A PREDICTION-BASED CLUSTERING ALGORITHM TO ACHIEVE QUALITY OF SERVICE IN MULTIHOP AD HOC NETWORKS

S. Sivavakeesar and G. Pavlou
CCSR, University of Surrey, UK

Abstract - This paper presents a novel framework for dynamically organising mobile nodes (MNs), in order to support Quality of Service (QoS) in multihop mobile ad hoc networks (MANETs). We name the proposed model the (p, t, d)-clustering model. This dynamic & fully distributed clustering approach is based on intelligent mobility prediction and enables MNs of a dynamic MANET to have a consistent view of the network at any time. QoS requirements of multimedia traffic necessitate this kind of consistency, which is achieved with an introduction of virtual cluster that is geographically oriented to a particular region. Making intelligent decisions regarding mobility and usage of location information [09] are highly desirable and, in fact, inevitable in MANETs, as the channel bandwidth and transmission power are scarce. Our proposed approach will also enable to capture context in future pervasive computing environments. Simulation results demonstrate the performance improvement of our approach with respect to similar approaches.

1. INTRODUCTION

This paper presents the results of our initial attempt to support QoS for multimedia traffic in ad hoc networks. The latter offer unique benefits and versatility for a variety of situations and applications. However, wireless communication and the lack of centralised administration in such networks pose numerous challenges [01]. In MANETs, the quality of wireless links is highly variable, and this situation is further aggravated by the fact that these links are formed between MNs that are likely to be moving independently of each another. Unlike in a fixed wired network, the rate of link failure due to node mobility is the primary obstacle to effective QoS support in MANETs. Hence, in such a situation, the effectiveness of any adaptive QoS mechanism depends heavily on the timeliness and detailed topology information available to them. Since it is expected that MANETs will be composed of a large number of MNs, it is better to adopt hierarchical rather than a flat structure. This fact makes researchers focusing their attention in partitioning the multihop network into clusters, and electing cluster heads (CH). This kind of clustering technique will bring in a number of benefits as stated in [03]. The primary step in clustering is the election of CHs and the formation of clusters around them. In this work, a large-scale deployment of multihop MANETs similar to mobile telephony system is envisioned (and this work does not necessarily consider a creation of spontaneous[1] networks). Because of the above-mentioned reasons, we are also adopting a hierarchical clustering technique in our work. The clustering approach proposed here is fully distributed and dynamic in nature. Our approach differs from previous related work in the way a CH is elected based on mobility prediction and the fact we introduce virtual clusters. Location information may be obtained using the Global Positioning System (GPS), or a self-positioning algorithm as specified in [02]. Although storing the location information in a scalable manner and making the prediction in an effective way demand storage & calculation capacity, these are affordable if we can limit the need for updates as much as possible while maintaining up-to-date topology information. That way we can limit the waste of wireless bandwidth & transmission power that are scarce. Minimisation of frequent update means less interference to other MNs. Our work is further motivated by the fact that both memory and processing power continue to get cheaper every year.

The rest of this paper is organized as follows. Section II reviews related previous work. The proposed new clustering technique based on mobility prediction is clearly described in section III. Section IV evaluates the proposed scheme by using simulations, and it demonstrates that our clustering scheme leads to more stable cluster formation than the other methods proposed in the literature, and hence yields better QoS performance. Section V presents our conclusions.

2. PREVIOUS WORK & OUR MOTIVATION

A number of techniques to select CH have been devised in the recent past, the most popular ones being the lowest identifier (Lowest-ID) [07] and maximum-connectivity [03]. But these two do not result in stable clusters: in the former, a highly mobile low ID CH will cause severe re-clustering while if this CH moves into another region may pose danger to an existing CH. In the latter, as well as in Lowest Distance Value (LDV)[2] and the Highest In-Cluster Traffic (ICT)[2] [04], depending on MNs movement and traffic characteristics, the criterion values used in the election process can keep on varying for each MN, and hence result in instability. A clustering scheme referred to as ($\propto$, t)-cluster has been proposed in [05], and it focuses on mathematical characterisation of the probability of link and path availability as a function of a random walk based mobility model. In the latter, it is considered that a link is an active link between two MNs at time $t_1 + t_0$ ($t_1 > t_0$) given that there is an active link between them at time t_0. This scheme leads to an ambiguity as to how big is t_1 and also it does not consider things that might have happened in the interval

[1] spontaneous networks come into life and exist only for a short period, and hence, in such kind of networks problems due to mobility of MNs are not significant.
[2] Clustering algorithms developed for HiperLAN type 2.

