Dynamic Clustering with Quality of Service Guarantees and Forwarder Selection in Wireless Ad Hoc Networks

Jörg Habetha(1), Martin Nadler(2), Diego Calvo de Nó(2)
Email: habe@comnets.rwth-aachen.de
(1)Philips Research Laboratories, Aachen, Germany
(2)Aachen University of Technology, Chair of Communication Networks (ComNets), Aachen, Germany

Abstract – In [1] the concept of a centralised multihop ad hoc network has been presented. Dynamic Clustering of stations and Forwarder Selection are two of the most important procedures needed to build such a network. Due to the high dynamic of the system both procedures have to be used very often. Therefore, new and more efficient algorithms are required. In this paper a new clustering approach called *Highest-ID-with-traffic* is presented. The new algorithm is compared with other clustering algorithms previously developed by the authors in a realistic office scenario. Furthermore a new algorithm for the selection of forwarders is presented. This algorithm allows every cluster to determine its optimal forwarding constellation.

I. Introduction

Unlike traditional basestation-oriented wireless networks, ad hoc networks function without the use of any pre-installed, fixed infrastructure. Every station of an ad hoc network may serve as an access point to the fixed networks.

Classic application scenarios for ad hoc networks are battlefield communications, disaster recovery as well as search and rescue. More recent applications for the systems are Personal Communication Networks (PCN) especially in the home or office. Possible scenarios for the latter are business meetings, wireless computer LANs, conferences, expositions as well as the wireless interconnection of consumer and computer devices at home.

Considering the protocol stack, elements of centralisation have first been introduced on network layer by the concept of hierarchical routing. Hierarchical routing algorithms divide a network into logical zones or clusters with the objective to reduce the complexity of the routing process and the storage requirements. Inside a cluster only one station is responsible for the routing to and from stations situated in different clusters. All other nodes of the cluster only know the routes within their own cluster. Such routing schemes have an element of centralisation in the sense that some stations have to perform special tasks and other have not. Clusters can themselves be grouped in higher level clusters thereby creating a cluster hierarchy [2].

Elements of centralisation can not only be introduced on the network layer but also on every other protocol layer. A centralisation of the Medium Access Control (MAC) has been proposed in [3] for the first time. There, stations are grouped into clusters, in which one station, called the clusterhead, is responsible for organising the access to the air interface for all Mobile Terminals (MTs) that are members of its cluster.

One important advantage of a centralised MAC scheme lies in the fact that base-station oriented protocols can be reused. This reuse is beneficial from a cost of development and economies of scale perspective. However, sometimes it is the only possible way to get access to a certain part of the frequency spectrum. Considering this, we have presented in [1] a centralised ad hoc network based on the HIPERLAN/2 standard. HIPERLAN/2 is a wireless LAN standardised by the European Telecommunications Standardisation Institute (ETSI) for the 5 GHz band.

The basic mode of operation of HIPERLAN/2 foresees basestations supported by a fixed infrastructure. Nevertheless an ad hoc mode of operation has been defined in an extension to the basic standard. This ad hoc network uses the same MAC protocol as the infrastructure-based alternative. The equivalent of the base station in the infrastructure-based network is called Central Controller (CC) in the HIPERLAN/2 ad hoc network. The CC is responsible for generating the MAC frames and granting access to the air interface for all MTs inside its cluster.

The current HIPERLAN/2 ad hoc network extends over only one cluster. To cover larger areas multiple clusters are needed, especially in indoor scenarios where walls strongly attenuate the signals. In [1] a concept has been presented how to extend the HIPERLAN/2 ad hoc mode over several clusters. In such a multi-cluster network, like in decentralised ad hoc networks, communication between two stations (situated in different clusters) involves several other stations that have to forward the data. Connections extend over multiple hops whereas infrastructure-based communication uses only one hop (uplink or downlink).

It has been illustrated in [1] that in such a centralised multihop network three main problems have to be solved:

- dynamic clustering of the mobile stations,
- forwarding of data between the clusters
- and routing of packets.

In this paper we will focus on the first two tasks.

II. Dynamic Clustering

We propose 1-level clustering where stations are grouped into clusters but clusters are not further grouped in higher level clusters. There is one CC per cluster that is responsible for registration and radio access of MTs that are roaming in its cluster. A MT can become a member of the cluster if it is in hear range of the CC. In a free space scenario clusters would therefore have a circular form.

As MTs move, cluster-membership changes dynamically. Even CCs may be mobile. The variable data rate requirements increase the complexity of the system. In addition to the variable bit rate, connections will start and end frequently. Thus, it is obvious that clustering becomes a dynamic process and frequent task. As clusters have a limited traffic capacity (in HIPERLAN/2 a maximum of 54 Mbit/s on physical layer), the capacity requirements can only be met by opening new clusters.

