Distributed Dynamic Channel Allocation for Mobile Computing

Ravi Prakash
Department of Computer and Information Science
The Ohio State University
Columbus, OH 43210.
e-mail: prakash@cis.ohio-state.edu

Niranjan G. Shivaratri
4 Independence Way
NEC Systems Laboratory, Inc.
Princeton, NJ 08540.
e-mail: niran@syl.nj.nec.com

Mukesh Singhal
Department of Computer and Information Science
The Ohio State University
Columbus, OH 43210.
e-mail: singhal@cis.ohio-state.edu

Abstract

Efficient allocation of communication channels is critical for the performance of wireless mobile computing systems. The centralized channel allocation algorithms proposed in literature are neither robust, nor scalable. Distributed channel allocation schemes proposed in the past are complicated and require active participation of the mobile nodes. These algorithms are unable to dynamically adjust to spatial and temporal fluctuations in channel demand. We present a dynamic distributed channel allocation algorithm that can quickly adapt to changes in load distribution. The algorithm described in this paper requires minimal involvement of the mobile nodes, thus conserving their limited energy supply. The algorithm is proved to be deadlock free, starvation free and fair. It prevents co-channel interference and is scalable.

1 Introduction

Mobile computing has found increased applications and gained importance in recent years [4]. Mobile computing makes use of cellular/wireless communication networks to provide communication among stationary and mobile hosts. In such environments, efficient allocation of wireless channels for communication sessions is of vital importance as the bandwidth alloted for cellular communication is limited.

Cellular communication networks divide the geographical area they serve into smaller regions, called cells. Each cell has a base station, also referred to as the *mobile service station (MSS)*. The mobile service stations are connected to each other by a fixed wire network. To establish a communication session/place a call, a *mobile host (MH)* has to send a request to the MSS of the cell in which it is present [2]. The call can be supported if a wireless channel can be allocated for communication between the mobile host and the mobile service station. If a particular wireless channel is used concurrently by more than one call originating in a cell, or in neighboring cells, the calls will interfere with each other. Such an interference is called *co-channel interference*. However, the same wireless channel can be used to support calls in geographically separated cells such that their signals do not interfere with each other. This is known as *frequency reuse*.

The limited frequency spectrum allocated for cellular communication is divided into a finite number of wireless channels. An efficient channel allocation strategy should exploit the principle of frequency reuse to increase the availability of wireless channels to support calls. The strategy should have the following features:

1. minimize the connection set-up time
2. maximize the number of communication sessions that can be supported concurrently across the entire network
3. ability to adapt to changing load distribution in the network. The load on a cell is the rate at which new requests for establishing communication sessions originate in the cell.

To support mobile computing, the strategy should also meet the following requirements:

1. *Energy conservation:* most of the communication between mobile computers is in the form of several

PODC 95 Ottawa Ontario CA © 1995 ACM 0-89791-710-3/95/08..$3.50

short bursts of data transfer. The mobile hosts have a limited energy source, in the form of a battery pack. Wireless communication drains the energy of the mobile hosts. Hence, energy should be conserved at a mobile host by keeping its involvement in the channel allocation process to a minimum. This can be achieved by minimizing the number of messages it has to exchange with the mobile service station during channel selection.

2. *Minimize hand-offs:* voice communication can tolerate hand-offs as short breaks in communication go undetected by the human ear. However, such breaks can lead to complications in data transfer to/from a mobile host. So, a channel allocation algorithm should not induce any hand-offs, over and above those caused by the movement of the mobile hosts between cells.

3. *Exploit locality of reference:* most computer applications exhibit high temporal and spatial locality of data reference. If the data items in great demand reside in a mobile host, they should be moved to a mobile service station from where they can be accessed over the fixed wire network. Until such a transfer takes place, or if such a transfer is not possible, the data references translate into frequent arrivals of requests at the mobile service station to establish communication sessions with the mobile host. A channel allocation strategy should be able to adapt to such traffic.

The channel allocation algorithms proposed in the past can be classified as

1. *Fixed Channel Allocation (FCA) strategy*: the set of channels allocated to a cell does not change with time. Mutually disjoint sets of wireless channels are assigned to neighboring cells. Each cell can use only its set of channels (the *nominal channels* of the cell) to support the calls originating from and/or directed towards the mobile hosts in its region.

2. *Dynamic Channel Allocation (DCA) strategy*: the set of channels allocated to a cell varies with time [14, 15]. A central network switch, referred to as the *Mobile Telecommunication Switching Office (MTSO)*, determines the channel(s), if any, a cell can borrow from neighboring cells, when the cell cannot support calls using its own set of channels. The central switch ensures that the borrowing does not lead to any co-channel interference.