$t_1 + t_0$. If this interval is too big, the clustering process will take longer time to converge, and on the other hand if it is too small, then the resulting clusters will not be effective. A clustering scheme based on a mobility-metric is proposed in [06]. It defines a relative mobility metric that is actually a ratio of received power between two successive packet transmissions from a neighbour MN. Since this mobility metric uses power measurements, its accuracy depends heavily on how well a varying channel condition is modelled, and hence not practical.

Our technique in mostly motivated by the fact that in MANETs link bandwidth and MNs transmission power are scarce [01] , and any effective solution should take this into account and try to conserve these two. However, in order to achieve QoS for real-time multimedia applications, all the MNs should be able to predict the resources available in the network. This in turn necessitates that each MN in a MANET should have up-to-date information about the network topology. In order to achieve a compromise between these two extremes, in such a network accurate prediction of future state is necessary for the network control algorithms to keep pace with rapid and frequent state changes. Hence, this paper proposes a mobility prediction scheme based on past history in a scalable manner. In MANETs, the uncertainty as to network topology arises mainly due to MN movement. If, however, the movement pattern of the MNs is deterministic, then the lifetime of a link can be determined with accurate precision, and the entire task becomes much easier. In typical mobile networks, MNs exhibit some degree of regularity in their mobility pattern and this is what we are using in our proposed solution. The main objectives of our algorithm are to ensure the realisation of the followings:

- The algorithm should achieve stable cluster topology.
- It should do so with minimal communications overhead.
- In highly dynamic environments like MANET, the algorithm should be highly distributed and should operate asynchronously.

Also, various clustering algorithms that appear in the literature such as Lowest-ID and max-connectivity, do not necessarily preserve a particular clusters identity after a CH change. This means that the number of MNs attached will vary after a changeover, because it may often result in cluster splitting and/or merger. As a result, each CH changeover necessitates increased update or signalling overhead, the occurrence of which could have otherwise been minimized by selecting a CH that will preserve the previous cell identity (meaning that it will keep the cluster-size nearly constant & serve the same area as the previous CH), and stay in that particular cluster for a long time.

3. (p, t, d)-CLUSTERING MODEL

Having taken into account the common deficiencies of all other approaches, our algorithm selects a node as CH that has the least probability (when compared to others within the same virtual cluster) to move out of the current virtual cluster, and it has the minimum distance from the respective virtual cluster centre (VCC). The first requirement is to ensure that a highly mobile MN related to its neighbours is *not* elected as a CH. The second is to make sure that in subsequent CH changes, the area covered would not be impaired. Our proposed model is known as (p, t, d)-clustering model. More accurately it is (p_{ik}, t_{ik}, d_{ik})-clustering model, where 'p_{ik}' is the probability that i^{th} MN within k^{th} virtual cluster having a distance, from the center of the virtual cluster, of 'd_{ik}' stays within that cluster for some specified time period 't_{ik}'. Any i^{th} MN within k^{th} virtual cluster, having $p_{ik} = p_{max}$ for $t_{ik} \geq t_c$ (where t_c is system dependent) and $d_{ik} = d_{min}$, can become a CH of that virtual cluster. Here 'p_{ik}' is determined based on the mobility prediction model described below. The following subsection explains the key aspects of this model.