One objective of the clustering process could be to spread the traffic as far as possible across all clusters by appropriate dynamic re-clustering of the network. Another objective of the clustering process can be a minimum of broken connections and MT handovers between the clusters. The most important feature of a clustering algorithm is its stability. Re-clustering can only be achieved by a Central Controller Handover as illustrated in [4]. However, this requires a large amount of the network resources.

A. Previously proposed Clustering Schemes

Two of the first proposed clustering schemes have been the so-called *Lowest ID* (LID) and the *Highest Connectivity* (HIC) algorithms [3, 5].

In the *Lowest ID* algorithm all stations have a network-wide unique identifier (ID). Periodically every stations broadcasts its ID to all stations that are in hear range. It is then possible for a station to compare its ID with the ID of its direct neighbours. A station autonomously decides to become a CC if its own ID is lower than all other received station IDs.

The *Highest Connectivity* algorithm is based on the number of stations, that a station is able to hear. Each station can calculate this number and broadcast it to its neighbours. Similarly as in the *Lowest ID* algorithm, every station then compares its connectivity value with the connectivity values of its neighbours. The station with the highest connectivity becomes the CC.

The shortcoming of these algorithms is that they do not consider the traffic situation inside the clusters. For this reason we have proposed in [1] to take into account not only the reception level of a signal from another station, called *Received Signal Strength* (RSS), but also the *Data Rate* (DR) that two stations have with one another. Once the CC has acquired these two values for every pair of MTs, it can build the matrix shown in Fig. 1.

	MT-ID1	*MT-ID2*	*MT-ID3*	…	*MT-IDn*
MT-ID1		RSS, DR 1 ← 2	RSS, DR 1 ← 3	…	RSS, DR 1 ← n
MT-ID2	RSS, DR 2 ← 1		RSS, DR 2 ← 3	…	RSS, DR 2 ← n
MT-ID3	RSS, DR 3 ← 1	RSS, DR 3 ← 2		…	RSS, DR 3 ← n
…	…	…	…	…	…
MT-IDn	RSS, DR n ← 1	RSS, DR n ← 2	RSS, DR n ← 3	…	

Fig. 1: RSS2 and traffic matrix of the network

In order to increase the efficiency of the network, a new and more appropriate CC has to be chosen in the own and the surrounding clusters. To achieve this, the RSS/DR matrix has to be exchanged between CCs. Thereby each CC can obtain a regional or even global topology map of the network.

Based on this approach two new clustering algorithms have been presented in [1]: the *Lowest Distance Value* (LDV) algorithm and the *Highest In-Cluster Traffic* (ICT) algorithm.

The *Lowest Distance Value* (LDV)[1] bases the decision on the RSS with which a station receives all its neighbours. Each station calculates the sum of all RSS values to its direct neighbours divided by the number of the 1-hop neighbours. The station with the lowest decision value becomes the CC. All 1-hop neighbours join this sub-net (as long as capacity is available). There are two possibilities how the decision values of the stations can be compared against one another. One possibility would be that all stations broadcast their own decision values to their neighbours (like in case of the LID and HIC algorithms). Another possibility is that the CC takes the decision based on the RSS matrix described above or that the CC broadcasts this matrix to the MTs which subsequently decide whether to become CC or not.

The *Highest In-Cluster Traffic* (ICT) algorithm builds clusters based on the traffic of each station with its direct neighbours, to minimise the forwarding traffic between clusters. Every station knows its 1-hop neighbours and can calculate its total traffic with them. The station with the highest direct neighbour traffic is selected as CC. All 1-hop neighbours of this station join the sub-net (as long as capacity is available). There are again the possibilities that either each station broadcasts its

[1] The term Lowest Distance refers to the fact that in a free-space-scenario the distance is equivalent to the RSS value.

total data rate to its neighbours or that the CC decides based on the global traffic matrix.

All algorithms have in common that a re-clustering is carried out every time the decision criterion recommends it. This leads to quite frequent re-clustering events and therefore to an unstable network structure. To cope with this problem we propose the *Check Changes* algorithm which can be applied in combination with any of the above algorithms. At first, the algorithm tests if the old cluster configuration can be kept further on. Only if the old configuration is not possible, the normal clustering algorithm is called.

B. A new Clustering Algorithm

Simulations in a free-space-scenario in [1] have shown that the LID algorithm is the most stable one, but that is has not been applicable because it does not consider the traffic situation inside the clusters.

