Network Working Group
Request for Comments: 2009
Category: Experimental

T. Imielinski
J. Navas
Rutgers University

November 1996

GPS-Based Addressing and Routing

Status of this Memo

IANA Note:

This document describes a possible experiment with geographic addresses. It uses several specific IP addresses and domain names in the discussion as concrete examples to aid in understanding the concepts. Please note that these addresses and names are not registered, assigned, allocated, or delegated to the use suggested here.

Table of Contents

1. Introduction

In the near future GPS will be widely used allowing a broad variety of location dependent services such as direction giving, navigation, etc. In this document we propose a family of protocols and addressing methods to integrate GPS into the Internet Protocol to enable the creation of location dependent services such as:

- Multicasting selectively only to specific geographical regions defined by latitude and longitude. For example, sending an emergency message to everyone who is currently in a specific area, such as a building or train station.

- Providing a given service only to clients who are within a certain geographic range from the server (which may be mobile itself), say within 2 miles.

- Advertising a given service in a range restricted way, say, within 2 miles from the server,

- Providing contiguous information services for mobile users when information depends on the user's location. In particular providing location dependent book-marks, which provides the user with any important information which happens to be local (within a certain range) possibly including other mobile servers.

The solutions which we present are flexible (scalable) in terms of the target accuracy of the GPS. We also discuss cases when GPS cannot be used (like inside buildings).

The main challenge is to integrate the concept of physical location into the current design of the Internet which relies on logical addressing. We see the following general families of solutions:

a) Unicast IP routing extended to deal with GPS addresses

b) GPS-Multicast solution

c) Application Layer Solution using extended DNS

The first two solutions are presented in this memo. We only sketch the third solution.

1b. General Architecture

We will assume a general cellular architecture with base stations called Mobile Support Stations (MSS). We will consider a wide variety of cells, including outdoor and indoor cells. We will discuss both cases when the mobile client has a GPS card on his machine and cases when the GPS card does not work (i.e. - inside buildings).

We will assume that each MSS covers a cell with a well defined range specified as a polygon of spatial coordinates and that the MSS is aware of its own range.

1c. Scenarios of Usage and Interface Issues

Below, we list some possible scenarios of usage for the geographic messaging.

Consider an example situation, of an area of land near a river. During a severe rain storm, the local authorities may wish to send a flood warning to all people living within a hundred meters of the river.

For the interface to such messaging system we propose to use a zoom-able map similar to the U.S. Census Bureau's Tiger Map Service. This map would allow a user to view a geographical area at varying degrees of magnitude. He could then use a pointing device, such as a mouse, to draw a bounding polygon around the area which will receive the message to be sent. The computer would then translate the drawn polygon into GPS coordinates and use those coordinates when sending and routing the message. Geographical regions specified using this zoom-able map could be stored and recalled at a later time. This zoom-able map is analogous to the IP address books found in many email programs.

To continue with the above example, local officials would call up a map containing the river in danger of overflowing. They would then hand-draw a bounding polygon around all of the areas at least a hundred yards from the river. They would specify this to be the destination for a flood warning email to all residents in the area. The warning email would then be sent. Similar applications include traffic management (for example, reaching vehicles which are stuck in traffic) and security enforcement.

Other applications involve general client server applications where servers are selected on the basis of the geographic distance. For example, one may be interested in finding out all car dealers within 2 miles from his/her location. This leads to an extension of the Web concept in which location and distance play important roles in selecting information. We are currently in the process of implementing location dependent book-marks (hot lists) in which pages associated with static and mobile servers which are present within a certain distance from the client are displayed on the client's terminal.

2. Addressing Model

Two-dimensional GPS positioning offers latitude and longitude information as a four dimensional vector:

<Direction, hours, minutes, seconds>

where Direction is one of the four basic values: N, S, W, E; hours ranges from 0 to 180 (for latitude) and 0 to 90 for longitude, and, finally, minutes and seconds range from 0 to 60.

Thus <W, 122, 56, 89> is an example of longitude and <N, 85, 66, 43> is an example of latitude.

Four bytes of addressing space (one byte for each of the four dimensions) are necessary to store latitude and four bytes are also sufficient to store longitude. Thus eight bytes total are necessary to address the whole surface of earth with precision down to 0.1 mile! Notice that if we desired precision down to 0.001 mile (1.8 meters) then we would need just five bytes for each component, or ten bytes together for the full address (as military versions provide).

The future version of IP (IP v6) will certainly have a sufficient number of bits in its addressing space to provide an address for even smaller GPS addressable units. In this proposal, however, we assume the current version of IP (IP v4) and we make sure that we manage the addressing space more economically than that. We will call the smallest GPS addressable unit a GPS-square.

2a. Using GPS for Destination Addresses

A destination GPS address would be represented by one of the following:

- Some closed polygon such as:

  circle(center point, radius)

  polygon(point1, point2, point3, ... , pointn)

  where each point would be expressed using GPS-square addresses. This notation would send a message to anyone within the specified geographical area defined by the closed polygon.

- site-name as a geographic access path

  This notation would simulate the postal mail service. In this manner, a message can be sent to a specific site by specifying its location in terms of real-world names such as the name of a specific site, city, township, county, state, etc. This format would make use of the directory service detailed later.

