Role-based address allocation and distribution mechanism Technical report

Damien LEROY

October 2, 2007

Contents

1 Motivations

The growth of the BGP routing tables in the default-free zone is again a concern for many network operators. So as to tackle this problem, the IRTF has chartered the Routing Research Group to propose new solutions to improve the scalability of the Internet routing architecture. A key reason for the growth of the BGP routing tables is the way IP addresses are allocated and used. The pool of available IP addresses is managed by the regional registries (RIPE, APNIC, ...). These registries define two types of addresses : Provider Aggregatable (PA) and Provider Independent (PI). To limit the size of the BGP routing tables, only large ISPs should obtain PI addresses while customer networks should receive PA addresses from the PI block of their upstream provider. Unfortunately, if a corporate network uses a PA address block, it should change the addresses of all its network when it changes from its upstream provider. For this reason, most corporate networks insist on obtaining PI addresses. Combined with the growth of multihoming, this explains the growth of the BGP routing tables. This growth could be avoided if corporate networks and smaller ISP networks were able to more easily use PA addresses. Unfortunately, with the current Internet architecture, using PA addresses implies that the corporate networks must be renumbered each time it changes from provider and several studies have shown this to be painful with both IPv4 and IPv6.

During the last years, extensions to the Internet architecture have been proposed. Several of these solutions rely on the separation between the two distinct roles of IP addresses : **identifier** and **locator**. An identifier is an address used to identify (the applications running on) one endsystem. An endsystem usually has one identifier. It can be a cryptographic identifier (such as with HIP) or an IP address (such as with LISP) obtained from PI space. A locator indicates the attachment point of an endsystem or a router. A router and a multihomed host have multiple locators assigned to them. A mapping mechanism is used to derive one locator from an identifier. When an endsystem moves or its upstream provider changes, its locator(s) change(s) but its identifier remains the same.

In the early days, IP addresses were allocated manually to both routers and endsystems. However, this manual allocation was a cause of errors and problems. As a consequence, most endsystems now obtain their IP address automatically either via DHCP or via autoconfiguration. Despite the widespread use of automatic configuration of endsystems, the addresses used by the routers are still manually configured (except in small networks by using DHCP extensions) and several studies have shown that configuration errors are responsible for a large number of operational problems.

In this paper, we propose a distributed mechanism that allows IP addresses used as locators to be automatically distributed and assigned to routers inside the network. The routers then are responsible for the suballocation of these locators to their locally connected endsystems and customers.

The zero-configuration protocol proposed in [CFT05] tackles to similar problems. Nevertheless, their solution is limited to small networks and does not have any security consideration. The *autoconf* working group at IETF is also offering close solutions than ours [BMRS07]. However, their work is focussed on ad-hoc networks, thus they do not have the same hypothesis and objectives.

2 Terminology

2.1 Definitions

Node names

We call **router** a device in our network that runs our protocol, **host** an end-device in our network that cannot run our protocol. What we called **subnet** is an edge sub-network containing several hosts and routers on the same LAN, i.e. linked together with wires, hubs and switches.

CPE (Customer Premise Equipment): router that is directly connected with one or several providers.

PE (Provider Egde device): router directly connected with one or several customers.

SER (Subnet Egress Router): egress router of a subnet .

Note that a router can combine several of the previous roles.

Parts of an IPv6 address

An IPv6 address can be viewed as three different parts as shown on Fig. 1 and seen through an enterprise network view:

1. A global routing prefix that we will simply call **prefix** in this document. The number of bits of this prefix will be referred as P_S. P_S should be equals to 48 in most edge networks (an enterprise, ...) [II01, rip06]. In our case, we will consider $3 < P_S < 64$. This prefix will be attributed to us by our provider. We can receive several prefixes if we are multihomed.

2. A subnet ID (**SID**) uniquely identifies a subnet in our network. Its size is equals to $64 - P_S$ and has 16 as a typical value. For our protocol, we will split it into 2 parts:

   (a) An attributed subnet ID (**ASID**) that is uniquely attributed by the network to a customer. The number of bits allocated to ASID will be referred as $ASID_S$.

   (b) A delegated subnet ID (**DSID**) that is the part of the address that under the customer's authority. The number of bits for customer allocation will be referred as $DSID_S$.