In this paper we present a distributed, dynamic channel allocation algorithm. The algorithm does not need a central network switch. The mobile service station of a cell makes all the decisions about channel allocation in that cell, based on the information available locally. The MSS only needs to exchange information with its neighbors within the co-channel interference range. Unlike the FCA algorithms, the proposed algorithm can adapt to changing load distribution in the network. It is more robust than existing DCA algorithms as it does not depend on a central network switch whose failure can bring down the entire network. The algorithm also exploits the temporal locality of load distribution to make quick decisions about channel allocation. Moreover, a fast and expensive mainframe acting as the MTSO can be replaced by a set of microprocessor based switches at the $MSSs$. These switches can collectively outperform the mainframe and cost much less. The symmetry of the channel allocation procedure across the entire network makes the system scalable. The proposed algorithm meets the requirements mentioned above: it conserves energy at the mobile hosts, does not induce any hand-offs of its own, and exploits locality of reference to improve the performance. Preliminary results from a simulation study support the above assertions.

The proposed algorithm also has the following features:

1. *Bounded latency*: no mobile user that wishes to acquire a wireless channel for a communication session is made to wait indefinitely before it is either allocated a channel or is informed of a failure to do so. Bounded latency is desirable to guarantee a certain quality of service to the users.

2. *Deadlock freedom*: there is no possibility of finding a set of mobile service stations involved in a circular wait while trying to satisfy channel allocation requests. So, the algorithm always makes progress. Resources are not wasted in detecting or resolving deadlocks.

3. *Symmetry*: All the cells follow the same procedure for channel allocation. This makes the system scalable. There is no need to design new hardware, or develop new software if more cells are to be added.

4. *Finite number of messages*: Each new request for channel allocation, to support a communication session, originating in the system leads to the exchange of a finite number of messages between the mobile service stations, before a decision to allocate a channel to it is made. Thus the fixed wire network connecting the mobile service stations is not unduly burdened.

5. *Low System Overhead and Network Traffic*: As the proposed algorithm adapts to the locality of load

distribution, each new channel allocation request is handled with an exchange of zero or a small number of messages between the mobile service stations.

6. *Concurrency*: requests for channel allocation originating independently and concurrently in different cells can be processed simultaneously.

Section 2 describes the system model. Section 3 compares the channel allocation problem with the problem of mutual exclusion in distributed systems and describes why the channel allocation problem is more complex than mutual exclusion. In Section 4, a dynamic distributed channel allocation algorithm is presented. The algorithm can adjust to changes in the temporal and spatial distribution of channel demand. In Section 5, we prove that the proposed algorithm avoids co-channel interference, is deadlock free and has low communication overheads. The advantages of the proposed algorithm over previous centralized and distributed channel allocation algorithms are described in Section 6. Section 7 presents enhancements to the algorithm that prevent cells from being starved for channels, and also relax channel transfer constraints which will lead to a reduction in blocking probability of channel requests. Finally, conclusions are presented in Section 8.

2 System Model and Definitions

We assume a cellular communication system that divides the geographical region served by it into hexagonal cells, with a mobile service station in the center of each cell. A mobile service station can be in wireless communication with the mobile hosts in its cell (for example, through an omni-directional antenna). A mobile host can either be a cellular telephone or a mobile computer. Calls involving cellular telephones and data transfers involving mobile computers will collectively be referred to as *communication sessions*. All the cells, except those at the boundaries of the region, have six neighbors. The system has been assigned a frequency band that is divided into a finite number of wireless channels. These channels are independent (orthogonal) of each other. So, adjacent channel interference can be neglected. However, a channel should not be concurrently used for more than one communication session in the same cell or in neighboring cells. Some of the wireless channels are set aside to be used exclusively for the control messages sent during link set-up between a mobile host and the mobile service station of the cell in which the mobile host is present (*control channels*). The remaining channels are used to support calls (*communication channels*).

A mobile host can communicate with other units, mobile or static, only through the mobile service station of the cell in which it is present. A mobile host initiates the channel allocation protocol when it wants to establish a new communication session, or when it is informed by the mobile service station about the arrival of a communication request from some other unit. Thus, from the point of view of channel allocation, the two cases are similar. If the mobile service station determines that the connection request can be satisfied, it allocates a communication channel for the mobile host to communicate with the mobile service station for the duration of the session. From the mobile service station the signals can be forwarded along the fixed wire network, or along another wireless channel, depending on whether the other party involved in the communication session is a unit outside the cell or a mobile host in the same cell, respectively. After the session is over, the same channel can be used to support another session, either in the same cell or in neighboring cells.