For example, if we were to send a message to city hall in Fresno, California, we could send it by specifying either a bounding polygon or the mail address. If we specify a bounding polygon, then we could specify the GPS limits of the city hall as a series of connected lines that form a closed polygon surrounding it. Since we have a list of connected lines, we just have to record the endpoints of the lines. Therefore the address of the city hall in Fresno could look like:

polygon([N 45 58 23, W 34 56 12], [N 23 45 56, W 12 23 34], ...)

Alternatively, since city hall in Fresno is a well-defined geographical area, it would be simpler to merely name the destination. This would be done by specifying "postal-like" address such as city_hall.Fresno.California.USA.

For "ad hoc" specified areas such as, say a quad between 5th and 6th Avenue and 43 and 46 street in New York, the polygon addressing will be used.

Unfortunately, we will not be able to assume that we have enough addressing space available in the IP packet addressing space to address all GPS squares. Instead we will propose a solution which is flexible in terms of the smallest GPS addressable units which we call atoms. In our solution, a smaller available addressing space (in the IP packet) will translate into bigger atoms. Obviously, we can use as precise addressing as we want to in the body of the geographic messages - the space limitations apply only to the IP addressing space.

By a geographic address we mean an IP address assigned to a geographic area or point of interest. Our solution will be flexible in terms of the geographic addressing space.

Below, we will use the following two terms:

- Atoms: for smallest geographic areas which have geographic address.

  Thus, atoms could be as small as GPS squares but could be larger

- Partitions: These are larger, geographical areas, which will also have a geographic address. A state, county, town etc. may constitute a partition. A partition will contain a number of atoms.

Here are some examples of possible atoms and partitions:

- A rectangle, defined by truncating either longitude or latitude part of the GPS address by skipping one or more least significant digits
- A circle, centered in a specific GPS address with a prespecified radius.
- Irregular shapes such as administrative domains: states, counties, townships, boroughs, cities etc

Partitions and Atoms (which are of course special atomic partitions) will therefore have geographic addresses which will be used by routers. Areas of smaller size than atoms, or of "irregular shape" will not have corresponding geographic addresses and will have to handled with the help of application layer.

3. Routing

Let us now describe the suggested routing schemes responsible for delivering a message to any geographical destination.

We will distinguish between two legs of the connection from the sender to the receiver: the first leg from the sender to the MSS (base station) and the second leg from the MSS to the receiver residing in its cell. Our two solutions will differ on the first leg of the connection and use the same options for the second leg, which we call "last mile".

3a. GPS-Multicast Routing Scheme

Here, we discuss the first leg of routing: from the sender to the MSS. We start with the multicasting solution.

Each partition and atom is mapped to a multicast address. The exact form of this mapping is discussed further in this subsection. We first sketch the basic idea.

This solution provides flexible mix of the multicast and application level filtering for the geographic addressing. The key idea here is to approximate the addressing polygon of the smallest partition which contains it and using the multicast address corresponding to that partition as the IP address of that message. The original polygon is a part of the packet's body and the exact matching is done on the application layer in the second leg of the route.

How is the multicast routing performed?

3a-i. Multicast Trees

The basic idea for the first level of routing using multicast is to have each base station join multicast groups for all partitions which intersect its range. Thus, MSS is not only aware of its own range but also has a complete information about system defined partitions which its range intersects. This information can be obtained upon MSS installation, from the geographic database stored as a part of DNS.

If the proper multicast trees are constructed (using for example link state multicast protocol) than the sender can simply determine the multicast address of the partition which covers the original polygon he wants to send his message to, use this multicast address as the address on the packet and put the original polygon specification into the packet content. In this way, multicast will assure that the packet will be delivered to the proper MSS.

Example

For instance the MSS in New Brunswick may have its range intersect the following atoms and partitions: Busch, College Avenue, Douglass and Livingston Campuses of Rutgers University (atoms), New Brunswick downtown area (atom), the Middlesex county partition and the NJ state partition. Each of these atoms and partitions will be mapped into a multicast address and the New Brunswick's MSS will have to join all such multicast groups.

The message will be then specified and sent as follows:

The user will obtain the map of the New Brunswick area possibly from the DNS extended properly with relevant maps. He will specify the intended destination by drawing a polygon on the map which will be translated into the sequence of coordinates. In the same time the polygon will be "approximated" by the smallest partition which contains that polygon. The multicast address corresponding to that partition will be the IP address for packets carrying our message. The exact destination polygon will be a part of each packet's body. In this way the packet will be delivered using multicast routing to

the set of MSS which are members of the specified multicast group (that is all MSS whose ranges intersect the given partition). Each such MSS now will follow the "last mile" routing which is described in detail, further in the proposal. Briefly speaking, the MSS could then multicast the message further on the same multicast address and the client will perform the final filtering o application layer, matching its location (obtained from GPS) with the polygon specified in the packet's body. Other solutions based entirely on multicasting are also possible as described below.

End_Example

However, things cannot be as simple as described. For such a large potential number of multicast groups if we build entire multicast trees, the routing tables could be too large. Fortunately it is not necessary to build complete multicast trees. Indeed, it in not important to know precise location of each atom in California, from a remote location, say in NJ.

Thus, we modify our simple solution by implementing the following intuition:

The smaller is the size of the partition (atom) the more locally is the information about that partition (atom) propagated.

Thus, only multicast group membership for very large partitions will be propagated across the whole country.