   Note that $DSID_S$ and $ASIS_S$ are ranged between 0 and $64 - P_S$. In the case of a local subnet or a customer that only need a single subnet, $DSID_S$ is equals to zero. $ASIS_S$ will never be set to zero (i.e. $DSID_S = 64 - P_S$) since there would be no more address available for our network.

3. An interface ID (noted **IID**) that is uniquely attributed in a subnet and that identify the host in it. The size of IID will always be 64, it can be attributed by egress router using DHCP [DBV+03] or by the host itself using the Neigbor Discovery protocol [NNS98] or SEND, the secure version of it [AKZN05].

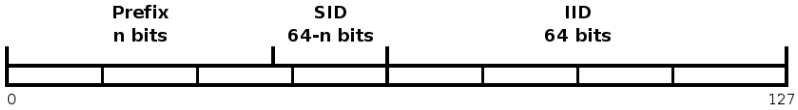


Figure 1: Structure of an global unicast address according to [II01]

If we consider that our customers use the same definitions than ours, customer's prefix corresponds to our prefix concatened with the DSID we have attributed to him.

Address status

This section describes an extension to the standard IPv6 address status [II01]. These new status are of course only used for internal conventions in this protocol. They can be used with either a single address or an address block.

Here a list of these status with a short word about them. A representation of them on a timeline is showed below the list.

Free : it is not really a status, an address is considered as "free" when it has no other status i.e. when it is not reserved by anyone.

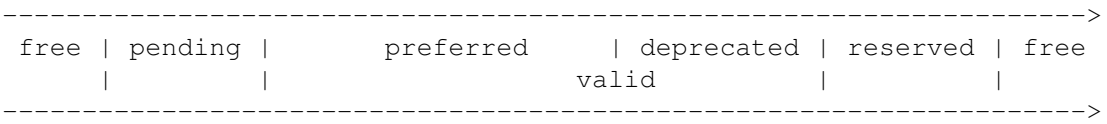
Pending is used to annonce that this address will be used in a few second and is waiting for collision notification if any.

Preferred status means that this address is used for the moment and is defined as preferred to hosts to which it has been attributed.

Deprecated is used to annonce that an address is still used by hosts but that it should be avoided as much as possible for new connections.

Valid is corresponding to an address with either the preferred or deprecated status.

Reserved status is used in our protocol when an address valid time is elapsed but is still reserved for use to the former router in charge of it.



2.2 Notations

P_S : the prefix size (number of bits)

$ASID_S$: the ASID size

$DSID_S$: the DSID size

ISP : Internet Service Provider

"+" represents concatenation in address representation

$t_{preferred}$ is the time in seconds the address is "preferred", default 3600.

t_{valid} is the time in seconds the address is "valid", default 7200.

3 Introduction to secure role-based locator distribution

3.1 Goals

The goals of the protocol are to :

- Number a whole network from the routers connected to the provider to the ones connected to the customers or to subnets. It must work whatever the topology is.
- Have no need for human intervention on each router while renumbering.
- Observe the Provider Aggregatable (PA) address space base concept: customers receive subsets of the addresses owned by their providers.
- Despite that PA addresses are used, avoid provider lock-up by making provider transition easier.
- Enable IPv6 multihoming, for the network itself and its clients.
- Ensure stability in address allocation especially for end-hosts, i.e. avoid as much as possible address renumbering for end-hosts.
- Bear an HD-Ratio [DH] equals to 0.94 following [dra] (draft of modifications to [rip06]).
- Starting from scratch, obtain a stable allocation in the whole network after a few seconds.
- For large networks, allow grouping routers in areas.
- Support firewalls, DNS and other related issues

3.2 Non-goals

- Unlike the zeroconf protocol, the objective is not to avoid all the router administration tasks. We just want a simple one and to do it only once.

3.3 Protocol overview

Our locator distribution mechanism is targetted for edge networks as well as ISP networks that need to provide locators to their customers. We encode locators as IPv6 addresses but there is no hindrance to apply the mechanism to another format. A full description of the protocol can be found in [Ler07]. The objective of the mechanism we propose is to dynamically assign and securely distribute locators in an entire network. We assume that each router is configured with an X.509 certificate indicating that it belongs to the network. In association with the corresponding public key, it will also be used to sign the locators attributed to client networks. Furthermore, we allow a network to divide its locator block in different roles (e.g. a sub-block reserved for servers, another one for customers, another one for loopback addresses, ...). These roles are very important because they will allow the locators to be assigned in a way that simplifies the configuration of packet filters on the routers.

