IP Addresses for Low Bandwidth Networks

Chandana Gamage and Kanchana Kanchanasut

Technical Report No. TR-CS-95-001

March 1995

Division of Computer Science
School of Advanced Technologies
Asian Institute of Technology (AIT)
G.P.O Box 2754
Bangkok 10501
Thailand

IP ADDRESSES FOR LOW BANDWIDTH NETWORKS

Chandana Gamage and Kanchana Kanchanasut

Computer Science Program
School of Advanced Technologies
Asian Institute of Technology
P.O.Box 2754, Bangkok 10501
Thailand

E-mail:{a94255,kk}@cs.ait.ac.th

ABSTRACT

The Internet Protocol (IP) is based on a network architecture in which each host on the Internet is identified by a unique number called an IP address which in the present version of IP (IPv4) is 32 bits long. With the rapid global expansion of the Internet, it is projected that the present IP address space will soon be exhausted making IP numbers a scarce resource. The proposed new generation IP (IPv6) addresses will have 128 bits to accommodate future need for address space. IPv6, while having an IP number that is four times larger has an IP header double the size that of IPv4. Thus, this solution is certainly not an easy one to accommodate on a low bandwidth network. In this paper, we examine how a limited set of IP addresses can be shared among a number of hosts in a low bandwidth networking environment.

1 INTRODUCTION

To allow host computers to communicate with one another, the *Internet* relies on assigning a unique Internet Protocol (IP) address [10] for each host on the network. Beginning from late 80's we have witnessed a growing need for IP address space in the South East Asia (SEA) region. This can be largely attributed to the fast expanding economic activities and growth in the technological sector. This trend has been reinforced by the stated objectives of the regions countries to move out from mainly agro based economies and achieve Newly Industrialized Country (NIC) status.

Inspite of the remarkable achievements in infrastructure development in the SEA region, numerous obstacles prevent an overhaul of the regions communication infrastructure which is mainly owned and operated by stated owned agencies that lack funding for major upgrades of the external plant. Thus, there is every likelihood that we will continue to be burdened by a low bandwidth networking environment.

IP is the network layer communication mechanism for transfer of packets in a interconnected system of computers using packet switching technologies. IP transfers messages termed *datagrams* to fixed length addresses called IP addresses or IP numbers. IP provides*only* a source to destination delivery and has no mechanism to enhance its service with end–to–end reliability, flow control, sequencing or security. Such functionality is typically provided through upper layers of the communication stack which uses the basic host–to–host datagram transfer capability of IP layer.

In this paper we review early efforts in making effective use of IP address space, namely, the Classless Inter Domain Routing (CIDR) [5] and the use of Private Address Space [4] reserved by Internet Assigned Numbers Authority (IANA). To increase the address space, the Internet Engineering Task Force (IETF) has proposed the next generation IP (IPv6) [12] as a long term solution. We discuss the reasons for our preference to stay with the current IPv4 rather than move to proposed IPv6 and propose a model to increase the access to the Internet while limiting the number of IP addresses.

Our model for the near term should be one of *local power – global access*, where we satisfy our processing requirements in local IT centers but make ramps to the information superhighway to derive benefits for regions need for access to the worlds centers of research excellence. This will lead us to a situation where in each country there will be a number of small sized networks (class C) each having only a limited set of IP addresses to assign. Thus the network administrators will have a need for IP address reuse and dynamic management of address space.

In the next section, we review the IP addressing scheme, the IANA's private address space, CIDR and the IPv6 proposal. Then we describe our model to multiply a given IP number where its operational, efficiency and security issues are examined.

2 IP ADDRESSING

With the TCP/IP protocol stack [13], which is the communication protocol used by *Internet*, two important ideas were implemented:

1. Each host is uniquely identified by a network layer number called the IP address, for example, 192.41.170.15.

2. Applications were identified with a transport layer port number, for example, TCP NNTP number 119.

For host–to–host communications, the IP and port numbers were concatenated to form a socket, hence it is uniquely identified, for example, 192.41.170.15.119) is the stream–based NNTP port on host 192.41.170.15.

Although this gave a very efficient and streamlined implementation of TCP layer, it made the two TCP and IP layers to be tightly coupled rather than be truly independent.

