A Protocol for Seamless Communication in a Picocellular Network

Rohit Ghai and Suresh Singh
University of South Carolina
singh@cs.scarolina.edu

Abstract

In this paper we present an architecture for a picocellular network *and present a novel communication protocol for mobile users in such networks. The cell size in these networks is very small – typically of the order of a room in a building, resulting in an environment where the movement between cells is very rapid. This makes the problem of maintaining communication between a pair of mobile users non-trivial.*

Our communication protocol uses groups *to maintain communication between two mobile users. Groups are special collections of picocells and represent regions where a mobile user is likely to be located in the near future. Thus, messages addressed to this mobile user are multicast to all the picocells in the group guaranteeing delivery and obviating the need for messages to be forwarded between picocells. The mobile users will therefore see lower delays in message delivery times.*

1 Introduction

Developments in cellular communication and wireless LAN technology will make it possible, in the near future, for mobile users to transparently access communication networks from anywhere at any time. The mobile user will typically use powerful palmtop or laptop computers that are equipped with wireless communication capability to connect to wired networks. This type of computing environment is referred to as *mobile computing*. The mobility of the nodes and the small size (in terms of storage space) of the hand-held devices make traditional solutions to network problems such as addressing, routing and flow-control inapplicable.

Recently several authors (see for instance [6, 1, 8]) have developed solutions to some of these problems for *microcellular* or *intra-campus* networks. A microcellular network is one where a geographical area (such as a university campus) is divided into regions called 'microcells'. Each microcell has a diameter of the order of several hundered meters and contains a *Mobile Support Station* (MSS) that provides a connection point for mobile users in that microcell. All the MSSs are connected via a wired network.

In this paper we consider an *in-building* or *picocellular* network and develop a protocol to setup and maintain connections between pairs of mobile users. A picocellular network is similar to a microcellular network with the difference that the cell size is now of the order of 10 meters and is typically a room in a building (see [4, 8]). The advantage of a smaller cell size is higher throughput (because there will be fewer users per cell), greater frequency reuse (i.e., while adjacent cells use different radio frequencies, non-adjacent cells may use the same frequency) and low-power transmitters (resulting in longer battery life). The main drawback is that small cells imply *faster* inter-cell node mobility resulting in a host of routing, tracking and hand-off problems. Figure 1 shows an example of a typical layout of such a network.

The rest of the paper is organized as follows. In the next section we present a brief literature review. In section 2 we present our system model. An overview of the protocol is provided in section 3. In section 4 we present the protocol in detail and flow control issues are discussed in section 5. Our conclusions are presented in Section 6.

1.1 Literature review

Problems associated with microcellular networks have been quite extensively studied in the literature. [6] present a set of IP-based protocols to provide continuous network access to mobile hosts in a microcellular environment. [1] discusses the problem of multicasting in similar environments. [4] introduced the idea of a cellular packet switch to efficiently handle the problem of hand-off in cellular communications.

The solutions presented in these and other papers unfortunately do not scale to our model. The main reason is the small cell size in picocellular networks and high user mobility (in terms of both both speed of mobile users and the number of times mobile users cross cell boundaries). In such situations, the solutions presented in the literature require each MSS to maintain a complicated set of forward pointers along which packets have to be forwarded to the current location of the MH.

2 The Model

The target environment of the proposed protocol is a typical office building. The building is assumed to have a wired backbone network overlaid on top of which is the "picocellular wireless network". A cell in this network is typically restricted to a room and corridors may be made up of several cells. The cell shape may not be uniform and is constrained by the architecture of the building as is movement between cells of mobile users, see Figure 1.

The system can be viewed as a three level hierarchy. At the lowest level is a dense collection of Mobile Hosts (MHs) which communicate with each other and with the stationary hosts in the network as they move around between cells. Communication in each cell is handled by Mobile Support Stations (MSSs). The MSSs are all connected to the underlying wired LAN. Also each subnet of the wired network has an assigned supervisor machine called the Supervisor Host (SH). If the subnet spans a large physical area we may choose to have more than one SH per subnet e.g., if a subnet straddles many floors then we may have one SH per floor.

The MHs are small battery powered handheld devices equipped with a wireless transciever. The MSSs are also small in size and are typically placed on the ceiling of each room. Each MSS contains a large amount of buffer space since the MHs contain very little. The SH is a server machine connected to the wired network and it handles most of the routing and other protocol details.

