Address-Swapping Scheme for On-Demand Assignment of Global Addresses in a TCP/IP Network

Reuven Cohen
HP Israel Science Center*

Keywords: TCP/IP, IP addressing, dynamic address allocation, community network, address swapping.

Abstract

The paper proposes a new scheme for on-demand allocation of global IP addresses to hosts of an autonomous network. Such a scheme is needed in order to overcome the problem of IP address exhaustion. According to the new scheme, each host of the autonomous network is assigned a fixed local address which appears in the source field of all the IP datagrams the host sends and in the destination field of all the IP datagrams it receives. A host that needs an IP-level connectivity with external hosts is allocated a global address for a limited time. Such a host continues using its fixed local address as the source field of every sent datagram. However, if the datagram is destined for an external host, the autonomous network border gateway swaps the fixed local address of the source with the leased global one. When an external host sends a datagram to a local host, it uses the global address of the local host as destination address. When the datagram enters the autonomous network, the leased global address is swapped by the fixed local one. As the paper shows, the proposed scheme that employs IP address swapping has several important advantages. Most of these advantages are due to the fact that the leased global addresses are not visible to the internal routing gateways, and that local hosts always use a single local address.

*Address: HP Israel Science Center, Technion City, Haifa 32000, Israel. Email: rcohen@hp.technion.ac.il

1 Introduction

With the worldwide proliferation of TCP/IP technology, CATV companies offer their customers connectivity to the Internet via local community data networks. The customers will be able to get data services from local servers, located inside the local community network, or from remote servers, located elsewhere on the Internet. However, as the Internet has evolved and grown over in recent years, it faces several serious scaling problems. One of them is the exhaustion of the class-B network IP address space and the eventual exhaustion of the entire IP address space. It becomes evident that after passing the trial phase, the operator of community TCP/IP networks will not be able to assign a unique IP address to every customer.

In order to alleviate the problem of IP address exhaustion, the Internet Assigned Numbers Authority (IANA) has reserved three blocks of IP address space for private networks. These addresses will be referred to as *local addresses,* in contrast to *global addresses* which are worldwide unique. An autonomous community network can assign to its hosts local addresses without any coordination with IANA. This address space can therefore be used by many autonomous networks. A host to which a local IP address has been assigned has a unique IP address within the autonomous network but not within the entire Internet. Thus, such a host may have IP-level connectivity with all other hosts inside the autonomous network, but not with external hosts.

In general, connecting local hosts to external hosts can be performed in several layers:

- In the Application layer, by means of an Application-level gateway (Figure 1(a)).
- In the Transport layer, by means of a Transport-level gateway (Figure 1(b)).
- In the IP level, by means of an IP-level gateway (a router)[1] (Figure 1(c)).

The most popular services, like e-mail, telnet, and ftp, can be provided to the hosts of an autonomous community network by means of Application-level gateways. Local hosts accessing an Application-level gateway can use their local address. Only the Application gateway needs an IP address which is unique within the entire Internet. The main drawback of using Application-level gateways is the necessity for a dedicated user interface or program for every provided service [2]. Thus, only few services are supported in this way.

Transport-level gateways (sometimes referred to as "Circuit-level gateways" [2, 6]) relay TCP connections. The source host of the application connects to a TCP port on the Transport-level gateway, which connects to some destination on the other side of the gateway. During the call, a relay program at the gateway copies bytes from one connection to the other. In order to employ Transport-level gateways, new programs should be executed at the clients. The best known strategy for making the necessary changes is the *socks* package [6], which consists of a set of replacements for various system calls like *socket, connect, bind* etc. In terms of addressing, Transport-level gateways are very similar to Application-level gateways, since in both cases the local host has a TCP connection with the gateway instead of with the external host, and therefore needs no global address.

[1] Throughout this paper, when the type of a gateway is not indicated, an IP-level gateway (router) is considered.

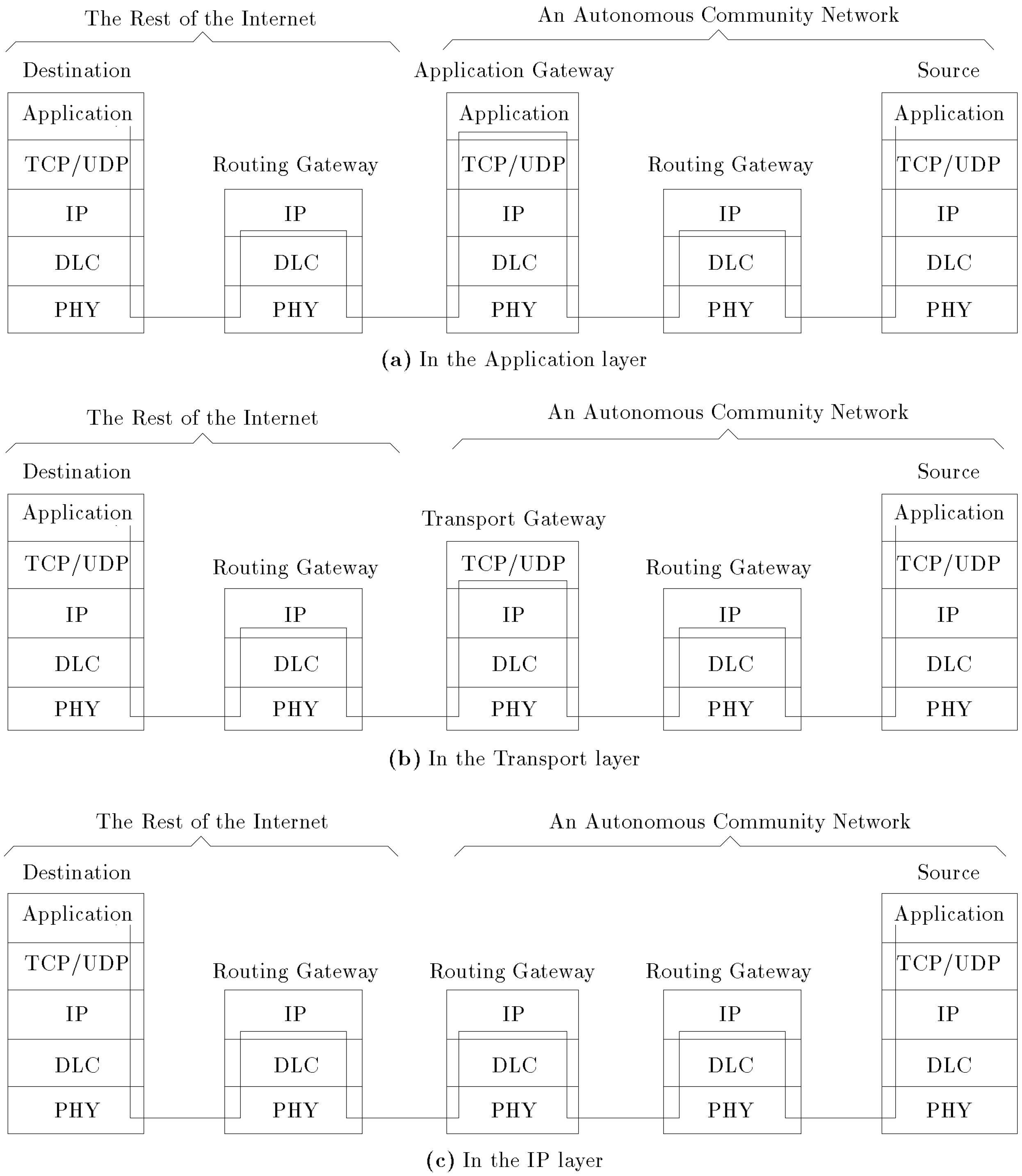
The only way to avoid making changes in user programs while allowing local hosts to communicate with external hosts is to employ IP-level gateways all the way from the source to the destination and vice versa. Such gateways enable the local hosts to have a Transport (TCP/UDP) connection with external hosts. IP-level connectivity with external hosts requires the local host to have a global IP address. This paper concentrates upon solutions that will enable local hosts to have IP-level connectivity with external hosts despite of the problem of IP address exhaustion.