Fig. 2 represents a typical environment where our mechanism could be used. In this figure, arrows represent the direction of locator allocation. Our mechanism is composed of three main parts: One or several prefixes are obtained from our upstream ISP by border routers; Our child networks needing a locator block ask for it at border routers; Core routers negotiate automatically which parts of the obtained address block have to be attributed to child networks.

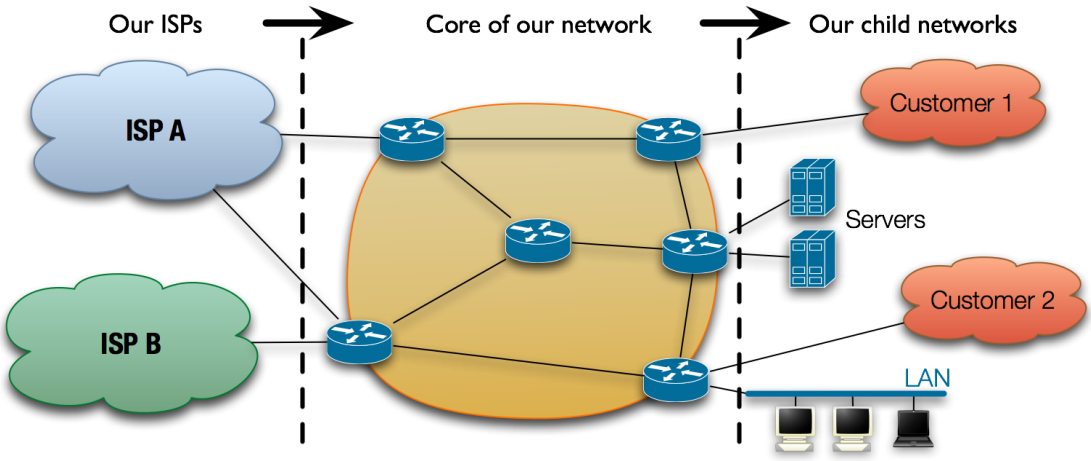


Figure 2: The protocol does apply on hierarchical topology

Our locator distribution protocol behaves as follows. Rou-ters communicate hop-by-hop, connections are only made with direct neighbors after authentication. The messages received by a router will be, if necessary, flooded to neighbors so that each router in the core receives the information. Global prefixes negotiated with ISPs are flooded to all core routers. These will append ASIDs to prefixes in order to obtain locators to deliver to child networks.

Each router in charge of one or several authenticated child networks starts off with choosing among free SIDs an address block where it can place all of them. Next, this address block is advertised through the network and the router waits a short while for a potential collision notification. If a collision appears, the procedure is restarted. Otherwise, all the routers retain this reservation and its initiator attributes locators to all of its child networks.

4 Messages

This section describes of the messages used in this protocol. All these messages must be sent with and to link-local addresses. A message from another address should be dropped. Messages should also be sent with hop-limit (TTL) of 255 and received ones should have this hop-limit value.

4.1 Discover message

This message is used to discover other routers that use the protocol on a link. It is sent to the link-local router multicast group (ff02::1). It is a ICMP message.

Fields:

router id: the router id of the sender of this message

4.2 Hello request message

This message is sent to initiate a new connection with a neighbor that we know thanks to its *discover* message.

Fields:

router id: the router id of the sender of this message

my session id: nonce session id chosen by the sender and unique for a connecion, it must be put in each response message

PKC: the public key certificate of the sender

AC: the attribute certificate of the sender

Signature: The previous fields signed with the private key of the sender

4.3 Hello response message

This message is sent as a reply to a new connection request from a neighbor.