Internet packet transfer involves three notions

1. A host name denoting the destination we want to reach
2. An IP address uniquely identifying the location of the host
3. A route consisting of intermediate gateways forming a path to the destination

IP layer deals only with addresses, it is the responsibility of higher layers to perform Name–Address mappings. While IP layer maps *Internet* addresses to local network addresses, the lower layers must map the local network addresses to routes for actual transfer of data.

IP addresses are of fixed length and in the current version of IP (IPv4) is 4–bytes (32–bits) long. There are three major address classes

Class A Format

0	1–7	8–31
0	network	local address

Class B Format

0	1	2–15	16–31
1	0	network	local address

Class C Format

0	1	2	3–23	24–31
1	1	0	network	local address

This class oriented addressing scheme is used to provide flexibility in assigning addresses to networks of varying sizes in a uniform manner.

It is possible for a single host machine to have multiple physical interfaces and for each interface to have several logical *Internet* addresses.

Although the current 32 bit IPv4 address structure can enumerate over 4 billion hosts on as many as 16.7 million networks (compare this with the projected 100,000 networks by 1996), the actual address assignment efficiency is far less than that. This inefficiency is exacerbated by the granularity of assignment using Class A, B and C addresses [11]. Using the current allocation statistics, the Address Lifetime Expectations (ALE) Working Group of IETF projected that the Internet would exhaust the IPv4 address space between 2005 and 2011. Efforts to select a successor to IPv4 was started in late 1990 primarily to overcome the limitations in the *Internet* address space.

2.1 EFFICIENT USE OF IP ADDRESSES

The *Internet* consists of thousands of networks consisting of millions of host computers[1]. Not every one of these hosts need full network layer connectivity to the larger internet. In an organizations only few hosts that provide server functions require

[1] As of 01 Oct 1994, *Internet* consisted of 3,898,233 hosts having unique IP addresses. *Source: Network Wizards, Menlo Park CA, USA.*

full IP level connectivity. By categorizing hosts into different classes depending on their connectivity requirements and thereafter allocating IP address based on such classification, we may be able to slow down the rate at which IP address space is used up. In this section, we review various methods which have been proposed to reduce the wasteful use of IP address.

1. **Private Address Space**
   The Internet Assigned Numbers Authority (IANA) has reserved the three sets of contiguous blocks of IP numbers for use by private networks. For example, 16–bit block: 192.168.0.0 - 192.168.255.255, a set of 256 contiguous address blocks can be used by private class C networks. These network IP addresses are unique only within an organization. They can be freely deployed within an organization without following any official allocation procedures of IANA or any *Internet* registry.

   Apart from the use of a reduced number of globally unique IP addresses, private IP address space allows network administrators to use a large block of IP addresses for configuring of their internal private hosts. This could be used to great advantage by doing appropriate subnetting which in turn improves routing efficiency, network trouble shooting and maintenance.

   Private hosts can communicate *only* with internal hosts which could either be private or public hosts. They cannot communicate with *any* external hosts. External services are typically provided through application layer relays (gateways).

2. **Classless Inter–Domain Routing**
   The class–based IP address block allocation scheme has a severe anomaly in IP number partitioning. In the allocation of IP address space, the Class C address blocks consisting of 254 nodes are too small for most of the requesting organizations whereas the Class B addresses giving 65534 nodes are far too large. Thus the allocation of Class B addresses to most organizations caused

   - The *allocatable* address space to decline rapidly while those organizations holding the address blocks had large numbers of unused IP addresses.
   - The Class B address space itself to rapidly dwindle.

   The CIDR mechanism was developed as a solution to two problems, to slow the growth of backbone routing tables and decrease the rate at which Class B addresses are used up.

   CIDR technique proposes that future IP addresses be allocated *topographically.* This is to be accomplished by allocating segments of the IP address space to transit routing domains (or service providers). Also, subnetting is to be done by variable length breakdown of IP address to *domain–host* parts at all levels of address assignment [6, 7].

3. **Next Generation IP Numbers**
   The *Internet Assigned Number Authority* (IANA) has assigned version number 6 to IPng and the protocol will be called IPv6. Important features of IPv6 includes expanded addressing and routing capabilities. The IP address size is increased from 32 bits to 128 bits providing support for a greater number of addressable nodes, more levels of addressing hierarchy and simpler auto–configuration of addresses [9, 8].