For simplicity of explanation, inter-cell movement, and the resultant hand-off, can be treated as the end of the communication session in the cell from which the mobile host has moved out, and the beginning of a new communication session in the cell to which it has moved. However, to maintain continuity of service, the resultant channel allocation for hand-off should be assigned higher priority than requests for new communication sessions. Two priority schemes for hand-offs have been proposed and evaluated in [5].

3 Channel Allocation vs. Mutual Exclusion

In the context of a cell and its neighbors, the use of a particular channel to support a communication session is equivalent to a critical section execution by the cell in which the channel is being used. Several neighboring cells may be concurrently trying to choose channels to support sessions in their region. This can lead to conflicts because the number of communication channels is limited. The resolution of such conflicts is similar to the mutual exclusion problem [6, 11].

However, the channel allocation problem is more general than the mutual exclusion problem. Firstly, a cell may be supporting multiple communication sessions, from different mobile hosts, in its region, each session using a different communication channel. This is equivalent to a cell being in multiple, distinct critical sections concurrently. Secondly, existing mutual exclusion algorithms for distributed systems [6, 7, 9, 10, 11] assume that a node specifies the identity of the resource it wants to access in a critical section. Depending on the availability of that resource, appropriate decisions can be made. However, in distributed channel allocation, a cell asks for *any* channel as long as there is no co-channel interference. Due to the non-specificity of the request and because neighboring mobile service stations make channel allocation decisions independently based on locally available information, the decision process becomes more difficult.

Moreover, existing distributed mutual exclusion algorithms do not impose any upper bound on the time from the instant a node issues a request for the resource to the instant the node is granted that resource. These algorithms are not suitable for the channel allocation problem that requires the decisions to be made quickly, in real-time. So, a conservative approach that makes the channel allocation decisions quickly needs to be adopted. Such an approach may drop calls/communication requests that a more general but time consuming approach would have supported. This is a trade-off that has to be accepted.

4 A Dynamic Channel Allocation Algorithm

In the proposed algorithm, a mobile service station makes all the channel allocation decisions on behalf of the mobile hosts in its cell. Requests timestamped with Lamport's clock [7] are sent by a mobile service station to neighboring mobile service stations to determine the channel to be assigned for a communication session. Sometimes a channel needs to be deleted from a cell's set of allocated channels and transferred to another cell's set of allocated channels to support communication sessions in the latter. The distributed nature of the algorithm, and the finite but non-deterministic propagation delays of messages between mobile service stations can lead to co-channel interference if a naive channel transfer strategy is employed: multiple cells in each other's interference range may concurrently and independently decide to transfer the same channel from a mutually adjacent cell. Such a possibility is prevented as follows: having selected a communication channel for transfer, based on a round of message exchange with its neighbors, the mobile service station sends the channel identity to the neighboring mobile service stations. Only if all the neighboring mobile service stations approve of the selection is the channel transferred, otherwise not.

The set of channels allocated to a cell varies with time. *Unlike existing DCA algorithms [14, 15], a newly acquired channel is not relinquished by a cell on completion of the communication session it was supporting in the cell.* Instead, the channel remains allocated to the same cell until it has to be transferred to a neighboring cell. This enables the algorithm to adapt to temporal and spatial changes in load distribution. It also helps reduce the traffic due to channel allocation requests in the fixed wire network.

4.1 Data Structures

All the communication channels in the system are collectively represented by a set *Spectrum*. We assume that all the channels are ordered. The channel with the lowest frequency band is considered to be the first channel and the channel with the highest frequency band is the n^{th} channel, where n is the total number of channels available.

The set of channels allocated to cell C_i is represented by $Allocate_i$. Initially, $Allocate_i$ is an empty set for every cell C_i. A subset of $Allocate_i$, known as $Busy_i$, represents the set of channels being used by C_i to support communication sessions at a particular instant of time. When a new communication request originates in C_i, one of the non-busy channels in $Allocate_i$ is assigned to support the communication session. If there is no such channel, then after a round of message exchange with the neighbors, a channel that is in the *Spectrum*, but not in the *Allocate* set of the cell or any of its neighbors is added to $Allocate_i$ as well as $Busy_i$. This channel is used to support the session. If such an attempt fails, C_i tries to transfer a non-busy channel from the *Allocate* set of its neighbors to $Allocate_i$. If such a transfer is not possible, the communication request is dropped. Otherwise, the communication is successfully completed. The set $Transfer_i$ at C_i consists of the channels earmarked for transfer from C_i to one of its neighbors. *Transfer* sets are initially empty at all the cells. All these sets are maintained by the corresponding mobile service stations.