Since not all the hosts of an autonomous network will be powered on at the same time, and only part of the powered-on hosts will need IP-level connectivity with external hosts, dynamic (on-demand) allocation of global IP addresses from a shared limited pool could solve the problem. Such a scheme is supported by the Dynamic Host Configuration Protocol (DHCP) [4]. Each host has a fixed local IP address which can be used for IP-level connectivity with other local hosts. A host that needs an IP-level connectivity with external hosts can acquire a global address from the shared pool for a limited duration. After the address is released, it can be re-assigned to another host.

This paper illuminates the potential drawbacks of such on-demand allocation of global addresses. Most of the drawbacks are due to the fact that local hosts may have different addresses at different times. This complicates the operation of the network, and may cause wrong delivery of IP datagrams. Then, the paper presents a unique scheme that eliminates the potential drawbacks. *The new scheme is based on IP address swapping performed by the border gateways of the autonomous network.* According to the new scheme an IP datagram originated at a local host will have in its SOURCE_IP_ADDRESS field the fixed local address of the host as long as it travels inside the autonomous network. If the datagram is destined for an external host, a border gateway will swap the fixed local address of the host with its temporary global address. Similarly, an IP datagram sent by an external host to a local one will carry the global address of the destination as long as it travels outside the autonomous network. Upon entering the autonomous network, the global address of the destination host will be swapped by the border gateway with the fixed local address. Due to this scheme, the autonomous network routers and name-servers will not have to deal with the various global addresses assigned to local hosts. In addition, the assignment of global addresses to local hosts is facilitated, because every local host can be assigned every global address regardless of the location of the host in the autonomous network. As the paper shows, this reduces the probability that a local host that needs an IP-level connectivity with external hosts will be blocked due to temporary lack of global addresses.

Circuit-switched networks like ATM uses "label swapping" as a routing strategy [7]. Label swapping is needed in these networks in order to allow every node along a circuit to determine the label carried by the packets it receives over the circuit. This simplifies the set-up of circuits and the management of the routing tables. The present paper is probably the first to show that address swapping is sometimes useful in packet-switched networks as well.

The rest of this paper is organized as follows. Section 2 deals with the distinction between local and global addresses. Section 3 shows how global addresses can be dynamically assigned to local hosts that need IP-level connectivity with external hosts. This section also illuminates the various problems arising due to such on-demand allocation. Section 4 presents the new scheme, which is based on IP-address swapping performed by the border gateways of the autonomous networks. This section explains how the new scheme eliminates the problems described in Section 3. Section 5 deals with recovery issues. Section 6 shows that the new scheme significantly reduces the probability that a local host that needs an IP-level



(a) In the Application layer

(b) In the Transport layer

(c) In the IP layer

Figure 1: Connecting the Community Network to the Internet in Various Layers

connectivity with external hosts will be blocked due to temporary lack of global addresses. Section 7 concludes the paper.

2 Fixed Assignment of Local and Global IP Addresses

A host H of an autonomous community network $\mathcal{N}$ will most of the time be connected to local hosts and servers. During this time, the host can use a local address, which is unique only within $\mathcal{N}$. Such an address cannot be used when the host needs to send IP datagrams to external hosts outside $\mathcal{N}$.

The Internet Assigned Numbers Authority (IANA) has reserved 3 blocks of local IP addresses for private autonomous networks [9]:

(a) 10.0.0.0 – 10.255.255.255 (a single class-A network number);

(b) 172.16.0.0 – 172.31.255.255 (16 contiguous class-B network numbers);

(c) 192.168.0.0 – 192.168.255.255 (255 contiguous class-C network numbers).

Let $\mathcal{A}_L$ represents the union of these three address spaces and $\mathcal{A}_G$ represents the union of the remaining class-A, B and C IP addresses. An autonomous network can use addresses from $\mathcal{A}_L$ without any coordination with IANA. Thus, each address in $\mathcal{A}_L$ can be assigned to many hosts in different networks. Since no address from $\mathcal{A}_L$ will be assigned by IANA to hosts that need a global IP address, a border gateway (connecting an autonomous private network to the Internet) G that receives on a local port a datagram whose DESTINATION_IP_ADDRESS field contains an address from $\mathcal{A}_L$ knows that the datagram is destined for a local host.

An approach that uses the address space $\mathcal{A}_L$ in order to overcome the exhaustion of addresses in $\mathcal{A}_G$ is described in [9]. Consider a private network $\mathcal{N}$, and let $\mathcal{A}_{G'} \subset \mathcal{A}_G$ be the set of global IP addresses assigned by IANA to $\mathcal{N}$. According to this approach, the authority of $\mathcal{N}$ should determine which hosts need and which hosts do not need network layer connectivity with external hosts. Hosts that need or may need IP-level connectivity with external hosts are assigned global addresses from $\mathcal{A}_{G'}$. The other hosts, that do not need IP-level connectivity with external hosts, are assigned local addresses from $\mathcal{A}_L$. Consequently, every host with a local address from $\mathcal{A}_L$ will be able to send IP datagrams only to hosts in $\mathcal{N}$, whereas every local host with a global address from $\mathcal{A}_{G'}$ will be able to send IP datagrams to hosts inside and outside $\mathcal{N}$. This scheme, where each host has either a global address *or* a local address will be referred to as the *fixed global address or fixed local address* scheme.