We therefore propose an algorithm, called *Highest-ID-with-traffic* (HID), which is an extension of the basic LID algorithm. The HID foresees that the station with the highest ID becomes the first CC. This new CC associates its direct neighbours to the cluster in ascending order starting with the one with the lowest ID. In contrast to the LID MTs can be associated to the cluster only as long as capacity is available. If all capacity in the cluster is occupied, an additional cluster will be opened. The CC of the additional cluster will be the station with the second highest ID. This station could certainly not be a member of the first cluster, because the association of MTs proceeds from lower to higher IDs in ascending order. Additional clusters are opened as long as MTs have not become members of a cluster.

The station with the highest ID will become the CC of the first cluster. The station with the second highest ID will notice if it has been accepted in this new cluster or not. If it has not been accepted, the station will decide locally to open an additional cluster, as it knows that it is the station with the second highest ID. The same procedure applies to all further clusters that might be opened. Note that a station with the second highest ID, which is not accepted in the cluster of the station with the highest ID and which therefore opens an additional cluster, may not associate all the remaining direct neighbours of the station with the highest ID to the additional cluster. Instead, the station with the second highest ID will associate all those MTs to its cluster that are *its own* neighbours and that have not yet been associated to a cluster of *any* possible CC in the previous clustering stage.

III. Mobility and Channel Model

We will compare the performance of the new algorithm with the other clustering algorithms in a more realistic scenario than the free-space-scenario in [1], namely in an office scenario. For this purpose we have implemented a new mobility and channel model which will be presented in the next sections.

A. An Office Mobility Model

In this model stations move on straight lines between a set of nodes. The nodes are either endpoints or cross-over points of lines. They are stored in a matrix which contains for each node: its position, the mean waiting time of a station at the node as well as a list of those nodes that are connected to the respective node by a line. An example scenario is shown in Fig. 2.

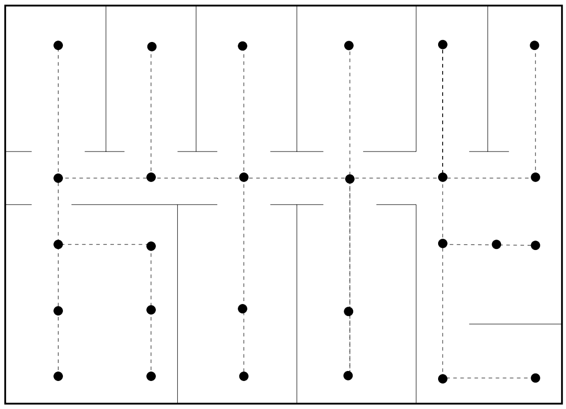

Fig. 2: Example scenario

At the start of the simulation run, each station has a starting position, which will be associated to a starting node. Each time a station leaves a node, it randomly chooses one of the neighbouring nodes as its new destination. Subsequently it moves to this node on a straight path. Arrived at the destination node, a station stays there for a randomly chosen waiting time, before it leave in direction of the next drawn destination.

B. The Multi-Wall Model

At the frequency used for HiperLAN/2 (5.2 GHz) the reduction of the gain due to path-losses is a very important factor. This value is needed in order to calculate the C/I value every time a PDU-train is sent from one WT to another.

For a stationary scenario the path-losses between the stations are usually calculated at their fixed positions by a ray-tracing algorithm before the simulation starts.

The computation of the ray-tracing algorithm is very time-consuming. Therefore, as applied to mobile systems it is only practical to simulate the fading channel of systems in which either the transmitter or the receiver have a fixed position.

Unfortunately, mobility is a very important characteristic of our scenario. Hence, our simulator needed a faster way to calculate the path-losses. These sort of algorithms are based on *Empirical Indoor Models*. The empirical model chosen for our simulator is the *Multi-Wall Model* (MWM) and has been investigated in COST Action 231 [6].

The multi-wall model gives the path loss as the sum of the free space loss and the losses introduced by the walls and floors penetrated by the direct path between the transmitter and the receiver. The attenuation in the model can then be expressed as

$$L = 10\log\left(\frac{(4\pi f)^2 d^n}{c_0}\right) + \sum_{i=1}^{I} k_{wi} L_{wi} + k_f^{\frac{k_f+2}{k_f+1}-b} L_f$$

where

f = frequency of the signal,
c_0 = propagation velocity of the signal,
n = power decay index (2 for air),
d = distance between transmitter and receiver,
k_{wi} = number of penetrated walls of type i,
L_{wi} = loss of wall type i,
I = number of wall types,
k_f = number of penetrated floors,
L_f = loss between adjacent floors,
b = empirical parameter.

