

Intelligent Reduction in Signaling Load of Location Management in Mobile Data Networks

Kashif Munir

FCIT, King Abdulaziz University, Jeddah, 21589, KSA
E-mail: kmunir@kau.edu.sa

Ehtesham Zahoor, Waseem Shahzad, Syed Junaid Hussain

National University of Computer and Emerging Sciences, Islamabad, 44000, Pakistan
E-mail: {Ehtesham.Zahoor, Waseem.Shahzad}@nu.edu.pk, junaidjanus@gmail.com

Abstract—Massive increase in the mobile data traffic volume has recently resulted in a big interest towards the distributed mobility management solutions that aim to address the limitations and drawbacks of centralized mobility management. Location management is an important requirement in a distributed mobility management environment. To provide seamless Internet data services to a mobile node, the location of a mobile node is stored and periodically updated on a location server through a location update message that is sent by the mobile node. In this paper, we propose an intelligent approach of setting the period of sending location update messages on the basis of a mobile node's patterns of data sessions and IP handovers. We use a machine learning approach on the location server. The results show that our approach significantly reduces the signaling load of the location management and the overall reduction is more than 50%.

Index Terms—Location management, distributed mobility management, mobile network, location signaling, machine learning.

I. INTRODUCTION

Location management [1] is a primary concern in mobility management services to make it possible for a Mobile Node (MN) to uninterruptedly receive data services while moving from one location to another.

For location management, the current location of an MN is stored and periodically updated in a Location Server (LS). Any Correspondent Node (CN) can get the current location of an MN by sending a request to the LS.

The idea is to implement an intelligent prediction mechanism on the LS, which can intelligently fine-tune the frequency of Location Update Messages (LUMs) from the MN to the LS. The “intelligent” message for adapting the pace of sending the LUMs will be sent to an MN by the LS, whenever there is a need to change the period of update messages for a particular MN.

The LS can record the following information related to an MN:

- i. The location query (from a CN) about the MN

- ii. Its location (obtained from the LUM)

Fig. 1 represents the simplest layout of a CN functionality of requesting and getting location (IP address) of a distinct MN, functionality of LS that how an MN updates its location and how it is passed on to a CN, and finally, the functionality of an MN that how it updates its location to the LS and how it is being contacted by the CN.

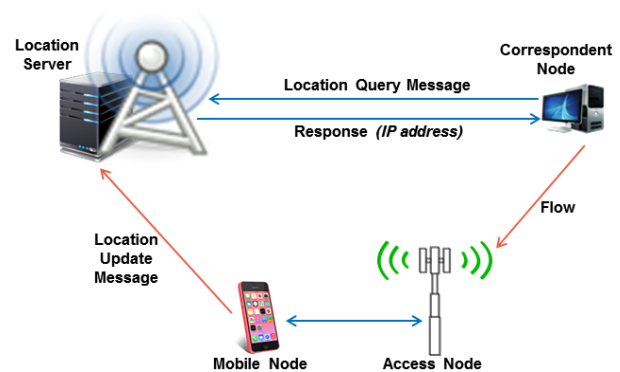


Fig.1. Basic Layout for LUM, CN and LS Functionality

From the location queries, the LS can calculate how often the MN is being contacted (by CNs) for creation of the data sessions. From the LUMs, the LS can measure how often the location of the MN is updated (measuring its mobility pattern). An MN registers the same Access Node (AN) in the LUM, if it stays attached to the same AN. However, as it is mobile, it keeps on changing the ANs. We define AN-Area as the area covered by an AN.

The idea is to have a Machine Learning (ML) [2] based intelligent prediction solution that can reduce the LUMs from those MNs which are:

- i. not mobile (attached to the same AN),
- ii. having low mobility (low or mild AN change pattern), and
- iii. having very few or no data sessions.

The ML tools, techniques and algorithms are implemented to reduce the cost of LUMs. Data Mining (DM) is a field of soft computing in which hidden

patterns from data are extracted and usually these patterns are unknown to the user [2].

We intend to minimize LUMs for those MNs which do not utilize data services or having low or no mobility. We formulate distinct classes for the classification of MNs into categories and then we send a message to MNs (which have low mobility and fewer or no sessions) in order to change their LUM frequency.

The rest of the paper is organized as follows. Section II discusses the related literature. Section III explains the proposed technique. The evaluation has been done in Section IV, and finally, the conclusions have been drawn in Section V.

II. RELATED WORK

Several location management schemes have been proposed in the literature. However, we discuss some approaches related to our proposed approach.