The *fixed global address or fixed local address* scheme has two major drawbacks. The first drawback is that each host in the autonomous network $\mathcal{N}$ should be classified in advance as either a local host or a global host. Thus, no more than $|\mathcal{A}_{G'}|$ hosts of $\mathcal{N}$ may ever have IP-level connectivity with external hosts. The second drawback is due to the fact that TCP/IP uses hierarchical routing which relies on hierarchical addressing. Routing is based on the destination host address, as appears in the DESTINATION_IP_ADDRESS field of each IP datagram. The address has two parts: net-id and host-id. Using the net-id portion of the address, the Internet gateways know the network $\mathcal{N}$ to which the destination is connected, and forward the IP datagram to that network. Within the destination network $\mathcal{N}$, a similar approach, called *sub-net routing,* is used in order to deliver the datagram to the destination host H. To this end, the host-id portion of the destination host IP address is sub-divided into two parts: sub-net_id and host_id. By interpreting the sub-net_id portion, the internal gateways of $\mathcal{N}$ deliver the datagram to the sub-net of the destination host H. This approach, where the host-id is divided into net-id and host-id, can be repeated several times, until a gateway with a direct physical connection to H forward the datagram to H. Since this routing approach is based on hierarchical addressing, mixing local addresses from $\mathcal{A}_L$ with global addresses from $\mathcal{A}_G$ in a single private network $\mathcal{N}$ would result in big routing tables at the internal gateways. These gateways will need to store an individual route for each individual host with a global address. This problem can be reduced if hosts in the same sub-net are assigned global addresses with the same prefix. In such a case, however, the address space of $\mathcal{A}_L$ will have to be much larger than the number of local hosts to which a global address should be assigned.

3 On-Demand Assignment of Global IP Addresses

In order to overcome the first drawback of the *fixed global address or fixed local address* scheme, hosts can be assigned global addresses only when they need IP-level connectivity with external hosts. Such an approach has been suggested in the context of the Dynamic Host Configuration Protocol (DHCP) [4], which is an extension of the TCP/IP Bootstrap Protocol (BOOTP) [3]. According to this approach, each host is assigned a fixed local address from $\mathcal{A}_L$. Upon bootstrapping, the host determines whether it will need a global address from $\mathcal{A}_G$, or it can manage with its local address. If the local address is sufficient, it can be found in the local disk, or can be acquired from a remote server using RARP, BOOTP, or DHCP. If, however, the host needs a global address in order to establish an IP connection with external hosts, a request DHCP message is sent by the host to a DHCP server. The server maintains a pool $\mathcal{A}_{G'} \subset \mathcal{A}_G$ of global addresses which has been assigned by IANA to $\mathcal{N}$. It selects an available address from $\mathcal{A}_{G'}$ to be allocated to H, and sends a response DHCP message that informs H of the allocated address. The global address is allocated to H for some pre-determined period of time, referred to as "lease" [4]. A host can extend its lease by sending the server another request message.

This scheme, where each host has either a leased global address or a fixed local address will be referred to as the *leased global address or fixed local address* approach. The scheme eliminates the first drawback associated with the *fixed global address or fixed local address* approach, because using the same global address space $\mathcal{A}_{G'}$, the network can accommodate at any time any $|\mathcal{A}_{G'}|$ local hosts that need IP-level connectivity with external hosts. In particular, there is no need to determine in advance the group of $|\mathcal{A}_{G'}|$ local hosts to which global addresses will be assigned.

On the other hand, the scheme complicates the management and administration of the autonomous private network. The most significant issue is that the routing tables of the internal gateways must be refreshed very often, whenever a global address is allocated or released. Every internal gateway must have in its routing table an entry for every allocated global address. When a global address a is allocated to a host H' in sub-net $\mathcal{N}'$ of the private network $\mathcal{N}$, all the gateways have to be informed that datagrams for IP address a should be forwarded to sub-net $\mathcal{N}'$. When a is released, all these routing table entries should be eliminated. This implies that in addition to routing information exchanged by the gateways in order to determine the best route(s) to every sub-net $\mathcal{N}'$, gateways should exchange information with other gateways, with hosts, and with DHCP servers, in order to know the latest assignment of each global address.

This problem can be reduced by dividing the set $\mathcal{A}_{G'}$ of dynamic global addresses into S mutually disjoint sub-sets, ${\mathcal{A}_{G'}}^1, {\mathcal{A}_{G'}}^2 \cdots {\mathcal{A}_{G'}}^S$, where S is the number of sub-nets in $\mathcal{N}$, and by allocating addresses from every sub-set ${\mathcal{A}_{G'}}^i$ to hosts in sub-net $\mathcal{N}^i$ only. Consequently, hosts using the same global address a (at different time) will be connected to the same sub-net. Thus, internal gateways will have to be informed only once about the binding between sub-sets of $\mathcal{A}_{G'}$ and sub-nets of $\mathcal{N}$. However, as shown in Section 6, such an approach may significantly increase the blocking probability of global address allocation, where 'blocking' is defined as the case where a host in sub-net $\mathcal{N}^i$ needs a global address while all the addresses in ${\mathcal{A}_{G'}}^i$ are allocated.

Other drawbacks of the *leased global address or fixed local address* approach are as follows:

- The same problem mentioned above regarding the internal gateways holds for the internal Domain Name System (DNS) server(s) as well. Each time a host H acquires a global address, every DNS server must be informed and updated. The DNS servers must also be informed about the expiration time of every lease in order to update the Time-To-Live (TTL) value associated with the relevant record.

- The local hosts cannot be addressed using a fixed address independent of the global address allocation mechanism, to which local servers can apply (e.g. in order to fix problems related to the global address allocation). This problem can be solved if each host is required to accept datagrams that carry in their DESTINATION_IP_ADDRESS either its fixed local address or its leased global one. However, multi-homing is known to introduce considerable confusion and complexity into the protocol suit [5].