We assume that every MH has a globally unique address regardless of where it is in the network as in [6]. The SH maintains the current location of every MH within its domain (i.e., its subnet). In addition, the SHs and MSSs together track the MHs as they move between cells. The MSSs are mainly responsible for flow-control and maintaining communication with all the MHs within its cell. In addition, a MSS buffers messages to and from all MHs.

3 Overview of the protocol

The basic concept used in our protocol to maintain a connection with a rapidly moving MH is the idea of a *group*. Let us assume that the MH is in cell c_i at some given time. A collection of cells sorrounding c_i that the MH can move into is defined as the current group of the MH and is denoted by g_i. Any message destined for the MH is multicast to all the cells in g_i. This ensures that even if an MH moves rapidly between cells, messages are not lost and no pointer structures need to be maintained.

The shape and composition of a group is determined by the architecture of the building, the speed at which the MH moves between cells and the direction of motion. Cells that are seperated by a wall will almost never belong to the same group, likewise cells on different floors of the same building will not belong to the same group as in Figure 2. The speed at which an MH moves and the direction of motion have a bearing upon the number of cells that are included in a group. In Figure 3 the arrow represents the direction of motion of MH a and MH b. The shaded cells indicate the composition of their respective current groups.

The SH keeps track of the current group of every active MH within its domain. When a message needs to be deliverd to an MH, the SH multicasts it to the current group of the MH. The group is updated each time the MH moves between cells and all the MSSs that belong to the new group are informed by the SH of the group identity number and the identity of the corresponding MH.

An MSS buffers all messages for an MH that it receives from the SH. If the MH is local, the MSS transmits the messages to the MH. If the MH is not local, the MSS holds on to the messages until either the MH enters its cell or until the SH informs it that it is no longer in the group. In the latter case, all messages are discarded. In the former case, during the initial 'handshake' between the MH and MSS , the MH informs the MSS of the *sequence number* of the next message it expects. The MSS transmits all messages starting with that sequence number. This method ensures that duplicate messages are not received at the MH.

4 Protocol description

The protocol we describe is a connection-oriented protocol. Whenever an MH wants to communicate with another MH, it requests a connection. The SH establishes a *virtual circuit* between the two MHs and returns a virtual circuit number to the MH. Every subsequent packet will contain this virtual circuit number. The protocol requires that the following data structures be maintained at the three levels:

- **MH** : The MH maintains the following information. It keeps the identity of the current MSS and the identity of the current SH. In addition, it maintains a list of its current connections and the attributes of each connection. Each connection has a *virtual circuit* number, sequence number of last packet received and a status flag. The status of a connection can be one of:

  - ACTIVE REMOTE : this means that the connection is active and the other end of the connection is in a different subnet
  - ACTIVE LOCAL : this means that the connection is active and the other end of the connection is in the same subnet
  - NASCENT : this means that the connection is currently in the establishment phase

  If the status of a connection is ACTIVE LOCAL, the group id of the MH at the other end of the circuit will also be recorded. This is because if the two communicating MHs are in the same subnet then the communication will not be through the SH. The MSS will directly multicast the message to the group of the destination MH.

- **MSS :** Each MSS maintains two tables. The first contains "static" information and lists the ID of all the groups it can belong to and the multicast address of each group. The second table contains the ID of all MHs in cells that belong to one of the groups the MSS is in.
- **SH :** At each SH there is a table that maintains a mapping from the current location of a MH (i.e., the MSS) and its direction of motion into a group ID and a group multicast address. The SH also maintains a table of all the virtual circuits that are in operation in its subnet. The table contains the following information for each entry:
  - vc_{local} :the local virtual circuit number
  - vc_{remote} :the remote virtual circuit number of the MH at the other end of the connection. Note that while the virtual circuit number is only locally (i.e., within each subnet) unique, the pair $< SH, vc\# >$ is globally unique.
  - status :the status of the connection which can be one of ACTIVE REMOTE, ACTIVE LOCAL, NACENT or POINT.
    - POINT : this means that the circuit has been adopted by another SH because the MH moved out of the local subnet into the subnet of another SH (SH_{new}) and this SH (SH_{local}) is currently simply forwarding the packets for virtual circuit vc_{local}.
  - SH_{remote} :if status is POINT then this entry is the ID of the SH to which packets with this connection number are to be forwarded.If status is ACTIVE REMOTE then it is the ID of the SH at the other end of the connection
  - g_{local} :the group of the MH which is the local end the connection

  Finally, each SH also keeps a table containing the mapping of the group IDs to the IDs of the member MSS's of the group.