A dynamic location management scheme is presented in [3] for the low mobility rate of an MN and a static scheme is presented for the high mobility rate. Furthermore, the authors in [3] claim that the dynamic scheme is better than the static scheme for high paging cost or when the number of cells in an LA is large. Our work is different from [3], as we use an intelligent mechanism on an LS (Location Server) to reduce the number of LUMs (Location Update Messages) sent by an MN to the LS.

Several metrics involved with location management are described in [4]. Some of the metrics are location update and paging costs. The static schemes described in [4] are based on the topology of the network, whereas the dynamic schemes are based on time, movement, or distance. The different location management schemes are also classified in [5]. A simulation environment is described in [5] with a claim that Location Area based location management using a profile or history-based direction information gives the best performance in terms of location management cost.

In [6], the authors have used an Intelligent Agent Quantization (IAQ) approach, which is based on the prediction of movements and the distance between node and BSC to locate the MNs.

A cluster based technique for the mobile location management is presented in [7]. The technique exploits the data available on the users (age, function, address, workplace etc.), existing infrastructure (roads, location of base stations etc.), and the users' displacements history.

A multilayer neural network technique is described in [8] to solve the location management problem. The model predicts the future prediction of the mobile subscriber based on its past predicted information.

The authors in [9] have proposed a distributed and dynamic location management scheme to reduce the signaling traffic and signaling delay as compared those of the IETF Mobile IP. Depending on the up-to-date mobility and traffic load for each MN, the regional network boundaries are dynamically adjusted in order to distribute the signaling burden.

An improved DHT-based handover management mechanism was proposed in [10]. A distributed localization service for wireless mesh networks is also proposed in [11]. To reduce the packet loss due to IP handovers, a mechanism utilizing a mobile stream control transmission protocol (mSCTP) for improved transport and a Chord DHT [12] for location management was proposed in [13]. In [14], a two-tier Chord DHT for distributed location management was introduced that uses the nodes with high computation power and stability as location servers.

The contribution of our work compared to the related works is that we use an intelligent machine learning approach on the LS to reduce the number of LUMs that are sent by an MN to the LS. The proposed approach considers the sessions arrival activity on an MN as well as its activity related to the mobility related IP handovers.

III. THE PROPOSED MECHANISM

In this section, we describe the proposed mechanism to fine-tune the expected period of sending LUMs to those MNs that are not actively changing ANs and at the same time not having many data sessions or the MNs that do not have many incoming sessions but changing ANs often or the MNs that have many incoming sessions but they are passive or not actively changing ANs.

A. Traffic Model

The real data of mobile subscribers are not available due to privacy concerns. Even for research purposes, the telecom operators do not provide real data of their mobile subscribers. Also, we propose this reduction in LUMs for DMM [15], which is still in the experimental phase. So, we generate the input data using the known models for mobile networks.

As explained in Section I, the LS stores current location of an MN. A CN queries this location in order to create a data session on the MN. We generate random variables for incoming sessions on the MN and random variables for AN-Area handovers (IP handovers).

There are two types of logs. One is the exponentially distributed incoming sessions (started by a CN) on an MN and the other is the exponentially distributed AN-Area residence times (time spent by an MN in an AN-Area) of the MN.

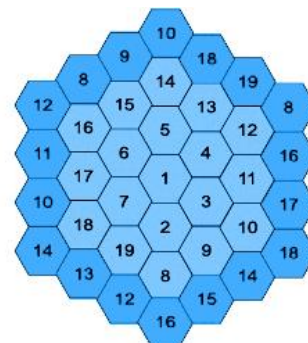


Fig.2. Topology of the Cellular Network

Fig. 2 shows the network topology used in the simulations. The topology consists of 19 ANs which interestingly make a giant hexagon itself. Each hexagon is connected to six other hexagons or ANs. We have considered different scenarios of AN-Areas as shown in Table 1.

Table 1. AN-Area Scenarios

Scenario #	Area of an (km ²)
1	10,000
2	5,000
3	1,000
4	100
5	10
6	1
7	0.01

Scenario 1 depicts Long Term Evolution (LTE) network. Every AN is equivalent to a Serving Gateway. In scenario 6, we consider a set of 3 sectorized radio cells in LTE in an urban area. In scenario 7, we consider the AN to be a wireless access point. Usually, the coverage range of a wireless access point is around 60 meters, so the area of the AN in this case is 0.01 square km. Scenarios 2-5 are just intermediate scenarios.

B. Parameter Settings for Incoming Sessions and AN-Area Residence Times

We generate inter-arrival time of incoming sessions using Exponential distribution. We generate the session duration using Uniform distribution. We set the average session duration to 350 seconds in our simulations. We use Exponential distribution to generate AN-Area residence times. The chosen parameter settings for the generation of the data and the mobility related traffic are consistent with the ones used in the related literature [16-25].