Fields:

router id: the router id of the sender of this message

my session id: nonce session id chosen by the sender and unique for a connecion, it must be put in each response message

session id: session id chosen by the recipient

PKC: the public key certificate of the sender

AC: the attribute certificate of the sender

Signature: The previous fields signed with the private key of the sender

4.4 Address Advertisement message

router id: the router id of the sender of this message

seq number: a sequence number greater that the previous ones sent to the recipient

session id: session id chosen by the recipient

number of address entries: the number of address entries in this packet

address entries: A list of address entries, each one with the following fields:

Router id of the router in charge of the entry

Customer id if any

Role of the customer in this network

P_S for which this ASID is attributed

$ASID_S$

$ASID$

Preferred time left

Valid time left

Sequence number associated with the triplet <*router id, customer id,* P_S>

Signature: The previous fields signed with the private key of the sender

4.5 Prefix Advertisement message

router id: the router id of the sender of this message

seq number: a sequence number greater that the previous ones sent to the recipient

session id: session id chosen by the recipient

number of prefix entries: the number of address entries in this packet

prefix entries: A list of prefix entries, each one with the following fields:

prefix: the address prefix

prefix length: the number of bits to consider in the prefix

preferred timeout

valid timeout

Entry seq number: a sequence number that is proper to the data associated with this prefix, it is incremented when new information about this prefix is generated.

Signature: The previous fields signed with the private key of the sender

5 Security statements

5.1 Goals

The goals in terms of security:

- Only a set of nodes is allowed to send messages using the protocol.

- Allowed nodes can be authenticated in the network by any other node.
- Integrity of messages exchanged has to be maintained.
- Replays should also avoided.

Note that confidentiality is not needed.

5.2 Infrastructure

The security is based on a Public Key Infrastructure (PKI) deployed in the network. The Certification Authority (CA) does not need to connected to the network.

In order to be authorized to send this protocol messages through the network, each router owns:

- An unique public/private key pair.
- An Internet X.509 Certificate [HPFS02] that associates its key pair with a network-unique 64-bits identifier and signed by the CA.
- An Attribute Certificate [FH02] (AC) that associates the identifier of the router with its role "router" and signed by the CA.

In order to authenticate the other ones, a router needs:

- The public key of the CA.

Validity conditions and revocation mechanisms have to be discussed since no check to the CA's revocation list can be done before it has a connectivity. Some ideas in [EFL+99].

5.3 Mechanisms

Before being "inserted" in the network, a router has to obtain its ID certificate (Internet X.509 Certificate) and AC. Using them, it will sign all the messages it sends except the *discover* ones. When a connection is started, the routers first exchange their cerficates.

When a *discover* message has been received, two messages are exchanged to initiate the authenticated connection. Some features has to be added in order to achieve the security goals:

- In order to avoid replay attack, a *sequence number* field is included in each message. A sequence number counter is unique for a router on a connection with a neighbor. The sequence number starts from 0 and is incremented each time a message is sent. This receiver must drop messages with a sequence number that is smaller or equals to the last received one.
- Each router will generate a nonce session id for each connection it starts. The session id of the recipient will be included in each message sent for this connection. It is used to avoid replay attack from a connection to another one. This number has also to be checked by the receiver.
- The signature applies on all the message fields.

6 Model of the network

6.1 Networks components

The protocol is designed to be used in an enterprise-like or an ISP network. We will consider several components in it:

Routers are the nodes that run our protocol. A router can be connected either to another one, to a subnet, to a customer, to an ISP or to several of theses. The routers in the network are organized as a connected graph. If we consider the routers as nodes and an edge each time two routers are on the same subnet, there must be a path beween each pair of nodes.

Subnets are directly connected to routers. Following the IPv6 specifications, they need a /64 prefix. The router (or routers) to which the subnet is directly connected is a SER. Several subnet can be connected to a router and a subnet can have several SER. In this latest case, the default behaviour for the SERs is to advertise the same prefix. From now, we will consider a subnet as a simple customer that need a /48 prefix ($ASID_S = 64 - P_S$ and $DSID_S = 0$). A subnet is uniquely identified by a customer id.

Customers are consider as directly connected to some routers, called PE, that store all the information about each customer attached to themself. This information includes a customer id and its prefix size requested. Several customers can be connected to one PE and a customer can be connected to several PE. Protocols to delegate addresses to the customer premise equipement are discussed in section **??**.

ISPs are considered as a black box connected to some routers, called CPE. The CPE are in charge of negociating network prefixes with the ISP. An ISP can be connected to several CPE and a CPE can be connected to several ISP.