For example, a base station in Menlo Park, California can intersect several atoms) and several larger which cover Menlo Park, such say a partition which covers the entire San Mateo county, next which cover the entire California and finally next which may cover the entire west coast. This base station will have to join multicast groups which correspond to all these rectangles. However, only the information about multicast group corresponding to the West Coast partition will be propagated to the East Coast routers.

However, a simple address aggregation scheme in which only a "more significant portion" of address propagates far away would not work. Indeed, in this case a remote router, say in NJ, could have several aggregate links leading to California - in fact, in the worst case, all its links could point to California since it could have received a routing information to some location in California on any of those links.

To avoid this, for each partition we distinguish one or a few MSS which act as designated router(s) for that partition. For example, the California partition, may have only three designated routers, one

in Eureka, another in Sacramento and yet another in LA. Only the routing entries from the designated routers would be aggregated into the aggregate address for California. Information coming from other city routers will simply be dropped and not aggregated at all. This, in addition to a standard selection of the shortest routes, would restrict the number of links which lead to an aggregate address. In particular, when there is only one designated router per partition, there would only be one aggregate link in any router. This could lead to non-optimal routing but will solve the problem of redundant links.

Even with a designated routers, it may happen that the same packet will arrive at a given base station more than once due to different alternative routes. Thus, a proper mechanism for discarding redundant copies of the same packet should still be in place. In fact, due to the possible intersections between ranges of the base stations the possibility of receiving redundant copies of the same packets always exist and has to be dealt with as a part of any solution.

3a-ii. Determining the geographic Multicast Addressing

Here we describe more specifically, the proposed addressing scheme and the corresponding routing.

The addressing will be hierarchical. We will use the following convention - each multicast address corresponding to a partitions or an atoms will have the following format:

1111.GPS.S.C.x

where GPS is the specific code corresponding to the geographic addressing subspace of the overall multicast addressing space. The S, C and x parts are described below:

S - Encoding of the state.
Each state partition will have the address S/0/0.

C - County within a state.
Each county partition having the address S/C/0.

x - Atom within a county.

where 0's refer to the sequences of 0 bits on positions corresponding to the "C part" and "x part" of address.

For example if GPS part is 6 bit,s which gives 1/64 of existing multicast addresses to the geographic addressing we have 22 bits left. The S part will take first 6 bits, C part next 6 bits (say) and then the next 10 bits encode different atoms (within a county).

Thus, in our terminology the proposed addressing scheme has two types of partitions: states and counties.

We will assume that the GPS network will consist of all base stations (MSS) in addition the rest of the fixed network infrastructure. The designated GPS routers however, will only be selected from the population of MSS. Specifically, there will be state dedicated and county dedicated routers.

The concept of the designation will be implemented as follows. From the set of all MSS, only certain MSS will play a role of designated routers for county and state partitions. Non-designated MSS will only join multicast groups which correspond to the GPS atoms but not GPS partitions that they intersect. The MSS which is a designated router for a county partition will join the multicast group of the county in which it is located, but not the state. Finally the state designated router will also join the multicast address corresponding to the state it is located in.

3a-iii. Building Multicast Trees

We assume that each router has geographic information attached to it - in the same format as we use for multicast mapping, S/C/x - it encodes the atom that contains the router.

The multicast tree is built by a router propagating its multicast memberships to the neighboring routers. A given router will only retain certain addresses though, to follow the intuition of not retaining a specific information which is far away.

This is done as follows: the router (not necessarily the MSS based router) with the address S/C/x will only retain addresses about S'/0/0, S/C'/0 for S' and C' different from S and C and S/C/x for all x. Thus, it will drop all the addresses of the form S'/C'/y for all S' different that S except those with C'=0 and y=0, as well as all the addresses of the form S/C'/y with C' different from C except those with y=0. Hence, these addresses will not be forwarded any further either.

Thus, notice that only the information coming from designated routers will be forwarded further away, since the non-designated routers are not allowed to join the multicast groups which correspond to the states and counties. Consequently, their multicast membership information will be not be propagated.

In this way a router at S/C/x will not bother about specific locations within S'/C'/y since they are "too far".

Notice that this service may not be provided everywhere so we may not have to use all multicast addresses even within those assigned for geographic addresses.

Notice also that all of this is flexible - if we have more multicast addresses available (IP v 6) we will get more precise addressing due to smaller atoms.

3a-iv. GPS Routing

Given a packet we always look for the "closest" match in the routing table. If there is a complete match we follow such a link, if not we follow the address with the x-part 0'd in (county address) if there is none with the county which agrees with the destination county than we look at the entry which agrees with the state part of the destination address.

3a-v. DNS Issues

How does the client find out the multicast address on which the packet is to be sent? We assume that the local name server has the complete state/county hierarchy and that each county map can be provided possibly with the "grid" of atoms and partitions already clearly marked.

Points of interests within a county can be attached multicast address just as atoms. Then a given base station would have to join multicast groups of the points of interests that it covers.

The final stage is for the receiver to look at the polygon (point of interest) which is encoded in the body of the multicast packet and decide on the basis of its own GPS location if this packet is to be received or not. Doing it on the application layer simplifies many routing issues. There is a tradeoff, however, specially when we have very short S/C/x addresses and base stations which do not cover the given polygon in fact are reached unnecessarily. This may happen and it needs to be determined what is the number of the multicast addresses which are necessary to reduce this "false" alarms to the minimum.