4.1 Connection Establishment Phase

We present the protocol followed to establish a connection between two mobile hosts. The steps are:

1. The MH (MH_s) wishing to initiate communication with another MH (MH_d), informs its MSS(MSS_s) of its intentions which in turn conveys this request to the SH (SH_s). The request includes the ID of the MH with which a connection is desired and the group id of MH_s.
2. SH_s assigns a virtual circuit number (the smallest number not currently in use) and this number is conveyed to MH_s. MH_s includes this circuit number in its list of connections. The status assigned to the connection at MH_s is NASCENT. MH_s sends an ACK to SH_s.
3. On receiving the ACK, the SH_s places the virtual circuit in its table of connections. The status is NASCENT and the group of MH_s is set from the latest available information. If no ACK is received before a TIMEOUT interval ends SH_s gives up trying to establish the connection and terminates the protocol.
4. SH_s tries to locate MH_d. If MH_d is local, it broadcasts a LOCATE packet in its own subnet containing the virtual circuit number and the ID of MH_s. MH_d includes this virtual circuit into its list of connections, sets the status of the circuit to ACTIVE LOCAL and responds with an ACK.
5. If MH_d is remote, SH_s broadcasts the LOCATE request to all other SHs. The LOCATE packet contains the virtual circuit number assigned at the initiating end and the ID of SH_s.
6. Upon receiving a LOCATE from another SH, the SH attempts to locate the MH_d in its subnet. The SH that does find it (SH_d) then tries to establish the other end of the connection. It assigns the lowest available virtual circuit number to this connection and sends a packet to MH_d containing this virtual circuit number. MH_d responds with an ACK and adds the virtual circuit number to its list of connections with the status set to ACTIVE REMOTE.
7. SH_d on receiving the ACK then includes this circuit number in it's table of connections and adds the remote virtual circuit number and the id of the remote SH (SH_s) to the entry. The status assigned to the connection is ACTIVE REMOTE. SH_d responds to SH_s with an ACK. This notification is accompanied by the virtual circuit number assigned at it's end and SH_d's id.
8. On receiving the ACK from SH_d, SH_s changes the status of the connection to ACTIVE REMOTE and informs MH_s which also changes the status of the connection to ACTIVE REMOTE.

This procedure completes the establishment of the connection. It is worth mentioning that the reason for requiring an ACK for each connection notification to an MH is to account for a possible movement of the MH into another subnet.

4.2 Communication between two MHs

Once the connection has been established the MHs at the two ends of the connection can communicate. We describe the journey of a packet as it travels from MH_i to MH_j. There are two distinct cases to be considered:

MH_i and MH_j are in the same subnet. In this case MH_i will give the packet to be sent to the MSS (MSS_i) of the current cell. The packet is labelled with the virtual circuit number and the current group of MH_j as known by MH_i (g_j). MSS_i includes the current group of MH_i (g_i) and multicast it to the group g_j. Recall that since the status of the connection would be ACTIVE LOCAL, MH_i will have the current group information of MH_j. The multicast packet will reach MH_j and it will use the value of g_i to update its information about the current group of MH_i and vice versa. It is important for MH_i and MH_j to exchange packets quite frequently to keep each other informed (perhaps by exchanging NULL packets) about their latest group IDs.

MH_i and MH_j are in different subnets. MH_i gives the packet to MSS_i . The packet contains the virtual circuit number (vc_i). MSS_i sends it to SH_i. SH_i looks up the entry corresponding to the virtual circuit number vc_i and finds the values for the remote virtual circuit number (vc_j) and the ID of the SH at the other end of the connection (SH_j). SH_i now sends the packet labelled with vc_j to SH_j. SH_j on receiving the packet labelled with vc_j looks up the entry corresponding to this connection and finds out the status of the connection. The status will be either ACTIVE REMOTE or POINT. If the status is POINT then SH_j is simply serving to forward packets for this connection. SH_j looks up the entry SH_{remote} in the table and also vc_{remote}. It forwards the packet to SH_{remote} and labels the packet with the virtual circuit number vc_{remote}. If however the status of the circuit is ACTIVE REMOTE then SH_j will multicast the packet to the group g_{local}.

5 Flow Control and Buffer Management

The problem of flow-control and buffer management is quite interesting in our system because of the three-tiered architecture used. We use one flow-control mechanism for communication between an MH and an MSS, and a second, similar, flow-control mechanism for communication between an MSS and its SH. In order to explain these flow-control schemes better, we first need to discuss the buffer management protocol. Let us focus on a specific MH, m, that has k open connections and an MSS, M.