Several new communication requests may originate in a cell concurrently. These new requests, originating in the same cell, may be ordered according to a policy decided *a priori*. Only after the mobile service station has made a channel allocation decision about one locally originating request, does it process the next locally originating communication request in the sequence.

4.2 The Algorithm

(A) When a communication session is to be set-up in cell C_i, the following actions are taken by its mobile service station (MSS):

1. If $Available_i \leftarrow Allocate_i - Busy_i - Transfer_i \neq \Phi$, then

    A highest order channel k from $Available_i$ is selected to set-up the session;
    $Busy_i \leftarrow Busy_i \cup \{k\}$;
    Go to step 8;

   else /* $Available_i = \Phi$ */

Send timestamped REQUEST messages to each neighbor C_j.

2. When C_i's MSS has received REPLY messages from each of its neighbors, containing their *Allocate, Busy and Transfer* sets, it takes the union of $Allocate_i$ and the *Allocate* sets received in the REPLY messages, and stores the result in $Interfere_i$.

3. If $Free_i \leftarrow Spectrum - Interfere_i \neq \Phi$, then a channel of the highest order is selected from $Free_i$ and added to $Allocate_i$. This channel is used to support the communication session. So, it is added to $Busy_i$ as well. Then go to step 8.

4. If $Free_i = \Phi$, it does not mean that no channel is available for allocation. Perhaps, the communication session can be supported by transferring a channel. C_i's MSS takes the union of $Busy_i$, $Transfer_i$, and $Busy$ and $Transfer$ sets received in the REPLY messages in step 2, and stores the result in $Interfere_i$.

5. If $Free_i \leftarrow Spectrum - Interfere_i = \Phi$, then the communication request is dropped. Otherwise, the channel of the lowest order in $Free_i$ is chosen for the transfer.

6. Let the channel selected for transfer be k.
   $Busy_i \leftarrow Busy_i \cup \{k\}$;
   $Allocate_i \leftarrow Allocate_i \cup \{k\}$;
   C_i's MSS sends TRANSFER(k) messages to all the neighbors whose *Allocate* sets have k as a member and waits for replies. Let S denote the set of these neighbors.

7. If all the cells in S reply AGREED:

   Channel k is used to support the communication session.
   C_i's MSS sends RELEASE(k) messages to all the cells in S.
   Go to Step 8.

   Otherwise: /* Some cells have sent REFUSE message. */

   $Allocate_i \leftarrow Allocate_i - \{k\}$;
   $Busy_i \leftarrow Busy_i - \{k\}$;
   C_i's MSS sends KEEP(k) messages to all the cells in S.
   C_i's MSS selects the next channel from $Free_i$, with order greater than that of k, and steps 6 and 7 are repeated [1]. To avoid excessive channel transfer overheads, under heavy load situations, the number of transfer attempts can be limited to the minimum of a THRESHOLD value (parameter of the algorithm) and the cardinality of $Free_i$. If all attempts to transfer a channel fail, the communication request is dropped.

8. Once a cell has decided to drop a request or to use a channel to support the corresponding communication session, it sends all the deferred REPLYs to its neighbors.

9. When a communication session terminates in C_i, the corresponding channel is deleted from the set $Busy_i$.

(B) When a cell C_j's MSS receives a REQUEST message from C_i's MSS:

C_j's sends a REPLY message to C_i if C_j is not requesting a channel, or if C_j is requesting a channel and C_i's request's timestamp is smaller than C_j's request's timestamp. The REPLY message contains $Allocate_j$, $Busy_j$, and $Transfer_j$. Otherwise, the REPLY is deferred.

(C) When a cell C_j's MSS receives TRANSFER(k) message from C_i:

If ($k \in Busy_j$) OR ($k \in Transfer_j$) then send REFUSE(k) message to C_i. Otherwise $Transfer_j \leftarrow Transfer_j \cup \{k\}$; Send AGREED($k$) message to C_i.

(D) When C_j's MSS receives a RELEASE(k) message, the following actions take place.

$Allocate_j \leftarrow Allocate_j - \{k\}$;
$Transfer_j \leftarrow Transfer_j - \{k\}$;

(E) When C_j's MSS receives KEEP(k) message, the following actions take place.

$Transfer_j \leftarrow Transfer_j - \{k\}$;

5 Correctness Proof

Lemma 1 *The channel allocation algorithm ensures that neighboring cells do not use the same channel concurrently.*

Proof: Let Nbr_i denote the set of neighboring cells of C_i such that concurrent use of a channel in C_i and a cell in Nbr_i will lead to co-channel interference. We have to prove the following assertion: $Busy_i \cap Busy_j = \Phi$,

[1] The KEEP messages can be piggybacked on TRANSFER messages, if they are going to the same cell.