IV. Performance Evaluation in an Office Scenario

An overview of the chosen office scenario can be found in Fig. 2. One can identify 10 rooms and one narrow hall. 14 stations move from one room to another according to the mobility model. Simulations have been carried out using our simulation tool WiLMA (Wireless LAN Multihop Ad-hoc).
In Fig. 3 the LID, LDV and ICT are compared against the HID as far as the average number of clusters is concerned.

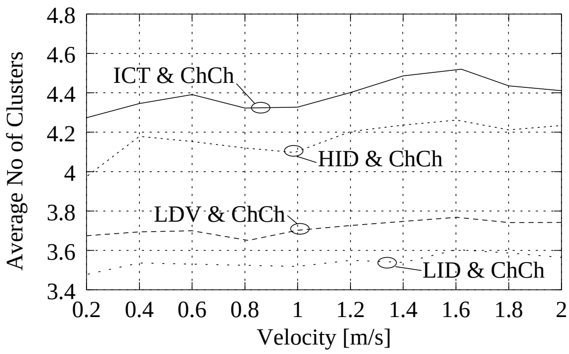

Fig. 3: Average number of clusters

The lowest number of clusters can be achieved with the LDV. This is a logical result, as the LDV is able to consider geographical clouds of stations. The HID gives a slightly higher number of clusters than the LID, because additional clusters are opened if capacity limits in one cluster are reached. The ICT results in the highest number of clusters.
The number of clusters is not a very decisive performance measure, because a higher number of clusters may not necessarily result in poorer system performance. A much more important criterion of the applicability of an algorithm is its stability. The stability can be measured by the average number of CC Handovers, that is to say re-clustering events, per time as illustrated in Fig. 4. In the simulations the Check Changes algorithm has always been activated as it has proven to significantly improve the stability.

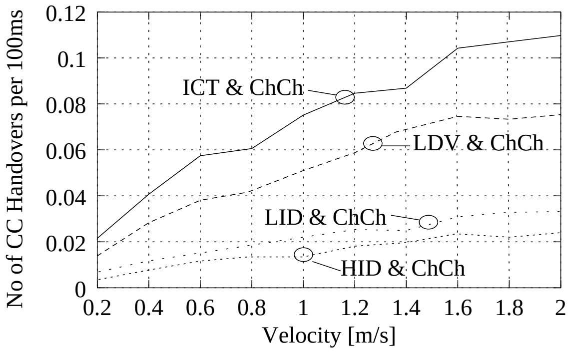

Fig. 4: Number of CC Handovers over velocity

It can be seen that the HID results in a network stability which is comparable to the stability of the LID. This is owing to the fact that obviously only rarely a MT is entering an existing cluster that has a higher ID than the current CC. LDV and ICT have not only a higher level of instability but also a stronger dependence on the velocity of the stations. The reason is that the algorithms react quite sensibly to topological changes of the network, because the optimisation criterion proposes a "better" constellation.

V. The Forwarding Problem

Once the clusters have been created, the next problem is to allow them to communicate with each other. In an ad-hoc network forwarders have to be chosen between the MTs that are in the range of at least two cluster heads (Central Controllers CC*s*). This is not a trivial problem, because the CCs have only access to local information about the network structure. They only *"know"* which MTs are situated in their own cluster. CCs are mostly not allowed to change their frequency in order to find MTs from other clusters.
Thus, MTs that are situated at a certain distance from their CCs must regularly scan the whole allowed frequency spectrum in order to find out if they are on the coverage area of other CCs. If it is the case, they have to inform the CCs that they can be used as forwarder to establish a link between two clusters. This will permit the CCs to create a list of possible forwarders. This information has to be treated by every CC in order to choose the best forwarding constellation for its own cluster.

A. Example of a Forwarding Problem

An example of a forwarding problem is shown on Fig. 5. It illustrates the optimal forwarding choices for a Central Controller called CC1.
In Fig. 5 forwarders T2 and T3 are both possible candidates for interconnection of cluster 1 and

cluster 3. Forwarding between clusters 2 and 3 can only be performed by forwarder T2.

For the efficiency of the forwarding process one forwarder can only interconnect two clusters. If forwarder T2 was chosen for interconnection of clusters 1 and 3, no forwarder would be left for interconnection of clusters 2 and 3. This situation has to be avoided. Therefore, first of all forwarders have to be installed between those clusters, where the least number of forwarding candidates exist.

A new algorithm is presented here which allows every CC to determine its optimal forwarding constellation.