3a-vi. Estimations

Assume average cell size of, say, 2km x 2km and the average state size: say 200,000 square km, the average county size: say 4,000 square km.

A reasonable size of the atom is around the size of the cell since then we do not hit wrong cells too often.

Therefore we need the x addressing part of the S/C/x to encode 4,000/4 cells: 1.000 atoms. Thus we need 10 bits for x part. With 6 bits for the state and 6 bits for the county that gives 22 bits which is 1/64 of the total IP v4 multicast addressing space.

With IPv6 we will have, of course, much more addressing space which we can use for the GPS multicast routing.

3b. "Last Mile" Routing

Multicasting will be used for the last mile routing in both our solutions (i.e the one just discussed and the geometric routing solution described next), but in different ways.

3b-i. Application Level Filtering

The MSS will forward the geographic message on its wireless link under a multicast address. This multicast address will either be the same for all locations in the range of the MSS's cell or, there will be several addresses corresponding to atoms which intersect the given cell. Additionally, a complete GPS address (for example in the form of the polygon) will be provided in the body of the packet and the exact address matching will be performed on the application layer. The receiver, knowing its GPS position uses it to match against the polygon address. The GPS position can be obtained by the receiver either from the GPS card or, indoors, from the indoor base station which itself knows its GPS position as a part of configuration file.

3b-ii. Multicast Filtering

In multicast level filtering, the base station assigns a temporary multicast address to the addressing polygon in a message. It will send out a directive on the cell's specially assigned multicast address. All mobile clients who reside in that cell are members of that special multicast group (one per MSS). The directive sent by the MSS will contain the pair consisting of the temporary multicast address together with the polygon. To improve the reliability this message will be multicast several times. The clients, knowing their GPS positions will than join the temporary multicast groups if their current locations are within the advertised polygon. The MSS will then send out the real message using the temporary multicast address.

The temporary multicast address would be cached for a period of time. If more packets for the same polygon arrive in a short period of time, they will be sent out on the same multicast address. If not, then the multicast address is dropped and purged from the cache. Filtering on the client's station is then performed entirely on the IP level. This solution introduces additional delay (needed to join

the temporary multicast group) but reduces the number of irrelevant packets received by the client. This especially important for very long messages.

3b-iii. Computers on Fixed Networks

Fixed-network computers should also monitor all of the mandatory multicast addresses for their site and GPS square. In this manner, the fixed computers will also receive messages sent to specific GPS-addresses.

Modified base stations would still be in charge of multicasting the messages to the computers. These base stations would have the same GPS-routing functionality as the mobile computer base stations. Their main difference would be that the mobile computer base stations would use radio frequencies to multicast their messages and the fixed network base stations use the local Ethernet or Token Ring network.

The next scheme differs from the GPS multicast scheme described above only on the first leg of the route, from the sender to the MSS. The "last mile" from the MSS to the final destination will have the same options as described above.

3c. Geometric Routing Scheme (GEO)

The Geometric Routing Scheme (GEO) uses the polygonal geographic destination information in the GPScast header directly for routing. GEO routing is going to be implemented in the Internet Protocol (IP) Network layer in a manner similar to the way multicast routing was first implemented. That is, a virtual network which uses GPS addresses for routing will be overlayed onto the current IP internetwork. We would accomplish this by creating our own GPS-address routers. These routers would use tunnels to ship data packets between them and between the routers and base stations.

3c-i. Routing Overview

Sending a GPScast message involves three steps: sending the message, shuttling the message between routers, and receiving the message.

Sending a GPScast message is very similar to sending a UDP datagram. The programmer would use the GPScast library routine SendToGPS(). Among other parameters, this routine will accept the GPS polygonal destination address and the body of the message. The SendToGPS() routine will encapsulate the GPScast message in a UDP datagram and send it to the class E address 240.0.0.0. Previously, the system administrator will have specified in the /etc/rc.local or /etc/rc.ip file a route command that will specify that packets with the address

240.0.0.0 will instead be sent to the address of the local GPS router. This will have the effect of sending the datagram to the nearest GPS router.

Before explaining how the GPS routers shuttle the GPScast message to its destination, an introduction to routers and their different parts is in order. For scalability purposes, GPS routers are arranged in a hierarchical fashion. Each layer would correspond to a distinct geographic area, such as a state or a city. At the top would be country-wide routers in charge of moving messages from one end of the country to another. At the bottom would be campus or department routers in charge of moving messages between the base stations. See Figure 1.

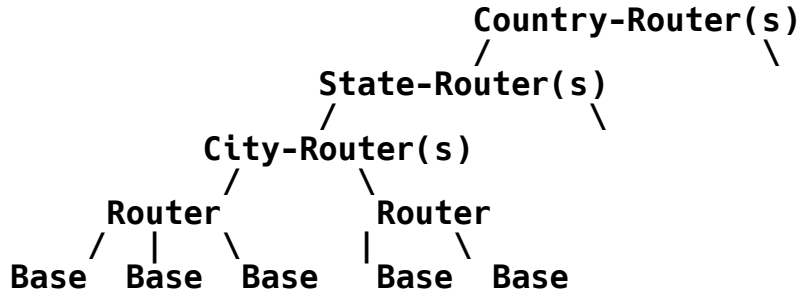


Figure 1: Hierarchy of routers.