3.1. MOBILITY PREDICITION MODEL

Our mobility model is a semi-deterministic movement model that integrates deterministic behaviour with randomness in an attempt to mimic actual human movement behaviour. This model attempts to derive the probabilistic prediction of user mobility by utilising the accumulated behaviour history of the specific mobile node. Since our model is mainly interested in mobile users' inter-virtual cluster movement behaviour, the randomness associated with intra-virtual cluster movement behaviour can be minimised to a great extent. This is similar to the global mobility model (GMM) as opposed to the local mobility model (LMM) specified in [10]. As in GMM, our consideration is motivated by the fact that most MNs exhibit some regularity in their daily movement, and this regularity can best be characterised by a number of user mobility patterns (UMP) [10]. In addition to specifying the movement pattern in terms of a sequence of virtual clusters, the residence time of a particular MN in each cluster is also specified.

The mobility database of each MN at a specific day (of a week) is represented as a mobility trie [12], which is to be constructed using the character-based version of the Ziv-Lempel [12] algorithm. As in data compression, this mobility trie is built by each node in an online fashion in such away that it preserves the relevant statistics that can be used to predict the probability of future events. Together with transition probabilities, each leaf of the trie consists of the 2-tuple information {virtual cluster identifier, residence time in that virtual cluster}. In this work, we use a finite k-state (S_1, S_2, …, S_k) continuous time Markov process, where state S_i denotes the i^{th} virtual cluster that an MN visits during a day of a week. Accordingly, each mobile has its own such Markov-chain and assuming a steady state behaviour of this chain, the state transition probabilities of $p_{x,j,k}(t)$ can be determined, based on past historical data of a specific MN using the Ziv-Lempel algorithm. $p_{x,j,k}(t)$ is the probability for a specific MN $x \in X_j$, where X_j is the set of all MNs in the virtual cluster 'j', to travel from j^{th} virtual cluster to k^{th} virtual cluster at time 't'. Another important point to be noted is that a particular MN would not visit or exert influence on all the virtual clusters in a particular region (visits, say, only k number of virtual clusters, and hence k-state Markov chain). This is similar to the 'shadow clusters' proposed in [11]. The root of the above trie represents the virtual cluster from where a specific mobile is generated.

3.2. VIRTUAL CLUSTER

In order to make our clustering scheme scalable, this paper introduces the notion of virtual clusters. The region to be covered by a MANET has static virtual clusters associated to geographical areas. Each MN to be used in this region is supposed to know or have in its memory a complete picture of the locations of virtual cluster centres (VCC). The unique identifier associated with each VCC can be calculated, given location information, using a publicly known function. This is similar to the LDA scheme and VHR of [02], although our approach is different in every aspect. In this context, each VCC is assumed to be 2-hops[1] away from each other. A virtual cluster may contain an actual cluster or not, and if the node degree happens to increase beyond a certain limit, an internal cluster-splitting[2] would occur based on hierarchical principles. Again here, it is ensured that waste of scarce resources is kept within limits. Each MN is supposed to construct its mobility trie with respect to these VCCs.

3.3. CLUSTERING ALGORITHM

In this scheme, the clustering computations are performed in each MN in a distributed manner when triggered. The decision as to the next CH to be selected OR elected depends on whether the current CH is available or not. If there is a primary CH or deputy CH available, then they will make the decision on behalf of all MNs, after getting the values of Λ (see equation 2 below) from each MN in the present virtual cluster. On the other hand if any of them does not exist, because of abrupt failure or error in prediction, then MNs within the virtual cluster elect one as their future CH in a distributed manner. The set-up time for the cluster formation in the latter case is higher when compared to that of the former. For this algorithm to work well, each MN is supposed to maintain its past history in terms of a mobility trie. If however, a MN is unable to construct its mobility trie, it can still use its distance in the criterion calculation, as $t_{ik} = 0$ in this case. The MN that has the highest Λ can become the CH. In forming clusters, the CH has to make sure that the area covered would not be impaired. Therefore, the CH makes a virtual cluster of k-hop, whose value is not necessarily uniform and the same between any border MN and the CH. Some of the terms used in this paper, such as 'Hello' message, adjacent cluster, and gateway MNs, are as those specified in [06], [07], [08].