6.2 Manual configuration

In this section, we describe the parameters that has to be configured manually in each router. It corresponds to some static information needed to authenticate the router in the network and to know the addresses it has to request.

Information to store about the router:

- Its 64-bits router id,
- Its Internet X.509 Cerficate,
- Its Attribute Certificate,
- The public key of the CA,
- A list of interfaces on which the protocol is active,
- The id or name of the network.

Information to store in a router about each of its subnets or customers:

- The customer id,
- Interface it is linked to (and eventually the gateway protocol to configure),
- The $DSID$ requested,
- Whether or not it needs site local addresses.

7 Areas

In a large network containing a large number of routers or high delay links, it is sometimes interesting to split the network into smaller areas. It is even more important for our protocol since otherwise flooding can be quite long and tables stored pretty large. Moreover, areas can be useful for efficently improving the address aggregation in FIBs.

To deal with areas, the decision has been made to let them be configured manually by network administrator. Indeed, we believe that these are large scale choices that still need human decisions. Moreover, when a router move from an area to another, some other reconfigurations have still to be made.

The way areas work in our protocol is quite simple. If they are used, each router has an area number associated with it. This number is coded with few bits. These bits will corresponds to the first bits of the ASID that will be choosen. All the area number must have the same length in a network.

Areas number is configured for each interface, so a router can have interfaces on different areas. Only the prefix advertisement messages are forwarded from an area to another one.

8 Network discovery

This section describes the first steps of the protocol, these are the probes sent by routers to discover their direct neighbors and initiate connections with them. Neighbor information will be stored in a table we will call the *neighbor table*.

Before starting the next steps, link-local addresses have to be assigned to each router. Neighbor Discovery [NNS98] is sufficient to do it and is normally complete in a very short time but an adresse generated with EUI-64 or with a manual solution is also admissible.

At startup and every $t_{discover}$ seconds, a router send a `discover` message on each its interfaces that run our protocol. This message is sent to the link-local router multicast group (ff02::1) that have to be listened by all the interface using the protocol. This message contains the router id of the sender.

When a `discover` message is received with r as router id.

- If r belongs to the *neighbor table*, ignore the message.
- If not, initiate a TCP connection with this node and send it a `hello req` message.

When a `hello req` message is received with r as router id.

- If r belongs to the *neighbor table*, close the connection.
- Check the signature of the message with the supplied PKC, if it is wrong, close the connection.
- Check the validity of the PKC and AC and their matching with r. If it failed, close the connection.
- If three previous checks passed, add an entry in the *neighbor table*: (r, PKC_r, AC_r, $seqnumber = 0$, $sessionid$).
- Send a `hello resp` message in response.

When a `hello resp` message is received with r as router id.

- Check the session id in the message that has to be the one the router has generated, if it is wrong, close the connection.
- Check the signature of the message with the supplied PKC, if it is wrong, close the connection.
- Check the validity of the PKC and AC and their matching with r. If it failed, close the connection.
- If three previous checks passed, add an entry in the *neighbor table*: (r, PKC_r, AC_r, $seqnumber = 0$, $sessionid$).

When the `hello resp` message has been sent or received, the address allocation protocol itself can start. (see section 9.2)

9 Address allocation protocol

This section describes the addess allocation protocol itself, i.e. the messages exchanged and how to deal with the addresses.

9.1 Stored information

In addition to the *neighbor table* defined in the previous section, some more dynamic tables are stored for address allocation.

9.1.1 Address table

An address table is used to store all the addresses attributed in the network for the moment. An entry in it has the following fields:

Router id of the router in charge of this address

Customer id if any

P_S for which this ASID is attributed

$ASID_S$

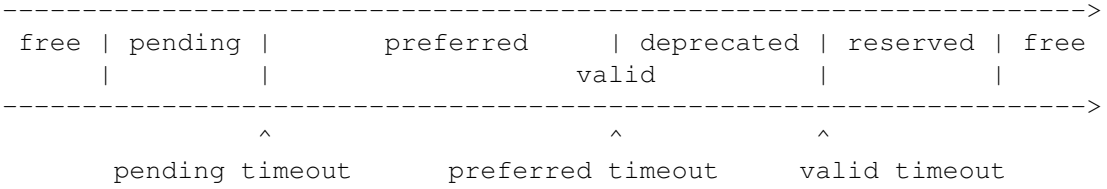
$ASID$

Pending timeout . See the timeline above.