- IP datagrams can be misrouted due to an IP-to-MAC address binding which is no more correct. For instance, suppose that a global address a has been assigned to host H in sub-net $\mathcal{N}^i$. Suppose that gateway G needs to forward a datagram to H. Gateway G uses ARP in order to translate the IP address a of H to its constant MAC address $\mu(H)$. Suppose that sometime later H releases the global IP address a, and the latter is allocated to another host is subnet $\mathcal{N}^i$, host H' say. If G has a datagram to forward to host H', and its cache still contains the binding between a and $\mu(H)$, the datagram will be forwarded to H instead of to H'.

- There might be cases where due to some error the same global address is employed by two or even more hosts. The recovery from such cases might be difficult.

4 The New Scheme of IP Address Swapping

This section presents a new approach that allows on-demand allocation of global IP address, while overcoming the drawbacks of the approach presented in the previous section. In particular, internal IP gateways and DNS servers will not have to be updated whenever a global address is allocated or release. Moreover, there is no need to pre-divide the set $\mathcal{A}_{G'}$ of global addresses assigned to the autonomous network $\mathcal{N}$ into sub-sets and to allow only hosts in sub-net $\mathcal{N}^i$ to acquire an address from ${\mathcal{A}_{G'}}^i$. Rather, every address in $\mathcal{A}_{G'}$ can be assigned to every host in $\mathcal{N}$. Consequently, the blocking probability, as previously defined, is significantly reduced.

In the new scheme, every host H of the autonomous network $\mathcal{N}$ has a fixed local address from $\mathcal{A}_L$, which depends on the location of H in $\mathcal{N}$ in order to allow an efficient hierarchical sub-net routing. A host H that needs a global IP address, in addition to the fixed local one, sends a request DHCP message to a DHCP server. The server maintains the pool $\mathcal{A}_{G'} \subset \mathcal{A}_G$ of the global addresses assigned by IANA to $\mathcal{N}$. Regardless of the sub-net of H in $\mathcal{N}$, any unused global address in $\mathcal{A}_{G'}$ can be allocated to H.

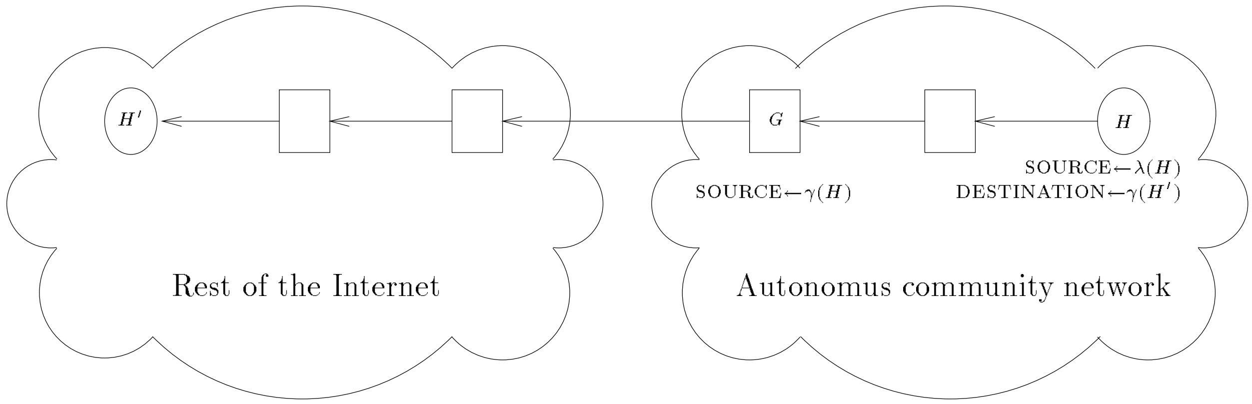
Though a single DNS server can be used in order to allow mapping of domain names to local or global addresses, the scheme works better when two kinds of servers are used. An *external* DNS server should resolve queries received from external hosts, whereas a set (whose exact cardinality depends upon the size of the autonomous network) of *local* DNS servers should resolve queries received from internal hosts. Since the local addresses are fixed, the information kept in the internal DNS server rarely changes. The external DNS server, in contrary, must be informed of any lease of a global address.

Let $\mathcal{G}$ be the set of border gateways connecting the autonomous network $\mathcal{N}$ to the Internet. When a global address a is leased to host H, the DHCP server should inform all the gateways in $\mathcal{G}$ about the local address of H, the assigned global address, and the duration of the lease. In addition, the DHCP server should inform the external DNS server about the name of the host, the leased global address and the lease duration. The simple case is when $\mathcal{G}$ includes a single gateway only, and the latter performs also the functionalities of the DHCP server and of the external DNS server. In such a case no update message should be sent.

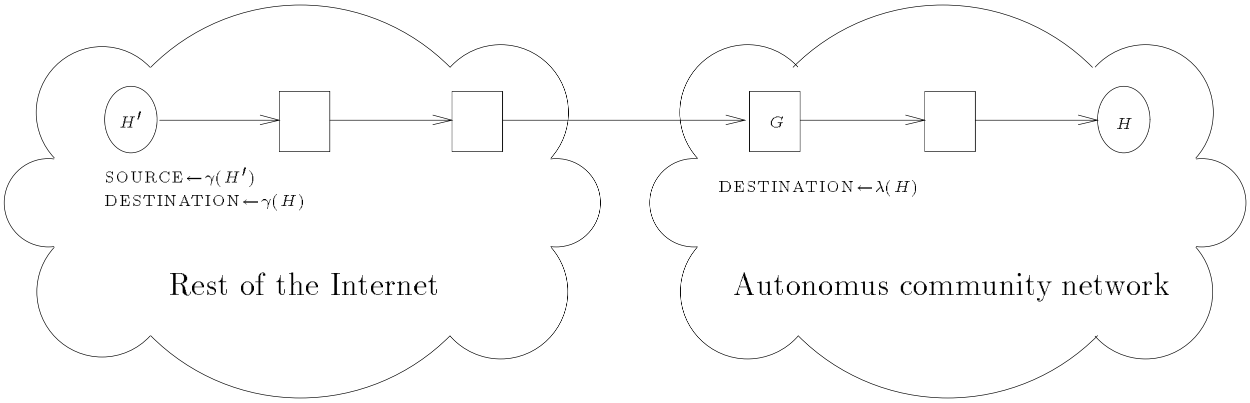
When a local host H sends an IP datagram to another local hosts H', host H uses its local fixed address $\lambda(H)$ in the SOURCE_IP_ADDRESS field of the datagram header, even if it has a leased global address $\gamma(H)$ as well. Similarly, H writes the local address $\lambda(H')$ of H' in the DESTINATION_IP_ADDRESS field, whether or not H' has a global address $\gamma(H')$ as well. In fact, H should not be aware of the leased global address of H' since it approaches only the local DNS server(s) in order to map the name H' into an IP address.

