# Mobile Ip

Mobile internet is wildly used today (laptops, palmtops, PDAs) so wireless communications are needed: several technologies can be used, 802.11, Blue tooth, GPRS in order to support data delivery traffic.

The wireless channel, however, has not so much good properties: the error probability is higher than cables and also propagation conditions are different; moreover nodes can move.

An important distinction has to be mentioned: the difference between *portable* and *mobile* network: a portable network forecast that nodes, when move, change connection by changing point of attach to the Internet while mobile networking forecast that a node moving is still attached to the Internet through the same connection. The second aspect is much more complex to manage with respect to the first one. Indeed, in the Internet, nodes are identified thanks to the Ip address that describe the subnet of the user: this notion is geographical, so if the node moves into another subnet it can not still use the old address: it needs a new one. This problem, at layer 4, is also valid at layer 4: a TCP/UDP connection is identified through a quadruplet: Ip source and destination, source and destination port; if one change the connection is lost.

Solutions that can be adopted are:

. change the Ip address when the attach point changes;

. specify routes using source routing;

. use another level of indirection (mobile Ip).

Mobility at network layer has several advantages:

. is transparent to upper layers;

. is a layer present in all nodes;

. is the layer entitled to provide routing;

. is independent on the physical medium.

Mobile Ip has the following design goals:

. a node, after change attachment, still can communicate in a secure manner (authentication phase) with other nodes, also if they do not support mobile Ip;

. be transparent to upper layers;

. be scalable (reduce overhead due to management messages);

. not require constraints on the assignment of addresses.

### Terminology

. Mobile node (MN): host or router that change point of attachment;

. home network (HN): the original subnet of the MN;

. foreign network (FN): the currently visited subnet;

. correspondent node (CN): the node in which the MN is communicating with;

. home agent (HA): a router on the HN that tunnels datagram to the FN;

. foreign agent (FA): a router in teh FN that provides routing functionalities;

. encapsulation: incorporate an Ip packet into another one;

. tunnelling: similar to encapsulation, but provides additional information on routing.

## Procedure

A node can uses tow addresses:

. the home address is the original one: it is static and assigned to identify the user in the HN;

. the care-of-address is the address assigned on the foreign network.

When a source (CN) has to deliver traffic to a node that moves from a subnet to another one, the protocol acts like this:

. the destination MN informs the home agent of his movement (registration procedure);

. if the registration procedure succeeds the HA re-route traffic to the FA thanks to encapsulation: this is tunnelling;

. once packets reach the FA, they are forwarded to the destination: it is easy because the care-of-address belong exactly to that FN;

. packets that go from the CN to the MN can be sent immediately to the source if there are nod firewalls, or to the HA that forward them.

## Protocol

The protocol is composed of:

. *discovery phase*: the care-of-address to assign to the MN has to be founded;

. *registration*: the procedure for which the MN registers his care-of-address to the HA;

. *data-delivery*: how packets are sent to the MN (tunnelling).

### Advertisement and solicitation

To discover the care-of-address is possible use functionalities already present in the Internet: the router discovery function of ICMP. Advertisement are sent periodically by routing, but if a MN does not want to wait can explicitly solicits routes.

Agent advertisement:

. allows the detection of MNs;

. MNs can detect their subnet (Net Id) so they can understand if it is a HN or FN and consequently discover HA or FA;

. in case it is FA, that agent is entitled to inform the MN about his encapsulation techniques and lists one or more care-of-addresses that are available.

A MN detects the movements if:

. it receives an agent advertisement with network prefix different from the one owned;

. the lifetime (it is a field of the packet) of the last received agent advertisement has expired and the MN has not received a new agent advertisement.

### Registration procedure

The registration procedure is used by MN to inform HA of its current care-of-address. HAs creates *bindings* composed of:

. MN's home address;

. MN's care-of-address;

. registration lifetime.

Bindings contains several care-of-address and are stored into the mobile node's home address. If the MN comes back to the HN, he has to de-register himself. Registration is also used to renew binding about to expire.

**Request**

. Inform the HA of MN's care-of-address.

. Inform the HA for how long the MN is going to be out of the HN.

. Inform the HA of FA's features.

The HA can block the request or reduce the lifetime of a binding.

**Reply**

. Notify if the request has been accepted.

. In negative case, a reason is explained.

During the registration the FA can:

. limit binding lifetime to the value that it has inserted into agent advertisements;

. maintain a list for every pending and current registration;

. use the list to delivery packets.

Since the registration is an important phase, malicious users can try to force it; two mechanism can be used:

. *redirection attack*: it happens when a malicious user disrupts the communication between MN and HA; a false registration request is sent and it contains a bogus care-of-address: after it all traffic will be re-directed;

. *reply attack:* the malicious user senses requests and replies pretending to be the HA, therefore no traffic will be re-directed to FAs and, therefore, to the destination.

To avoid these problems, a values is included into the identifier and changes for each registration; it could be:

. timestamps;

. nonces.

For timestamps digital signatures are needed to avoid that attacks will predict the time. Nonces are pseudo random numbers generated by an algorithm, equal for source and destination. The seed used is a secret key share, again, by source and destination such that the nonce can be decrypted only by them.

### Data delivery

Once the Mn has registered his care-of-address, the source can send packets being sure that they reach the destination thanks to tunnelling: the HA forward them, by encapsulation, to the FA that is in charge of deliver them to the destination. Encapsulation techniques used are:

. Ip-in-Ip;

. minimal.

The first one is very simple: an Ip packet is directly encapsulated in a new Ip packet and few fields change (done to recognize that this packet is an encapsulated one, to show the new source and destination and to have reliable checksum and length).

Minimal encapsulation not repeat headers, so introduces less overhead while requires much more processing capability. The inner header is copied because the outer is very similar, then few fields are modified.

### Broadcast

HAs forward everything a part form ARP packets, so broadcast packets can be either sent directly or encapsulated again.

ARP packets are detected by MNs, when they are in FN, tanks to the fact that HAs use proxy ARP (tunnelling) because they can not transmit or receive ARPs directly.

## Route optimization

The tunnelling, based on triangulation, is not so much efficient. Route optimization tries to tunnel datagrams directly to the care-of-address of the MN; it consists in two parts:

. binding caches;

. smooth handoff.

### Binding caches

When a HA receives a datagram to forward to the MN sends also a binding update to the source: this message is sent for every packets, so it does not require acks.

The procedure need authentication and, after that, the source can sends packets immediately to the FN without crossing first the HN.

### Smooth handoff

This procedure takes place when a node visits several FNs in sequence: the basic protocol does not notify that fact to previous FAs. Thanks to route optimization MN and Fa can *share* a registration key to make later movements smooth and avoid packets loss.

The procedure is:

. when a MN registers to a new FA, it sends a registration request including the previous Foreign Agent Notification Extension;

. the new FA creates a binding update to the previous FA, requesting a binding acknowledgement: this is part of the registration procedure;

. the authentication needed is guaranteed by the MN through the secret key shared with the previous FA;

. when a FA receives a tunnelled datagram, it informs the HA with a binding warning that the MN has changed FN and the HA sends a binding update the the CN.

In this way the source still sends packets to the first FA, that is entitled to forward them to the correct FA: this reduces issues in traffic delivery.