Each CH is supposed to broadcast a HELLO message periodically – say every CH_HELLO_INTERVAL. This message carries the ID of the virtual cluster that it is covering, VCC, and virtual cluster's radius from VCC, in addition to other information. Whenever, a new MN receives this message from a CH, it can send a JOIN message immediately, if it is within the region (i.e. virtual cluster) covered by the specific CH. Similarly, whenever a CH receives a JOIN message from any MN, it has to check whether it is within its coverage area. If it is, the CH is supposed to include this MN in its cluster, and append its information in its neighbour table [07]. There is an exception to this case, when the MN's residence time within a specific virtual cluster is minimal, and hence it will not be enlisted. On the other hand, in case the MN is not within the virtual cluster concerned, the MN will not be included. In either case, the MN has to wait at least for the next two successive CH_HELLO_INTERVAL to check whether it has been appended in the neighbour table, which is transmitted as part of periodic HELLO messages by the present CH. If it is not included, the new MN has to re-transmit the JOIN message. Likewise, a MN can be a member of up to a four maximum adjacent virtual clusters. This specific MN would then behave as gateway [07] or forwarder [04] between those virtual clusters. Every MN within a particular virtual cluster is supposed to unicast a HELLO message to its respective CH periodically – say every MN_HELLO_INTERVAL, where every MN_HELLO_INTERVAL >> CH_HELLO_INTERVAL. In the HELLO message, an MN specifies in "MN TYPE" field, whether it is acting as a gateway or an ordinary node. These signalling messages, except periodic HELLO messages by CHs, are relayed by intermediate MNs only within the virtual cluster in which those messages are destined for. Each intermediate MN will determine this by checking the virtual cluster ID or whether the sender is within a specific virtual cluster. On the other hand, periodic HELLO messages by CHs have to be unicast by gateways between CHs of adjacent virtual clusters. This is to enable CHs to get the topology information of adjacent virtual clusters. Since each CH knows about the predicted residence time of each MN within its virtual cluster, it has to delete an entry, associated with a particular MN, from the neighbour table when its residence time expires. This effect will be reflected in every HELLO message a CH broadcasts periodically. The unique aspect of our protocol is that, before a particular CH becomes unavailable, it has to trigger the "CH changeover event". It will broadcast "CH

[1] The 2-hop is taken as twice the maximum transmission range of an average MN.

[2] Up to a maximum of three internal cluster-splitting can occur.

Changeover Event" message within its virtual cluster, and all member MNs are supposed to start performing the clustering criterion calculation process immediately. When this event occurs,

- Each MN has to calculate its distance from the centre of a particular virtual cluster (such information is broadcast periodically by each CH). Assuming an MN with an identifier 'i', whose location co-ordinates at time 't' are ($x_{ik}(t)$, $y_{ik}(t)$), in the k^{th} virtual cluster, whose centre's Cartesian co-ordinates are (x_{ck}, y_{ck}), its distance – $d_{ik}(t)$ – at time 't' can be calculated by :

$$d_{ik}(t) = \sqrt{(x_{ik}(t) - x_{ck})^2 + (y_{ik}(t) - y_{ck})^2} \quad \text{---------------------(01)}$$

- Each MN has to predict as to how long it can remain in the present virtual cluster. Let the time duration that MN with an identifier 'i' is going to spend in k^{th} cluster be t_{ik}, and p_{ik} be the probability for the i^{th} MN to stay in k^{th} virtual cluster during its residence time.

Based on the above, each MN 'i' is required to calculate, the clustering criterion factor Λ_i which is given by :

$$\Lambda_i = \begin{cases} p_{ik}\left(\dfrac{t_{ik} - (t_{th})}{d_{ik}(t)}\right) & \forall\, d_{ik}(t) \neq 0 \\ p_{ik}\ \Lambda_{max} & \text{Otherwise} \end{cases} \quad \text{------------(02)}$$

Where (t_{th}) is the threshold values (system dependent) for the residence time, Λ_{max} is the maximum possible value defined. At the end of calculation, each will unicast Λ **-**message to the CH. Based on the information received from its members, the CH will select the MN, that has the highest value for 'Λ **',** as the new primary CH. It will also select two assistant (deputy) CHs for reliability purposes – again based on Λ-values**.** The present CH will then broadcast this information using a SUCCESSOR Message. As soon as the new CH receives this, it will assume its status as the new primary CH, and so will the two assistant CHs.