A GPS router essentially consists of three parts: a service area table containing the geographic area serviced by the router and each of its hierarchical children, a hashed cache of previous actions, and a table containing the IP addresses of at least the router's children and the router's parent. In the case of a bottom-layer campus router, the service area table will contain polygons describing the geographic reach of each child base station's cell. The polygon created from the union of all of the router's child base stations' polygons defines the service area of the router.

Once the datagram arrives at a GPS router, the router strips the datagram off, thereby, leaving it with the original GPScast message. First the router must determine if it services any part of the area of the destination polygon. To do this, the router finds the intersection between the destination polygon and the polygon describing the router's service area. The polygon intersection algorithm used is described by O'Rourke in his paper, A New Linear Algorithm for Intersecting Convex Polygons. This algorithm requires order N-squared time in the worst case. If the intersection result is null, then the router simply sends the message to its parent router.

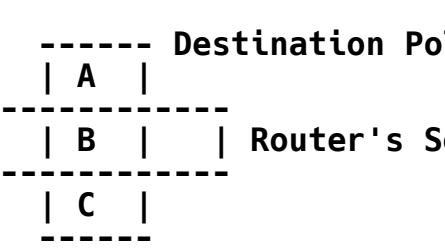


Figure 2: Polygon Difference

However, if the result is not null, then the router does service the area described by the intersection polygon. The router now subtracts its service area from the destination polygon and sends the rest to it's parent router. This subtraction step is actually a by-product of the intersection algorithm. Using the example in Figure 2, the destination polygon and the router's service area polygon intersect at the region labeled B. Therefore, the router will subtract out the B section and send the remaining sections A and C to its parent router.

Continuing with the example, the router now uses the intersection polygon B to to determine which base station (or stations) will receive the GPScast message. The router finds the intersection between the region B and the polygon of each base station's cell. Those base station polygons which intersect the region B will be sent the GPScast message. Processes on Mobile Hosts serviced by these base stations will now use the routine RecvFromGPS() to receive the GPScast message.

3c-ii. Supporting Long-Duration GPScasts

Most likely, there will be a need to support sending real-time continuous media to a GPS destination. This continuous media could be an audio GPScast or a video GPScast. This would require that jitter be reduced in order to minimize disturbing artifacts in the audio or video playback. Continually checking the destination geometry of each packet would incur unnecessary delays and may promote jitter.

Therefore, the router will keep a hashed cache of the latest GPScast packets and their destinations. Each cache item will be hashed using the Sender Identification included in the header of GPScast messages as the key. Each cache item will contain a time stamp and a list of the next hops for that GPScast. When the time stamp exceeds a certain limit, then the cache item will be dropped. The list of next hops is a list of the IP addresses of the base stations, peer routers, and parent router which are to receive a copy of the GPScast messages.

When a router receives a GPScast packet, it will use the incoming packet's Sender Id as a key into the hashed cache. If this is not the first packet to arrive for this destination and if the timer on the hash table entry has not yet expired, then the hashed cache will return a list of all of the destination addresses to which copies of the packet must be sent. Copies of the packet are sent to all of these destinations and the hash entry's time stamp is updated.

If no hash table entry is found (i.e.- this is the first packet encountered for this destination address), then the normal geometry checking routine would take over. A new cache entry is made recording all of the next-hop destination addresses of the GPScast. In this manner, if several other packets with the same GPS destination follow this first packet, the router can use the hash table to look-up the destination base stations instead of calculating it using geometry.

3c-iii. Discovering A Router's Service Area

When the router is initiated, it will consult its configuration file. One of the items it will find in the file will be the multicast address of the base station group to which all of its child base stations are members. The router will join this group and then send out Service Area Query messages to this multicast group periodically to discover and to refresh its knowledge of its children base stations and the geographical areas serviced by them.

Queries are issued infrequently (no more than once every five minutes) so as to keep the IGPSMP overhead on the network very low. However, since the query is issued using unreliable multicast datagrams, there is a chance that some base stations may not receive the query. This is important in two cases: when a child node fails and when a router first boots up. The case of a failed child node will be explained later. However, when a router first boots up, it can issue several queries in a small amount of time in order to guarantee that base stations will receive the query and to, therefore, build up its knowledge about its child base stations quickly.

Base stations respond to a Service Area Query by issuing a Service Area Report. This report is issued on the same multicast group address that all of the base stations have joined. The report contains the geographical service area of the base station. In order to avoid a sudden congestion of reports being sent at the same time, each base station will initiate a random delay timer. Only when the timer expires will the base station send its report.

For every base station that responds, the router will create an IP tunnel between it and the base station. This tunnel will carry the GPScast packet traffic between the base station and the router. Each responding base station and its geographic area of service will also be included in the router's geometric routing table as a possible destination for GPScast packets. Any base station that does not respond for ten continuous Service Area Queries will be considered unreachable and will be dropped from the routing table.

3c-iv. Hierarchical Router Structure and Multicast Groups

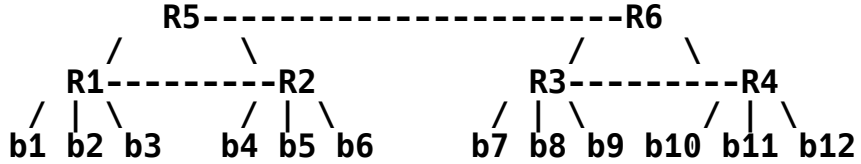


Figure 3: Two peer routers (R5 and R6) cooperatively servicing four child routers (R1 - R4).

