

# UNIVERSAL MOBILE ADDRESSING

## Extended Abstract

Jorge A. Cobb

Chris C. Edmondson-Yurkanan

Mohamed G. Gouda

Department of Computer Sciences  
University of Texas at Austin  
Austin, Texas 78712-1188

### Abstract

We discuss a novel scheme for assigning mobile hosts temporary addresses while they visit foreign networks. These temporary addresses are used in routing data messages that are sent to the mobile hosts during their visits of foreign networks. The main advantage of our scheme is that it does not require any agents for assigning, reclaiming, and keeping track of temporary addresses in any network in the system.

### 1. Introduction

We consider a system, like the Internet [2] and [6], that consists of several computer networks. The computers in each network are classified into hosts and routers. Each network has one or more hosts and one or more routers. For convenience, we assume that each host is attached to exactly one network, and each router is attached to two or more networks.

Hosts generate and send data messages destined to other hosts, possibly in other networks, in the system. Hosts also receive and consume data messages that are destined to them. The function of

routers is to forward the sent data messages from one network to another until each message arrives at its destination host.

Each host has a permanent address  $(N, d)$ , where  $N$  is the network to which the host is attached, and  $d$  is a unique identifier for the host in network  $N$ .

Each router has a set of two or more permanent addresses

$$\{ (N_0, d_0), \dots, (N_r, d_r) \}$$

where  $N_0, \dots, N_r$  are the networks to which the router is attached, and every  $d_k$  is the unique identifier of the router in network  $N_k$ .

When a host generates a data message, it augments the message with the permanent address of its destination host. Thus, each data message is of the form

$$\text{data}(N, d, \text{text})$$

where  $(N, d)$  is the permanent address of the destination host.

When a router receives a  $\text{data}(N, d, \text{text})$  message from some attached network  $M$ , the router uses  $N$  to compute another attached network  $K$ , and forwards the message to another router in network  $K$ , and the cycle repeats. This cycle

continues until finally some router, attached to the destination network  $N$ , receives the message and forwards it to the destination host  $(N, d)$ .

These addressing and routing schemes are adequate as long as hosts are "stationary", i. e. remain attached to their networks indefinitely. However, other schemes are needed to accommodate "mobile" hosts that can move between different networks in the system. Existing schemes for accommodating mobile hosts are surveyed in [4], and some schemes to solve the problem in local-area or campus networks are proposed in [1] and [3].

One scheme for accommodating mobile hosts in wide-area networks is suggested in [5] and [7]. In this scheme, when a mobile host  $(N, d)$  moves to a network  $M$ , it is assigned a temporary address  $(M, c)$  by some agent in network  $M$ . In this case, any data message destined to host  $(N, d)$  is eventually re-routed to host  $(M, c)$ . When host  $(N, d)$  leaves network  $M$ , the temporary address  $(M, c)$  is returned to the addressing agent in network  $M$  so that it can be assigned to other mobile hosts. The problem of this scheme is that it requires an addressing agent in each network in the system.

In this paper, we discuss another scheme to accommodate mobile hosts. This scheme does not require addressing agents for assigning and reclaiming temporary addresses in any network in the system.

## 2. Mobile Addresses

Our scheme for accommodating mobile hosts is as follows. When a host  $(N, d)$  moves to a network  $M$ , different from  $N$ , it assumes the mobile address  $(M, N, d)$ . Host  $(N, d)$  continues to assume this mobile address as long as it is

attached to network  $M$ . When host  $(N, d)$  finally leaves network  $M$  for another network  $K$ , it assumes the mobile address  $(K, N, d)$ , and so on.

When a host  $(N, d)$  moves to a network  $M$  and assumes the mobile address  $(M, N, d)$ , it sends an  $\text{upd}(M, N, d)$  message to each router in network  $N$  to inform the router that host  $(N, d)$  is currently attached to network  $M$ . Later, when another host sends a  $\text{data}(N, d, \text{text})$  message destined to host  $(N, d)$ , the message ends up in a router in network  $N$ . This router recognizes that host  $(N, d)$  is currently in network  $M$ . Thus, it modifies the message, making it of the form  $\text{fdata}(M, N, d, \text{text})$ , and forwards the modified message towards network  $M$ . When the data message arrives at a router in network  $M$ , the router sends the original message to host  $(N, d)$ .

It is possible that by the time the  $\text{fdata}(M, N, d, \text{text})$  message arrives at a router in network  $M$ , host  $(N, d)$  has already left network  $M$ . In this case, the message is discarded.

Before we present these addressing and routing protocols in more detail below, we discuss next how one computer, host or router, can send a message to another computer, host or router, in the same network.

Each computer, host or router, in each network in our system has a unique hardware address in the range  $1..n-1$ . Henceforth, we adopt  $i$  and  $j$  to denote hardware addresses in our system.

For a computer  $i$  to send a message  $\text{msg}(f_0, \dots, f_r)$  to a computer  $j$ , where both  $i$  and  $j$  are in the same network  $M$ , computer  $i$  executes:

$\text{send msg}(j, f_0, \dots, f_r)$  over  $M$

Thus, computer  $i$  needs to know beforehand the hardware address  $j$  of computer  $j$ .