If the first assistant CH sees that it has not received any HELLO message from the primary CH in the last two consecutive CH_HELLO_INTERVAL, then it will take over as the primary CH, informing its deputy as its first assistant CH. In this case, the CH changeover event will be triggered sometime before the present CH becomes unavailable, as usual. If, however, the second assistant has not received any HELLO message either from the primary or first assistant CH in the last four consecutive CH_HELLO_INTERVAL, it will assume duty as the primary CH. In this case, the CH changeover event will be triggered immediately. In case an ordinary MN notices no HELLO message from any of the CH – whether it to be a primary or assistant CHs –, within the period of six consecutive CH_HELLO_INTERVAL, the first noticed MN will assume the duty as the temporary CH, and it will immediately trigger the CH changeover event by broadcasting CH Changeover Event message. However, in this case MNs will broadcast their Λ **-**message within their virtual cluster. Accordingly, each MN becomes aware of other MNs' Λ**-**values. Each MN then compares its own value with that of each MN of the same virtual cluster, and one that has the highest value for 'Λ **'** will unilaterally be elected as the new primary CH. This new CH then selects its assistant CHs based on Λ **-**values, and soon informs this information by broadcasting the SUCCESSOR message. However, in this case, both the 'CH ID' and the 'New CH ID' fields of this SUCCESSOR message will be the same and constitute the new elected CH's ID. The new CH will then immediately start broadcasting HELLO message as usual. If however, an ordinary MN has not received any of the above signalling messages for more than eight last consecutive CH_HELLO_INTERVAL, then it will unilaterally elect itself as the CH. In this algorithm, if more than two MNs have the same value for 'Λ **',** the one with the lowest ID will be selected as the new CH. Unlike in any other clustering algorithm, this has another unique feature, whereby whenever a CH leaves the virtual cluster it has served, it will loose its status. In this way this algorithm ensures that no other MN can challenge or pose danger to an existing CH within a particular virtual cluster. Also, since in our algorithm each MN is supposed to be aware of its neighbours residence time within a specific virtual cluster, the frequency of HELLO message transmissions can be made lower than those of other known clustering schemes. In this way, our algorithm minimises the waste of channel bandwidth and transmission power.

4. EVALUATION THROUGH SIMULATION

The simulation work attempts to compare the performance of our (p, t, d)-clustering algorithm with the Lowest-ID clustering algorithm, in terms of the stability of clusters being formed. The y-axis shows the frequency of CH changes, and hence measures the (in)stability associated with each clustering algorithm. (The less frequency of CH changes, the more stable the cluster is). As it can be seen from Fig. 1, the (p, t, d)-clustering algorithm results in lower number of CH changeovers, and hence leads to more stable cluster formation, when compared to that of

Lowest-ID. The x-axis represents the number of MNs considered in this simulation. We performed our simulations using the GloMoSim simulation package in which we implemented and compared the Lowest-ID and our algorithm.