For scalability purposes, a hierarchy of routers is used to transport messages from a sender to a receiver. Each layer of peer routers would have its own multicast group address for the exchange of Service Area Queries and Reports between the peer routers. However, routers in distinct subtrees need not know about the routers in other subtrees. Therefore, multicast group addresses will also differ between hierarchy subtrees. See figure 3. For instance, routers R1 and R2 would share a multicast group and would know about each other. At the same time, routers R3 and R4 would share a different multicast group and would know about each other. However, routers R1 and R2 would not know about R3 and R4, and vice versa.

But how will the router know the location and number of its peer routers and who its parent router is? As mentioned before, the router consults its configuration file upon start-up. Included in this configuration file will be the the address of its parent router and the multicast group address that the peer routers will use. This peer multicast group address will be used in the same manner as the base station multicast group address. It will be used to send and receive Service Area Queries and Reports between the parent router and the peer routers. There is only one difference. When a router sends a Service Area Report, in addition to reporting its geographical service area, a router will include the multicast address of its children base stations. The reason for this is explained in the router-failure recovery scheme described below.

3c-v. Routing Optimizations

The optimization described here attempts to reduce the latency of a GPScast. It does so by reducing the the number of hops a packet must traverse before finding its destination. The intuition behind the idea is this: instead of going to the parent router and then to the sibling, simply go to the sibling directly. As an additional benefit, this method prevents the parent router from becoming a bottleneck or a point of failure in the routing scheme.

In this optimization, when a router attempts to determine who will receive the GPS packet, it considers its peer routers as if they were also its children in the routing hierarchy. This means that the router will consider its service area to be the union of the service areas of its children and its peer routers. Also, when the destination polygon intersects the router's service area polygon, the router will forward a copy of the GPScast packet to any child or peer router whose geographic service area contains or touches the packet's GPS destination polygon.

However, before it sends a copy of the packet to a peer router, it first finds the polygon:

$$P = D \wedge S$$

where D stands for the packet's destination GPS polygon, S is the polygon representing the service area of the peer router, and P is the polygon that represents the intersection of D and S. The polygon P is substituted for the destination polygon D in the packet and only then is the packet forwarded to the peer router. This is necessary because the peer router will be using that same routing algorithm. Therefore, if the peer router receives a packet with the original destination polygon D, it will also route copies of the packet to all of its qualifying peer routers causing a chain of packet copies being bounced back and forth.

3c-vi. Router-Failure Recovery Scheme

In the case of a router failure, the system should be able to route around the failed router and continue to service GPScast messages. The responsibility of detecting whether a router has failed or not falls to the parent router. Using Figure 3 as an example router hierarchy, the parent router R5 periodically sends out Service Area Query IGPSMP messages on its children's multicast group address. Thus, the child routers R1 and R2 will both receive this query. Normally, both routers will respond with a Service Area Report message. This message contains a polygon describing their service areas and the multicast group address of their children.

However, if a router, R1, does not respond to ten continuous queries, then it must be considered to have failed. Upon detecting this, the parent router R5 will send a Set Service Area message to the child router, R2 telling it to assume responsibility for the base stations underneath the failed R1 router. In this Set Service Area message, the parent router includes the multicast group address of R1's children. The R2 router uses this multicast address to learn the service areas and IP addresses of R1's children. The R2 router then issues a Service Area Report advertising its new enlarged service area responsibilities. All peer and parent routers will then update their routing tables to include this new information. When the failed router, R1, restarts, it will declare that it is alive and that it is again servicing its area. All routers will then again update their routing tables.

In the case that there is no parent router, such as at the top of the routing hierarchy, then each peer router will keep track of its neighbors. If a neighbor router fails, then the first neighbor router to declare that it is taking over the base stations for the failed router will take responsibility. The rest continues as before.

3c-vii. Domain Name Service Issues

Domain Name Servers (DNS) could be used to facilitate the use of GPS geographic addressing for sites of interest. The aim is to describe specific geographic sites in a more natural and real-world manner using a postal-service like addressing method. Essentially, the DNS would resolve a postal-service like address, such as City_Hall.New_York_City.New_York, into the IP address of the GPS router responsible for that site. The GPS router would then route the message to all available recipients in the site.

The DNS would be used when a message is sent using the

site-code.city-code.state-code.country-code

addressing scheme. The DNS would evaluate the address in reverse starting with the country code, then the state code, etc. This is the same method used currently by the IP DNS service to return IP addresses based on the country or geographic domains.

4. Router Daemon and Host Library

4a. GPS Address Library - SendToGPS()

A library for GPS address routing will be constructed. The main routines contained in this library will be the SendToGPS() and RecvFromGPS() commands. SendToGPS() has the following syntax:

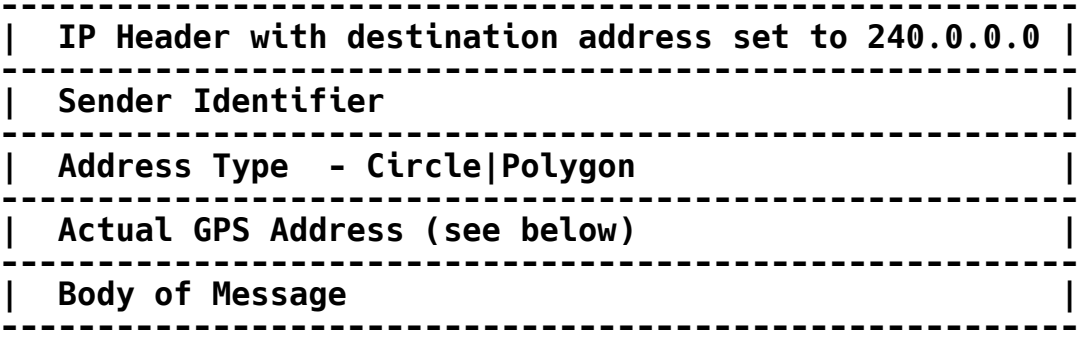
SendToGPS(int socket, GPS-Address *address, char *message, int size)

where socket is a previously created datagram socket, address is a filled GPS-Address structure with the following form:

```
typedef _GPS-Address {
        enum { point, circle, polygon } type;
        char *mail-address;
        struct
        {
                enum { North, South, West, East } dir;
                int hours, minutes, seconds;
        } *points;

} GPS-Address;
```

and message and size specify the actual message and its size. The SendToGPS() routine will take the GPS-addressed message, encapsulate it in an IP packet, and then send it as a normal IP datagram. The message is encapsulated in the following manner:



where the Sender Identifier would consist of a combination of the sender's process id, host IP address, and the center of the destination polygon. The Actual Address would be one of the following:

circle - single GPS address and range measured in centiminutes.

polygon - list of GPS addresses terminated by the impossible address: N 255 255 255.

RecvFromGPS() has the following syntax:

RecvFromGPS(int socket,GPS-Address *address,char *message,int size)

where socket is a previously created datagram socket, address is an empty GPS-Address structure, and message and size specify message buffer and its size.

4b. Establishing A Default GPS Router

The default GPS router is determined using the unicast routing table found in the UNIX kernel. The local system administrator will have previously adjusted the table so that all GPScast messages are sent to the local GPS router. However, if there is no route for GPScast messages in the table, then all messages will, by default, be sent to the default gateway. If the default gateway does not support GPScast messages, then all attempts to send a GPScast will return an error.

By default, all GPScast messages will initially have as their destination the class E address 240.0.0.0. A route will be added to the kernel routing table by the system administrator for this address. The route will specify the location of the local GPS router. The "route" command will be used to affect the routing table and it can be placed in the /etc/rc.local or /etc/rc.ip files so that it will take effect each time the computer is booted. For example, to specify that GPScast messages addressed to 240.0.0.0 should, by default, be sent to the router which resides on a computer on the same subnet with local address 128.6.5.53, use the following:

```
/etc/route add host 240.0.0.0 128.6.5.53 0
```

If the default destination for GPScast messages is a host that does not support GPS addressing, then Network Unreachable errors will be returned to any process attempting to route GPScasts through that host.

4c. GPSRouteD

In order to provide the capability of GPS address routing throughout an IPv4-based internetwork, special-purpose routers will be created to support GPS address routing on top of the current Internet. These routers, which will be called GPSRouteD, will use virtual point-to-point links called tunnels in order to connect two GPSRouteDs together over regular unicast networks. The tunnels work by encapsulating the GPS address messages in IP datagrams and then

transmitting the message to the host on the other end of the tunnel. In this manner, the GPS address messages look like normal unicast packets to all IPv4 routers in between the two GPS address routers. At the end of the tunnel, the receiving GPSRouteD removes the GPS address message from the datagram and continues the routing process.

By using tunnels, the GPS routers can be established as a virtual internetwork throughout the current Internet without regard for the physical properties of the underlying networks. Moreover, the use of tunnels means that the host on which the router daemon is running need not be connected to more than one subnet in order for the router to forward GPS messages. This virtual internetwork would be responsible for routing GPS address messages only. This virtual network, however, is not intended to be a permanent solution and is only intended to provide a means of supporting GPS address routing until it gains wider acceptance and support in the Internet infrastructure.

4c-i. Configuration

When a GPSRouteD initially executes, it first checks the file /etc/GPSRouteD.conf for configuration commands to add tunnel and multicast links to other GPS address routers. There are two kinds of configuration commands:

```
multicast  <multicast-address> <peer|child>

tunnel  <local-addr> <remote-addr>
        <parent|peer|child|host> <service-area>
```

The tunnel command is used to create a tunnel between the local host on which the GPSRouteD executes and a remote host on which another GPSRouteD executes. The tunnel must be set up in the GPSRouteD.conf files at both ends before it will be used.

The multicast command tells the router which multicast addresses to join. These addresses will carry IGPSMP messages and replies. The router will use these IGPSMP messages to build up and keep current its own internal routing table.

4d. Multicast Address Resolution Protocol (MARP)

Of course, this begs the question, how will the individual computers know which multicast addresses to join? For example, an MH would have to join the multicast address of its current cell so that it can receive GPScast messages (using application-level filtering) or directions to join other multicast groups (using multicast filtering). We have designed a protocol called Multicast Address

Resolution Protocol (MARP) that works the same way as Reverse Address Resolution Protocol (RARP). However, instead of returning the IP address of the MH, it will return multicast group address of the cell the MH is currently in. The MH would then join this multicast group.

4e. Internet GPS Management Protocol (IGPSMP)

The Internet GPS Management Protocol (IGPSMP) is used by GPS routers to report, query, and inform their router counterparts about their geographical service areas. The IGPSMP will also be used to verify that routers are correctly functioning.