$\forall C_j \in Nbr_i$.
Initially, the assertion is trivially true as the sets are empty. Also, $Busy_i \subseteq Allocate_i$ under all circumstances. $Busy_i$ can change under three situations:

1. *In step (A).1, when $Available_i \neq \Phi$*: Let cell C_i select channel k (an element of $Allocate_i$) to support a new communication session. Assuming $Busy_i \cap Busy_j = \Phi$ and $Allocate_i \cap Allocate_j = \Phi$ prior to the addition of k to $Busy_i$, $(Busy_i \cup \{k\}) \cap Busy_j = \Phi$. So, the assertion holds after k is selected to support a call in cell C_i.

2. *$Available_i = \Phi$ in step (A).1 and $Free_i \neq \Phi$ in step (A).3*: Channel $k \in Spectrum$ - $(Allocate_i \bigcup_{j \in Nbr_i} Allocate_j)$ is added to $Busy_i$ and $Allocate_i$. The assertion is proved by contradiction. Let us assume that cell C_i, and its neighbor C_j, are using channel k concurrently. Cell C_j does not transfer channel k to its neighbor C_i as long as $k \in Busy_j$. This implies that the co-channel interference mentioned above can arise only if the *Allocate* sets in the REPLYs received by the mobile service stations from each other in step (A).2 did not contain k. Based on the pattern of REQUEST and REPLY messages exchanged between the two nodes, the following three situations arise:

   (a) C_i sends a REPLY to C_j before sending its own REQUEST. So, C_i's REQUEST has a higher timestamp than C_j's REQUEST. When C_j receives this REQUEST, it defers the REPLY until it has decided to use k. Then C_j sends its *Allocate* set, containing k, in the REPLY to C_i. So, C_i cannot select channel k.

   (b) C_j sends a REPLY to C_i before sending its own REQUEST. This is the similar to the previous case. So, C_i selects channel k, while C_j does not.

   (c) Both C_i and C_j receive each other's REQUEST after sending their own REQUESTs. Both the cells compare their own channel request timestamp with that received in the REQUEST message from the other. As the timestamps are fully ordered by the Lamport's clock system, the cell whose request happens to have the lower timestamp among the two requests, will defer its REPLY until it has made its own decision. The other cell will send a REPLY. Let C_i be the cell that deferred the REPLY. If C_i decides to use k, then C_j receives this information (Allocate set) in the REPLY it receives from C_i. So, C_j will not use channel k.

   Thus, two neighboring cells will not be allocated the same channel concurrently.

3. *$Free_i \neq \Phi$ in step (A).5 and AGREED messages received from all cells in S in step (A).7*: Channel $k \in Spectrum- (Busy_i \bigcup Transfer_i \bigcup_{j \in Nbr_i} Busy_j \bigcup_{j \in Nbr_i} Transfer_j)$ is added to $Allocate_i$ and $Busy_i$. TRANSFER(k) is sent to the neighbors. If any neighboring cell is using k, it sends a REFUSE message. So, channel k is not used in cell C_i. From steps (C), (D), and (E) it can be inferred that in response to a TRANSFER(k), a cell C_j sends AGREED to at most one neighbor at any time. All other TRANSFER(k) messages received by C_j, after k is added to $Transfer_j$ and before RELEASE(k) or KEEP(k) are received, are responded to with a REFUSE message. Therefore, two neighboring cells cannot simultaneously acquire channel k as a result of a transfer attempt.

∎

Lemma 2 *Each new request for a communication session originating in a cell C_i causes a finite number of messages to be exchanged between the mobile service stations of the cell and its neighbors.*

Proof: Three situations can arise. If the channel request can be satisfied locally (step (A).1), no messages are exchanged between the mobile service stations. If $Free_i \neq \Phi$ in step (A).3 at most $2N$ messages are generated to allocate a channel to the communication session, where N is the number of neighboring cells in the co-channel interference range: N REQUESTs from C_i to its neighbors, and a REPLY from each neighbor to C_i. If $Free_i = \Phi$ in step (A).5, the request is dropped after the exchange of the same $2N$ messages. Finally, if $Free_i \neq \Phi$ in step (A).5, $5N \leq$ *messages needed to make a channel allocation decision* $\leq 2N + 3N \times minimum(\mid Free_i \mid, THRESHOLD)$. Besides the $2N$ messages already mentioned, at most N TRANSFER(k) messages from C_i to its neighbors, an AGREED or REFUSE message from each neighbor to C_i, and finally a RELEASE(k) or KEEP(k) to each neighbor are needed per channel transfer attempt. The number of attempts is upper bound by the minimum of $\mid Free_i \mid$ and THRESHOLD. As THRESHOLD is a constant value chosen as a parameter of the algorithm, the message complexity is $O(N)$. ∎

Lemma 3 *The channel allocation algorithm is deadlock free.*

Proof: New channel requests originating concurrently in different cells get totally ordered by their timestamps.