   With the increased address length, 24 more bytes in each IP datagram will be used for IP addressing thus seriously increasing the link overhead for data transmissions. This overhead leads to useless bandwidth consumption from the user point of view, particularly when you only have low bandwidth connections. Besides if IPv6 is going to be adopted worldwide, the transition period has to be carefully planned for.

   For these reasons, we would prefer to stay with the IPv4 scheme while trying to make effective use of the address space and finding ways to increase the addressing space without effecting the existing IPv4. This has led us to the question of how to make IP address sharable among many different hosts.

3 SHARABLE IP ADDRESS SCHEME (SIPAS)

Within any given organization, where many host machines are linked together, not all of them need to have access to the Internet at the same time. Some people may be doing local processing or require occasional access for a short period of time while others may want to occupy the connection over an extended period. While doing local processing users do not use the IP number, hence an IP connection can be viewed as a shared resource in a multi–user environment.
In SIPAS, We propose that an IP address can be shared among many hosts within the same organization so that we can reduce the growth of the overall IP address consumption. To share an IP address, we need to have a mechanism to dynamically allocating an IP number to a requesting host, as done using the Dynamic Host Configuration Protocol (DHCP) [2]. With such a protocol, we can multiply the address space without having to change the IP addressing scheme.

With dynamic IP allocation, the domain name and IP address binding is done dynamically as opposed to the existing static binding. We must maintain and make known this binding to the world to allow a host to use the IP number allocated to it as if it was assigned by the network manager statically.

At present, the binding of domain name and IP addresses are kept in the Domain Name System (DNS) servers of the domain in which the host belongs to [15]. The

DNS server maintains a table of such binding. Information within the responsibility of a DNS server could be cached by other hosts for efficiency reasons.

We propose to have a name server which keeps the most up–to–date bindings of fully qualified domain names (FQDN) and IP addresses within an organization. Hence, two new protocols are necessary, one for dynamic IP allocation (DIPA) and another for local name server (LNS) operation. We discuss our proposed change to the DNS to accommodate the proposed scheme, SIPAS, with **dynamic binding**. Within an organization, one IP number must be given to its DIPA and LNS server or this server must be accessible at all times. The following diagram demonstrate SIPAS architecture.

```
                  outside world

     -----------+-------------------------------------------
                |
IP Address: xxx |     Domain: yy.zz
     +----------+--------+
     | LNS      | Router |
     | Server   |        |
     +----------+        |
     | DIPA     |        |
     | Server   |        |
     +----------+--------+
                |
          --+---+---+---+---+---+--
      Host  |   |   |   |   |   |

            1   2       i
                        ^
                        Hi.yy.zz


          Figure 1: SIPAS
```

The server at `xxx` is known to the outside world as well as to the machines within the same organization. IP number allocation for machines within the organization is done by adopting IANA's private addressing scheme.

3.1 DYNAMIC NAME ADDRESS BINDING

With the introduction of dynamic binding of name and IP address, the DNS must be correspondingly updated. This could cause inconsistencies in the DNS distributed databases due to dynamic updates [15]. We propose that in order to cope with dynamic nature of the network, the DNS should also make use of dynamic data structure. An entry in the DNS server table for a host whose IP address is dynamically allocated, can have a pointer field indicating a secondary reference instead of an absolute IP address. Hence, an entry for `Hi.yy.zz` would look like

```
+------------+------------+
| FQDN       | IP Address |
+------------+------------+
| ro.yy.xx   | a.b.c.d    | <-- router for the sub-domain
| hi.yy.xx   | ^a.b.c.d   | <-- indirect address for a dynamic-host

Figure 2: Format of ANY DNS server
          (both zone-transferred and cached.
           ^ indicates indirect addressing)
```

```
+------------+------------+
| FQDN       | IP Address |
+------------+------------+
| ro.yy.xx   | a.b.c.d    | <-- router for the sub-domain
| hi.yy.xx   | a.b.e.f    | <-- IP address for a dynamic-host

Figure 3: Format for the LNS server
          (which now act as the primary server)
```

The entry can be cached by any DNS server without causing consistency problems. No updates to this entry in the DNS databases are required. The only place that needs to be updated is the table within the LNS server.

Our scheme has potential inter–operability problems in the presence of secondary DNS servers in the subnet.