In case of incoming sessions, there is only one important factor that is to be considered and it is the arrival rate of sessions. But, in case of AN residence times, there are many things to be taken into account. The most important aspect is the area of an AN. If AN-Area is very large but an MN stays inside the boundary of the AN, then in this case, the LS will not record change of AN. On the other hand, if AN-Area is very small; then there can be frequent AN-Area handovers. Table 1 describes the AN-Area details we use for generating time intervals that an MN spends in an AN.

Table 2. Settings for AN-Area Residence Times; S. Stands For Scenario

Parameter	S. # 1	S. # 2	S. # 3	S. # 4	S. # 5	S. # 6
Area (km ²) of AN	10 ⁴	5×10 ³	1000	100	1	0.01
Average velocity of an MN	1 ms ⁻¹					
Perimeter of the AN-Area	3.72×10 ⁵	2.63×10 ⁵	1.17×10 ⁵	3.72×10 ⁴	3.72×10 ³	3.72×10 ²
AN-Area crossing rate	1.18×10 ⁻⁵	8.37×10 ⁻⁵	3.74×10 ⁻⁵	1.18×10 ⁻⁴	1.18×10 ⁻³	1.18×10 ⁻²

Table 2 describes parameter settings for generating AN residence times for an MN. The first row of the table describes the areas of an AN. The second row explains the average velocity of an MN in the AN. It is considered to be 1 ms⁻¹. The third row shows the perimeter of an AN-Area. The last row shows the AN-Area handover rates consistent with the rates in [26].

C. Preprocessing of Input Data

It is evident that the input data are all numerical values of incoming session arrival times, session durations, and AN-Area handover times, which do not give any idea of an MN's mobility and data usage patterns. So, we preprocess the data to formulate a strategy to extract useful information from the given session data and location data logs.

According to the devised pre-processing strategy, every incoming session is checked for AN-Area handovers and the time interval between two consecutive incoming sessions (to check if they are far apart in terms of time or there is a sudden burst in the incoming sessions for few minutes or hours and then MN becomes idle etc.).

D. Discretizing Interval Between Two Consecutive Incoming Sessions

It is important to understand that all MNs do not use data services all the time. So, there may be some cases in which an MN receives many incoming sessions in a short span of time and then becomes idle. To cater for this immediate burst pattern in the incoming sessions, we devise a strategy to check the inter-arrival time of sessions. On the basis of this time, each MN is classified in a category. We use heuristics and get intuition from [27-32] in the field of ML and data mining.

Table 3. Discretizing Inter-Arrival Time of Sessions

S #	Time Interval Between Two Consecutive Incoming Sessions (minutes)	Discretized Category
1	≥ 240	High
2	≥ 120 and < 240	Moderately High
3	≥ 50 and < 120	Medium
4	≥ 30 and < 50	Moderately Medium
5	> 0 and < 30	Low
6	0	Inactive

Table 3 shows the time interval between two consecutive incoming sessions on an MN. The six categories defined here are the discretized values. By having these specific values, we can easily classify an MN to some distinct category as well as extract the incoming session pattern.

E. Outcome of Preprocessing

After applying miscellaneous filters on the data logs, we are finally at the stage to get the preprocessed data. Preprocessed data is in the form of records. Each record provides information about recently created session, its mobility pattern that suggests whether MN has changed its location from previous AN to current AN, and finally,

the discretized values of inter-arrival times of two successive sessions created on the MN. So, we can conclude that our data is now in the form of information and by applying classification algorithms, we can extract patterns from the preprocessed data. Table 4 depicts the outcome of the preprocessing.

Table 4 summarizes the final outcome of preprocessing of data logs. First of all, there must be an incoming session on the MN from CN, only then the rest of the processing starts. If there are very few incoming sessions or if their number is below a certain threshold, then we can simply classify that an MN as inactive. The discretized values of time interval between two consecutive incoming sessions have already been explained. The last column is reserved for IP handover activity of an MN. It simply distinguishes whether an MN has changed the AN or handover is performed after the last session was created on the MN. By combining all these values, we form a proper record of each session on the MN, which is very healthy information to classify an MN into a category by analyzing simulation data of one year.

Table 4. Outcome of the Preprocessing

S #	Session Created	Time Interval Between Two Consecutive Incoming Sessions	IP Handover
1	Yes	Moderately Medium	Yes
2	Yes	Moderately Medium	No
3	Yes	Medium	Yes
4	Yes	Medium	No
5	Yes	Low	Yes
6	Yes	Low	No
7	Yes	Moderately High	Yes
8	Yes	Moderately High	No
9	Yes	High	Yes
10	Yes	High	No

F. Classification of Preprocessed Data

Once, we preprocess the data generated using theoretical distributions [33], we define a strategy to categorize this data into the six predefined distinct classes. Every MN will be assigned a distinct category through the ML process. The criteria for the selection of a class for an MN is based on checking every record of incoming session, AN-Area handover activity, and immediate burst of sessions (the discretized inter-arrival times of two sessions on an MN).

