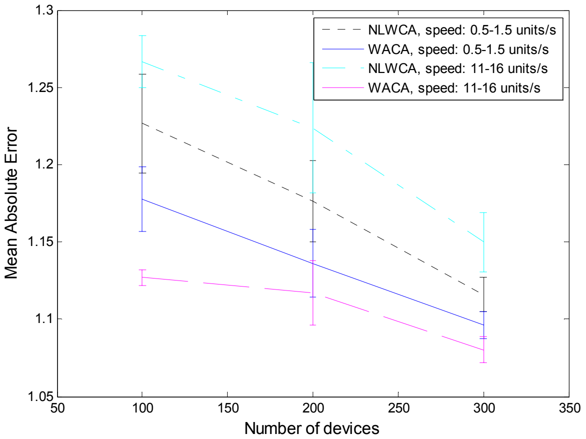

Figure 10: Mean absolute error of calculated predictions

Thus, crossing devices, which are only in the communication range for a short period of time, will be not part of the cluster and hence, do not participate on a profile exchange. Since, the precision of calculated predictions also depends on the number of discovered similar neighbors, it was expected that the precision of the calculated predictions will be inferior when using NLWCA. Nevertheless, using NLWCA instead of WACA significantly reduces the overhead concerning communication and computational load at the cost of a relatively small loss in precision.

CONCLUSION AND FUTURE WORK

In this work we optimized HCNR in order to reduce the communication and computational overhead of the collaborative filtering based recommender system, which was designed to overcome the potential problem of information overload in mobile ad hoc networks. To achieve this goal we employed a network link stability aware clustering protocol, which provides HCNR with a better suited topology than the previous used mechanism WACA.

Several simulations were done using the JANE simulator and the results show that the new mechanism significant improves the communication and computational load on the network nodes. However, the prediction precision is slightly lowered since the devices communicate only with the set of neighbors that are considered to be stable connected.

In order to improve the prediction precision, in future work we will employ an inter-cluster communication protocol, thus enabling an inter-cluster similarity matrix exchange.

REFERENCES

Andronache, A., M.R. Brust and S. Rothkugel. 2006. "Multimedia Content Distribution in Hybrid Wireless Networks using Weighted Clustering". In *Proceedings of the 2nd ACM Workshop on Wireless Multimedia Networking and Performance Modeling (WMuNeP'06)*, Torromolinos, Spain, 1-19.

Andronache, A., M.R. Brust and S. Rothkugel. 2008. "NLWCA—Node and Link Weighted Clustering Algorithm for Backbone-assisted Mobile Ad Hoc Networks". *The Seventh International Conference on Networking (ICN)*, Cancun, Mexico, 460-467.

Belotti, R., C. Decurtins, M.C. Norrie, B. Signer, and L. Vukelja. 2005. "Experimental Platform for Mobile Information Systems". In *Proceedings of the 11th Annual International Conference on Mobile Computing and Networking (MobiCom 2005)*, Cologne, Germany, 258-269.

Blazevic, L., S. Giordano, J. Y. Le Boudec. 2002. "Self Organized Terminode Routing". In *Cluster Computing Vol.5, No. 2,* Springer Netherlands, 205-218.

Breese, J.S., D. Heckerman and C. Kadie. 1998. "Empirical analysis of predictive algorithms for collaborative filtering". In *Proceedings of the 14th Conference on Uncertainty in Artifical Intelligence (UAI-98)*, 43-52.

Cöster, R. and M. Svensson. 2005. "Incremental Collaborative Filtering for Mobile Devices". In *Proceedings of the 2005 ACM symposium on Applied Computing* (*SAC'05),* Santa Fe, New Mexico, USA, 1102-1106.

Görgen, D., H. Frey and C. Hiedels. 2007. "JANE-The Java Ad Hoc Network Development Environment". In *40th Annual Simulation Symposium (ANSS'07),* 163-176.

Gratz, P., A. Andronache and S. Rothkugel. 2008. "Ad Hoc Collaborative Filtering for Mobile Networks". *The Second IEEE International Workshop on Ad Hoc and Ubiquitous Computing (AHUC),* Taichung, Taiwan.

Herlocker, J. L., J.A. Konstan, A. Borchers, and J. Riedl. 1999. "An algorithmic framework for performing collaborative filtering". In *Proceedings of the 22nd International Conference on Research and Development in Information Retrieval (SIGIR'99),* 230-237.

Jacobsson, M., M. Rost and L.E. Holmquist. 2006. "When Media Gets Wise: Collaborative Filtering with Mobile Media Agents". In *Proceedings of the 11th International Conference on Intelligent User Interfaces* (*IUI'06),* Sydney, Australia, 291-293.

Miller, B.N., J. A. Konstan, and J. Riedl. 2004. "PocketLens: Toward a Personal Recommender System". In *ACM*

Transactions on Information Systems, *Vol. 22, No. 3*, 437-476.

Pazzani, M.J. 1999. "A Framework for Collaborative, Content-Based and Demographic Filtering". In *Artificial Intelligence Review*, 393-408.

Resnick, P., N. Iacovou, M. Suchak, P. Bergstrom, and J.Riedl. 1994. "Grouplens: An open architecture for collaborative filtering of netnews". I*n Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, Chapel Hill, North Carolina: ACM, 175-186.

Sarwar, B., G. Karypis, J. Konstan and J. Riedl. 2001. "Item-based Collaborative Filtering Recommendation Algorithms". In *Proceedings of the 10th International World Wide Web Conference (WWW10)*, Hong Kong, Hong Kong, 285-295.

Shardanand, U. and P. Maes. 1995. "Social information filtering: Algorithms for automating "word of mouth"". In *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, vol. 1, 210-217.

Spindler, de A., M.C. Norie, M. Grossniklaus and B. Signer. 2006. "Spatio-Temporal Proximity as a Basis for Collaborative Filtering in Mobile Environments". In *Workshop on Ubiquitous Mobile Information and Collaboration Systems (UMICS 2006),* Luxembourg, Grand Duchy of Luxembourg.

Tveid, A. 2001. "Peer-to-peer based Recommendations for Mobile Commerce". In *Proceedings of the 1st International Workshop on Mobile Commerce* (*WMC 01,* Rome, Italy, 26-29.