1. We can not prevent information that should *only* be held by the LNS being transferred to secondary NS servers through a zone transfer.

   As a solution we propose to modify NS zone transfer code to prevent moving of mappings belonging to dynamically allocated hosts. Secondaries will still be consistent with LNS as far as the limited fixed IP number allocations and root–server mappings are concerned. And they will cache the indirect address bindings for all dynamically serviced hosts.

2. As the secondaries are *authoritative* NS servers, a resolve coming through a secondary for it's zone–of–authority will *not* do a recursive resolve attempt to reach the LNS for the complete mapping. It will return the indirect address.

   To solve this we propose modifications to NS server algorithm, such that they perform recursive resolving (even if the request is for it's zone–of–authority) for bindings held in it's tables that have indirect–IP–entries.

4 EXISTING TECHNOLOGIES

In this section, we examine existing technologies which could be applicable to our scheme.

DHCP allows hosts to setup their own network addresses without creating conflicts with other hosts already on a network while a DNS server allow hosts to publish their name–IP address bindings. In PC network environments, *Microsoft* offers DHCP functionality with a name server (NS) that operates under their Windows for Workgroups platform. Windows Internet Name Server (WINS) [16] coupled with DHCP allow organizational networks to grow, evolve and integrate to larger networks easily and quickly [1].

DHCP consists of two components: a protocol for delivering host–specific configuration parameters from a DHCP server to a host and a mechanism for allocation of network addresses to hosts. A DHCP server which runs on a router or a centralized machine maintains a collection of addresses and assigns them to users. Whenever a new host connects to the network it assigns an available address to it. Addresses obtained from the DHCP server are not owned by the host requesting it. DHCP server only *leases* IP addresses to requesting hosts with a set expiration time.

The network administrator can decide upon a suitable lease period depending upon the number of IP addresses available, number of users supported and dominant type of users in the subnet (fixed hosts, mobile hosts). Use of a centralized DHCP server removes the possibility of any conflicts occuring between hosts trying to use the same IP address.

SIPAS is similar to DHCP in that it requires the IP address to be allocated to different categories of hosts. DHCP technology can be adopted by SIPAS. However, to provide an efficient operation, additional features are necessary as discussed in the next section.

An NS allows hosts to look up other computers by their names rather than by IP addresses. This allows a machine to move from one subnet to another (thus changing the IP address) but to carry the same name as it originally had. For this NS layer tracks the movements of hosts and accordingly make changes in the listing information it maintains.

However WINS from *Microsoft* is still a local solution to providing a name space and reuse of IP addresses. This is because hosts outside of the subnet in which a particular machine resides can no longer get the IP address of that machine. This will prevent a machine that lists its name–IP binding in a WINS server being accessed by outside hosts for finger, ftp, telnet, etc. Only services such as mail which are routed through a gateway will be unaffected. This is due to the implicit assumption made by *Internet* that all hosts are fixed having static IP-addresses.

In SIPAS, this problem is resolved by demanding that DNS allows indirect addressing. But with the dynamic nature of name–address binding, LNS must allow for selective updating of its binding table. It has to provide record locking to prevent incorrect binding information from being used.

5 IMPLEMENTATION ISSUES

In this section, we examine how SIPAS could be implemented with respect to network efficiency and security issues.

1. **Efficiency**
   Since we need indirect addressing for a name–address lookup for any dynamic hosts, this could cause extra traffic for the the network. To avoid such problem, we should allow classifications of hosts within a given organization, such that some hosts can be assigned an IP address on permanent basis as allowed in DHCP [3].

   We can identify 3 distinct categories of hosts

   (a) Hosts that operate only within an organization and does not connect to outside.
   These nodes only require IP addresses that are unique within the organization. They could possibly be ambiguous outside of the organization.

   (b) Hosts that require only a limited set of external services such as E–mail, FTP, NNTP and Telnet.
   This set of services can be provided by application layer gateways. In this instance the hosts requiring such services will have IP addresses that are unique only within the organization and will connect to a gateway to obtain the required services. This gateway will have a unique IP address and will allow the services for internal hosts through an IP–name mapping facility. This is similar to the operation of firewall hosts[2].

   (c) Hosts requiring full IP level network connectivity.
   Such nodes must be assigned globally unique IP addresses for their correct operation.

[2]A firewall host is an application gateway that provides a proxy service by controlling traffic between a protected network and outside *Internet*.