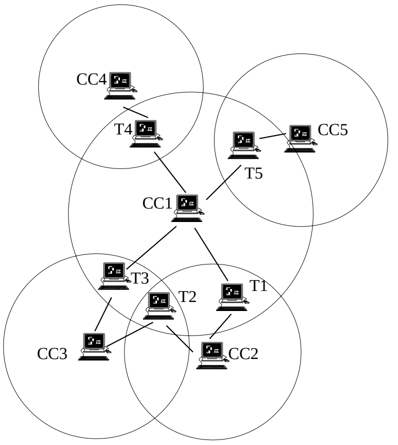

Fig. 5: Example of a Forwarding Problem

B. A new Forwarder Selection Algorithm (FSA)

The whole algorithm is repeated by every CC. The data is stored locally at every *CC*. To prevent confusion, the CC that is performing the algorithm at this moment is called *Processing* Central Controller (PCC).

All possible forwarders are inscribed on a three-dimensional array **F(**1..*n*;1..*n*;1..*t***)**. *n* being the number of CCs known by the PCC and *t* being the number of possible forwarders. If *m* forwarders are able to establish a connection between two CCs (*i* and *j*, *i*<*j*), their Ids will be entered on the array at **F(***i*,*j*,1..*m***)**. The Ids of the forwarders are sorted by link quality, so that the Id of the forwarder which offers the best link quality is registered at **F(***i*,*j*,*1***)**. The array is illustrated in Fig. 7.

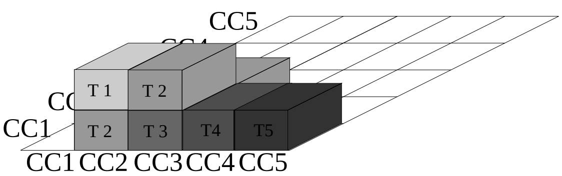

Fig. 7: Three-dimensional array

The following algorithm is repeated until the matrix is empty.

-At every step the PCC scans the array for the link (*i*,*j*, *i*<*j*) with the smallest *m*.

-The terminal with the Id t_k registered at **F(***i*,*j*,1**)** is chosen as forwarder between *i* and *j*.

-All the entries for this link **F(***i*,*j*,1..*m***)** are removed.

-The Id t_k is searched and removed over the whole array.

This algorithm chooses the best forwarders between the MTs situated on the coverage area of the *PCC*. It takes not only decisions concerning the forwarders connecting its cluster with other clusters, but also concerning links between clusters, other than its own cluster. This allows the algorithm, if performed by all *CCs*, to find not only local optimal constellations, but also a global optimal forwarding system.

VI. Conclusion

In this paper a new clustering algorithm called Highest-ID-with-traffic (HID) as well as a new Forwarder Selection Algorithm (FSA) for multihop wireless ad hoc networks have been presented. The new clustering algorithm has been evaluated in a realistic office scenario using appropriate mobility and channel models. The HID has proven to be much more stable than previously developed clustering algorithms. It is superior to the well-known LID algorithm in the sense that the traffic capacity of clusters is considered thereby maintaining the Quality of Service in the network.

The new Forwarder Selection Algorithm guarantees that all clusters are interconnected, if a physical possibility of interconnection does exist.

Our current work focuses on the combination of clustering rules and the consideration of new aspects like the velocity of the stations in the sense that stations with lower velocity have a higher chance of becoming a Central Controller.

REFERENCES

[1] J. Habetha, M. Nadler, “Concept of a Wireless Centralised Multihop ad hoc Network”, in *Proc. European Wireless 2000,* Sept. 2000.

[2] F. Kamoun, L. Kleinrock, “Hierarchical routing for large networks: Performance evaluation and optimization”, *Comput. Networks*, Vol. 1, pp. 155-174, 1977.

[3] M. Gerla, J. Tzu-Chieh Tsai, “Multicluster, mobile, multimedia radio network”, *Wireless Networks*, Vol. 1, pp. 255-265, Oct. 1995.

[4] J. Habetha, A. Hettich, J. Peetz, Y. Du, “Central Controller Handover Procedure for ETSI-BRAN HiperLAN/2 Ad Hoc Networks and Clustering with Quality of Service Guarantees”, in IEEE Annual Workshop on Mobile Ad Hoc Networking and Computing (MOBIHOC), Aug. 2000.

[5] C. R. Lin, M. Gerla, “Adaptive Clustering for Mobile Wireless Networks”, *IEEE J. Select. Areas Commun.*, Vol. 15, No. 7, pp. 1265-1275, Sept. 1997.

[6] J. Lähteenmäki, “Indoor Propagation Models”, in *COST 231 “Digital mobile radio towards future generation systems” - Final report*, European Commission, 1999.