The vocabulary of IGPSMP will consist of six words:

- set service area - Used by the parent router to set the geographic service area of a router. This is needed in order to automatically respond to router failure or new router boot-up.

- confirm service area - confirms that a router has received its service area.

- geographical service area query - This message will be used by a router to build up its geographical routing table. It is sent to all routers on the same level.

- service area report - This message is sent in response to a query request. It contains a bounding closed polygon described using GPS coordinates which contains the service area for the router.

- ping - This message is sent periodically to ascertain whether the router is currently functioning properly. Usually sent by the parent router in the hierarchy tree.

- alive signal - Usually sent as a reply to the ping message. Used by a router to indicate that it is functioning correctly. It is also sent immediately after a router boots.

All of IGPSMP messages will be sent on an all-routers multicast address for a particular hierarchy level. The exact multicast address can be set in the router configuration file.

Note that for the GPS-Multicast routing scheme, the time-to-live value of the service area reports will be varied in order to control the distribution of the information. In GPS-Multicast routing, only

the multicast group membership for very large partitions will be distributed throughout the country. Smaller partition may only be distributed to neighbor routers.

5. Working Without GPS Information

5a. Users Without GPS Modules

Mobile users without GPS modules can still participate - though at a very reduced level. When an MH enters a cell, it can use an MARP to discover the local multicast group for that cell or atom. As the user roams from cell to cell, the mobile host can keep track of the current cell that the user is in and adds or drops the multicast groups pertaining to those cells. The user's GPS address can be set to be the center of the current cell.

5b. Buildings block GPS radio frequencies. What then?

Each room can have a radio beacon placed on the ceiling. The beacon will be weak enough so that it will not penetrate walls. Each radio beacon will have its own GPS-address associated with it which it will broadcast. When a mobile user enters a room, his MH will detect the beacon and read the beacon's GPS address. The GPS-address of the MH will be set to the GPS-address of the beacon. The MH will then use this beacon's GPS address in order to perform any message filtering that it needs to do. Now the mobile user can have a GPS-address associated with him even though he is indoors and his GPS-module is useless.

6. Application Layer Solution

In this subsection we sketch a third solution which relies more heavily on the DNS.

In the application layer solution the geographic information is added to the DNS which provides the full directory information down to the level of the IP address of each base station and its area of coverage represented as a polygon of coordinates.

A new first level domain - "geographic" is added to the set of first level domains. The second level domain names include states, the third, counties and finally, the fourth: polygons of coordinates, or so called points of interests. We can also allow, polygons to occur as elements of second, third domains to enable sending messages to larger areas.

Thus a typical geographic address can look like

city-hall-Palo-Alto.San-Mateo-County.California.geographic

or

Polygon.San-Mateo-County.California.geographic

where Polygon is a sequence of coordinates.

This geographic address is resolved in a similar way as the standard domain addresses are resolved today into a set of IP addresses of base stations which cover that geographic area. There are several possibilities here:

a. A set of unicast messages is sent to all base stations corresponding to the IP addresses returned by the DNS. Each base station then forwards the message using either of the two last link solutions: application level or network level filtering.

b. All the base stations join the temporary multicast group for the geographic area specified in the message. In this way we may avoid sending the same message across the same link several times. Thus, after the set of relevant base stations is determined by the DNS, the temporary multicast group is established and all packets with that multicast address are sent on that multicast address.

c. Only one, central to the polygon base station is returned by the DNS just as in the IP unicast solution. However that "central" base station will have to forward messages to the other base stations within the polygon.

Notice that we should distinguish between "small area" and "wide area" geographic mail. The "small area" mail will be most common and will most likely involve just one base station, favoring a simple form of solution (a).

7. Reliability

Should the geographic messages be acknowledged?

Since we have no control if users are present in the target geographic area where the mail is distributed we do not see a need for individual acknowledgments from the message recipients. However, we believe that the base stations (MSS) covering the target area of geographic mail should acknowledge the messages.

Typically only a few base stations will be involved since typically we will not cover very broad geographic areas anyway. We assume that the base stations, additionally to forwarding the the messages on their wireless interfaces will buffer them, either to periodically multicast them (emergency response) or to provide them to users who just entered a cell and download the "emergency stack" of messages for that area as a part of the service hand-off protocol.

8. Security Considerations

Some method of determining who has permission to send messages to a large geographical area is needed. For instance, perhaps only the mayor of New York City has permission to send a message to all of New York City.

9. References

Deering, S., "Host Extensions for IP Multicasting", STD 5, RFC 1112, August 1989.

S. Deering. Multicast Routing in a Datagram Internetwork. Ph.D. Thesis, Stanford University, (December 1991).

J. O'Rourke, C.B. Chien, T. Olson, and D. Naddor, A new linear algorithm for intersecting convex polygons, Computer Graphics and Image Processing 19, 384-391 (1982).

J. Ioannidis, D. Duchamp, and G. Q. Maquire. IP-Based Protocols for Mobile Internetworking. Proc. of ACM SIGCOMM Symposium on Communication, Architectures and Protocols, pages 235-245, (September, 1991).

10. Authors' Addresses

Tomasz Imielinski and Julio C. Navas
Computer Science Department
Busch Campus
Rutgers, The State University
Piscataway, NJ
08855

Phone: 908-445-3551
EMail: {imielins,navas}@cs.rutgers.edu