Suppose now that H needs to send an IP datagram to an external host H' whose global address is $\gamma(H')$. To this end, H has leased a global address $\gamma(H)$ in advance. Host H writes in the DESTINATION_IP_ADDRESS field of the IP datagram the global adders $\gamma(H')$ of H'. However, in the SOURCE_IP_ADDRESS field it does not write the global address $\gamma(H)$ of itself, but the local one $\lambda(H)$. The internal gateways of the autonomous network $\mathcal{N}$ deduce from the DESTINATION_IP_ADDRESS field that the datagram is destined for an external host. Thus, they route the datagram to one of the border gateways in $\mathcal{G}$, which connect the private network to the Internet. The gateway from $\mathcal{G}$ that receives the datagram uses the local address of H, as appears in the SOURCE_IP_ADDRESS field, as a key to a table which contains a list of all the global leased addresses. The gateway G deduces that H's global address is $\gamma(H)$. Then, it changes the SOURCE_IP_ADDRESS field from $\lambda(H)$ to $\gamma(H)$ and forwards the datagram to an external gateway. Consequently, all the external gateways on the route from $\mathcal{N}$ to H', as well as the destination host H', are not aware of the local address of H, but only of its global address. The local gateways, in contrast, are aware of the fixed local address only. The whole process is described in Figure 2(a).

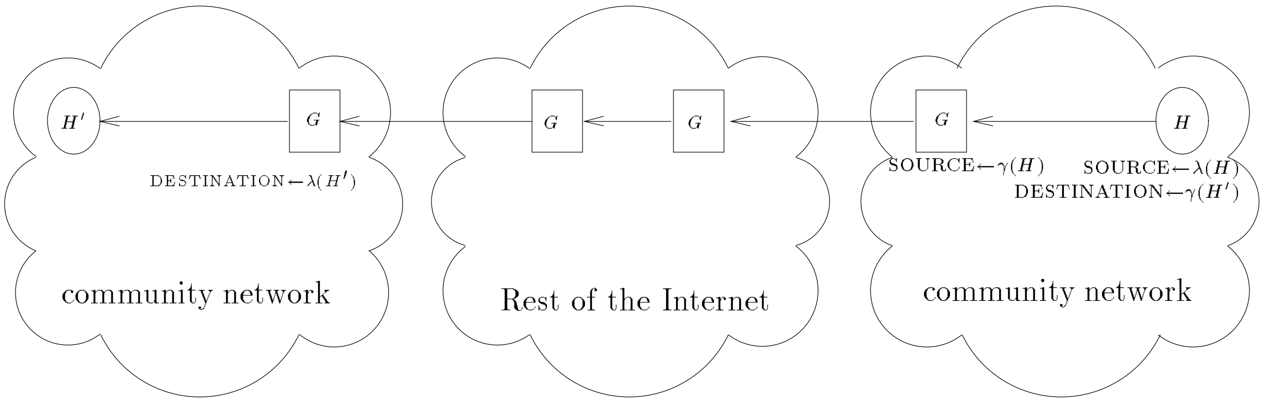
A similar process, in reverse order, is performed when H' sends an IP datagram to H (Figure 2(b)). The datagram created by H' contains the global address $\gamma(H)$ of H in the DESTINATION_IP_ADDRESS field. The Internet gateways deduce from $\gamma(H)$ that the destination is in $\mathcal{N}$, and route the datagram accordingly. Upon entering $\mathcal{N}$, the datagram is received by a gateway in $\mathcal{G}$, which changes the DESTINATION_IP_ADDRESS field from



(b) Routing of an IP datagram from the autonomous community network to an external host



(b) Routing of an IP datagram from an external host into the autonomous network



(c) Routing of an IP datagram between two autonomous networks

Figure 2: IP Address Swapping

$\gamma(H)$ to $\lambda(H)$. Then, the datagram is routed by the internal gateways of $\mathcal{N}$ until reaching H.

Figure 2(c) shows the case where H sends an IP datagram to H' while each of them resides in a different autonomous network. In such a case, address swapping is performed twice. When the IP datagram leaves the source network the SOURCE_IP_ADDRESS field changes from $\lambda(H)$ to $\gamma(H)$. When the datagram enters the destination network, the DESTINATION_IP_ADDRESS address field changes from $\gamma(H')$ to $\lambda(H')$.

The new scheme takes advantage of an important feature of the Domain Name System: the use of a Time-To-Live (TTL) field that specifies the time interval that a given name-to-address binding may be cached before the source of the information should again be consulted. Using the TTL, it is possible to make sure that a global address will not be associated with a host after the expiration of the lease. When a global address is allocated to a local host H, the DNS server of the autonomous network to which external hosts and servers apply in order to get the IP addresses associated with names of local hosts, is informed. The DNS server is also informed about the lease duration. It creates a new Resource Record (RR), which contains the name of the host H and the assigned global address $\gamma(H)$. The TTL value associated with the new record is determined to be less than the lease duration. When information about this record is obtained and cashed by DNS servers outside the autonomous network, the TTL value is attached. Thus, no external host will use the information of the entry after the lease is expired.

The proposed scheme, which is based on address swapping performed by the gateway(s) connecting the autonomous community network to the rest of the Internet, eliminates all the drawbacks related to the *leased global address or fixed local address* approach. First, any available address in $\mathcal{A}_{G'}$ can be allocated to any host H in $\mathcal{N}$ regardless of the sub-net $\mathcal{N}'$ in $\mathcal{N}$ to which H belongs. Nevertheless, routing information of internal gateways do not have to change each time a global address is allocated or released. This is simply because the leased addresses are not used within the autonomous network. For similar reason there is no need to update the internal DNS server(s) when a global address is allocated or released. The only nodes that have to be informed about any allocation and release of global addresses are the border gateways $\mathcal{G}$ and the external DNS server. However, this group is considerably smaller than the group of internal gateways and DNS servers. In fact, in most community networks the group $\mathcal{G}$ will consist of a single gateway which will function also as an external DNS server and will be in charge, as a DHCP server, of assigning on-demand global addresses to local hosts. In such a case, *no update message will have to be sent whenever a lease of an address starts or expires.*

In the new scheme every local host H has a constant local address $\lambda(H)$ which is independent of the global address allocation mechanism. All the packets destined for H have $\lambda(H)$ in their DESTINATION_IP_ADDRESS field, regardless of the time and the source station location. Thus, multi-homing is not needed. Similarly, all the IP datagrams originated at H have $\lambda(H)$ in their SOURCE_IP_ADDRESS field, regardless of the time and the destination station location. This implies that the communication among local hosts is not affected by the allocation of global addresses. In particular, all the local hosts are accessible by other local hosts even if the DHCP server or some DNS server fails. Consequently, the maintenance of the autonomous network is facilitated.