Preferred timeout . See the timeline above.

Valid timeout . See the timeline above.

Latest sequence number is the sequence number associated with the latest modification of this entry.



9.1.2 Prefix table

A prefix table is also used to store the prefix advertised. An entry in it has the following fields:

Prefix

Prefix length : the number of bits to consider in the prefix.

Preferred timeout

Valid timeout

9.2 When a connection with a new neighbor has just started

When the *hello* messages have been exchanged between two hosts, each host sends to the other one its entire address table in an Address Advertisement message. When the table of the other one is received, the tables are merged with the same mechanism as explained in section 9.6.2. Once it is done, router can start generating addresses for its customers. Of course, there no new address allocation to make if the current table already contains addresses for these customers.

9.3 Address allocation protocol "in general"

9.3.1 Router address attribution

When a router learn a new prefix to use, the first thing it does is to compute its own address using this prefix. Note that it can be choosen not to attribute globally routable address for router. We suppose here that they have to obtain an address in each prefix.

The router address is obtain with the SID bits set to zero and its router id as IID. The result is: *PREFIX* + 0000...0000 ($64 - P_S$ zeros) + *router id*. Since the router id is unique, we know that this address is not used by another router. These addresses are not advertissed to the whole network. If a message has to be sent to a router, the address can be easily recreated from its router id by any other host that knows the prefixes.

9.3.2 Customer address allocation

Next, the router allocates one address block for all its customer. To do that it computes the size of the prefix it needs to place all its customers. It is a good idea to prevent a future extension of 1.5 of the address needed by customer. Once the prefix is choosen, ASID can be flooded. Once it is done and no collision detected, ASID for each of its customer are chosen.

9.4 Address allocation for a network without a global routable prefix

The non-connectivity to the Internet should not be a barrier to communication between hosts in the network. That is why site local addresses are allocated first. These addresses, defined in [HH05], are only routable in the network and not through the Internet. The local 48-bit prefix is built from concatenation of "FD" (8 bits) and a 40-bit global ID that is common to the whole network. This ID will be made from the 40 first bits of the network ID configured in each router. This prefix is implicit and has not to be advertised between the routers. So when a network has just been started, routers directly distribute ASID as they received a 48-bit prefix. The only difference with other prefixes is that customers can be configured not to obtain local addresses. Actually, real customers do not often need local addresses.

9.5 Address allocation for global prefixes

Things are not really different with global prefixes. When a new global prefix is known, we can consider two cases:

- If the prefix size is equals to another one that is already used, routers automatically use the ASID used with the other one (Actually ASID are not associated with a prefix but with a prefix size). So, no new advertisement are needed between routers. The only exception is when the former prefix is only the site-local one. In this case, customers that did not obtain local addresses must obtained an ASID.
- Otherwise, an ASID has to be reattribute to each customer with the protocol described in section 9.3.

9.6 Address Advertisement Messages exchanged

9.6.1 Messages sent

Each time a new address or a group of addresses is added in a router address table, the router has to send an Address Advertisement message. The router id is the id of the router that has generated this address. The sequence number is the previous one +1 and is proper to each neighbor.

The entries sent are the new ones that have just been created and the ones that have just been modified. For the new entries, the preferred time left is $t_{preferred}$ and the valid time left is t_{valid} and the sequence number is 0.

When this is not a new entry, the sequence number is incremented and the entry is added to the message.

9.6.2 Messages received

Each time an Address Message is received, all the entries are compared with the ones stored. For each couple *router,customer* (identified by their ids), the sequence number of the entry is compared. If the received entry is strictly greater than the stored one, the received replaces the former one. Once it is done for each received entry, all the entries that have replaced older ones, are sent in a new message to all the neighbors except the ones on the link from which the incoming message was received. If there was no new information in the message received, no message is sent.

If two routers choose at the same time addresses block that overlap, a collision occurs. It can also occur when there are address tables merging. In this case, one of the proposal is kept in each router and the other one is dropped. This dropping is silent since the sender of the message will also receive the second message. The rule is the following:

- If the status of the two entries are different, the order is the following (if a<b, a is dropped and b is kept): *reserved* < *pending* < *deprecated* < *preferred*
- If the two requested address blocks have a different size, only the larger block is kept.
- If the size are the same, the entry with the higher router id is kept.

add explanation prefix

9.7 Prefix Messages exchanged

Prefix messages are introduced by a human intervention or another protocol, by the gateway typically. When a prefix message is received, if it is new of if the entry sequence number is greater than the stored one, new information is stored and message is sent to all neighbor except the ones on the message incoming link.