A mobile service station with REPLYs pending to its own REQUESTs, sends REPLYs to all REQUESTs with a lower timestamp and defers other REPLYs. As the same ordering of calls is seen by all the nodes, there is no circular deferring of REPLYs among the mobile service stations.

During the interval between sending a TRANSFER(k) message to the neighbors, and receiving either a REFUSE or an AGREED message from each neighbor, a cell does not suspend replying to TRANSFER(k) messages it may itself receive from the neighbors. Instead, it responds to such transfer attempts with a REFUSE message during this interval. This conservative policy may lead to some requests, that could have otherwise been supported, being dropped. However, it avoids any circular wait during the channel transfer attempts, thus preventing deadlocks. ∎

6 Comparison with Earlier Work

The proposed algorithm has several advantages over existing channel allocation algorithms. In the centralized algorithms, the central network switch is the single point of failure that can bring down the entire network, and can become a bottleneck during high load situations. The proposed algorithm is more robust and does not have a bottleneck as the traffic is distributed over the entire network. Each mobile service station shoulders responsibility. The size of the messages is also small because very little information is exchanged.

The algorithm adapts well to changing load distribution. Due to statistical fluctuations, there may be temporally and spatially distributed pockets of high load in the system. High load situations may also arise due to the locality of reference to data items residing on mobile hosts present in those hot-spots. The channel transfer feature of the proposed algorithm ensures that unused channels are moved from lightly loaded cells to the heavily loaded cells. Therefore, most of the channel requests that originate in heavily loaded cells can be satisfied locally by selecting a free channel from the *Available* sets. Moreover, if a mobile host, containing frequently accessed data, moves from a cell to a neighboring cell, channels are transferred from the *Allocate* set of the former to the latter, over a period of time. Thus the size of the *Allocate* sets of cells can adapt with time to support the locality of data reference.

The algorithm has low computational overheads. Most of the steps involve union, intersection or subtraction of sets of channels, which can be efficiently carried as operations on bit-streams, with a bit for each channel. As already mentioned, the hardware cost can be reduced, while maintaining the performance, by replacing an expensive mainframe, acting as the central network switch, with inexpensive microprocessor based controllers at the mobile service stations.

In [3], channel allocation is done by the mobile host and the mobile service station working together. In mobile computing, most of the communication is in the form of several short bursts of unidirectional data transfer, with unpredictable interval between two successive bursts. If the mobile host had to expend energy in channel allocation each time such a transmission is needed, it would soon become a significant overhead. In the proposed algorithm, the involvement of the mobile host in channel selection is limited to sending a request to its mobile service station for uplink connectivity and receiving a message from the mobile service station carrying the identity of the selected channel, if any. This leads to significant energy savings at the mobile host.

Complicated channel allocation strategies that constantly monitor the *signal-to-interference ratio* of channels, and employ cell sectoring [12] or cell overlaying [8], may be able to support higher loads than the proposed algorithm. However, these strategies have a higher probability of needing hand-offs [13]. In addition to inter-cell hand-offs, intra-cell hand-offs may be needed. For example, if cell sectoring is employed a cell is divided into multiple sectors. Some channels can be used to support communication sessions in only particular sectors of a cell. If a mobile host continues to use the same channel as it moves from one sector to another, in the same cell, co-channel interference may result. Hence, an intra-cell hand-off may be required. In the case of cell overlaying, the regular hexagonal grid of cells is overlaid with smaller cells. Channels used in the overlaid cells can be used at smaller distances. However, if a mobile host using such a channel moves out of the smaller overlaid cell while remaining in the bigger underlaid cell, co-channel interference may occur. Once again hand-offs may be required to avoid interference. Increased rate of hand-offs will lead to a significant degradation in the quality of service for data transmission to and from mobile hosts for two reasons. First, hand-offs impose computational overheads on the MHs and $MSSs$ in the system. Second, data communication has a much lower tolerance for hand-off induced temporary breaks in communication than voice communication. The proposed algorithm does not induce any intra-cell hand-offs. Hence, it does not suffer from the two problems mentioned above.