Each internal gateway may have in its cache the binding between the fixed local address $\lambda(H)$ and the fixed MAC address $\mu(H)$ of every host H in the local sub-net. This information remains valid regardless of allocations and releases of global addresses. Thus, mis-routing due to incorrect IP-to-MAC address bindings is avoided.

The new proposed scheme also eliminates potential problems caused due to the use of the same global address by more than a single local host. This issue is discussed in the next section.

5 Checksum Related Issues

Handling the IP and the TCP/UDP Checksums

An IP datagram has in its header a 16-bit HEADER_CHECKSUM field to ensure integrity of header values. The checksum applies only to the IP header and not to the data. The IP layer of the source station calculates the checksum and sets the HEADER_CHECKSUM field accordingly. In the proposed scheme, a border gateway that performs address swapping should re-calculate the checksum and update the HEADER_CHECKSUM field. However, this does not put new processing burden on the border gateways since regardless of the proposed scheme gateways should update the Time-To-Live (TTL) field in the IP datagram header and re-calculate the IP header checksum. If a border gateway is not supposed to perform any change in the IP header except address swapping, it can make an incremental update of the checksum without scanning the entire header [1, 8]. To this end, the gateway can pre-compute and store the difference $\gamma(H) - \lambda(H)$ between the allocated global address and the fixed local address of every local host H to which a local address $\gamma(H)$ has been allocated. When a datagram is received, the new HEADER_CHECKSUM field can be computed according to the received checksum and the stored value of $\gamma(H) - \lambda(H)$.

In addition to the HEADER_CHECKSUM field in the IP layer datagram, both UDP and TCP have in their datagram header a 16-bit CHECKSUM field to verify the integrity of the entire Transport layer datagram. The checksum is optional in UDP datagrams and mandatory in TCP datagrams. It covers not only the datagram header and data, but also a *pseudo header* which is not a part of the datagram [3]. The pseudo header is shown in Figure 3. It contains the IP address of the source, the IP address of the destination, the IP protocol type code (17 for UDP, 6 for TCP) and the length of the entire UDP/TCP datagram. This pseudo header is used during checksum computation in order to let the receiving host to make sure that the datagram has reached the correct destination. Upon receiving a datagram, the software verifies the checksum using the sender IP address and the local IP address, as obtained from the IP layer.

The use of a pseudo header that contains IP layer information during the computation of a Transport layer checksum violates basic layering rules [3]. As shown in the following, this violation imposes some difficulty on the proposed IP address swapping approach, but on the other hand facilitates the recovery from wrong use of global addresses.

To understand the difficulty caused by the pseudo header, consider a host H in an autonomous community network $\mathcal{N}$, which sends a TCP or UDP datagram to an external host H'. As explained in Section 4, H writes in the SOURCE_IP_ADDRESS and DESTINATION_IP_ADDRESS fields of the IP datagram the local address of itself $\lambda(H)$ and the global address of H' $\gamma(H')$ respectively. However, due to the address swapping the destination host H' receives the packet with $\gamma(H)$ instead of $\lambda(H)$ in the SOURCE_IP_ADDRESS field. Obviously, if H computes the TCP/UDP checksum using $\lambda(H)$ as the SOURCE_IP_ADDRESS of

SOURCE_IP_ADDRESS		
DESTINATION_IP_ADDRESS		
ZERO	PROTO	LENGTH

Figure 3: The 12-byte Pseudo-Header Used for UDP/TCP Checksum Computation

the pseudo header, whereas H' computes the checksum using $\gamma(H)$ as the SOURCE_IP_ADDRESS of the pseudo header, the datagram will be rejected by H'.

The problem is avoided if the sender H uses its global address $\gamma(H)$ instead of its local address $\lambda(H)$ during checksum calculation upon sending a Transport layer (TCP/UDP) datagram *to an external host.* Assuming that each local host knows the set of local addresses $\mathcal{A}_L$ used by the local community network, host H can determine the SOURCE_IP_ADDRESS to be used during the checksum calculation in the following way:

– if DESTINATION_IP_ADDRESS is in $\mathcal{A}_L$ use $\lambda(H)$, else use $\gamma(H)$

Similarly, when a local host H receives a UDP/TCP datagram, it needs to determine whether to use $\lambda(H)$ or $\gamma(H)$ as the DESTINATION_IP_ADDRESS of the pseudo header, during checksum calculation. The decision is made according to the SOURCE_IP_ADDRESS in the following way:

– if SOURCE_IP_ADDRESS is in $\mathcal{A}_L$ use $\lambda(H)$, else use $\gamma(H)$

Except for checksum calculation, an internal host H does not need to be aware of its global address $\gamma(H)$. All the IP datagrams it receives contain its local address $\lambda(H)$ in their DESTINATION_IP_ADDRESS field, regardless of the sender location. Similarly, all the IP datagrams it sends contains in their SOURCE_IP_ADDRESS its local address $\lambda(H)$, regardless of the destination location.

Misrouting due to Allocation Error

As already mentioned, another advantage of the new proposed scheme is that it eliminates potential problems caused due to the dynamic assignment of global addresses. To simplify the following discussion, it is assumed that the group of border gateways $\mathcal{G}$ in the considered autonomous network $\mathcal{N}$ consists of a single gateway which functions also as an external DNS server and as a DHCP server. The concerned issues are as follows:

1. A global address a from $\mathcal{A}_{G'}$ was allocated to host H. The lease has expired, and the address has been re-allocated to another local host H'. However, due to some error host H continues viewing a as its global address.

2. A global address a was assigned to host H. The lease has not expired yet. However, due to some error host H views another global address a' as its global address.

3. An external host H' sends to a local host H a datagram using a global address a which is no more used by H.

Note that under normal operation none of the above cases is possible. The DHCP protocol is supposed to allocate only available global address to any host that needs IP-level connectivity with external hosts. Each global address is allocated for a limited duration, during which it cannot be assigned to any other host. After the lease expires, the host is not supposed to use its old global address unless it has requested to extend the lease using another DHCP request message [4]. Thus, the first two cases are not supposed to take place. As explained earlier, the third case is not supposed to take place due to the association of an appropriate TTL value with every Resource Record that contains the name-to-address mapping of the autonomous network hosts at the DNS server.