The classes are as follows:

- I. Highly active
- II. Active
- III. Moderately active
- IV. Somehow active
- V. Rarely active
- VI. Inactive

The six classes mentioned above are categorizing an MN from the highest mobility and incoming session activity to the lowest mobility and little or no incoming session activity. Only six classes are defined to cover

every possible scenario. More than six classes can increase the computational cost of the algorithm. The classification categories are closely adjusted to their nearest neighbor so that there should be a competition in the process of finding the class of an MN.

According to the features of data, we assign these classes to records, and hence, make it a supervised learning problem. In supervised learning problems, the final output is already known. We prefer supervised learning scheme in order to classify MNs in their deserved class as well as to make it possible for all MNs to adjust themselves into the categorization.

Table 5 displays the final classification of an MN after preprocessing the generated data.

Table 5. Distinct Classes According to Preprocessed Data Attributes

S #	Session Created	Time Interval Between Two Consecutive Incoming Sessions	Handover	Predicted Class
1	Yes	Moderately Medium	Yes	Highly Active
2	Yes	Moderately Medium	No	Rarely Active
3	Yes	Medium	Yes	Active
4	Yes	Medium	No	Rarely Active
5	Yes	Low	Yes	Moderately Active
6	Yes	Low	No	Rarely Active
7	Yes	Moderately High	Yes	Somehow Active
8	Yes	Moderately High	No	Rarely Active
9	Yes	High	Yes	Rarely Active
10	Yes	High	No	Inactive

Table 5 summarizes the different set of records and classes into distinct categories. Every record is checked for the following properties; (i) has the incoming session been created? (ii) what is the discretized interval of current and previous incoming session? (iii) is there an IP handover? There are six classes for each record to fall-in. At the end, we can calculate the total number of records for each category and the category having maximum number of records will be selected as the class for an MN.

G. Proposed LUMs to be Sent

Table 6. Proposed LUMs to be sent according To the Classification

S#	Category	IP Handover	Predicted Class	LUMs to be sent in 24 Hours
1	Moderately Medium	Yes	Highly Active	24
2	Medium	Yes	Active	18
3	Low	Yes	Moderately Active	16
4	Moderately High	Yes	Somehow Active	12
5	High	Yes	Rarely Active	6
6	High	No	Inactive	2

Table 6 summarizes the proposed scheme for sending the LUMs when an incoming session is created on an MN.

H. Classification Algorithms

The main purpose of using ML algorithms for classification of MNs into distinct categories is to extract rules for classification. These rules are followed to classify given data. Every algorithm constructs different set of rules. The rules work the same way as the if-else structure in high-level languages.

A combined set of attributes are the outcome of pre-processing. The pre-processed data is then provided to well formulated, ML algorithms including:

- i. J48 decision trees [34,35]
- ii. J-Rip [36]
- iii. Ridor [37,38]
- iv. Naïve Bayes [39]

The ML algorithms listed above are first tested with the actual preprocessed data in the ML evaluation tool WEKA [40], and then, we incorporate error of 5% and 10% in the pre-processed data. For all six classification categories, the data logs are checked.

Each classification case is tested 1000 times with and without incorporating noise. Average of logs without noise is 92.70%. With 5% noise, it is 87.56% and with 10% noise, it is 84.35%. So, we can conclude that the classification algorithms are working efficiently on the pre-processed data with and without adding noise. In the next section, we discuss the testing of the preprocessed data with and without noise.

IV. RESULTS AND DISCUSSION

After classifying MNs into distinct categories, we run miscellaneous tests on the MNs' classification and history logs. The simulation program is written in JAVA. We consider the following performance metrics to evaluate our proposed mechanism.

A. Signaling Load

To measure the signaling load, we count the number of LUMs that are sent by an MN in one year.

B. Mean Tunnel Time

Tunnel time is the time for which a tunnel exists after its creation (between two ANs) for providing the continuity of the data sessions. It is evident that when we increase the period of sending an LUM for inactive or moderately active MNs, it increases the overall mean tunnel time. We use EWMA (Exponential Weighted Moving Average) [41] for adapting the activity of an MN, initially, for a complete month and then after every four hours. By having this important functionality, if initially, there is some excessive tunnel time due to the high time interval of LUMs, it is catered for as we run the proposed algorithm after every four hours to monitor and predict the activity of an MN.