The distributed nature of the proposed algorithm makes the cellular network scalable. Channel allocation decisions are made by each mobile service station locally. All the messages needed to set up a communication session in a cell are restricted to that cell and its immediate neighbors. So, the traffic on the wired network between adjacent mobile service stations does not increase with increasing number of cells; it only increases with increasing load in the cell and its neighbors. For the centralized algorithms, the traffic on the communication paths leading to the central network switch increases with increasing network size. So, with the centralized algorithm, as the network expands, existing links will have to be replaced with those with a higher bandwidth.

7 Enhancements

While the algorithm is deadlock free, there is a possibility of some cells being starved for channels. For example, all the channels of a cell may be transferred to its neighbors. Later when the cell needs a channel, the neighbors may happen to be so highly loaded that the cell cannot acquire any channel from them.

Sometimes channel transfer attempts may be denied even if the transfer would not have caused co-channel interference. For example, let cell C_j add channel k to $Transfer_j$ on receiving a TRANSFER(k) message from C_i. Later, when another neighbor C_l of C_j tries to transfer channel k from C_j, it will be sent a REFUSE(k) message even if C_l and C_i are not neighbors of each other and cannot cause co-channel interference in each other's region.

The two issues mentioned above can be handled by making the following enhancements to the algorithm:

1. In cell C_i, instead of maintaining the set $Transfer_i$ for all the channels earmarked for transfer from C_i to its neighbors, the following sets are maintained: $\forall j : C_j \in Nbr_i$, $Transfer_{ij}$ is maintained. $Transfer_{ij}$ consists of the channels earmarked for transfer from C_i to C_j or to a cell that is a neighbor of C_i as well as C_j. Then $Transfer_i = \cup_{j \in Nbr_i} Transfer_{ij}$.

2. In step (B) of the algorithm, on receiving a REQUEST from C_i, C_j sends $Transfer_{ji}$ in the REPLY instead of $Transfer_j$.

3. In steps (D) and (E) of the algorithm, on receiving a RELEASE(k) or KEEP(k) message from C_i, C_j deletes the channel k from $Transfer_{ji}$ and $Transfer_{jl}$ for all C_l such that C_l is a neighbor of C_i as well as C_j.

4. Step (C) of the algorithm can be rewritten as:

   If ($k \in Busy_j$) OR ($k \in Transfer_{ji}$) OR ($\mid Allocate_j \mid < LOW\_THRESHOLD$) then send REFUSE($k$) message to C_i. Otherwise $Transfer_{ji} \leftarrow Transfer_{ji} \cup \{k\}$, $Transfer_{jl} \leftarrow Transfer_{jl} \cup \{k\}$ for all C_l such that C_l is a neighbor of C_i as well as C_j; Send AGREED(k) message to C_i.

LOW_THRESHOLD is a constant value introduced as a parameter of the algorithm. A cell will agree to transfer a channel to its neighbor only if it has at least LOW_THRESHOLD number of channels. Thus, channel starvation in cells is avoided. The degree of dynamism of the algorithm can be varied by changing the value of LOW_THRESHOLD.

Maintaining a transfer set with respect to each neighbor avoids the above mentioned problem of transfer requests being denied even if such transfers would not lead to co-channel interference.

The use of logical clocks to timestamp channel requests ensures fairness. If cell C_i's channel request causally precedes a neighboring cell C_j's channel request, C_i's request is processed before C_j's request. Thus, with the enhancements mentioned above, the algorithm ensures deadlock freedom, avoids starvation, and guarantees fairness.

A communication session of a mobile host may be disrupted if the mobile host moves from one cell to a neighboring *target* cell and no channel is available in the target cell to support the session. Such a disruption is referred to as hand-off failure.

Two priority based hand-off strategies have been proposed in [5]. These strategies can be easily incorporated in the algorithm described in Section 4. The two strategies are:

1. **Channel Reservation:** a certain number of channels (H) are reserved exclusively for hand-offs. Any available channel in the target cell can be used to support handed-off communication sessions. However, if a channel is needed for a new communication session arising in the cell and the number of available channels is less than H, the channel request is denied.

2. **Hand-off Queue:** channel requests for hand-offs are queued up on an FCFS basis in the target cell. If a channel becomes available in the target cell, and its hand-off queue is non-empty, the channel is used to satisfy the hand-off request at the head of the queue. An upper bound is imposed on the length of the queue and the duration for which a hand-off request can be queued up. This is to ensure that hand-offs, if possible, are done fast enough so that the disruption in the communication session during hand-off is not noticeable.

Simulation results in [5] show that both the strategies reduce the hand-off failure probability significantly. However, the first strategy leads to a corresponding increase in the probability of new channel requests being dropped. Hence, the hand-off queuing strategy appears to be the better of the two strategies.