Buffer allocation at M: Let us assume that initially M does not lie in m's current groups but after some time, m moves close enough that M is included in a current group. Let us further assume that m *does not* lie in the cell belonging to MSS M.

When M is first included in m's group, it allocates b buffers for m. b is smaller than some maximum. If there are no free buffers available at M (i.e., M is already included in several groups), it simply deallocates *one* buffer from some other MHs allocation and uses it for m.

The b buffers are treated as a FIFO queue with new packets (for m) put in at the tail of the queue. If M receives more than b packets for m, it retains only the latest b packets (i.e., once all b buffers are full, packets are dropped from the head of the buffer queue to make room for more recent packets), discarding duplicates.

Flow-Control at M: While m is not in M's cell, M does not communicate with m and therefore does not participate in any type of flow-control for m. Let us assume that m eventually moves into M's cell.

During the initial handshake, m sends a packet to M containing the sequence number of packets m expects to receive for *each* of its k open connections. In addition, for every open connection m also includes a *permit*. A permit for a connection is a number representing the number of packets that m can buffer locally for that connection.

Upon receiving the above information, M begins transmitting packets to m starting with the appropriate sequence numbers for each connection and upto the maximum allowed by the permit for that connection. The order in which the connections are serviced may depend upon some priority mechanism (e.g., real-time video is transmitted before a data file). In this paper, however, we are not concerned with this aspect of the problem. If m stays in Ms cell long enough, it transmits packets with sequence number and permit information as above whenever its local buffers become free.

Periodically (while m is in M's cell) M transmits a message to the SH containing a sequence number for each of m's connections along with a permit for each connection (the value of this permit depends upon b and the number of packets of each connection that are in M's buffer). Thus for some connection i if M has packets numbered upto $j + l$ in its buffers but if the latest packet transmitted to m is numbered j then the sequence number for this connection sent to the SH is $j + 1$ (not $j + l$). On receiving this message the SH *discards* all packets for this connection (for m) numbered upto j because m has successfully received them. Note that if M had instaed transmitted $j + l$ to SH then it is possible that m may not receive all packets. For example, if m moved out of Ms cell into a neighboring cell which does not have the packets numbered between $j + 1$ and $j + l$ (because of little available buffer space) then m never receives these packets since the SH has already discarded them.

Upon receiving the above packet from M, the SH begin multicasting packets for each connection starting from the given sequence number and upto a maximum sequence number as determined by the value of the permits.

6 Conclusions and Future Work

In this paper we have presented a novel way to implement efficient seamless communication between mobile users in a picocellular environment using the idea of groups. It will be interesting to quantitatively compare the efficiency of using pointers (as in most other work) and the idea of using groups as in this paper. It will be useful to study different buffering mechanisms at MSSs and develop mathematical or simulation models to study their behavior. The authors are currently looking at some of these questions.

References

[1] A. Acharya and R. Badrinath, "Delivering multicast messages in networks with mobile hosts" *13th Intl. Conf. on Distributed Computing Systems*, May 1993.

[2] Dale Buchholz,Paul Odlyzko,Mark Taylor and Richard White, "Wireless In-Building Network Architecture and Protocols" *IEEE Network Magazine*, pp 31-38, November 1991.

[3] D. Duchamp, Steven K. Feiner and G. Q. Maguire, "Software technology for wireless Mobile computing" *IEEE Network Mag.*, pp 12-18, November 1991.

[4] David J. Goodman, "Cellular Packet Communications" *IEEE Trans. on Comm.*, vol. 38, no. 8, pp 1272-1280, August 1990.

[5] David J. Goodman, "Trends in Cellular and Cordless Communications" *IEEE Communications Magazine*, pp 31-40, June 1991.

[6] J. Ioanidis, D. Duchamp and G. Q. Maguire, "IP-based protocols for mobile internetworking" *Proc. of ACM SIGCOMM'91*, pp 235-245, September 1991.

[7] Fumino Teraoka, Yasuhiko Yokote and Mario Tokoro, "A network architecture providing host migration transparency" *Proc. of ACM SIGCOMM'91*, pp 209-220, September 1991.

[8] R. Steele, "The cellular environment of lightweight handheld portables" *IEEE Commun. Mag.*, vol 27, no. 7, pp 20-29, July 1989.