Table 7. Number of Incoming Sessions and Session Arrival Rate

S #	Total # of Sessions	Incoming Session Arrival Rate (λ): Average # of sessions per second
1	26280	0.00083
2	13140	0.00041
3	8760	0.00027
4	2920	0.000091
5	2190	0.000069

Table 7 summarizes the total number of incoming sessions and the arrival rate of the sessions on an MN. The table shows that if the arrival rate value is large, there are many incoming sessions on an MN from CNs and vice versa.

Table 8. AN-Area Residence Times

S #	Total Handovers	AN-Area Handover Rate (α)	AN-Area Residence Time ($1/\alpha$)
1	373950	0.01185	1 min. 40 sec.
2	37370	0.001185	14 min.
3	3790	0.0001185	138 min.
4	1215	0.00003747	432 min.
5	775	0.00002370	675 min.
6	350	0.00001185	1460 min.

The second column of Table 8 shows the total number of handovers. The third column lists the AN-Area handover rates and the fourth column lists the AN-Area residence times.

C. Classification Accuracy Testing

Now, we present the activity of an MN in terms of AN-Area, average AN-Area residence time, total number of handovers, total number of incoming sessions on an MN, total number of tunnels, and mean tunnel time with respect to the specific classification categories.

Table 9. Parameter Settings: Upper Part Gives the Input Values. Lower Part Lists the Calculated Values for the Highly Active Category; S. Stands for Scenario

Parameter	S. # 1	S. # 2	S. # 3	S. # 4	S. # 5	S. # 6
Area(km ²) of AN	10,000	5000	1000	100	1	0.01
Average AN-Area residence time	1390 min.	712 min.	439 min.	139 min.	14 min.	1 min. 25 sec.
# of handovers	375	735	1175	3750	37500	375000
# of incoming sessions	13300	13300	13300	13300	13300	13300
Calculated values for the highly active category						
# of tunnels	5	14	30	130	6160	13050
Mean tunnel time (hrs)	2	7	17	243	3060	6570

Table 9 shows the overall input parameter settings and calculated output parameters for the highly active classification setting. When behavior of a particular MN

is classified into highly active category, the above mentioned input and output parameters are the results of the simulation. The results of the remaining five classification categories are omitted.

D. Overall Results of Reduction in Signaling Load

The main contribution of our work is to reduce the signaling load of location management.

We have explained the usage of machine learning in the research for mainly structuring data from all numerical values to discretized values and to pre-processed form, and from there to classify each MN in distinct categories. In a realistic environment, an MN does not stay idle or highly active all the time. There are off days from work, public holidays, and most importantly the relaxing time. So, we test each incoming session with AN change activity and the time interval between two consecutive sessions. If an MN is receiving incoming sessions at relaxing time and if that MN is not mobile (*which normally happens*), we classify that MN as rarely active. In this case, on average after every 4 hours an LUM will be sent.

Table 10. Comparison of the Proposed Scheme with the Scheme without Intelligence

S #	Number of LUMs Using Proposed Scheme	Number of LUMs Without Intelligence
1	4603	8760
2	4309	8760
3	4425	8760
4	4311	8760
5	4166	8760
6	4736	8760
7	4724	8760
8	3599	8760
9	4179	8760
10	4143	8760

Table 10 lists the total number of LUMs that are sent by an MN in a time period of one year. It can be seen that the values of the proposed scheme are around half of the values of the scheme without having intelligence in it.

Fig. 3 shows that the proposed scheme is easing signaling load of location management by more than 50%.

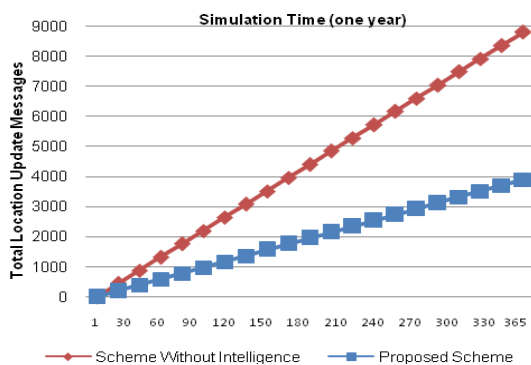


Fig.3. Total Number of LUMs: Comparison of the Proposed Scheme with the Scheme without Intelligence

E. Use of Exponentially Weighted Moving Average

The reason for using EWMA is to handle the abrupt changes in setting the period of sending the LUMs. For example, if an MN is highly active for a few days and then it becomes moderately or rarely active, the frequency of sending the LUMs should not be abruptly changed and instead the change should be smooth.

We take one month data of incoming sessions and mobility patters of an MN. We calculate the overall rating and assign the numerical weight to the selected classification category. Then, we run our algorithm after every four hours and determine the new classification category for the MN. If the newly predicted class is different from the current class, we assign 90% weight to the current class and 10% weight to the newly predicted class. So, in order to switch the class of an MN from highly active to moderately active, we need consecutive predictions of moderately active. EWMA serves our purpose by not allowing algorithm to change classification category abruptly.