10 Address allocation algorithm

The address allocation algorithm needed for our purpose is a very specific one. Although there are some algorithms proposed for MANET address or for disk space allocation, no such problem are really equivalent to our needs.

Our needs:

- We have an address space of $64 - P_S$ bits, i.e. 2^{64-P_S} slots available
- The blocks that have to be allocated need $DSID_S$ bits, the $DSID$ must be different for each customer.

- Address space allocation should observe 0,8 as HD-Ratio. (or 0,94 if we want to be compliant with the RIPE draft [dra])
- Allocation should first aggregate address of a same role as much as possible
- Next, allocation should aggregate address of a same router as much as possible
- Customer reservations could be added or removed at any time
- Address changes should be avoided as much as possible and when it is absolutely needed, the former and the new address block has be reserved for the customer for a while.
- It should be possible for a customer to grow without changing its allocated addresses.

11 Evaluation

In order to validate our assumptions, we have written a simulator that allows to evaluate the performances of the locator distribution protocol. We are also working on including the protocol in XORP. More evaluations are shown in [Ler07].

Simulations are run on a 110-routers-topology coming from a true ISP network. Child networks with random size are uniformally associated to routers to obtain 0.8 as an HD-Ratio[1]. So we have around 4,500 client networks. The tests consist of starting this configuration and applying regular modifications to the child network topology. Modifications include client adding, removing, reducing and enlarging. We monitor 10,000 changes for each experiment.

Let us begin with the protocol overhead. First, we can consider the handshake messages between neighbors, i.e. the authenticated hello exchange and the keepalives. These messages never exceed 10 messages per minute with one hop. Next, let us consider the messages sent by the core routers when one of their child networks is modified. Since a router makes the reservation of an address block for all its clients, only one message is sent at startup by router having clients. Next modifications do not generate lots of messages since most changes can be done within the reserved blocks of the router. We have measured a mean rate of 0.03 messages received per router and per modification, i.e. 3 messages on each link for 100 modifications.

Convergence time measured is usually very short. Since routers do not start reserving a block at the same time (in order to avoid collisions), the convergence time will be equals to this bootstrap time. While running, modifications will be applied either directly if no new block has to be requested or after a short while corresponding to collision notification waiting otherwise.

For scalability experiments, evaluations have been made with an larger number of customers for a same HD-Ratio. The maximum value for customers was 7,131 with HD-Ratio of 0.8. This configuration does not cause any difference to the normal protocol performance. If the HD-Ratio is raised, routers have more difficulties to find a block to reserve. Our simulations show that it has to be maintained under 0.94 to avoid address block allocation difficulties. Note that in our algorithms, routers always reserve a block a bit bigger than what it needs; this behavior should be avoided if we work with high HD-Ratio.

Another interesting result is the number of entries stored in router tables about other router reservations. The router forwarding table will also be proportional to this number. Since each router makes reservation of one prefix for several clients, we known that the lower bound of prefixes known is the number of routers that have clients. The upper bound is the case in which each client has a specific prefix. With the test bed describes above, we observe a size of 110 most of the time and a maximum value of 112. Table 11 summurize all these results and fig. 3 represents through the time.

[1] HD-Ratio is defined in RFC 3194. 0.8 is the threshold defined by regional registries from which additional address allocation is justify (see ripe-388).