Consider the first possible failure, where host H uses an old global address a while it has no global address. Note that as long as H sends IP datagrams to local hosts there will be no confusion, since global addresses are not involved. However, when H sends an IP datagram to an external host, the datagram will be received by the border gateway. The later will realize that H has no global address, drop the datagram, and send an appropriate ICMP message back to H.

In the second case, where H views a instead of a' as its global address, the datagram cannot be dropped by the border gateway. Rather, it will be received by the external destination host H' with SOURCE_IP_ADDRESS=a'. However, since the SOURCE_IP_ADDRESS field in the pseudo header created by H was a whereas the SOURCE_IP_ADDRESS field in the IP datagram header is a', the destination will drop the TCP/UDP datagram due to wrong checksum. An interesting point here is that H' can send an ICMP message back to H, which will not be dropped by H. This is because the checksum of ICMP messages is not a function of the pseudo header.

Next, consider the third case where an external host H' sends to a local host H a datagram using a global address a which is no more used by H. If a has not been allocated to another local host, the IP packet will be discarded by a gateway in $\mathcal{G}$ upon entering the autonomous network, and an appropriate ICMP (e.g. "destination unreachable") will be sent back to the sender. If a has already been allocated to another local host H'', the datagram will be delivered to that wrong host, and the error will not be detected. It is therefore important that the autonomous network will advertise the allocated global addresses of local hosts with correct TTL values, and that the DHCP server will always allocate the least recently used address.

6 Blocking Probability

As already mentioned, one of the advantages of the the new IP address swapping scheme over the *fixed global address or fixed local address* scheme presented in Section 3 is that the new scheme enables to reduce the probability that a local host that needs an IP-level connectivity with external hosts will be blocked due to a lack of an available global IP address. This property is proven in the following.

Consider an autonomous network $\mathcal{N}$, where the *fixed global address or fixed local address* scheme (i.e. no address swapping) is used. In order to reduce the size of the routing tables at the internal gateways and the need to inform every local gateway about every allocation or release of a global address, network $\mathcal{N}$ is divided into S sub-nets $\mathcal{N}^1, \mathcal{N}^2 \cdots \mathcal{N}^S$. The set $\mathcal{A}_{G'}$ of global addresses assigned to $\mathcal{N}$ is also divided into S mutually disjoint sub-sets, ${\mathcal{A}_{G'}}^1, {\mathcal{A}_{G'}}^2 \cdots {\mathcal{A}_{G'}}^S$. Stations from every sub-net $\mathcal{N}^i$ can be allocated global addresses from ${\mathcal{A}_{G'}}^i$ only. In such a case, a host H from $\mathcal{N}^i$ is blocked if it needs a global address while all the addresses of ${\mathcal{A}_{G'}}^i$ are allocated to other hosts in $\mathcal{N}^i$. In the new address swapping scheme, in contrary, there is no need to divide the set $\mathcal{A}_{G'}$ into sub-sets. Rather, every host in $\mathcal{N}$, regardless of its sub-net $\mathcal{N}^i$, can be allocated any available address in $\mathcal{A}_{G'}$. Thus, a host will be blocked only if all the addresses in $\mathcal{A}_{G'}$ are occupied.

To analyze the blocking probability in each of these two schemes, we model the dynamic assignment of global addresses as an m-server loss system. Each of the $|\mathcal{A}_{G'}|$ global addresses assigned to the autonomous network is considered as a server, and each request for a global address is considered as a customer of the system. Assuming that the requests arrive according to a Poisson process with rate λ, that the average duration of each request is $1/\mu$, and that requests that cannot be satisfied due to the exhaustion of the available addresses are lost (rather than queued) we get an M/G/m/m system. In an M/G/m/m system, the probability that an arrival will find all m servers busy and will therefore be lost is given by the following Erlang B formula:

$$Prob(\text{blocking}) = \frac{\frac{(\lambda/\mu)^m}{m!}}{\sum_{n=0}^{m} \frac{(\lambda/\mu)^n}{n!}} \tag{1}$$

Suppose that $\mathcal{N}$ is allocated one set of Class C addresses. Thus, $m = |\mathcal{A}_{G'}| = 256$. Figure 4 shows the blocking probability for various loads for the case where global addresses are allocated from a shared pool and for the cases where several pools are held, one for each sub-net. The *load* is defined as λ/μ. The figure shows the graphs for loads ranging between $m/4$ and $2m$. For heavier loads ($\lambda/\mu > 2m$), the blocking probability is larger than 0.5 even if all the 256 addresses are considered as a single shared pool. Thus, for such loads the new scheme does not help in getting reasonable blocking probability. For smaller loads ($\lambda/\mu < m/4$), on the other hand, the blocking probability is very small ($< 0.3 * 10^{-5}$) even if the pool of global addresses is divided into 16 sub-sets, and the addresses of every sub-set ${\mathcal{A}_{G'}}^i$ can be allocated only to the stations of sub-net of $\mathcal{N}^i$. Thus, for such loads the new approach is not needed in order to achieve reasonable blocking probability. Note, however, that all the other advantages related to the new scheme, as discussed in Section 4, apply regardless of the load.

The table in Figure 5 outlines the blocking probability with and without address swapping for the case where the autonomous network is divided into 16 sub-nets. As the table shows, for a load of $\lambda/\mu = 2m$, address swapping reduces blocking probability from 0.525 (the blocing probability for $m = 16$) to 0.501 (blocing probability for $m = 256$). However, as the load decreases, the advantage of address swapping substantially increases: for $\lambda/\mu = 3m/4$,