Table 11 shows the numerical values we attach with each distinct classification category. So, with every new prediction, the value of a class is changed by 0.1.

Table 11. EWMA Numerical Values for the Classification Categories

S #	Classification	Range
1	Highly Active	5.0 - 6.0
2	Active	4.0 - 4.9
3	Moderately Active	3.0 - 3.9
4	Somehow Active	2.0 - 2.9
5	Rarely Active	1.0 - 1.9
6	Inactive	0.0 - 0.9

F. Overall Results for Total Number of Tunnels and Mean Tunnel Time

The increase in mean tunnel time is also an important aspect of our research. There is a trade-off between LUM interval and tunnel time. If LUM interval increases, the tunnel time also increases and vice versa.

Our proposed scheme periodically checks the activity of an MN and on the basis of that activity, we change the LUM interval. If the MN is classified into inactive or some less active category, we increase the time interval of sending the LUMs and it eventually increases the tunnel time.

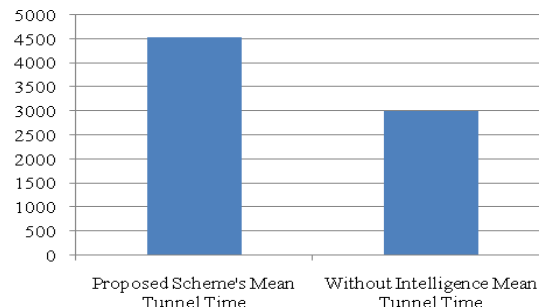


Fig.4. Mean Tunnel Time (in Seconds): Comparison of the Proposed Scheme with the Scheme without Intelligence

Fig. 4 shows that the mean tunnel time of the proposed scheme is higher than that of the scheme without having intelligence in it.

Fig. 5 shows the total number of tunnels created in the simulation time period of one year. It is obvious that the proposed scheme of reduction in LUMs results in more number of tunnels than an unintelligent scheme in which an MN sends an LUM after a fixed period of time. Due to the availability of enough memory and processing resources at an AN to process tunnels, the increased number of tunnels due to our intelligent mechanism is not a concern.



Fig.5. Total Number of tunnels: Comparison of the Proposed Scheme with the Scheme without Intelligence

V. CONCLUSION

In this paper, we have proposed an intelligent mechanism to check the data sessions and mobility patterns of a mobile node (MN), and then take decision about the MN in order to classify it in some distinct category and manage its frequency of sending the LUMs (Location Update Messages). As a result, there is reduction in the cost of sending the LUMs, which consequently reduces the traffic load on a network. We have rigorously compared our intelligent mechanism with an unintelligent mechanism that sends a LUM after a constant period. The results show more than 50% reduction in LUMs with the proposed intelligent approach.