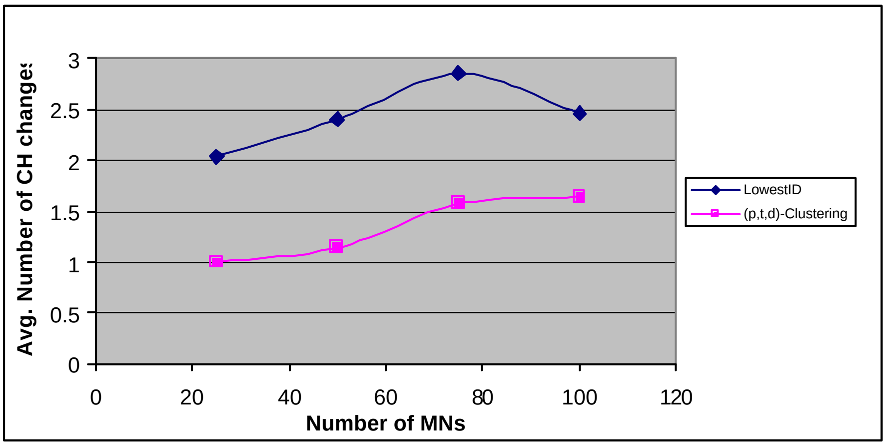

Fig. 1. Clustering Instability : Lowest-ID vs (p, t, d)-Clustering

5. CONCLUSIONS & FUTURE WORK

In this paper we presented a new clustering approach that makes use of intelligent mobility prediction and location information. We have demonstrated that this clustering scheme results in more stable clusters than those of other well-known schemes such as the Lowest-ID one using simulation. This in turn results in performance improvements, and this way of making intelligent mobility prediction is highly beneficial in pervasive computing environments that rely on accurate capturing of context. Our future work on QoS routing and resource reservation mechanisms will be built on this clustering scheme. The main objective is to minimise the handoff dropping probability by making use of mobility prediction and careful resource reservation by respective cluster heads or by location managers [08]. The clear demarcation of cluster boundary facilitates graceful handover of MNs as they move from one virtual cluster to another. Based on this we can build a DiffServ-like QoS mechanism tailored to MANETs, and this would be beneficial – together with our intelligent mobility management scheme– to future pervasive computing environments.

6. REFERENCES

[01] S.Chakrabarthi, and A.Mishra, QoS Issues in Ad Hoc Wireless Networks, IEEE Communications Magazine, Feb. 2001. pp 142 –148.

[02] L.Blazevic, L.Buttyan, S.Capkun, S.Giordona, J-P.Hubaux and J-Y.Boudec, Self-Organisation in MobileAd Hoc Networks: TheApproach to Terminodes, IEEE Communications Magazine, Jun. 2001. pp 166 –173.

[03] T-C.Hou, and T-J.Tsai, An Access-Based Clustering Protocol for Multihop Wireless Ad Hoc Networks, IEEE Journal on Selected Areas in Communications, vol. 19, July. 2001. pp 1201 –1210.

[04] J.Habetha, A.Hettich, J.Peetz, and Y.Du, Central Controller Handover Procedure for ETSI_BRAN HiperLAN2 Ad Hoc Networks and Clustering with Quality of Service Guarantees, Mobile and Ad Hoc Networking and Computing (MobiHOC) 2000.

[05] B.McDonald, and F.Znati, A Mobility-Based Framework for Adaptive Clustering in Wireless Ad Hoc Networks, IEEE Journal on Selected Areas in Communications, vol. 17, Aug. 1999. pp 1466 – 1487.

[06] P.Basu, N.Khan, and D.C.Little, Mobility Based Metric for Clustering in Mobile Ad Hoc Networks, Workshop on Distributed Computing Systems, 2001, pp. 413 – 418.

[07] M.Jiang, J.Li, and Y.C.Tay, Cluster Based Routing Protocol (CBRP) Function Specifications, IETF Draft, Aug. 1999.

[08] R.Ramanathan, and M.Steenstrup, Hierarchically-organised, multihop mobile wireless networks for quality-of-service support, Mobile Networks and Applications, vol. 3, June 1998.

[09] Y-B. Ko, and N. H. Vaidya, Location-Aided Routing (LAR) Mobile Ad Hoc Networks, MOBICOM'98, Oct.1998, Dallas.

[10] T.Liu, P.Bahl, and I.Chlamtac, Mobility Modeling, Location Tracking, and Trajectory Prediction in Wireless ATM Networks, IEEE Journal on Selected Areas in Communications, vol. 16, Aug. 2001, pp. 922 – 936.

[11] D.A.Levine, I.F.Akyilidiz, and M.Naghshineh, A Resource Estimation and Call Admission Algorithm for Wireless Multimedia Networks Using the Shadow Cluster Concept, IEEE & ACM Transactions on Networking, vol. 5, Feb. 1997.

[12] F.Yu, and V.C.M.Leung, Mobility-Based Predictive Call Admission Control and Bandwidth Reservation in Wireless Cellular Networks, IEEE INFOCOM 2001, pp. 518 – 526.