	#routers	#clients	#prefixes
min	110	4,225	110
max	110	4,765	112
mean	110	4,453	110.4
std dev	0	151	1.37

Table 1: Number of entries in prefix tables

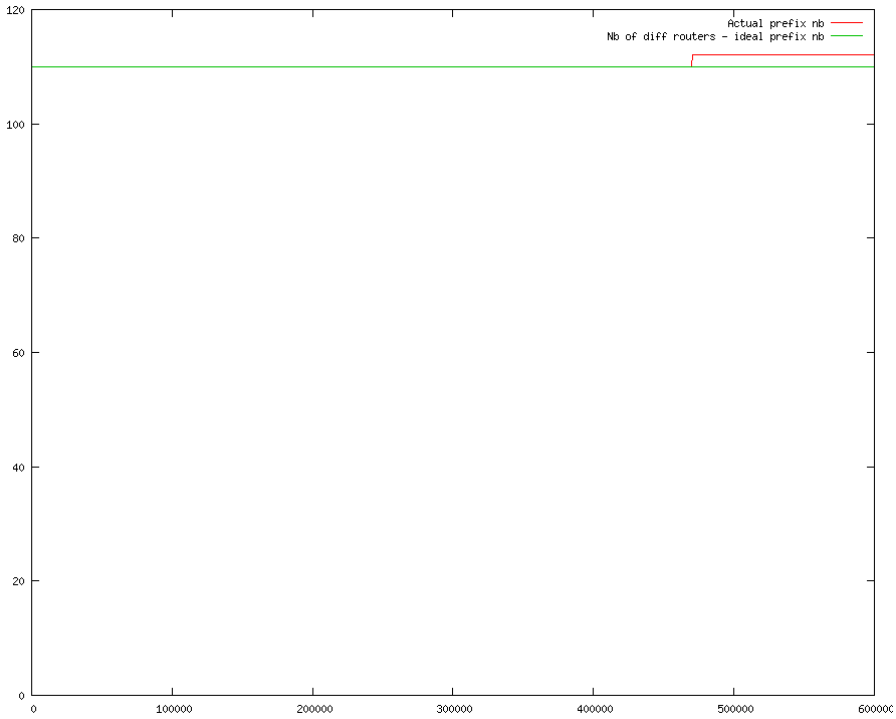


Figure 3: Table size through the time

References

[AKZN05] J. Arkko, J. Kempf, B. Zill, and P. Nikander. SEcure Neighbor Discovery (SEND). RFC 3971, Internet Engineering Task Force, March 2005.

[BMRS07] E. Baccelli, K. Mase, S. Ruffino, and S. Singh. Address autoconfiguration for MANET: Terminology and problem statement. Internet Draft "draft-ietf-autoconf-statement-01", Internet Engineering Task Force, August 2007.

[CFT05] G. Chelius, E. Fleury, and L. Toutain. No Administration Protocol (NAP) for IPv6 router auto-configuration. *Int. J. Internet Protocol Technology*, 1(2), 2005.

[DBV+03] R. Droms, J. Bound, B. Volz, T. Lemon, C. Perkins, and M. Carney. Dynamic host configuration protocol for IPv6 (DHCPv6). RFC 3315, Internet Engineering Task Force, July 2003.

[DH] A. Durand and C. Huitema. The host-density ratio for address assignment efficiency: An update on the h ratio. Technical report.

[dra] Draft: Ipv6 address allocation and assignment policy. Technical report, RIPE Community.

[EFL+99] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen. SPKI certificate theory. RFC 2693, Internet Engineering Task Force, September 1999.

[FH02] S. Farrell and R. Housley. An internet attribute certificate - profile for authorization. RFC 3281, Internet Engineering Task Force, April 2002.

[HH05] R. Hinden and B. Haberman. Unique Local IPv6 Unicast Addresses. RFC 4193, Internet Engineering Task Force, October 2005.

[HPFS02] R. Housley, W. Polk, W. Ford, and D. Solo. Internet X.509 Public Key Infrastructure - Certificate and Certificate Revocation List (CRL) Profile. RFC 3280, Internet Engineering Task Force, April 2002.

[II01] IAB and IESG. IAB/IESG recommendations on IPv6 address allocations to sites. RFC 3177, Internet Engineering Task Force, September 2001.

[Ler07] D. Leroy. A secure role-based address allocation and distribution mechanism - draft. Technical report, Université catholique de Louvain (UCL), September 2007. http://inl.info.ucl.ac.be/proto-addr-distrib.

[NNS98] T. Narten, E. Nordmark, and W. Simpson. Neighbor discovery for IP Version 6 (IPv6). RFC 2461, Internet Engineering Task Force, December 1998.

[rip06] Ipv6 address allocation and assignment policy. Technical Report “ripe-388”, APNIC, ARIN, RIPE NCC, September 2006.