REFERENCES

- [1] James F. Kurose, and Keith W. Ross, *Computer Networking: A Top Down Approach*, 6e. Pearson Education, 2012.
- [2] Mark Hall, Ian Witten, and Eibe Frank, "Data mining: Practical machine learning tools and techniques", *Kaufmann, Burlington*, 2011.
- [3] Y. Xiao, Y. Pan, and J. Li, "Design and analysis of location management for 3G cellular networks", *IEEE Transactions on Parallel and Distributed Systems*, 15(4), 2004, pp. 339-349.
- [4] B. Sidhu, and H. Singh, "Location management in cellular networks", *World Academy of Science, Engineering and Technology*, 25, 2007, pp. 314-319.
- [5] K. Kyamakya, and K. Jobmann, "Location management in cellular networks: classification of the most important paradigms, realistic simulation framework, and relative performance analysis," *IEEE Transactions on Vehicular Technology*, 54(2), 2005, pp. 687-708.
- [6] C. Selvan, R. Shanmugalakshmi, and V. Nirmala, "Location management technique to reduce complexity in cellular networks", *IJCSI International Journal of Computer Science Issues*, 2010.
- [7] Malika Belkadi, Rachida Aoudjit, Mehmed Daoui, and Mustapha Lalam, "Mobile Localization Based on Clustering", *International Journal of Computer Network and Information Security (IJCNIS)*, 5(9), 2013, p. 37.
- [8] Smita Parija, Santosh Kumar Nanda, Prasanna Kumar Sahu, and Sudhansu Sekhar Singh, "Location Prediction of Mobility Management Using Soft Computing Techniques in Cellular Network", *International Journal of Computer Network and Information Security (IJCNIS)*, 5(6), 2013, pp. 27-33.
- [9] J. Xie, and I.F. Akyildiz, "A novel distributed dynamic location management scheme for minimizing signaling costs in mobile IP", *IEEE Trans. Mobile Computing*, 1(3), 2002, pp. 163-175.
- [10] YuJia Zhai, XinYu Mao, Yue Wang, Jian Yuan, and Yong Ren, "A DHT-based fast handover management scheme for mobile identifier/locator separation networks", *Science China Information Sciences*, 56(12), 2013, pp. 1-15.
- [11] Mehdi Bezhaf, Luigi Iannone, Marcelo Dias De Amorim, and Serge Fdida, "Transparent and distributed localization of mobile users in wireless mesh networks", *In International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness, Springer Berlin Heidelberg*, 2009, pp. 513-529.
- [12] Stoica, R. Morris, D. Karger, M.F. Kaashoek, H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications", *In Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '01) ACM*, pp. 149-160.
- [13] Waqas Ahmed Imtiaz, Muhammad Afaq, and Muhammad Asmatullah Khan Babar, "mSCTP Based Decentralized Mobility Framework", *International Journal of Advanced Computer Science and Applications*, 2(9), 2011, pp. 106-112.
- [14] W.A. Imtiaz, "Two-Tier CHORD for Decentralized Location Management", *International Journal of Computer Applications*, 2013, 69(4).
- [15] "DMM Working Group", <http://tools.ietf.org/wg/dmm/>
- [16] Hassan Ali-Ahmad, Meryem Ouzzif, Philippe Bertin, and Xavier Lagrange, "Comparative performance analysis on dynamic mobility anchoring and proxy mobile IPv6", *In Wireless Personal Multimedia Communications (WPMC), 2012 15th International Symposium on*, pp. 653-657.
- [17] Kashif Munir, Xavier Lagrange, Philippe Bertin, Karine Guillouard, and Meryem Ouzzif, "Performance analysis of mobility management architectures in cellular networks", *Springer Telecommunication Systems*, 59(2), 2015, pp. 211-227.
- [18] Tien-Thinh Nguyen, and Christian Bonnet, "DMM-based inter-domain mobility support for Proxy Mobile IPv6", *In Wireless Communications and Networking Conference (WCNC), 2013 IEEE*, pp. 1998-2003.
- [19] Jong-Tae Park, Seung-Man Chun, Jun-Hyuk Choi, and Seung-Mu Lee, "Simple mobility management protocol for global seamless handover", *In Consumer Communications and Networking Conference (CCNC), 2012 IEEE*, pp. 677-681.
- [20] Christian Makaya, and Samuel Pierre, "An analytical