We should also allow a host to always ask for a fixed address. Although this feature deviates from the dynamic concept, by utilizing this facility only for major servers (that provide FTP, Telnet like facilities) in a subnet that are accessed by machines from outside the network, while the network administrator can still have effective IP address reuse for a large number of machines in the subnet.

2. **Security**
   A potentially serious problem is security.

   (a) **Denial of service** - A rouge machine can impersonate a valid host that may be temporarily disconnected from the subnet by requesting a listing with an IP address obtained from a DHCP server and giving the name of the legal machine. This will cause a name registration refusal for the valid host if it tries to reconnect to the network later. Also, a malicious client host could issue `DHCPDECLINE` replies to the DHCP server, preventing the committing of an IP binding offer made to another host.

   (b) **Spoofing** - This is a type of forgery in which a rouge host can get an IP address from the subnet DHCP server and could potentially interfere with the activities of the legal (regular) users. Conversely, a host could masquerade as a DHCP server disrupting network activities. In this case an unauthorized DHCP server set up in a subnet could disrupt the network by allocating duplicate IP addresses, giving incorrect routing information, incorrect DNS addresses, etc.

   Use of broadcasting mechanisms to establish IP leases prevents the effective use of network layer authorization methods. Thus, it is difficult to prevent an unauthorized host coming on-line in a particular subnet.

   We recommend application-layer level authentication mechanisms such as **crypto-challenge** based on challenge/response exchanges, so that both the host and DHCP server can verify authenticity or legality of each other. Such authentication can be done prior to registration with NS server or as part of that process.

6 DISCUSSION

In this paper, we have proposed a model to expand the usage of IP addresses without extending IP address length. We propose that DNS be modified to use dynamic data structure mechanism to accommodate dynamic bindings. We discussed how to overcome the efficiency problem which could arise from introducing indirect DNS lookups. Our scheme as presented has one gateway consisting of a router and a server for DIPA and LNS, however we feel that an extension of the scheme to accommodate a network of DIPA and LNS servers could be achieved. For cooperation among LNS servers to optimize IP number use, a new protocol is required.

References

[1] Peter Wayner. *Automating TCP/IP in NT*. BYTE, November 1994.

[2] Ralph Droms. *Dynamic Host Configuration Protocol*. RFC[3] 1531, October 1993.

[3] Ralph Droms. *Interoperation between DHCP and BOOTP*. RFC 1534, October 1993.

[4] Yakov Rekhter, Robert G Moskowitz, Daniel Karrenberg and Geert Jan de Groot. *Address Allocation for Private Internets*. RFC 1597, March 1994.

[5] Robert Hinden. *Applicability Statement for the Implementation of Classless Inter–Domain Routing (CIDR)*. RFC 1517, September 1993.

[6] Yakov Rekhter and Tony Li. *An Architecture for IP Address Allocation with CIDR*. RFC 1518, September 1993.

[7] Vince Fuller, Tony Li, Jessica Yu and Kannan Varadhan. *Classless Inter–Domain Routing (CIDR): an Address Assignment and Aggregation Strategy*. RFC 1519, September 1993.

[8] Steve Deering. *Simple Internet Protocol Plus (SIPP) Specification*. Work in Progress, February 1994.

[9] Robert Hinden. *Simple Internet Protocol Plus White Paper*. RFC 1710, October 1994.

[10] Jon Postel. *Internet Protocol. DARPA Internet Program Protocol Specification*. RFC 791, September 1981.

[11] Scott Bradner and Allison Mankin. *The Recommendation for the IP Next Generation Protocol*. RFC 1752, January 1995.

[12] Richard Carlson and Domenic Ficarella. *Six Virtual Inches to the Left: IPng Problems*. RFC 1705, October 1994.

[13] Jon Postel. *Transmission Control Protocol*. RFC 793, September 1981.

[14] Richard Stevens. *UNIX Network Programming*. Prentice–Hall International Inc., 1991.

[15] Susan Thomson, Yakov Rekhter and Jim Bound. *Extending the Domain Name System (DNS) to Perform Dynamic Updates*. Work in Progress, January 1995.

[16] Microsoft Windows NT Server. *Reviewers Guide*. June 1994.

[3] Request for Comments are working notes of the *Internet* research and development community.