Computer  $i$  can compute the hardware address of computer  $j$  from the permanent address  $(N, d)$  of computer  $j$  by executing:

```
adr := arp.(N, d, M)
```

In this command, `arp` is a function that takes as inputs a permanent address  $(N, d)$  and a network  $M$  and outputs either the hardware address of computer  $(N, d)$  if this computer is currently attached to network  $M$ , or 0 if this computer is currently not attached to network  $M$ .

For a computer  $i$  to send a message  $\text{msg}(f_0, \dots, f_r)$  to each computer in its network  $M$ , computer  $i$  executes:

```
send msg(0, f0, ..., fr) over M
```

We are now ready to define our addressing and routing protocols in detail.

### 3. Advertisement of Mobile Addresses

Periodically, each router in the system broadcasts its permanent addresses using `here(0, N, d)` messages, where  $(N, d)$  is a permanent address of the router. The code for mobile host  $i$  is as follows.

```
var
  CN : the network to which mobile
        host  $i$  is currently attached,
  cr : id of a router in network CN
```

**action**

```
rcv here(0, N, d) →
  CN, cr := N, d
```

**end**

A router  $j$  periodically broadcasts its permanent addresses. Each permanent address  $(N, d)$  of router  $j$  is broadcasted over network  $N$ .

**const**

```
SA : the set of permanent addresses
      of router  $j$ 
```

**action**

```
timeout every  $\tau$  msec →
  for every  $(N, d) \in SA$  do
    send here(0, N, d) over N
```

**rof**

**end**

### 4. Propagation of Mobile Addresses

The above action of mobile host  $i$  is modified to become as follows.

**const**

```
PN : the network in the permanent
      address of mobile host  $i$ ,
pd : the id in the permanent address
      of mobile host  $i$ 
```

**action**

```
rcv here(0, N, d) →
  if CN ≠ N
  then
    CN, cr := N, d;
    adr := arp.(CN, cr, CN);
    send upd(adr, PN, pd, CN) over CN
```

**fi**

**end**

The following two actions are added to the code of router  $j$ .

**const**

```
SN : the set of networks to which
      router  $j$  is attached
```

**var**

```
cnet: array [network, id] of network
```

**action**

```
rcv upd(j, N, d, M) →
  if N ∈ SN
  then
    cnet[N, d] := M;
    send upd(0, N, d, M) over N
```

**else**

```
(K, c) := route.N;
adr := arp.(K, c, K);
send upd(adr, N, d, M) over K
```

**fi**

```
□ rcv upd(0, N, d, M) →
  cnet[N, d] := M
```

**end**

## 5. Routing Using Mobile Addresses

The following two actions are added to the code of mobile host *i*.

**action**

```
true →
/* send a data msg to host (M, c) */
Define M and c;
if M = CN
then
  adr := arp.(M, c, M);
  send data(adr, M, c, text) over M
else
  adr := arp.(CN, cr, CN);
  send data(adr, M, c, text) over CN
fi
```

```
[] rcv data(i, N, d, text) →
  store text
end
```

The following two actions are added to the code of router *j*.

**action**

```
rcv data (j, N, d, text) →
  if ¬(N ∈ SN)
  then
    (M, c) := route.N;
    adr := arp.(M, c, M);
    send data(adr, N, d, text) over M
  else
    if cnet[N, d] ∈ SN
    then
      adr:=arp.(N, d, cnet[N, d]);
      if adr ≠ 0
      then
        send data(adr, N, d, text)
        over cnet[N, d]
      else
        skip /*discard data msg*/
    fi
  else
    (M, c) := route.cnet[N, d];
    adr := arp.(M, c, M);
    send
```

```
fdata(adr, cnet[N, d], N, d, text)
over M
```

```
fi
fi
>[] rcv fdata (j, M, N, d, text) →
  if ¬(M ∈ SN)
  then
    (K, c) := route.M;
    adr := arp.(K, c, K);
    send fdata(adr, M, N, d, text) over K
  else
    adr := arp.(N, d, M);
    if adr ≠ 0
    then
      send data(adr, N, d, text) over M
    else
      skip /*discard data msg*/
    fi
  fi
end
```

## References

- [1] D. Cohen, J. B. Postel, and R. Rom, IP addressing and routing in a local wireless network, Proc. INFOCOM 1992, pp. 626-632.
- [2] D. Comer, Internetworking with TCP/IP, Volume I, Second Edition, Prentice-Hall Inc., 1991.
- [3] J. Ioannidis, D. Duchamp, G. Q. Maguire Jr., IP based protocols for mobile internetworking, Proc. ACM SIGCOMM 1991, pp. 235-245.
- [4] A. Myles and D. Skellern, Comparing four IP based mobile host protocols, Computer Networks and ISDN Systems, Vol. 26, 1993, pp. 349-355.
- [5] W. Simpson, IP mobility support, work in progress, July 1994.
- [6] W. R. Stevens, TCP/IP Illustrated, Volume 1, Addison-Wesley, 1994.
- [7] F. Teraoka, Y. Yokote, and M. Tokoro, A network architecture for providing host migration transparency, Proc. ACM SIGCOMM 1991, pp. 209-220.