- framework for performance evaluation of IPv6-based mobility management protocols”, *IEEE transactions on wireless communications*, 7(3), 2008, pp. 972-983.
- [21] Karen Church, Barry Smyth, Paul Cotter, and Keith Bradley, “Mobile information access: A study of emerging search behavior on the mobile Internet”, *ACM Transactions on the Web (TWEB)*, 1(1), 2007.
- [22] Gou-feng Zhao, Qing Shan, Shasha Xiao, and Chuan Xu, “Modeling web browsing on mobile internet”, *Communications Letters, IEEE*, 15(10), 2011, pp. 1081-1083.
- [23] Anindya Ghose, and Sang Pil Han, “An empirical analysis of user content generation and usage behavior on the mobile Internet”, *Management Science*, 57(9), 2011, pp. 1671-1691.
- [24] Ram Kumar Singh, Amit Asthana, Akanksha Balyan, Shyamji Gupta, and Pradeep Kumar, “Vertical handoffs in fourth generation wireless networks”, *International Journal of Soft Computing and Engineering (IJSCE)* 2, no. 2, 2012, pp. 481-490.
- [25] Hassan Ali-Ahmad, Kashif Munir, Philippe Bertin, Karine Guillouard, Meryem Ouzzif, and Xavier Lagrange, “Processing loads analysis of distributed mobility management and SIP-based reachability”, *Springer Telecommunication Systems* (2016): 1-16. DOI:10.1007/s11235-016-0148-4.
- [26] Leonhard Korowajczuk, LTE, WiMAX and WLAN network design, optimization and performance analysis, *John Wiley & Sons*, 2011.
- [27] Konstantinos Drakakis, and Dragan Radulovic, “A discretized version of the self-similar model for internet traffic”, *Applied Mathematical Sciences*, 2(55), 2008, pp. 2743-2756.
- [28] Adriano Pereira, Leonardo Silva, W. Meira, and Walter Santos. “Assessing reactive qos strategies for internet services”, *In Applications and the Internet*, 2006.
- [29] Yang Chu, Sanjay Rao, Srinivasan Seshan, and Hui Zhang. “Enabling conferencing applications on the internet using an overlay multicast architecture”, *ACM SIGCOMM Computer Communication Review*, 31(4), 2001, pp. 55-67.
- [30] César Cárdenas, Maurice Gagnaire, Víctor Lopez, and Javier Aracil, “Performance evaluation of the Flow-Aware Networking (FAN) architecture under Grid environment”, *In Network Operations and Management Symposium, NOMS 2008. IEEE*, pp. 481-487.
- [31] Xing Dongshan, Song Qinbao, and Shen Junyi, “Research on a New Algorithm of Web Session Fuzzy Clustering”, *JOURNAL-XIAN JIAOTONG UNIVERSITY*, 36(8), 2002, pp. 822-825.
- [32] A.A. Ojugo, E. Ben-Iwhiwhu, O. Kekeje, M.O. Yerokun, I.J.B. Iyawa, “Malware Propagation on Social Time Varying Networks: A Comparative Study of Machine Learning Frameworks”, *IJMECS*, 6(8), 2014, pp. 25-33.
- [33] Yuguang Fang, and Yi Zhang. “Call admission control schemes and performance analysis in wireless mobile networks”, *IEEE Transactions on Vehicular Technology*, 51(2), 2002, pp. 371-382.
- [34] Ljupčo Todorovski, and Sašo Džeroski. “Combining classifiers with meta decision trees”, *Machine learning*, 50(3), 2003, pp. 223-249.
- [35] Neeraj Bhargava, Girja Sharma, Ritu Bhargava, and Manish Mathuria, “Decision tree analysis on j48 algorithm for data mining”, *Proceedings of International Journal of Advanced Research in Computer Science and Software Engineering*, 3(6), 2013.
- [36] Yong Wang, Margaret Martonosi, and Li-Shiuan Peh, “A new scheme on link quality prediction and its applications to metric-based routing”, *In Proceedings of the 3rd international conference on Embedded networked sensor systems*, ACM, 2005, pp. 288-289.
- [37] Hassan Najadat, and Izzat Alsmadi, “Enhance Rule Based Detection for Software Fault Prone Modules”, *International Journal of Software Engineering and Its Applications*, 6(1), 2012, pp. 75-86.
- [38] Bing Liu Wynne Hsu Yiming Ma, “Integrating classification and association rule mining”, *In Proceedings of the fourth international conference on knowledge discovery and data mining*, 1998.
- [39] Jingnian Chen, Houkuan Huang, Shengfeng Tian, and Youli Qu, “Feature selection for text classification with Naïve Bayes”, *Expert Systems with Applications*, 36(3), 2009, pp. 5432-5435.
- [40] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten, “The WEKA data mining software: an update”, *ACM SIGKDD Explorations Newsletter*, 11(1), 2009, pp. 10-18.
- [41] James M. Lucas, and Michael S. Saccucci, “Exponentially weighted moving average control schemes: properties and enhancements”, *Technometrics*, 32(1), 1990, pp. 1-12.

Authors' Profiles



Kashif Munir did his PhD from University of Innsbruck, Austria in 2009. He was a post doctoral researcher at Telecom Bretagne, France from February 2011 to December 2012. He is currently working as an Associate Professor at the department of Computer Science, FCIT, King Abdulaziz University, Jeddah, KSA. Previously, he worked at National University of Computer and Emerging Sciences, Islamabad, Pakistan. He is an author of numerous peer-reviewed conference and journal publications.



Ehtesham Zahoor is currently working as an Assistant Professor at National University of Computer and Emerging Sciences, Islamabad, Pakistan. He is head of the Secure Networks and Distributed Systems Lab (SENDS). He did his PhD from Université de Lorraine, France in 2011. He has been an active researcher in the domain of distributed and cloud computing and author of numerous peer reviewed conference/journal publications.



Waseem Shahzad is currently working as an Associate Professor and Head of the department of Computer Science at National University of Computer and Emerging Sciences, Islamabad, Pakistan. His areas of interest are machine learning, data mining and knowledge discovery, data warehousing, big data, and evolutionary computing. He has several research publications in these areas.



Syed Junaid Hussain did his MS (Computer Science) in 2014 from National University of Computer and Emerging Sciences, Islamabad, Pakistan. His research interests include mobile networks, machine learning, and artificial intelligence. He has work experience in software

architecture, open-source web application development, and database design.

How to cite this paper: Kashif Munir, Ehtesham Zahoor, Waseem Shahzad, Syed Junaid Hussain, "Intelligent Reduction in Signaling Load of Location Management in Mobile Data Networks", International Journal of Computer Network and Information Security(IJCNIS), Vol.8, No.11, pp.23-31, 2016.DOI: 10.5815/ijcnis.2016.11.03