Disruptions in the performance of the algorithm, due to the failure of some $MSSs$, can be handled by replicating the channel allocation information in an MSS at a few other $MSSs$ as described in [1]. Information replication can be done in either a pessimistic or an optimistic manner.

8 Conclusions and Future Work

An efficient channel allocation algorithm is important for high utilization of a cellular communication network. In this paper, we presented an efficient distributed dynamic channel allocation algorithm. The algorithm distributes the responsibility for channel allocation among the mobile service stations of the network. This is a departure from the algorithms proposed in the past, which employed a centralized controller.

The algorithm is especially suited for supporting mobile computing. It keeps the involvement of mobile hosts in channel selection to a minimum, thereby conserving the limited energy at their disposal. The algorithm keeps the number of hand-offs to a minimum as it does not induce any intra-cell hand-offs, unlike some strategies proposed in the past. It also exploits the locality of reference. The algorithm is dynamic and easily adapts to changes in the network load distribution by transferring allocated channels from lightly loaded cells to highly loaded cells.

The algorithm is deadlock free, fair and has low computational and communication overheads. The proposed enhancements prevent channel starvation in the cells. The cost of new hardware needed to implement the algorithm is low. The distributed nature of the algorithm and the symmetry of the channel allocation procedure across the entire network makes the system scalable. The simplicity of the algorithm makes it easy to implement on an actual network.

We are currently evaluating the performance of the algorithm through extensive simulations. Simulation results will be collected to determine the mean number of wireline messages needed per channel allocation request, the latency time for channel allocation, and the probability of channel requests being denied under high load situations. We intend to compare the performance of the algorithm with the performance of algorithms proposed in the past. We also intend to observe the performance of the algorithm for different distributions of request arrival rate and channel use duration per communication session.

References

[1] S. Alagar, R. Rajagopalan, and S. Venkatesan. Tolerating Mobile Support Station Failure. Technical Report UTDCS-16-93, University of Texas, Dallas, 1993.

[2] B. R. Badrinath, A. Acharya, and T. Imielinski. Structuring Distributed Algorithms for Mobile Hosts. In *Proceedings of the 14^{th} International Conference on Distributed Computing Systems*, June 1994.

[3] J. C.-I. Chuang. Performance Issues and Algorithms for Dynamic Channel Assignment. *IEEE Journal on Selected Areas in Communications*, 11(6):955–963, August 1993.

[4] G. H. Forman and J. Zahorjan. The Challenges of Mobile Computing. *IEEE Computer*, 27(4):38–47, April 1994.

[5] P.-O. Gaasvik, M. Cornefjord, and V. Svensson. Different Methods of Giving Priority to Handoff Traffic in a Mobile Telephone System with Directed Retry. In *Proceedings of the 41^{st} Vehicular Technology Conference*, pages 549–553. IEEE, 1991.

[6] K.M.Chandy and J.Misra. The Drinking Philosophers Problem. *ACM Transactions on Programming Languages and Systems*, 6(4):632–646, October 1984.

[7] L. Lamport. Time, Clocks and the Ordering of Events in a Distributed System. *Communications of the ACM*, 21(7):558–565, July 1978.

[8] W. C. Y. Lee. New Cellular Schemes for Spectral Efficiency. *IEEE Transactions on Vehicular Technology*, VT-36, November 1987.

[9] M. Maekawa. A $\sqrt{N}$ Algorithm for Mutual Exclusion in Decentralized Systems. *ACM Transactions on Computer Systems*, pages 145–159, May 1985.

[10] G. Ricart and A. K. Agrawala. An Optimal Algorithm for Mutual Exclusion in Computer Networks. *Communications of the ACM*, 24(1):9–17, January 1981.

[11] M. Singhal. A Dynamic Information-Structure Mutual Exclusion Algorithm for Distributed Systems. *IEEE Transactions on Parallel and Distributed Systems*, 3(1):121–125, January 1992.

[12] V. H. MacDonald. The Cellular Concept. *Bell System Technical Journal*, 58(1), January 1979.

[13] T.-S. P. Yum and W.-S. Wong. Hot-Spot Traffic Relief in Cellular Systems. *IEEE Journal on Selected Areas in Communications*, 11(6):934–940, August 1993.

[14] M. Zhang and T.-S. P. Yum. Comparisons of Channel-Assignment Strategies in Cellular Mobile Telephone Systems. *IEEE Transactions on Vehicular Technology*, 38(4):211–215, November 1989.

[15] M. Zhang and T.-S. P. Yum. The Nonuniform Compact Pattern Allocation Algorithm for Cellular Mobile Systems. *IEEE Transactions on Vehicular Technology*, 40(2):387–391, May 1991.