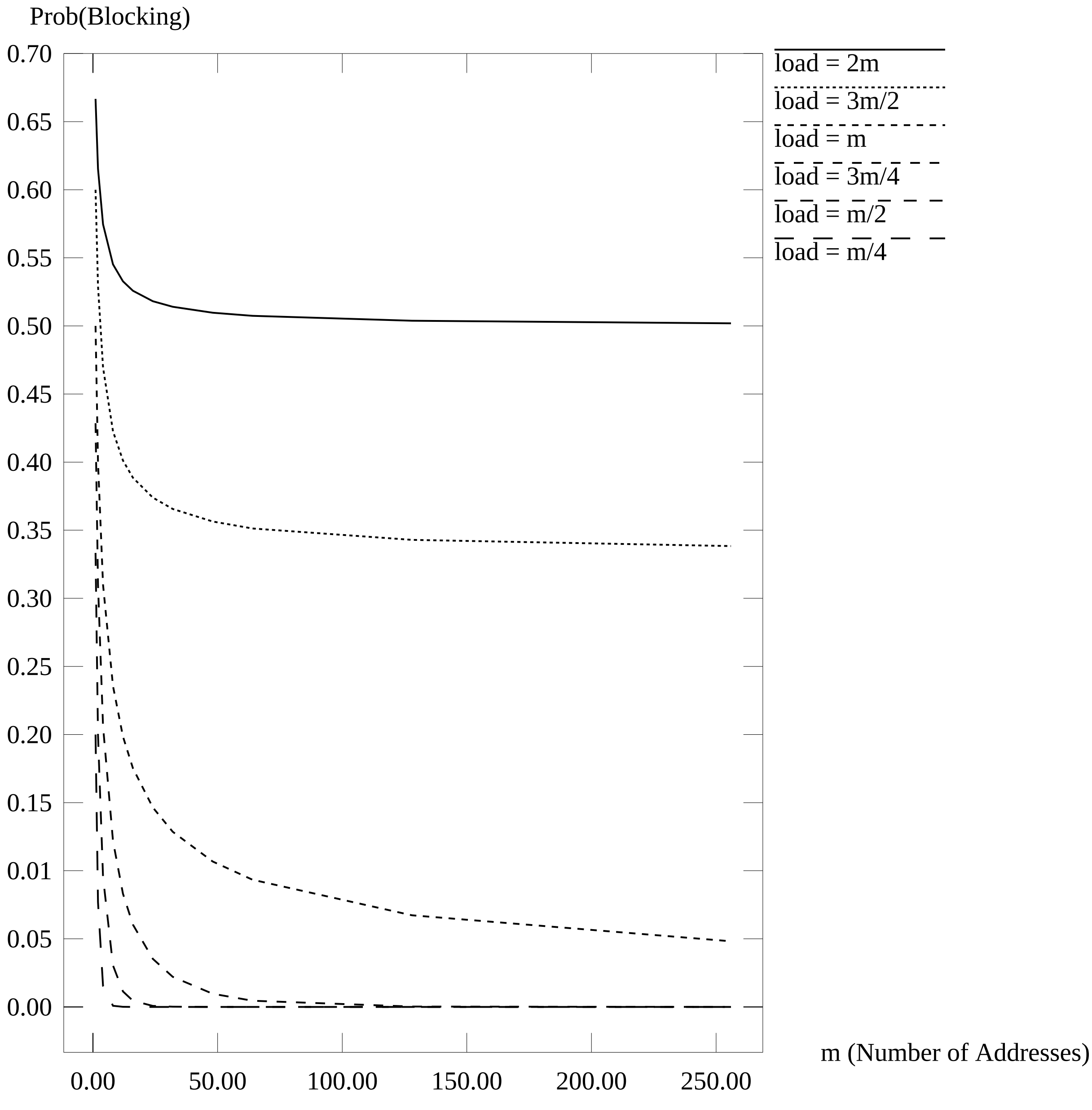


Figure 4: Blocking Probability Vs. Address Pool Size for Various Loads

load	Blocking Probability	
(λ/μ)	without address swapping	with address swapping
2m	0.525	0.501
3m/2	0.388	0.338
m	0.175	0.048
3m/4	0.060	$0.161 \cdot 10^{-5}$
m/2	0.004	$0.836 \cdot 10^{-23}$

Figure 5: Blocking Probability for 16 Sub-Nets

the blocking probability is reduced from 0.06 to 0.16·10−5, and for $\lambda/\mu = m/2$ it is reduced from 0.004 to $0.836 \cdot 10^{-23}$.

7 Conclusions

The paper proposed a new scheme for on-demand allocation of global IP addresses to hosts of an autonomous network. According to the new scheme, each host of the autonomous network is assigned a fixed local address. A Host that needs an IP-level connectivity with external hosts uses the Dynamic Host Configuration Protocol in order to lease a global address. All the IP datagrams traveling inside the autonomous network carry only the local fixed addresses of the autonomous network hosts. If an IP datagram is sent by a local host to an external one, the fixed local address of the local host is swapped by the leased global address at the autonomous network border gateway. If an IP datagram is sent by an external node to an internal one, the datagram carries in the DESTINATION_IP_ADDRESS field the leased global address of the destination host. Upon entering the autonomous network, the leased global address is swapped by the fixed local address of the local destination host. Circuit-switched networks uses "label swapping" in order to simplify the set-up of circuits and the management of the routing tables. The present paper is probably the first to show that address swapping can be useful for packet-switched networks as well.

The new address swapping scheme has several advantages in the management of an autonomous community network that use dynamic assignment of global addresses. Firstly, it allows to allocate any available global address to any host regardless of the sub-net to which that host belongs, without requiring the internal gateways to change their routing tables each time a global address is allocated or released. Secondly, there is no need to update the internal DNS server(s) whenever a global address is allocated or released. Thirdly, the scheme allows the local hosts to use their fixed local address only. In addition, the scheme avoids mis-routing due to incorrect IP-to-MAC address bindings. And finally, the scheme eliminates potential problems caused due to wrong use of global addresses by local hosts.

8 References

[1] R. Barden, D. Borman and C. Partridge. Computing the Internet checksum. RFC-1071, September 1988.

[2] W. Cheswick and S. Bellovin. *Firewalls and Internet security.* Addison-Wesley, 1994.

[3] D. Comer. *Internetworking With TCP/IP.* Prentice Hall, 1991.

[4] R. Droms. Dynamic host configuration protocol. RFC-1541, October 1993.

[5] Internet Engineering Task Force (R. Braden, Editor), Requirements for Internet Hosts – Communication Layers. RFC-1122, October 1989.

[6] D. Koblas and M. Koblas. Socks. In *UNIX Security III Symposium*, September 1992.

[7] J. Le Boudec. The asynchronous transfer mode: A tutorial. *Computer Networks and ISDN Systems*, 24:279–309, 1992.

[8] T. Mallory and A. Kullberg. Incremental Updating of the Internet checksum. RFC-1141, January 1990.

[9] Y. Rekhter, B. Moskowitz, D. Karrenberg, and G. de Groot. Address allocation for private internets. RFC-1597, March 1994.